

UNIVERSIDADE DE SÃO PAULO
FACULDADE DE ECONOMIA, ADMINISTRAÇÃO E CONTABILIDADE
DEPARTAMENTO DE ADMINISTRAÇÃO

**APLICAÇÃO DE REDES NEURAS ARTIFICIAIS NA ANÁLISE DE SÉRIES
TEMPORAIS ECONÔMICO-FINANCEIRAS**

Mauri Aparecido de Oliveira
Orientador: Prof. Dr. Fernando Carvalho de Almeida

SÃO PAULO
2007

Prof^a. Dr^a. Suely Vilela

Reitora da Universidade de São Paulo

Prof. Dr. Carlos Roberto Azzoni

Diretor da Faculdade de Economia, Administração e Contabilidade

Prof. Dr. Isak Kruglianskas

Chefe do Departamento de Administração

Prof. Dr. Lindolfo Galvão de Albuquerque

Coordenador do Programa de Pós-Graduação em Administração

MAURI APARECIDO DE OLIVEIRA

**APLICAÇÃO DE REDES NEURAS ARTIFICIAIS NA ANÁLISE DE SÉRIES
TEMPORAIS ECONÔMICO-FINANCEIRAS**

Tese apresentada ao Departamento de
Administração da Faculdade de Economia,
Administração e Contabilidade da
Universidade de São Paulo, para a obtenção
do título de Doutor em Administração

Orientador: Prof. Dr. Fernando Carvalho de Almeida

SÃO PAULO

2007

FICHA CATALOGRÁFICA

Elaborada pela Seção de Processamento Técnico do SBD/FEA/USP

Oliveira, Mauri Aparecido de

Aplicação de redes neurais artificiais na análise de séries temporais econômico-financeiras / Mauri Aparecido de Oliveira. – São Paulo, 2007.

317 p.

Tese (Doutorado) – Universidade de São Paulo, 2007

Bibliografia

1. Econometria 2. Análise de séries temporais 3. Redes neurais 4. Previsão (Análise de séries temporais) I. Universidade de São Paulo. Faculdade de Economia, Administração e Contabilidade. II. Título.

CDD – 330.015195

Para Mauro Aparecido de Oliveira

Agradecimentos

Ao Prof. Dr. Fernando Carvalho de Almeida, meu orientador, pelo apoio e dedicada orientação durante o doutorado e durante a elaboração desse trabalho.

À Prof.^a Dr.^a Alessandra de Ávila Montini, as reuniões técnicas que tivemos foram muito produtivas e estimulantes.

À Prof.^a Dr.^a Solange Oliveira Rezende do ICMC-USP/São Carlos pelo incentivo desde o curso de graduação e pela minha iniciação científica que tanto contribuiu para minha formação acadêmica e profissional.

Ao Prof. Dr. Luiz Paulo Lopes Fávero, que desde o início do programa de pós-graduação possibilitou discussões construtivas sobre a aplicação de métodos quantitativos em ciências sociais aplicadas.

Aos meus pais, aos meus irmãos, e ao meu sobrinho Kepler Silveira pelo carinho e incentivo.

A todos os professores da FEA-USP pelos seus ensinamentos.

Às pessoas que trabalham na FEA-USP (Biblioteca, Secretaria, UPD, etc.) e a tornam um ambiente profícuo para os alunos de pós-graduação.

Aos colegas de pós-graduação, e em especial ao Daniel Reed Bergmann que é um excelente econometrista e teve paciência de escutar muitas das minhas teorias.

À Kérya Sayuri Shiguemori Oliveira, por ser a filha maravilhosa que me aceitou em seu mundo quando o meu não funcionava bem.

À Cristina Mayumi que durante estes anos cuidou de nossa família, sem o seu amor e encorajamento eu não teria conseguido continuar meu trabalho.

À Senhora Fumiko Shiguemori pelo auxílio incondicional durante todas as etapas de elaboração desse trabalho.

APLICAÇÃO DE REDES NEURAS ARTIFICIAIS NA ANÁLISE DE SÉRIES TEMPORAIS ECONÔMICO-FINANCEIRAS

Resumo

Diversas metodologias são empregadas para realizar a análise de séries temporais, dentre as quais destaca-se o uso das redes neurais artificiais (RNA). Neste trabalho são utilizados quatro métodos para realizar previsão de séries temporais univariadas: os modelos ARIMA-GARCH, RNA *feedforward*, RNA treinada com filtro de Kalman estendido (EKF) e RNA treinada com o filtro de Kalman *unscented* (UKF). Sendo que o uso de RNA-UKF é um avanço recente na área de sistemas de inteligência computacional. O uso de redes neurais treinadas com filtro de Kalman é uma metodologia que tem trazido bons resultados em uma ampla variedade de aplicações nas áreas comercial, militar e científica. Em 2002 aproximadamente 250 bilhões de dólares eram gerenciados em fundos de investimentos por modelos quantitativos (tais como lógica fuzzy, redes neurais, algoritmos genéticos, fractais e modelos de Markov). Desde 2006 estima-se que três em cada dez destes fundos utilizem estes modelos quantitativos. A capacidade das RNA em lidar com não linearidades é uma vantagem normalmente destacada quando são realizadas previsões de séries temporais. São apresentadas simulações de Monte Carlo que mostram a influência dos parâmetros dos modelos ARIMA-GARCH na predição de redes neurais artificiais do tipo *feedforward*, treinadas com o algoritmo de Levenberg-Marquardt. Pelos resultados obtidos verificou-se que a RNA *feedforward* realizou melhores previsões a medida que o parâmetro ligado a estacionariedade aumenta. Também é aplicada a teoria para construção de intervalos de predição (IP) e de confiança (IC) para RNA *feedforward*. As séries temporais analisadas são univariadas e compostas de dados reais do setor financeiro (Bradesco PN, Bradespar PN, Itausa PN e Itaú PN), setor de alimentos (Perdigão PN, Sadia PN, Saca da Soja de 60Kg e Saca de Açúcar de 50Kg), setor industrial (Marcopolo PN, Petrobrás PN, Embraer ON, Ripasa PN, Souza Cruz ON e Gerdau PN) e setor de serviços (Pão de Açúcar PN, Eletropaulo PNA, Eletrobras PNB, Brasil Telecom PN, Cesp PNA e Lojas Americanas PNA). Os resultados obtidos mostram que a RNA-UKF apresentou-se superior quando comparada com as técnicas concorrentes.

APLICAÇÃO DE REDES NEURAS ARTIFICIAIS NA ANÁLISE DE SÉRIES TEMPORAIS ECONÔMICO-FINANCEIRAS

Abstract

Many technologies has been applied to time series analysis, among these artificial neural networks (RNA). In this work, four methods are used to univariate time series forecasting: ARIMA-GARCH, RNA feedforward, RNA trained using extended Kalman filter (EKF) and RNA trained using unscented Kalman filter (UKF). RNA-UKF is a recent method in computational intelligence field. The use of neural networks trained using Kalman filter is a methodology that has brought good results in a wide variety of applications such as commercial, military and scientific field. In 2002 approximately 250 billions of dollars were managed in investment funds by quantitative models (such as fuzzy logic, neural networks, genetic algorithms, fractals and Markov models). Since 2006 it is estimated that three in ten investment funds use these quantitative models. The RNA power to deal with non linearities is a highlited advantage when time series forecasting are performed. This work presents Monte Carlo simulations showing the ARIMA-GARCH parameters influence in the feedforward artificial neural networks predictions, trained with Levenberg-Marquardt algorithm. According to the results, RNA feedforward performed best forecasts to the extent stacionarity parameter increase. Moreover, the theory for confidence (IC) e prediction (IP) intervals are applied to RNA feedforward.

This work presents analysis to real data univariate time series from financial sector (Bradesco PN, Bradespar PN, Itausa PN and Itaú PN), food sector (Perdigão PN, Sadia PN, Soybean 60Kg and Sugar 50Kg), factory sector (Marcopolo PN, Petrobrás PN, Embraer ON, Ripasa PN, Souza Cruz ON and Gerdau PN) and service sector (Pão de Açúcar PN, Eletropaulo PNA, Eletrobras PNB, Brasil Telecom PN, Cesp PNA and Lojas Americanas PNA). The results showed RNA-UKF upper hand when compared with the competitors techniques.

Abreviações e Símbolos

Abreviações

ADF	Augmented Dickey Fuller - Dickey Fuller Aumentado
AIC	Akaike Information Criteria – Critério de Informação de Akaike
AR	Auto-regressivo
ARCH	Autoregressive Conditional Heterocedasticity – Auto-regressivo com heterocedasticidade condicional
ARIMA	Auto-regressivo Integrado de Média Móvel
ARMA	Auto-regressivo de Média Móvel
BIC	Bayesian Information Criteria – Critério de Informação Bayesiano
SBC	Schwartz Bayesian Criteria– Critério de Informação de Schwartz
BP	Back Propagation - Retropropagação
BPTT	Back Propagation Through Time - Retropropagação Através do Tempo
CORR	Coefficiente de Correlação de Pearson
EKF	Extended Kalman Filter - Filtro de Kalman Estendido
FAC	Função de Auto-correlação
FACP	Função de Auto-correlação Parcial
FIR	Finite Impulse Response Filter – Filtro de Resposta a Impulso de Duração Finita
FK	Filtro de Kalman
FM	Frequência Modulada
GARCH	Generalized Autoregressive Conditional Heterocedasticity – Auto-regressivo com Heterocedasticidade Condicional Generalizada
IA	Inteligência Artificial
IC	Intervalo de Confiança
IP	Intervalo de Predição
LM	Levenberg- Marquardt
MA	Moving Average - Média Móvel
MAE	Mean Absolute Error - Erro Absoluto Médio
MAPE	Mean Absolute Percentage Error - Erro Absoluto Médio Percentual
MLP	Multi-Layer Perceptron - Perceptron de Camadas Múltiplas
MRNA	Modelos de Redes Neurais Artificiais
MQO	Mínimos Quadrados Ordinários
MSE	Mean Squared Error - Erro Quadrático Médio
MSPE	Mean Squared Prediction Error - Erro Quadrático Médio de Predição
NID	Normal e Identicamente Distribuído
RB	Ruído Branco
RMSE	Root Mean Square Error – Raiz do Erro Quadrático Médio
RMSPE	Root Mean Squared Prediction Error – Raiz do Erro Quadrático Médio de Predição
RNA-EKF	Rede Neural Artificial Treinada com Filtro de Kalman Estendido
RNA-UKF	Rede Neural Artificial Treinada com Filtro de Kalman Unscented

RSPE	Root Squared Prediction Error – Raiz do Erro Quadrado de Predição
RTRL	Real Time Recurrent Learning - Aprendizado Recorrente em Tempo Real
SIC	Schwarz Information Criteria – Critério de Informação de Schwarz
SPE	Square prediction error – Erro Quadrado de Predição
SQR	Soma de Quadrado dos Resíduos
SSE	Sum of Squared Errors – Soma Quadrática dos Erros
TDNN	Time Delay Neural Network – Rede Neural com Atraso de Tempo
TIC	Coefficiente de Desigualdade de Theil
TLFN	Time Lagged Feedforward Network – Rede Alimentada Adiante Atrasada no Tempo
UKF	Filtro de Kalman Unscented
USPTO	United States Patent and Trademark Office

Símbolos Importantes - Em negrito significa que são vetores

\mathbf{r}_t	Ruído de Medida no Tempo t
\mathbf{R}_t	Matriz de Covariância do Ruído de Medida
\mathbf{q}_t	Processo-ruído
\mathbf{Q}_t	Matriz de Covariância do Processo Ruído
\mathbf{y}_t	Vetor de Medida
\mathbf{x}_t	Vetor de Estado
$\hat{\mathbf{x}}_{t t}$	Estimativa de Estado a Posterior na Teoria de Kalman
$\hat{\mathbf{x}}_{t t-1}$	Estimativa de Estado a Priori
$\mathbf{P}_{t t}$	Matriz de Correlação Estado-erro a Posterior
$\mathbf{P}_{t t-1}$	Matriz de Correlação Estado-erro a Priori
\mathbf{K}_t	Ganho de Kalman
\mathbf{H}_t	Matriz de Medida
\mathbf{F}_t	Matriz de Transição de Estado
$h_t(\bullet)$	Função Vetor de Medida Não linear
$f_t(\bullet)$	Função Vetor Estado Não linear
\mathbf{H}_n	Jacobiana – Matriz de Medida
$\mathbf{F}_{t+1,t}$	Jacobiana – Matriz Transição de Estado
\mathbf{I}	Matriz Identidade
t_d	Profundidade de Truncamento
p_p	Parâmetro que Inicializa $\mathbf{P}_{t t}$
p_r	Parâmetro Taxa de Aprendizagem que Inicializa \mathbf{R}_t
t_i	Número Total de Unidades na Rede

t_o	Número de Unidades de Saída na Rede
t_i	Número de Unidades de Entrada na Rede
t_w	Número Total de Pesos Sinápticos na Rede
\mathbf{u}_t	Vetor de Entrada da Rede Neural
$\phi(\bullet)$	Função de Ativação
\mathbf{w}_t	Vetor de Pesos Sinápticos Ordenados
w_i	Pesos Individuais
s_t	t-ésima Amostra de Série Temporal
τ	Atraso

Sumário

Agradecimentos	iv
Resumo	v
Abstract	vi
Abreviações e Símbolos	vii
Lista de Tabelas	xv
Lista de Figuras	xix

1 Introdução

1.1 Motivação	1
1.2 Objetivos da Pesquisa e Considerações Iniciais	3
1.3 Delineamento do Trabalho	7

2 Aplicação Econométrica de Séries Temporais

2.1 Séries Temporais.....	8
2.1.1 Processos Estocásticos.....	9
2.1.2 Modelos Estacionários.....	10
2.1.3 Modelos Auto-regressivos de Médias Móveis (ARMA).....	11
2.1.4 Modelos Auto-regressivos (AR).....	12
2.1.5 Modelos de Médias Móveis (MA).....	13
2.1.6 Modelos Não-estacionários.....	13
2.1.7 Heterocedasticidade Condicional.....	14
2.1.8 Modelos ARCH.....	15
2.1.9 Modelos GARCH.....	17
2.2 Modelos ARIMA: Metodologia de Box e Jenkins.....	18
2.2.1 Medidas para Avaliar Previsão.....	20

3 Redes Neurais Artificiais

3.1 Características das Redes Neurais.....	22
3.2 Motivação para utilizar Redes Neurais na Previsão.....	25
3.2.1 Redes Neurais e Estatística.....	27
3.3 Modelo de um Neurônio Artificial.....	28
3.4 Algoritmo de Retropropagação (<i>Backpropagation</i>).....	30
3.5 Redes Neurais Alimentadas Adiante (<i>Feedforward</i>).....	34
3.5.1 Incorporação do Tempo na Rede Neural.....	36
3.5.2 Algoritmo de Retropropagação Temporal.....	41

3.6	Intervalos de Predição para Redes Neurais Artificiais.....	46
3.7	Teoria de Intervalo de Predição.....	48
3.8	A influência dos parâmetros de modelos ARIMA-GARCH na predição de redes neurais feedforward.....	52
3.9	Rede Neural e Função de Ativação.....	54
3.10	Algoritmo de Levenberg-Marquardt	56
3.11	Comportamento da Função de Ativação para processos AR-GARCH.....	58
3.12	Resultados de predições de séries AR-GARCH para diferentes topologias de RNA.....	63
4	Filtro de Kalman	
4.1	Filtro de Kalman.....	74
4.2	Filtro de Kalman Estendido.....	78
4.3	Aplicação do EKF a Redes Neurais.....	79
4.4	Filtro de Kalman <i>Unscented</i>	82
4.4.1	Representações de Modelos de Espaço de Estado.....	87
4.5	Teste para não linearidade.....	90
5	Metodologia	
5.1	Metodologia de Pesquisa.....	92
5.2	Cálculo e Obtenção dos Resultados.....	97
6	Resultados Experimentais	
6.1	Normalização dos Valores de Entrada da RNA.....	98
6.2	Setor Financeiro – Bradesco, Itaú, Bradespar e Itausa.....	99
6.2.1	Análise da série do Bradesco.....	99
6.2.1.1	Análise da série do Bradesco – Método ARIMA-GARCH.....	99
6.2.1.2	Análise da série do Bradesco – Método RNA <i>feedforward</i>	103
6.2.1.3	Análise da série do Bradesco – Método RNA EKF.....	105
6.2.1.4	Análise da série do Bradesco – Método RNA UKF.....	106
6.2.2	Análise da série do Itaú.....	108
6.2.2.1	Análise da série do Itaú – Método ARIMA-GARCH.....	108
6.2.2.2	Análise da série do Itaú – Método RNA <i>feedforward</i>	111
6.2.2.3	Análise da série do Itaú – Método RNA EKF.....	113
6.2.2.4	Análise da série do Itaú – Método RNA UKF.....	114

6.2.3	Análise da série da Bradespar.....	115
6.2.3.1	Análise da série da Bradespar – Método ARIMA-GARCH.....	116
6.2.3.2	Análise da série da Bradespar – Método RNA <i>feedforward</i>	119
6.2.3.3	Análise da série da Bradespar – Método RNA EKF.....	121
6.2.3.4	Análise da série da Bradespar – Método RNA UKF.....	122
6.2.4	Análise da série da Itausa.....	124
6.2.4.1	Análise da série da Itausa – Método ARIMA-GARCH.....	124
6.2.4.2	Análise da série da Itausa – Método RNA <i>feedforward</i>	127
6.2.4.3	Análise da série da Itausa – Método RNA EKF.....	129
6.2.4.4	Análise da série da Itausa – Método RNA UKF.....	130
6.3	Setor de Alimentos – Perdigão, Sadia, Soja e Açúcar.....	132
6.3.1	Análise da série da Perdigão.....	132
6.3.1.1	Análise da série da Perdigão – Método ARIMA-GARCH.....	132
6.3.1.2	Análise da série da Perdigão – Método RNA <i>feedforward</i>	135
6.3.1.3	Análise da série da Perdigão – Método RNA EKF.....	137
6.3.1.4	Análise da série da Perdigão – Método RNA UKF.....	138
6.3.2	Análise da série da Sadia.....	140
6.3.2.1	Análise da série da Sadia – Método ARIMA-GARCH.....	140
6.3.2.2	Análise da série da Sadia – Método RNA <i>feedforward</i>	143
6.3.2.3	Análise da série da Sadia – Método RNA EKF.....	145
6.3.2.4	Análise da série da Sadia – Método RNA UKF.....	146
6.3.3	Análise da série da Soja.....	148
6.3.3.1	Análise da série da Soja – Método ARIMA-GARCH.....	148
6.3.3.2	Análise da série da Soja – Método RNA <i>feedforward</i>	151
6.3.3.3	Análise da série da Soja – Método RNA EKF.....	153
6.3.3.4	Análise da série da Soja – Método RNA UKF.....	154
6.3.4	Análise da série do Açúcar.....	156
6.3.4.1	Análise da série do Açúcar – Método ARIMA-GARCH.....	156
6.3.4.2	Análise da série do Açúcar – Método RNA <i>feedforward</i>	159
6.3.4.3	Análise da série do Açúcar – Método RNA EKF.....	161
6.3.4.4	Análise da série do Açúcar – Método RNA UKF.....	162
6.4	Setor Industrial – Marcopolo, Petrobrás, Embraer, Ripasa, Souza Cruz e Gerdau.....	164
6.4.1	Análise da série da Marcopolo.....	164
6.4.1.1	Análise da série da Marcopolo – Método ARIMA-GARCH.....	164
6.4.1.2	Análise da série da Marcopolo – Método RNA <i>feedforward</i>	167
6.4.1.3	Análise da série da Marcopolo – Método RNA EKF.....	169
6.4.1.4	Análise da série da Marcopolo – Método RNA UKF.....	170

6.4.2	Análise da série da Petrobras.....	172
6.4.2.1	Análise da série da Petrobras– Método ARIMA-GARCH.....	172
6.4.2.2	Análise da série da Petrobras– Método RNA <i>feedforward</i>	175
6.4.2.3	Análise da série da Petrobras– Método RNA EKF.....	177
6.4.2.4	Análise da série da Petrobras– Método RNA UKF.....	178
6.4.3	Análise da série da Embraer.....	180
6.4.3.1	Análise da série da Embraer – Método ARIMA-GARCH.....	180
6.4.3.2	Análise da série da Embraer – Método RNA <i>feedforward</i>	183
6.4.3.3	Análise da série da Embraer – Método RNA EKF.....	185
6.4.3.4	Análise da série da Embraer – Método RNA UKF.....	186
6.4.4	Análise da série da Ripasa.....	188
6.4.4.1	Análise da série da Ripasa – Método ARIMA-GARCH.....	188
6.4.4.2	Análise da série da Ripasa – Método RNA <i>feedforward</i>	191
6.4.4.3	Análise da série da Ripasa – Método RNA EKF.....	193
6.4.4.4	Análise da série da Ripasa – Método RNA UKF.....	194
6.4.5	Análise da série da Souza Cruz.....	196
6.4.5.1	Análise da série da Souza Cruz – Método ARIMA-GARCH.....	196
6.4.5.2	Análise da série da Souza Cruz – Método RNA <i>feedforward</i>	199
6.4.5.3	Análise da série da Souza Cruz – Método RNA EKF.....	201
6.4.5.4	Análise da série da Souza Cruz – Método RNA UKF.....	202
6.4.6	Análise da série da Gerdau.....	204
6.4.6.1	Análise da série da Gerdau – Método ARIMA-GARCH.....	204
6.4.6.2	Análise da série da Gerdau – Método RNA <i>feedforward</i>	207
6.4.6.3	Análise da série da Gerdau – Método RNA EKF.....	209
6.4.6.4	Análise da série da Gerdau – Método RNA UKF.....	210
6.5	Setor de Serviços – Pão de Açúcar, Eletropaulo, Eletrobras, Brasil Telecom, Cesp e Lojas Americanas.....	212
6.5.1	Análise da série do Pão de Açúcar.....	212
6.5.1.1	Análise da série do Pão de Açúcar – Método ARIMA-GARCH.....	212
6.5.1.2	Análise da série do Pão de Açúcar – Método RNA <i>feedforward</i>	215
6.5.1.3	Análise da série do Pão de Açúcar – Método RNA EKF.....	217
6.5.1.4	Análise da série do Pão de Açúcar – Método RNA UKF.....	218
6.5.2	Análise da série da Eletropaulo.....	220
6.5.2.1	Análise da série da Eletropaulo – Método ARIMA-GARCH.....	220
6.5.2.2	Análise da série da Eletropaulo – Método RNA <i>feedforward</i>	223
6.5.2.3	Análise da série da Eletropaulo – Método RNA EKF.....	225
6.5.2.4	Análise da série da Eletropaulo – Método RNA UKF.....	226

6.5.3	Análise da série da Eletrobrás.....	228
6.5.3.1	Análise da série da Eletrobras – Método ARIMA-GARCH.....	228
6.5.3.2	Análise da série da Eletrobras – Método RNA <i>feedforward</i>	231
6.5.3.3	Análise da série da Eletrobras – Método RNA EKF.....	233
6.5.3.4	Análise da série da Eletrobras – Método RNA UKF.....	234
6.5.4	Análise da série da Brasil Telecom.....	236
6.5.4.1	Análise da série da Brasil Telecom – Método ARIMA-GARCH.....	236
6.5.4.2	Análise da série da Brasil Telecom – Método RNA <i>feedforward</i>	239
6.5.4.3	Análise da série da Brasil Telecom – Método RNA EKF.....	241
6.5.4.4	Análise da série da Brasil Telecom – Método RNA UKF.....	242
6.5.5	Análise da série da Cesp.....	244
6.5.5.1	Análise da série da Cesp– Método ARIMA-GARCH.....	244
6.5.5.2	Análise da série da Cesp – Método RNA <i>feedforward</i>	247
6.5.5.3	Análise da série da Cesp – Método RNA EKF.....	249
6.5.5.4	Análise da série da Cesp – Método RNA UKF.....	250
6.5.6	Análise da série da Lojas Americanas.....	252
6.5.6.1	Análise da série da Lojas Americanas– Método ARIMA-GARCH.....	252
6.5.6.2	Análise da série da Lojas Americanas – Método RNA <i>feedforward</i>	255
6.5.6.3	Análise da série da Lojas Americanas – Método RNA EKF.....	257
6.5.6.4	Análise da série da Lojas Americanas – Método RNA UKF.....	258
6.6	Síntese dos Resultados Obtidos.....	260
7	Conclusões.....	262
8	Referências Bibliográficas	265
	Apêndice.....	276
	Apêndice-A.....	276
	Apêndice-B.....	277
	Apêndice-C.....	284

Lista de Tabelas

3.1	Terminologias usadas em modelos estatísticos e em redes neurais.....	28
3.2	Resumo do algoritmo de retropropagação temporal para uma rede alimentada adiante.....	46
3.3	Percentis do TIC obtidos pela RNA 10-1-1 e 10-9-1.....	72
4.1	Equações do filtro de Kalman.....	76
4.2	Equações do filtro de Kalman estendido.....	79
4.3	Equações do filtro de Kalman unscented.....	86
6.1	Teste ADF para série original do Bradesco.....	100
6.2	Teste ADF para série original do NDLBRAD.....	100
6.3	Resultados obtidos pelo teste BDS nas respectivas dimensões.....	101
6.4	Parâmetros estimados para a série NDLBRAD.....	101
6.5	Estatísticas para diferentes topologias de RNA para a série.....	103
6.6	Estatísticas de previsão para o método de ARIMA-GARCH e de RNA....	107
6.7	Teste ADF para série original do Itaú.....	108
6.8	Teste ADF para série original do NDLITAU.....	108
6.9	Resultados obtidos pelo teste BDS nas respectivas dimensões.....	109
6.10	Parâmetros estimados para a série NDLITAU.....	109
6.11	Estatísticas para diferentes topologias de RNA para a série.....	111
6.12	Estatísticas de previsão para o método de ARIMA-GARCH e de RNA....	115
6.13	Teste ADF para série original da Bradespar.....	116
6.14	Teste ADF para série original do NDLBPAR.....	116
6.15	Resultados obtidos pelo teste BDS nas respectivas dimensões.....	117
6.16	Parâmetros estimados para a série NDLBPAR.....	117
6.17	Estatísticas para diferentes topologias de RNA para a série.....	119
6.18	Estatísticas de previsão para o método de ARIMA-GARCH e de RNA....	123
6.19	Teste ADF para série original da Itausa.....	124
6.20	Teste ADF para série original do NDLIUSA.....	124
6.21	Resultados obtidos pelo teste BDS nas respectivas dimensões.....	125
6.22	Parâmetros estimados para a série NDLIUSA.....	125
6.23	Estatísticas para diferentes topologias de RNA para a série.....	127
6.24	Estatísticas de previsão para o método de ARIMA-GARCH e de RNA....	131
6.25	Teste ADF para série original da Perdigão.....	132
6.26	Teste ADF para série original do NDLPERD.....	132
6.27	Resultados obtidos pelo teste BDS nas respectivas dimensões.....	133
6.28	Parâmetros estimados para a série NDLPERD.....	133
6.29	Estatísticas para diferentes topologias de RNA para a série.....	135
6.30	Estatísticas de previsão para o método de ARIMA-GARCH e de RNA....	139

6.31	Teste ADF para série original da Sadia.....	140
6.32	Teste ADF para série original do NDLSADI.....	140
6.33	Resultados obtidos pelo teste BDS nas respectivas dimensões.....	141
6.34	Parâmetros estimados para a série NDLSADI.....	141
6.35	Estatísticas para diferentes topologias de RNA para a série.....	143
6.36	Estatísticas de previsão para o método de ARIMA-GARCH e de RNA....	147
6.37	Teste ADF para série original da Soja.....	148
6.38	Teste ADF para série original do NDLSOJA.....	148
6.39	Resultados obtidos pelo teste BDS nas respectivas dimensões.....	149
6.40	Parâmetros estimados para a série NDLSOJA.....	149
6.41	Estatísticas para diferentes topologias de RNA para a série.....	151
6.42	Estatísticas de previsão para o método de ARIMA-GARCH e de RNA....	155
6.43	Teste ADF para série original do Açúcar.....	156
6.44	Teste ADF para série original do NDLCANA.....	156
6.45	Resultados obtidos pelo teste BDS nas respectivas dimensões.....	157
6.46	Parâmetros estimados para a série NDLCANA.....	157
6.47	Estatísticas para diferentes topologias de RNA para a série.....	159
6.48	Estatísticas de previsão para o método de ARIMA-GARCH e de RNA....	163
6.49	Teste ADF para série original da Marcopolo.....	164
6.50	Teste ADF para série original do NDLMARC.....	164
6.51	Resultados obtidos pelo teste BDS nas respectivas dimensões.....	165
6.52	Parâmetros estimados para a série NDLMARC.....	165
6.53	Estatísticas para diferentes topologias de RNA para a série.....	167
6.54	Estatísticas de previsão para o método de ARIMA-GARCH e de RNA....	171
6.55	Teste ADF para série original da Petrobras.....	172
6.56	Teste ADF para série original do NDLPETR.....	172
6.57	Resultados obtidos pelo teste BDS nas respectivas dimensões.....	173
6.58	Parâmetros estimados para a série NDLPETR.....	173
6.59	Estatísticas para diferentes topologias de RNA para a série.....	175
6.60	Estatísticas de previsão para o método de ARIMA-GARCH e de RNA....	179
6.61	Teste ADF para série original da Embraer.....	180
6.62	Teste ADF para série original do NDLEMBR.....	180
6.63	Resultados obtidos pelo teste BDS nas respectivas dimensões.....	181
6.64	Parâmetros estimados para a série NDLEMBR.....	181
6.65	Estatísticas para diferentes topologias de RNA para a série.....	183
6.66	Estatísticas de previsão para o método de ARIMA-GARCH e de RNA....	187
6.67	Teste ADF para série original da Ripasa.....	188
6.68	Teste ADF para série original do NDLRIPA.....	188
6.69	Resultados obtidos pelo teste BDS nas respectivas dimensões.....	189

6.70	Parâmetros estimados para a série NDLRIPA.....	189
6.71	Estatísticas para diferentes topologias de RNA para a série.....	191
6.72	Estatísticas de previsão para o método de ARIMA-GARCH e de RNA....	195
6.73	Teste ADF para série original da Souza Cruz.....	196
6.74	Teste ADF para série original do NDLSOUZ.....	196
6.75	Resultados obtidos pelo teste BDS nas respectivas dimensões.....	197
6.76	Parâmetros estimados para a série NDLSOUZ.....	197
6.77	Estatísticas para diferentes topologias de RNA para a série.....	199
6.78	Estatísticas de previsão para o método de ARIMA-GARCH e de RNA....	203
6.79	Teste ADF para série original da Gerdau.....	204
6.80	Teste ADF para série original do NDLGERD.....	204
6.81	Resultados obtidos pelo teste BDS nas respectivas dimensões.....	205
6.82	Parâmetros estimados para a série NDLGERD.....	205
6.83	Estatísticas para diferentes topologias de RNA para a série.....	207
6.84	Estatísticas de previsão para o método de ARIMA-GARCH e de RNA....	210
6.85	Teste ADF para série original do Pão de Açúcar.....	212
6.86	Teste ADF para série original do NDLPAOD.....	213
6.87	Resultados obtidos pelo teste BDS nas respectivas dimensões.....	213
6.88	Parâmetros estimados para a série NDLPAOD.....	213
6.89	Estatísticas para diferentes topologias de RNA para a série.....	215
6.90	Estatísticas de previsão para o método de ARIMA-GARCH e de RNA....	219
6.91	Teste ADF para série original da Eletropaulo.....	220
6.92	Teste ADF para série original do NDLELET.....	220
6.93	Resultados obtidos pelo teste BDS nas respectivas dimensões.....	221
6.94	Parâmetros estimados para a série NDLELET.....	221
6.95	Estatísticas para diferentes topologias de RNA para a série.....	223
6.96	Estatísticas de previsão para o método de ARIMA-GARCH e de RNA....	227
6.97	Teste ADF para série original da Eletrobrás.....	228
6.98	Teste ADF para série original do NDLELBR.....	228
6.99	Resultados obtidos pelo teste BDS nas respectivas dimensões.....	229
6.100	Parâmetros estimados para a série NDLELBR.....	229
6.101	Estatísticas para diferentes topologias de RNA para a série.....	231
6.102	Estatísticas de previsão para o método de ARIMA-GARCH e de RNA....	235
6.103	Teste ADF para série original da Brasil Telecom.....	236
6.104	Teste ADF para série original do NDLBRAS.....	236
6.105	Resultados obtidos pelo teste BDS nas respectivas dimensões.....	237
6.106	Parâmetros estimados para a série NDLBRAS.....	237
6.107	Estatísticas para diferentes topologias de RNA para a série.....	239
6.108	Estatísticas de previsão para o método de ARIMA-GARCH e de RNA....	243

6.109	Teste ADF para série original da Cesp.....	244
6.110	Teste ADF para série original do NDLCESP.....	244
6.111	Resultados obtidos pelo teste BDS nas respectivas dimensões.....	245
6.112	Parâmetros estimados para a série NDLCESP.....	245
6.113	Estatísticas para diferentes topologias de RNA para a série.....	247
6.114	Estatísticas de previsão para o método de ARIMA-GARCH e de RNA.....	251
6.115	Teste ADF para série original da Lojas Americanas.....	252
6.116	Teste ADF para série original do NDLLLOJA.....	252
6.117	Resultados obtidos pelo teste BDS nas respectivas dimensões.....	253
6.118	Parâmetros estimados para a série NDLLLOJA.....	253
6.119	Estatísticas para diferentes topologias de RNA para a série.....	255
6.120	Estatísticas de previsão para o método de ARIMA-GARCH e de RNA.....	259

Lista de Figuras

1.1	Procedimento para modelagem de processos ARIMA (Jhee e Lee, 1993).	5
1.2	(a) Arquitetura de rede alimentada adiante (<i>feedforward</i>) (b) rede recorrente.....	6
3.1	Modelo de um neurônio artificial (Haykin, 2001).....	29
3.2	Rede <i>Feedforward</i> – Alimentada Adiante.....	35
3.3	Exemplo de uma rede TDNN - <i>Time Delay Neural Network</i>	37
3.4	(a) Filtro Neural Focado, (b) Rede TLFN – <i>Time Lagged Feedforward Network</i>	38
3.5	Filtro FIR aplicado à uma rede neural.....	39
3.6	Representação do somatório da Equação (3.46).....	43
3.7	Derivadas $\partial v/\partial y$	44
3.8	Rede neural alimentada adiante de três camadas.....	47
3.9	Modelo de uma rede neural.....	55
3.10	Apresenta oito funções de ativação: (a) seno, (b) cosseno, (c) tangente hiperbólica, (d) $y/(1+ y)$, (e) arco tangente, (f) linear por partes, (g) sigmóide e (h) erf. Todas as figuras apresentam ganhos de 0,25, 0,5, 1 e 2.....	56
3.11	Curva da derivada da função de ativação, ganho igual a 1.....	59
3.12	Histogramas de $\varphi(a_0)$ e $\varphi'(a_0)$. α_1 e β_1 variam no intervalo [0,1;0,2] e ϕ_1 varia nos intervalos: (a) [0,1;0,2], (b) [0,3;0,4], (c) [0,5;0,6] e (d) [0,7;0,8].....	62
3.13	Histogramas de $\varphi(a_0)$ e $\varphi'(a_0)$. ϕ_1 e β_1 variam no intervalo [0,1;0,2] e α_1 varia nos intervalos: (a) [0,3;0,4], (b) [0,5;0,6] e (c) [0,7;0,8].....	63
3.14	Histogramas de $\varphi(a_0)$ e $\varphi'(a_0)$. ϕ_1 e α_1 variam no intervalo [0,1;0,2] e β_1 varia nos intervalos: (a) [0,3;0,4], (b) [0,5;0,6] e (c) [0,7;0,8].....	64
3.15	Árvore de combinação de parâmetros para simulação das séries AR[1]-GARCH[1,1].....	65
3.16	(a) Box-plot do TIC para a RNA 10-1-1 considerando aumento gradativo de ϕ_1 , (b) TIC de acordo com o aumento de ϕ_1 para a RNA 10-1-1.....	66
3.17	(a) Box-plot do TIC para a RNA 10-1-1 considerando ϕ_1 variando no intervalo [0,1;0,2] e várias combinações de α_1 e β_1 , (b) ϕ_1 varia no intervalo [0,3;0,4], (c) ϕ_1 varia no intervalo [0,5;0,6], (d) ϕ_1 varia no intervalo [0,7;0,8].....	68
3.18	(a) Box-plot do TIC considerando aumento gradativo de α_1 , (b) TIC com o aumento de α_1	68
3.19	(a) Box-plot do TIC para a RNA 10-1-1 considerando aumento gradativo da volatilidade β_1 , (b) variação do TIC de acordo com o aumento de β_1 para a RNA 10-1-1.....	69

3.20	(a) Box-plot do TIC para a RNA 10-3-1, (b) Box-plot do TIC para a RNA 10-5-1, (c) Box-plot do TIC para a RNA 10-7-1, (d) Box-plot do TIC para a RNA 10-9-1.....	70
3.21	a) Box-plot do TIC para a RNA 10-9-1 considerando aumento gradativo de ϕ , (b) variação do TIC de acordo com o aumento de ϕ_1 para a RNA 10-9-1.....	71
3.22	Quatro séries AR(1)-GARCH(1,1) geradas por simulação e previsões (linhas rosas) versus valores reais (linhas azuis).....	73
4.1	Estrutura recursiva preditor-corretor do filtro de Kalman.....	84
5.1	Fluxograma para treinar a RNA utilizando o filtro de Kalman.....	96
6.1	(a) Série no nível do preço do Bradesco, (b) Série transformada NDLBRAD.....	99
6.2	(a) Correlograma da série NDLBRAD, (b) Estatísticas dos resíduos.....	100
6.3	(a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado.....	102
6.4	Previsão obtida pelo modelo ARIMA-GARCH para NDLBRAD.....	102
6.5	(a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLBRAD.....	103
6.6	Previsão obtida pela RNA <i>feedforward</i> para NDLBRAD.....	104
6.7	(a) Intervalo de confiança, (b) Intervalo de predição para NDLBRAD.....	104
6.8	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA <i>feedforward</i> para NDLBRAD.....	105
6.9	Previsão obtida pela RNA EKF para NDLBRAD.....	105
6.10	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLBRAD.....	106
6.11	Previsão obtida pela RNA UKF para NDLBRAD.....	106
6.12	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLBRAD.....	107
6.13	Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA.....	107
6.14	(a) Série no nível do preço do Itaú, (b) Série transformada NDLITAU....	108
6.15	(a) Correlograma da série NDLITAU, (b) Estatísticas dos resíduos.....	109
6.16	(a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado.....	110
6.17	Previsão obtida pelo modelo ARIMA-GARCH para NDLITAU.....	110
6.18	(a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLBITAU.....	111
6.19	Previsão obtida pela RNA <i>feedforward</i> para NDLITAU.....	112
6.20	(a) Intervalo de confiança, (b) Intervalo de predição para NDLITAU.....	112
6.21	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA <i>feedforward</i> para NDLITAU.....	113
6.22	Previsão obtida pela RNA EKF para NDLITAU.....	113
6.23	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLITAU.....	114
6.24	Previsão obtida pela RNA UKF para NDLITAU.....	114

6.25	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLITAU.....	115
6.26	Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA	115
6.27	(a) Série no nível do preço da Bradespar, (b) Série transformada NDLBPAR.....	116
6.28	(a) Correlograma da série NDLBPAR, (b) Estatísticas dos resíduos.....	117
6.29	(a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado.....	118
6.30	Previsão obtida pelo modelo ARIMA-GARCH para NDLBPAR.....	118
6.31	(a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLBPAR.....	119
6.32	Previsão obtida pela RNA <i>feedforward</i> para NDLBPAR.....	120
6.33	(a) Intervalo de confiança, (b) Intervalo de predição para NDLBPAR.....	120
6.34	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA <i>feedforward</i> para NDLBPAR.....	121
6.35	Previsão obtida pela RNA EKF para NDLBPAR.....	121
6.36	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLBPAR.....	122
6.37	Previsão obtida pela RNA UKF para NDLBPAR.....	122
6.38	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLBPAR.....	124
6.39	Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA	124
6.40	(a) Série no nível do preço da Itausa, (b) Série transformada NDLIUSA.....	124
6.41	(a) Correlograma da série NDLIUSA, (b) Estatísticas dos resíduos.....	125
6.42	(a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado.....	126
6.43	Previsão obtida pelo modelo ARIMA-GARCH para NDLIUSA.....	126
6.44	(a) Normal Q-Q Plot, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLIUSA.....	127
6.45	Previsão obtida pela RNA <i>feedforward</i> para NDLIUSA.....	128
6.46	(a) Intervalo de confiança, (b) Intervalo de predição para NDLIUSA.....	128
6.47	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA <i>feedforward</i> para NDLBIUSA.....	129
6.48	Previsão obtida pela RNA EKF para NDLIUSA.....	129
6.49	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLIUSA.....	130
6.50	Previsão obtida pela RNA UKF para NDLIUSA.....	130
6.51	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLIUSA.....	131
6.52	Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA	131

6.53	(a) Série no nível do preço da Perdigão, (b) Série transformada NDLPERD.....	132
6.54	(a) Correlograma da série NDLPERD, (b) Estatísticas dos resíduos.....	133
6.55	(a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado.....	134
6.56	Previsão obtida pelo modelo ARIMA-GARCH para NDLPERD.....	134
6.57	(a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLPERD.....	135
6.58	Previsão obtida pela RNA <i>feedforward</i> para NDLPERD.....	136
6.59	(a) Intervalo de confiança, (b) Intervalo de predição para NDLPERD.....	136
6.60	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA <i>feedforward</i> para NDLPERD.....	137
6.61	Previsão obtida pela RNA EKF para NDLPERD.....	137
6.62	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLPERD.....	138
6.63	Previsão obtida pela RNA UKF para NDLPERD.....	138
6.64	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLPERD.....	139
6.65	Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA.....	139
6.66	(a) Série no nível do preço da Sadia, (b) Série transformada NDLSADI.....	140
6.67	(a) Correlograma da série NDLSADI, (b) Estatísticas dos resíduos.....	141
6.68	(a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado.....	142
6.69	Previsão obtida pelo modelo ARIMA-GARCH para NDLSADI.....	142
6.70	(a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLSADI.....	143
6.71	Previsão obtida pela RNA <i>feedforward</i> para NDLSADI.....	144
6.72	(a) Intervalo de confiança, (b) Intervalo de predição para NDLSADI.....	144
6.73	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA <i>feedforward</i> para NDLSADI.....	145
6.74	Previsão obtida pela RNA EKF para NDLSADI.....	145
6.75	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLSADI.....	146
6.76	Previsão obtida pela RNA UKF para NDLSADI.....	146
6.77	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLSADI.....	147
6.78	Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA.....	147
6.79	(a) Série no nível do preço da Soja, (b) Série transformada NDLSOJA.....	148
6.80	(a) Correlograma da série NDLSOJA, (b) Estatísticas dos resíduos.....	149

6.81	(a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado.....	150
6.82	Previsão obtida pelo modelo ARIMA-GARCH para NDLSOJA.....	150
6.83	(a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLSOJA.....	151
6.84	Previsão obtida pela RNA <i>feedforward</i> para NDLSOJA.....	152
6.85	(a) Intervalo de confiança, (b) Intervalo de predição para NDLSOJA.....	152
6.86	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA <i>feedforward</i> para NDLSOJA.....	153
6.87	Previsão obtida pela RNA EKF para NDLSOJA.....	153
6.88	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLSOJA.....	154
6.89	Previsão obtida pela RNA UKF para NDLSOJA.....	154
6.90	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLSOJA.....	155
6.91	Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA.....	155
6.92	(a) Série no nível do preço do Açúcar, (b) Série transformada NDLCANA.....	156
6.93	(a) Correlograma da série NDLCANA, (b) Estatísticas dos resíduos.....	157
6.94	(a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado.....	158
6.95	Previsão obtida pelo modelo ARIMA-GARCH para NDLCANA.....	158
6.96	(a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLCANA.....	159
6.97	Previsão obtida pela RNA <i>feedforward</i> para NDLCANA.....	160
6.98	(a) Intervalo de confiança, (b) Intervalo de predição para NDLCANA.....	160
6.99	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA <i>feedforward</i> para NDLCANA.....	161
6.100	Previsão obtida pela RNA EKF para NDLCANA.....	161
6.101	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLCANA.....	162
6.102	Previsão obtida pela RNA UKF para NDLCANA.....	162
6.103	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLCANA.....	163
6.104	Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b) RMSE.....	163
6.105	(a) Série no nível do preço da Marcopolo, (b) Série transformada NDLMARC.....	164
6.106	(a) Correlograma da série NDLMARC, (b) Estatísticas dos resíduos.....	165
6.107	(a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado.....	166
6.108	Previsão obtida pelo modelo ARIMA-GARCH para NDLMARC.....	166

6.109	(a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLMARC.....	167
6.110	Previsão obtida pela RNA <i>feedforward</i> para NDLMARC.....	168
6.111	(a) Intervalo de confiança, (b) Intervalo de predição para NDLMARC.....	168
6.112	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da predição da RNA <i>feedforward</i> para NDLMARC.....	169
6.113	Previsão obtida pela RNA EKF para NDLMARC.....	169
6.114	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da predição da RNA EKF para NDLMARC.....	170
6.115	Previsão obtida pela RNA UKF para NDLMARC.....	170
6.116	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da predição da RNA UKF para NDLMARC.....	171
6.117	Estatísticas de predição obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b) RMSE.....	171
6.118	(a) Série no nível do preço da Petrobras, (b) Série transformada NDLPETR.....	172
6.119	(a) Correlograma da série NDLPETR, (b) Estatísticas dos resíduos.....	173
6.120	(a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado.....	174
6.121	Previsão obtida pelo modelo ARIMA-GARCH para NDLPETR.....	174
6.122	(a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLPETR.....	175
6.123	Previsão obtida pela RNA <i>feedforward</i> para NDLPETR.....	176
6.124	(a) Intervalo de confiança, (b) Intervalo de predição para NDLPETR.....	176
6.125	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da predição da RNA <i>feedforward</i> para NDLPETR.....	177
6.126	Previsão obtida pela RNA EKF para NDLPETR.....	177
6.127	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da predição da RNA EKF para NDLPETR.....	178
6.128	Previsão obtida pela RNA UKF para NDLPETR.....	178
6.129	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da predição da RNA UKF para NDLPETR.....	179
6.130	Estatísticas de predição obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b) RMSE.....	179
6.131	(a) Série no nível do preço da Embraer, (b) Série transformada NDLEMBR.....	180
6.132	(a) Correlograma da série NDLEMBR, (b) Estatísticas dos resíduos.....	181
6.133	(a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado.....	182
6.134	Previsão obtida pelo modelo ARIMA-GARCH para NDLEMBR.....	182
6.135	(a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLEMBR.....	183
6.136	Previsão obtida pela RNA <i>feedforward</i> para NDLEMBR.....	184

6.137	(a) Intervalo de confiança, (b) Intervalo de predição para NDLEMBR.....	184
6.138	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA <i>feedforward</i> para NDLEMBR.....	185
6.139	Previsão obtida pela RNA EKF para NDLEMBR.....	185
6.140	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLEMBR.....	186
6.141	Previsão obtida pela RNA UKF para NDLEMBR.....	186
6.142	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLEMBR.....	187
6.143	Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b) RMSE.....	187
6.144	(a) Série no nível do preço da Ripasa, (b) Série transformada NDLRIPA.....	188
6.145	(a) Correlograma da série NDLRIPA, (b) Estatísticas dos resíduos.....	189
6.146	(a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado.....	190
6.147	Previsão obtida pelo modelo ARIMA-GARCH para NDLRIPA.....	190
6.148	(a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLRIPA.....	191
6.149	Previsão obtida pela RNA <i>feedforward</i> para NDLRIPA.....	192
6.150	(a) Intervalo de confiança, (b) Intervalo de predição para NDLRIPA.....	192
6.151	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA <i>feedforward</i> para NDLRIPA.....	193
6.152	Previsão obtida pela RNA EKF para NDLRIPA.....	193
6.153	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLRIPA.....	194
6.154	Previsão obtida pela RNA UKF para NDLRIPA.....	194
6.155	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLRIPA.....	195
6.156	Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b) RMSE.....	195
6.157	(a) Série no nível do preço da Souza Cruz, (b) Série transformada NDLSOUZ.....	196
6.158	(a) Correlograma da série NDLSOUZ, (b) Estatísticas dos resíduos.....	197
6.159	(a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado.....	198
6.160	Previsão obtida pelo modelo ARIMA-GARCH para NDLSOUZ.....	198
6.161	(a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLSOUZ.....	199
6.162	Previsão obtida pela RNA <i>feedforward</i> para NDLSOUZ.....	200
6.163	(a) Intervalo de confiança, (b) Intervalo de predição para NDLSOUZ.....	200
6.164	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA <i>feedforward</i> para NDLSOUZ.....	201
6.165	Previsão obtida pela RNA EKF para NDLSOUZ.....	201

6.166	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLSOUZ.....	202
6.167	Previsão obtida pela RNA UKF para NDLSOUZ.....	202
6.168	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLSOUZ.....	203
6.169	Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b) RMSE.....	203
6.170	(a) Série no nível do preço da Gerdau, (b) Série transformada NDLGERD.....	204
6.171	(a) Correlograma da série NDLGERD, (b) Estatísticas dos resíduos.....	205
6.172	(a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado.....	206
6.173	Previsão obtida pelo modelo ARIMA-GARCH para NDLGERD.....	206
6.174	(a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLGERD.....	207
6.175	Previsão obtida pela RNA <i>feedforward</i> para NDLGERD.....	208
6.176	(a) Intervalo de confiança, (b) Intervalo de predição para NDLGERD.....	208
6.177	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA <i>feedforward</i> para NDLGERD.....	209
6.178	Previsão obtida pela RNA EKF para NDLGERD.....	209
6.179	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLGERD.....	210
6.180	Previsão obtida pela RNA UKF para NDLGERD.....	210
6.181	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLGERD.....	211
6.182	Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b) RMSE.....	211
6.183	(a) Série no nível do preço do Pão de Açúcar, (b) Série transformada NDLPAD.....	212
6.184	(a) Correlograma da série NDLPAD, (b) Estatísticas dos resíduos.....	213
6.185	(a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado.....	214
6.186	Previsão obtida pelo modelo ARIMA-GARCH para NDLPAD.....	214
6.187	(a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLPAD.....	215
6.188	Previsão obtida pela RNA <i>feedforward</i> para NDLPAD.....	216
6.189	(a) Intervalo de confiança, (b) Intervalo de predição para NDLPAD.....	216
6.190	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA <i>feedforward</i> para NDLPAD.....	217
6.191	Previsão obtida pela RNA EKF para NDLPAD.....	217
6.192	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLPAD.....	218
6.193	Previsão obtida pela RNA UKF para NDLPAD.....	218

6.194	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLP AOD.....	219
6.195	Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b) RMSE.....	219
6.196	(a) Série no nível do preço da Eletropaulo, (b) Série transformada NDLELET.....	220
6.197	(a) Correlograma da série NDLELET, (b) Estatísticas dos resíduos.....	221
6.198	(a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado.....	222
6.199	Previsão obtida pelo modelo ARIMA-GARCH para NDLELET.....	222
6.200	(a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLELET.....	223
6.201	Previsão obtida pela RNA <i>feedforward</i> para NDLELET.....	224
6.202	(a) Intervalo de confiança, (b) Intervalo de predição para NDLELET.....	224
6.203	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA <i>feedforward</i> para NDLELET.....	225
6.204	Previsão obtida pela RNA EKF para NDLELET.....	225
6.205	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLELET.....	226
6.206	Previsão obtida pela RNA UKF para NDLELET.....	226
6.207	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLELET.....	227
6.208	Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b) RMSE.....	227
6.209	(a) Série no nível do preço da Eletrobras, (b) Série transformada NDLELBR.....	228
6.210	(a) Correlograma da série NDLELBR, (b) Estatísticas dos resíduos.....	229
6.211	(a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado.....	230
6.212	Previsão obtida pelo modelo ARIMA-GARCH para NDLELBR.....	230
6.213	(a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLELBR.....	231
6.214	Previsão obtida pela RNA <i>feedforward</i> para NDLELBR.....	232
6.215	(a) Intervalo de confiança, (b) Intervalo de predição para NDLELBR.....	232
6.216	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA <i>feedforward</i> para NDLELBR.....	233
6.217	Previsão obtida pela RNA EKF para NDLELBR.....	233
6.218	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLELBR.....	234
6.219	Previsão obtida pela RNA UKF para NDLELBR.....	234
6.220	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLELBR.....	235
6.221	Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b) RMSE.....	235

6.222	(a) Série no nível do preço da Brasil Telecom, (b) Série transformada NDLBRAS.....	236
6.223	(a) Correlograma da série NDLBRAS, (b) Estatísticas dos resíduos.....	237
6.224	(a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado.....	238
6.225	Previsão obtida pelo modelo ARIMA-GARCH para NDLBRAS.....	238
6.226	(a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLBRAS.....	239
6.227	Previsão obtida pela RNA <i>feedforward</i> para NDLBRAS.....	240
6.228	(a) Intervalo de confiança, (b) Intervalo de predição para NDLBRAS.....	240
6.229	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA <i>feedforward</i> para NDLBRAS.....	241
6.230	Previsão obtida pela RNA EKF para NDLBRAS.....	241
6.231	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLBRAS.....	242
6.232	Previsão obtida pela RNA UKF para NDLBRAS.....	242
6.233	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLBRAS.....	243
6.234	Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b) RMSE.....	243
6.235	(a) Série no nível do preço da Cesp, (b) Série transformada NDLCESP.....	244
6.236	(a) Correlograma da série NDLCESP, (b) Estatísticas dos resíduos.....	245
6.237	(a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado.....	246
6.238	Previsão obtida pelo modelo ARIMA-GARCH para NDLCESP.....	246
6.239	(a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLCESP.....	247
6.240	Previsão obtida pela RNA <i>feedforward</i> para NDLCESP.....	248
6.241	(a) Intervalo de confiança, (b) Intervalo de predição para NDLCESP.....	248
6.242	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA <i>feedforward</i> para NDLCESP.....	249
6.243	Previsão obtida pela RNA EKF para NDLCESP.....	249
6.244	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLCESP.....	250
6.245	Previsão obtida pela RNA UKF para NDLCESP.....	250
6.246	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLCESP.....	251
6.247	Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b) RMSE.....	251
6.248	(a) Série no nível do preço da Lojas Americanas, (b) Série transformada NDLLLOJA.....	252
6.249	(a) Correlograma da série NDLLLOJA, (b) Estatísticas dos resíduos.....	253
6.250	(a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado.....	254

6.251	Previsão obtida pelo modelo ARIMA-GARCH para NDLLLOJA.....	254
6.252	(a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLLLOJA.....	255
6.253	Previsão obtida pela RNA <i>feedforward</i> para NDLLLOJA.....	256
6.254	(a) Intervalo de confiança, (b) Intervalo de predição para NDLLLOJA.....	256
6.255	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA <i>feedforward</i> para NDLLLOJA.....	257
6.256	Previsão obtida pela RNA EKF para NDLLLOJA.....	257
6.257	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLLLOJA.....	258
6.258	Previsão obtida pela RNA UKF para NDLLLOJA.....	258
6.259	(a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLLLOJA.....	259
6.260	Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b) RMSE.....	259
6.261	Resultados das estatísticas TIC para cada uma das vinte séries analisadas.	260
6.262	Resultados das estatísticas RMSE para cada uma das vinte séries analisadas.....	261

Capítulo 1

Introdução

1.1 Motivação

Modelar sistemas do mundo real é uma tarefa de alta demanda e complexidade. Geralmente, objetos mal definidos por ruídos nos dados, medidas imprecisas e existência de não linearidades, descrevem os sistemas do mundo real (OLIVEIRA et al., 2007e, 2007f, 2007h). Técnicas de modelagem convencionais como modelos estatísticos (ARIMA-GARCH) e aqueles baseados em modelos matemáticos teóricos têm produzido previsões insatisfatórias. As suas desvantagens poderiam ser sumarizadas pelo alto grau de subjetividade com respeito a estrutura do modelo e sua base de informação limitada. Na área de finanças podem ser encontradas críticas sobre o uso de modelos ARIMA-GARCH e redes neurais para explicar os mercados, como por exemplo a dificuldade desses modelos em lidar com eventos extremos (TALEB, 1997, p. 102). Porém, a estrutura não linear e a habilidade da rede neural de estender o número de argumentos de entrada fazem dela um procedimento de modelagem adequado para muitos problemas em finanças (OLIVEIRA e FÁVERO, 2003). As redes neurais têm sido utilizadas com sucesso em uma ampla variedade de aplicações de sistemas de modelagem financeiro, provando em muitos casos sua superioridade sobre os modelos estatísticos ARIMA-GARCH (MEDEIROS et al., 2006, FELDKAMP et al., 2001, SCARBOROUGH e SOMERS, 2006).

Considerando o fato de que métodos padrões para computar intervalos de predição em regressões não lineares podem ser efetivamente aplicados às redes neurais quando o número de valores de treinamento é grande. É apresentada aqui a teoria para construção de intervalos de predição (IP) e de confiança (IC) para redes neurais artificiais (RNA) alimentadas adiante de múltiplas camadas baseada em regressão multivariada e destaca-se

sua importância em aplicações econométricas. Com essa abordagem busca-se criar uma ligação entre os resultados fornecidos pela rede neural e a teoria estatística de intervalos de predição.

O filtro de Kalman estendido (*extended* Kalman filter – EKF) é uma técnica de estimação para sistemas não lineares. Pode-se acelerar o processo de aprendizagem da RNA explorando a teoria do filtro de Kalman.

A análise de séries temporais é uma das áreas-chave em ciências sociais aplicadas (DELGADO et al., 2006, MEDEIROS et al., 2006, VAN DIJK, 2006, TERASVIRTA et al., 2005, GHIASSI et al., 2005, KIANI, 2005, JASIC, 2004).

Dadas algumas observações passadas do comportamento de alguns sistemas, surge a questão de como fazer previsões sobre o comportamento futuro e quão precisas essas previsões podem ser (TERASVIRTA et al., 2005). As condições presentes determinam em algum grau o futuro de tal forma que pode haver muitas interações e complexas relações entre as variáveis envolvidas. Tais processos ocorrem em muitos dos ramos da ciência, sendo que alguns exemplos incluem meteorologia, oceanografia, epidemiologia, economia e finanças (MORETTIN, 2006, LIMA, 2004, ALMEIDA, 2004).

Existindo informação disponível suficiente sobre o sistema em estudo, uma abordagem estatística-matemática pode ser utilizada, por meio de equações construídas com o intuito de modelar os mecanismos responsáveis pela geração das séries temporais. No entanto, em muitos problemas reais não tem-se informações e/ou condições ideais suficientes para construir equações que governem o comportamento das variáveis que se quer prever. Na ausência de informações suficientes para gerar as equações é mais atrativo usar uma abordagem baseada em redes neurais.

Uma abordagem baseada em redes neurais artificiais é um exemplo de abordagem baseada em modelos, na qual uma rede é treinada para abstrair as equações geradoras de uma série temporal. Portanto, ao invés de explicitamente deduzir equações que descrevam o comportamento subjacente do sistema em estudo, um capturador automático de um modelo não linear usando um algoritmo de aprendizagem neural é construído para aproximar as equações ideais. As redes neurais são atrativas para modelar sistemas não lineares porque elas são inerentemente não lineares, devido as funções de ativação não lineares, e também são bem estabelecidas como funções de aproximação (FORSGREN e KLING, 2003). Tem

sido mostrado que uma rede neural com um número arbitrário de unidades (ou neurônios) e possuindo duas ou mais camadas pode aproximar qualquer função uniformemente contínua (WAN e MERWE, 2001a).

Entre as muitas técnicas que desempenharam um papel importante no desenvolvimento teórico dos métodos para previsão destaca-se a metodologia de Box e Jenkins (BOX, JENKINS e REINSEL, 1994). Esta metodologia foi desenvolvida no começo da década de 70 para possibilitar a construção de modelos auto-regressivos integrados de média móvel (ARIMA) que uma vez estimados são utilizados para realizar previsão. São utilizados os modelos ARIMA como ponto inicial para a análise das séries temporais porque muitos modelos neurais para processamento temporal são construídos a partir de variações de modelos ARIMA.

Diversos algoritmos que utilizam redes neurais para processamento temporal têm sido propostos, entre estes destacam-se o algoritmo de retropropagação através do tempo (BPTT, *backpropagation through time*), aprendizado recorrente em tempo real (RTRL, *real-time recurrent learning*) e os algoritmos de redes recorrentes utilizando o filtro de Kalman (WILLIAMS, 1992; WILLIAMS e ZIPSER, 1995).

Atualmente existem cerca de 10130 patentes que utilizam redes neurais, registradas no escritório de patentes e marca registrada dos Estados Unidos (USPTO – *United States Patent and Trademark Office*). As diversas aplicações que podem ser feitas com redes neurais fazem com que esta seja uma área de estudo e aplicação das mais dinâmicas.

1.2 Objetivos da Pesquisa e Considerações Iniciais

O objetivo deste trabalho é mostrar que RNA treinadas com o filtro de Kalman *unscented* (UKF) realizam melhores previsões para séries temporais univariadas econômico-financeiras quando comparadas com os modelos auto-regressivos de médias móveis (ARIMA), RNA que utilizam o filtro de Kalman estendido (EKF) e as RNA *feedforward*. Também é aplicado um método para construção de intervalos de confiança (IC) e de predição (IP). Esse método além de melhorar a análise econométrica pode ser usado como ferramenta para determinar a melhor topologia da RNA (OLIVEIRA et al., 2007d). As

redes neurais são técnicas de aprendizagem no sentido de serem capazes de extrair características relevantes de grandes conjuntos de dados e isto é útil em aplicações envolvendo sistemas complexos que devem ser analisados baseando-se em conhecimentos prévios.

Embora muitas técnicas de regressão não linear estejam sendo desenvolvidas, as propriedades de não linearidade das redes neurais fazem com que estas sejam preferidas para previsão. Colocando as redes neurais no contexto dos métodos estatísticos tradicionais, elas podem ser consideradas como técnica de estimação não paramétrica (ou seja, não necessitam de presunção da distribuição de probabilidade), não linear e multivariada (indica que a entrada e a saída da rede neural pode conter duas ou mais variáveis distintas) (OLIVEIRA et al., 2007b).

Um importante aspecto das redes neurais é a sua capacidade de construir relações não lineares entre os dados de entrada e as saídas desejadas (OLIVEIRA e SIQUEIRA, 2004). A linearidade (por exemplo, $y_t = \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \dots$ sendo que y_t é uma série temporal e os α_i são parâmetros) apenas descreve a relação mais simples que pode ocorrer em uma série temporal. As redes neurais são ferramentas de análise que apresentam características desejáveis para análise de dados e possuem um esquema de implementação relativamente eficiente em termos de velocidade de computação e memória necessária para processamento (OLIVEIRA et al., 2007c, 2007g).

De acordo com Lippman (1987), estes modelos fornecem um alto grau de robustez e tolerância à falha. Em adição a estas características, os MRNA (modelos de redes neurais artificiais) podem encontrar as transformações corretas para as variáveis, detectar relações lineares fracas e trabalhar com valores que estejam discrepantes das demais observações (*outliers*).

O procedimento para o modelagem ARIMA normalmente é realizado em quatro etapas distintas, como mostrado na Figura 1.1 (JHEE e LEE, 1993), conhecidas como identificação, estimação, diagnóstico e análise de resíduos.

A modelagem ARIMA é uma ferramenta gerencial para previsão amplamente utilizado como método de referência na criação de modelos lineares de dados temporais, principalmente por causa de sua elaboração teórica e precisão nas previsões de curto prazo (OLIVEIRA et al., 2007a). Os modelos ARIMA são construídos através de passos

iterativos, como mostrados na Figura 1.1. Entre as etapas de modelagem, a identificação do modelo que determina a ordem p do processo AR e a ordem q do processo MA é crucial para representar a série temporal adequadamente. Podem ser estimados vários modelos concorrentes para representar a série temporal, porém deve-se escolher o modelo estimado com menor quantidade de parâmetros para não contrariar o princípio da parcimônia de Box e Jenkins. Portanto, é importante o julgamento do especialista que realiza o modelagem na interpretação de algumas informações estatísticas tal como as funções de autocorrelação (FAC) e autocorrelação parcial (FACP).

Alguns sistemas para realizar a tarefa de previsão são construídos de maneira a suprimir o julgamento do especialista. Isso é feito a partir do uso de uma rede neural que atua como um classificador de padrões na etapa de identificação (JHEE e LEE, 1993).

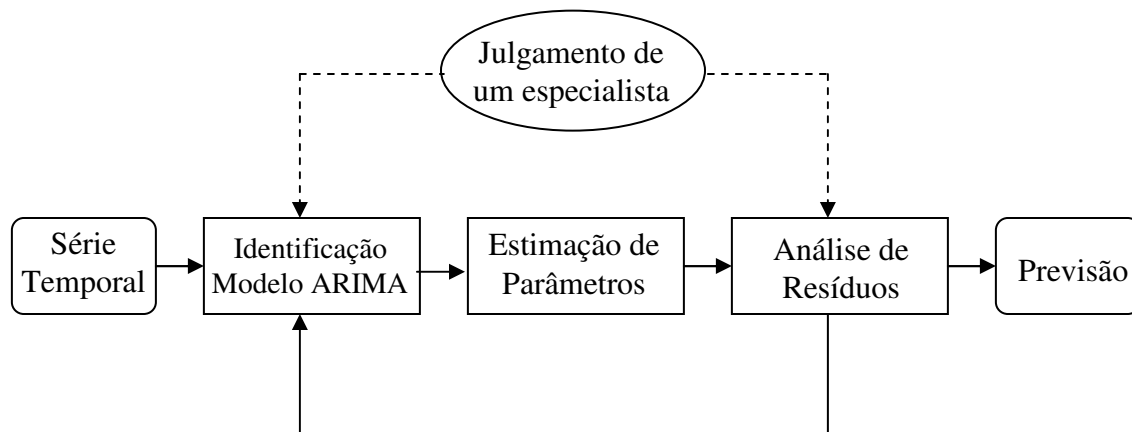


Figura 1.1 Procedimento para modelagem de processos ARIMA.

No caso das redes neurais tem-se o interesse em estudar quais as arquiteturas de rede que devem ser utilizadas para o processamento temporal e principalmente como derivar os algoritmos de aprendizagem necessários para realizar o processo de treinamento. A forma mais simples de fazer com que uma rede neural realize processamento temporal por meio da adição de memória à estrutura da rede. Estas memórias podem ser de longo prazo ou de curto prazo. A memória é inserida na rede através de atrasos de tempo que podem ser implementados no nível sináptico dentro da rede ou na camada de entrada da rede. Uma rede com memória de curto prazo muito utilizada é rede neural de atrasos de tempo

(TDNN, *time delay neural network*). As redes com memórias de longo prazo mais conhecidas são as *redes recorrentes* de Elman(1990) e de Jordan(1986).

Os algoritmos de treinamento de redes neurais recorrentes mais utilizados são o algoritmo de aprendizagem em tempo real (RTRL) e o de retropropagação através do tempo (BPTT). No entanto, muitos autores (JULIER e UHLMAN, 2002, WAN e MERWE, 2001b, DREYFUS, 2005) têm notado que um algoritmo utilizando variantes do filtro de Kalman (EKF, *extended Kalman filter* e UKF *unscented Kalman filter*) pode ser usado para treinar redes neurais com a vantagem de possuir uma velocidade de convergência superior e baixa tendência de ficar fixado num mínimo local quando comparado com os outros métodos baseados no método do gradiente descendente. A Figura 1.2 mostra as arquiteturas de redes neurais artificiais mais utilizadas para processamento temporal.

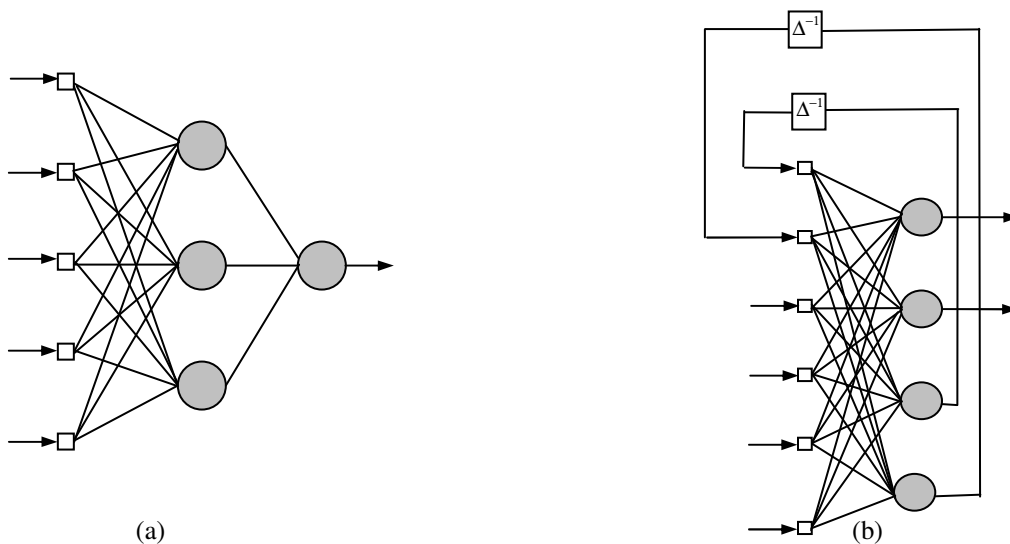


Figura 1.2 (a) Arquitetura de rede alimentada adiante (*feedforward*) (b) Rede recorrente

Na rede neural alimentada adiante a rede é alimentada com as observações das variáveis de interesse de entrada da série temporal a cada passo de tempo e o objetivo é prever um passo à frente. As redes recorrentes são redes neurais com um ou mais laços de realimentação. A rede opera dependendo de suas próprias saídas a cada passo de tempo. Para prever vários passos de tempo à frente é de fato uma tarefa difícil e desafiadora para a rede neural realizar, pois os erros de saída não poderão ser amplificados e deverão tender a um valor pré-estabelecido a cada passo do processamento (ELMAN, 1990 e MOZER, 1988). Utilizando o algoritmo de retropropagação o erro quadrático médio será minimizado se o

modelo não ficar preso em um mínimo local e se houver um número adequado de neurônios na camada escondida. Para assegurar que o erro seja minimizado é de costume construir a rede com uma camada escondida, ou intermediária, e dobrar de forma sucessiva os neurônios pertencentes a ela até que o erro seja reduzido (ERSOY e HONG, 1990).

1.3 Delineamento do Trabalho

Esta tese consiste de oito capítulos e um apêndice devotados à teoria econométrica de séries temporais, redes neurais, filtro de Kalman e aplicações dos algoritmos que são desenvolvidos no decorrer do trabalho.

A tese esta organizada como segue.

O Capítulo 1 apresenta uma introdução consistindo da motivação e dos objetivos que levaram ao problema de pesquisa.

O Capítulo 2 apresenta a fundamentação teórica dos modelos de séries temporais. São apresentados os modelos ARIMA-GARCH e a metodologia de Box e Jenkins para identificação, estimação, diagnóstico e previsão em séries temporais.

O Capítulo 3 apresenta a teoria sobre redes neurais, dando destaque aos algoritmos de retropropagação (*Backpropagation*), redes neurais alimentadas adiante (*Feedforward*), construção de intervalos de confiança e de predição e um estudo através de simulações de Monte Carlo mostrando a influência da função de ativação sobre a rede neural. Cada uma dessas técnicas é apresentada detalhadamente.

O Capítulo 4 apresenta a teoria do filtro de Kalman, apresentando a teoria do EKF e UKF (filtro de Kalman Estendido e *Unscented*). É mostrada aqui a aplicação do filtro à redes neurais e sua utilização na estimação de parâmetros da rede. Também são apresentados alguns testes para não linearidade.

O Capítulo 5 é dedicado a apresentar a metodologia utilizada.

No Capítulo 6 são apresentados os resultados experimentais de todos os testes e simulações realizados com a teoria apresentada nos capítulos anteriores.

O Capítulo 7 é dedicado às conclusões.

No Capítulo 8 tem-se as referências bibliográficas utilizadas.

Por fim, tem-se o Apêndice com demonstrações para entendimento de algumas passagens teóricas e programas criados para realizar as implementações dos modelos.

Capítulo 2

Aplicação Econométrica de Séries Temporais

2.1 Séries Temporais

A primeira razão para o interesse na análise de séries temporais é adquirir conhecimento de padrões que se desenvolvem a medida que os eventos acontecem em um período de tempo. Isto é genericamente feito através de observações e análises dos dados passados sobre ocorrência dos eventos. Tipicamente, tais processos envolvem a consolidação das experiências históricas em modelos econométricos que descrevem o comportamento dos eventos através da passagem do tempo. Esta tarefa deve ser feita da forma mais concisa possível e pode ser chamada de construção do modelo. Num segundo estágio, estes modelos econométricos são utilizados para projetar o que é provável de ocorrer durante algum curto período de tempo. Resultados obtidos por esses processos são usualmente chamados de “previsão”, ou seja, uma extrapolação para além do domínio temporal conhecido. A combinação destes dois processos formam essencialmente o que se convencionou chamar de análise de séries temporais. Os resultados dessa análise podem ser utilizados, principalmente, nas áreas de Economia e Finanças.

Uma série temporal é um conjunto de observações geradas sequencialmente no tempo (BOX e JENKINS, 1976). Se o conjunto é contínuo, a série temporal é chamada de contínua. Se o conjunto é discreto, a série temporal é chamada de discreta. As observações de uma série temporal feitas nos tempos $t_1, t_2, \dots, t_i, \dots, t_n$ podem ser denotadas por $y(t_1), y(t_2), \dots, y(t_i), \dots, y(t_n)$. Tendo n sucessivos valores de uma série disponível para análise também pode-se escrever $y_1, y_2, \dots, y_i, \dots, y_n$ para denotar as observações feitas em intervalos de tempo equidistantes $t_0 + h, t_0 + 2h, \dots, t_0 + ih, \dots, t_0 + nh$. Na prática, muitos dados de séries

temporais econômicas e financeiras são coletadas em períodos de tempo discretos. Dessa forma, utiliza-se séries temporais discretas ao invés de contínuas, sendo que y_t representará uma variável aleatória observável da série temporal num dado instante t .

Normalmente, séries temporais podem possuir cinco características: (i) tendência, (ii) sazonalidade, (iii) algum ponto de influência discrepante, (iv) uma variância que se altera no tempo - heterocedasticidade condicional e (v) não linearidade. Tipicamente, uma série temporal econômica apresenta pelo menos duas ou três dessas características (FRANSES, 1998).

2.1.1 Processos Estocásticos

Se os valores futuros de uma série temporal são determinados exatamente por alguma função matemática, então esta série é denominada de determinística. Se os valores futuros podem ser descritos apenas em termos de uma distribuição de probabilidade a série temporal é chamada de não-determinística ou simplesmente série temporal estocástica. Um fenômeno estatístico que envolva leis probabilísticas no tempo é chamado de processo estocástico. De acordo com Gouriéroux (1997) a análise da dinâmica em economia é baseada em séries temporais observadas e tomadas como realizações de um processo estocástico. Sendo que um processo estocástico é uma seqüência de variáveis aleatórias que estão definidas em um conjunto Ω , chamado de espaço de estado. Estas variáveis aleatórias podem ser unidimensionais e multidimensionais e serão indexadas pelo tempo nas aplicações que serão apresentadas. Ou seja, a utilização da metodologia para analisar as séries temporais refere-se a uma realização particular de um processo estocástico.

Definição 2.1: Um processo estocástico $Y = \{Y_{t,\theta}; t \in T, \theta \in \Omega\}$ é um conjunto de variáveis aleatórias definidas sobre um espaço de probabilidade $\{\Omega, \mathfrak{F}, P\}$ e indexadas pelos elementos do conjunto de parâmetros T . O conjunto T é usualmente o conjunto dos números inteiros ou o conjunto dos números reais. Para cada $\theta \in \Omega$ fixado tem-se uma séries temporal $Y_{t,\theta}$. Visando a simplificação da notação, a série temporal $Y_{t,\theta}$ será denotada por Y_t .

Definição 2.2: As distribuições finito-dimensionais de uma série temporal Y_t são dadas por

$$F(y_1, \dots, y_n; t_1, \dots, t_n) = P(Y_{t_1} \leq y_1, \dots, Y_{t_n} \leq y_n),$$

em que $n \geq 1; \leq t_1, \dots, t_n \in T$ e $y_1, \dots, y_n \in Y$.

Definição 2.3: As funções média, variância, auto-covariância e auto-correlação de uma série temporal são dadas, respectivamente, por

$$E[Y_t] = \int_{-\infty}^{+\infty} Y dF(y; t) = \mu_t, \quad (2.1)$$

$$Var[Y_t] = \int_{-\infty}^{+\infty} (Y - \mu_t)^2 dF(y; t) = \gamma_Y(t; t), \quad (2.2)$$

$$Cov[Y_j, Y_k] = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (Y_j - \mu_j)(Y_k - \mu_k) dF(y_j, y_k; j, k) = \gamma_Y(j; k). \quad (2.3)$$

2.1.2 Modelos Estacionários

Para restringir a classe de processos estocásticos considerada nesse trabalho, serão introduzidas duas condições: a condição de estacionariedade e a suposição de o processo é um processo de Markov. A suposição de Markov significa que o conteúdo informacional de todo o passado pode ser sumarizado por um número finito de variáveis, chamadas de variáveis de estado (GOURIÉROUX, 1997, CAMPBELL et al., 1997, p.357 DUFFIE, 2001, p. 52).

Definição 2.4: Uma série temporal Y_t , $t \in T$, é estritamente estacionária se todas as distribuições finito-dimensionais forem invariantes sobre translações em t, ou seja,

$$F(y_1, \dots, y_n; t_1, \dots, t_n) = F(y_1, \dots, y_n; t_1 + \Delta t, \dots, t_n + \Delta t), \text{ para quaisquer } t_1, \dots, t_n, \Delta t \in T.$$

Definição 2.5: Uma série temporal Y_t , $t \in T$, é fracamente estacionária se e somente se

$$E[Y_t] = \mu_t = \mu, \forall t \in T, \quad (2.4)$$

$$Var[Y_t] = \sigma^2 < \infty, \forall t \in T, \quad (2.5)$$

$$e \gamma_Y(j;k) = \gamma(|j-k|). \quad (2.6)$$

Isto quer dizer que $\gamma_Y(j;k)$ depende apenas da defasagem entre j e k e, conseqüentemente, $\rho_Y(j;k) = \rho(|j-k|)$.

Definição 2.6: Um processo estocástico Y_t diz-se gaussiano, ou normal, se para qualquer conjunto $t_1, \dots, t_n, \in T$, as variáveis aleatórias Y_{t_1}, \dots, Y_{t_n} possuem distribuição conjunta normal n-variada, isto é,

$$f(\mathbf{Y}) = \frac{1}{(2\pi)^{\frac{n}{2}}} |\Sigma|^{-\frac{1}{2}} \exp\left(-(\mathbf{Y}-\boldsymbol{\mu})^t \Sigma^{-1} (\mathbf{Y}-\boldsymbol{\mu})\right). \quad (2.7)$$

Definição 2.7: Um processo de variáveis aleatórias independentes $\varepsilon_t, t \in T$, é chamado ruído branco (RB) se e somente se

$$E[\varepsilon_t] = \mu \quad e \\ Var[\varepsilon_t] = \sigma_\varepsilon^2.$$

Definição 2.8: O processo é Markoviano de ordem k se e somente se:

$$f_{t/t-1, \dots, t-k}(y_t, y_{t-1}, \dots, y_{t-k}) = f_{t/t-1, \dots, t-K}(y_t, y_{t-1}, \dots, y_{t-K}), \quad \forall t, \forall k \geq K$$

Toda a informação é então incluída nos valores mais recentes de K.

2.1.3 Modelos Auto-regressivos de Médias Móveis (ARMA)

Os modelos auto-regressivos (AR) e de médias móveis (MA) são casos especiais do modelo geral denominado auto-regressivo de médias móveis (ARMA).

Definição 2.8: Uma série temporal $Y_t \{t = 0, \pm 1, \pm 2, \dots\}$ segue um modelo auto-regressivo de médias móveis de ordem p,q, ARMA(p,q), se satisfizer a equação

$$Y_t - \mu = \phi_1(Y_{t-1} - \mu) + \phi_2(Y_{t-2} - \mu) + \dots + \phi_p(Y_{t-p} - \mu) + \varepsilon_t - \theta_1\varepsilon_{t-1} - \theta_2\varepsilon_{t-2} - \dots - \theta_q\varepsilon_{t-q} \quad (2.8)$$

em que ε_t é um processo ruído branco com média zero e variância σ_ε^2 constante, $\phi_1, \phi_2, \dots, \phi_p, \theta_1, \theta_2, \dots, \theta_q$ são parâmetros do modelo.

A expressão (2.8) pode ser reescrita na forma reduzida:

$$\phi(B)\tilde{Y}_t = \theta(B)\varepsilon_t . \quad (2.9)$$

em que

$$\tilde{X}_t = X_t - \mu .$$

$\phi(B) = (1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p)$ é o operador auto-regressivo de ordem p;

$\theta(B) = (1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q)$ é o operador de médias móveis de ordem q.

O modelo (2.9) será estacionário se a série $\theta(B)$ convergir para $|B| \leq 1$ e será invertível se $\phi(B)$ convergir para $|B| \leq 1$, ou seja, o modelo será estacionário se as raízes de $\phi(B) = 0$ forem, em módulo, maiores que um e será invertível se as raízes de $\theta(B) = 0$ forem, em módulo, maiores que um (BOX, JENKINS, REINSEL, 1994).

Será suposto de agora em diante que $\mu = 0$.

2.1.4 Modelos Auto-regressivos (AR)

Se na expressão (2.8), $\phi_j = 0$ para $j > p$ e $\theta_q = 0$ para todo q, obtém-se o modelo auto-regressivo de ordem p, que será denotado por AR(p):

$$Y_t = \phi_1(Y_{t-1}) + \phi_2(Y_{t-2}) + \dots + \phi_p(Y_{t-p}) + \varepsilon_t , \quad (2.10)$$

em que ϕ_1, \dots, ϕ_p são parâmetros reais e $\varepsilon_t \sim RB(0, \sigma_\varepsilon^2)$.

Ou utilizando o operador retroativo B através de $B^s Y_t = Y_{t-s}, s \geq 1$, pode-se escrever (2.10) como:

$$\phi(B)\tilde{Y}_t = \varepsilon_t , \quad (2.11)$$

em que $\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p$ é o operador auto-regressivo de ordem p e

$$\tilde{Y}_t = Y_t - \mu .$$

O modelo (2.11) será estacionário se a série $\phi(B)$ convergir para $|B| \leq 1$ e será sempre invertível, ou seja, o modelo será estacionário e invertível se as raízes de $\phi(B) = 0$ forem em módulo maiores que um.

2.1.5 Modelos de Médias Móveis (MA)

Se na expressão (2.8), $\theta_j = 0$ para $j > p$ e $\phi_p = 0$ para todo p , obtém-se o modelo de médias móveis de ordem q , que será denotado por MA(q):

$$Y_t = \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q} \quad (2.12)$$

ou

$$Y_t = \theta(B) \varepsilon_t . \quad (2.13)$$

O modelo (2.13) será sempre estacionário e será invertível se a série $\theta(B)$ convergir para $|B| \leq 1$, ou seja, o modelo será estacionário e invertível se as raízes de $\theta(B) = 0$ forem em módulo maiores que um.

2.1.6 Modelos Não-estacionários

Na prática, muitas vezes são observadas séries não-estacionárias e que possuem tendência determinística, porém observa-se a estacionariedade na primeira diferença da série. Um processo ARMA é estacionário se as raízes da equação característica $\Phi(B)$ estiverem fora do círculo unitário. O modelo ARIMA foi proposto para a modelagem de processos na qual pelo menos uma das raízes se encontra sobre o círculo unitário. Processos dessa forma são denominados processos não-estacionários homogêneos.

Considere o modelo:

$$\varphi(B)Y_t = \theta(B)\varepsilon_t , \quad (2.14)$$

em que $\varphi(B)$ é um operador não-estacionário, ou seja, pelo menos uma das raízes da equação característica encontra-se sobre o círculo unitário. Pode-se também representar o modelo utilizando o operador diferença Δ , da seguinte forma:

$$\phi(B)\Delta^d Y_t = \theta(B)\varepsilon_t, \quad (2.15)$$

em que $\Delta^d = (1-B)^d$.

Este processo é denominado auto-regressivo e de médias móveis integrados denotado por ARIMA(p,d,q). Verifica-se que este modelo assume que a d-ésima diferença da série pode ser representada por um processo ARMA(p,q) estacionário e invertível.

2.1.7 Heterocedasticidade Condicional

Campbell, Lo, e MacKinlay (1997, p.481) argumentam que “é tanto logicamente inconsistente como estatisticamente ineficiente utilizar medidas de volatilidade que são baseadas na suposição de volatilidade constante ao longo do mesmo período de tempo quando as séries resultantes movem-se através do tempo”. No caso de dados financeiros, por exemplo, grandes e pequenos erros tendem a ocorrer em clusters (agrupamentos), ou seja, grandes retornos são seguidos por grandes retornos e pequenos retornos por pequenos retornos. Isto sugere que os retornos são serialmente correlacionados. Quando trabalham com não linearidades, Campbell, Lo, e MacKinlay (1997), fazem a distinção entre:

- Séries Temporais Lineares: os choques são assumidos como sendo não correlacionados mas não independentes e identicamente distribuídos (iid).
- Séries Temporais Não lineares: choques são assumidos iid, mas existe uma função não linear relacionando a série temporal observada e os choques subjacentes.

Uma vez que os dados refletem resultados de negócios entre compradores e vendedores, por exemplo no mercado acionário, várias fontes de eventos econômicos exógenos podem ter um impacto sobre o padrão dos preços de capital. Dado que algumas notícias podem levar a várias interpretações diferentes e também que certos eventos econômicos específicos, tal como uma crise de petróleo, podem durar por algum tempo, freqüentemente observa-se que grandes quantidades positivas e grandes quantidades negativas de observações em séries temporais financeiras tendem a aparecer em clusters (agrupamentos). Uma abordagem para tratar esta situação é explorar o fato de que os outliers aparecem em clusters e tentar construir um modelo de série temporal para os próprios outliers. Uma vez que a seqüência de outliers pode refletir a volatilidade do período, estes modelos de séries

temporais podem ser utilizados para prever a volatilidade. Ou seja, devido a presença de valores discrepantes, a variância de séries temporais financeiras varia em função do tempo e conseqüentemente os intervalos de previsão para cada nível também deverão variar. A intuição é que em períodos de maior volatilidade existe maior incerteza sobre a próxima observação do que em períodos de menor volatilidade e, conseqüentemente, em períodos que apresentam maior volatilidade, os intervalos de previsão serão maiores.

2.1.8 Modelos ARCH

O modelo ARCH (*Autoregressive Conditional Heteroskedasticity*), ou seja modelo autoregressivo com heterocedasticidade condicional foi proposto inicialmente por Engle (1982). Desde então, têm aparecido inúmeros estudos com refinamentos e modificações do modelo ARCH básico e também com aplicações empíricas em taxa de câmbio, mercado de ações e muitos outros tipos de ativos (BOLLERSLEV, ENGLE e NELSON,1994).

Definição 2.9 (Lei das Esperanças Iteradas): Seja Ω_1 e Ω_2 dois conjuntos de variáveis aleatórias tal que $\Omega_1 \subseteq \Omega_2$. Seja Y uma variável aleatória escalar.

Então, $E[Y|\Omega_1] = E[E[Y|\Omega_2]|\Omega_1]$.

Nota (Condicionalidade versus Incondicionalidade):

Se $\Omega_1 = \emptyset$, então $E[Y|\Omega_1] = E[E[Y|\Omega_2]] = E[Y]$.

Definição 2.10: Um modelo ARCH(r) é definido por:

$$\varepsilon_t = \eta_t \sqrt{h_t}, \quad (2.16)$$

$$h_t = \alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \dots + \alpha_r \varepsilon_{t-r}^2, \quad (2.17)$$

em que η_t é i.i.d. $(0,1)$, $\alpha_0 > 0, \alpha_i \geq 0, i > 0$.

Na prática, usualmente supõem-se $\eta_t \sim N(0,1)$ ou $\eta \sim t_\nu$ (MORETTIN, 2002 pág. 110).

Seja um processo ARCH(1) como sendo representado por

$$\varepsilon_t = \eta_t \sqrt{\alpha_0 + \alpha_1 \varepsilon_{t-1}^2},$$

em que η_t é um processo estocástico ruído branco. Johnston e DiNardo (1997) mencionam as seguintes propriedades dos modelos ARCH:

i) ε_t tem média zero.

Prova:

$$\varepsilon_t = \eta_t \sqrt{\alpha_0 + \alpha_1 \varepsilon_{t-1}^2}$$

$$E_{t-1}[\varepsilon_t] = \underbrace{E_{t-1}[\eta_t]}_0 \sqrt{\alpha_0 + \alpha_1 \varepsilon_{t-1}^2}$$

$$E_{t-1}[\varepsilon_t] = 0$$

$$E_{t-2}E_{t-1}[\varepsilon_t] = 0$$

(...)

$$E[\varepsilon_t] = 0$$

ii) ε_t tem variância condicional dada por $\sigma_t^2 = \alpha_0 + \alpha_1 \varepsilon_{t-1}^2$.

Prova:

$$\varepsilon_t^2 = \eta_t^2 [\alpha_0 + \alpha_1 \varepsilon_{t-1}^2]$$

$$E_{t-1}[\varepsilon_t^2] = \sigma_t^2 [\alpha_0 + \alpha_1 \varepsilon_{t-1}^2]$$

$$E_{t-1}[\varepsilon_t^2] = 1 [\alpha_0 + \alpha_1 \varepsilon_{t-1}^2]$$

$$E_{t-1}[\varepsilon_t^2] = \sigma_t^2$$

iii) ε_t tem variância incondicional dada por $\sigma^2 = \frac{\alpha_0}{1 - \alpha_1}$.

Prova:

$$\varepsilon_t^2 = \eta_t^2 [\alpha_0 + \alpha_1 \varepsilon_{t-1}^2]$$

$$E_{t-2}E_{t-1}[\varepsilon_t^2] = E_{t-2}[\alpha_0 + \alpha_1 \varepsilon_{t-1}^2] = \alpha_0 + \alpha_1 E_{t-2}[\varepsilon_{t-1}^2] = \alpha_0 + \alpha_0 \alpha_1 + \alpha_1^2 \varepsilon_{t-2}^2$$

$$E_{t-3}E_{t-2}E_{t-1}[\varepsilon_t^2] = E_{t-3}[\alpha_0 + \alpha_0 \alpha_1 + \alpha_1^2 \varepsilon_{t-2}^2] = \alpha_0 + \alpha_0 \alpha_1 + \alpha_1^2 E_{t-3}[\varepsilon_{t-2}^2]$$

$$E_{t-3}E_{t-2}E_{t-1}[\varepsilon_t^2] = \alpha_0 + \alpha_0 \alpha_1 + \alpha_0 \alpha_1^2 + \alpha_1^3 \varepsilon_{t-3}^2$$

(...)

$$E_0 E_1 E_2 (\dots) E_{t-2} E_{t-1} [\varepsilon_t^2] = \alpha_0 (1 + \alpha_1 + \alpha_1^2 + \dots + \alpha_1^{t-1}) + \alpha_1^t \varepsilon_0^2 = \frac{\alpha_0}{1 - \alpha_1} = \sigma^2$$

Portanto, não-condicionalmente o processo é homocedástico.

iv) ε_t tem auto-covariâncias zero.

Prova:

$$E_{t-1} [\varepsilon_t \varepsilon_{t-1}] = \varepsilon_{t-1} E_{t-1} [\varepsilon_t] = 0$$

Com relação a curtose, que será chamada de K, Bera e Higgins (1993) e Morettin (2002, p. 111) mostram que o processo tem uma cauda mais pesada do que a distribuição Normal, sendo que:

$$K = 3 \left(\frac{1 - \alpha_1^2}{1 - 3\alpha_1^2} \right) > 3$$

Caudas pesadas são um aspecto comum dos dados financeiros, e conseqüentemente os modelos ARCH são populares neste campo de análise.

2.1.9 Modelos GARCH

O modelo ARCH generalizado, conhecido como GARCH (*Generalized ARCH*) foi primeiramente proposto por Bollerslev (1986).

Definição 2.11: Um modelo GARCH(r,s) é definido por:

$$\varepsilon_t = \eta_t \sqrt{h_t}, \quad (2.18)$$

$$h_t = \alpha_0 + \sum_{i=1}^s \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^r \beta_j h_{t-j}, \quad (2.19)$$

em que η_t é i.i.d. (0,1), $\alpha_0 > 0, \alpha_i \geq 0, \beta_j \geq 0, \sum_{i=1}^q (\alpha_i + \beta_j), q = \max(r, s)$.

Como no caso dos modelos ARCH, usualmente supõe-se que os η_t são normais ou seguem uma distribuição t de Student.

2.2 Modelos ARIMA: Metodologia de Box e Jenkins

De acordo com Maddala e Kim (2003) o método de Box e Jenkins é uma das metodologias mais amplamente utilizada para análise de séries temporais. O influente trabalho de Box e Jenkins (1970) mudou a atenção dos profissionais para o paradigma dos modelos ARIMA(p,d,q). Estes modelos são populares devido a sua generalidade, eles podem manipular quaisquer séries, estacionárias ou não e existe uma grande quantidade de programas computacionais que documentam bem as análises. Este último fator é um dos mais importantes para a sua popularidade. Embora Box e Jenkins não tenham sido criadores e nem os contribuidores mais importantes no campo dos modelos ARMA, eles popularizaram estes modelos e os tornaram acessíveis para todos, de tal forma que os modelos ARMA muitas vezes são referidos como modelos de Box-Jenkins.

Os passos básicos na metodologia de Box-Jenkins consistem dos seguintes cinco passos:

i) Diferenciar para obter estacionariedade. Para verificar se a série é estacionária ou não pode-se estudar o gráfico do correlograma da série, de onde obtém-se características chamadas de auto-correlação (FAC) e auto-correlação parcial (FACP). O correlograma de uma série estacionária cai com o aumento do número de defasagens. O procedimento comum é plotar o correlograma de uma dada série y_t e das sucessivas diferenças Δy , $\Delta^2 y$ e assim por diante, observando o correlograma em cada um desses estágios. Continua-se diferenciando até o correlograma cair.

ii) Identificação de um modelo. Uma vez que foi utilizado o procedimento de diferenciar para obter uma série temporal estacionária, o correlograma é examinado para decidir a ordem apropriada dos componentes AR e MA. O correlograma de um processo MA é zero depois de um certo ponto, enquanto o de um processo AR declina geometricamente. Os correlogramas dos processos ARMA mostram diferentes padrões, mas todos caem depois de um ponto.

iii) Estimação do modelo. O próximo passo é a estimação do modelo identificado no passo anterior. A estimação de um modelo AR é direta, ele é estimado utilizando o método dos mínimos quadrados ordinários (MQO) para minimizar a soma dos erros quadrado, $\sum \varepsilon_i^2$. No caso dos modelos MA, não se pode escrever a soma dos erros quadrado simplesmente como uma função dos y_s observados e dos parâmetros como no modelo. O que pode-se fazer é escrever a matriz de covariância do erro da média-móvel e, assumindo normalidade, usar o método de máxima verosimilhança para estimação (MVE). Para modelos ARMA, novamente aparece o problema com o componente MA, porém pode-se também utilizar o (MVE).

iv) Diagnóstico. Quando um modelo AR, MA ou ARMA é ajustado para uma dada série temporal, é aconselhável checar que o modelo dá uma adequada descrição dos dados. Existem dois critérios freqüentemente usados que refletem a qualidade do ajuste e o número de parâmetros estimado. Um deles é o Critério de Informação de Akaike (AIC) e o outro é o Critério de Informação Bayesiano de Schwartz (BIC). Se k é o número total de parâmetros estimado, tem-se que:

$$AIC(k) = n \log \hat{\sigma}^2 + 2k \quad (2.20)$$

e

$$SIC(k) = n \log \hat{\sigma}^2 + k \log n \quad (2.21)$$

Sendo que n é o tamanho da amostra. Se $RSS = \sum \hat{\varepsilon}_i^2$ é a soma dos resíduos quadrados, então $\hat{\sigma}^2 = RSS / (n - p)$. Se vários modelos ARMA estão sendo considerados, escolhe-se aquele com os menores AIC e BIC. Os dois critérios podem levar a diferentes conclusões. Além disso, é necessário checar o padrão de correlação serial dos resíduos. Box e Pierce (1970) sugerem avaliar não apenas a auto-correlação de primeira ordem mas as auto-correlações de todas as ordens dos resíduos. Eles sugerem o cálculo de $Q = N \sum_{\tau=1}^m \rho_\tau^2$, onde ρ_τ é a auto-correlação do lag τ e N é o número de observações na série. Se o modelo ajustado é apropriado, eles argumentam que Q tem uma distribuição χ^2 assintótica com $m - p - q$ graus de liberdade, onde p e q são, respectivamente, as ordens dos componentes do AR e do MA. Embora as estatísticas Q sejam amplamente utilizadas, elas não são apropriadas

em modelos auto-regressivos (MADDALA e KIM, 2003). Um argumento contra o seu uso é que as estimativas de MQO dos coeficientes das variáveis dependentes defasadas são inconsistentes na presença de erros correlacionados serialmente.

v) Previsão. Supondo que um modelo tenha sido estimado com as observações, quer se prever y_{n+k} , chamada de previsão de k-períodos à frente. Primeiro é necessário escrever a expressão para y_{n+k} e então substituir todos os valores futuros y_{n+j} ($0 < j < k$) pelas suas previsões e ε_{n+j} ($j > 0$) por zero (uma vez que seu valor esperado é zero). Também são substituídos todos os ε_{n-j} ($j \geq 0$) pelos resíduos preditos.

2.2.1 Medidas para Avaliar Previsão

Existem muitos trabalhos escritos sobre o desempenho de previsão dos modelos GARCH (CHAKRABORTY, 1992; DIEBOLD e MARIANO, 1995; BARUCCI e RENO, 2002; ANDERS, 2006) e tradicionalmente algumas medidas são mais utilizadas para comparar o ajuste e a precisão da previsão de modelos e técnicas alternativas: o RMSE (raiz do erro quadrado médio), MAE (erro absoluto médio), MAPE (erro percentual absoluto médio), CORR (coeficiente de correlação de Pearson), Sign (proporção de vezes em que o sinal é corretamente previsto) e o TIC (coeficiente de desigualdade de Theil).

$$MAE = \frac{\sum_{t=T+1}^{T+h} |\hat{y}_t - y_t|}{h}. \quad (2.22)$$

O MAE é uma medida estatística que depende da escala da variável dependente.

$$MAPE = \frac{100}{h} \sum_{t=T+1}^{T+h} \left| \frac{\hat{y}_t - y_t}{y_t} \right|. \quad (2.23)$$

O MAPE é uma medida estatística invariante com a escala da variável dependente.

$$CORR = \frac{\sum_{t=T+1}^{T+h} (y_t - \bar{y}_t)(\hat{y}_t - \bar{\hat{y}}_t)}{\sqrt{\sum_{t=T+1}^{T+h} (y_t - \bar{y}_t)^2 \sum_{t=T+1}^{T+h} (\hat{y}_t - \bar{\hat{y}}_t)^2}}. \quad (2.24)$$

$$Sign = \frac{1}{h} \sum_{t=T+1}^{T+h} z_t \quad (2.25)$$

sendo que,

$$z_t = \begin{cases} 1 & \text{se } y_{t+1} \cdot \hat{y}_{t+1} > 0 \\ 0 & \text{caso contrário} \end{cases}$$

Neste trabalho são utilizadas, como medidas de desempenho, o RMSE e o TIC dados pelas expressões (2.26) e (2.27), respectivamente.

$$TIC = \frac{\sqrt{\sum_{t=T+1}^{T+h} \frac{(\hat{y}_t - y_t)^2}{h}}}{\sqrt{\sum_{t=T+1}^{T+h} \frac{\hat{y}_t^2}{h}} + \sqrt{\sum_{t=T+1}^{T+h} \frac{y_t^2}{h}}}. \quad (2.26)$$

O TIC varia entre zero e um, sendo que zero indica ajuste perfeito.

$$RMSE = \sqrt{\sum_{t=T+1}^{T+h} \frac{(\hat{y}_t - y_t)^2}{h}}. \quad (2.27)$$

Quanto menor o RMSE significa menor o erro obtido pelo modelo utilizado para realizar a previsão.

Capítulo 3

Redes Neurais

3.1 Características das Redes Neurais

Historicamente, o estudo das redes neurais tem progredido por várias décadas e é originado das pesquisas sobre Inteligência Artificial (IA), a qual tem o interesse de entender o cérebro e/ou criar uma máquina que mimetize suas funções. As primeiras investigações sistemáticas que tentaram definir um modelo computacional do processo de pensamento do cérebro iniciou com os trabalhos de McCulloch e Pitts, Hebb e Rosenblat entre os anos de 1940 e 1950. No entanto, o progresso subsequente foi gradual e sofreu da falta de algoritmos de treinamentos eficientes, sendo que grande parte do desenvolvimento destes algoritmos foi desenvolvido na área de engenharia de controle. A renovação das atividades em redes neurais aconteceu grandemente devido ao desenvolvimento no treinamento de redes neurais com profundidade, ou camadas escondidas (também chamadas de camadas intermediárias), conhecidas como perceptrons de múltiplas camadas; MLP – *multilayer perceptron* (ROSENBLATT, 1958). Este foi o resultado de pesquisas de psicólogos como Rummelhart que publicou em 1986 um novo método de treinamento, chamado de *backpropagation* (retropropagação). A rápida disseminação desta técnica levou à uma nova onda de atividades de pesquisa. O MLP com *backpropagation* é o exemplo mais bem sucedido de uma rede neural, com a maioria das aplicações industriais (exemplo: controle e robótica) implementando esta arquitetura, a despeito de haver um grande número alternativo de arquiteturas (AZOFF, 1994).

O nome “rede neural” deriva da estrutura neural do cérebro humano e foi inventada por biólogos e psicólogos. Este assunto é hoje interdisciplinar, indo dos biólogos e psicólogos, os quais têm um grande entendimento sobre o processo computacional no

cérebro, até físicos que têm formulado com sucesso modelos mecânico-estatísticos de redes, engenheiros interessados em projetos de filtros massivamente paralelos, matemáticos, cientistas de computação e estatísticos para os quais a fundamentação rigorosa, a lógica do raciocínio e o reconhecimento de padrões são áreas ativas de pesquisa. Uma rede neural pode ser considerada como uma técnica de processamento de dados que mapeia, ou relaciona, algum tipo de entrada de informação à uma saída de dados. Por exemplo, a entrada pode estar na forma de uma imagem bidimensional com alguma parte faltante ou segmento distorcido e o processamento da saída pode produzir uma imagem corrigida. Nas aplicações em séries temporais, a entrada pode ser uma série temporal unidimensional e a saída a melhor estimativa para o próximo item na série (AZOFF, 1994). Em geral, a tarefa executada pela rede neural pode ser dividida em quatro tipos distintos de aplicação (DUDA e HART, 1993):

1. Classificação: decidir em qual categoria uma determinada entrada deverá ser inserida;
2. Associação: atua como uma memória de conteúdo endereçável que recorda uma saída, dada alguma parte desta na forma de uma entrada;
3. Codificação: comprime e codifica uma entrada através da produção de uma saída com dimensão reduzida. Também pode ser realizada a operação inversa, ou seja, uma decodificação;
4. Simulação: a criação de uma nova saída para uma entrada que atua como um estímulo, no caso da rede ter sido exposta a uma amostra de estímulo possível.

Qualquer tarefa particular pode envolver um certo número das funções descritas acima. Por exemplo, a previsão de séries temporais por um *perceptron* pode envolver um elemento de compressão dos dados de entrada (codificação), um elemento de redução de ruído (associação), a capacidade de determinar padrões recorrentes que levem a comportamento preditível (classificação) e a previsão de um único evento (simulação).

Algumas tarefas de processamento desempenhadas pela rede neural podem também ser realizadas por técnicas tradicionais, tal como a análise de regressão (AZOFF, 1994). No entanto, nenhuma compartilha das capacidades únicas da rede neural, que podem ser sumarizadas como:

- **Generalização:** a rede neural é inicialmente estabelecida a partir de uma fase de treinamento, onde entradas de exemplo são apresentadas a rede que é treinada para extrair informações relevantes desses padrões. Subseqüentemente, a rede tem a capacidade de generalizar, ou seja, uma entrada ainda não vista e não utilizada também poderá ser processada;
- **Flexibilidade:** o limite de tarefas para as quais uma rede neural pode ser aplicada, excede qualquer uma das técnicas tradicionais;
- **Modelagem não linear:** o processo de mapeamento da rede neural envolve funções não lineares que podem conseqüentemente cobrir um limite maior da complexidade do problema. Embora outras técnicas não lineares existam, a rede neural é superior na sua generalidade e fácil implementação prática.

Existem numerosas arquiteturas de redes neurais, porém estes podem ser divididos em três grandes classes, tomando como base as técnicas usadas para treinar iterativamente os parâmetros livres na rede, conhecidos como pesos. As classes de aprendizagem são as seguintes (HAYKIN, 2001):

Treinamento supervisionado: as saídas desejadas da rede são conhecidas durante o treinamento, de tal forma que o erro, a diferença entre o valor desejado e saída real, podem ser realimentadas para a rede para melhorar o desempenho. Um exemplo dessa arquitetura é o perceptron de múltiplas camadas.

Aprendizagem por reforço: valores desejados para cada um dos exemplos de entrada da rede são não conhecidos (ou não usados), apenas um critério para verificar se o desempenho da rede está correto ou não é utilizado na retropropagação, nenhuma informação referente a saída correta real está disponível para treinamento. Em alguns casos o julgamento para avaliação está disponível apenas depois que uma seqüência de entradas tenha sido apresentada, a qual cria uma separação entre entradas individuais e o desempenho desejado da rede.

Aprendizagem não supervisionada: existe uma falta total de realimentação para dizer se o objetivo da aprendizagem foi alcançado ou não. Estes tipos de rede são alto organizáveis, usando correlações entre os padrões de entrada para categorizá-los de acordo com as classes auto-descobertas. Um exemplo deste tipo de rede são os mapas de Kohonen auto-organizáveis.

3.2 Motivação para utilizar Redes Neurais na Previsão

A razão para o uso de redes neurais nesse trabalho é a previsão de um dado valor desejado ou variável de saída y , a partir de um conjunto de variáveis de entrada observadas \mathbf{x} . Em séries temporais, o conjunto de variáveis de entrada \mathbf{x} pode incluir variáveis atrasadas no tempo das variáveis atuais de \mathbf{x} , bem como valores atrasados de y . No estudo de previsão, usualmente inicia-se com um modelo de regressão dado pela seguinte equação:

$$y_t = \sum_{k=1}^K \beta_k x_{k,t} + \varepsilon_t, \quad (3.1)$$

em que $\varepsilon_t \sim N(0, \sigma^2)$ NID

sendo que a variável ε_t é considerado um distúrbio aleatório, assumido como sendo normalmente distribuído com média zero e variância σ^2 e onde $\{\beta_k\}$ representa os parâmetros a serem estimados. O conjunto de estimadores dos parâmetros é denotado por $\{\hat{\beta}_k\}$, enquanto que o conjunto de previsões de y geradas pelo modelo com o conjunto de coeficientes $\{\hat{\beta}_k\}$ é denotado por $\{\hat{y}_t\}$. O objetivo é selecionar $\{\hat{\beta}_k\}$ de tal forma que seja minimizada a soma das diferenças quadráticas entre as observações atuais y e as observações previstas pelo modelo linear, \hat{y} .

Em séries temporais, as variáveis de entrada e saída $[x,y]$, tem subscripto t , denotando a observação particular de um dado, com a observação mais antecipada iniciando em $t=1$. Em econometria existe uma grande variedade de métodos para estimar o conjunto de parâmetros $\{\beta_k\}$, sujeitos a uma variedade de suposições alternativas sobre a distribuição dos distúrbios ε_t , sobre a constância de sua variância, σ^2 , bem como sobre a independência da distribuição das variáveis de entrada x_k com relação as perturbações ε_t .

O objetivo do processo de estimação é encontrar o conjunto de parâmetros para o modelo de regressão, dado por $\{\hat{\beta}_k\}$, para minimizar E , definido como a soma das diferenças quadráticas, ou resíduos, entre o valor desejado ou variável de saída y e a variável gerada

pelo modelo \hat{y} , sobre todas as observações. O problema de regressão é formulado da seguinte forma:

$$\underset{\hat{\beta}}{\text{Min E}} = \sum_{t=1}^T \varepsilon_t^2 = \sum_{t=1}^T (y_t - \hat{y}_t)^2, \quad (3.2)$$

$$\hat{\beta} = \arg \min_{\beta} \sum_{t=1}^T \varepsilon_t^2,$$

$$y_t = \sum \beta_k x_{k,t} + \varepsilon_t \text{ e}$$

$$\hat{y}_t = \sum \hat{\beta}_k x_{k,t}.$$

Um modelo linear comumente utilizado para previsão é o modelo autoregressivo:

$$y_t = \sum_{i=1}^m \beta_i y_{t-1} + \sum_{j=1}^k \gamma_j x_{j,t} + \varepsilon_t, \quad (3.3)$$

no qual há k variáveis independentes k com γ_j coeficientes para x_j e m lags (ou valores atrasados, ou ainda, valores passados) para a variável independente y , portanto, tendo $k + m$ parâmetros a serem estimados. Dessa forma, quanto maior for a estrutura de atrasos, maior a quantidade de parâmetros que devem ser estimados e menor o número de graus de liberdade das estimativas de regressão.

O número de variáveis de saída, obviamente, pode ser maior do que um. No entanto, no modelo de regressão padrão, pode ser estimada a previsão de cada variável de saída y_j , $j = 1, 2, \dots, j^*$, com uma série de j^* modelos de regressão independentes. Para j^* variáveis de saída deve-se estimar $j^* \times k$ parâmetros (PATEL, 2000).

O modelo linear tem a propriedade útil de ter uma solução de forma fechada para resolver o problema de regressão, ou seja, de minimizar a diferença quadrática entre y e \hat{y} . Deste modo, o método de regressão linear (ou auto-regressão linear) é rápido. Para a previsão de um período curto o modelo linear é um ponto de partida razoável, ou “benchmark”, visto que em muitos mercados, observam-se apenas pequenas mudanças simétricas na variável a ser predita, ao longo de uma tendência de longo prazo. No entanto, isto pode não ser preciso o bastante para mercados financeiros voláteis em que pode haver um processo não linear dos dados. Movimentos lentos no aumento dos preços dos ativos, seguidos por colapsos repentinos conhecidos como “bolhas” são comuns de acontecer. Dessa forma, o

modelo linear pode falhar em capturar ou prever pontos de mudança acentuada nos dados, por esta razão é que são utilizadas técnicas não lineares de previsão.

A rede neural é uma alternativa aos modelos lineares e a algumas abordagens não-paramétricas para a aproximação de sistemas não lineares. A razão para o uso de uma rede neural é simples e direta. O objetivo é encontrar uma abordagem ou método o qual realize bem previsões para dados gerados para processos que freqüentemente são desconhecidos e altamente não lineares, com uma quantidade pequena de parâmetros e o qual seja mais fácil de estimar que os modelos não lineares paramétricos.

3.2.1 Redes Neurais e Estatística

Existem algumas similaridades que podem ser destacadas entre os modelos estatísticos e as redes neurais. Por exemplo, no caso da resolução de um problema utilizando a técnica de regressão linear que apresente o seguinte formato geral (MEDEIROS et al., 2006).

$$Y_i = \beta_0 + \sum_{j=1}^{p-1} \beta_j X_{ij} + \varepsilon_i, \text{ verifica-se uma similaridade de forma com a saída de um neurônio}$$

que pode ser expressada por: $v_j(n) = \sum_{i=0}^m w_{ji}(n)x_i(n)$. Assim, pode-se construir as seguintes

relações (PEREIRA e RODRIGUES, 1998):

Redes Neurais	Modelos Estatísticos
Entrada da rede	Variáveis independentes, regressores
Saída da rede	Valores previstos
Valores de treinamento, alvos	Variáveis dependentes
Erros	Resíduos
Treinamento, aprendizagem, adaptação, organização própria	Estimação
Função de erro, função de custo	Critério de estimação
Padrões	Observações
Pesos	Estimativas dos parâmetros
Neurônios intermediários	Interações
Conexões funcionais	Transformações

Aprendizado supervisionado	Regressão e análise discriminante
Aprendizado não-supervisionado	Redução dos dados, análise de componentes principais
Aprendizado competitivo	Análise de grupos

Tabela 3.1 Terminologias usadas em modelos estatísticos e em redes neurais

As redes neurais atuam como um tipo de modelo de regressão não paramétrica, possibilitando a modelagem de formas funcionais complexas. As redes neurais são úteis quando não se sabe o relacionamento funcional entre variáveis dependentes e independentes, a rede usa os dados para determinar o relacionamento funcional entre as variáveis dependentes e independentes. Sendo a rede neural dependente dos dados, ela apresenta um desempenho melhor em função do tamanho da amostra. A regressão tem desempenho melhor quando a teoria ou a experiência indicam um padrão subjacente. A regressão também pode ser a melhor alternativa para amostras com tamanho extremamente pequenos (WARNER e MISRA, 1996). O modelo de regressão por postos é a técnica mais adequada para ser utilizada quando tem-se o interesse em construir uma função que represente dados que apresentam monotonicidade linear (IMAN e CONOVER, 1979). Uma das vantagens de utilizar regressão utilizando método dos mínimos quadrados ordinários (MQO) é a de utilizar pouca quantidade de dados observados, quando comparado a outros algoritmos, para chegar a valores assintóticos (GALINDO-FLORES, 1998, p. 71).

3.3 Modelo de um Neurônio Artificial

Um neurônio é uma unidade de processamento que é fundamental para a operação de uma rede neural (HAYKIN, 2001, p. 36). A Figura 3.1 apresenta o modelo de um neurônio, que forma a base para o projeto de redes neurais (artificiais).

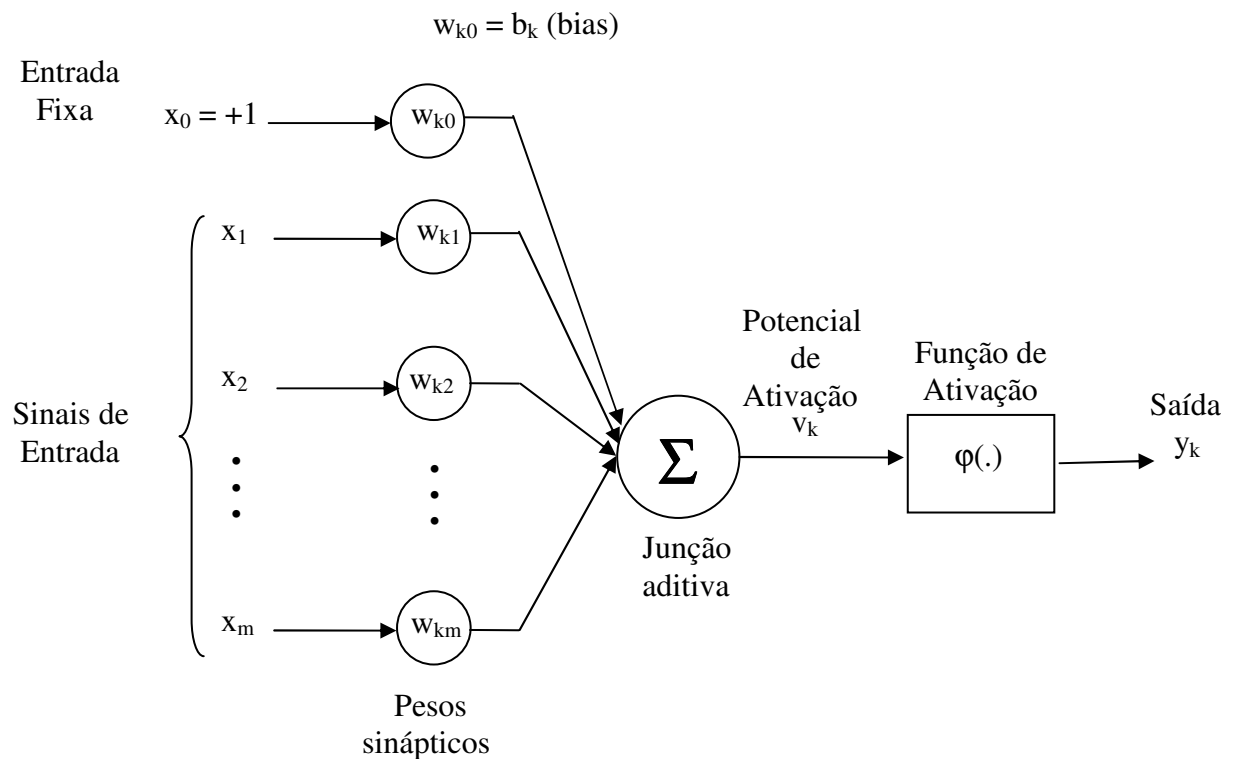


Figura 3.1 Modelo de um neurônio artificial (Haykin, 2001)

Neste modelo neural da Figura 3.1 tem-se:

- Um conjunto de *sinapses* ou *elos de conexão*, sendo que cada sinapse é caracterizada por um peso ou força própria. Especificamente, um sinal x_j na entrada da sinapse j conectada ao neurônio k é multiplicado pelo peso sináptico w_{kj} . É importante notar como são escritos os índices do peso sináptico w_{kj} . O primeiro índice se refere ao neurônio em questão e o segundo refere ao terminal de entrada da sinapse à qual o peso se refere;
- Um *somador* ou *integrador* para somar os sinais de entrada, ponderados pelas respectivas sinapses do neurônio; as operações descritas aqui constituem um combinador linear;
- Uma *função de ativação* para restringir a amplitude da saída de um neurônio. A função de ativação é também referida como *função restritiva* já que restringe (limita) o intervalo permissível de amplitude do sinal de saída a um valor finito.

Tipicamente, o intervalo normalizado da amplitude da saída de um neurônio é escrito como intervalo unitário fechado $[0,1]$ ou alternativamente $[-1,1]$.

O modelo neural da Figura 3.1 inclui também um *bias* aplicado externamente, representado por b_k . O bias b_k tem o efeito de aumentar ou diminuir a entrada líquida da função de ativação, dependendo se ele é positivo ou negativo, respectivamente.

Em termos matemáticos, pode-se descrever um neurônio k a partir das seguintes equações:

$$u_k = \sum_{j=1}^m w_{kj} x_j, \quad (3.4)$$

$$y_k = \varphi(u_k + b_k), \quad (3.5)$$

sendo que 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; $\varphi(\bullet)$ é a função de ativação e y_k é o sinal de saída do neurônio. O uso do bias b_k tem o efeito de aplicar uma *transformação afim* à saída u_k do combinador linear no modelo da Figura 3.1, como mostrado por:

$$v_k = u_k + b_k. \quad (3.6)$$

Em particular, dependendo se o bias b_k é positivo ou negativo, a relação entre o *campo local induzido* ou *potencial de ativação* v_k do neurônio k e a saída do combinador linear u_k pode ser alterada.

3.4 Algoritmo de Retropropagação (*Backpropagation*)

A Figura 3.1 apresenta um perceptron de múltiplas camadas formado por cinco elementos principais: camada de entrada, pesos, camada oculta, bias e camada de saída. O primeiro elemento é a **camada de entrada**, mostrada como quadrados, sendo que estes nós é que recebem os valores dos dados de entrada. As linhas de interconexão indicam que o valor de saída de um neurônio é passado ao longo desta linha até o próximo neurônio. Quando todos os dados da camada de entrada tiverem passado através da última camada, conhecida como camada de saída, um ciclo ou época terá sido realizado. Cada linha de interconexão tem um valor, chamado de **peso**, os pesos são parâmetros os que operam sobre os dados associados a cada linha (ou seja, é realizada uma multiplicação do valor do dado pelo peso). Estes pesos são incrementalmente ajustados durante a fase de treinamento de tal forma a alcançar

o resultado de saída desejado para um determinado dado de entrada. Inicializações típicas para estes pesos encontram-se no intervalo de $[-1;1]$ ou menos.

A segunda camada, e todas as camadas subseqüentes, contêm nós de processamento, conhecidos como neurônios artificiais e são mostrados como círculos na Figura 3.2. qualquer camada entre as camadas de entrada e saída são chamadas de **camadas ocultas** (ou intermediárias). O processamento realizado em um neurônio artificial pode ser dividido em três passos:

- 1- os dados que passam ao longo das linhas de entrada para os neurônios são multiplicados pelos de pesos;
- 2- todos esses dados que foram multiplicados pelos pesos são somados dentro do neurônio;
- 3- o valor total dessa soma é passado através de uma função de transferência, cuja saída representa o valor de saída do neurônio.

A função de transferência mais simples é a função linear (adequada para um espaço linearmente separável), a qual iguala a saída à entrada. A utilização de funções de transferência não lineares (adequadas para um espaço não linearmente separável) é uma das características principais da rede neural.

Na Figura 3.1 também é mostrado um nó especial, este nó é conhecido como bias e tem o seu valor fixado como sendo 1 e alimenta todos os neurônios dentro da rede. A utilização do bias elimina a necessidade de tratar o limiar como uma característica especial do neurônio e leva a uma implementação de algoritmo mais eficiente.

No final da rede encontra-se o neurônio de saída ou a camada de saída na qual a saída é o estado final da rede. A seleção do número de neurônios na camada de saída e a escolha de suas funções de transferência deverá ser feita atendendo os requisitos do problema de que se dispõem.

A seguir são apresentadas todas as etapas do algoritmo de *backpropagation* (AZOFF, 1994).

1. Definir a função de pesos da rede, $E(w)$, conhecida como a energia ou função de custo, a qual deseja-se minimizar. Considerando a metade da soma quadrática dos erros (SSE – *sum of square errors*) tem-se que:

$$E_{total} = \sum_{\mu} \sum_o E(w) , \quad (3.7)$$

$$E(w) = \frac{1}{2} [T - O]^2 , \quad (3.8)$$

$$\frac{\partial E}{\partial O} = -(T - O) , \quad (3.9)$$

sendo que a soma em (3.7) é realizada sobre os padrões μ e os neurônios de saída o , T é o valor alvo e O é a saída observada da rede, uma função dos pesos da rede w . A expressão (3.7) significa que deve-se minimizar o erro de todos os processadores da camada de saída para todos os padrões que são apresentados à rede;

2. Definir a função de transferência, $\varphi(v)$, para os neurônios. Escolhendo a função sigmóide tem-se que:

$$v = \sum_{i=0}^n w_i x_i , \quad (3.10)$$

$$O = \varphi(v) , \quad (3.11)$$

$$\varphi(v) = \frac{1}{1 + e^{-v}} , \quad (3.12)$$

$$\varphi'(v) = O(1 - O) , \quad (3.13)$$

tendo-se que v é a entrada da função de transferência, sendo que a soma é realizada sobre todos os neurônios de $i = 1, \dots, n$, ou seja, o produto de cada peso w pelo valor de entrada x .

E $\varphi'(v)$ é a primeira derivada de $\varphi(v)$;

3. Definir as medidas de desempenho da rede. A raiz do erro quadrático médio (RMSE) é a medida mais utilizada. Um conjunto de apresentações de todos os vetores de treinamento constitui uma época. Para cada medida é selecionado um nível de tolerância para determinar o ponto no qual o treinamento da rede deve parar;

4. Inicializar os pesos randomicamente, na prática utilizam-se pesos no intervalo de $[-0,5; 0,5]$. Se a constante de momento for utilizada na atualização dos pesos então tomar Δw inicialmente como zero;
5. Inicializar a lista de ordem das épocas, ou seja, a ordem de apresentação de cada padrão de treinamento, usando um procedimento de aleatoriedade. Tomar o primeiro padrão indicado na lista e apresentar aos nós de entrada da rede;
6. Propagar os valores dos nós de entrada adiante através da rede. Estimar para cada neurônio a sua entrada a na função de transferência. Estimar a saída da função de transferência $\varphi(v)$, este é o valor de saída do neurônio o qual pode ser propagado adiante através das sinapses entre neurônios. Este processo é continuado até os neurônios da camada de saída terem sido alcançados e a rede ter um novo estado de saída observável;
7. Comparar os valores observados nos neurônios da camada de saída, O , com os valores alvo, T , e atualizar as medidas de desempenho com o resultado;
8. Calcular o valor δ_i para cada neurônio i na camada de saída, definido como:

$$\delta_i = -\varphi'(v_i) \frac{\partial E}{\partial O} \quad (3.14)$$

$$\text{e } \delta_i = (1 - O_i) O_i (T_i - O_i); \quad (3.15)$$

9. Retropropagar os deltas através da rede até a camada precedente dos neurônios j , conectado a saída dos neurônios i , e calcular os novos valores de delta:

$$\delta_j = -\varphi'(v_j) \sum_i w_{ij} \delta_i. \quad (3.16)$$

Dessa forma para um neurônio particular j na camada escondida anterior a camada de saída, o peso conectando-o ao neurônio i da camada de saída é multiplicado pelo último valor de delta e somado com todos os neurônios da camada de saída i . Esta soma é então multiplicada pela derivada da função de transferência do neurônio j , ou seja, $\varphi'(v_j)$. Este processo é continuado para trás na rede até que todos os deltas tenham sido calculados para cada neurônio. Se forem substituídos os neurônios i por k tem-se $\delta_j = -\varphi'(v_j) \sum_k w_{kj} \delta_k$;

10. O peso conectando quaisquer dois neurônios é atualizado pelo produto da saída dos neurônios da camada precedente, V_p , e o valor de delta dos neurônios da camada precedente, δ_q :

$$p \rightarrow q, \quad (3.17)$$

$$\Delta w_{qp} = \eta \delta_q V_p, \quad (3.18)$$

$$w_{qp}^{novo} = w_{qp}^{antigo} + \Delta w_{qp}, \quad (3.19)$$

sendo que a seta na expressão (3.17) indica uma estrutura de caminho adiante, e η é a taxa de aprendizagem. Todos os pesos são atualizados de acordo com esta regra;

11. A expressão de atualização de pesos acima para Δw pode ser modificada com a adição de um termo de momento:

$$\Delta w_{qp}^{novo} = \eta \delta_q V_p + \alpha \Delta w_{qp}^{antigo}, \quad (3.20)$$

sendo que o coeficiente de momento α recebe um valor no intervalo de $[0;1]$. A consequência disto é dar à atualização dos pesos uma memória de sua última atualização, fornecendo uma suavização nas forças que afetam as mudanças dos pesos;

12. Se a lista de épocas foi exaurida, deve-se ir até o próximo passo, caso contrário apresentar o próximo conjunto de treinamento à camada de entrada e voltar ao passo 6;

13. Tendo completado uma época, e acumulado os erros de todos os padrões de entrada, os erros obtidos são avaliados e comparados com os níveis de tolerância. Se a rede não tiver convergido então voltar até o passo 5, caso contrário encerrar o treinamento.

3.5 Redes Neurais Alimentadas Adiante (*Feedforward*)

A aplicação de redes neurais na previsão de séries temporais envolve três estágios principais. O primeiro é o estágio de pré-processamento no qual a arquitetura da rede é definida. O estágio de pré-processamento também inclui a construção de um vetor de entrada e sua transformação para uma forma adequada para treinamento da rede. O segundo é o procedimento de treinamento no qual a rede aprende as relações subjacentes que governam os dados de entrada e saída. Neste estágio os parâmetros da rede são adaptados de acordo com o algoritmo de aprendizagem de tal forma a minimizar o erro de

aproximação na camada de saída. Finalmente, há o estágio de teste da rede onde um conjunto separado dos dados que não foi apresentado à rede é usado para estimar o desempenho levando a seleção do melhor da rede treinada. Muitas medidas de desempenho, tal como a raiz do erro quadrático médio (RMSE – *root mean square error*) e a medida de correlação podem ser computadas para medir o desempenho.

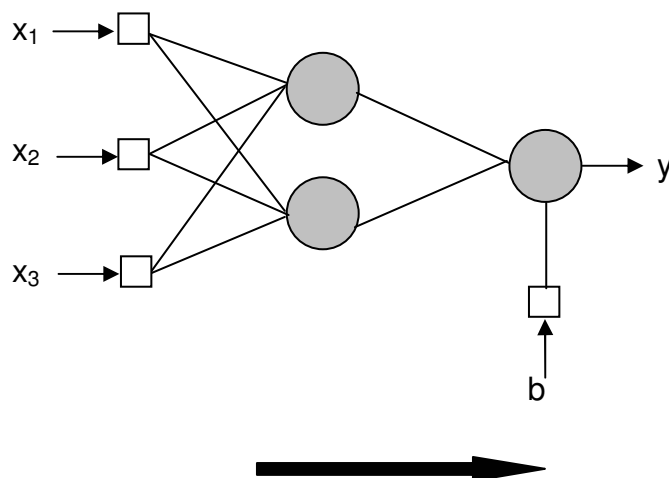


Figura 3.2 Rede *Feedforward* – Alimentada Adiante

Será descrito agora o algoritmo de aprendizagem de retropropagação temporal para uma rede neural alimentada adiante com atraso no tempo. A Figura 3.2 ilustra uma rede neural com uma camada escondida contendo dois neurônios, três variáveis de entrada $\{x_i\}, i = 1, 2, 3$ e uma saída y .

O que dá à rede neural seu poder de previsão é o processamento paralelo. Adicionalmente ao processamento seqüencial dos sistemas lineares típicos, no qual apenas entradas observadas são utilizadas para predizer uma saída observada, pela ponderação dos neurônios de entrada, os neurônios na camada oculta processam as entradas de forma paralela, de tal forma a melhorar as predições. Esta rede de camada simples alimentada adiante, ou multiperceptron, com uma camada escondida é a rede neural mais básica e comumente utilizada em aplicações econômicas e financeiras. Mais genericamente, a rede representa o modo como o cérebro humano processa os dados sensoriais de entrada, recebidos como neurônios de entrada, e internamente reconhecidos como um neurônio de saída.

3.5.1 Incorporação do Tempo na Rede Neural

A incorporação do tempo na operação de uma rede neural pode ser realizada de duas formas (HAYKIN, 2001): na primeira forma, chamada de representação implícita, o tempo é representado pelo efeito que tem sobre o processamento de sinais de uma maneira implícita. Por exemplo, o sinal de entrada é mostrado uniformemente, e a seqüência de pesos sinápticos de cada neurônio conectados à camada de entrada da rede sofre uma convolução com uma seqüência diferente de amostras de entrada. Fazendo assim, a estrutura temporal do sinal de entrada é inserida na estrutura espacial da rede. Na segunda forma, chamada de, representação explícita, o tempo recebe sua própria representação particular. O sistema de ecolocalização de um morcego, por exemplo, opera emitindo um curto sinal de frequência modulada (FM), de modo que o mesmo nível de intensidade é mantido para cada canal de frequência restrito a um período muito curto dentro da varredura de FM. São realizadas múltiplas comparações entre várias frequências diferentes codificadas por um arranjo de receptores auditivos com a finalidade de extrair informação precisa sobre a distância (alcance) até o alvo. Quando um eco é recebido do alvo com um atraso desconhecido, um neurônio (no sistema auditivo) com uma linha de atraso casada com este sinal responde, fornecendo dessa forma uma estimativa do alcance do alvo (HAYKIN, 2001).

Neste trabalho será utilizada a representação implícita do tempo, de tal forma que uma rede neural estática (por exemplo, um perceptron de múltiplas camadas) é suprida de propriedades dinâmicas. E dessa maneira a rede torná-se sensível à estrutura temporal dos sinais portadores de informação. Para que uma rede neural seja dinâmica, ela deve ter memória.

A memória pode ser dividida em memória de *curto prazo* e de *longo prazo*, dependendo do tempo de retenção. Nesse estudo necessita-se de alguma forma de memória de curto prazo para tornar a rede dinâmica. Uma forma simples de inserir memória de curto prazo na estrutura de uma rede neural é através de atrasos de tempo, que podem ser implementados a nível sináptico dentro da rede ou na camada de entrada da rede.

Portanto, o papel principal da memória é transformar uma rede estática em uma rede dinâmica. Em particular, incorporando memória na estrutura de uma rede estática como um perceptron de múltiplas camadas ordinário, a saída da rede se torna uma função do tempo. Sendo assim, a abordagem para construir um sistema dinâmico não linear é direta porque fornece uma clara separação de responsabilidades: a rede estática é responsável pela não linearidade, e a memória é responsável pelo tempo.

Esta abordagem é chamada de rede neural com atraso de tempo (TDNN – *time delay neural network*) e tem sido utilizada em diversas áreas, tais como reconhecimento de fala. Formalmente, os atrasos de tempo são a janelas de tempo e podem então ser visualizadas como modelos autoregressivos. Na Figura 3.3 tem-se uma TDNN.

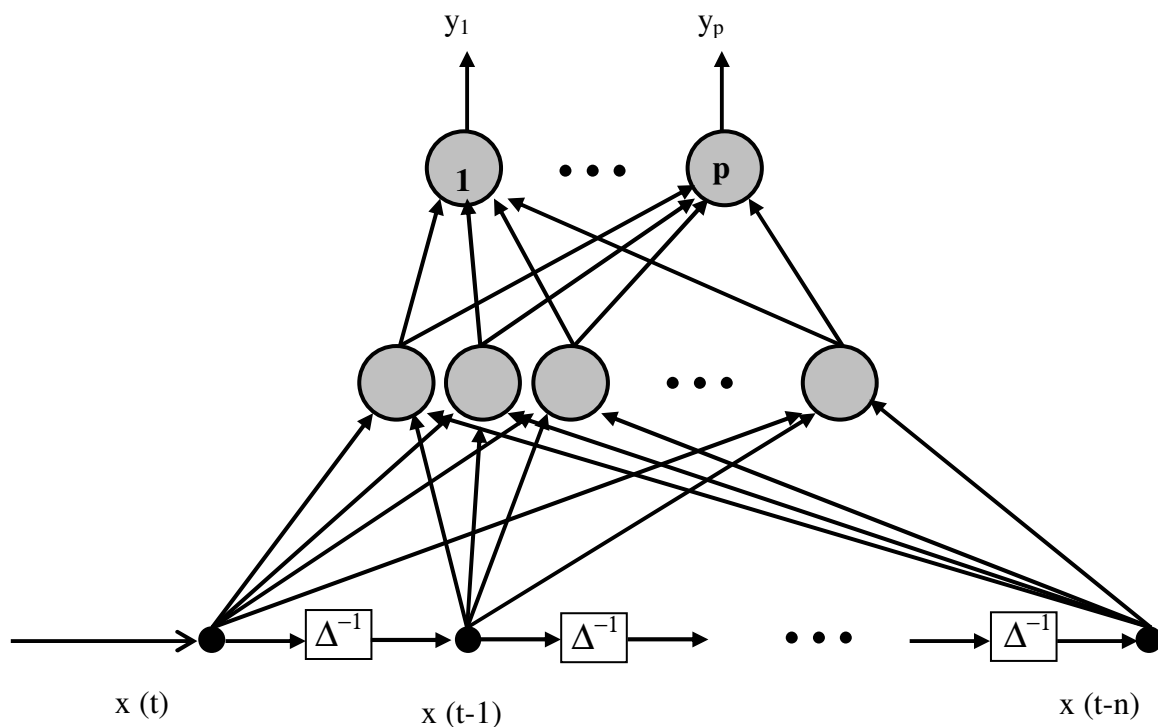


Figura 3.3 Exemplo de uma rede TDNN - *Time Delay Neural Network*

Na Figura 3.4a tem-se uma unidade de processamento temporal composta de uma memória de linha de atraso derivada com suas derivações conectadas às sinapses de um neurônio. A memória de linha de atraso derivada captura a informação temporal contida no sinal de entrada e o neurônio insere esta informação em seus próprios pesos sinápticos. Esta unidade de processamento é chamada de filtro neural focado, focado no sentido de que a estrutura inteira da memória é localizada no terminal de entrada da unidade.

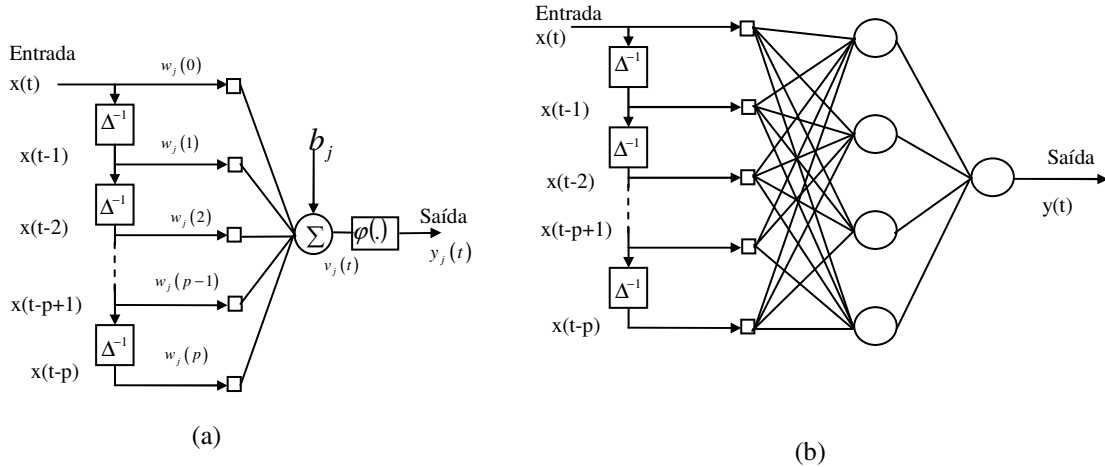


Figura 3.4 Filtro Neural Focado (a), Rede TLFN – *Time Lagged Feedforward Network* (b)

A rede representada na Figura 3.4b é chamada de rede alimentada adiante focada defasada no tempo (TLFN – *focused time lagged feedforward network*), consiste em uma memória de linha de atraso com derivações de ordem p e um perceptron de múltiplas camadas.

A saída do filtro na Figura 3.4a, em resposta à entrada $x(t)$ e aos seus valores passados $x(t-1), \dots, x(t-p)$, é dada por:

$$y_j(t) = \varphi \left(\sum_{l=0}^p w_j(l) \cdot x(t-l) + b_j \right). \quad (3.21)$$

No caso da Figura 3.4b onde tem-se uma camada oculta a saída $y(t)$ será dada por:

$$y(t) = \sum_{j=1}^{m_1} w_j \cdot y_j(t), \quad (3.22)$$

substituindo $y_j(t)$ da expressão anterior implica que a saída na rede da Figura 3.4b será:

$$y(t) = \sum_{j=1}^{m_1} w_j \cdot \varphi \left(\sum_{l=0}^{m_1} w_j(l) \cdot x(t-l) + b_j \right) + b_0. \quad (3.23)$$

A Figura 3.5 mostra o diagrama de um filtro de resposta a impulso de duração finita (FIR – *finite impulse response filter*). O filtro FIR é um dos blocos construtivos básicos na área de processamento digital de sinais.

Tem-se então que a combinação de elementos de atraso unitário e os pesos sinápticos associados pode ser vista como um filtro FIR que gera uma saída $s_j(t)$ que é a convolução

entre os sinais de entrada e os pesos sinápticos. A convolução pode ser entendida como um método matemático que combina dois sinais para produzir um terceiro sinal.

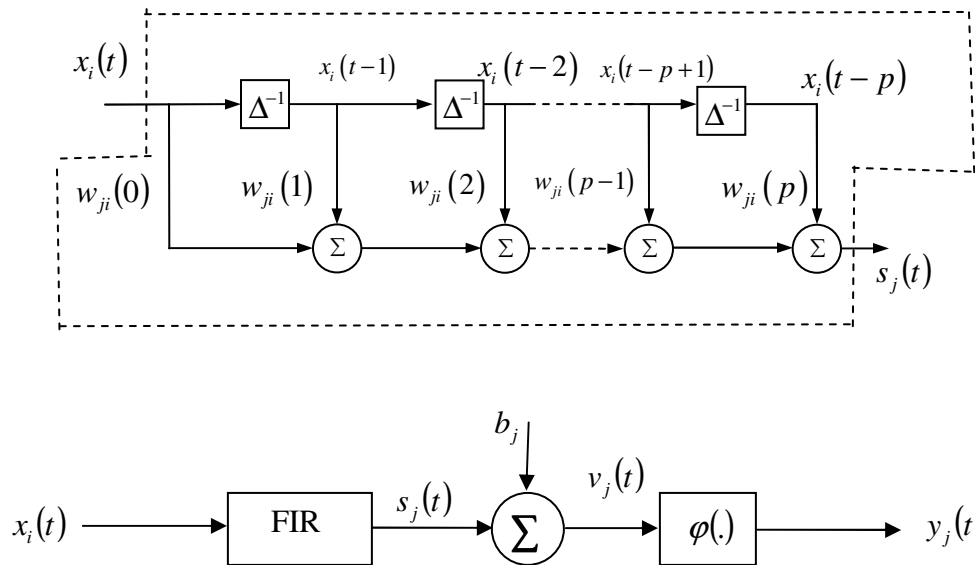


Figura 3.5 Filtro FIR aplicado à uma rede neural

A saída do filtro FIR da Figura 3.5 é dada por:

$$s_j(t) = \sum_{k=0}^p w_{ji}(k)x_i(t-k). \quad (3.24)$$

Pode-se então, em termos matemáticos, descrever o processamento espaço-temporal realizado no filtro neural da Figura 3.5 expressando sua saída $y_j(n)$, como:

$$y_j(t) = \varphi \left(\sum_{i=1}^m \sum_{l=0}^p w_{ji}(l)x_i(t-l) + b_j \right), \quad (3.25)$$

sendo que $w_{ji}(l)$ é o peso da l -ésima sinapse secundária pertencente à i -ésima sinapse primária, $x_i(t)$ é a entrada aplicada à i -ésima sinapse primária no tempo t e b_j é o viés aplicado ao neurônio. Na Figura 3.5, o neurônio possui m sinapses primárias que são responsáveis pela dimensão espacial do processamento da série, sendo que cada sinapse primária possui $(p+1)$ sinapses secundárias que são responsáveis pela dimensão temporal do processamento da série e o número de total de pesos sinápticos é igual a $m(p+1)$.

Uma limitação da TLFN é de ser adequada apenas para o uso em ambientes estacionários (ou seja, invariantes no tempo). Para superar esta limitação utiliza-se uma rede alimentada

adiante atrasada no tempo (TLFN) distribuída. Esta rede é distribuída porque a influência implícita do tempo é distribuída através da rede.

O modelagem de uma rede neural alimentada adiante atrasada no tempo distribuída pode ser realizada conforme mostrado a seguir.

Seja $w_{ji}(l)$ o peso conectado a l -ésima derivação do filtro FIR que modela a sinapse que conecta a saída do neurônio i ao neurônio j , então o sinal $s_{ji}(t)$ que aparece na i -ésima sinapse do neurônio j é dado por:

$$s_{ji}(t) = \sum_{l=0}^p w_{ji}(l) x_i(t-l). \quad (3.26)$$

Escrevendo vetorialmente tem-se:

$$s_{ji}(t) = \mathbf{w}_{ji}^T \mathbf{x}_i(t), \quad (3.27)$$

sendo que:

$$\mathbf{x}_i(t) = [x_i(t), x_i(t-1), \dots, x_i(t-p)]^T \quad (3.28)$$

$$\text{e } \mathbf{w}_{ji} = [w_{ji}(0), w_{ji}(1), \dots, w_{ji}(p)]^T. \quad (3.29)$$

O sobrescrito T significa operador de transposição.

A saída do neurônio j será dada então pelas seguintes expressões:

$$v_j(t) = \sum_{i=1}^m s_{ji}(t) + b_j = \sum_{i=1}^m \mathbf{w}_{ji}^T \mathbf{x}_i(t) + b_j, \quad (3.30)$$

$$y_j(t) = \varphi(v_j(t)), \quad (3.31)$$

sendo que $v_j(t)$ representa o campo local induzido do neurônio j (HAYKIN, 2001, p. 54),

b_j é o bias aplicado externamente e $\varphi(\cdot)$ representa a função de ativação não linear do

neurônio. É importante notar que se o vetor de pesos \mathbf{w}_{ji} e o vetor de estado $\mathbf{x}_i(t)$ forem

substituídos pelos escalares w_{ji} e x_i , respectivamente, o modelo dinâmico do neurônio se

reduz ao modelo estático do perceptron de múltiplas camadas.

3.5.2 Algoritmo de Retropropagação Temporal

Para realizar o treinamento da TLFN distribuída é necessário um algoritmo de aprendizagem supervisionada no qual a resposta real de cada neurônio na camada de saída é comparada com uma resposta desejada a cada instante de tempo (LANG e HINTON, 1988) (WAIBEL, et al. 1989).

Seja $y_j(t)$ a resposta real do neurônio j da camada de saída no tempo t e $d_j(t)$ a resposta desejada do neurônio j da camada de saída no tempo t . O valor instantâneo para a soma dos erros quadráticos será dado por:

$$E(t) = \frac{1}{2} \sum_j e_j^2(t), \quad (3.32)$$

o erro é definido como:

$$e_j(t) = d_j(t) - y_j(t). \quad (3.33)$$

O objetivo é minimizar o erro quadrático para todos os tempos:

$$E_{total} = \sum_t E(t). \quad (3.34)$$

Diferenciando em relação ao vetor de pesos:

$$\frac{\partial E_{total}}{\partial \mathbf{w}_{ji}} = \sum_t \frac{\partial E(t)}{\partial \mathbf{w}_{ji}}. \quad (3.35)$$

Também pode-se considerar um modo alternativo de expressar a derivada parcial da função de custo E_{total} em relação ao vetor de pesos $\mathbf{w}_{ji}(t)$:

$$\frac{\partial E_{total}}{\partial \mathbf{w}_{ji}} = \sum_t \frac{\partial E(t)}{\partial v_j(t)} \cdot \frac{\partial v_j(t)}{\partial \mathbf{w}_{ji}}. \quad (3.36)$$

A derivada parcial $\frac{\partial E(t)}{\partial v_j(t)}$ pode ser interpretada como a variação da função de custo E_{total}

produzida por uma variação no campo local induzido v_j do neurônio j no tempo t .

Entretanto, é importante notar que:

$$\frac{\partial E_{total}}{\partial v_j(t)} \cdot \frac{\partial v_j(t)}{\partial \mathbf{w}_{ji}} \neq \frac{\partial E(t)}{\partial \mathbf{w}_{ji}}. \quad (3.37)$$

A igualdade é válida somente quando é realizada uma soma sobre todo t , tal como nas Equações (3.35) e (3.36).

A atualização do vetor de pesos sinápticos será realizada da seguinte maneira:

$$\mathbf{w}_{ji}(t+1) = \mathbf{w}_{ji}(t) - \eta \frac{\partial E_{total}}{\partial v_j(t)} \frac{\partial v_j(t)}{\partial \mathbf{w}_{ji}(t)} \quad (3.38)$$

porém,

$$\frac{\partial v_j(t)}{\partial \mathbf{w}_{ji}(t)} = \mathbf{x}_i(t), \quad (3.39)$$

sendo que $\mathbf{x}_i(t)$ é o vetor de entrada aplicado à sinapse i do neurônio j .

Definindo o gradiente local para o neurônio j como:

$$\delta_j(t) = -\frac{\partial E_{total}}{\partial v_j(t)}, \quad (3.40)$$

a atualização do vetor de pesos sinápticos será obtida da seguinte maneira:

$$\mathbf{w}_{ji}(t+1) = \mathbf{w}_{ji}(t) + \eta \delta_j(t) \mathbf{x}_i(t). \quad (3.41)$$

Tal como na derivação do algoritmo de retropropagação padrão, a forma explícita do gradiente local $\delta_j(t)$ depende se o neurônio j está na camada de saída ou na camada oculta da rede. Serão considerados os dois casos: o neurônio j está na camada de saída e depois a situação em que o neurônio j está em uma camada oculta.

Caso 1: O neurônio j está na camada de saída.

Neste caso tem-se simplesmente que:

$$\delta_j(t) = -\frac{\partial E_{total}}{\partial v_j(t)} = -\frac{\partial E(t)}{\partial v_j(t)} = -\frac{\partial E(t)}{\partial y_j(t)} \frac{\partial y_j(t)}{\partial v_j(t)}. \quad (3.42)$$

Porém, $E(t) = \frac{1}{2} \sum_j e_j^2(t)$ e $e_j(t) = d_j(t) - y_j(t)$, dessa forma as derivadas parciais na

expressão (3.42) podem ser substituídas por:

$$\frac{\partial E(t)}{\partial y_j(t)} = e_j(t) \frac{\partial e_j(t)}{\partial y_j(t)} = e_j(t) \frac{\partial [d_j(t) - y_j(t)]}{\partial y_j(t)} = -e_j(t), \quad (3.43)$$

$$\frac{\partial y_j(t)}{\partial v_j(t)} = \varphi'(v_j(t)) . \quad (3.44)$$

No caso em que o neurônio j está na camada de saída o gradiente local é obtido através da seguinte expressão:

$$\delta_j(t) = e_j(t) \varphi'(v_j(t)) . \quad (3.45)$$

Caso 2: O neurônio j está na camada oculta.

Seja Ψ o conjunto de todos os neurônios cujas entradas são alimentadas pelo neurônio j e considerando $v_r(t)$ o potencial do neurônio $r \in \Psi$. Pode-se então escrever que:

$$\delta_j(t) = -\frac{\partial E_{total}}{\partial v_j(t)} = -\sum_{r \in \Psi} \sum_k \frac{\partial E_{total}}{\partial v_r(k)} \frac{\partial v_r(k)}{\partial v_j(t)} . \quad (3.46)$$

Para fazer a equivalência entre $\frac{\partial E_{total}}{\partial v_j(t)}$ e $\sum_{r \in \Psi} \sum_k \frac{\partial E_{total}}{\partial v_r(k)} \frac{\partial v_r(k)}{\partial v_j(t)}$ precisa-se somar todos os campos locais induzidos das k sinapses do neurônio r e depois somar todos os r neurônios que pertencem ao conjunto de neurônios Ψ . A Figura 3.6 mostra esquematicamente como deve ser realizado o duplo somatório da Equação (3.46).

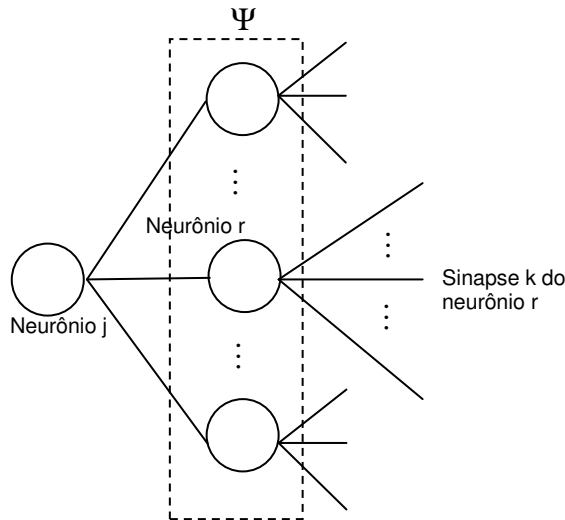


Figura 3.6 Representação do somatório da Equação (3.46)

Fazendo $\delta_r(k) = -\frac{\partial E_{total}}{\partial v_r(k)}$ implica em:

$$\delta_j(t) = \sum_{r \in \Psi} \sum_k \delta_r(k) \frac{\partial v_r(k)}{\partial v_j(t)} = \sum_{r \in \Psi} \sum_k \delta_r(k) \frac{\partial v_r(k)}{\partial y_j(t)} \frac{\partial y_j(t)}{\partial v_j(t)}, \quad (3.47)$$

lembrando que $y_j(t)$ é a saída do neurônio j e $\frac{\partial y_j(t)}{\partial v_j(t)} = \varphi'(\partial v_j(t))$ tem-se:

$$\delta_j(t) = \varphi'(\partial v_j(t)) \sum_{r \in \Psi} \sum_k \delta_r(k) \frac{\partial v_r(k)}{\partial y_j(t)}. \quad (3.48)$$

$\partial v_r(k)$ é o potencial do neurônio r que é alimentado pela saída do neurônio j dado por:

$$\partial v_r(k) = \sum_{j=0}^{m_0} \sum_{l=0}^p w_{rj}(l) y_j(k-l). \quad (3.49)$$

Com $w_{r0}(l) = b_r$ e $y_0(k-l) = 1$ representando o bias do neurônio r .

O cálculo de $\frac{\partial v_r(k)}{\partial y_j(t)}$ é mostrado na Figura 3.7.

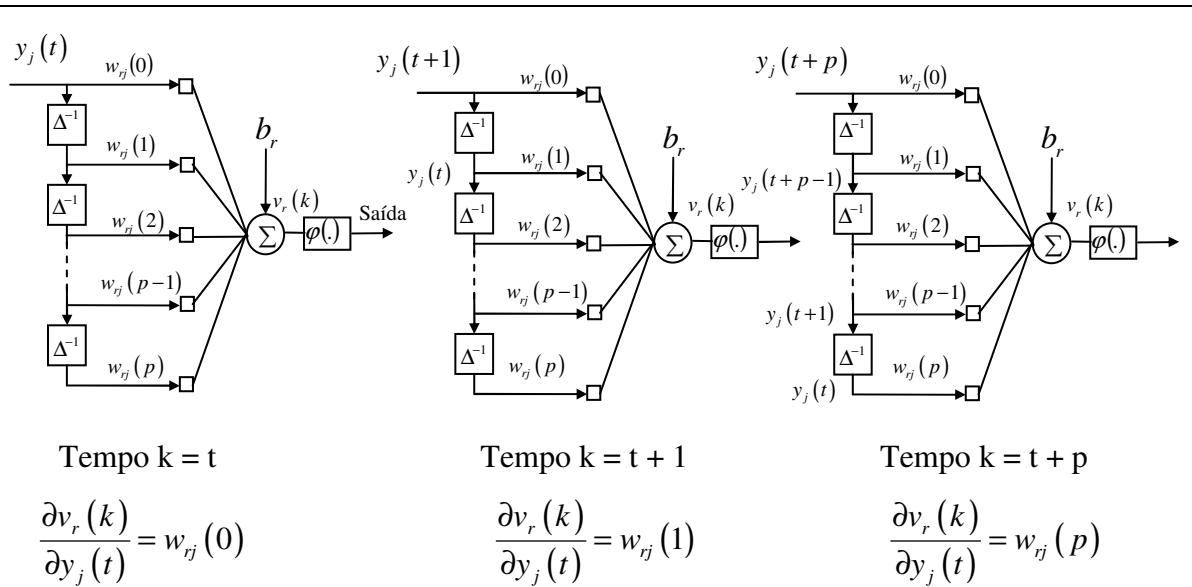


Figura 3.7 Derivadas $\partial v/\partial y$

Dessa forma a diferenciação da Equação (3.49) com relação a y_j pode ser sintetizada como:

$$\frac{\partial v_r(k)}{\partial y_j(t)} = \begin{cases} w_{rj}(k-t), & \text{para } t \leq k \leq t+p \\ 0, & \text{caso contrário} \end{cases} \quad (3.50)$$

Com isso o gradiente local será dado por:

$$\begin{aligned} \delta_j(t) &= \varphi'(\partial v_j(t)) \sum_{r \in \Psi} \sum_k \delta_r(k) \frac{\partial v_r(k)}{\partial y_j(t)}, \\ \delta_j(t) &= \varphi'(\partial v_j(t)) \sum_{r \in \Psi} \sum_{k=t}^{t+p} \delta_r(k) w_{rj}(k-t), \\ \delta_j(t) &= \varphi'(\partial v_j(t)) \sum_{r \in \Psi} \sum_{k=0}^p \delta_r(k+t) w_{rj}(k). \end{aligned} \quad (3.51)$$

Definindo dois novos vetores (p+1)-por-1:

$$\Delta_r(t) = [\delta_r(t), \delta_r(t+1), \dots, \delta_r(t+p)]^T, \quad (3.52)$$

$$\mathbf{w}_{rj}(t) = [w_{rj}(0), w_{rj}(1), \dots, w_{rj}(p)]^T. \quad (3.53)$$

Dessa forma a Equação (3.51) pode ser reescrita em notação matricial:

$$\delta_j(t) = \varphi'(\partial v_j(t)) \sum_{r \in \Psi} \Delta_r^T(t) \mathbf{w}_{rj}, \quad (3.54)$$

sendo que $\Delta_r^T \mathbf{w}_{rj}$ é o produto interno dos vetores Δ_r^T e \mathbf{w}_{rj} , ambos tendo dimensão (p+1).

A equação (3.54) completa avaliação do gradiente local $\delta_j(t)$ para o neurônio j na camada oculta.

A atualização dos pesos para a retropropagação temporal pode ser resumida a partir das seguintes relações (WAN, 1990, 1994):

$$\mathbf{w}_{ji}(t+1) = \mathbf{w}_{ji}(t) + \eta \delta_j(t) \mathbf{x}_i(t), \quad (3.55)$$

$$\delta_j(t) = \begin{cases} e_j(t) \varphi'(v_j(t)), & \text{o neurônio } j \text{ está na camada de saída} \\ \varphi'(v_j(t)) \sum_{r \in \Psi} \Delta_r^T(t) \mathbf{w}_{rj}, & \text{o neurônio } j \text{ está em uma camada oculta} \end{cases} \quad (3.56)$$

O cálculo do gradiente local em (3.56) para um neurônio na camada oculta está em função dos valores futuros de $\delta_j(t)$, para tornar o vetor gradiente local Δ_r dependente apenas dos valores presentes e passados é feita a seguinte reindexação:

$$\Delta_r(t-p) = [\delta_r(t-p), \delta_r(t+1-p), \dots, \delta_r(t)]^T. \quad (3.57)$$

Perceba na Equação (3.57) que $\Delta_r(t-p)$ representa um vetor de $p+1$ elementos por 1 (Veja a Figura 3.7).

A Tabela 3.2 resume o algoritmo de retropropagação temporal para uma rede neural alimentada adiante diante - *feedforward*.

Algoritmo de Retropropagação Temporal

1. Propagar a série de entrada para frente através da rede, camada por camada. Determinar o sinal de erro para o neurônio j da camada de saída subtraindo sua saída .

2. Para o neurônio j da camada de saída, calcular:

$$\delta_j(t) = e_j(t) \varphi'(v_j(t))$$

$$\mathbf{w}_{ji}(t+1) = \mathbf{w}_{ji}(t) + \eta \delta_j(t) \mathbf{x}_i(t)$$

sendo que $\mathbf{x}_i(t)$ é o estado da sinapse i de um neurônio oculto conectado ao neurônio de saída j .

3. Para o neurônio j em uma camada oculta, calcular:

$$\delta_j(t-lp) = \varphi'(\partial v_j(t-lp)) \sum_{r \in \Psi} \Delta_r^T(t-lp) \mathbf{w}_{rj}$$

$$\mathbf{w}_{ji}(t+1) = \mathbf{w}_{ji}(t) + \eta \delta_j(t-lp) \mathbf{x}_i(t-lp)$$

sendo que p é a ordem de cada filtro sináptico FIR e o índice l identifica a camada oculta em questão. Para redes com múltiplas camadas ocultas, $l=1$ corresponde a uma camada atrás da camada de saída, $l=2$ corresponde a duas camadas atrás da camada de saída, e assim por diante.

Tabela 3.2 Resumo do algoritmo de retropropagação temporal para uma rede alimentada adiante

3.6 Intervalos de Predição para Redes Neurais Artificiais

Uma rede alimentada adiante tem todos os sinais indo em uma direção, dos neurônios de entrada para os neurônios de saída. As redes neurais alimentadas adiante de múltiplas camadas também são chamadas de *multilayer feedforward neural networks (MLFN)*. Uma conexão entre dois neurônios tem um valor numérico (peso) representando a influência do

neurônio de entrada sobre o neurônio de saída. Um caso especial desse tipo de rede, com apenas uma camada oculta e uma saída, é mostrado na Figura 3.8.

Usando a Figura 3.8 como exemplo, os sinais de entrada x_1, x_2, \dots, x_r são linearmente combinados com os vários pesos resultando em vários sinais de entrada para a segunda camada. Estes sinais de entrada são então passados através de uma função de ativação $f(\bullet)$ para produzir os sinais de saída dos neurônios da segunda camada. Uma função de ativação que normalmente é utilizada é a função logística: $f(x) = \frac{1}{1 + e^{-x}}$.

Continuando com a Figura 3.8 como exemplo, tendo uma função de ativação $f(\bullet)$, a saída total da rede dada a entrada $\mathbf{x} = (x_1, x_2, \dots, x_r)'$ é então:

$$y_{\mathbf{w}}(\mathbf{x}) = w_{20} + \sum_{i=1}^k w_{2i} f(\mathbf{w}'_{1i} \mathbf{x} + w_{1i0}), \quad (3.58)$$

sendo que $\mathbf{w}'_i = (w_{1i1}, w_{1i2}, \dots, w_{1id})$ e os pesos, ou parâmetros não conhecidos, são representados pela matriz dada por $\mathbf{W} = (w_{20}, \dots, w_{2k}, w_{110}, \dots, w_{1k0}, \mathbf{w}'_{11}, \dots, \mathbf{w}'_{1k})$, sendo que negrito indica notação matricial e o sobrescrito ' denota a transposta (HWANG e DING, 1997). Neste trabalho é assumido um modelo estatístico que relaciona y e $y_{\mathbf{w}}(\mathbf{x})$ da seguinte forma:

$$y = y_{\mathbf{w}}(\mathbf{x}) + \mathcal{E}. \quad (3.59)$$

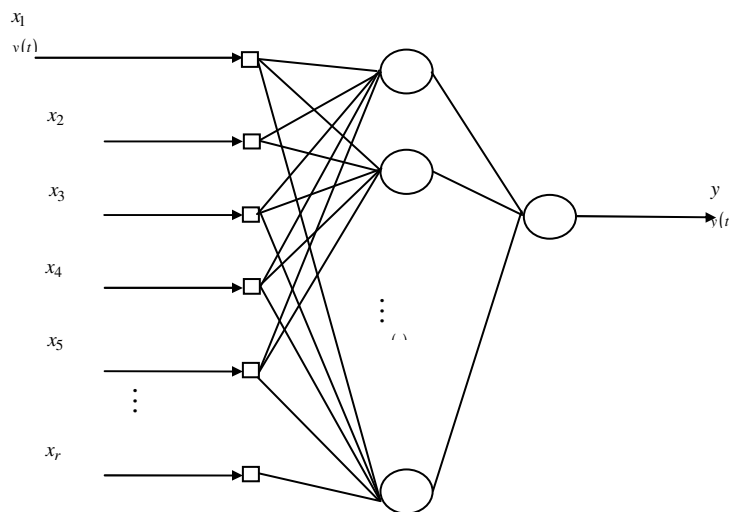


Figura 3.8 Rede neural alimentada adiante de três camadas

Considerando o ε como sendo um erro aleatório com a seguinte especificação:

1. $E[\varepsilon] = 0$ e no caso multivariado $E[\boldsymbol{\varepsilon}] = \mathbf{0}$

2. $Cov[\varepsilon] = E[\varepsilon\varepsilon'] = \sigma^2$ e no caso multivariado tem-se que $Cov[\boldsymbol{\varepsilon}] = E[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}'] = \sigma^2\mathbf{I}$.

A rede é treinada com os conjuntos de dados $(y_1, \mathbf{x}_1) \dots (y_n, \mathbf{x}_n)$; ou seja estes dados são usados para determinar o estimador $\hat{\mathbf{W}}$ dos pesos \mathbf{W} . Continuando, esta rede é usada para prever a saída futura de uma nova entrada \mathbf{x}_{n+1} por $\hat{y} = y(\mathbf{x}_{n+1})$. Assumindo que para todo $1 \leq i \leq n+1$, (y_i, \mathbf{x}_i) satisfaz (3.58) e (3.59), ou seja $y_i = y_{\mathbf{W}}(\mathbf{x}_i) + \varepsilon_i$, $i = 1, 2, \dots, n+1$, onde y_{n+1} é uma variável aleatória não observável que é o objetivo da predição. Além disso, assumindo os \mathbf{x}_i 's são independentes dos erros ε_i 's e que $(\mathbf{x}_i, \varepsilon_i)$, $1 \leq i \leq n+1$ são iid (independentes e identicamente distribuídos).

Nosso objetivo principal é construir os intervalos de predição para y_{n+1} e os intervalos de confiança para $y(\mathbf{x}_{n+1})$, que é a esperança condicionada de y_{n+1} dado \mathbf{x}_{n+1} .

3.7 Teoria de Intervalo de Predição

O problema de estimar os parâmetros na equação (3.58) utilizando a função objetivo de mínimos quadrados pode também ser formulado como um problema de regressão não linear. Os métodos-padrão que fornecem intervalos de predição assintóticos para regressão não linear podem, em teoria, ser aplicados diretamente às redes neurais. Nesta seção é feita uma revisão dessa teoria (SEBER e WILD, 1989, NETER et al., 1996, JOHNSON e WICHERN, 2002).

Considere o modelo de regressão não linear definido por:

$$\mathbf{y} = f(\mathbf{x}, \mathbf{w}) + \varepsilon, \quad (3.60)$$

sendo que \mathbf{w} é um vetor de parâmetros e $\varepsilon \sim N(\mathbf{0}, \sigma^2\mathbf{I})$. No caso da rede neural, f é dada pelo lado direito da equação (3.58). Outra forma de apresentar a equação (3.60) é:

$$\mathbf{y} = \mathbf{X} \cdot \mathbf{w} + \boldsymbol{\varepsilon} \quad (3.61)$$

$\begin{matrix} (n \times 1) & (n \times (r+1)) & ((r+1) \times 1) & (n \times 1) \end{matrix}$

Os parâmetros estimados serão denotados por $\hat{\mathbf{w}}$, sendo que é escolhido $\hat{\mathbf{w}}$ de tal forma que minimize a soma quadrada dos resíduos $SQR = \frac{1}{2}(\mathbf{y} - \hat{\mathbf{y}})'(\mathbf{y} - \hat{\mathbf{y}})$, onde $\hat{\mathbf{y}}$ é o vetor de valores preditos.

Na vizinhança dos valores verdadeiros dos parâmetros, que será denotada por \mathbf{w}^* , a função f pode ser aproximada por uma série de Taylor: $\hat{\mathbf{y}} = f(\mathbf{w}^*) + \mathbf{J}(\hat{\mathbf{w}} - \mathbf{w}^*)$, onde \mathbf{J} é a matriz cuja i -ésima entrada é $\partial f(\mathbf{x}_i) / \partial w_j$, avaliada no vetor de parâmetros verdadeiros \mathbf{w}^* , e \mathbf{x}_i é a i -ésima linha de \mathbf{X} , correspondendo ao i -ésimo conjunto de dados. Os métodos-padrão para estimar os pesos na rede neural, como por exemplo, o *backpropagation*, também calculam a matriz \mathbf{J} como parte de seus procedimentos. Pode-se aproximar a soma quadrada dos resíduos da seguinte forma:

$$SQR \approx \frac{1}{2}(\mathbf{y} - f(\mathbf{w}^*) + \mathbf{Jw}^* - \mathbf{J}\hat{\mathbf{w}})'(\mathbf{y} - f(\mathbf{w}^*) + \mathbf{Jw}^* - \mathbf{J}\hat{\mathbf{w}}) = \frac{1}{2}(\mathbf{k} - \mathbf{J}\hat{\mathbf{w}})'(\mathbf{k} - \mathbf{J}\hat{\mathbf{w}}), \quad (3.62)$$

em que $\mathbf{k} = \mathbf{y} - f(\mathbf{w}^*) + \mathbf{Jw}^* = \boldsymbol{\varepsilon}^* + \mathbf{Jw}^*$.

Para obtermos a estimativa de valores ótimos dos parâmetros, ou seja $\hat{\mathbf{w}}$, pode-se igualar a zero a derivada de SQR com relação a $\hat{\mathbf{w}}$.

A soma quadrada dos resíduos também pode ser escrita como:

$$\begin{aligned} SQR &\approx \frac{1}{2}[(\mathbf{y} - f(\mathbf{w}^*) + \mathbf{Jw}^*) - \mathbf{J}\hat{\mathbf{w}}]'[(\mathbf{y} - f(\mathbf{w}^*) + \mathbf{Jw}^*) - \mathbf{J}\hat{\mathbf{w}}] \\ &= \frac{1}{2}[\mathbf{k}'\mathbf{k} - 2\mathbf{J}'\hat{\mathbf{w}}'(\mathbf{y} - f(\mathbf{w}^*) + \mathbf{Jw}^*) + \hat{\mathbf{w}}'\mathbf{J}'\mathbf{J}\hat{\mathbf{w}}]. \end{aligned} \quad (3.63)$$

Para obter-se $\partial SQR / \partial \hat{\mathbf{w}}$ tem-se que $\partial \mathbf{k}'\mathbf{k} / \partial \hat{\mathbf{w}} = 0$ (tal como definido, o vetor \mathbf{k} não depende de $\hat{\mathbf{w}}$), $\partial / \partial \hat{\mathbf{w}}[-\mathbf{J}'\hat{\mathbf{w}}'(\mathbf{y} - f(\mathbf{w}^*) + \mathbf{Jw}^*)] = -\mathbf{J}'(\mathbf{y} - f(\mathbf{w}^*) + \mathbf{Jw}^*)$ e finalmente o último

resultado parcial é $\partial / \partial \hat{\mathbf{w}}\left[\frac{1}{2}(\hat{\mathbf{w}}'\mathbf{J}'\mathbf{J}\hat{\mathbf{w}})\right] = \mathbf{J}'\mathbf{J}\hat{\mathbf{w}}$. Com esses resultados segue então que:

$$\frac{\partial SQR}{\partial \hat{\mathbf{w}}} = 0 \Rightarrow \mathbf{J}'\mathbf{J}\hat{\mathbf{w}} = \mathbf{J}'(\mathbf{y} - f(\mathbf{w}^*) + \mathbf{Jw}^*) \text{ e conseqüentemente:}$$

$$\hat{\mathbf{w}} = (\mathbf{J}'\mathbf{J})^{-1} \mathbf{J}'(\mathbf{y} - f(\mathbf{w}^*) + \mathbf{Jw}^*). \quad (3.64)$$

A variância assintótica dos parâmetros estimados pode ser obtida pelo teorema de Gauss-Markov que resulta em: $Var[\hat{\mathbf{w}}] = \sigma^2 (\mathbf{J}'\mathbf{J})^{-1}$.

Dados os parâmetros estimados $\hat{\mathbf{w}}$ e um novo conjunto de dados \mathbf{x}_0 , pode-se aproximar o valor predito de f em \mathbf{x}_0 por:

$$\hat{y}_0 = f(\mathbf{x}_0, \hat{\mathbf{w}}) \approx f(\mathbf{x}_0, \mathbf{w}^*) + \mathbf{J}'_0 (\hat{\mathbf{w}} - \mathbf{w}^*), \quad (3.65)$$

em que \mathbf{J}_0 é um vetor cuja i -ésima entrada é $\partial f(\mathbf{x}_0)/\partial w_j$, avaliada no vetor de parâmetros verdadeiros \mathbf{w}^* . Para obter-se um estimador assintótico da variância de predição de \hat{y} dado \mathbf{x}_0 utiliza-se o resultado obtido para $\hat{\mathbf{w}}$ de onde fica-se com:

$$\begin{aligned} \hat{y}_0 &\approx f(\mathbf{x}_0, \mathbf{w}^*) + \mathbf{J}'_0 \left[(\mathbf{J}'\mathbf{J})^{-1} \mathbf{J}'(\boldsymbol{\varepsilon}^* + \mathbf{J}\mathbf{w}^*) - \mathbf{w}^* \right] \\ &= f(\mathbf{x}_0, \mathbf{w}^*) + \mathbf{J}'_0 \left[(\mathbf{J}'\mathbf{J})^{-1} \mathbf{J}'\boldsymbol{\varepsilon}^* + \mathbf{w}^* - \mathbf{w}^* \right] \\ &= f(\mathbf{x}_0, \mathbf{w}^*) + \mathbf{J}'_0 (\mathbf{J}'\mathbf{J})^{-1} \mathbf{J}'\boldsymbol{\varepsilon}^*. \end{aligned} \quad (3.66)$$

A variância de \hat{y} condicionada a \mathbf{x}_0 é $Var[\hat{y}|\mathbf{x}_0] = Var\left[f(\mathbf{x}_0, \mathbf{w}^*) + \mathbf{J}'_0 (\mathbf{J}'\mathbf{J})^{-1} \mathbf{J}'\boldsymbol{\varepsilon}^*\right]$, como a variância de $f(\mathbf{x}_0, \mathbf{w}^*)$ é igual a zero pelo fato de que os estados \mathbf{x}_0 e \mathbf{w}^* são conhecidos, e assumi-se que o vetor de erros é não correlacionado com nenhum dos outros vetores de parâmetros, concluí-se que a variância condicionada é dada por:

$$Var[\hat{y}|\mathbf{x}_0] = \sigma^2 \mathbf{J}'_0 (\mathbf{J}'\mathbf{J})^{-1} \mathbf{J}_0. \quad (3.67)$$

Pode-se estimar \mathbf{J} e \mathbf{J}_0 de forma consistente calculando as derivadas parciais em $\hat{\mathbf{w}}$. Para estimar a variância σ^2 , é necessário o valor esperado do SQR. A partir desse valor determina-se um estimador assintoticamente não-viesado de σ^2 que é mostrado na equação (3.67), ver Apêndice-A.

$$E[SQR] \approx E\left[\frac{1}{2}(\mathbf{k} - \mathbf{J}\hat{\mathbf{w}})'(\mathbf{k} - \mathbf{J}\hat{\mathbf{w}})\right] \approx \frac{1}{2} E\left[(\mathbf{k} - \mathbf{J}(\mathbf{J}'\mathbf{J})^{-1} \mathbf{J}'\mathbf{k})'(\mathbf{k} - \mathbf{J}(\mathbf{J}'\mathbf{J})^{-1} \mathbf{J}'\mathbf{k})\right].$$

Resultando em $E[SQR] \approx \sigma^2 (n - r - 1)$.

O estimador de σ^2 é:

$$s^2 = \frac{SQR}{(n - r - 1)}. \quad (3.68)$$

Tem-se que, n é o número de valores que correspondem as variáveis preditas e r é o número de parâmetros.

Para o cálculo do intervalo de predição o procedimento estatístico é inicialmente escolher a probabilidade desejada, ou seja, a probabilidade do valor predito pertencer ao intervalo. Uma das probabilidades que normalmente utiliza-se é de 95%, isso quer dizer que deve-se calcular um intervalo para que haja 95% de chance dele conter o verdadeiro valor. Este intervalo é chamado de intervalo de confiança e o valor de 95% é chamado de nível de confiança.

Como y_0 denota o valor de uma resposta quando a variável preditora tem os valores $\mathbf{x}'_0 = [1, x_{01}, \dots, x_{0r}]$, de acordo com o modelo da equação (3.61), o valor esperado de y_0 é:

$$E[y_0 | \mathbf{x}_0] = w_0 + w_1 x_{01} + \dots + w_r x_{0r} = \mathbf{x}'_0 \mathbf{w},$$

sendo que o estimador de mínimos quadrados é $\mathbf{x}'_0 \hat{\mathbf{w}}$.

Para o modelo de regressão da equação (3.61), $\mathbf{x}'_0 \hat{\mathbf{w}}$ é o estimador não-viesado de $E[y_0 | \mathbf{x}_0]$ com variância mínima dada por $Var[\hat{y} | \mathbf{x}_0] = \sigma^2 \mathbf{J}'_0 (\mathbf{J}'\mathbf{J})^{-1} \mathbf{J}_0$. Se os erros $\boldsymbol{\varepsilon}$ são normalmente distribuídos, então um intervalo de confiança (IC) de 100(1- α)% para $E[y_0 | \mathbf{x}_0] = \mathbf{x}'_0 \mathbf{w}$ é dado por (JOHNSON e WICHERN, 2002, p. 374):

$$\mathbf{x}'_0 \hat{\mathbf{w}} \pm t_{n-r-1} \left(\frac{\alpha}{2} \right) \sqrt{s^2 [\mathbf{J}'_0 (\mathbf{J}'\mathbf{J})^{-1} \mathbf{J}_0]}, \quad (3.69)$$

em que $t_{n-r-1}(\alpha/2)$ é o 100($\alpha/2$)-ésimo percentil superior de uma distribuição- t com $n-r-1$ graus de liberdade.

A predição de uma nova observação, tal como y_0 , em $\mathbf{x}'_0 = [1, x_{01}, \dots, x_{0r}]$ é mais incerta que estimar o valor esperado de y_0 . De acordo com o modelo de regressão (3.61) expresso como $y_0 = \mathbf{x}'_0 \mathbf{w} + \boldsymbol{\varepsilon}_0$, ou ainda

(nova resposta y_0) = (valor esperado de y_0 em \mathbf{x}_0) + (novo erro)

onde $\boldsymbol{\varepsilon}_0$ é distribuído como $N(0, \sigma^2)$ e é independente de $\boldsymbol{\varepsilon}$, de $\hat{\mathbf{w}}$ e de s^2 . Os erros $\boldsymbol{\varepsilon}$ influenciam os estimadores $\hat{\mathbf{w}}$ e s^2 através da resposta y , mas $\boldsymbol{\varepsilon}_0$ não.

Dado o modelo de regressão (3.58), uma nova observação y_0 tem o preditor não-viesado:

$$\mathbf{x}'_0 \hat{\mathbf{w}} = \hat{w}_0 + \hat{w}_1 x_{01} + \dots + \hat{w}_r x_{0r}.$$

A variância do erro de predição $y_0 - \mathbf{x}'_0 \hat{\mathbf{w}}$ é:

$$\text{Var}[y_0 - \mathbf{x}'_0 \hat{\mathbf{w}}] = \sigma^2 \left(1 + \mathbf{J}'_0 (\mathbf{J}'\mathbf{J})^{-1} \mathbf{J}_0 \right).$$

Quando os erros ε têm uma distribuição normal, um intervalo de predição de $100(1-\alpha)\%$ para y_0 é dado por (JOHNSON e WICHERN, 2002, p. 375):

$$\mathbf{x}'_0 \hat{\mathbf{w}} \pm t_{n-r-1} \left(\frac{\alpha}{2} \right) \sqrt{s^2 \left[1 + \mathbf{J}'_0 (\mathbf{J}'\mathbf{J})^{-1} \mathbf{J}_0 \right]}, \quad (3.70)$$

em que $t_{n-r-1}(\alpha/2)$ é o $100(\alpha/2)$ -ésimo percentil superior de uma distribuição- t com $n-r-1$ graus de liberdade.

Hwang e Ding (1997) utilizam esta teoria para a rede neural de múltiplas camadas alimentada adiante na forma mostrada na equação (3.58). Rivals e Personnaz (2000) apresentam resultados teóricos sobre a construção de intervalos de confiança para regressão não linear baseado na estimação de mínimos quadrados utilizando expansão de Taylor.

Outros métodos para computar intervalos de confiança baseados em regressão não linear têm sido propostos (DE VAUX et al., 1998).

3.8 A influência dos parâmetros de modelos ARIMA-GARCH na predição de redes neurais *feedforward*

A necessidade de processamento temporal é importante em diversas aplicações, e isso tem motivado o desenvolvimento de novas técnicas e metodologias que utilizam redes neurais para esse fim. As redes neurais podem ser classificadas como uma técnica de inferência não-paramétrica não linear multivariada, devido a possibilidade de introduzir variáveis diferentes cuja interdependência pode ser usada para prever valores futuros de uma seqüência temporal. Embora exista uma grande quantidade de técnicas que tratem de não linearidade, a capacidade das redes neurais em lidar com este tipo de comportamento tem sido bastante explorada para realizar predições (OLIVEIRA e MONTINI, 2007b). A característica não-paramétrica das redes neurais, ou seja, não fazer suposições sobre a

forma paramétrica do relacionamento funcional entre as variáveis a serem modeladas, tem sido uma vantagem adicional para sua utilização, que tem sido intensificada devido aos avanços da capacidade de processamento computacional.

Do ponto de vista teórico, o processamento de sinais não lineares (LAPÉDES e FARBER, 1987), incorporação do tempo na rede neural (ELMAN, 1988) e o modelagem não linear para predição de séries temporais caóticas (CASDAGLI, 1989) tem levado a aplicação de redes neurais como ferramenta na tomada de decisão em finanças (HAWLEY, 1990; REFENES et al., 1997; TAYLOR, 2000), análise de mercado (FISHMAN et al., 1991), modelagem não linear e previsão (CASDAGLI e EUBANK., 1992; AZZOF, 1993; CLEMENTS e HENDRY, 1999).

Mais recentemente tem havido a preocupação de comparar e relacionar a tecnologia de redes neurais com a abordagem estatística tradicional (CHENG e TITTERINGTON, 1994; RIPLEY, 1993, 1994, 1996; HWANG e DING, 1997; MEDEIROS et al., 2006), sob a perspectiva econométrica (KUAN e WHITE, 1994), de engenharia financeira (ABU-MOSTAFA et al., 2001) e macroeconômica (TERÄSVIRTA et al., 2005).

Neste trabalho são consideradas séries temporais ARIMA-GARCH univariadas construídas por simulação de Monte Carlo. O objetivo dos modelos ARIMA-GARCH é a previsão de valores futuros de uma seqüência $\{y_t\}$. As predições serão realizadas por redes neurais feedforward de três camadas, sendo que a camada intermediária utilizará, dentre as várias possibilidades existentes, a função sigmóide. Para uma revisão sobre redes feedforward ver Fine (1999) e sobre funções de ativação ver Duch e Jankowski (1997, 1999, 2000).

Através de simulações com números aleatórios, é analisado o comportamento da função de ativação sigmóide e da sua primeira derivada quando os parâmetros relacionados a estacionariedade, erro quadrado e volatilidade condicional dos modelos ARIMA-GARCH aumentam individualmente. Um número aleatório pode ser obtido de formas diferentes e assim recebendo uma classificação específica tal como número pseudo-aleatório e quasi-aleatório (ROSS, 2006; KNUTH, 1981; NIEDERREITER, 1992). Nesse trabalho as simulações são realizadas usando números pseudo-aleatórios.

Com respeito ao algoritmo de minimização utilizado destaca-se que atualmente estão disponíveis vários algoritmos de otimização, tal como: backpropagation, *steepest-descent*, backpropagation com momentum, Levenberg-Marquardt e Gauss-Newton. Todos estes

algoritmos realizam a tarefa de minimizar o erro quadrado médio, porém dadas as suas vantagens, utiliza-se neste trabalho o algoritmo de Levenberg-Marquardt.

As predições são realizadas um passo à frente, por isso tem-se apenas um neurônio na camada de saída da rede neural.

3.9 Rede Neural e Função de Ativação

Uma rede neural artificial (RNA) alimentada adiante, ou feedforward, de três camadas pode ser esquematizada como mostrado na Figura 3.9. Basicamente a RNA é constituída de linhas de ligações para as quais são atribuídos pesos e vértices que representam as conexões.

Na primeira camada, representada por quadrados, tem-se os valores de entrada para a rede, também chamados como nós de entrada. O modelo mostrado na Figura 1 inclui também um bias, y_0 , aplicado externamente, representado por y_0 , esse bias tem o efeito de aumentar ou diminuir a entrada líquida da função de ativação, dependendo se ele é positivo ou negativo, respectivamente.

As linhas de conexão indicam para qual neurônio o valor precedente está se direcionando. Cada uma dessas linhas contém um peso, estes pesos são parâmetros ajustados conforme são apresentados novos vetores de dados para a RNA. Uma RNA em que um neurônio em qualquer camada está conectado a todos os nós/neurônios da camada anterior é denominada de totalmente conectada. O fluxo de sinal através da rede progride para frente, da esquerda para a direita e de camada em camada.

A segunda camada, e todas as camadas subseqüentes, contêm neurônios de processamento, mostrados na forma de círculo. Quaisquer camadas entre as camadas de entrada e de saída são chamadas de camadas intermediárias. Uma apresentação completa de um vetor de dados que será processado constitui-se em uma época. No final da RNA a camada de saída fornece o estado final da rede.

O processamento realizado em um neurônio artificial pode ser dividido em três etapas. Na primeira etapa os dados passam ao longo das linhas de entrada de um neurônio e são multiplicados pelos pesos correspondentes. A seguir, todos os dados, após a multiplicação pelos respectivos pesos, são somados dentro do neurônio. Na etapa final o valor somado é

passado através de uma função de ativação, cuja saída representa o valor de saída do neurônio.

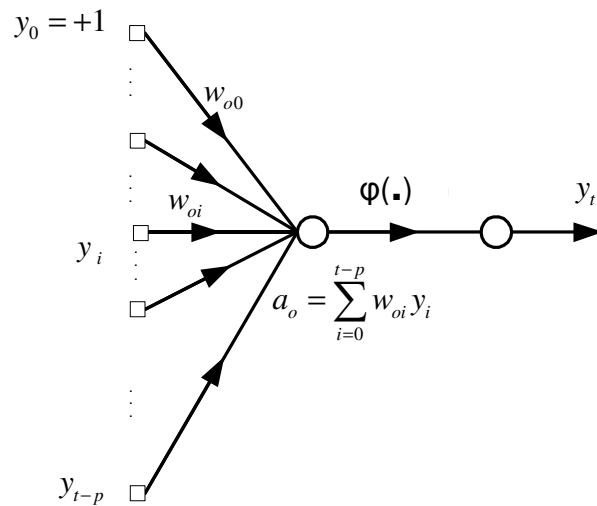


Figura 3.9 Modelo de uma rede neural

Uma RNA é uma ferramenta poderosa e flexível para realizar previsões de séries temporais, quando uma seleção adequada de amostras de entradas e saídas, número de neurônios na camada intermediária, função de ativação e recursos computacionais estão disponíveis. Muitos tipos diferentes de redes neurais têm sido desenvolvidos (BISHOP, 1995, HERTZ et al., 1991, WILLIAMS e ZIPSER, 1995, ROJAS, 1996) e aplicados em diversas áreas incluindo econometria e finanças (REFENES, 1993, TERÄSVIRTA et al., 2005).

A RNA também possui a vantagem de aproximar funções não lineares e ser capaz de resolver problemas onde o relacionamento de entrada e saída não é bem definido. Uma RNA do tipo feedforward é especialmente adequada para realizar previsão de séries que apresentam volatilidade, devido ao seu tratamento da não linearidade realizada pela utilização de funções sigmóides na camada de entradas.

A função de ativação, $\phi(\cdot)$, define a saída de um neurônio. A função de ativação é a regra para mapeamento das entradas somadas, a , do neurônio até sua saída e, por uma escolha adequada, isto significa a introdução de uma capacidade de processar não linearidade na rede. Na prática estas funções são escolhidas de tal forma a serem monotônicas e saturar nos extremos $[0,1]$ ou $[-1,1]$.

Existem vários tipos de funções de ativação, por exemplo: linear, sigmóide, linear por partes, função de limiar, arco tangente, seno, cosseno, tangente hiperbólica, função erf, $y/(1+|y|)$, Gaussiana, etc. Ver Figura 3.10.

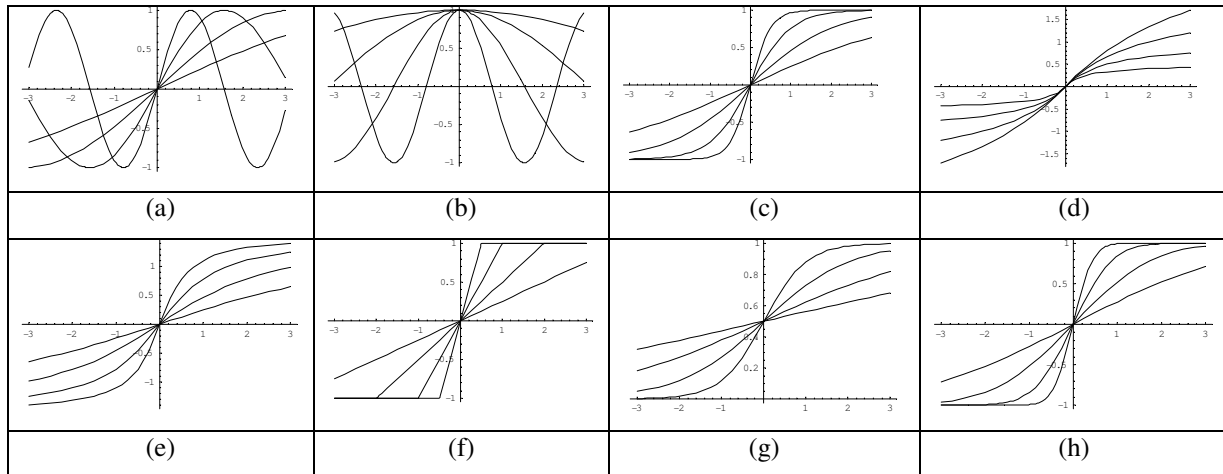


Figura 3.10 Apresenta oito funções de ativação: (a) seno, (b) cosseno, (c) tangente hiperbólica, (d) $y/(1+|y|)$, (e) arco tangente, (f) linear por partes, (g) sigmóide e (h) erf. Todas as figuras apresentam ganhos de 0,25, 0,5, 1 e 2.

A mais simples é a função linear, cuja saída é igual a entrada. A função sigmóide é a forma mais comum de função de ativação utilizada na construção de uma RNA:

$$\varphi(y) = \frac{1}{1 + e^{-gy}} \quad (3.71)$$

em que g é o ganho, ou parâmetro de inclinação da função sigmóide. Variando-se o parâmetro g , obtém-se funções sigmóides com diferentes inclinações.

Para escolha de outros tipos de funções de ativação, ver Chen et al. (2001).

3.10 Algoritmo de Levenberg-Marquardt

O algoritmo de backpropagation (RUMELHART et al., 1986a, 1986b; RUMELHART e McCLELLAND, 1986; WERBOS, 1988) é um dos mais importantes no desenvolvimento das redes neurais, no entanto sua taxa de convergência é muito pobre. Em função disso foram propostos outros algoritmos como alternativa ao backpropagation (ANDERSEN e WILAMOWSKI, 1995; BATTITI, 1992; CHARALAMBOUS, 1992; HAGAN e MENHAJ, 1994; SHAH e PALMIERI, 1990), sendo que o algoritmo de Levenberg-

Marquardt tem sido amplamente utilizado e aceito como algoritmo muito eficiente para solucionar o problema de minimização de mínimos quadrados não linear.

Pode ser mostrado que as iterações do método de Gauss-Newton e de descida do gradiente são complementares nas vantagens que eles fornecem. Levenberg (1944) propôs um algoritmo baseado nesta observação, cuja regra de atualização é uma mistura dos algoritmos mencionados acima e é dada por

$$w_{t+1} = w_t - [H + \lambda I]^{-1} \nabla f(w_t), \quad (3.72)$$

sendo que H é a matriz Hessiana calculada em w_t . Se o erro decresce após uma atualização, isto implica que a suposição de forma quadrática para $f(w)$ está valendo e é reduzido λ para reduzir a influência da descida do gradiente. Por outro lado, se o erro aumenta, pode-se seguir o gradiente e então λ é aumentado pelo mesmo fator. O algoritmo proposto por Levenberg é então composto basicamente por quatro passos. No passo um, é realizada uma atualização como descrita pela equação (3.72). A seguir, avalia-se o erro no novo vetor de parâmetros. No terceiro passo, é verificado se o erro diminuiu como resultado da atualização, e então deve-se retornar o passo (ou seja, retornar os pesos para os seus valores anteriores) e aumentar λ por um determinado fator, usualmente por um fator de 10, ou por algum fator significativo. Então voltar para o passo 1 e tentar atualizar novamente. No último passo, é avaliado se o erro diminuiu como resultado da atualização, então aceitar o passo (ou seja, manter os pesos em seus novos valores) e diminuir λ por um determinado fator e assim por diante.

Esse algoritmo tem a desvantagem de que se o valor de λ é grande, a matriz Hessiana calculada não é utilizada no todo. Pode-se obter alguma vantagem da segunda derivada em tais casos pelo escalonamento de cada componente do gradiente de acordo com a curvatura. Isto pode resultar em grandes movimentos ao longo das direções onde o gradiente é menor de tal forma que o problema dos vales não ocorra mais. A principal sugestão foi fornecida por Marquardt (1963). Ele substituiu a matriz identidade em (3.72) pela diagonal da matriz Hessiana, resultando na regra de atualização de Levenberg- Marquardt.

$$w_{t+1} = w_t - (H + \lambda \text{diag}[H])^{-1} \nabla f(w_t). \quad (3.73)$$

Problemas de minimização utilizando RNA freqüentemente são mal condicionados, isto torna o problema de minimização difícil de resolver e para tais problemas o algoritmo de

Levenberg- Marquardt (LM) é normalmente a melhor escolha. Por esta razão, o método de otimização utilizando o algoritmo LM será utilizado no treinamento das RNA utilizadas para predição das séries temporais deste trabalho.

O valor de λ é escolhido automaticamente pelo programa adotado, de tal forma que um passo decrescente é produzido. Em cada iteração, o algoritmo tenta decrescer o valor de λ por algum incremento $\Delta\lambda$. Se o valor atual de λ não decrescer o MSE (mean square error – erro quadrado médio), então λ é aumentado em passos de $\Delta\lambda$ até produzir um decréscimo.

Modificações do algoritmo LM têm sido propostas para melhorar o desempenho de aprendizagem de redes neurais de múltiplas camadas e reduzir a quantidade de oscilação no processo de aprendizagem (SURATGAR et al., 2005; WILAMOWSKI et al., 2001).

3.11 Comportamento da Função de Ativação para processos AR-GARCH

A diferenciabilidade é a única exigência que a função de ativação deve satisfazer. Um exemplo de uma função de ativação não linear, continuamente diferenciável normalmente utilizada em RNA do tipo feedforward é a sigmóide.

Considerando o índice o para indicar um neurônio a direita e i um neurônio, ou nó a esquerda, pode-se escrever o somador dos valores de entrada, ponderados pelos respectivos pesos como:

$$a_o = \sum_{i=0}^m w_{oi} y_i . \quad (3.74)$$

Compondo uma função de ativação utilizando uma função de entrada na forma de um modelo auto-regressivo com heterocedasticidade condicional auto-regressiva generalizada, tem-se:

$$\varphi \left[w_{o0} y_0 + \sum_{i=1}^m w_{oi} \left(\phi_0 + \sum_{i=1}^n \phi_i y_{t-i} + v_t \sqrt{\alpha_0 + \sum_{i=1}^p \alpha_i \varepsilon_{t-i}^2 + \sum_{i=1}^q \beta_i \sigma_{t-i}^2} \right) \right] . \quad (3.75)$$

Agora, fazendo o coeficiente ϕ_0 igual a zero, e considerando um modelo comumente utilizado AR[1]-GARCH[1,1], pode-se escrever (3.75) da seguinte maneira:

$$\varphi \left[w_{o0}y_0 + \sum_{i=1}^m w_{oi} \left(\phi_1 y_{t-1} + v_t \sqrt{\alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2} \right) \right]. \quad (3.76)$$

Durante o processo de minimização em que utiliza-se o algoritmo LM é necessário que se tenha a disposição a derivada da função de ativação, que no caso da função sigmóide é dada por:

$$\varphi'(y) = \frac{ge^{-gy}}{[1+e^{-gy}]^2}. \quad (3.77)$$

A curva que representa a derivada da função sigmóide é apresentada na Figura 3.11. Pode-se verificar que a função de ativação alcança seu valor máximo em 0,25, considerando um ganho g igual a 1.

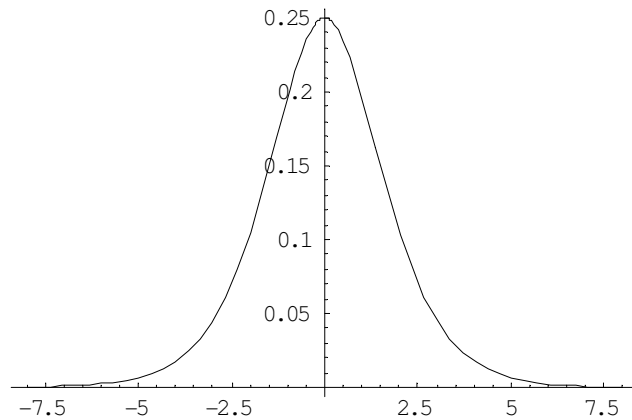


Figura 3.11 Curva da derivada da função de ativação, ganho igual a 1.

O valor da modificação do peso sináptico da RNA é proporcional a derivada da função de ativação, isto resulta que para uma função de ativação sigmóide, os pesos sinápticos são modificados mais intensamente para aqueles neurônios da RNA onde os valores dos sinais estão no meio do seu intervalo (HAYKIN, 2001, RUMELHART et al., 1986a).

Os pesos são aspectos cruciais para a RNA (BALKIN, 2000). Inicialmente são associados valores aleatórios em torno de zero. Os pesos são incrementalmente ajustados durante a fase de treinamento, inicializações típicas situam-se no intervalo [-1,1].

Pode-se avaliar o comportamento da função de ativação $\varphi(a_o)$ e sua derivada de primeira ordem $\varphi'(a_o)$ se, por exemplo, assumir-se que os pesos situem-se no intervalo sugerido

acima, que o bias tenha valor 1, o coeficiente α_0 seja um valor aleatório no intervalo $[0,1]$, v_t seja distribuído como $N(0;1)$ e que a série temporal tenha características de um modelo AR[1]-GARCH[1,1], como descrito em (16). Além disso coloca-se $g = 1$ em (17).

Se for considerada uma RNA 10-1-1 (ou seja, primeira camada com 10 sinais de entrada, um neurônio na camada intermediária e um neurônio na camada de saída) com função de ativação sigmóide, pode-se realizar uma simulação em que variam os parâmetros ϕ_1 , α_1 e β_1 da equação (3.76). O parâmetro α_1 , que está associado ao termo do erro quadrado e β_1 associado ao termo da volatilidade assumem valores aleatórios no intervalo $[0,1;0,2]$ para todas as simulações, ϕ_1 que é associado ao termo auto-regressivo variará em quatro etapas diferentes. Primeiro, será realizada a simulação com ϕ_1 assumindo como valor um número aleatório entre $[0,1;0,2]$. Na segunda simulação ϕ_1 é um valor aleatório no intervalo $[0,3;0,4]$. A terceira simulação tem ϕ_1 variando entre $[0,5;0,6]$. Na última simulação ϕ_1 varia no intervalo $[0,7;0,8]$.

O parâmetro ϕ_1 está ligado a estacionariedade da série temporal descrita por um processo AR(1):

$$y_t = \phi_0 + \phi_1 y_{t-1} + \varepsilon_t \quad (3.78)$$

Se $|\phi_1| < 1$ a seqüência $\{y_t\}$ será estacionária.

A medida que aumenta-se gradativamente o valor de ϕ_1 queremos observar o comportamento da função de ativação $\varphi(a_o)$ e de sua primeira derivada $\varphi'(a_o)$ para séries que estão se aproximando cada vez mais de modelos com características não estacionárias. Foram simuladas 1000 séries temporais AR(1)-GARCH(1,1) de tamanho 1000. Sendo que as primeiras 900 observações são utilizadas para treinamento da RNA. Como a rede tem 10 entradas e uma saída para cada série simulada foram calculadas 90 funções de ativação. Desta forma para as mil simulações foram calculadas 90000 funções de ativação de acordo com a equação (3.74) e o mesmo número de derivadas de primeira ordem da função de ativação dada por:

$$\varphi'(v_0) = \frac{\exp\left[-w_{o0}y_0 - \sum_{i=1}^m w_{oi} \left(\phi_1 y_{t-1} + v_t \sqrt{\alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2}\right)\right]}{\left[1 + \exp\left[-w_{o0}y_0 - \sum_{i=1}^m w_{oi} \left(\phi_1 y_{t-1} + v_t \sqrt{\alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2}\right)\right]\right]^2}. \quad (3.79)$$

Na Figura 3.12 tem-se os histogramas de $\varphi(a_0)$ e $\varphi'(a_0)$. Estes histogramas foram obtidos por meio da simulação dos parâmetros dos modelos. Estes parâmetros dos modelos AR(1)-GARCH(1,1) foram obtidos a partir de simulações de Monte de Carlo associados aos termos auto-regressivo ϕ_1 , erro quadrado α_1 e volatilidade β_1 . Em todas as simulações α_1 e β_1 variam no intervalo [0,1;0,2] e ϕ_1 varia nos intervalos: (a) [0,1;0,2], (b) [0,3;0,4], (c) [0,5;0,6] e (d) [0,7;0,8].

Na Figura 3.12a o histograma da função de ativação apresenta uma distribuição com relativa uniformidade em que os valores mais típicos estão concentrados em torno de 0,5. Para a derivada desta função, o histograma é assimétrico para a direita concentrando um maior número de resultados próximos do valor máximo 0,25.

Para as simulações apresentadas na Figura 3.12b, 3.12c e 3.12d pode-se observar a dinâmica da alteração da função de ativação e da derivada quando aumenta-se os valores do parâmetro auto-regressivo (ϕ_1). Aumentando o valor de ϕ_1 a distribuição dos valores obtidos para $\varphi(a_0)$ continua simétrica, porém ocorre um aumento na frequência em que esta variável assume valores próximos aos extremos. Isso vai se intensificando até que no caso (d) em que ϕ_1 varia no intervalo [0,7;0,8] obtem-se a maior concentração de resultados da função de ativação próximos de zero e um. A variação observada nos histogramas indica que aumentando o valor do parâmetro auto-regressivo $\varphi'(a_0)$ tem a sua concentração aumentada em valores próximos de zero.

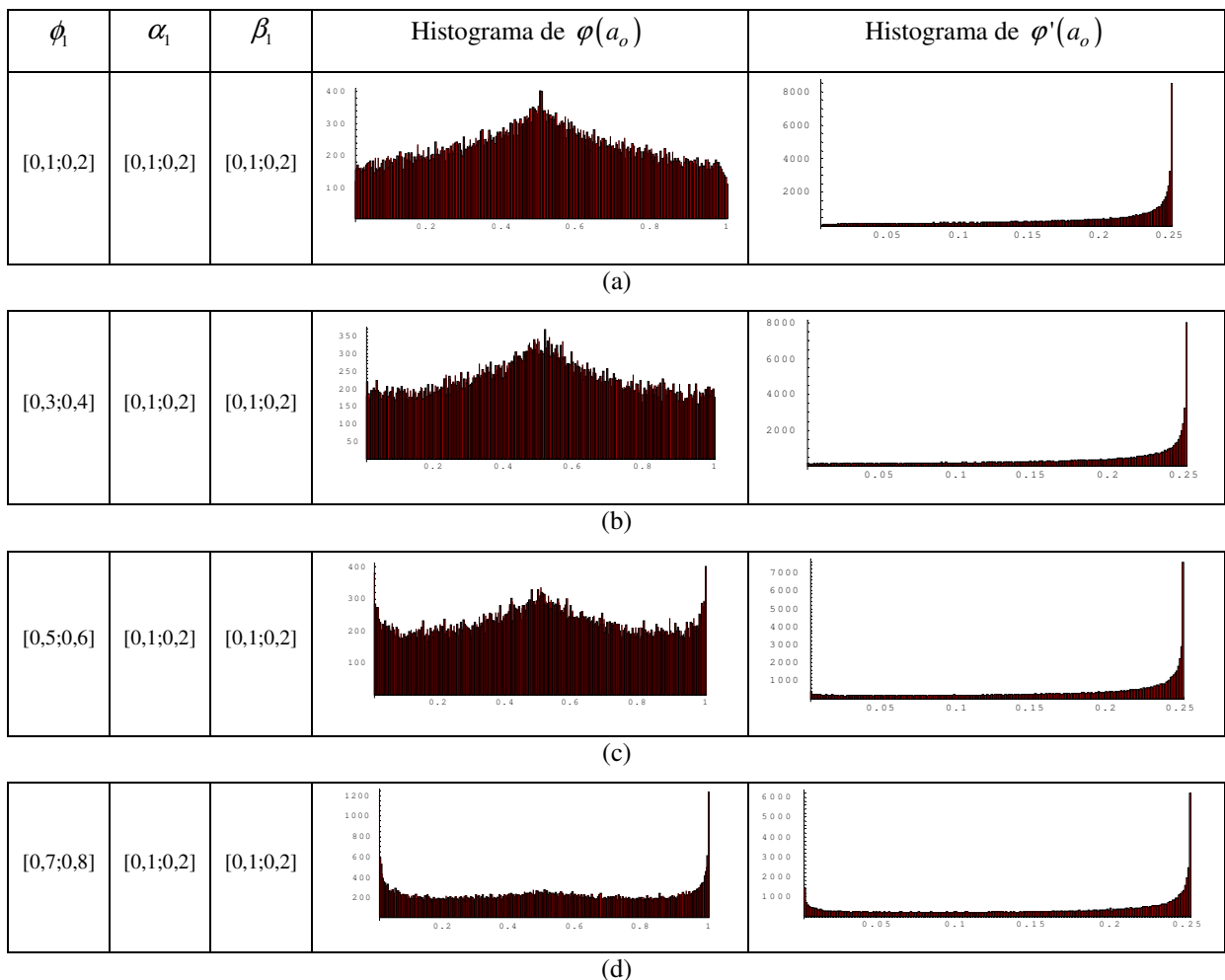


Figura 3.12 Histogramas de $\varphi(a_o)$ e $\varphi'(a_o)$. α_1 e β_1 variam no intervalo [0,1;0,2] e ϕ_1 varia nos intervalos: (a) [0,1;0,2], (b) [0,3;0,4], (c) [0,5;0,6] e (d) [0,7;0,8].

Na Figura 3.13 tem-se os histogramas de $\varphi(a_o)$ e $\varphi'(a_o)$. Estes histogramas foram obtidos por meio da simulação dos parâmetros dos modelos. Estes parâmetros dos modelos AR(1)-GARCH(1,1) foram obtidos a partir de simulações de Monte Carlo associados aos termos auto-regressivo ϕ_1 , erro quadrado α_1 e volatilidade β_1 . Em todas as simulações ϕ_1 e β_1 variam no intervalo [0,1;0,2] e α_1 varia nos intervalos: (a) [0,3;0,4], (b) [0,5;0,6] e (c) [0,7;0,8]. Os histogramas de $\varphi(a_o)$ e $\varphi'(a_o)$ possuem forma similar aos obtidos na Figura 4, ou seja quando aumentamos o parâmetro α_1 mantendo os demais parâmetros variando no intervalo [0,1;0,2] o histograma de $\varphi(a_o)$ reduz a concentração em torno de 0,5 e aumenta a

concentração de valores próximos aos valores extremos. O histograma de $\varphi'(a_0)$ reduz a concentração em torno do valor 0,25 e aumenta a concentração em torno do zero.

Na Figura 3.14 tem-se os histogramas de $\varphi(a_0)$ e $\varphi'(a_0)$. Estes histogramas foram obtidos por meio da simulação dos parâmetros dos modelos. Estes parâmetros dos modelos AR(1)-GARCH(1,1) foram obtidos a partir de simulações de Monte de Carlo associados aos termos auto-regressivo ϕ_1 , erro quadrado α_1 e volatilidade β_1 . Em todas as simulações ϕ_1 e α_1 variam no intervalo $[0,1;0,2]$ e β_1 varia nos intervalos: (a) $[0,3;0,4]$, (b) $[0,5;0,6]$ e (c) $[0,7;0,8]$. Os histogramas de $\varphi(a_0)$ e $\varphi'(a_0)$ possuem forma similar aos obtidos na Figura 3.14. Nota-se que o histograma de $\varphi(a_0)$ na Figura 6c tem-se a concentração de valores em torno de zero e um mais acentuada do que foi obtidos nas Figuras-3.12c e 3.12c. O mesmo comportamento é obtido para o histograma de $\varphi'(a_0)$.

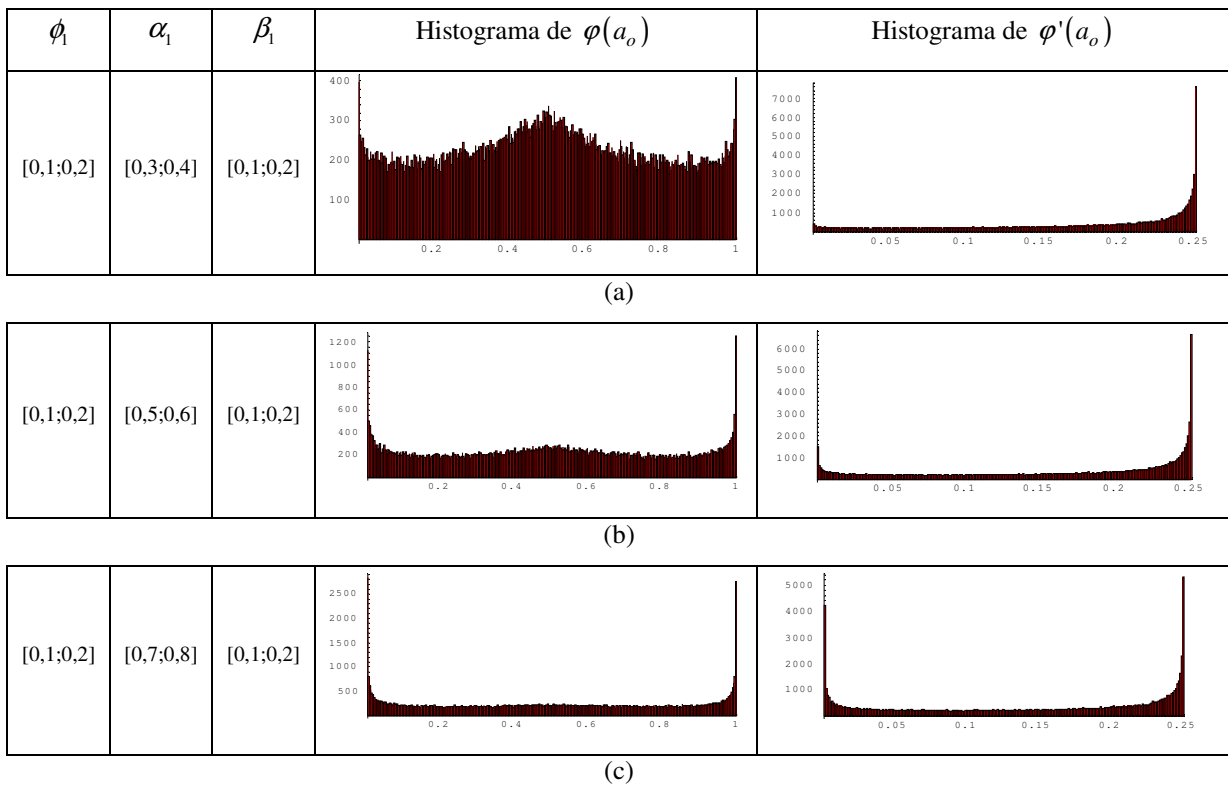


Figura 3.13 Histogramas de $\varphi(a_0)$ e $\varphi'(a_0)$. ϕ_1 e β_1 variam no intervalo $[0,1;0,2]$ e α_1 varia nos intervalos: (a) $[0,3;0,4]$, (b) $[0,5;0,6]$ e (c) $[0,7;0,8]$.

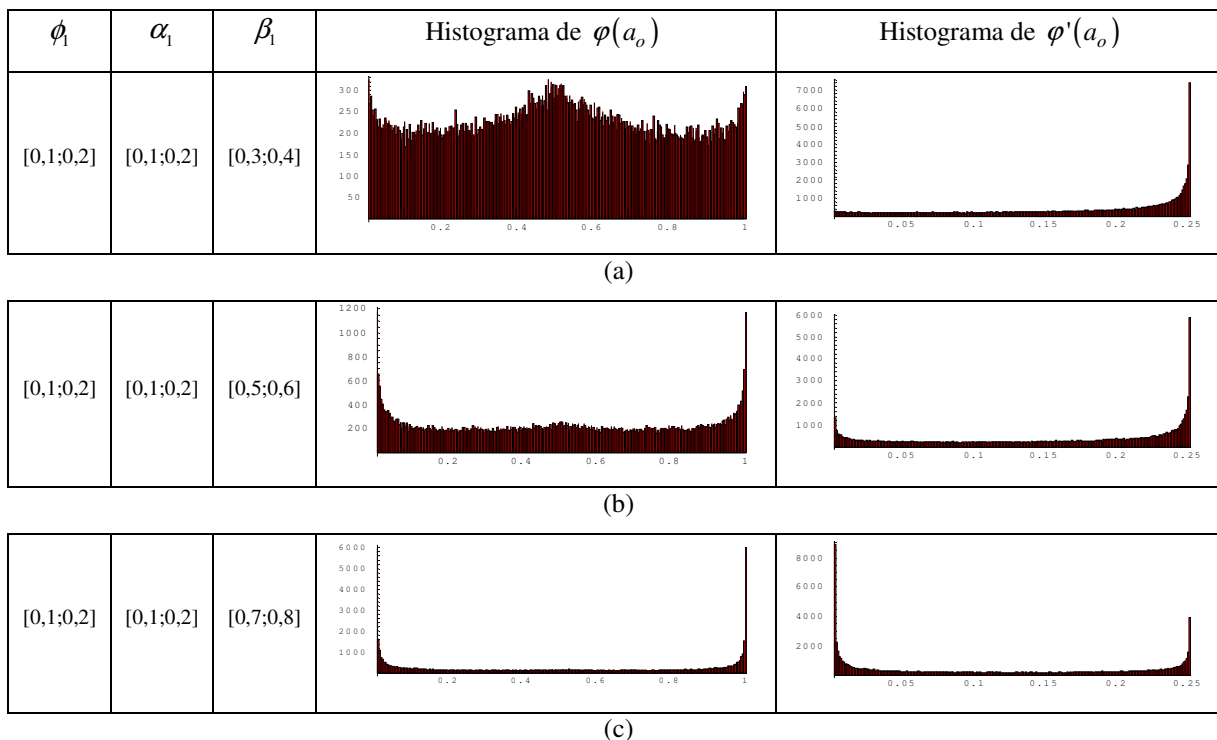


Figura 3.14 Histogramas de $\varphi(a_o)$ e $\varphi'(a_o)$. ϕ_1 e α_1 variam no intervalo [0,1;0,2] e β_1 varia nos intervalos: (a) [0,3;0,4], (b) [0,5;0,6] e (c) [0,7;0,8].

É sabido que grandes valores de α_1 e β_1 atuam de tal forma a aumentar a volatilidade condicional, mas fazem isso de forma diferente. Quando tem-se modelos com valores grandes de α_1 , o efeito deste choque é mais pronunciado no período subsequente. Em modelos em que β_1 assume maiores valores, sua variância condicional apresenta mais persistência auto-regressiva (ENDERS, 2004, HAMILTON, 1994, MORETTIN, 2004).

3.12 Resultados de previsões de séries AR-GARCH para diferentes topologias de RNA

Nesta seção é realizada a previsão de um passo a frente de processos AR[1]-GARCH[1,1] com diferentes combinações de parâmetros, usando cinco topologias de RNA feedforward. Cada um dos parâmetros do modelo AR(1)-GARCH(1,1) (ϕ_1 , α_1 , β_1), individualmente, irá variar nos intervalos: [0,1;0,2], [0,3;0,4], [0,5;0,6] e [0,7;0,8]. Desta forma, tem-se as 64 combinações de parâmetros apresentadas na Figura 3.15. Para cada combinação serão geradas via Monte Carlo 100 séries com tamanho 1000. As séries são divididas em duas partes, uma para treinamento constituindo de 900 valores e os 100 valores restantes serão utilizados para realizar a previsão. Esta estratégia de segmentação da série temporal é comum na análise de séries temporais, a primeira parte, contendo um maior número de observações, é usada para estimar o modelo e as observações finais são utilizadas para medir a capacidade de previsão do modelo estimado.

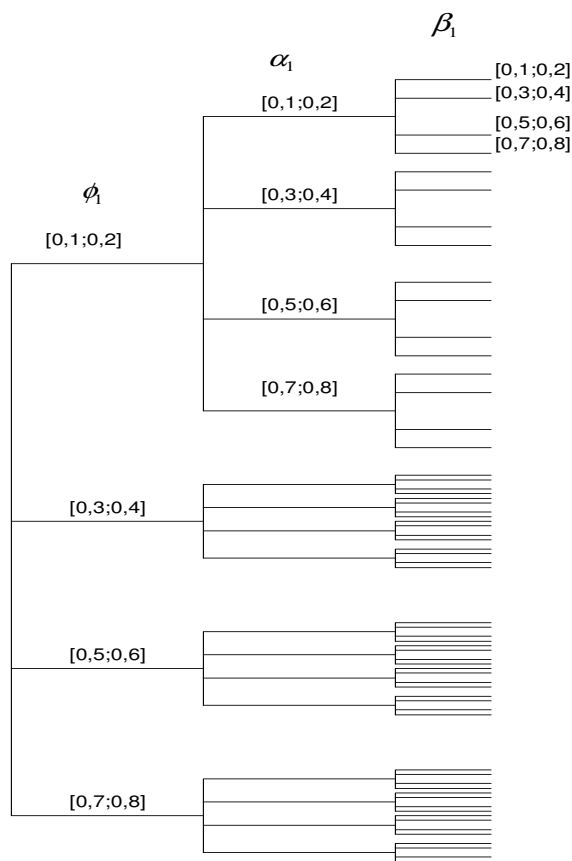


Figura 3.15 Árvore de combinação de parâmetros para simulação das séries AR[1]-GARCH[1,1].

Os valores observados e ajustados serão comparados pelo coeficiente de desigualdade de Theil (TIC) dado pela expressão (11). Este procedimento será realizado para redes neurais com as seguintes topologias 10-1-1, 10-3-1, 10-5-1, 10-7-1 e 10-9-1, função de ativação do tipo sigmóide e algoritmo de treinamento de Levenberg-Marquardt. Os dados na maioria dos casos necessitam ser normalizados, a exceção é para dados que em certo sentido já estão normalizados, tal como dados binários ou dados que compõem a mesma série temporal e são apresentados no mesmo formato (RECH, 2002). Neste parte não é realizada a normalização dos valores das séries temporais. Os valores de entrada da RNA são os mesmos obtidos pela simulação.

Primeiro caso: previsões obtidas pela RNA 10-1-1.

A Figura 3.16 apresenta o Box-plot e gráfico de linha dos coeficientes TIC para os 100 valores preditos pela RNA considerando α_1 e β_1 variando no intervalo $[0,1;0,2]$ e ϕ_1 variando nos intervalos $[0,1;0,2]$, $[0,3;0,4]$, $[0,5;0,6]$ e $[0,7;0,8]$. A medida que aumenta-se o valor do parâmetro ϕ_1 que está diretamente relacionado à estacionariedade da série, o coeficiente de desigualdade de Theil diminui. Ou seja, mantendo os parâmetros α_1 e β_1 variando no intervalo $[0,1;0,2]$, a previsão realizada pela RNA um passo à frente torna-se cada vez melhor quando o parâmetro ϕ_1 da série aumenta.

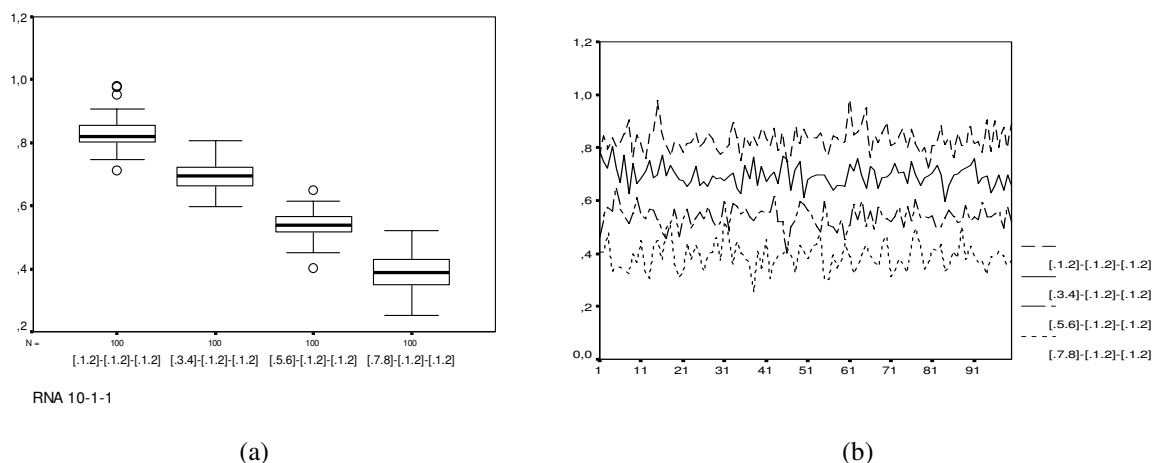
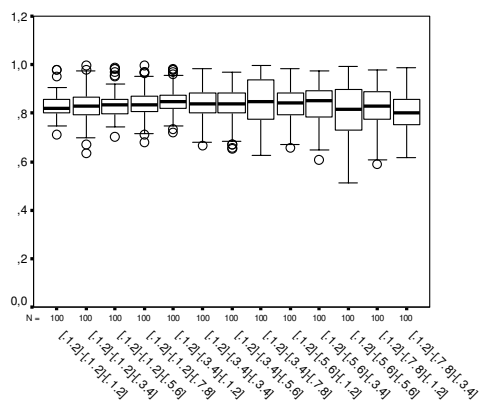


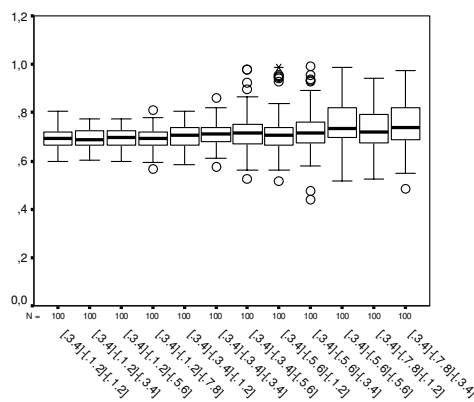
Figura 3.16 (a) Box-plot do TIC para a RNA 10-1-1 considerando aumento gradativo de ϕ_1 , (b) TIC de acordo com o aumento de ϕ_1 para a RNA 10-1-1.

Se for considerado que o parâmetro ϕ_1 continue variando no intervalo $[0,1;0,2]$ e que os parâmetros α_1 e β_1 variem, individualmente, dentro das várias possibilidades do primeiro ramo da árvore de combinações da Figura 3.15, pode-se ver na Figura 3.17a que a maior concentração do TIC para cada combinação dos parâmetros oscilou em torno de um valor próximo de 0,8, indicando que as predições não foram boas. Na Figura 3.17b são apresentados os box-plot relacionados aos coeficientes TIC originados pelas predições em que o parâmetro ϕ_1 varia no intervalo $[0,3;0,4]$ e α_1 e β_1 assumem, individualmente, valores de acordo com as combinações possíveis pré-definidas. Nota-se que os valores do TIC associados a estas predições concentraram-se em torno de 0,7, ou seja, foram menores quando comparados aos obtidos na Figura 3.17a.

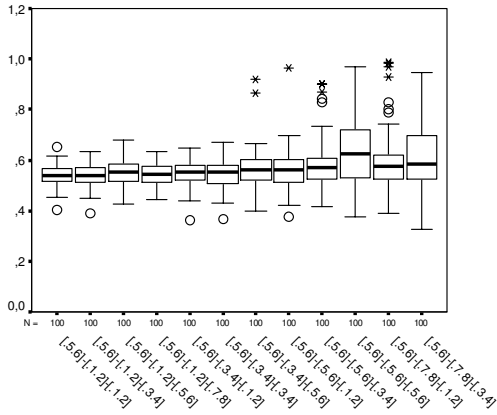
Nos dois casos seguintes, mostrados na Figura 3.17c e 3.17d em que o parâmetro ϕ_1 varia nos intervalos de $[0,5;0,6]$ e de $[0,7;0,8]$, respectivamente, tem-se que o TIC das predições continuou em sua tendência de declínio. Esse comportamento leva à conclusão que para uma rede neural com dez neurônios na camada de entrada, um neurônio na camada intermediária, função de ativação sigmóide e um neurônio na camada de saída, a RNA realiza melhores predições quando a série tem parâmetro ϕ_1 entre $[0,7;0,8]$, exceto para as seguintes combinações de parâmetros: ϕ_1 entre $[0,7;0,8]$, α_1 entre $[0,3;0,4]$ e β_1 entre $[0,7;0,8]$; ϕ_1 entre $[0,7;0,8]$, α_1 entre $[0,5;0,6]$ e β_1 entre $[0,5;0,6]$; ϕ_1 entre $[0,7;0,8]$, α_1 entre $[0,5;0,6]$ e β_1 entre $[0,7;0,8]$; ϕ_1 entre $[0,7;0,8]$, α_1 entre $[0,7;0,8]$ e β_1 entre $[0,5;0,6]$; ϕ_1 entre $[0,7;0,8]$, α_1 entre $[0,7;0,8]$ e β_1 entre $[0,7;0,8]$.



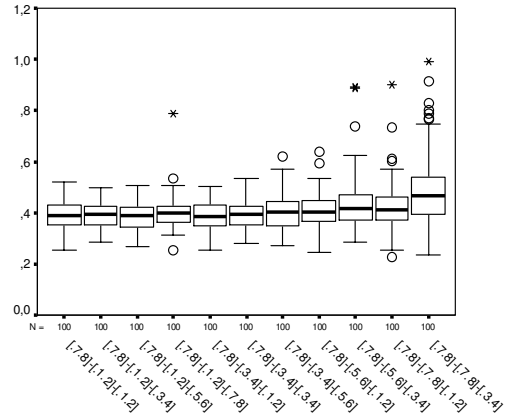
(a)



(b)



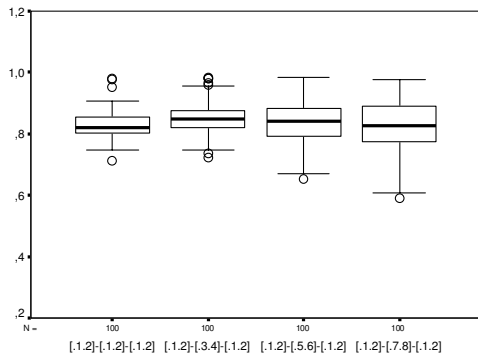
(c)



(d)

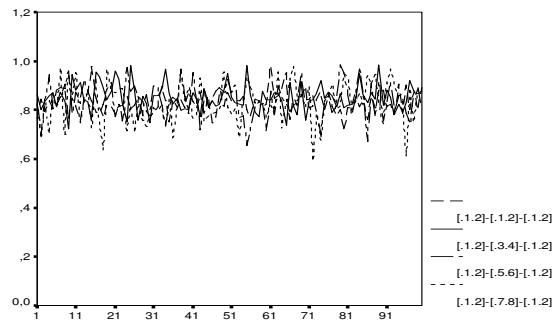
Figura 3.17 (a) Box-plot do TIC para a RNA 10-1-1 considerando ϕ_1 variando no intervalo $[0,1;0,2]$ e várias combinações de α_1 e β_1 , (b) ϕ_1 varia no intervalo $[0,3;0,4]$, (c) ϕ_1 varia no intervalo $[0,5;0,6]$, (d) ϕ_1 varia no intervalo $[0,7;0,8]$.

É importante ressaltar que, nas simulações realizadas, em séries em que o parâmetro relacionado a volatilidade podia assumir valores no intervalo $[0,7;0,8]$ se o parâmetro ϕ_1 ou α_1 também estivessem nesse intervalo ou no intervalo $[0,5;0,6]$ a variância apresentava persistência auto-regressiva acentuada, e nestes casos não é adequada a utilização desta topologia de RNA para realizar previsão. Pode-se verificar, na Figura 3.18, que aumentando gradativamente o valor do parâmetro α_1 as previsões de um passo à frente da RNA 10-1-1 não tem melhora significativamente como ocorreu quando aumenta-se os valores de ϕ . Os valores do TIC obtidos ficam oscilando em torno de 0,8.



RNA 10-1-1

(a)



(b)

Figura 3.18 (a) Box-plot do TIC considerando aumento gradativo de α_1 , (b) TIC com o aumento de α_1 .

Quando é alterado o parâmetro ligado a volatilidade, β_1 , e mantendo ϕ_1 e α_1 variando simultaneamente no intervalo $[0,1;0,2]$, ver Figura 3.19, verifica-se que as previsões de acordo com o TIC não tiveram melhora, pois seus valores oscilaram em torno de 0,8.

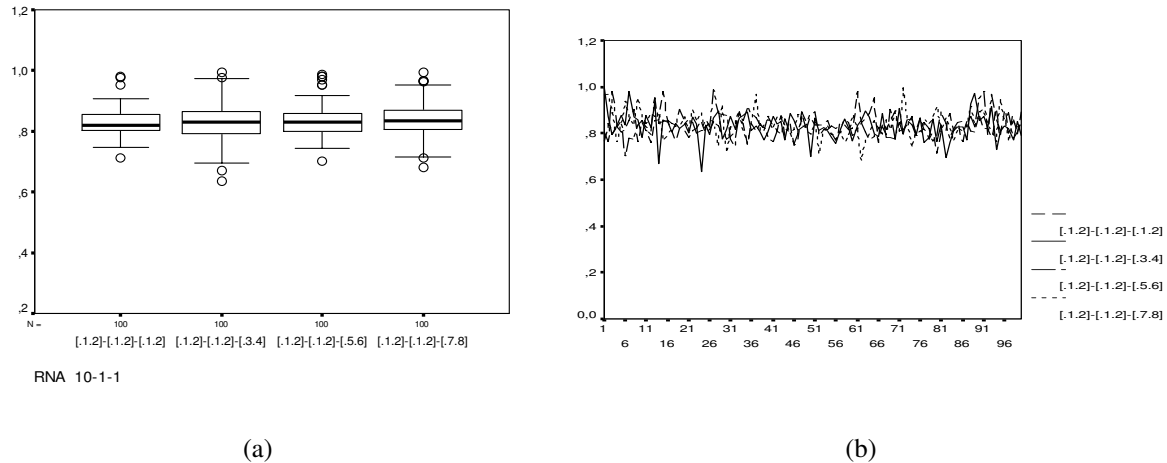


Figura 3.19 (a) Box-plot do TIC para a RNA 10-1-1 considerando aumento gradativo da volatilidade β_1 ,
 (b) variação do TIC de acordo com o aumento de β_1 para a RNA 10-1-1.

Segundo caso: previsões obtidos pelas RNA 10-3-1, 10-5-1, 10-7-1 e 10-9-1.

Neste segundo caso são apresentados os resultados obtidos pelas redes neurais com três, cinco, sete e nove neurônios na camada intermediária.

Na Figura 3.20 são apresentados o box-plot do coeficiente TIC para as topologias RNA 10-3-1, 10-5-1, 10-7-1 e 10-9-1 com os parâmetros variando nos intervalos $[0,1;0,2]$, $[0,3;0,4]$, $[0,5;0,6]$ e $[0,7;0,8]$. Pela figura pode-se identificar quatro agrupamentos em cada uma dessas topologias. A Figura 3.20a corresponde aos resultados de ajuste de previsão obtidos pela RNA 10-3-1 seguindo as combinações de parâmetros ϕ_1 , α_1 e β_1 . O primeiro agrupamento de box-plot da Figura 3.20a corresponde ao primeiro ramo mostrado na árvore de combinações de parâmetros da Figura 7 onde tem-se ϕ_1 variando em $[0,1;0,2]$ com as várias combinações de α_1 e β_1 , no segundo tem-se que ϕ_1 varia em $[0,3;0,4]$, no terceiro agrupamento ϕ_1 está variando entre $[0,5;0,6]$ e no quarto agrupamento ϕ_1 varia em $[0,7;0,8]$. Novamente aqui é preciso dizer que foram removidas as combinações de parâmetros que implicaram em séries com efeito auto-regressivo da volatilidade muito acentuado. Pode-se ver que para as quatro topologias de rede o mesmo efeito de melhora de ajuste foi obtido com o aumento do parâmetro ϕ_1 .

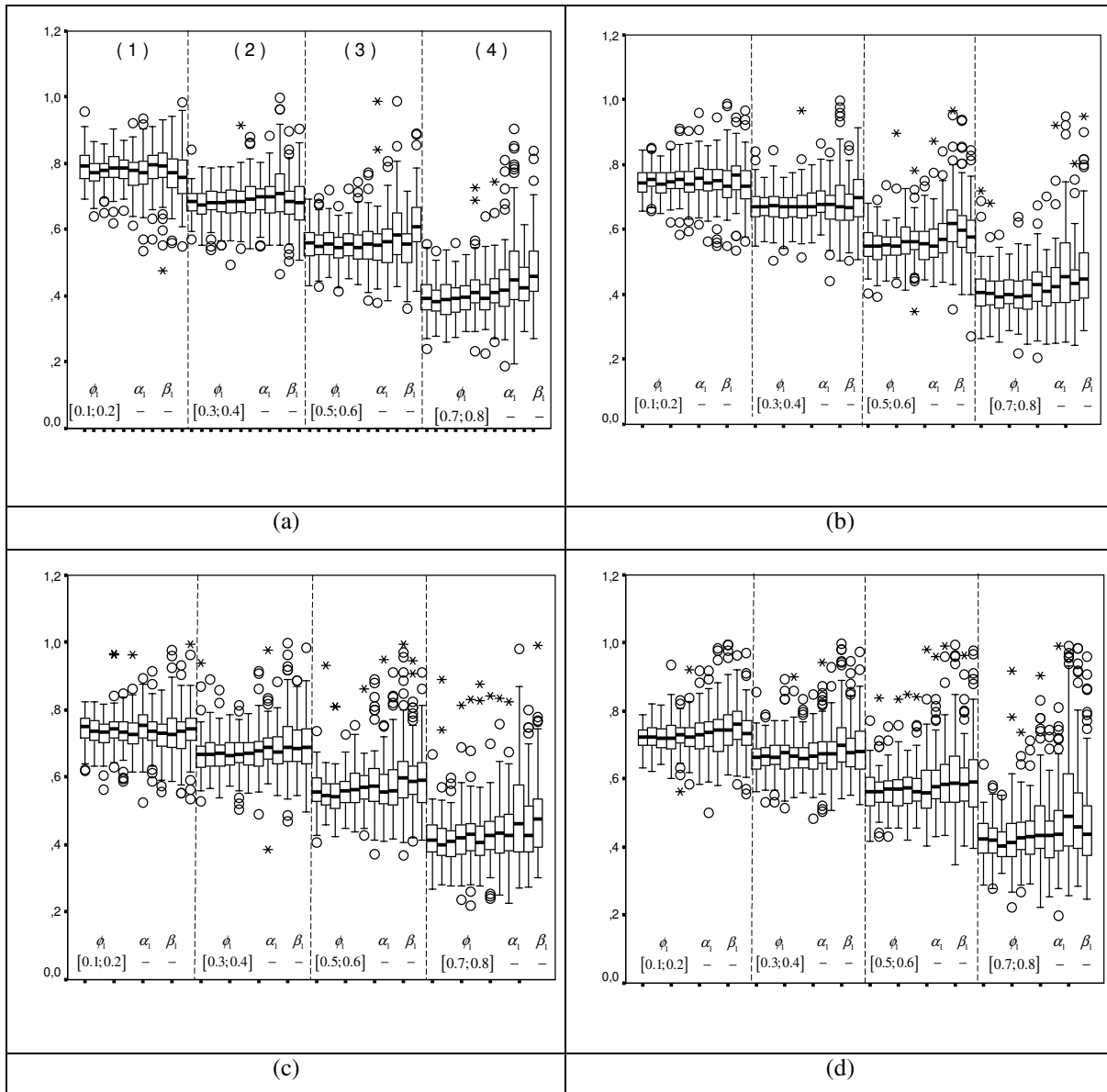


Figura 3.20 (a) Box-plot do TIC para a RNA 10-3-1, (b) Box-plot do TIC para a RNA 10-5-1, (c) Box-plot do TIC para a RNA 10-7-1, (d) Box-plot do TIC para a RNA 10-9-1.

Dos resultados obtidos é analisado se a rede com maior número de neurônios na camada intermediária mostrou desempenho de predição melhor do que a rede com um neurônio nessa camada. Inicialmente, é analisado qual foi o comportamento de predição para o caso em que o parâmetro ϕ_1 ligado a estacionariedade torna-se cada vez maior, a Figura 3.21 apresenta as distribuições dos coeficientes de desigualdade de Theil dessa situação para a rede 10-9-1. Verifica-se que a melhora do ajuste de predição continua acontecendo quando é aumentado o ϕ_1 , mantendo os termos de erro quadrado e volatilidade abaixo de 0,2.

Comparando os resultados obtidos pela RNA 10-9-1, ver Figura 3.21, com aqueles da RNA 10-1-1, ver Figura 3.16, nota-se que com ϕ_1 abaixo de 0,4 a rede com nove neurônios na camada intermediária apresentou menores TIC.

Na Tabela 3.3 tem-se os resultados dos percentis de 25, 50 e 75 % obtidos pela RNA 10-9-1 e pela RNA 10-1-1. Verifica-se que com ϕ_1 acima de 0,5 a RNA 10-1-1 teve desempenho de predição melhor que a RNA 10-9-1.

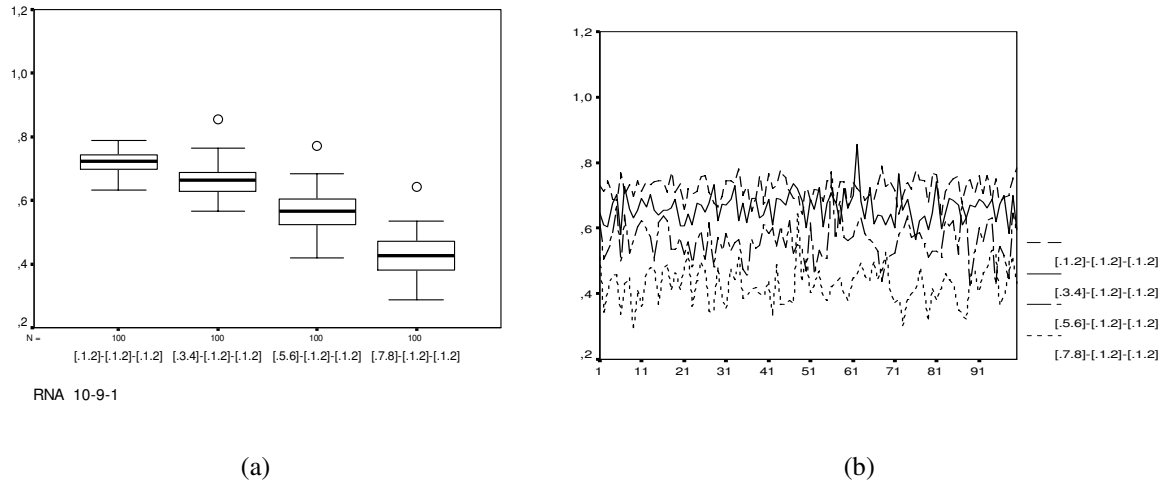


Figura 3.21 (a) Box-plot do TIC para a RNA 10-9-1 considerando aumento gradativo de ϕ_1 , (b) variação do TIC de acordo com o aumento de ϕ_1 para a RNA 10-9-1.

Foram construídas, por simulação, 32000 séries temporais AR(1)-GARCH(1,1) de tamanho 1000 em que os 100 últimos valores são utilizados para testar o poder de predição de cinco tipos de topologias de redes neurais alimentadas adiante. A Figura 3.22 apresenta quatro dessas séries com as respectivas predições realizadas usando a RNA 10-1-1 e a RNA 10-9-1. Claramente pode-se notar que em séries com ϕ_1 assumindo valores entre [0,7;0,8], mantendo α_1 e β_1 menores que 0,2, as RNA realizam predições melhores. Na primeira das séries mostradas na Figura 3.22 o TIC obtido foi de 0,7475 e na segunda série, em que a previsão foi feita com um número maior de neurônios na camada intermediária, o TIC foi 0,7454, ou seja houve uma melhora de apenas 0,27% no ajuste. Para todas as séries simuladas também foi calculado a raiz do erro quadrático médio – root mean square error, na primeira série o RMSE calculado foi de 0,416017 e na segunda série foi encontrado um RMSE de 0,397204. As duas últimas séries da Figura 3.22 apresentaram, respectivamente,

os seguintes coeficientes de Theil 0,448021 e 0,337136. Quanto ao RMSE para a penúltima série foi encontrado 0,336322 e para a última série 0,312307.

Intervalo de Variação dos Parâmetros				
	ϕ_1 [0,1;0,2]	ϕ_1 [0,3;0,4]	ϕ_1 [0,5;0,6]	ϕ_1 [0,7;0,8]
	α_1 [0,1;0,2]	α_1 [0,1;0,2]	α_1 [0,1;0,2]	α_1 [0,1;0,2]
	β_1 [0,1;0,2]	β_1 [0,1;0,2]	β_1 [0,1;0,2]	β_1 [0,1;0,2]
RNA 10-1-1				
Percentil 25	0,8019	0,6643	0,5175	0,3513
50	0,8202	0,6938	0,5385	0,3892
75	0,8546	0,7218	0,5670	0,4302
RNA 10-9-1				
Percentil 25	0,6977	0,6290	0,5232	0,3819
50	0,7224	0,6632	0,5642	0,4257
75	0,7454	0,6879	0,6055	0,4702

Tabela 3.3 Percentis do TIC obtidos pela RNA 10-1-1 e 10-9-1.

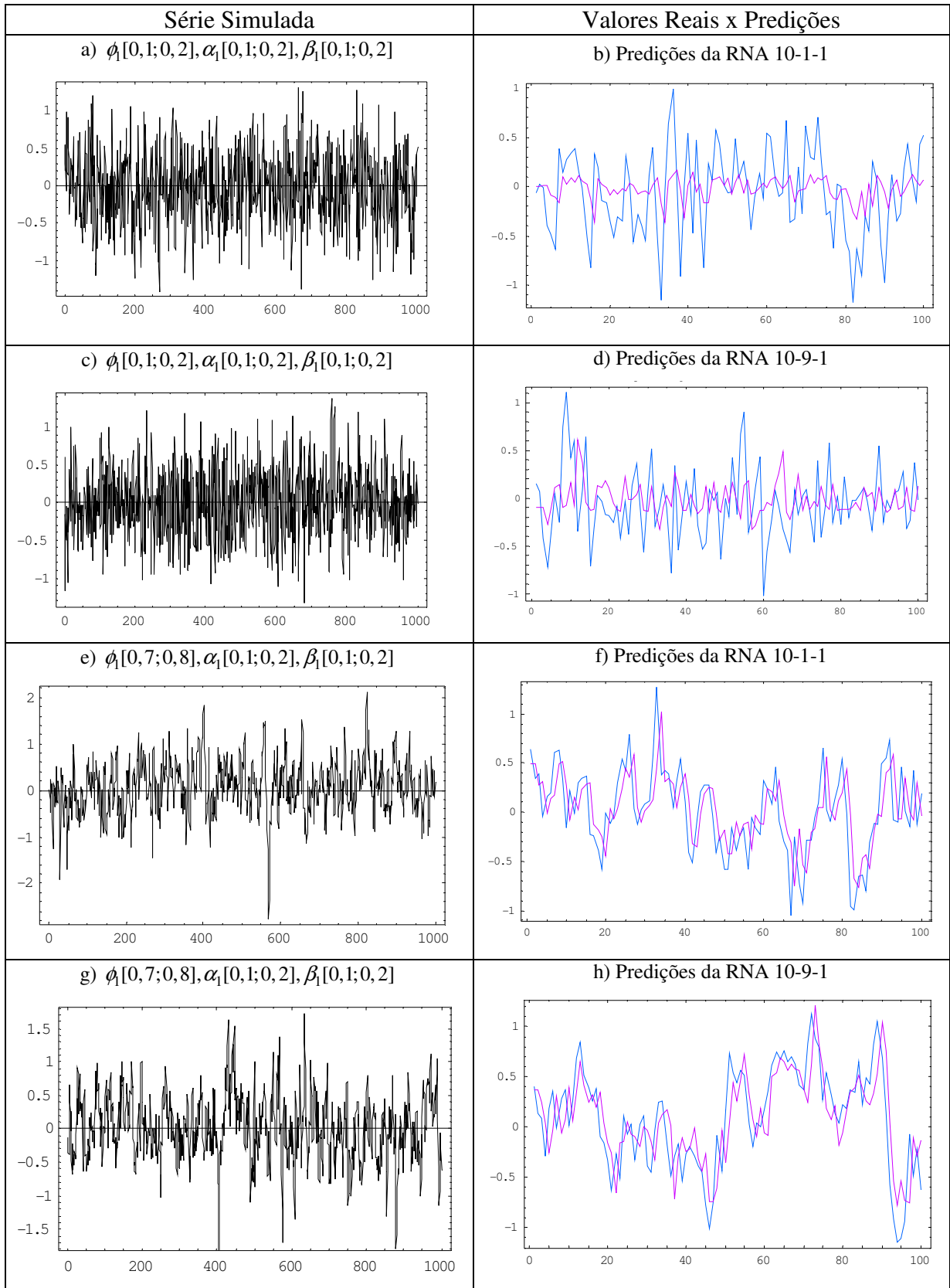


Figura 3.22 Quatro séries AR(1)-GARCH(1,1) geradas por simulação e previsões (linhas rosas) versus valores reais (linhas azuis).

Capítulo 4

Filtro de Kalman

4.1 Filtro de Kalman

O filtro de Kalman é um estimador ótimo, infere parâmetros de interesse a partir de observações indiretas, imprecisas e incertas. Alguns dos motivos que levam o filtro de Kalman a ser uma das maiores contribuições na teoria moderna de controle, incluindo a análise de séries temporais, são: bons resultados obtidos na prática devido a otimalidade e estrutura, apresentar uma forma conveniente para processamento em tempo real, fácil para formular e implementar, as equações de medida necessárias não são invertidas, etc. Assumindo um sistema linear e que o ruído é branco e Gaussiano, o filtro de Kalman é ótimo no sentido em que o erro quadrático médio (MSE) é minimizado. O filtro de Kalman consiste em um conjunto de equações matemáticas que computam recursivamente o estado estimado de um sistema dinâmico linear no tempo t , \mathbf{x}_t , partindo das estimativas prévias do estado no tempo $t - 1$, \mathbf{x}_{t-1} e dos novos dados de entrada \mathbf{y}_t . Uma vez que ele é recursivo, não há necessidade de armazenar todos os dados passados observados para estimar o estado do sistema. Esta é uma propriedade importante e permite a possibilidade de implementação em tempo real.

O estado de um sistema é simplesmente um vetor coluna contendo variáveis que são de interesse do analista. Frequentemente, estas variáveis (ou, os estados) não podem ser observadas diretamente. Pelo contrário, elas devem ser inferidas de um conjunto de medidas (muito provavelmente contaminadas com ruído) tomadas do sistema. Usando o filtro de Kalman estendido para treinar uma rede neural o vetor de estado deverá consistir dos pesos sinápticos da rede. Estas são as variáveis pelas quais tem-se interesse. Este é o

objetivo do filtro de Kalman, usar medidas tomadas do sistema para deduzir as melhores estimativas para as variáveis de estado (PATEL, 2000).

Para alicerçar o método do filtro de Kalman, considere o modelo de espaço de estado linear de um sistema dinâmico de tempo discreto descrito pelas seguintes duas equações:

$$\mathbf{y}_t = \mathbf{H}_t \mathbf{x}_t + \mathbf{r}_t, \quad (4.1)$$

$$\mathbf{x}_{t+1} = \mathbf{F}_{t+1,t} \mathbf{x}_t + \mathbf{q}_t, \quad (4.2)$$

sendo que,

- \mathbf{y}_t : $M \times 1$ vetor de medida no tempo t
- \mathbf{H}_t : $M \times N$ matriz de medida no tempo t
- \mathbf{r}_t : $M \times 1$ ruído de medida no tempo t
- \mathbf{R}_t : $M \times M$ matriz de covariância do ruído de medida
- \mathbf{x}_t : $N \times 1$ estado do sistema no tempo t
- \mathbf{x}_{t+1} : $N \times 1$ estado do sistema no tempo $t + 1$
- $\mathbf{F}_{t+1,t}$: $N \times N$ matriz de transição do estado do tempo t para o $t + 1$
- \mathbf{q}_t : $N \times 1$ ruído do processo no tempo t
- \mathbf{Q}_t : $N \times N$ matriz de covariância do ruído do processo

As Equações (4.1) e (4.2) são conhecidas como equações de medida e de transição, respectivamente. Nas Equações (4.1) e (4.2), tanto \mathbf{r}_t como \mathbf{q}_t são assumidos como sendo seqüências de ruído branco de média zero com matrizes de covariância diagonal (ou correlação) \mathbf{R}_t e \mathbf{Q}_t respectivamente, descritas como:

$$\mathbf{R}_t = E[\mathbf{r}_t \mathbf{r}_t^T], \quad (4.3)$$

$$\mathbf{Q}_t = E[\mathbf{q}_t \mathbf{q}_t^T]. \quad (4.4)$$

Na formulação do filtro de Kalman é assumido que o sistema sob consideração pode ser descrito pelo modelo de espaço de estado acima e que os parâmetros do modelo $\mathbf{F}_{t+1,t}$, \mathbf{H}_t , \mathbf{Q}_t e \mathbf{R}_t são conhecidos. Na prática, \mathbf{Q}_t e \mathbf{R}_t podem não ser conhecidos mas devem

também ser estimados considerando a informação a *priori* disponível ou serem aproximadas pela experiência específica que se tem sobre a aplicação (PATEL, 2000).

Computar o ganho de Kalman

$$\mathbf{K}_t = \frac{\mathbf{P}_{t|t-1} \mathbf{H}_t^T}{\mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^T + \mathbf{R}_t} \quad (4.5)$$

Atualizar (corrigir) Medidas

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t \left[\mathbf{y}_t - \mathbf{H}_t \hat{\mathbf{x}}_{t|t-1} \right] \quad (4.6)$$

$$\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{K}_t \mathbf{H}_t \mathbf{P}_{t|t-1} \quad (4.7)$$

Atualizar tempo (predição)

$$\hat{\mathbf{x}}_{t+1|t} = \mathbf{F}_{t+1|t} \hat{\mathbf{x}}_{t|t} \quad (4.8)$$

$$\mathbf{P}_{t+1|t} = \mathbf{F}_{t+1,t} \mathbf{P}_{t|t} \mathbf{F}_{t+1,t}^T + \mathbf{Q}_t \quad (4.9)$$

Tabela 4.1 Equações do filtro de Kalman

As equações do filtro de Kalman na Tabela 4.1 podem ser classificadas em dois grupos: equações de atualização de tempo e equações de atualização de medida. Estas equações mantêm ligadas as estimativas do estado $\hat{\mathbf{x}}$ e a estimativa da matriz de correlação estado-erro \mathbf{P} . As equações de atualização de medida corrigem as estimativas baseadas nas novas medidas \mathbf{y}_t que é feita disponível no passo t . As equações de atualização de tempo predizem valores das estimativas no futuro (um passo à frente) antes que a próxima medida seja tornada disponível. Este processo é repetido recursivamente até a estimativa de estado convergir.

Com a utilização desta notação tem-se então que, \mathbf{x}_t é o estado verdadeiro do sistema a ser determinado, $\hat{\mathbf{x}}_{t|t-1}$ refere-se a estimativa de estado a *priori* no passo t dado o conhecimento das medidas até o passo $t - 1$, e $\hat{\mathbf{x}}_{t|t}$ refere-se à estimativa de estado a *posteriori* no passo t dadas as medidas até o passo t . Similarmente, $\mathbf{P}_{t|t-1}$ é a matriz de correlação estado-erro a *priori* e $\mathbf{P}_{t|t}$ é a matriz de correlação estado-erro a *posteriori*, descrita como segue:

$$\mathbf{P}_{t|t-1} = E \left[\left(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t-1} \right) \left(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t-1} \right)^T \right], \quad (4.10)$$

$$\mathbf{P}_{t|t} = E \left[\left(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t} \right) \left(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t} \right)^T \right]. \quad (4.11)$$

Para iniciar o procedimento de interação, descrito acima, a estimativa de estado *a priori* e a estimativa de covariância estado-erro *a priori* devem primeiro ser inicializadas com valores apropriados, $\hat{\mathbf{x}}_{1|0}$ e $\mathbf{P}_{1|0}$, respectivamente.

No centro do filtro de Kalman está a noção que quando uma nova observação ou medida é tornada disponível, tanto o vetor de estado *a priori* $\hat{\mathbf{x}}_{t|t-1}$ como a matriz de covariância estado-erro *a priori* $\mathbf{P}_{t|t-1}$ são atualizadas. Isto é realizado via as equações de atualização de medida. As equações de atualização de medida obtêm as estimativas *a posteriori* das estimativas *a priori* baseadas nas novas medidas \mathbf{y}_t no passo t . O termo de diferença $\mathbf{y}_t - \mathbf{H}_t \hat{\mathbf{x}}_{t|t-1}$ na Equação (4.6) é conhecido como o *vetor de inovações*, uma vez ele representa novas informações na medida atual \mathbf{y}_t quando comparada a medida predita $\mathbf{H}_t \hat{\mathbf{x}}_{t|t-1}$ (HAYKIN, 1996). Próximo, as equações de atualização de tempo predizem as estimativas *a priori* no passo $t + 1$ baseadas nas estimativas *a posteriori* nos passos prévios t e então o procedimento inteiro é repetido. Juntas, as Equações (4.7) e (4.9) são chamadas equações de diferença finita de Riccati e elas computam recursivamente a matriz de correlação estado-erro \mathbf{P} (HAYKIN, 1996).

Este procedimento recursivo direto e de atualização leva à minimização do traço da matriz de covariância estado-erro \mathbf{P} . Uma vez que as diagonais da matriz \mathbf{P} são as médias quadradas dos erros no estado, como é dado pelas Equações (4.10) e (4.11), isto significa que o filtro de Kalman encontra a estimativa de erro quadrático mínimo do estado.

A entrada para o filtro de Kalman é o vetor de medida no tempo t , \mathbf{y}_t . Portanto, para operar o filtro de Kalman a cada passo de tempo, a medida correta deve ser tornada disponível. Por essa razão, pode-se dizer que o filtro de Kalman é de fato um algoritmo supervisionado.

4.2 Filtro de Kalman Estendido

Há duas suposições básicas na derivação do filtro de Kalman. A primeira é que o sistema é descrito por um modelo de espaço de estado linear e a segunda é que os ruídos são brancos e Gaussianos com média zero, e eles também são assumidos serem não correlacionados um com o outro e com o estado inicial. Quando estas suposições são satisfeitas, o filtro de Kalman é ótimo no sentido do erro quadrático médio. Quando o sistema sob consideração é não linear, a primeira condição é violada e o filtro de Kalman estendido (EKF – *extended Kalman filter*) é aplicado como um filtro sub-ótimo. No EKF os termos não lineares são aproximados pelos termos lineares de primeira ordem pela expansão em uma série de Taylor. Para começar, considere um sistema não linear descrito como:

$$\mathbf{y}_t = h_t \mathbf{x}_t + \mathbf{r}_t, \quad (4.12)$$

$$\mathbf{x}_{t+1} = f_{t+1,t} \mathbf{x}_t + \mathbf{q}_t. \quad (4.13)$$

As Equações (4.12) e (4.13) são as contra-partidas não lineares das Equações (4.1) e (4.2). Elas são as equações de medida e processo para o caso não linear, onde, h e f são funções de vetores não lineares do estado. Procedendo tal como foi afirmado acima, são linearizadas ambas as equações usando a expansão por série de Taylor sobre $\hat{\mathbf{x}}_{t|t-1}$ e $\hat{\mathbf{x}}_{t|t}$ como segue, respectivamente:

$$h_t(\mathbf{x}_t) = h_t(\hat{\mathbf{x}}_{t|t-1}) + \mathbf{H}_t(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t-1}) + \dots, \quad (4.14)$$

$$f_t(\mathbf{x}_t) = f_t(\hat{\mathbf{x}}_{t|t}) + \mathbf{F}_{t+1,t}(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t}) + \dots, \quad (4.15)$$

em que as matrizes Jacobianas $\mathbf{F}_{t+1,t}$ e \mathbf{H}_t são definidas como:

$$\mathbf{F}_{t+1,t} = \frac{\partial f_t(\hat{\mathbf{x}}_{t|t})}{\partial \mathbf{x}} \quad \text{e} \quad \mathbf{H}_t = \frac{\partial h_t(\hat{\mathbf{x}}_{t|t-1})}{\partial \mathbf{x}}.$$

Ignorando os termos de maior ordem nas expansões de Taylor acima, as equações de medida e processo não lineares podem ser aproximadas como segue:

$$\mathbf{y}_t = \mathbf{H}_t \mathbf{x}_t + \mathbf{u}_t + \mathbf{r}_t, \quad (4.16)$$

$$\mathbf{x}_{t+1} = \mathbf{F}_{t+1,t} \mathbf{x}_t + \mathbf{v}_t + \mathbf{q}_t, \quad (4.17)$$

sendo que:

$$\mathbf{u}_t = h_t(\hat{\mathbf{x}}_{t|t-1}) - \mathbf{H}_t \hat{\mathbf{x}}_{t|t-1},$$

$$\mathbf{v}_t = f_t(\hat{\mathbf{x}}_{t|t-1}) - \mathbf{F}_{t+1,t} \hat{\mathbf{x}}_{t|t-1}.$$

Computar o ganho de Kalman

$$\mathbf{K}_t = \frac{\mathbf{P}_{t|t-1} \mathbf{H}_t^T}{\mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^T + \mathbf{R}_t} \quad (4.18)$$

Atualizar (corrigir) Medidas

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t \left[\mathbf{y}_t - h_t \hat{\mathbf{x}}_{t|t-1} \right] \quad (4.19)$$

$$\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{K}_t \mathbf{H}_t \mathbf{P}_{t|t-1} \quad (4.20)$$

Atualizar tempo (predição)

$$\hat{\mathbf{x}}_{t+1|t} = f_t \hat{\mathbf{x}}_{t|t} \quad (4.21)$$

$$\mathbf{P}_{t+1|t} = \mathbf{F}_{t+1,t} \mathbf{P}_{t|t} \mathbf{F}_{t+1,t}^T + \mathbf{Q}_t \quad (4.22)$$

Tabela 4.2 Equações do filtro de Kalman estendido

Comparando as equações do filtro de Kalman na Tabela 4.1 com aquelas do filtro de Kalman estendido na Tabela 4.2, apenas umas poucas diferenças são notadas. Primeiro, os termos lineares $\mathbf{H}_t \hat{\mathbf{x}}_{t|t-1}$ e $\mathbf{F}_{t+1,t} \hat{\mathbf{x}}_{t|t-1}$ no filtro de Kalman são substituídas por $h_t(\hat{\mathbf{x}}_{t|t-1})$ e $f_t(\hat{\mathbf{x}}_{t|t-1})$ no EKF, respectivamente. A matriz de transição de estado $\mathbf{F}_{t+1,t}$ e a matriz de medida \mathbf{H}_t no filtro de Kalman também são substituídas pelas matrizes Jacobianas $\mathbf{F}_{t+1,t}$ e \mathbf{H}_t no EKF, respectivamente. As matrizes de derivadas devem ser recalculadas para cada iteração do filtro de Kalman.

4.3 Aplicação do EKF a Redes Neurais

O algoritmo do filtro de Kalman estendido foi primeiro aplicado para o treinamento de perceptrons multicamadas alimentados adiante por (SINGHAL e WU, 1989). Eles mostram que o algoritmo convergia em menos iterações que o algoritmo de retropropagação padrão

usando uns poucos exemplos artificiais. Eles também mostraram que, em alguns casos que quando o algoritmo de retropropagação falhava, o EKF convergia para uma boa solução. Seguindo esta demonstração, muitos autores usaram este método para treinar redes neurais. Puskorius e Feldkamp (1991) aplicaram o algoritmo para o treinamento de redes neurais recorrentes para problemas reais na área de engenharia. Williams (1992) também estudou o seu uso em redes recorrentes. Williams (1992) examina a relação entre EKF e os algoritmos de gradiente mais amplamente utilizados o RTRL e o BPTT. É observado que o algoritmo RTRL é uma forma degenerada do método EKF dando um considerável aumento de velocidade em termos dos passos requeridos para convergência quando comparado com o BPTT e o RTRL.

O treinamento dos pesos em redes neurais pode ser abordado como um problema de estimação de estado (PATEL, 2000). As redes neurais operam inerentemente de uma forma não linear e de tal forma que esta estimação de estado é realizada via o filtro de Kalman estendido como contrário ao algoritmo do filtro de Kalman, o qual é aplicada para sistemas lineares. Quando aplica-se o algoritmo EKF para o treinamento de redes neurais, os pesos da rede são organizados como um vetor \mathbf{w}_t e são tratados como o vetor de estado \mathbf{x}_t do sistema, descrito acima. Dessa forma, o vetor de estado poderá ser referido como \mathbf{w}_t em lugar de \mathbf{x}_t . O algoritmo EKF atualiza o estado de um passo de tempo para um outro e também atualiza a matriz \mathbf{P} de correlação estado-erro como descrito anteriormente.

Primeiro, considere uma rede neural arbitrária (*feedforward* ou recorrente) dado um vetor de objetivos desejado \mathbf{d}_t de tamanho $n_0 \times 1$ para nós de saída em cada passo. O objetivo é encontrar os pesos de tal forma que a diferença entre a saída da rede e o valor desejado (isto é, o erro) seja minimizado no sentido do erro quadrático médio. Seja uma rede com um total de t_w pesos (incluindo os pesos de bias) e t_o unidades de saída que têm valores desejados associados a elas. Pode-se lançar o problema de treinamento da rede neural dentro de uma estrutura de EKF, primeiro anotando as equações não lineares que governam a operação da rede. O modelo da rede neural do sistema de espaço de estado é então dado por:

$$\mathbf{d}_t = h_t(\mathbf{w}_t, \mathbf{u}_t) + \mathbf{r}_t, \quad (4.23)$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t. \quad (4.24)$$

Comparando as equações (4.23) e (4.24) com as equações de medida e processo (4.12) e (4.13), nota-se que $\mathbf{F}_{t+1,t} = I$, onde I é a matriz identidade e $\mathbf{Q} = 0$. Também nota-se que o vetor de valores desejados \mathbf{d}_t é tratado como o vetor de medida na terminologia Kalman e que a função vetor não linear h_t descreve a relação entre as entradas, pesos e saídas da rede neural. Note que \mathbf{u}_t é o vetor de entrada para a rede no passo t . Com a formulação de espaço de estado de uma rede neural arbitrária, dada pelas equações (4.23) e (4.24), as equações do EKF da Tabela 3.3 podem ser escritas como:

$$\mathbf{K}_t = \frac{\mathbf{P}_{t|t-1} \mathbf{H}_t^T}{\mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^T + \mathbf{R}_t}, \quad (4.25)$$

$$\hat{\mathbf{w}}_{t|t} = \hat{\mathbf{w}}_{t|t-1} + \mathbf{K}_t \left[\mathbf{d}_t - h_t(\hat{\mathbf{w}}_{t|t-1}), \mathbf{u}_t \right], \quad (4.26)$$

$$\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{K}_t \mathbf{H}_t \mathbf{P}_{t|t-1}, \quad (4.27)$$

$$\hat{\mathbf{w}}_{t+1|t} = \hat{\mathbf{w}}_{t|t}, \quad (4.28)$$

$$\mathbf{P}_{t+1|t} = \mathbf{P}_{t|t}, \quad (4.29)$$

pode ser mais simplificado em uma forma compacta, primeiro simplificando a notação como:

$$\hat{\mathbf{w}}_t = \hat{\mathbf{w}}_{t|t} \quad \text{e} \quad \mathbf{P}(t) = \mathbf{P}(t|t).$$

e então substituindo as Equações (4.28) e (4.29) nas Equações (4.25), (4.26) e (4.27), resultando em:

$$\mathbf{K}_t = \frac{\mathbf{P}_t \mathbf{H}_t^T}{\mathbf{H}_t \mathbf{P}_t \mathbf{H}_t^T + \mathbf{R}_t}, \quad (4.30)$$

$$\hat{\mathbf{w}}_{t+1} = \hat{\mathbf{w}}_t + \mathbf{K}_t \left[\mathbf{d}_t - h_t(\hat{\mathbf{w}}_t), \mathbf{u}_t \right], \quad (4.31)$$

$$\mathbf{P}_{t+1} = \mathbf{P}_t - \mathbf{K}_t \mathbf{H}_t \mathbf{P}_t. \quad (4.34)$$

Para linearizar a Equação (4.23), a matriz Jacobiana $H_t(t)$ deve ser calculada como:

$$\mathbf{H}_t = \frac{\partial h_n(\mathbf{w}_t, \mathbf{u}_t)}{\partial \mathbf{w}} = \begin{bmatrix} \frac{\partial h_1}{\partial w_1} & \frac{\partial h_1}{\partial w_2} & \dots & \frac{\partial h_1}{\partial w_{t_w}} \\ \frac{\partial h_2}{\partial w_1} & \frac{\partial h_2}{\partial w_2} & \dots & \frac{\partial h_2}{\partial w_{t_w}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_{t_0}}{\partial w_1} & \frac{\partial h_{t_0}}{\partial w_2} & \dots & \frac{\partial h_{t_0}}{\partial w_{t_w}} \end{bmatrix}, \quad (4.33)$$

sendo que, $h_t(\bullet) = [h_1, h_2, \dots, h_{t_0}]$ são as t_0 saídas da rede, e $\mathbf{w} = [w_1, w_2, \dots, w_{t_w}]$ são os t_w pesos ordenados da rede. As derivadas são avaliadas na estimativa corrente do estado $\hat{\mathbf{w}} = (t)$. A relação $h_t(\bullet)$ depende da arquitetura da rede que é empregada e desempenha um papel crucial no método pelo qual a matriz Jacobiana \mathbf{H}_t é calculada. A diferença entre a aplicação do EKF para o treinamento de redes feedforward e recorrentes reside no cálculo da Jacobiana (HAYKIN, 2001; WERBOS, 1990; PUSKORIUS e FELDKAMP, 1994 e PATEL, 2000).

4.4 Filtro de Kalman *Unscented*

Julier et al. (1995) e Julier e Uhlmann (1996 e 1997) propuseram a transformação *unscented* para calcular as estatísticas de uma variável x propagada através de uma função não linear $\mathbf{y} = f(\mathbf{x})$.

Considere a propagação de uma variável aleatória x (dimensão L) através de uma função não linear, $\mathbf{y} = f(\mathbf{x})$. Assumindo que x tem média \bar{x} e covariância \mathbf{P}_x . Para calcular as estatísticas de y , nós formamos a matriz \mathfrak{X} de $2L + 1$ vetores sigma \mathfrak{X}_i de acordo com as seguintes equações:

$$\mathfrak{X}_0 = \bar{\mathbf{x}}, \quad (4.34)$$

$$\mathfrak{X}_i = \bar{\mathbf{x}} + \left(\sqrt{(L + \lambda) \mathbf{P}_x} \right)_i, \quad i = 1, \dots, L, \quad (4.35)$$

$$\mathfrak{X}_i = \bar{\mathbf{x}} - \left(\sqrt{(L + \lambda) \mathbf{P}_x} \right)_{i-L}, \quad i = 1, \dots, 2L, \quad (4.36)$$

sendo que $\lambda = \alpha^2 (L + \kappa) - L$ é o parâmetro de escalonamento. A constante α determina a dispersão dos pontos sigma em torno de $\bar{\mathbf{x}}$, usualmente é fixado como um valor positivo pequeno ($1 \leq \alpha \leq 10^{-4}$). A constante κ é o segundo parâmetro de escalonamento, o qual é usualmente fixado como sendo $3 - L$, β é utilizado para incorporar conhecimento a priori da distribuição de \mathbf{x} (para distribuições Gaussianas, $\beta = 2$ é ótimo). $\left(\sqrt{(L + \lambda)\mathbf{P}_x}\right)_i$ é a i -ésima coluna da raiz quadrada da matriz (ou seja, fatoração triangular inferior de Cholesky).

Cada ponto sigma é propagado através da função $f(\cdot)$ para produzir o conjunto de pontos sigma transformados $\mathfrak{S}_i = f(\mathfrak{X}_0)$ e a média $\bar{\mathbf{y}}$ de uma distribuição transformada é estimada por:

$$\begin{aligned} \bar{\mathbf{y}} = \sum_{i=0}^{2L} W_i \varphi_i = \frac{\lambda}{L + \lambda} f(\bar{\mathbf{x}}) + f\left(\bar{\mathbf{x}} - \sqrt{L + \lambda} \cdot s_{x,i}\right) + \\ + \frac{1}{2(L + \lambda)} \sum_{i=1}^L \left(f\left(\bar{\mathbf{x}} + \sqrt{L + \lambda} \cdot s_{x,i}\right) \right). \end{aligned} \quad (4.37)$$

O estimador da covariância obtido pela transformada *unscented* é:

$$\begin{aligned} \mathbf{P}_y = \sum_{i=0}^{2L} W_i (\varphi_i - \bar{\mathbf{y}})(\varphi_i - \bar{\mathbf{y}})^T = \frac{\lambda}{L + \lambda} (f(\hat{\mathbf{x}}) - \bar{\mathbf{y}})(f(\hat{\mathbf{x}}) - \bar{\mathbf{y}})^T + \\ + \frac{1}{2(L + \lambda)} \sum_{i=1}^L \left(f\left(\bar{\mathbf{x}} + \sqrt{L + \lambda} \cdot s_{x,i}\right) f\left(\bar{\mathbf{x}} + \sqrt{L + \lambda} \cdot s_{x,i}\right)^T \right) + \\ + \frac{1}{2(L + \lambda)} \sum_{i=1}^L \left(\left(f\left(\bar{\mathbf{x}} - \sqrt{L + \lambda} \cdot s_{x,i}\right) - \bar{\mathbf{y}} \right) \left(f\left(\bar{\mathbf{x}} - \sqrt{L + \lambda} \cdot s_{x,i}\right) - \bar{\mathbf{y}} \right)^T \right). \end{aligned} \quad (4.38)$$

O Filtro de Kalman Estendido (EKF) foi aplicado com sucesso na estimação de parâmetros de redes neurais (FREITAS, 1999; SINGHAL e WU, 1989 e WILLIAMS, 1992). Também foi mostrado que as estatísticas estimadas pelo EKF podem ser utilizadas para estimar sequencialmente a estrutura (número de neurônios ocultos e conexões) bem como os parâmetros das redes *feedforward* (TODOROVIC' et al., 2000), Funções de Base Radial (RBF) e recorrentes (TODOROVIC' et al., 2002).

O estimador filtro de Kalman Unscented tem sido apresentado (JULIER e UHLMANN, 1996 e 1997) com resultados que superam a performance do EKF na estimação de estados

não lineares. Na estimação de parâmetros de redes neurais *feedforward* o UKF é comparável ou levemente melhor que o EKF (MERWE e WAN, 2001a e 2001b), com a vantagem significativa de que ele não demanda o cálculo da Jacobiana da rede neural.

Considerando o sistema dinâmico de tempo discreto não linear no qual o sinal (ou série) não observado x_t é modelado como um processo de Markov de alcance 1 (FERRARI e GALVES, 1997) de distribuição inicial $p(x_0)$ e equação de transição:

$$x_t = f(x_{t-1}, u_{t-1}) + q_t, \quad (4.39)$$

sendo que u_t denota a entrada exógena conhecida.

As observações são assumidas condicionalmente independentes dado o estado x_t :

$$y_t = h(x_t) + r_t, \quad (4.40)$$

O processo ruído q_t guia o sistema dinâmico, enquanto o ruído da observação é dado por r_t .

O estimador do Erro Médio Quadrático Mínimo (MMSE – *Minimum Mean Squared Error*) do estado x_t do sistema de tempo discreto não linear (1) satisfaz as condições de que o erro de estimação $\tilde{x}_t = x_t - \hat{x}_t$ é não viesado $E[\tilde{x}_t] = 0$ e também ortogonal a observação y_t , ou seja, $E[\tilde{x}_t y_t^T] = 0$. O EKF e o UKF fornecem um estimador do MMSE do estado x_t utilizando-se do esquema preditor-corretor tal como mostrado na Figura 4.1.

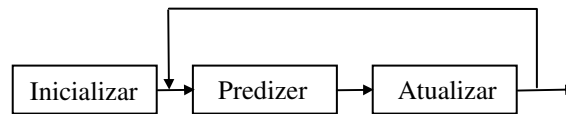


Figura 4.1 Estrutura recursiva preditor-corretor do filtro de Kalman

Dado o estimador do estado \hat{x}_{t-1} e sua covariância $P_{x(t-1)}$, obtidos do conjunto de informações até o passo de tempo $Y_{t-1} = \{y_1, y_2, \dots, y_{t-1}\}$, o filtro prediz o estado futuro utilizando o modelo do processo e o conhecimento sobre a distribuição do ruído do processo. A média e a covariância preditas são idealmente:

$$\hat{x}_t^- = E[x_t | Y_{t-1}] \quad (4.41)$$

$$P_{x(t)}^- = E \left[(x_t - \hat{x}_t^-)(x_t - \hat{x}_t^-)^T | Y_{t-1} \right]. \quad (4.42)$$

O estimador \hat{x}_t e sua covariância $P_{x(t)}$ são obtidos pela atualização (correção) da predição do estado $(\hat{x}_t^-, P_{x(t)}^-)$ com a observação atual y_t :

$$\hat{x}_t = \hat{x}_t^- + K_t (y_t - \hat{y}_t^-), \quad (4.43)$$

$$K_t = P_{xy(t)} P_{y(t)}^{-1}, \quad (4.44)$$

$$P_{x(t)} = P_{x(t)}^{-1} - K_t P_{y(t)}^{-1} K_t^T. \quad (4.45)$$

Sendo que $\hat{y}_t^- = E[y_t | Y_{t-1}]$ e $P_{y(t)} = E[(y_t - \hat{y}_t^-)(y_t - \hat{y}_t^-)^T | Y_{t-1}]$ são a predição da observação e sua covariância e $P_{xy(t)} = E[(x_t - \hat{x}_t^-)(y_t - \hat{y}_t^-)^T | Y_{t-1}]$ é a correlação condicionada. Estas equações dependem dos valores preditos dos dois primeiros momentos de x_t e y_t , dado o conjunto de observações Y_{t-1} .

Neste trabalho, será considerado o treinamento de uma rede neural auto-regressiva não linear com entradas endógenas utilizando o EKF e o UKF. Os resultados das predições obtidas a partir do treinamento da rede com o EKF e o UKF serão comparados com os resultados dos modelos GARCH (*Generalized Autoregressive Conditional Heteroskedasticity*). A Tabela 4.3 mostra o algoritmo do UKF.

Inicializar com

$$\hat{\mathbf{x}}_0 = E[\mathbf{x}_0]$$

$$\mathbf{P}_0 = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T]$$

$$\hat{\mathbf{x}}_0^a = E[\mathbf{x}^a] = [\hat{\mathbf{x}}_0^T \quad 0 \quad 0]^T$$

$$\mathbf{P}_0^a = E[(\mathbf{x}_0^a - \hat{\mathbf{x}}_0^a)(\mathbf{x}_0^a - \hat{\mathbf{x}}_0^a)^T] = \begin{bmatrix} \mathbf{P}_0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^v & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}^n \end{bmatrix}$$

para $t \in [1, \dots, \infty]$, calcular os pontos sigma:

$$\mathbf{X}_{t-1}^a = \begin{bmatrix} \hat{\mathbf{x}}_{t-1}^a & \hat{\mathbf{x}}_{t-1}^a + \gamma \sqrt{\mathbf{P}_{t-1}^a} & \hat{\mathbf{x}}_{t-1}^a - \gamma \sqrt{\mathbf{P}_{t-1}^a} \end{bmatrix}$$

As equações de atualização são

$$\mathbf{X}_{t|t-1}^x = \mathbf{F} \left(\mathbf{X}_{t|t-1}^x, \mathbf{u}_{t-1}, \mathbf{X}_{t-1}^x \right)$$

$$\hat{\mathbf{x}}_{t-1}^- = \sum_{i=0}^{2L} W_i^{(m)} \mathbf{X}_{i,t|t-1}^x$$

$$\mathbf{P}_{t-1}^- = \sum_{i=0}^{2L} W_i^{(c)} \left(\mathbf{X}_{i,t|t-1}^x - \hat{\mathbf{x}}_{t-1}^- \right) \left(\mathbf{X}_{i,t|t-1}^x - \hat{\mathbf{x}}_{t-1}^- \right)^T$$

$$\mathbf{y}_{t|t-1} = \mathbf{H} \left(\mathbf{X}_{i,t|t-1}^x, \mathbf{X}_{t-1}^n \right)$$

$$\hat{\mathbf{y}}_{t-1}^- = \sum_{i=0}^{2L} W_i^{(m)} y_{i,t|t-1}$$

as equações de inovação de medida são

$$\mathbf{P}_{\bar{\mathbf{y}}_t, \bar{\mathbf{y}}_t} = \sum_{i=0}^{2L} W_i^{(c)} \left(y_{i,t|t-1} - \hat{\mathbf{y}}_t^- \right) \left(y_{i,t|t-1} - \hat{\mathbf{y}}_t^- \right)^T$$

$$\mathbf{P}_{\mathbf{x}_t, \mathbf{y}_t} = \sum_{i=0}^{2L} W_i^{(c)} \left(x_{i,t|t-1} - \hat{\mathbf{x}}_t^- \right) \left(x_{i,t|t-1} - \hat{\mathbf{x}}_t^- \right)^T$$

$$\mathbf{K}_t = \mathbf{P}_{\mathbf{x}_t, \mathbf{y}_t} \mathbf{P}_{\bar{\mathbf{y}}_t, \bar{\mathbf{y}}_t}^{-1}$$

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t-1}^- + \mathbf{K}_t \left(\mathbf{y}_t - \hat{\mathbf{y}}_{t-1}^- \right)$$

$$\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{K}_t \mathbf{P}_{\bar{\mathbf{y}}_t, \bar{\mathbf{y}}_t}^{-1} \mathbf{K}_t^T$$

sendo que

$$\mathbf{x}^a = \begin{bmatrix} \mathbf{x}^T & \mathbf{v}^T & \mathbf{n}^T \end{bmatrix}^T, \quad \mathbf{X}^a = \begin{bmatrix} (\mathbf{X}^x)^T & (\mathbf{X}^v)^T & (\mathbf{X}^n)^T \end{bmatrix}^T, \quad \gamma = \sqrt{L + \lambda}$$

λ é o parâmetro de escala, L é a dimensão do estado, \mathbf{R}^v é a covariância do processo, \mathbf{R}^n é a covariância da medida e W_i são os pesos calculados como

$$W_0^{(m)} = \frac{\lambda}{L + \lambda}$$

$$W_0^{(c)} = \frac{\lambda}{L + \lambda} + 1 - \alpha^2 + \beta$$

$$W_i^{(m)} = W_i^{(c)} = \frac{\lambda}{2(L + \lambda)}, \quad i = 1, \dots, 2L$$

Tabela 4.3 Equações do filtro de Kalman *unscented*

4.4.1 Representações de modelos de espaço de estado

O modelo de espaço de estado geral para um sistema univariado pode ser escrito como uma variante da equação (4.46), que representa a equação de medida, e (4.47), que representa a equação de transição:

$$y_t = H_t x_t + r_t, \quad (4.46)$$

$$x_t = F_t x_{t-1} + q_t, \quad (4.47)$$

sendo que y_t é um vetor $m \times 1$, H_t é uma matriz $m \times n$, x_t é um vetor $n \times 1$ de coeficientes, e

$$\begin{pmatrix} r_t \\ q_t \end{pmatrix} \sim N \left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}; \begin{pmatrix} R_t & 0 \\ 0 & Q_t \end{pmatrix} \right],$$

em aplicações usuais destes modelos, considera-se que F_t , R_t e Q_t são conhecidas.

Para colocar-se um processo AR(1) em um modelo de espaço de estados tem-se que:

$$y_t = \phi y_{t-1} + \varepsilon_t, \quad (4.48)$$

sendo que

$$\varepsilon_t \sim iid N(0; \sigma^2), \quad (4.49)$$

nesse caso já tem-se o processo AR(1) na forma de espaço de estado. Dessa forma as equações de medida e de transição para o processo AR(1), respectivamente, são:

$$y_t = x_t, \quad (4.50)$$

$$x_t = \phi x_{t-1} + \varepsilon_t. \quad (4.51)$$

Para um processo univariado MA(1) pode-se construir o modelo de espaço de estado da seguinte maneira, considere:

$$y_t = \varepsilon_t + \theta \varepsilon_{t-1}, \quad (4.52)$$

sendo que

$$\varepsilon_t \sim iid N(0; \sigma^2). \quad (4.53)$$

Para que este processo fique na forma de espaço de estado é necessário termos que:

$$y_t = [1 \quad \theta] \begin{bmatrix} \varepsilon_t \\ \varepsilon_{t-1} \end{bmatrix} + r_t, \quad (4.54)$$

sendo que para x_t tem-se que:

$$x_t = \begin{bmatrix} \varepsilon_t \\ \varepsilon_{t-1} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \varepsilon_{t-1} \\ \varepsilon_{t-2} \end{bmatrix} + \begin{bmatrix} \varepsilon_t \\ 0 \end{bmatrix}. \quad (4.55)$$

Portanto, no caso de um processo MA(1) todos os componentes do modelo de espaço de estado são dados por:

$$r_t = 0, \quad (4.56)$$

$$H_t = [1 \quad \theta], \quad (4.57)$$

$$F_t = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad (4.58)$$

$$q_t = \begin{bmatrix} \varepsilon_t \\ 0 \end{bmatrix}. \quad (4.59)$$

Para um modelo AR(2) dado por:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \varepsilon_t, \quad (4.60)$$

sendo que

$$\varepsilon_t \sim iid \ N(0; \sigma^2). \quad (4.61)$$

Teremos:

$$y_t = [1 \quad 0] \begin{bmatrix} y_t \\ y_{t-1} \end{bmatrix} + r_t, \quad (4.62)$$

sendo que para x_t tem-se:

$$x_t = \begin{bmatrix} y_t \\ y_{t-1} \end{bmatrix} = \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y_{t-1} \\ y_{t-2} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \varepsilon_t. \quad (4.63)$$

No caso de um modelo ARMA(2,1) dado por:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \theta \varepsilon_{t-1} + \varepsilon_t, \quad (4.64)$$

sendo que

$$\varepsilon_t \sim iid \ N(0; \sigma^2). \quad (4.65)$$

Tem-se:

$$y_t = [1 \quad 0] \begin{bmatrix} y_t \\ \phi_2 y_{t-1} + \theta \varepsilon_t \end{bmatrix} + r_t, \quad (4.66)$$

sendo que para x_t tem-se:

$$x_t = \begin{bmatrix} y_t \\ \phi_2 y_{t-1} + \theta \varepsilon_t \end{bmatrix} = \begin{bmatrix} \phi_1 & 1 \\ \phi_2 & 0 \end{bmatrix} \begin{bmatrix} y_{t-1} \\ \phi_2 y_{t-2} + \theta \varepsilon_{t-1} \end{bmatrix} + \begin{bmatrix} 1 \\ \theta \end{bmatrix} \varepsilon_t. \quad (4.67)$$

Genericamente pode-se escrever um processo univariado ARMA(p,q) na forma de espaço de estado da seguinte maneira:

A forma de espaço de estado geral pode ser escrita como:

$$y_t = H_t x_t + G_t r_t, \quad (4.68)$$

$$x_t = F_t x_{t-1} + T_t q_t, \quad (4.69)$$

sendo que y_t é um vetor $m \times 1$, H_t é uma matriz $m \times n$, x_t é um vetor $n \times 1$ de coeficientes, e

$$\begin{pmatrix} r_t \\ q_t \end{pmatrix} \sim N \left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}; \begin{pmatrix} R_t & 0 \\ 0 & Q_t \end{pmatrix} \right],$$

em aplicações usuais destes modelos, considera-se que F_t , R_t , G_t , T_t e Q_t são conhecidas.

Para um modelo univariado com um vetor de estado x_t de dimensão $n \times 1$ e vetores de erros r_t e q_t de dimensão $m \times 1$, as matrizes H_t , F_t , G_t e T_t têm as dimensões $1 \times n$, $n \times n$, $1 \times m$ e $n \times m$, respectivamente.

A representação de espaço de estado para processos ARMA, considerando $H_t = H = (1, 0, \dots, 0)$, $G_t = G = 1$, pode ser escrita como (PEARLMAN, 1980):

$$F_t = F = \begin{bmatrix} \phi_1 & 1 & 0 & \dots & 0 \\ \phi_2 & 0 & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & 1 \\ \vdots & 0 & \dots & 0 & 0 \\ \phi_m & 0 & \dots & \dots & \end{bmatrix},$$

sendo que $m = \max\{p, q\}$. Em Hamilton (1994, p. 375) o processo ARMA(p,q) pode ser escrito em forma de espaço de estado, mas a representação $m = \max(p, q)$ é substituída por

$m = \max\{p, q + 1\}$. Isso acontece porque nesta versão H e F são iguais ao que foi definido anteriormente, mas para G é atribuído o valor zero e $T = [1 \ \theta_1 \ \theta_2 \ \dots \ \theta_{m-1}]$.

Neste trabalho os modelos ARCH e GARCH serão representados na forma de espaço de estado. Os modelos ARCH terão a seguinte representação na forma de espaço de estado:

$$y_t = x_t, \quad (4.70)$$

$$x_t = \left(\alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \dots + \alpha_r \varepsilon_{t-r}^2 \right)^{1/2} \eta_t, \quad (4.71)$$

sendo que η_t é i.i.d. $(0,1)$, $\alpha_0 > 0$, $\alpha_i \geq 0$, $i > 0$.

No caso dos modelos GARCH é utilizadas a seguinte representação na forma de espaço de estado:

$$y_t = x_t, \quad (4.72)$$

$$x_t = \left(\alpha_0 + \sum_{i=1}^r \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^s \beta_j h_{t-j} \right)^{1/2} \eta_t, \quad (4.73)$$

sendo que η_t é i.i.d. $(0,1)$, $\alpha_0 > 0$, $\alpha_i \geq 0$, $\beta_j \geq 0$, $\sum_{i=1}^q (\alpha_i + \beta_j)$, $q = \max(r, s)$.

4.5 Teste para Não linearidade

Na análise de séries heterocedásticas, antes de começar uma procura por uma especificação geral para encontrar um modelo que ajuste seu conjunto particular de dados, é importante testar a presença de não linearidade. Os testes para verificar não linearidade que têm sido mais utilizados são o BDS - Brock, Dechert, e Scheinkman (1987), o teste de McLeod e Li (1983), um teste desenvolvido por Hsieh (1989) e um teste sugerido por Teräsvirta e Granger (1993). Neste trabalho é considerado o teste BDS.

Na estimação de um modelo ARMA, a função de autocorrelação (ACF) pode auxiliar a selecionar os valores de p e q , e o ACF dos resíduos é uma importante ferramenta de diagnóstico. Infelizmente, o ACF tal como é usado em modelos lineares pode levar a falsas conclusões nos modelos não lineares. A razão é que os coeficientes de autocorrelação medem o grau de associação linear entre y_t e y_{t-i} . Dessa forma, o ACF pode falhar na

detecção de importantes relações não lineares presentes nos dados. Tendo interesse nas relações não lineares dos dados, uma ferramenta de diagnóstico útil é examinar o ACF dos quadrados e dos cubos dos valores de uma série.

O teste de McLeod-Li (1983) visa determinar se existem autocorrelações significantes nos resíduos quadrados de uma equação linear. Para realizar o teste, deve-se estimar a série utilizando o melhor modelo de ajuste linear e chamar os resíduos de $\hat{\epsilon}_t$. Tal como em um teste formal para erros ARCH, constrói-se as autocorrelações dos resíduos quadrados. Fazendo ρ_i denotar o coeficiente de correlação da amostra entre os resíduos $\hat{\epsilon}_t^2$ e $\hat{\epsilon}_{t-i}^2$ utiliza-se a estatística de Ljung-Box para determinar se os resíduos quadrados exibem correlação serial. Conseqüentemente, tem-se que:

$$Q = T(T+2) \sum_{i=1}^n \frac{\rho_i}{(T-i)}. \quad (4.74)$$

O valor de Q tem uma distribuição χ^2 assintótica com n graus de liberdade se a seqüência $\{\hat{\epsilon}_t^2\}$ é não-correlacionada. Rejeitar a hipótese nula é equivalente a aceitar que o modelo é não linear. Alternativamente, pode-se estimar a regressão $\hat{\epsilon}_t^2 = \alpha_0 + \alpha_1 \hat{\epsilon}_{t-1}^2 + \dots + \alpha_n \hat{\epsilon}_{t-n}^2 + v_t$. Se não existir não linearidade, α_1 até α_n deverão ser estatisticamente iguais a zero. Com uma amostra de T resíduos, se não houver não linearidades, a estatística teste TR^2 converge para uma distribuição χ^2 com n graus de liberdade. Este teste tem um poder substancial para detectar várias formas de não linearidade. Porém, a forma real da não linearidade não é especificada pelo teste. Rejeitar a hipótese nula de linearidade não informa a natureza da não linearidade presente nos dados.

Capítulo 5

Metodologia

5.1 Metodologia de Pesquisa

Nos primeiros capítulos, dedicados à revisão teórica, foram apresentados e discutidos os modelos e técnicas utilizados para análise e previsão de séries temporais. Cada uma das técnicas apresentadas: procedimento de Box e Jenkins, Redes Neurais (*feedforward* e recorrentes), RN treinadas com EKF e UKF apresentam formas diferentes de implementação e de análise de resultados. Visando concretizar a importância de se explorar a utilização de redes neurais treinadas com filtro de Kalman para análise de séries temporais, é especificada a metodologia usada em cada uma das tecnologias apresentadas.

Este trabalho está baseado em teorias quantitativas de análise, sendo assim os modelos matemáticos e estatísticos são utilizados a fim de atender o objetivo da pesquisa. Os estudos quantitativos são aqueles em que os pesquisadores estabelecem hipóteses e as testa por meio da mensuração de variáveis operacionais definidas, quantificando o resultado com o uso de variados métodos matemáticos e estatísticos. É um tipo de pesquisa tradicionalmente aceito como tendo confiabilidade, fidedignidade e validade, desde que realizada seguindo uma metodologia rigorosa. A validação dá-se pelo nível de significância adotado nos testes estatísticos (MARCONI e LAKATOS, 2007; LIMA, 2004). O nível de significância será chamado aqui de p-valor.

Os principais tipos de pesquisa são busca de fatos, consolidação, extensão, reformulação e criação de teoria (KNELLER, 1978). A *consolidação* consiste em desenvolver as implicações de uma lei ou teoria para as áreas em que se espera a sua aplicabilidade. A *extensão* é a aplicação de uma lei ou teoria a novas teorias. *Reformulação* é a revisão de uma teoria para torná-la mais clara, mais simples ou mais facilmente aplicável. A *construção de teoria*, incluindo a criação de novas leis e taxonomias, é a mais vital e original forma de pesquisa científica.

Neste trabalho serão analisadas séries temporais univariadas de dados reais de preços de fechamento de ações de empresas e commodities dos seguintes setores econômicos:

- Setor Financeiro: foram analisadas as séries do Bradesco PN, Bradespar PN, Itausa PN e Itaú PN;
- Setor de Alimentos: Perdigão PN, Sadia PN, Saca da Soja de 60Kg e Saca de Açúcar de 50Kg;
- Setor Industrial: Marcopolo PN, Petrobrás PN, Embraer ON, Ripasa PN, Souza Cruz ON e a Gerdau PN;
- Setor de Serviços: Pão de Açúcar PN, Eletropaulo PNA, Eletrobras PNB, Brasil Telecom PN, Cesp PNA e a Lojas Americanas PN;

As séries foram obtidas da Economática.

A abordagem de previsão defendida por Box e Jenkins pode ser delineada em quatro passos (HAMILTON, 1994): o primeiro passo é a transformação dos dados, se necessário, de tal forma que a suposição de covariância-estacionária seja a única razoável. O segundo passo é fazer uma suposição inicial de pequenos valores para p e q para um modelo ARMA(p , q) que poderia descrever a série transformada. O terceiro, estimar os parâmetros. O quarto passo é realizar análise de diagnóstico para confirmar que o modelo é de fato consistente com as características observadas dos dados.

Neste trabalho a construção dos modelos ARIMA-GARCH obedecerá a seqüência apresentada a seguir.

Plotar a série no nível e plotar a série transformada. Realizar o teste ADF para verificar a presença de raiz unitária na série no nível, depois o mesmo teste ADF será realizado para a série transformada. O teste de Dickey-Fuller aumentado (ADF) é dado pela expressão:

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \sum_{i=2}^p \delta_i y_{t-i+1} + \varepsilon_t. \quad (5.1)$$

Depois disso é identificado o melhor modelo ARIMA de acordo com o correlograma. A seguir são estimados os parâmetros do modelo ARIMA e os resíduos são testados quanto a normalidade. É realizado o teste BDS, desenvolvido por Brock et alii (1996), para verificar a presença de dependência não linear, dado por:

$$BDS_{\varepsilon,m} = \frac{\sqrt{N} \left[C_{\varepsilon,m} - (C_{\varepsilon,1})^m \right]}{\sqrt{V_{\varepsilon,m}}}. \quad (5.2)$$

O teste BDS é um teste *portmanteau* relacionado a dependência em séries de tempo. Conforme Fernandes e Preumont (2002), o BDS apresenta um alto poder contra uma variedade de modelos lineares, não lineares e não-estacionários. O teste BDS é executado da segunda até a sexta dimensão, portanto, uma determinada série temporal rejeitará a hipótese nula se, e somente se, pelo menos um dos cinco níveis descritivos do teste (p-valor) forem menores do que o nível de significância de 5%, o que justifica o ajuste de um modelo GARCH.

Em seguida é feito o teste ARCH-LM para os resíduos do modelo ajustado para a série NDL para verificar se há efeito ARCH nos resíduos. É ajustado o modelo ARIMA-GARCH, realizado o teste de Ljung-Box para verificar se os resíduos são RB. Finalmente são realizadas as previsões e obtidas as estatísticas de ajuste.

O treinamento e uso de redes neurais para previsão requer o seguinte conjunto de passos (GATELY, 1996): inicialmente deve-se determinar o que será previsto, depois disso coletar o conjunto de dados ou série de números que tenha relação com o que será previsto. A seguir é necessário pré-processar os dados para combinar informação ou torná-los mais útil. Verificar máximos e mínimos, determinar o intervalo dos dados e determinar valores máximos e mínimos. Extrair um conjunto de teste dos dados. Selecionar uma arquitetura de rede adequada. Selecionar um algoritmo de aprendizagem (treinamento) adequado. Treinar a rede neural. Usar a rede neural, aplicar novos dados para a rede treinada e criar uma previsão.

A aprendizagem de uma rede neural baseada na descida do gradiente, normalmente é lenta por ter de contar com as estimativas instantâneas do gradiente. Esta limitação pode ser superada se o processo de treinamento for considerado como um problema de filtragem ótima, cuja solução utiliza recursivamente a informação contida nos dados de treinamento voltando à primeira iteração do processo de aprendizagem (HAYKIN, 2001). Para isto será utilizado o filtro de Kalman. As características do filtro de Kalman incluem: formulação do espaço de estados. A estimativa do estado é calculada recursivamente; isto é, cada estimativa atualizada do estado é calculada a partir da estimativa anterior e dos dados

correntemente disponíveis, e com isso apenas a estimativa prévia necessita de armazenamento.

Pode-se utilizar uma RNA treinada com filtro de Kalman de duas formas distintas. A primeira é estimar o melhor modelo ARIMA-GARCH e usar este modelo estimado para gerar os valores usados nos cálculos de treinamento e previsão. A outra é tomar como entrada para a RNA os valores originais da série.

A Figura 5.1 mostra o fluxograma que ilustra o processo de utilização do filtro de Kalman no treinamento de uma RNA *multilayer* perceptron (MLP) quando o modelo ARIMA-GARCH é transformado em espaço de estado e usado no algoritmo.

Neste trabalho os dados originais da série temporal serão utilizados no algoritmo da RNA treinada com filtro de Kalman. Os parâmetros alfa, beta e kappa serão tal como em Haykin (2001, p. 229), ou seja, o algoritmo utiliza $\alpha = 1$, $\beta = 2$ e $\kappa = 0$. Os programas utilizados na construção de RNA treinadas com filtro de Kalman são baseados em códigos de Matlab do software Rebel (*Recursive Bayesian Estimation Library* - <http://choosh.bme.ogi.edu/rebel/download/index.html>) cedidos por Eric Wan (ericwan@csee.ogi.edu).

No caso da construção de intervalos de confiança e de previsão para as redes neurais tem-se basicamente cinco passos a serem seguidos. O primeiro passo é a construção do modelo ARIMA-GARCH e da rede neural que faz a previsão desse modelo. Na construção da teoria de construção desses intervalos serão utilizadas inicialmente séries ARIMA-GARCH criadas por simulação no programa Mathematica (ver Apêndice-B). As redes neurais serão do tipo *feedforward* e os dados utilizados divididos em duas parte, uma primeira para estimação e outra para validação.

As funções básicas utilizadas serão: TimeSeries, ARModel, GARCHModel, NeuralARFit, FeedForwardNet e NetSimulate. O programa fornece um *help browser* que pode ser utilizado para buscar informações detalhadas de cada uma dessas funções.

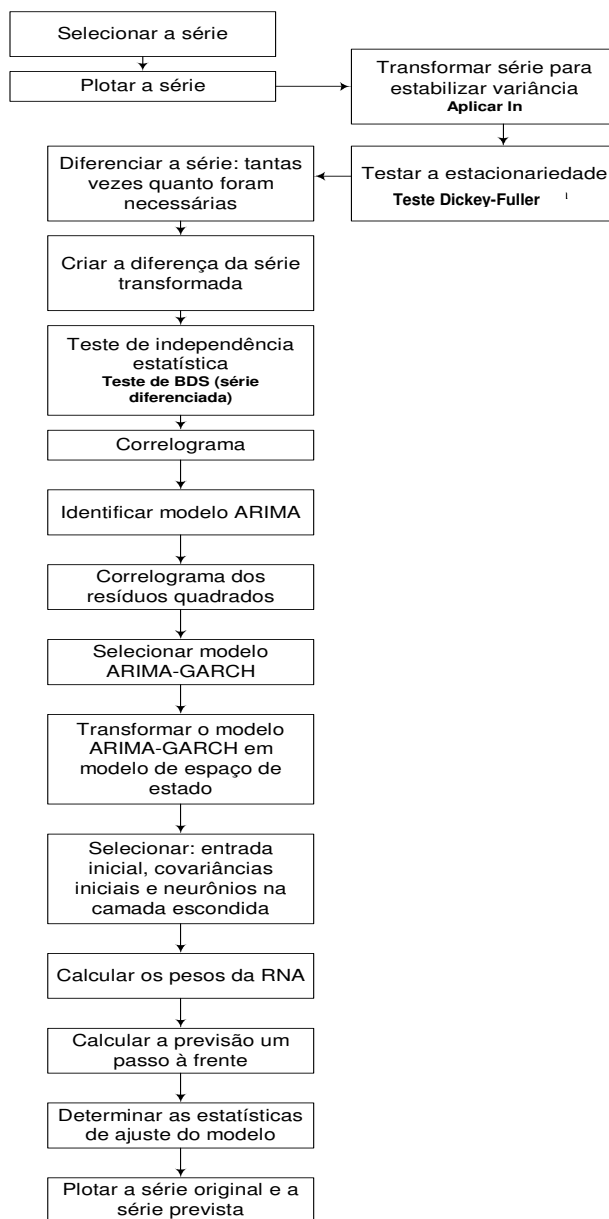


Figura 5.1 Fluxograma para treinar a RNA utilizando o filtro de Kalman.

No segundo passo, os resultados obtidos pela rede neural são utilizados para construir os intervalos de predição. De acordo com a teoria apresentada na revisão teórica.

O terceiro é a construção dos intervalos de confiança. De acordo com a teoria apresentada na revisão teórica. Para realizar os passos dois e três, várias funções de programação são necessárias. O programa não foi desenvolvido de forma ótima, do ponto de vista de explorar e reduzir ao máximo as linhas de comando. As principais funções utilizadas são: AppendTo, Sort, Partition, Transpose, MultipleListPlot e variantes da função Plot.

O quarto passo é plotar os intervalos de confiança e predição. O último passo é o cálculo do RMSE, coeficiente de desigualdade de Theil e Histograma dos erros de predição.

5.2 Cálculo e Obtenção dos Resultados

Para construção dos modelos e teste do desempenho de cada método, serão realizados vários experimentos com diferentes séries temporais univariadas. As simulações e cálculos foram conduzidos utilizando os programas Matlab v. 6.5, Eviews v. 5.0, SPSS v.10 e Mathematica v. 5.0 em um computador Pentium IV com 1,6-GHz CPU e 512-MB de memória RAM.

Capítulo 6

Resultados Experimentais

Todas as séries selecionadas possuem 1542 valores que iniciam em 3/01/2000. Estas séries foram divididas em duas partes, uma para estimação dos modelos ARIMA-GARCH e treinamento da RNA e a segunda parte para realizar as previsões e construir as medidas de performance. A primeira parte das séries contém 1300 valores e a segunda 242. Para cada uma das séries as topologias de RNA que foram testadas estavam concentradas dentro das seguintes possibilidades: 10-1-1; 10-2-1; 10-3-1; 10-4-1; 10-5-1; 10-6-1; 10-7-1 e 10-8-1. Desta forma, todas as redes possuem 10 neurônios na camada de entrada, a camada intermediária poderia ter no máximo nove neurônios e foi utilizado um neurônio na camada de saída.

Na análise de resultados das séries a ordem de apresentação terá a construção e resultados dos modelos ARIMA-GARCH, depois são apresentados os resultados da RNA *feedforward*, a seguir os resultados obtidos pela RNA usando o EKF e por último os resultados da RNA treinada com o UKF.

6.1 Normalização dos Valores de Entrada da RNA

Os dados de entrada da RNA na maioria dos casos necessitará de normalização. Este processo é aconselhável uma vez que coloca o intervalo de dados dentro de limites que facilitam a tarefa da rede para realizar a minimização do erro de saída. De acordo com Azoff (1994, pág. 25) existem quatro abordagens para normalização. Neste caso será utilizado o processo que considera um vetor de entrada de dimensão n , ou seja, os elementos no vetor variam de $i = 1, \dots, n$. A amostra total da série será normalizada ao longo da extensão $[0,1]$. Primeiro será construído o logaritmo da série, LSERIE, depois disso é construída a série da primeira diferença, DLSERIE e por fim a normalização.

A normalização tem a seguinte forma:

$$NDLSERIE_t = \frac{\Delta(\log(SERIE_t)) - \min(\Delta(\log(SERIE_t)))}{\max(\Delta(\log(SERIE_t))) - \min(\Delta(\log(SERIE_t)))},$$

$$NDLSERIE_t = \frac{DLSERIE_t - \min(DLSERIE_t)}{\max(DLSERIE_t) - \min(DLSERIE_t)}. \quad (6.1)$$

Tanto as previsões realizadas pelos modelos ARIMA-GARCH, quanto pelas RNA serão realizadas para séries normalizadas de acordo com (6.1).

6.2 Setor Financeiro – Bradesco, Itaú, Bradespar e Itausa

Para o setor financeiro as séries analisadas seguirão a seqüência: Bradesco, Itaú, Bradespar e Itausa.

6.2.1 Análise da série do Bradesco

Para o caso do Bradesco, inicialmente são apresentados os resultados obtidos pelo método ARIMA-GARCH, pela RNA feedforward, pela RNA-EKF e por final pela RNA-UKF.

6.2.1.1 Análise da série do Bradesco – Método ARIMA-GARCH

Na Figura 6.1a é apresentada a série original do preço de fechamento do Bradesco, na Figura 6.1b é apresentada a série normalizada da primeira diferença do logaritmo do preço, chamada aqui de NDLBRAD.

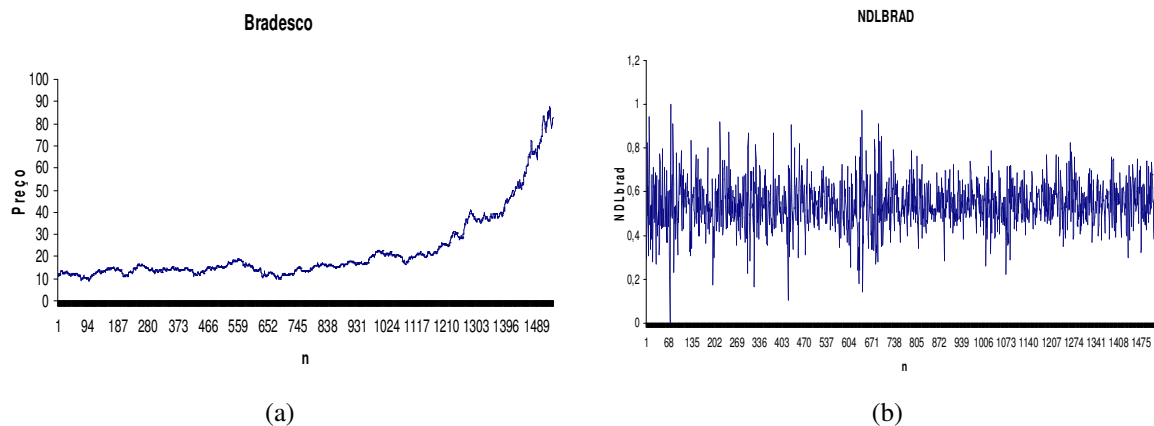


Figura 6.1 (a) Série no nível do preço do Bradesco, (b) Série transformada NDLBRAD

Para testar se a série possui raiz unitária foi aplicado o teste Dickey-Fuller aumentado (ADF). O resultado do teste para a série original é apresentado na Tabela 6.1 e para a série transformada na Tabela 6.2. Nota-se que não é rejeitada a hipótese de raiz unitária para a série no nível e é rejeitada a hipótese de raiz unitária para a série transformada.

Estatística do teste ADF para série original	valores críticos 1%	-3,43439
4,04312	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771

Tabela 6.1 Teste ADF para série original do Bradesco

Estatística do teste ADF para série transformada	valores críticos 1%	-3,43439
-28,5275	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771

Tabela 6.2 Teste ADF para série NDLBRAD

O correlograma da série NDLBRAD é apresentado na Figura 6.2a, a partir do correlograma foi estimado um modelo AR(2). As estatísticas e o histograma dos resíduos são mostrados na Figura 6.2b. A estatística de Jarque-Bera testa a hipótese nula dos resíduos padronizados serem normalmente distribuídos. Verifica-se que os resíduos padronizados são leptocúrticos e a estatística de Jarque-Bera rejeita fortemente a hipótese de distribuição normal.

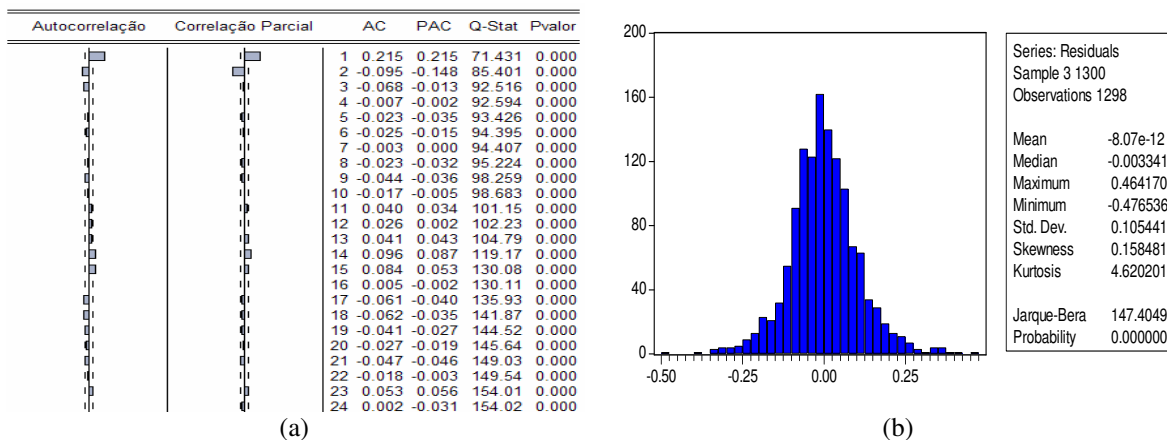


Figura 6.2 (a) Correlograma da série NDLBRAD, (b) Estatísticas dos resíduos

Os resultados do teste para não linearidade desenvolvido por Brock, Dechert e Scheinkman (BDS) são apresentados na Tabela 6.3. O teste BDS foi executado da segunda até a sexta dimensão para a série NDLBRAD e verifica-se que é rejeitada a hipótese nula ao nível de significância de 5%, o que justifica o ajuste de um modelo GARCH.

Dimensão	Estatística BDS	Erro-padrão	Estatística -z	p-valor
2	0,016847	0,001937	8,696789	0,0000
3	0,026464	0,003065	8,632803	0,0000
4	0,032478	0,003634	8,936365	0,0000
5	0,035915	0,003771	9,523393	0,0000
6	0,037347	0,003621	10,31510	0,0000

Tabela 6.3 Resultados obtidos pelo teste BDS nas respectivas dimensões.

Para verificar a existência de heterocedasticidade condicional foi aplicado o teste ARCH-LM. O teste apresentou estatística-F igual a 45,67105 e p-valor 0,000000. Isso implica em presença de heterocedasticidade condicional.

O melhor modelo ajustado para a série NDLBRAD é um AR(2)-GARCH(1,1). Os parâmetros estimados do modelo AR(2)-GARCH(1,1) são mostrados na Tabela 6.4.

Parâmetros	Coefficiente	Erro-padrão	Estatística z	Significância
ϕ_0	0,543588	0,002958	183,7440	0,0000
ϕ_1	0,251017	0,029238	8,585341	0,0000
ϕ_2	-0,144619	0,030050	-4,812674	0,0000
α_0	0,000534	0,000148	3,615884	0,0003
α_1	0,098978	0,016851	5,873796	0,0000
β_1	0,852434	0,025338	33,64257	0,0000

Tabela 6.4 Parâmetros estimados para a série NDLBRAD

Com o intuito de verificar se os resíduos e os resíduos ao quadrado pelo modelo AR(2)-GARCH(1,1) podem ser considerados RB foi aplicado o teste Ljung-Box. O resultado do teste para a série NDLBRAD é apresentado na Figura 6.3. Pode-se concluir, a partir da estatística-Q, que os resíduos e os resíduos quadrados gerados pelo modelo AR(2)-GARCH(1,1) podem ser considerados RB.

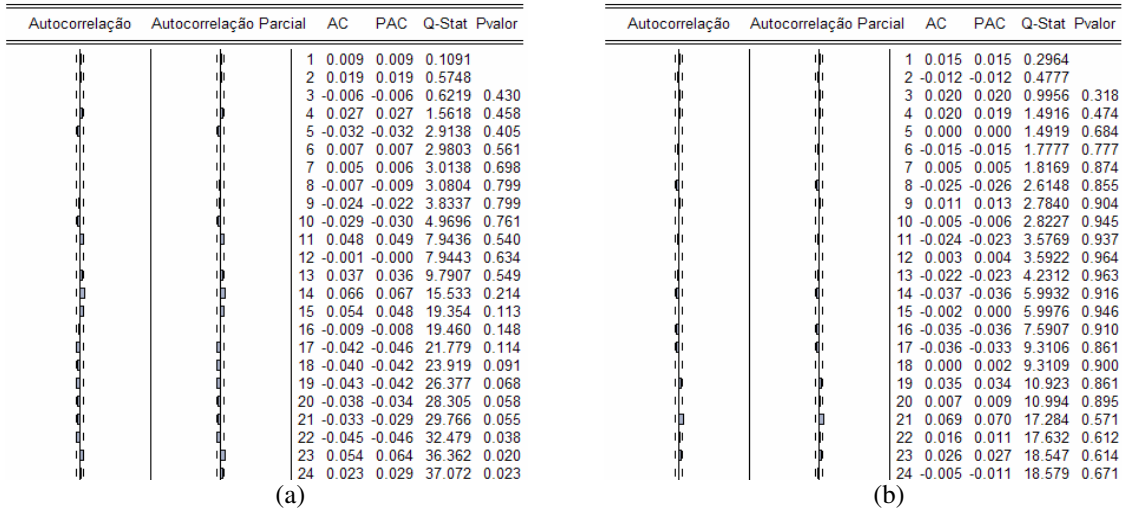


Figura 6.3 (a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado

As previsões realizadas utilizando o modelo ARIMA-GARCH são mostradas na Figura 6.4. A linha contínua indica o valor da série NDLBRAD e a linha central com marcador quadrado indica as previsões.

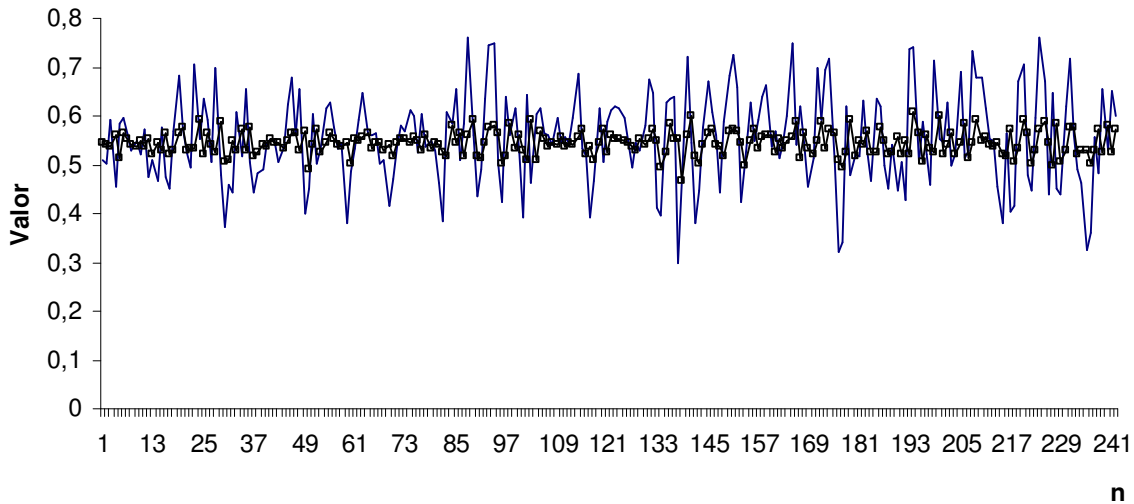


Figura 6.4 Previsão obtida pelo modelo ARIMA-GARCH para NDLBRAD

Na Figura 6.5a é mostrado o resultado da previsão da volatilidade obtida pelo modelo ARIMA-GARCH ajustado para a série. Na Figura 6.5b é apresentada a previsão e o intervalo de previsão para a série NDLBRAD

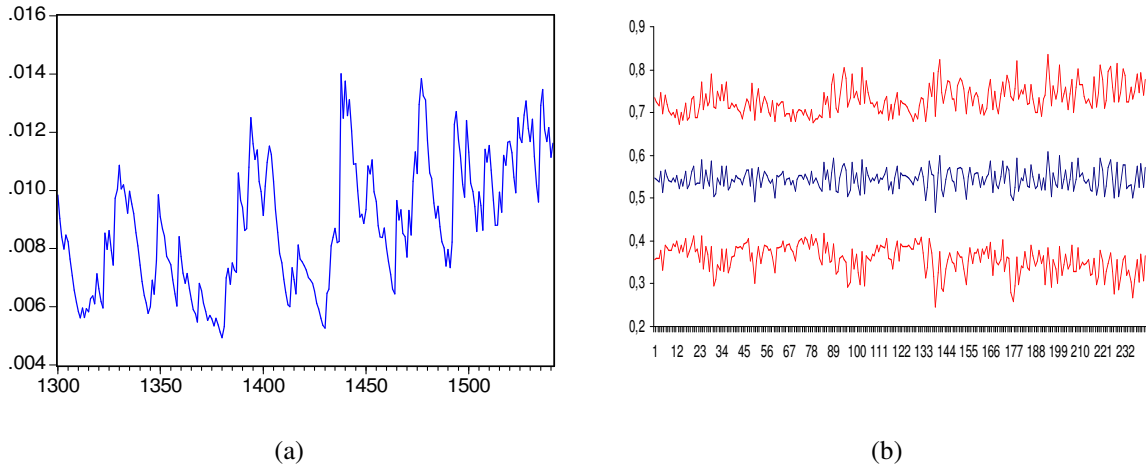


Figura 6.5 (a) Previsão da Volatilidade, (b) Intervalo de previsão obtida pelo modelo ARIMA-GARCH para NDLBRAD

6.2.1.2 Análise da série do Bradesco – Método RNA *feedforward*

Na Tabela 6.5 são apresentadas as medidas de performance FIC (fora do intervalo de confiança), FIP (fora do intervalo de previsão), TIC (coeficiente de desigualdade de Theil) e RMSE(raiz do erro quadrado médio) para diferentes topologias de RNA *feedforward*.

Topologia	FIC	FIP	TIC	RMSE
RNA 10-1-1	132	0	0,0800184	0,0882636
RNA 10-2-1	130	0	0,0828567	0,0913284
RNA 10-3-1	135	0	0,0808077	0,0889541
RNA 10-4-1	133	0	0,0820377	0,0905547
RNA 10-5-1	134	0	0,0810544	0,0895251
RNA 10-8-1	116	0	0,082537	0,0909994

Tabela 6.5 Estatísticas para diferentes topologias de RNA para a série

Os melhores resultados de previsão foram obtidos pela RNA 10-1-1, conforme pode ser verificado pelo TIC e pelo RMSE.

Na Figura 6.6 são apresentadas as previsões obtidas pela RNA 10-1-1.

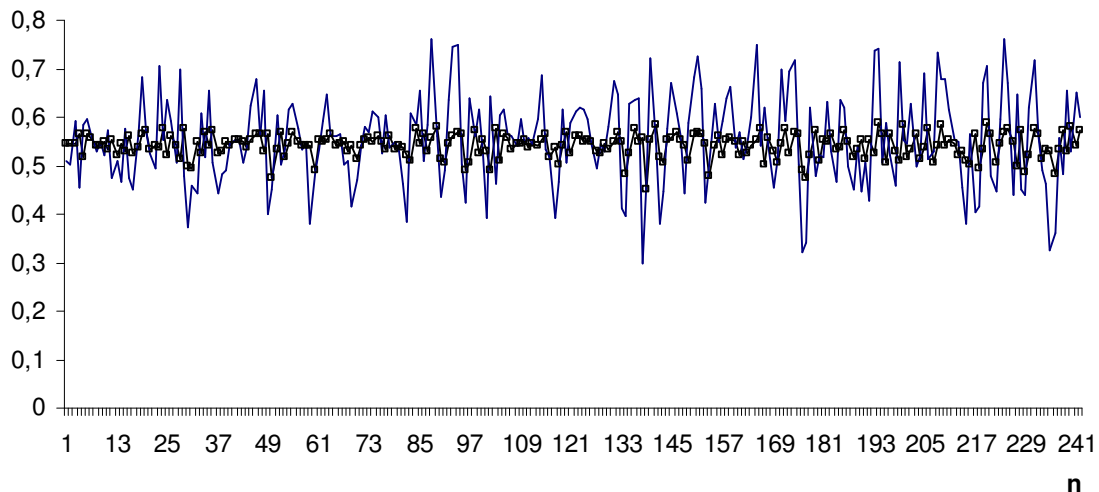


Figura 6.6 Previsão obtida pela RNA *feedforward* para NDLBRAD

Na Figura 6.7a é apresentado o intervalo de confiança para as previsões obtidas pela RNA 10-1-1 e na Figura 6.7b o intervalo de predição.

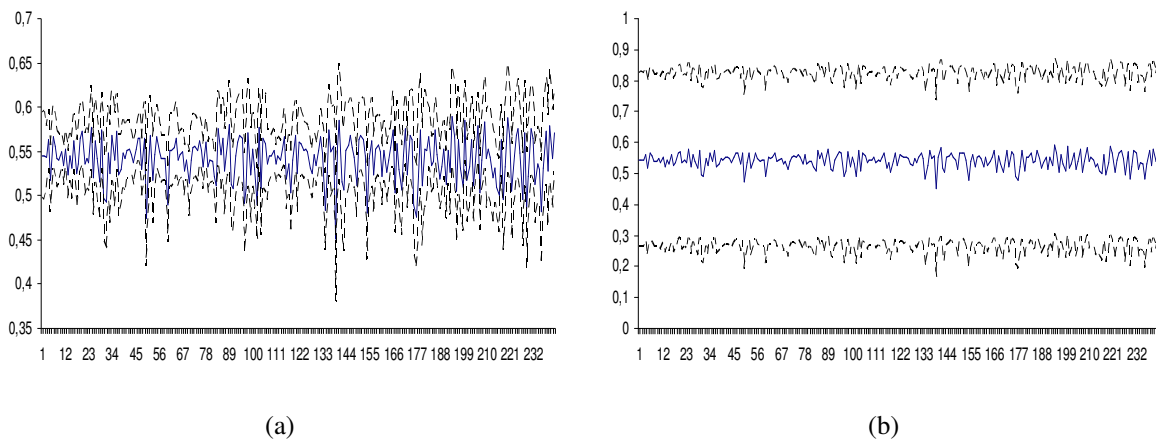


Figura 6.7 (a) Intervalo de confiança, (b) Intervalo de predição para NDLBRAD

Na Figura 6.8a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA 10-1-1 e na Figura 6.8b o histograma desses resíduos. Nestas figuras tem-se uma medida descritiva do comportamento dos resíduos com relação a normalidade.

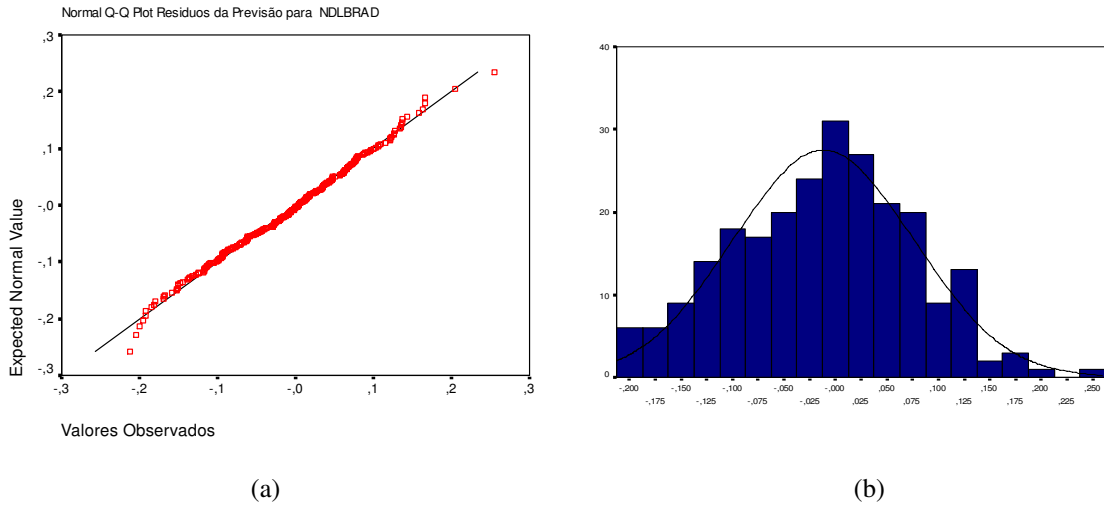


Figura 6.8 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA *feedforward* para NDLBRAD

6.2.1.3 Análise da série do Bradesco – Método RNA EKF

As previsões obtidas pela RNA treinada com o filtro de Kalman estendido (EKF) são mostradas na Figura 6.9.

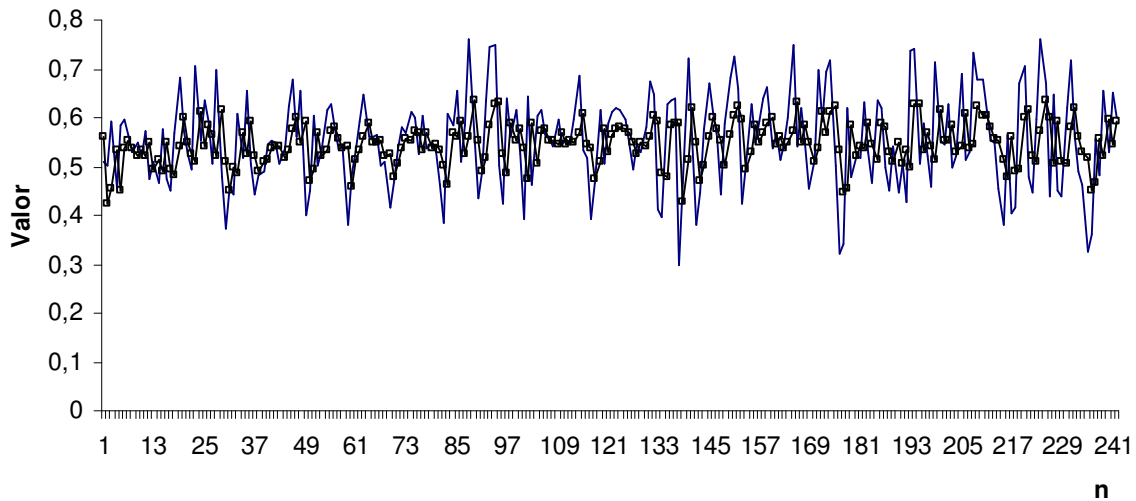


Figura 6.9 Previsão obtida pela RNA EKF para NDLBRAD

Na Figura 6.10a é apresentado o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman estendido e na Figura 6.10b o histograma desses resíduos. Os resíduos têm distribuição normal quanto mais se aproximarem da linha reta.

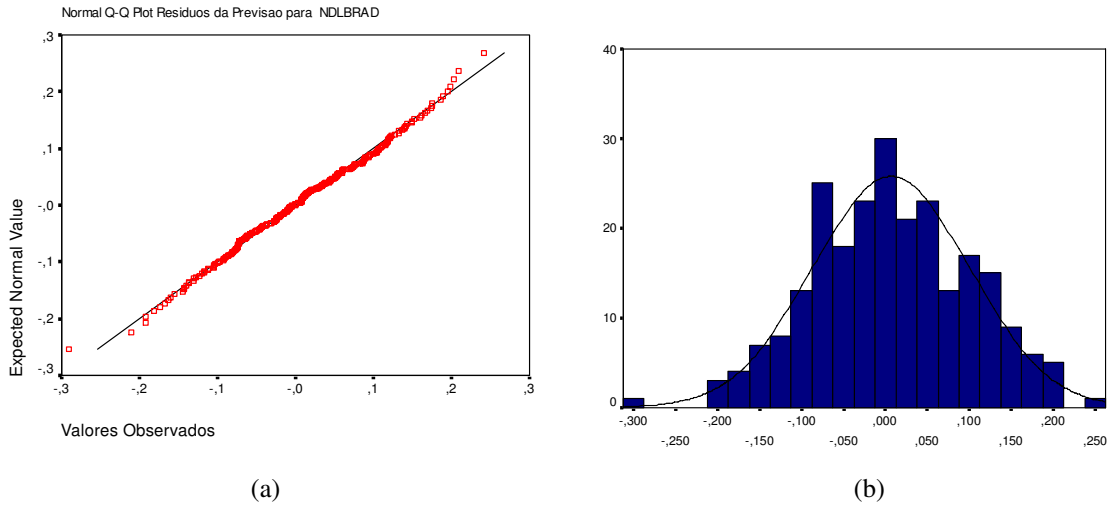


Figura 6.10 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLBRAD

6.2.1.4 Análise da série do Bradesco – Método RNA UKF

As previsões obtidas pela RNA treinada com o filtro de Kalman unscented são mostradas na Figura 6.11.

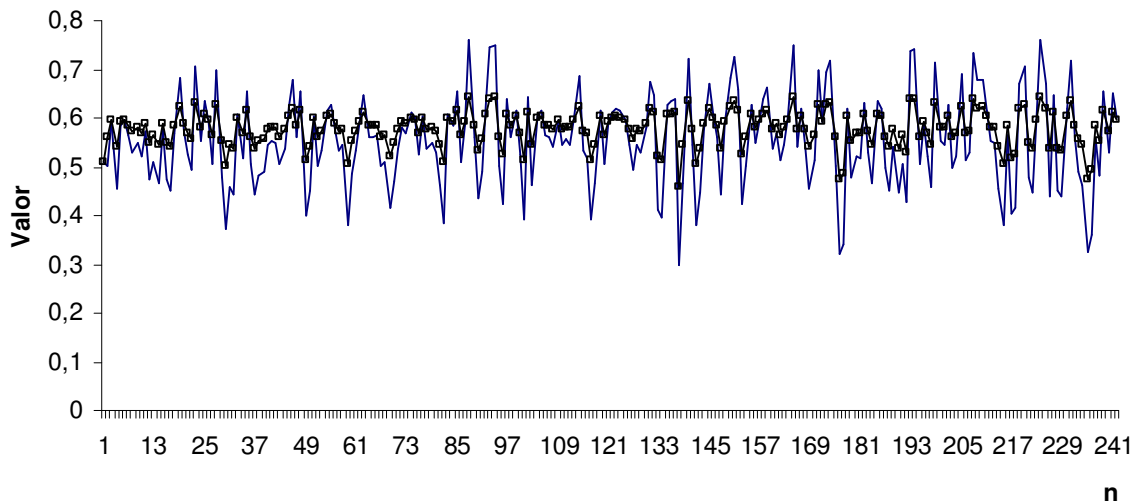


Figura 6.11 Previsão obtida pela RNA UKF para NDLBRAD

Na Figura 6.12a é apresentado o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman *unscented* e na Figura 6.12b o histograma desses resíduos.

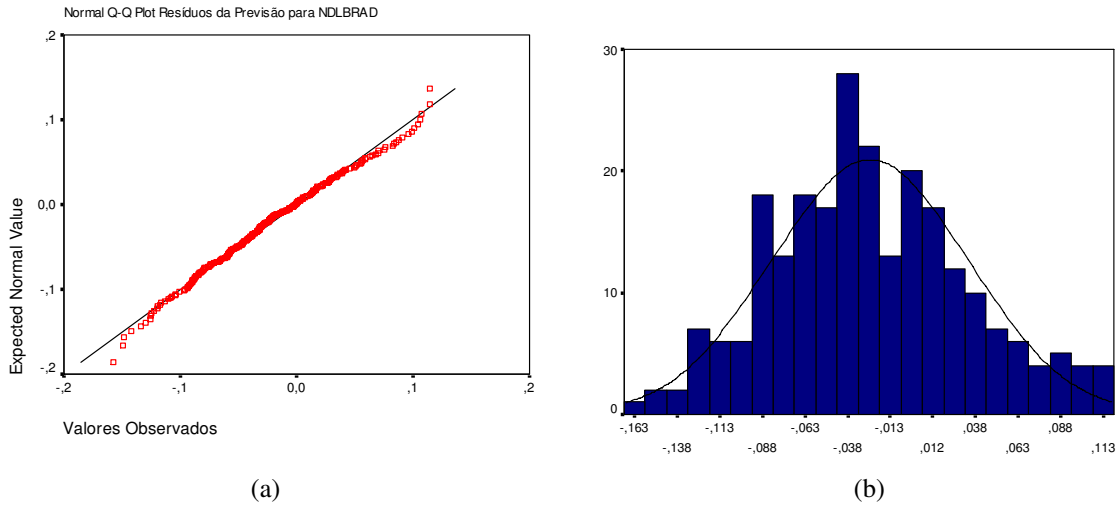


Figura 6.12 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLBRAD

As estatísticas de previsão obtidas pelos modelos ARIMA-GARCH, RNA *feedforward*, RNA EKF e RNA UKF são apresentadas na Tabela 6.6 e na Figura 6.13.

Método	TIC	RMSE
AR(2)-GARCH(1,1)	0,088158	0,079703
RNA 10-1-1	0,080018	0,088264
RNA EKF	0,084257	0,093454
RNA UKF	0,054779	0,062430

Tabela 6.6 Estatísticas de previsão para o método de ARIMA-GARCH e de RNA

As previsões dos modelos ARIMA-GARCH foram realizadas no programa Eviews versão 5.0, as previsões utilizam os modelos ARIMA-GARCH para realizar as previsões da média, erro padrão e variância condicional.

Pode-se verificar que as melhores previsões foram obtidas pelo método que empregou redes neurais treinadas com o filtro de Kalman *unscented*.

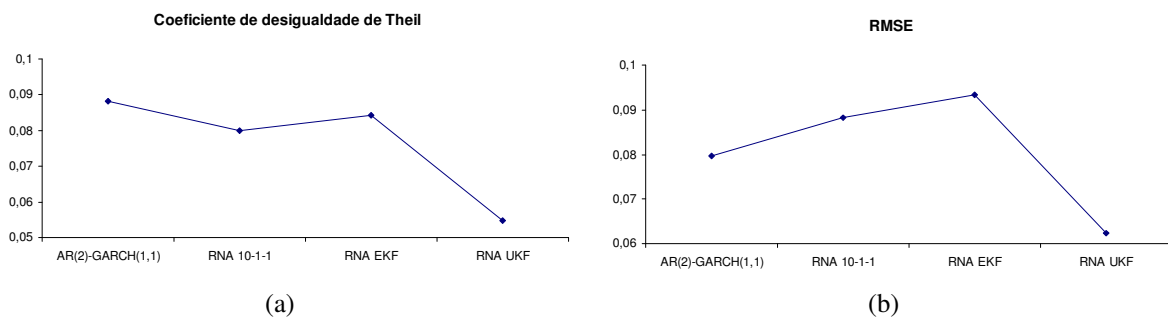


Figura 6.13 Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA

6.2.2 Análise da série do Itaú

A seguir são apresentados os resultados obtidos para a série do Itaú.

6.2.2.1 Análise da série do Itaú – Método ARIMA-GARCH

Na Figura 6.14a é apresentada a série original do preço de fechamento do Itaú, na Figura 6.14b é apresentada a série normalizada da primeira diferença do logaritmo do preço, NDLITAU.

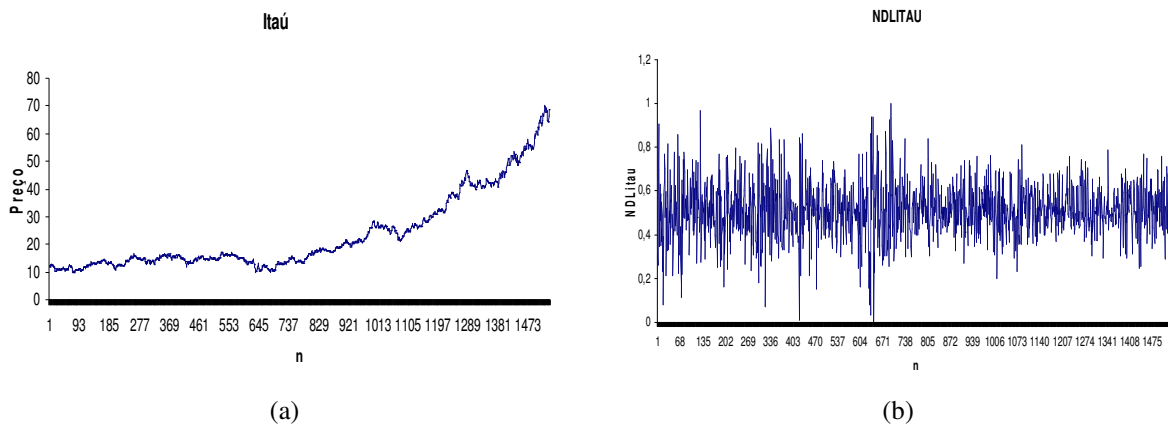


Figura 6.14 (a) Série no nível do preço do Itaú, (b) Série transformada NDLITAU

O resultado do teste ADF, para a série original, é apresentado na Tabela 6.7.

Estatística do teste ADF para série original	2,81294	valores críticos 1%	-3,43439
		valores críticos 5%	-2,86321
		valores críticos 10%	-2,56771

Tabela 6.7 Teste ADF para série original do Itaú

Não é rejeitada a hipótese de raiz unitária para a série no nível pelo teste ADF. O teste de presença de uma raiz unitária para a série transformada apresentou os seguintes resultados:

Estatística do teste ADF para série transformada	-28,4563	valores críticos 1%	-3,43439
		valores críticos 5%	-2,86321
		valores críticos 10%	-2,56771

Tabela 6.8 Teste ADF para série NDLITAU

Na Figura 6.15 é apresentada a estatística de Jarque-Bera, que rejeita fortemente a hipótese de distribuição normal.

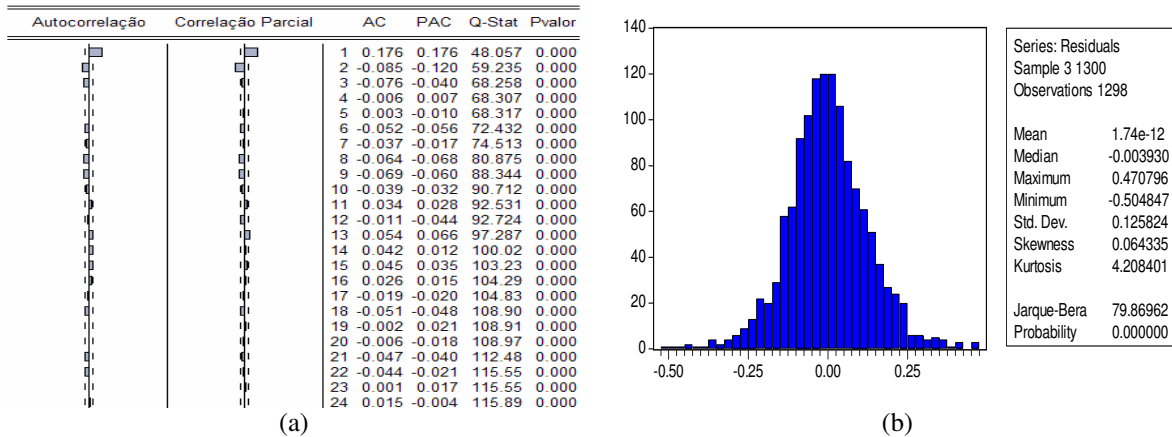


Figura 6.15 (a) Correlograma da série NDLITAU, (b) Estatísticas dos resíduos

Os resultados para o teste BDS são apresentados na Tabela 6.9.

BDS				
Dimensão	Estatística	Erro-padrão	Estatística-z	p-valor
2	0,017143	0,002015	8,506016	0,0000
3	0,031895	0,003204	9,955211	0,0000
4	0,041097	0,003816	10,77001	0,0000
5	0,046428	0,003978	11,67143	0,0000
6	0,047102	0,003837	12,27668	0,0000

Tabela 6.9 Resultados obtidos pelo teste BDS nas respectivas dimensões.

O melhor modelo ajustado para a série é um AR(2)-GARCH(1,1). Os parâmetros estimados do modelo para a série são mostrados na Tabela 6.10.

Parâmetros	Coefficiente	Erro-padrão	Estatística-z	p-valor
ϕ_0	0,514898	0,003465	148,5939	0,0000
ϕ_1	0,177387	0,028939	6,129767	0,0000
ϕ_2	-0,106001	0,028855	-3,673637	0,0002
α_0	0,000440	0,000149	2,947703	0,0032
α_1	0,075857	0,013525	5,608568	0,0000
β_1	0,895568	0,017606	50,86615	0,0000

Tabela 6.10 Parâmetros estimados para a série NDLITAU

O resultado do teste Ljung-Box para a série NDLITAU é apresentado na Figura 6.16.

Autocorrelação	Autocorrelação Parcial	AC	PAC	Q-Stat	Pvalor
		1 0.019	0.019	0.4755	
		2 0.016	0.015	0.8016	
		3 -0.021	-0.022	1.3956	0.237
		4 -0.004	-0.004	1.4197	0.492
		5 0.010	0.011	1.5474	0.671
		6 -0.032	-0.033	2.8739	0.579
		7 0.009	0.009	2.9734	0.704
		8 -0.043	-0.042	5.3950	0.494
		9 -0.058	-0.058	9.7273	0.205
		10 -0.013	-0.010	9.9490	0.269
		11 0.029	0.030	11.046	0.273
		12 -0.024	-0.029	11.789	0.299
		13 0.063	0.064	16.953	0.109
		14 0.019	0.017	17.410	0.135
		15 0.023	0.017	18.097	0.154
		16 -0.004	-0.004	18.118	0.201
		17 -0.020	-0.021	18.639	0.231
		18 -0.058	-0.066	23.146	0.110
		19 0.006	0.014	23.189	0.143
		20 -0.009	-0.008	23.289	0.180
		21 -0.016	-0.016	23.646	0.210
		22 -0.016	-0.007	23.972	0.244
		23 0.021	0.029	24.543	0.268
		24 0.020	0.013	25.088	0.293

Autocorrelação	Autocorrelação Parcial	AC	PAC	Q-Stat	Pvalor
		1 0.043	0.043	2.3560	
		2 0.009	0.007	2.4661	
		3 -0.002	-0.002	2.4694	0.116
		4 0.012	0.012	2.6615	0.264
		5 -0.008	-0.009	2.7516	0.432
		6 0.007	0.008	2.8226	0.588
		7 -0.015	-0.015	3.1086	0.683
		8 -0.014	-0.013	3.3478	0.764
		9 0.001	0.003	3.3489	0.851
		10 0.015	0.015	3.6341	0.889
		11 -0.034	-0.034	5.1097	0.825
		12 -0.011	-0.008	5.2631	0.873
		13 0.017	0.019	5.6454	0.896
		14 -0.005	-0.007	5.6774	0.931
		15 0.003	0.004	5.6884	0.957
		16 -0.033	-0.034	7.1480	0.929
		17 -0.043	-0.040	9.5519	0.847
		18 -0.003	0.001	9.5677	0.888
		19 0.009	0.008	9.6777	0.917
		20 0.006	0.006	9.7278	0.940
		21 0.001	0.002	9.7296	0.959
		22 -0.006	-0.007	9.7779	0.972
		23 -0.031	-0.032	11.015	0.963
		24 0.003	0.005	11.025	0.974

Figura 6.16 (a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado

As previsões realizadas utilizando o modelo ARIMA-GARCH são mostradas na Figura 6.17. A linha contínua indica o valor da série NDLITAU e a linha central com marcador quadrado indica as previsões.

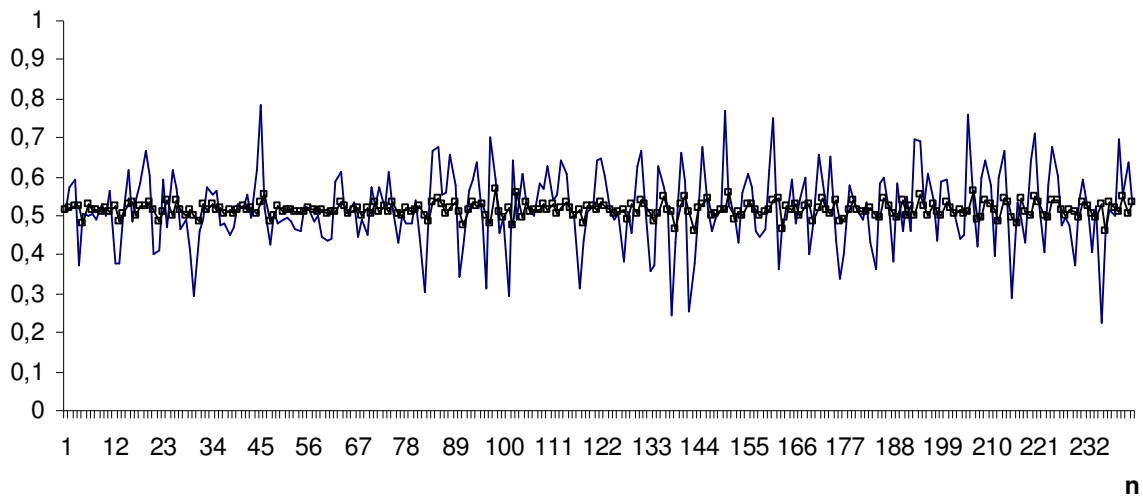


Figura 6.17 Previsão obtida pelo modelo ARIMA-GARCH para NDLITAU

Na Figura 6.18a é mostrada a previsão da volatilidade e na Figura 6.18b a série original e o intervalo de previsão para a série NDLITAU.

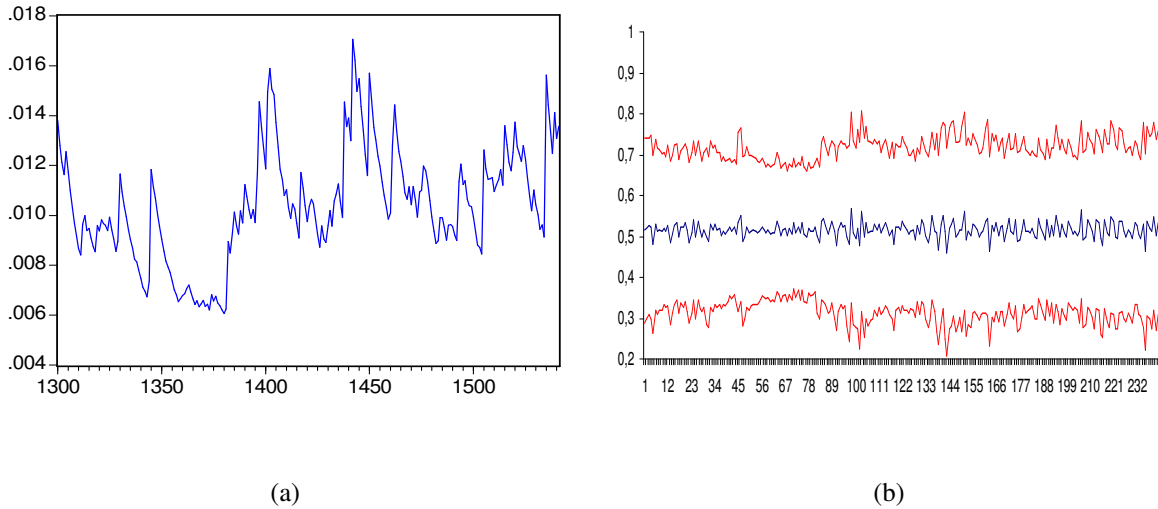


Figura 6.18 (a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLBITAU

6.2.2.2 Análise da série do Itaú – Método RNA *feedforward*

Para a série NDLITAU os resultados obtidos pelas diferentes topologias de RNA são apresentados na Tabela 6.11.

Topologia	FIC	FIP	TIC	RMSE
RNA 10-1-1	121	0	0,0888249	0,0923883
RNA 10-2-1	116	0	0,0901086	0,0933606
RNA 10-3-1	120	0	0,0901503	0,0937411
RNA 10-4-1	117	0	0,0906039	0,0941404
RNA 10-5-1	111	0	0,0903765	0,0941439
RNA 10-8-1	104	0	0,0934771	0,0971403

Tabela 6.11 Estatísticas para diferentes topologias de RNA para a série

Na Figura 6.19 são apresentadas as previsões obtidas pela RNA 10-1-1 para a série NDLITAU.

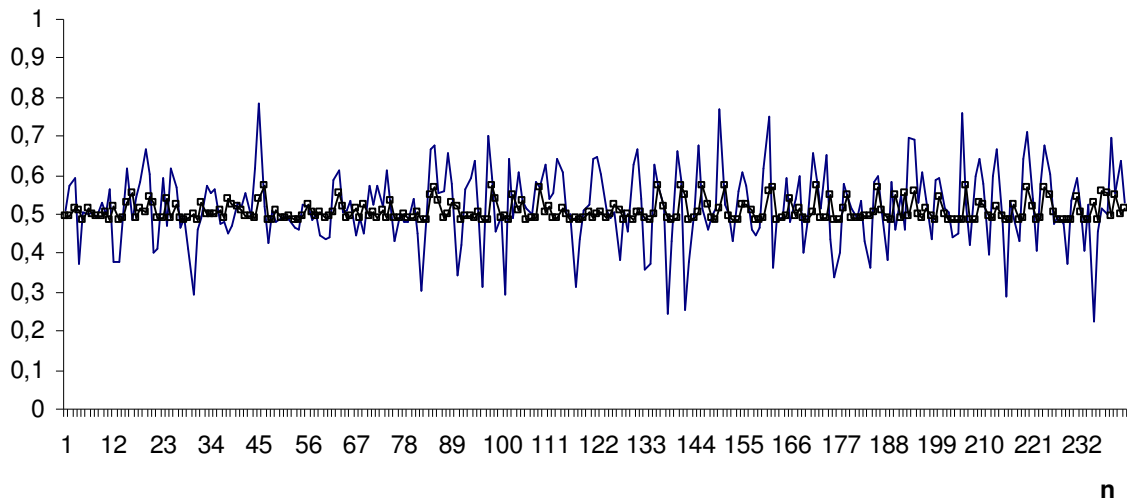


Figura 6.19 Previsão obtida pela RNA *feedforward* para NDLITAU

Na Figura 6.20a é mostrado o intervalo de confiança para as previsões obtidas pela RNA 10-1-1 e na Figura 6.20b o intervalo de predição para a série NDLITAU.

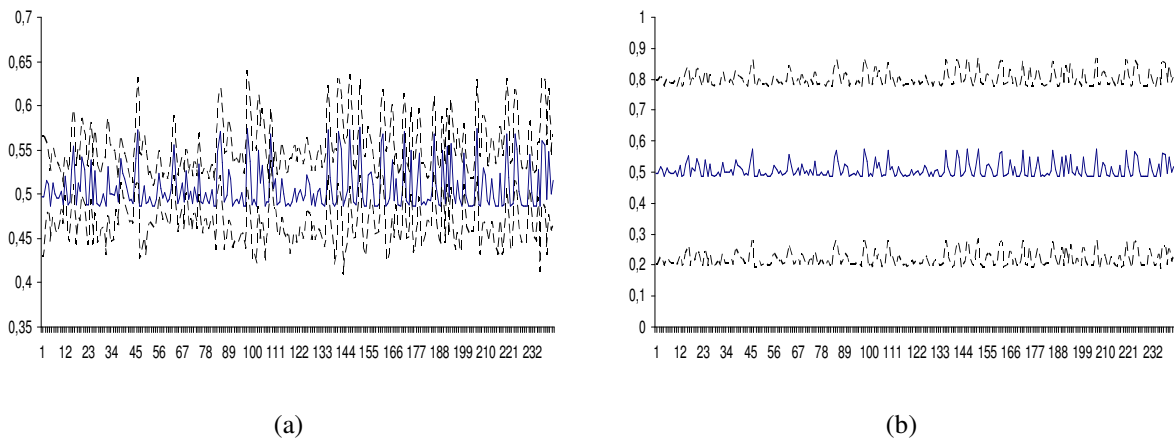


Figura 6.20 (a) Intervalo de confiança, (b) Intervalo de predição para NDLITAU

Na Figura 6.21a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA 10-1-1 e na Figura 6.21b o histograma desses resíduos.

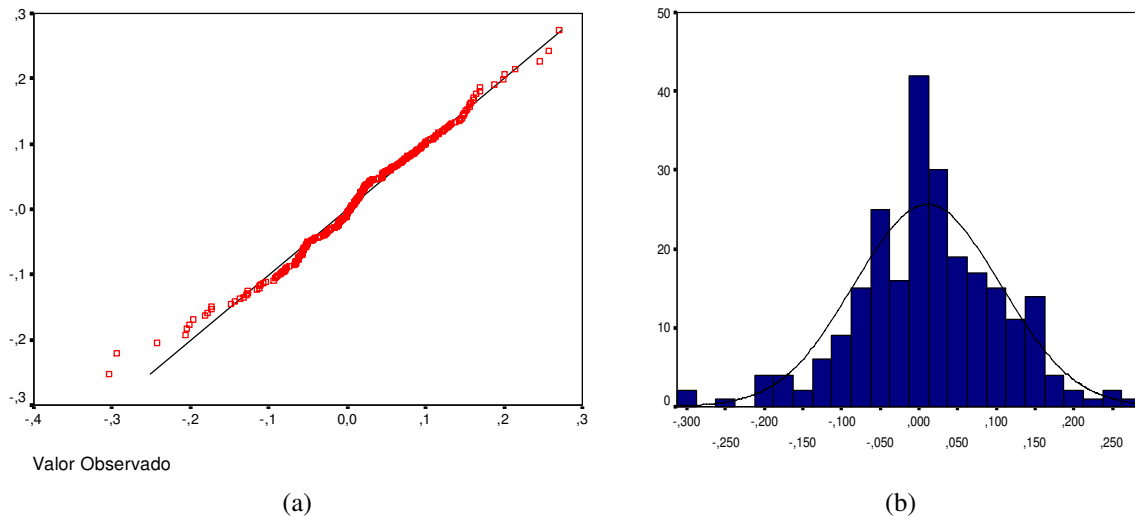


Figura 6.21 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA *feedforward* para NDLITAU

6.2.2.3 Análise da série do Itaú – Método RNA EKF

As previsões, para a série NDLITAU, obtidas pela RNA treinada com o filtro de Kalman estendido (EKF) são mostradas na Figura 6.22.

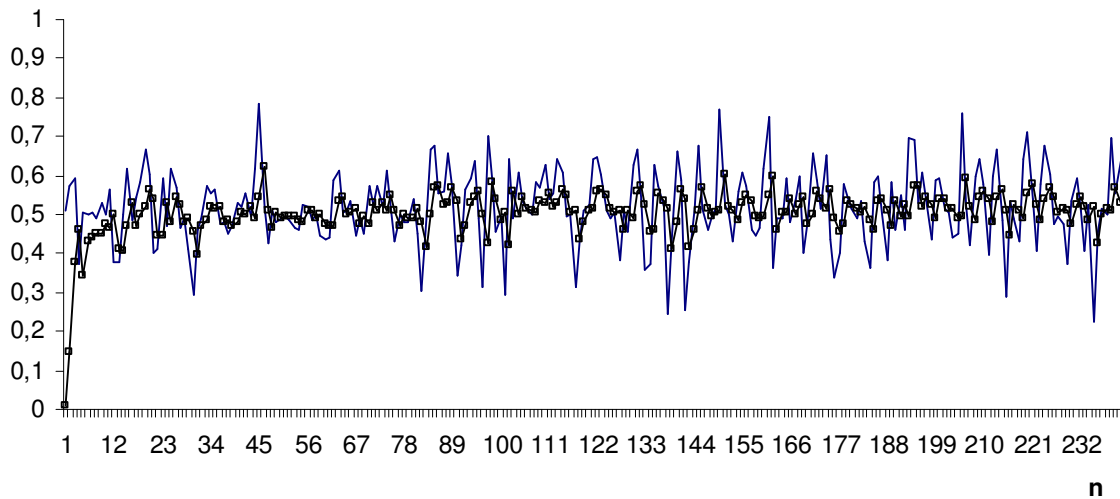


Figura 6.22 Previsão obtida pela RNA EKF para NDLITAU

Na Figura 6.23a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman estendido e na Figura 6.23b o histograma desses resíduos.

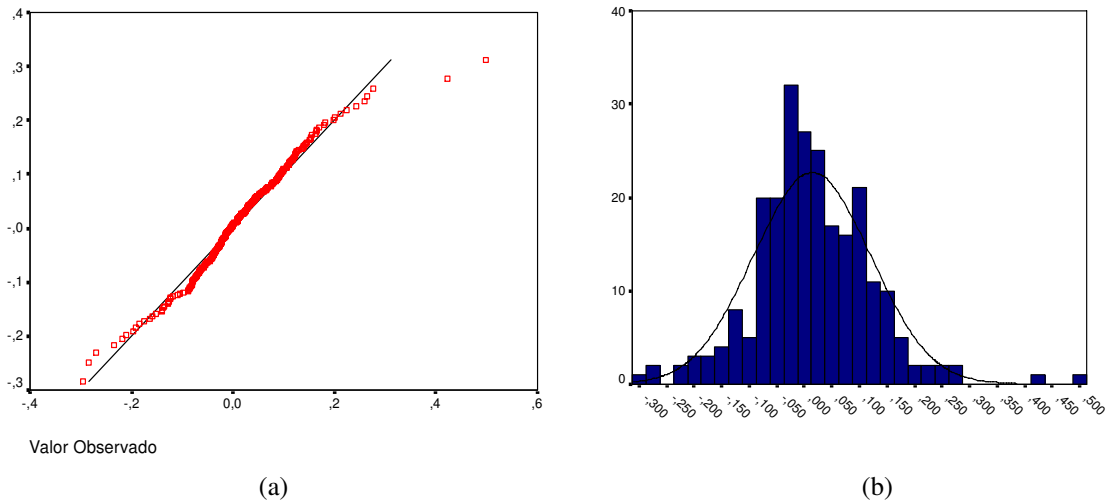


Figura 6.23 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLITAU

6.2.2.4 Análise da série do Itaú – Método RNA UKF

As previsões obtidas pela RNA treinada com o filtro de Kalman *unscented* são mostradas na Figura 6.24.

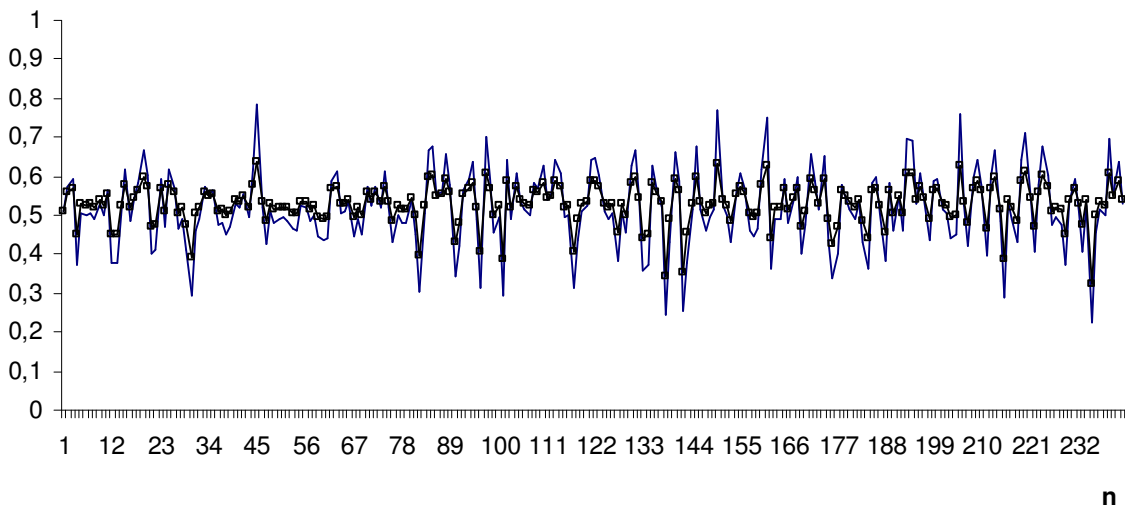


Figura 6.24 Previsão obtida pela RNA UKF para NDLITAU

Na Figura 6.25a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman *unscented* e na Figura 6.25b o histograma desses resíduos.

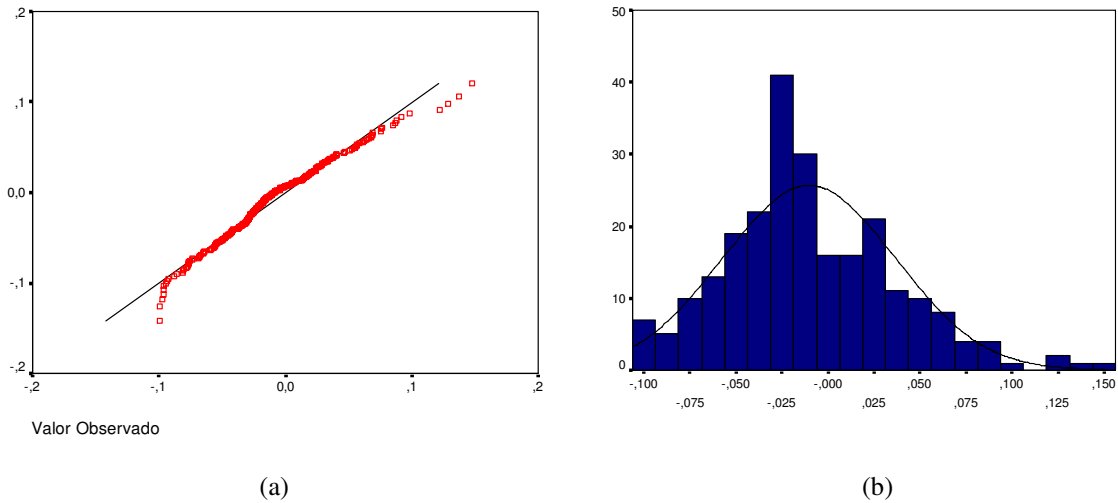


Figura 6.25 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLITAU

As estatísticas de previsão obtidas pelos modelos ARIMA-GARCH, RNA *feedforward*, RNA EKF e RNA UKF são apresentadas na Tabela 6.12 e na Figura 6.26.

Método	TIC	RMSE
AR(2)-GARCH(1,1)	0,109252	0,107997
RNA 10-1-1	0,110965	0,109702
RNA EKF	0,103411	0,107002
RNA UKF	0,045310	0,047958

Tabela 6.12 Estatísticas de previsão para o método de ARIMA-GARCH e de RNA

Pode-se verificar que as melhores previsões foram obtidas pelo método que empregou redes neurais treinadas com o filtro de Kalman *unscented*.

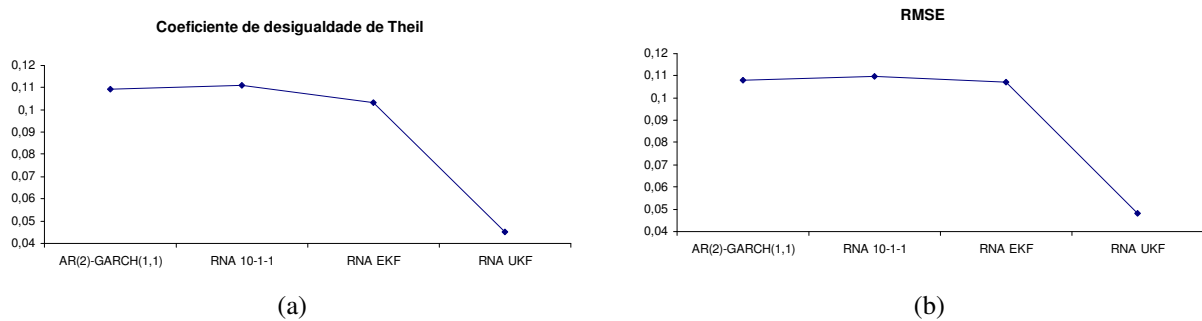


Figura 6.26 Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA

6.2.3 Análise da série da Bradespar

A seguir são apresentados os resultados obtidos para a série da Bradespar.

6.2.3.1 Análise da série da Bradespar – Método ARIMA-GARCH

Na Figura 6.27a é apresentada a série original do preço de fechamento do Itaú, na Figura 6.27b é apresentada a série normalizada da primeira diferença do logaritmo do preço, NDLBPAR.

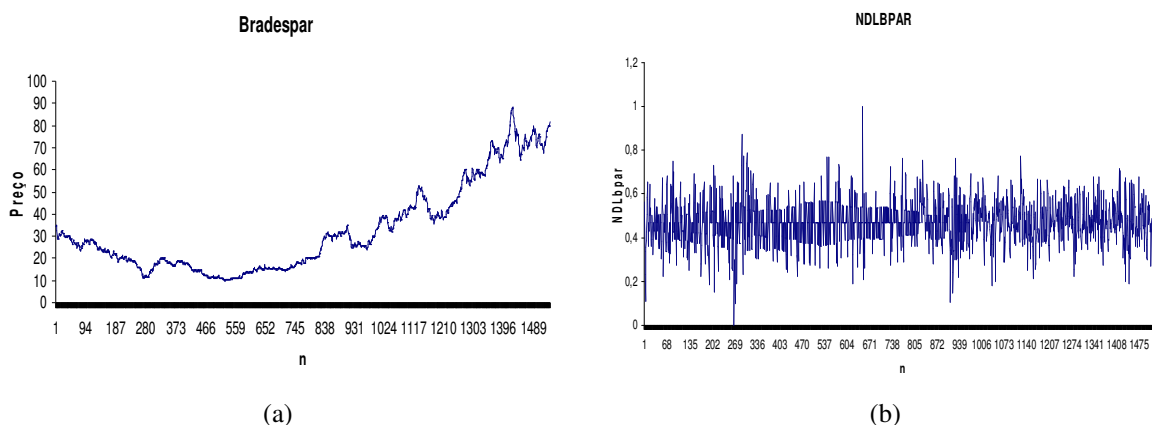


Figura 6.27 (a) Série no nível do preço da Bradespar, (b) Série transformada NDLBPAR

O resultado do teste ADF, para a série original, é apresentado na Tabela 6.13.

Estatística do teste ADF para série original	valores críticos 1%	-3,43439
0,79304	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771

Tabela 6.13 Teste ADF para série original da Bradespar

Não é rejeitada a hipótese de raiz unitária para a série no nível pelo teste ADF. O teste de presença de uma raiz unitária para a série transformada apresentou os seguintes resultados:

Estatística do teste ADF para série transformada	valores críticos 1%	-3,43439
-27,2139	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771

Tabela 6.14 Teste ADF para série NDLBPAR

Na Figura 6.28 é apresentada a estatística de Jarque-Bera, que rejeita fortemente a hipótese de distribuição normal.

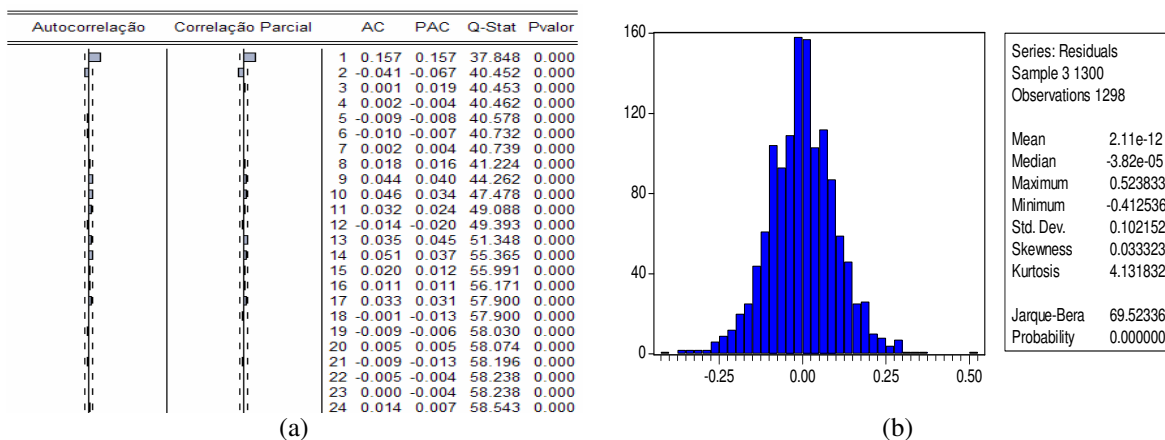


Figura 6.28 (a) Correlograma da série NDLBPAR, (b) Estatísticas dos resíduos

Os resultados para o teste BDS são apresentados na Tabela 6.15.

BDS				
Dimensão	Estatística	Erro-padrão	Estatística-z	p-valor
2	0,011597	0,001904	6,089998	0,0000
3	0,022113	0,003014	7,336243	0,0000
4	0,028470	0,003575	7,964577	0,0000
5	0,030789	0,003710	8,298934	0,0000
6	0,030450	0,003563	8,547009	0,0000

Tabela 6.15 Resultados obtidos pelo teste BDS nas respectivas dimensões.

O melhor modelo ajustado para a série é um AR(2)-GARCH(1,1). Os parâmetros estimados do modelo para a série são mostrados na Tabela 6.16.

Parâmetros	Coefficiente	Erro-padrão	Estatística-z	p-valor
ϕ_0	0,470231	0,003039	154,7287	0,0000
ϕ_1	0,179742	0,030519	5,889588	0,0000
ϕ_2	-0,091862	0,029676	-3,095551	0,0020
α_0	0,002356	0,000659	3,576356	0,0003
α_1	0,114771	0,027280	4,207129	0,0000
β_1	0,657739	0,083444	7,882397	0,0000

Tabela 6.16 Parâmetros estimados para a série NDLBPAR

O resultado do teste Ljung-Box para a série NDLBPAR é apresentado na Figura 6.29.

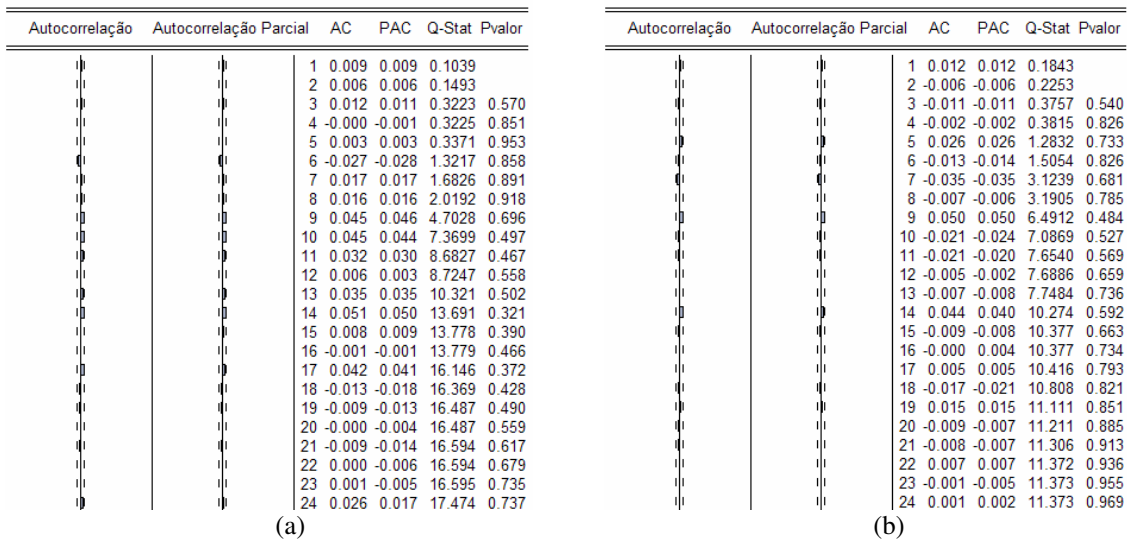


Figura 6.29 (a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado

As previsões realizadas utilizando o modelo ARIMA-GARCH são mostradas na Figura 6.30. A linha contínua indica o valor da série NDLBPARG e a linha central com marcador quadrado indica as previsões.

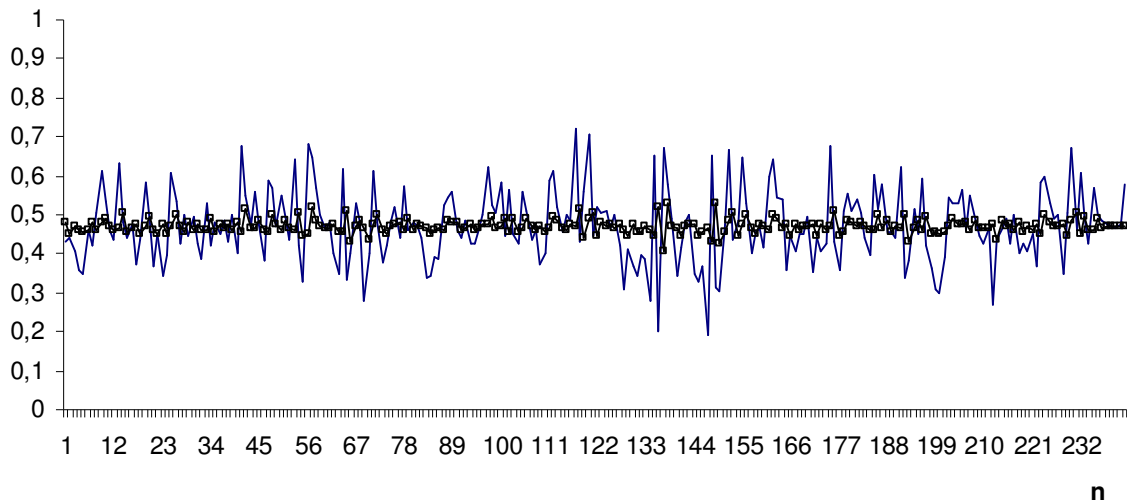


Figura 6.30 Previsão obtida pelo modelo ARIMA-GARCH para NDLBPARG

Na Figura 6.31a é mostrada a previsão da volatilidade e na Figura 6.31b a série original e o intervalo de previsão para a série NDLBPARG.

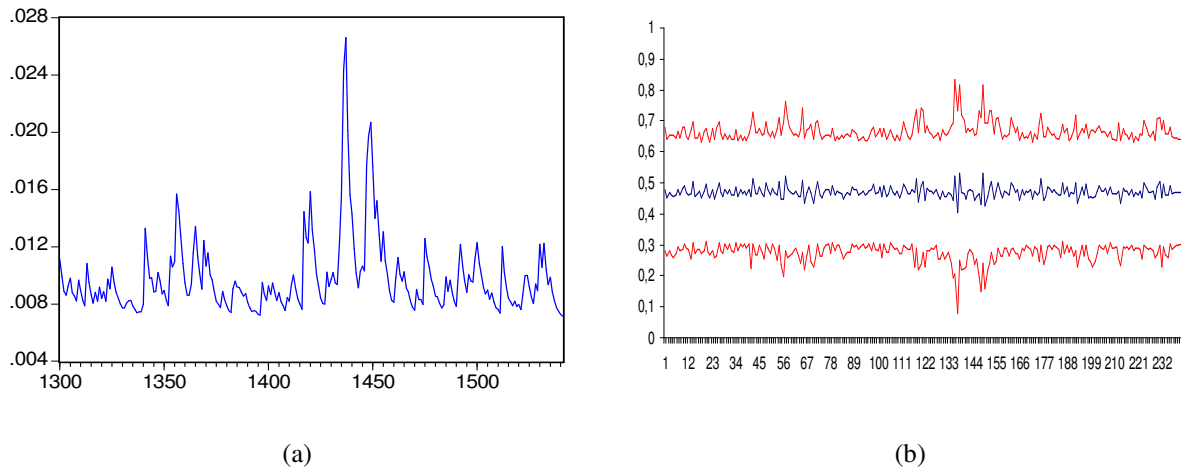


Figura 6.31 (a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLBPAR

6.2.3.2 Análise da série da Bradespar – Método RNA *feedforward*

Para a série NDLBPAR os resultados obtidos pelas diferentes topologias de RNA são apresentados na Tabela 6.17.

Topologia	FIC	FIP	TIC	RMSE
RNA 10-1-1	120	0	0,0980901	0,0934057
RNA 10-2-1	109	0	0,0995521	0,0947912
RNA 10-3-1	109	1	0,136706	0,130858
RNA 10-4-1	116	0	0,101252	0,0965534
RNA 10-5-1	102	1	0,160326	0,154261
RNA 10-8-1	94	0	0,102588	0,0976335

Tabela 6.17 Estatísticas para diferentes topologias de RNA para a série

Na Figura 6.32 são apresentadas as previsões obtidas pela RNA 10-1-1 para a série NDLBPAR.

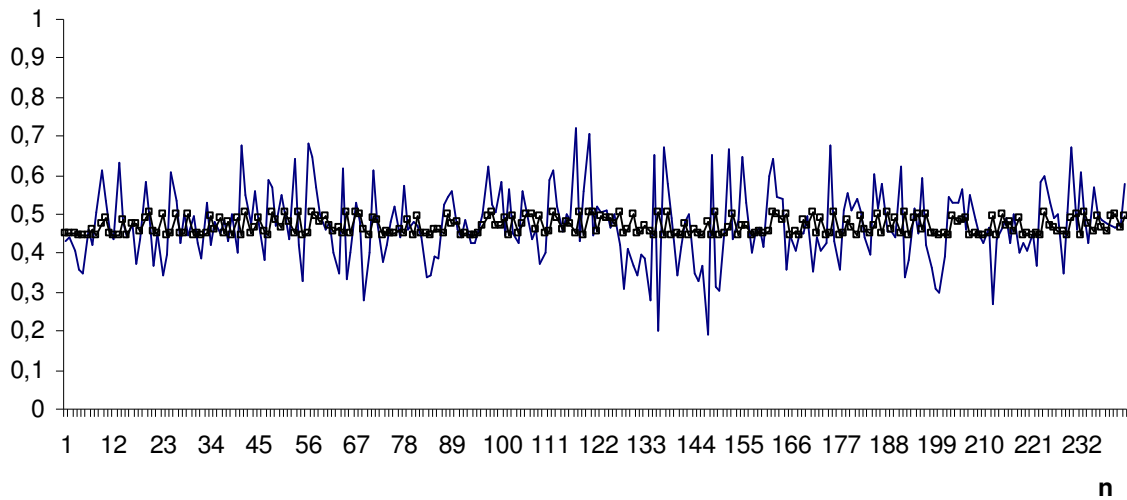


Figura 6.32 Previsão obtida pela RNA *feedforward* para NDLBP

Na Figura 6.33a é mostrado o intervalo de confiança para as previsões obtidas pela RNA 10-1-1 e na Figura 6.33b o intervalo de predição para a série NDLBP.

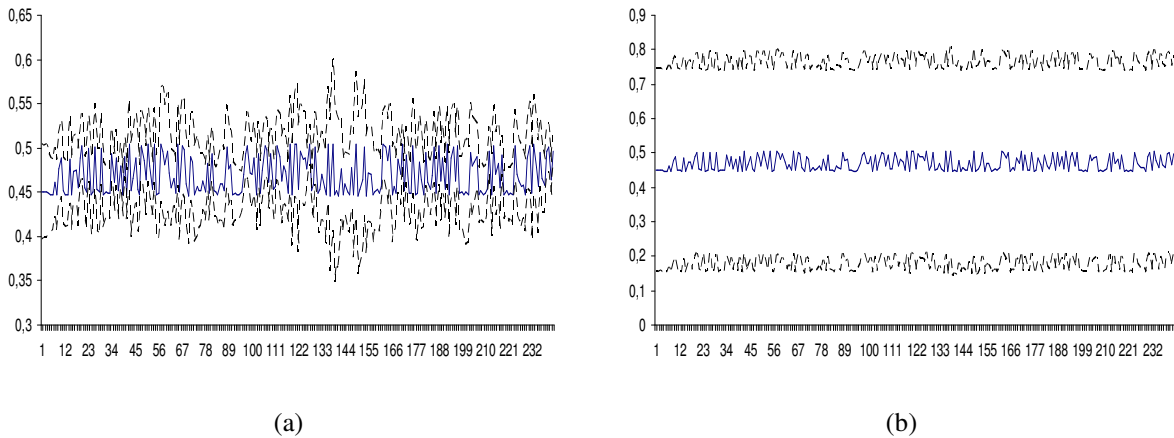


Figura 6.33 (a) Intervalo de confiança, (b) Intervalo de predição para NDLBP

Na Figura 6.34a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA 10-1-1 e na Figura 6.34b o histograma desses resíduos.

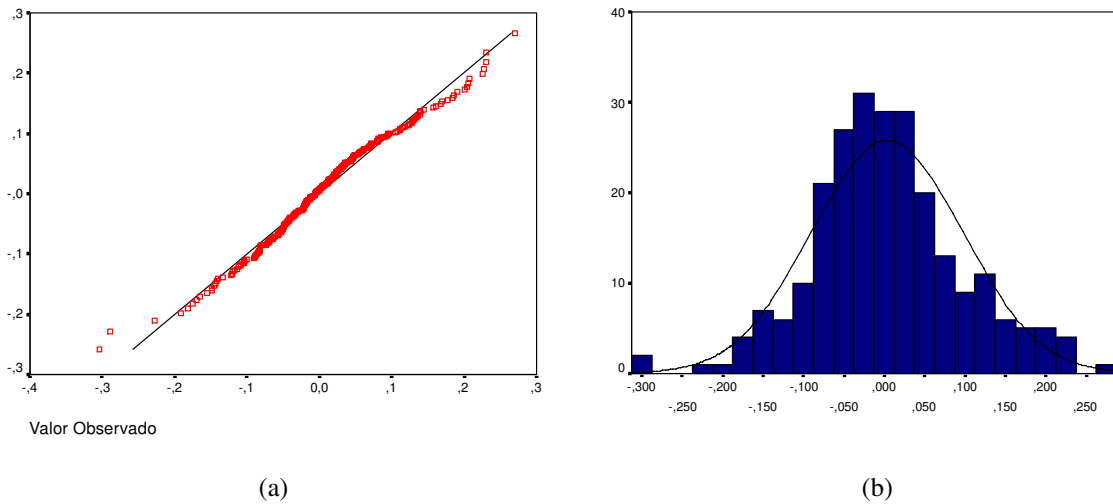


Figura 6.34 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA *feedforward* para NDLBPAR

6.2.3.3 Análise da série da Bradespar – Método RNA EKF

As previsões, para a série NDLBPAR, obtidas pela RNA treinada com o filtro de Kalman estendido (EKF) são mostradas na Figura 6.35.

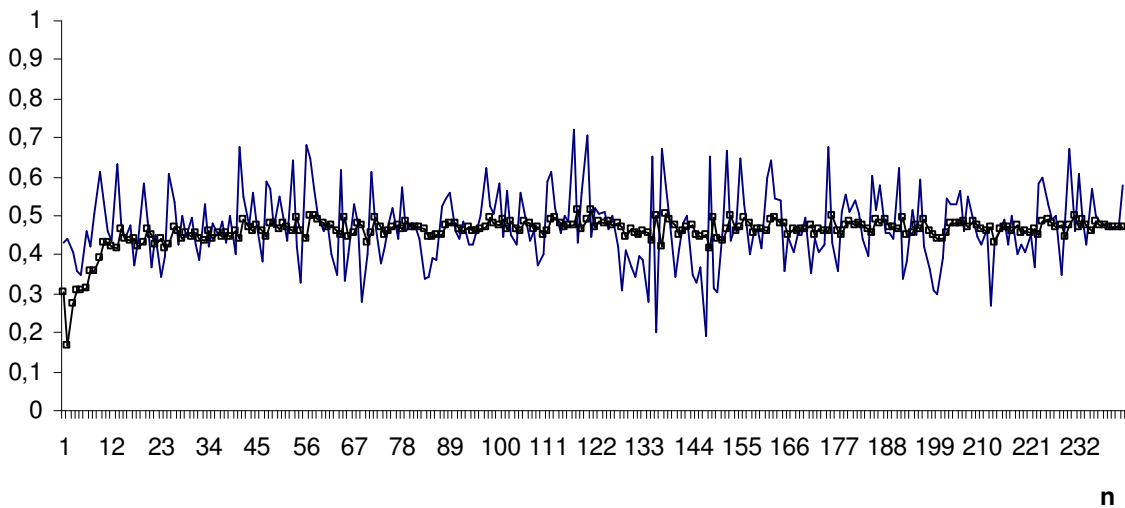


Figura 6.35 Previsão obtida pela RNA EKF para NDLBPAR

Na Figura 6.36a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman estendido e na Figura 6.36b o histograma desses resíduos.

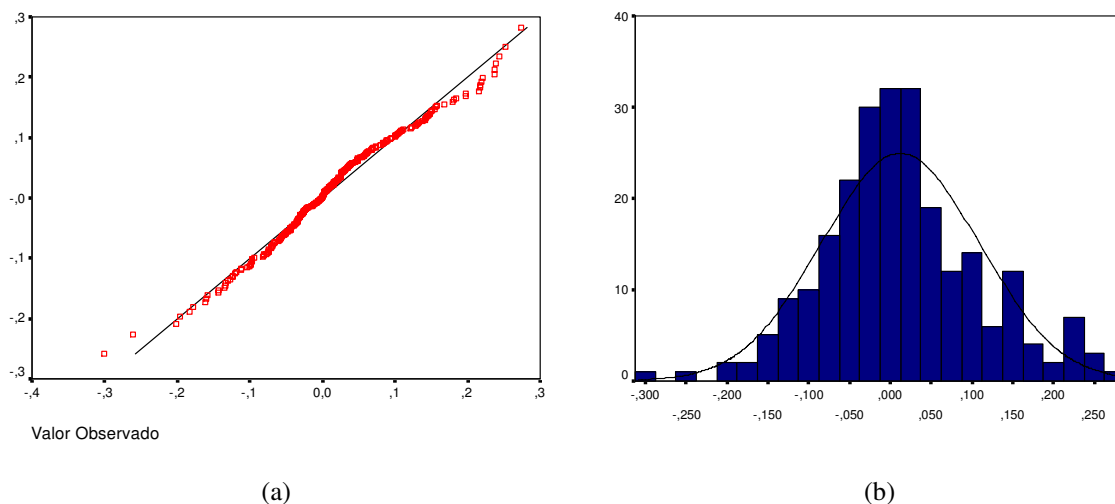


Figura 6.36 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLPAR

6.2.3.4 Análise da série da Bradespar – Método RNA UKF

As previsões obtidas pela RNA treinada com o filtro de Kalman *unscented* são mostradas na Figura 6.37.

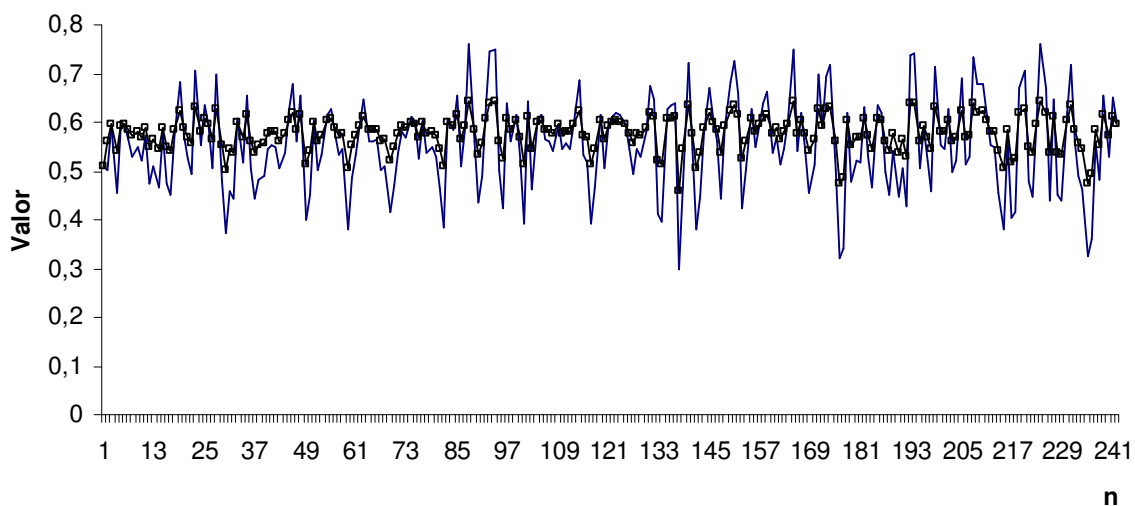


Figura 6.37 Previsão obtida pela RNA UKF para NDLPAR

Na Figura 6.38a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman *unscented* e na Figura 6.38b o histograma desses resíduos.

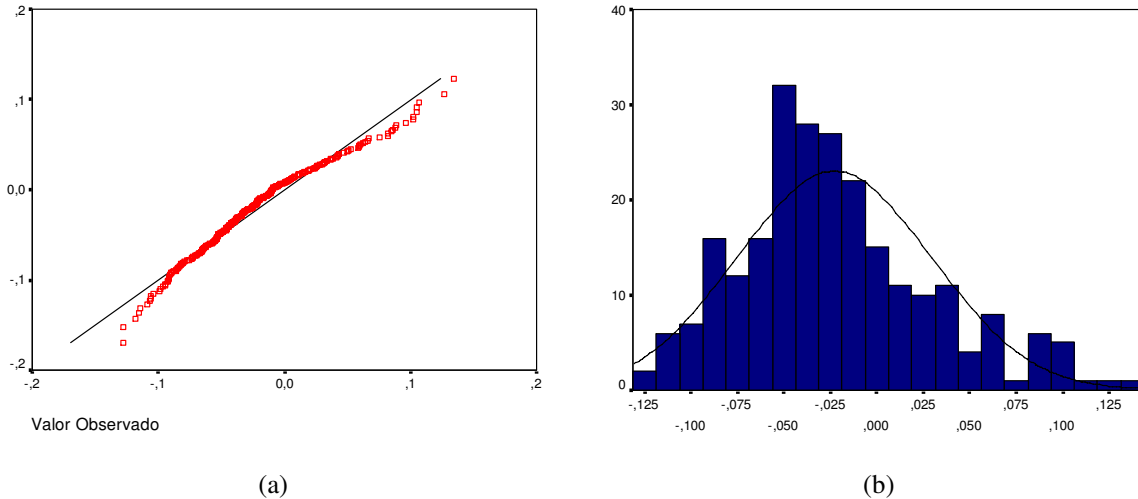


Figura 6.38 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLBPAR

As estatísticas de previsão obtidas pelos modelos ARIMA-GARCH, RNA *feedforward*, RNA EKF e RNA UKF são apresentadas na Tabela 6.18 e na Figura 6.39.

Método	TIC	RMSE
AR(2)-GARCH(1,1)	0,097353	0,092753
RNA 10-1-1	0,098090	0,093406
RNA EKF	0,102973	0,097193
RNA UKF	0,058222	0,057022

Tabela 6.18 Estatísticas de previsão para o método de ARIMA-GARCH e de RNA

Pode-se verificar que as melhores previsões foram obtidas pelo método que empregou redes neurais treinadas com o filtro de Kalman *unscented*.

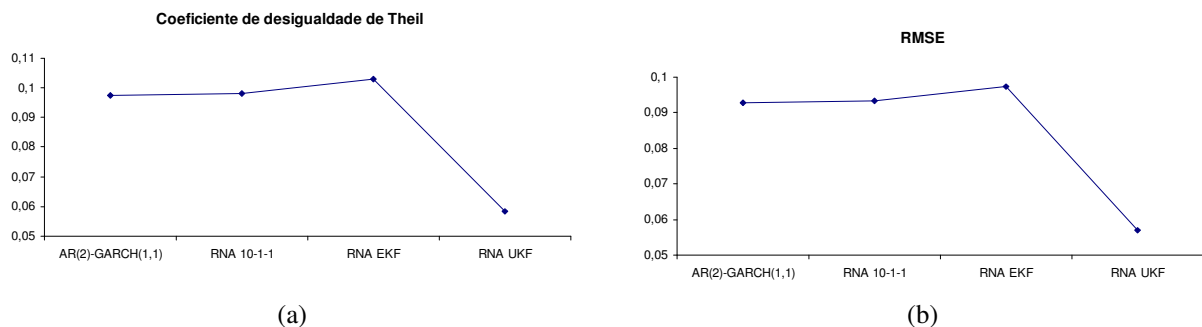


Figura 6.39 Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA

6.2.4 Análise da série da Itausa

A seguir são apresentados os resultados obtidos para a série do Itausa.

6.2.4.1 Análise da série da Itausa – Método ARIMA-GARCH

Na Figura 6.40a é apresentada a série original do preço de fechamento da Itausa, na Figura 6.40b é apresentada a série normalizada da primeira diferença do logaritmo do preço, NDLIUSA.

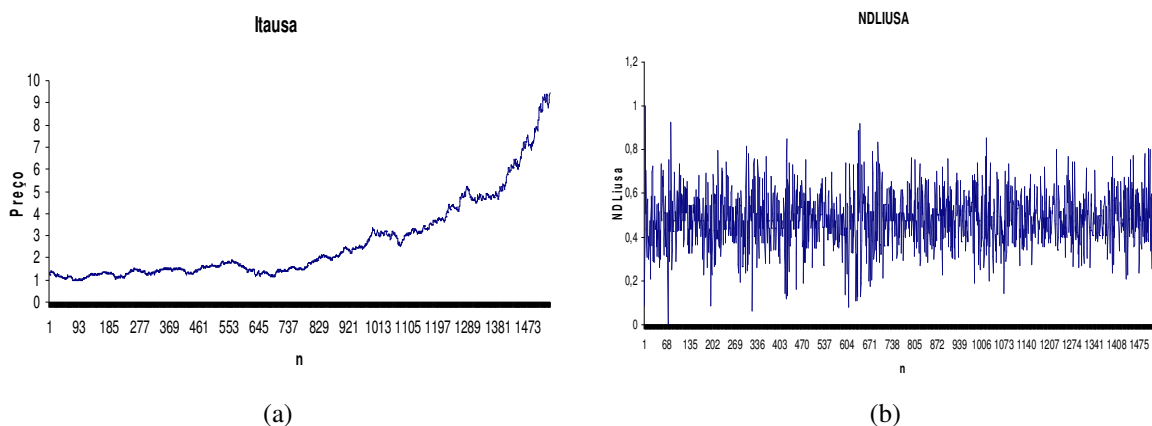


Figura 6.40 (a) Série no nível do preço do Bradesco, (b) Série transformada NDLIUSA

O resultado do teste ADF, para a série original, é apresentado na Tabela 6.19.

Estatística do teste ADF para série original	valores críticos 1%	-3,43439	
	4,59791	valores críticos 5%	-2,86321
		valores críticos 10%	-2,56771

Tabela 6.19 Teste ADF para série original do Itausa

Não é rejeitada a hipótese de raiz unitária para a série no nível pelo teste ADF. O teste de presença de uma raiz unitária para a série transformada apresentou os seguintes resultados:

Estatística do teste ADF para série transformada	valores críticos 1%	-3,43439	
	-28,5146	valores críticos 5%	-2,86321
		valores críticos 10%	-2,56771

Tabela 6.20 Teste ADF para série NDLIUSA

Na Figura 6.41 é apresentada a estatística de Jarque-Bera, que rejeita fortemente a hipótese de distribuição normal.

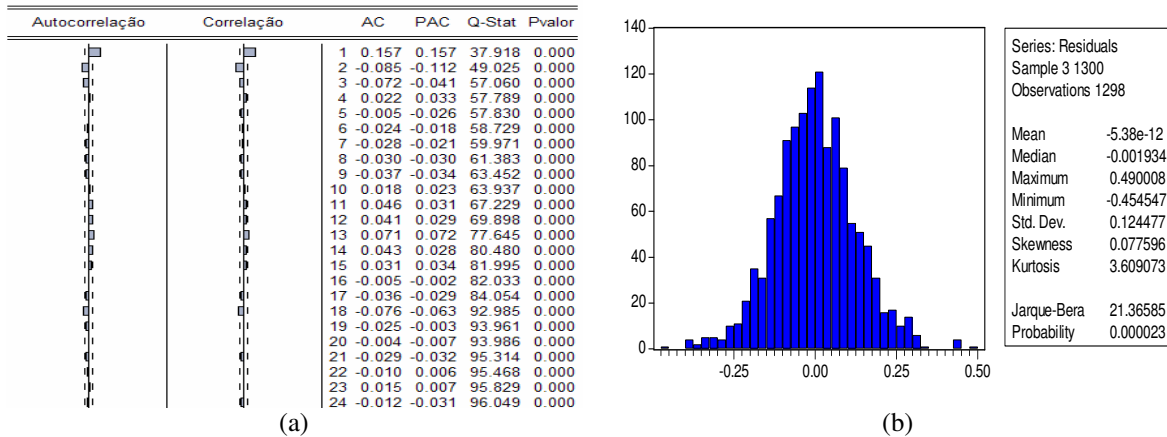


Figura 6.41 (a) Correlograma da série NDLIUSA, (b) Estatísticas dos resíduos

Os resultados para o teste BDS são apresentados na Tabela 6.21.

BDS				
Dimensão	Estatística	Erro-padrão	Estatística-z	p-valor
2	0,012195	0,001847	6,602719	0,0000
3	0,021300	0,002928	7,274068	0,0000
4	0,025387	0,003478	7,299345	0,0000
5	0,028330	0,003615	7,836109	0,0000
6	0,028372	0,003477	8,159878	0,0000

Tabela 6.21 Resultados obtidos pelo teste BDS nas respectivas dimensões.

O melhor modelo ajustado para a série é um AR(2)-GARCH(1,1). Os parâmetros estimados do modelo para a série são mostrados na Tabela 6.22.

Parâmetros	Coefficiente	Erro-padrão	Estatística-z	p-valor
ϕ_0	0,482154	0,003565	135,2536	0,0000
ϕ_1	0,170989	0,028759	5,945617	0,0000
ϕ_2	-0,097106	0,027453	-3,537166	0,0004
α_0	0,000989	0,000354	2,792137	0,0052
α_1	0,080108	0,019176	4,177485	0,0000
β_1	0,854186	0,035560	24,02119	0,0000

Tabela 6.22 Parâmetros estimados para a série NDLIUSA

O resultado do teste Ljung-Box para a série NDLIUSA é apresentado na Figura 6.42.

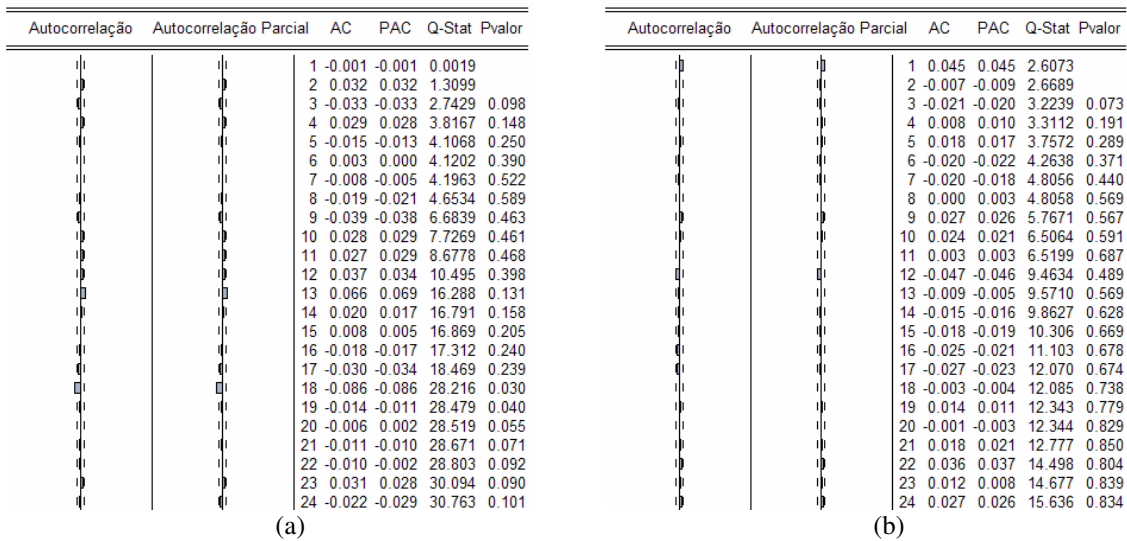


Figura 6.42 (a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado

As previsões realizadas utilizando o modelo ARIMA-GARCH são mostradas na Figura 6.43. A linha contínua indica o valor da série NDLIUSA e a linha central com marcador quadrado indica as previsões.

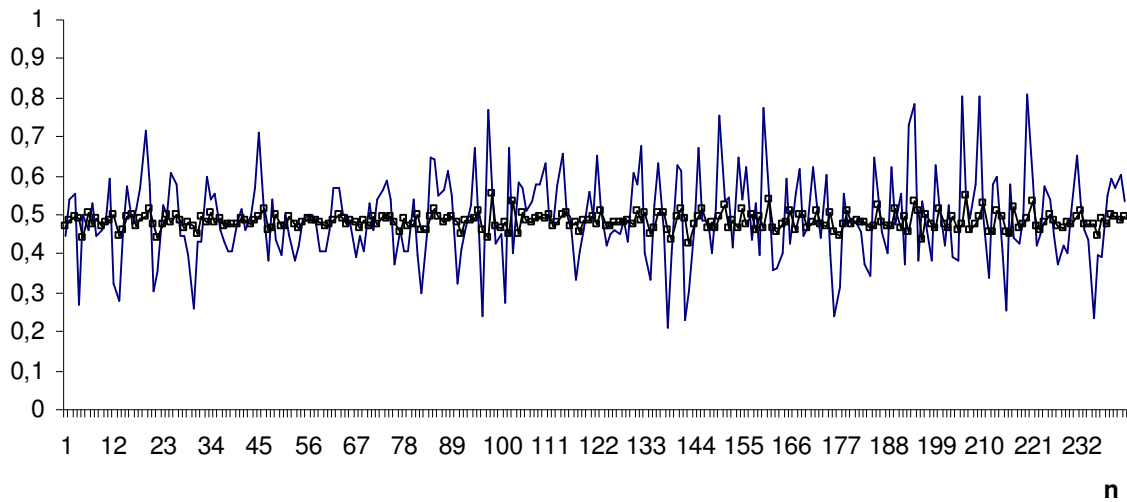


Figura 6.43 Previsão obtida pelo modelo ARIMA-GARCH para NDLIUSA

Na Figura 6.44a é mostrada a previsão da volatilidade e na Figura 6.44b a série original e o intervalo de previsão para a série NDLIUSA.

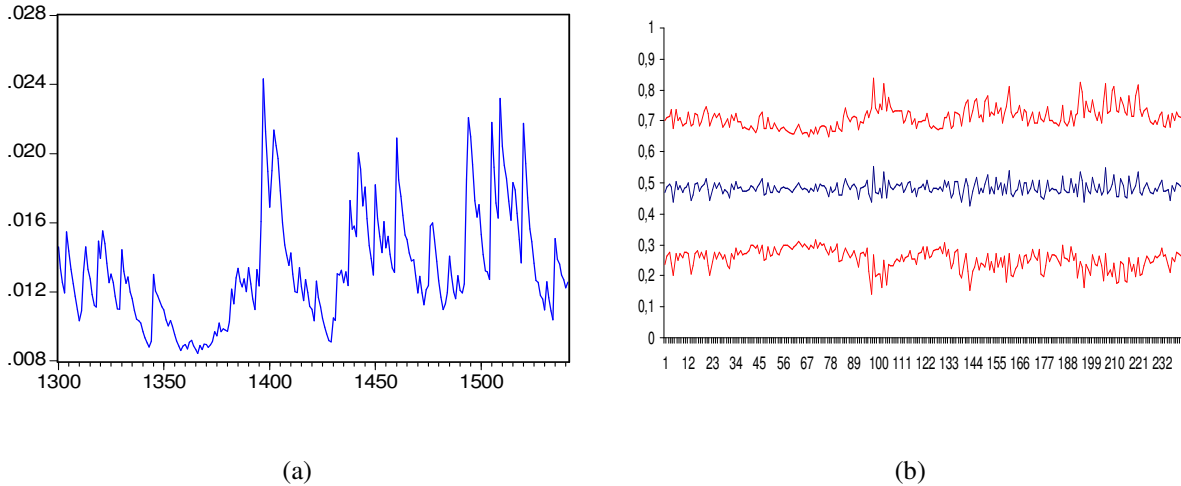


Figura 6.44 (a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLIUSA

6.2.4.2 Análise da série da Itausa – Método RNA *feedforward*

Para a série NDLIUSA os resultados obtidos pelas diferentes topologias de RNA são apresentados na Tabela 6.23.

Topologia	FIC	FIP	TIC	RMSE
RNA 10-1-1	140	0	0,119284	0,117562
RNA 10-2-1	131	0	0,110965	0,109702
RNA 10-3-1	95	0	0,112349	0,110828
RNA 10-4-1	127	1	0,129916	0,12821
RNA 10-5-1	119	0	0,112218	0,110692
RNA 10-8-1	116	0	0,122006	0,120261

Tabela 6.23 Estatísticas para diferentes topologias de RNA para a série

Na Figura 6.45 são apresentadas as previsões obtidas pela RNA 10-2-1 para a série NDLIUSA.

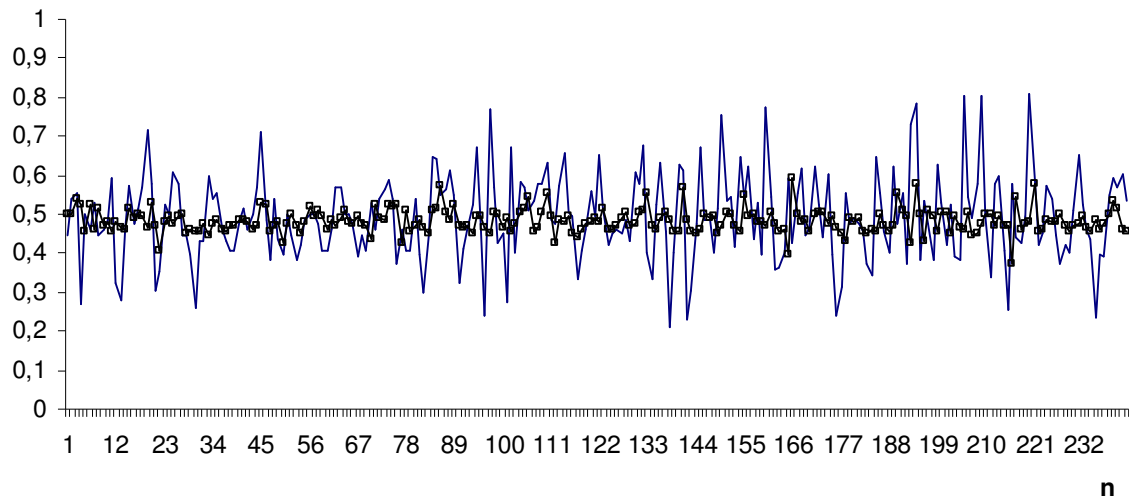
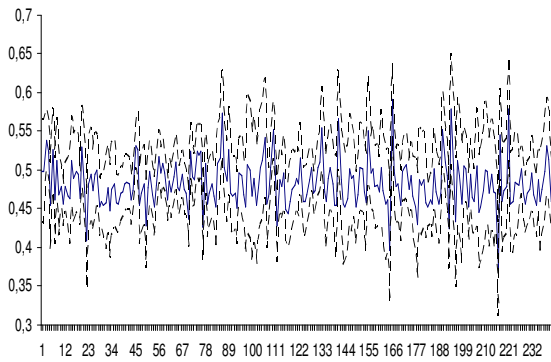
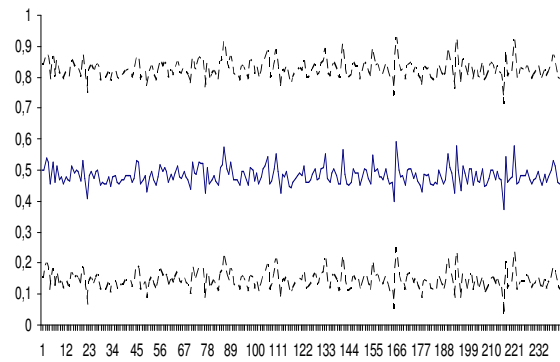


Figura 6.45 Previsão obtida pela RNA *feedforward* para NDLIUSA

Na Figura 6.46a é mostrado o intervalo de confiança para as previsões obtidas pela RNA 10-2-1 e na Figura 6.46b o intervalo de predição para a série NDLIUSA.



(a)



(b)

Figura 6.46 (a) Intervalo de confiança, (b) Intervalo de predição para NDLIUSA

Na Figura 6.47a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA 10-2-1 e na Figura 6.47b o histograma desses resíduos.

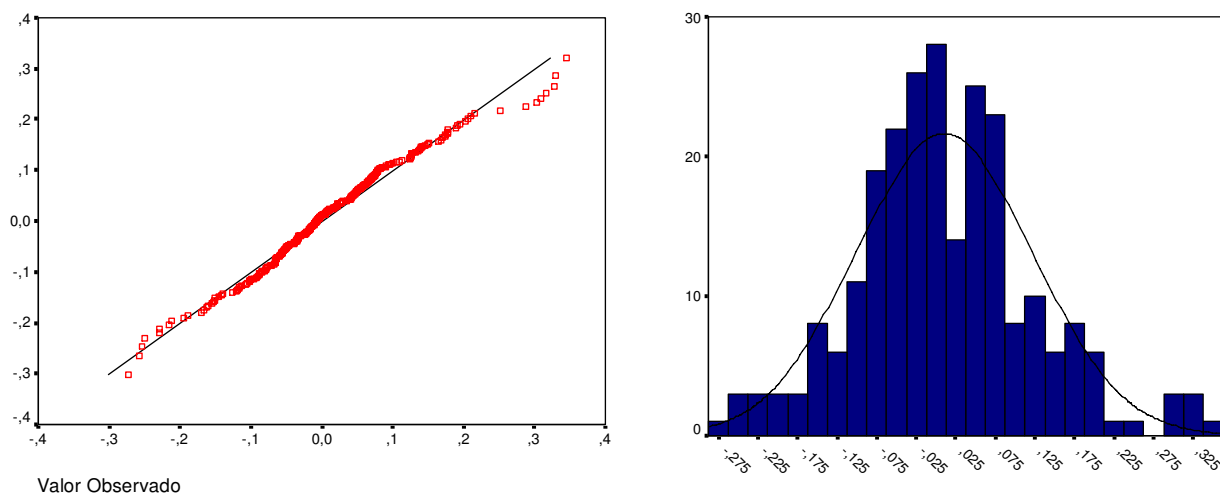


Figura 6.47 (a) Normal Q-Q Plot. (b) Histograma dos resíduos da previsão da RNA *feedforward* para NDLBIUSA

6.2.4.3 Análise da série da Itausa – Método RNA EKF

As previsões, para a série NDLIUSA, obtidas pela RNA treinada com o filtro de Kalman estendido (EKF) são mostradas na Figura 6.48.

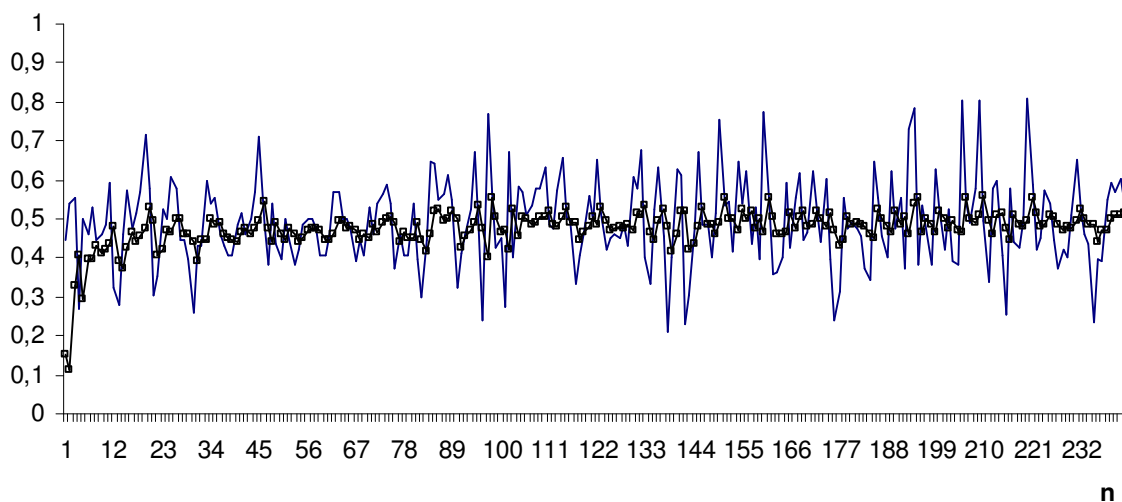


Figura 6.48 Previsão obtida pela RNA EKF para NDLIUSA

Na Figura 6.49a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman estendido e na Figura 6.49b o histograma desses resíduos.

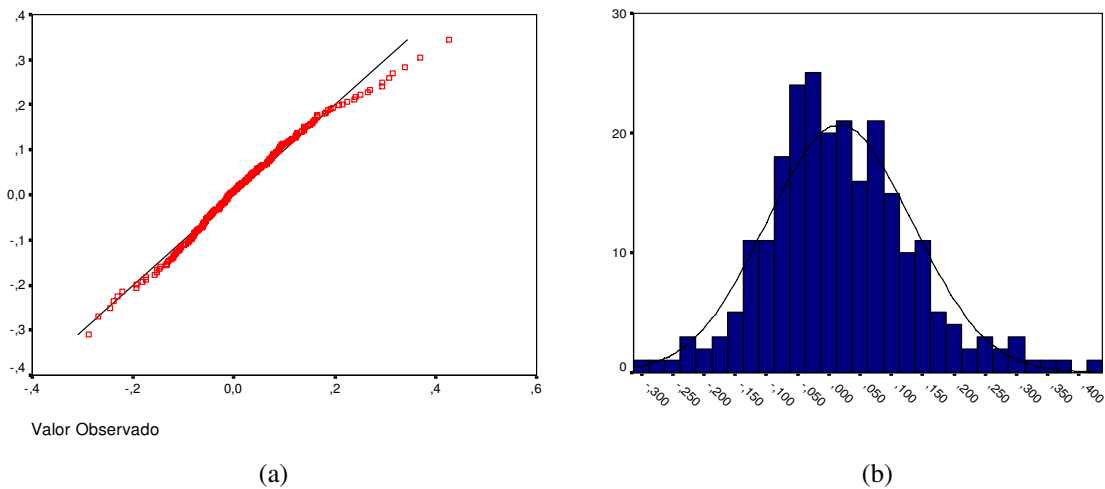


Figura 6.49 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLIUSA

6.2.4.4 Análise da série da Itausa – Método RNA UKF

As previsões obtidas pela RNA treinada com o filtro de Kalman unscented são mostradas na Figura 6.50.

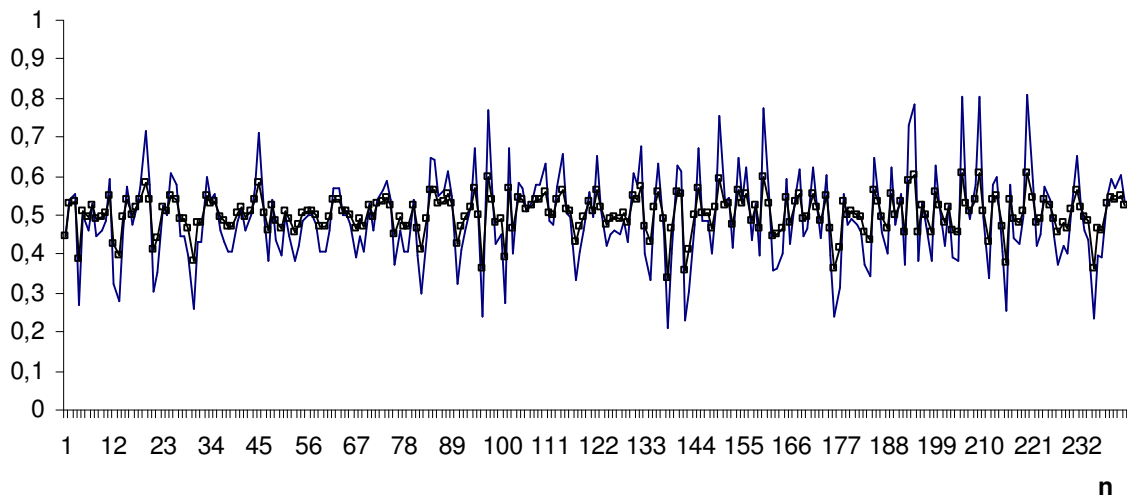


Figura 6.50 Previsão obtida pela RNA UKF para NDLIUSA

Na Figura 6.51a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman *unscented* e na Figura 6.51b o histograma desses resíduos.

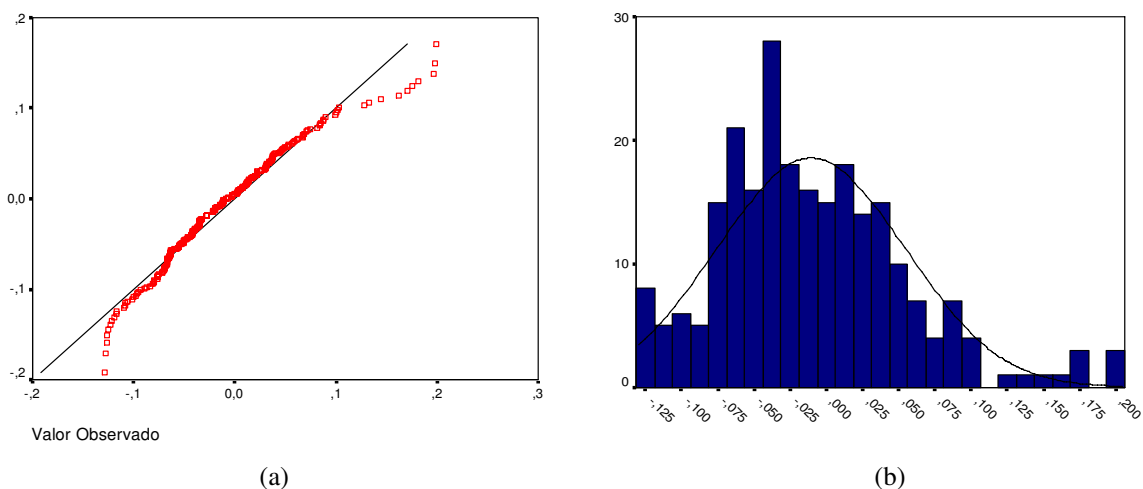


Figura 6.51 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLIUSA

As estatísticas de previsão obtidas pelos modelos ARIMA-GARCH, RNA *feedforward*, RNA EKF e RNA UKF são apresentadas na Tabela 6.24 e na Figura 6.52.

Método	TIC	RMSE
AR(2)-GARCH(1,1)	0,109252	0,107997
RNA 10-2-1	0,110965	0,109702
RNA EKF	0,119568	0,117599
RNA UKF	0,064831	0,065538

Tabela 6.24 Estatísticas de previsão para o método de ARIMA-GARCH e de RNA

Pode-se verificar que as melhores previsões foram obtidas pelo método que empregou redes neurais treinadas com o filtro de Kalman *unscented*.

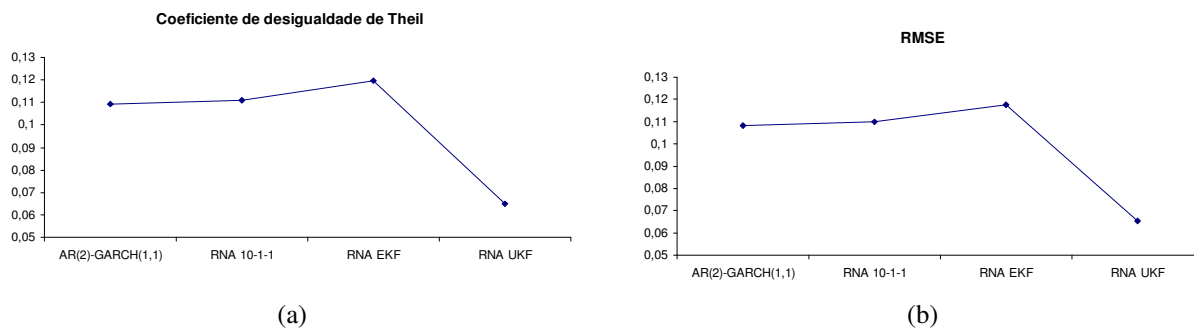


Figura 6.52 Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA

6.3 Setor de Alimentos – Perdigão, Sadia, Soja e Açúcar

No caso do setor de alimentos as séries analisadas seguirão a seqüência: Perdigão, Sadia, Soja e Açúcar.

6.3.1 Análise da série da Perdigão

A seguir são apresentados os resultados obtidos para a série da Perdigão.

6.3.1.1 Análise da série da Perdigão – Método ARIMA-GARCH

Na Figura 6.53a é apresentada a série original do preço de fechamento da Perdigão, na Figura 6.53b é apresentada a série normalizada da primeira diferença do logaritmo do preço, NDLPERD.

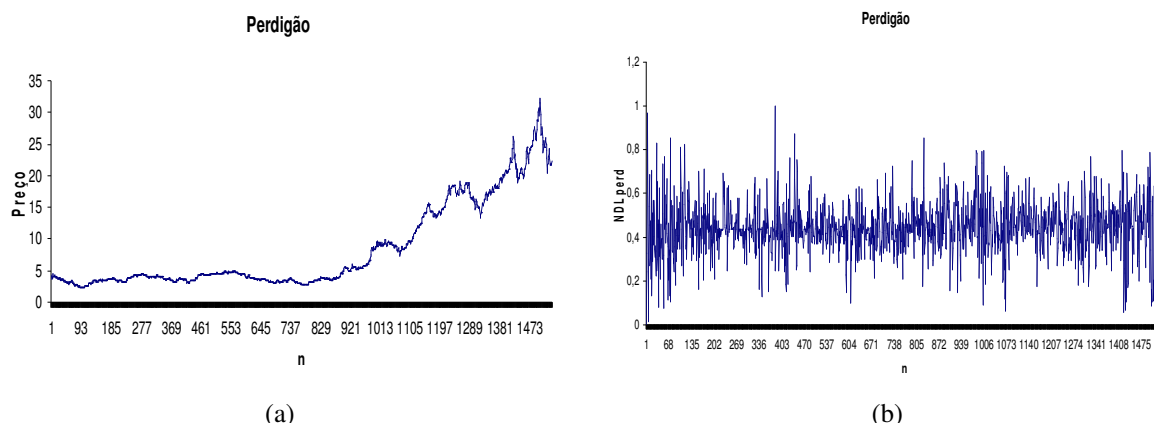


Figura 6.53 (a) Série no nível do preço do Bradesco, (b) Série transformada NDLPERD

O resultado do teste ADF, para a série original, é apresentado na Tabela 6.25.

Estatística do teste ADF para série original	valores críticos 1%	-3,43439
0,23910	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771

Tabela 6.25 Teste ADF para série original da Perdigão

Não é rejeitada a hipótese de raiz unitária para a série no nível pelo teste ADF. O teste de presença de uma raiz unitária para a série transformada apresentou os seguintes resultados:

Estatística do teste ADF para série transformada	valores críticos 1%	-3,43439
-33,1227	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771

Tabela 6.26 Teste ADF para série NDLPERD

Na Figura 6.54 é apresentada a estatística de Jarque-Bera, que rejeita fortemente a hipótese de distribuição normal.

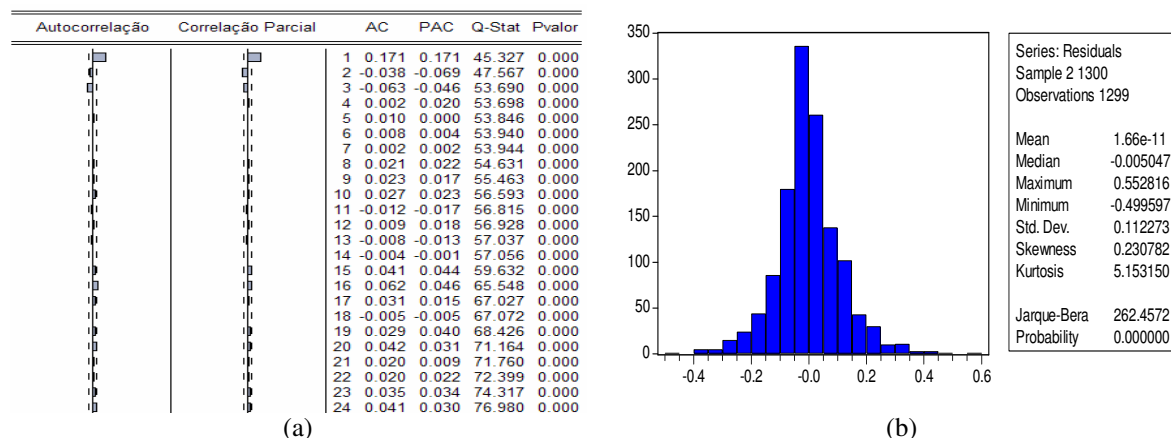


Figura 6.54 (a) Correlograma da série NDLPED, (b) Estatísticas dos resíduos

Os resultados para o teste BDS são apresentados na Tabela 6.27.

BDS				
Dimensão	Estatística	Erro-padrão	Estatística-z	p-valor
2	0,025258	0,002271	11,11986	0,0000
3	0,045749	0,003606	12,68843	0,0000
4	0,058545	0,004289	13,65097	0,0000
5	0,064846	0,004465	14,52239	0,0000
6	0,065240	0,004302	15,16668	0,0000

Tabela 6.27 Resultados obtidos pelo teste BDS nas respectivas dimensões.

O melhor modelo ajustado para a série é um AR(1)-GARCH(1,1). Os parâmetros estimados do modelo para a série são mostrados na Tabela 6.28.

Parâmetros	Coefficiente	Erro-padrão	Estatística-z	p-valor
ϕ_0	0,444602	0,003247	136,9117	0,0000
ϕ_1	0,142823	0,026878	5,313849	0,0000
α_0	0,001384	0,000315	4,390602	0,0000
α_1	0,117129	0,019798	5,916306	0,0000
β_1	0,770346	0,041048	18,76713	0,0000

Tabela 6.28 Parâmetros estimados para a série NDLPED

O resultado do teste Ljung-Box para a série NDLPED é apresentado na Figura 6.55.

Autocorrelação	Autocorrelação Parcial	AC	PAC	Q-Stat	Pvalor
		1 0.006	0.006	0.0509	
		2 0.016	0.016	0.3957	0.529
		3 -0.030	-0.030	1.5543	0.460
		4 0.008	0.008	1.6376	0.651
		5 0.002	0.003	1.6445	0.801
		6 0.045	0.044	4.3304	0.503
		7 -0.001	-0.001	4.3325	0.632
		8 0.032	0.031	5.6957	0.576
		9 0.029	0.032	6.8357	0.554
		10 0.024	0.022	7.5815	0.577
		11 -0.019	-0.018	8.0414	0.625
		12 -0.007	-0.009	8.1139	0.703
		13 -0.001	0.001	8.1147	0.776
		14 -0.006	-0.010	8.1594	0.833
		15 0.035	0.032	9.7413	0.781
		16 0.046	0.044	12.586	0.634
		17 -0.006	-0.008	12.633	0.699
		18 0.011	0.010	12.796	0.750
		19 0.033	0.036	14.245	0.713
		20 0.031	0.031	15.527	0.689
		21 0.011	0.009	15.688	0.736
		22 0.023	0.021	16.385	0.748
		23 0.028	0.029	17.460	0.737
		24 0.057	0.052	21.792	0.533

Autocorrelação	Autocorrelação Parcial	AC	PAC	Q-Stat	Pvalor
		1 0.032	0.032	1.3474	
		2 -0.009	-0.010	1.4439	0.230
		3 -0.017	-0.017	1.8272	0.401
		4 -0.015	-0.014	2.1304	0.546
		5 -0.024	-0.023	2.8876	0.577
		6 0.010	0.011	3.0244	0.696
		7 -0.005	-0.007	3.0582	0.802
		8 0.010	0.010	3.1933	0.867
		9 -0.013	-0.014	3.4143	0.906
		10 -0.010	-0.009	3.5431	0.939
		11 0.031	0.032	4.7902	0.905
		12 -0.014	-0.017	5.0565	0.928
		13 0.003	0.004	5.0663	0.956
		14 -0.014	-0.015	5.3355	0.967
		15 0.010	0.011	5.4664	0.978
		16 0.021	0.021	6.0502	0.979
		17 -0.009	-0.012	6.1530	0.986
		18 -0.045	-0.044	8.8570	0.945
		19 -0.015	-0.013	9.1412	0.956
		20 0.044	0.047	11.702	0.898
		21 0.013	0.009	11.936	0.918
		22 0.020	0.017	12.478	0.926
		23 -0.007	-0.008	12.540	0.945
		24 0.036	0.038	14.298	0.918

Figura 6.55 (a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado

As previsões realizadas utilizando o modelo ARIMA-GARCH são mostradas na Figura 6.56. A linha contínua indica o valor da série NDLPERD e a linha central com marcador quadrado indica as previsões.

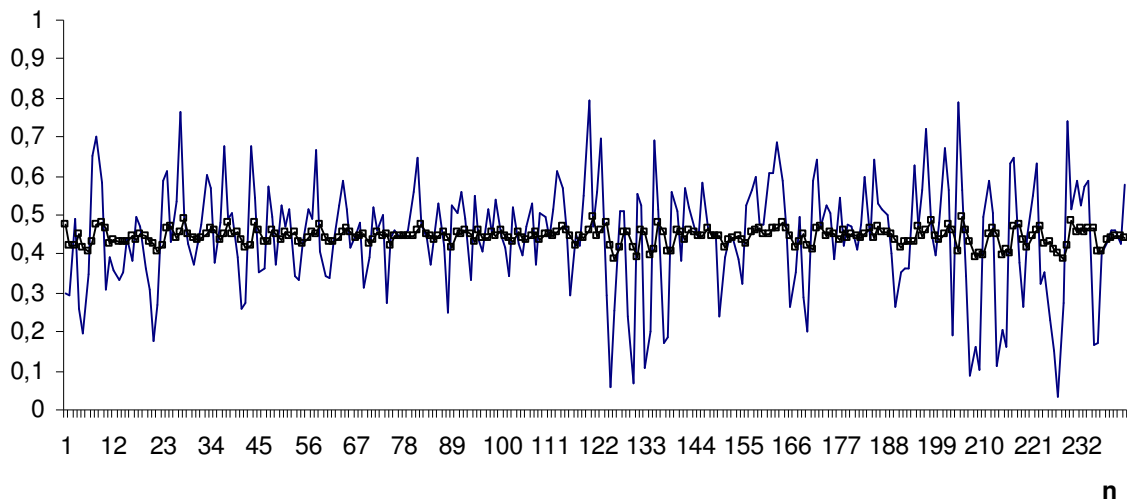
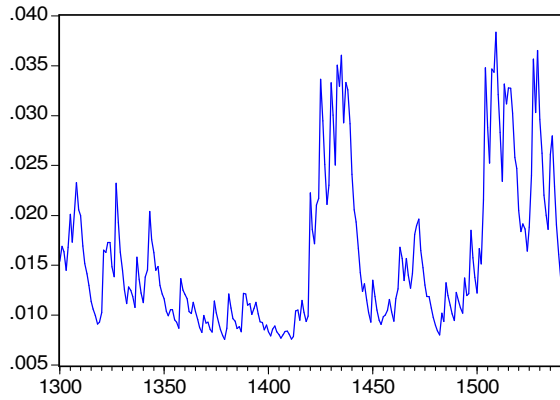
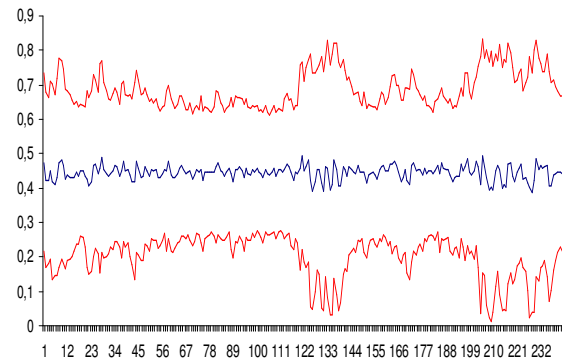


Figura 6.56 Previsão obtida pelo modelo ARIMA-GARCH para NDLPERD

Na Figura 6.57a é mostrada a previsão da volatilidade e na Figura 6.57b a série original e o intervalo de previsão para a série NDLPERD.



(a)



(b)

Figura 6.57 (a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLPERD

6.3.1.2 Análise da série da Perdigoão – Método RNA *feedforward*

Para a série NDLPERD os resultados obtidos pelas diferentes topologias de RNA são apresentados na Tabela 6.29.

Topologia	FIC	FIP	TIC	RMSE
RNA 10-1-1	122	0	0,152813	0,140041
RNA 10-2-1	145	0	0,171551	0,156073
RNA 10-3-1	136	0	0,170983	0,157927
RNA 10-4-1	143	0	0,157963	0,143877
RNA 10-5-1	127	0	0,162571	0,147175
RNA 10-8-1	88	0	0,203454	0,187000

Tabela 6.29 Estatísticas para diferentes topologias de RNA para a série

Na Figura 6.58 são apresentadas as previsões obtidas pela RNA 10-1-1 para a série NDLPERD.

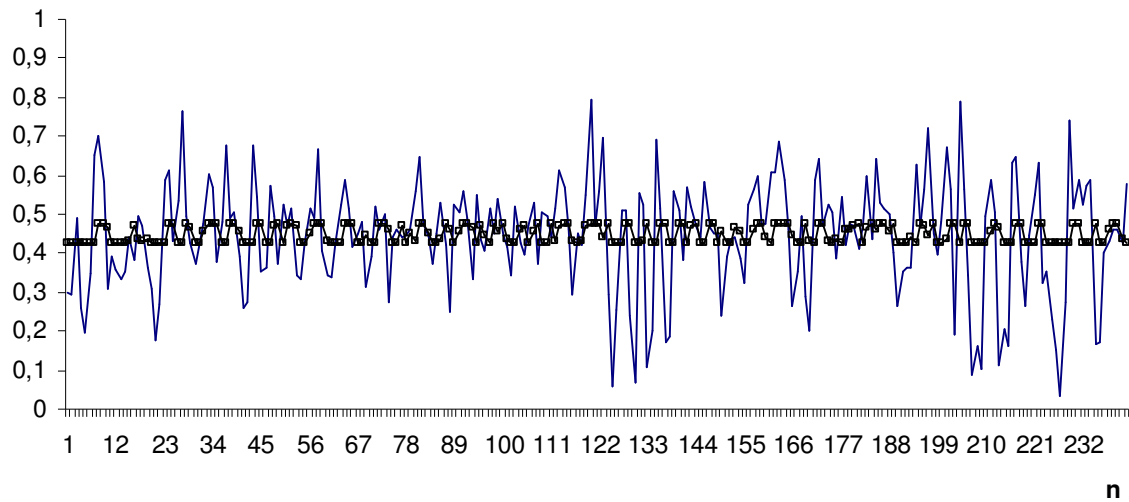


Figura 6.58 Previsão obtida pela RNA *feedforward* para NDLPERD

Na Figura 6.59a é mostrado o intervalo de confiança para as previsões obtidas pela RNA 10-1-1 e na Figura 6.59b o intervalo de predição para a série NDLPERD.

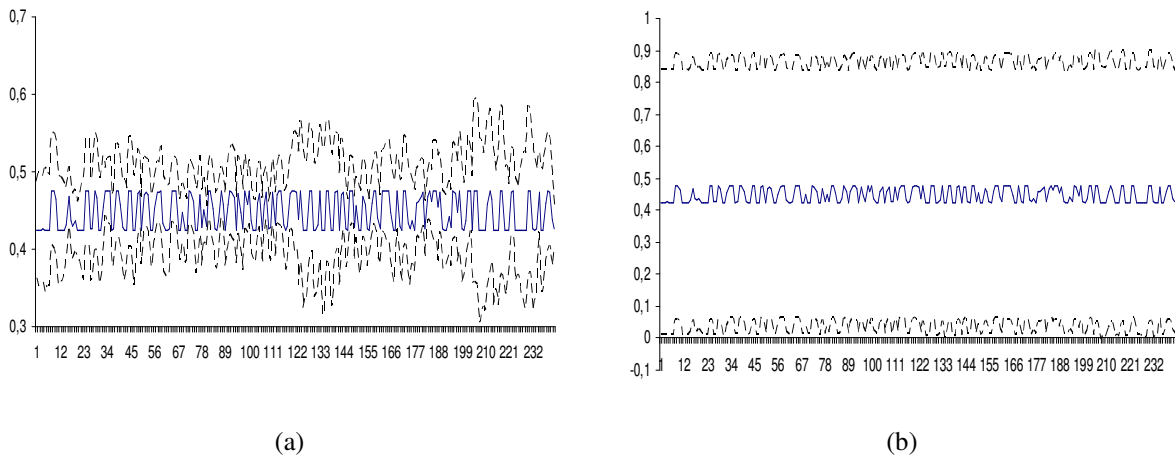


Figura 6.59 (a) Intervalo de confiança, (b) Intervalo de predição para NDLPERD

Na Figura 6.60a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA 10-1-1 e na Figura 6.60b o histograma desses resíduos.

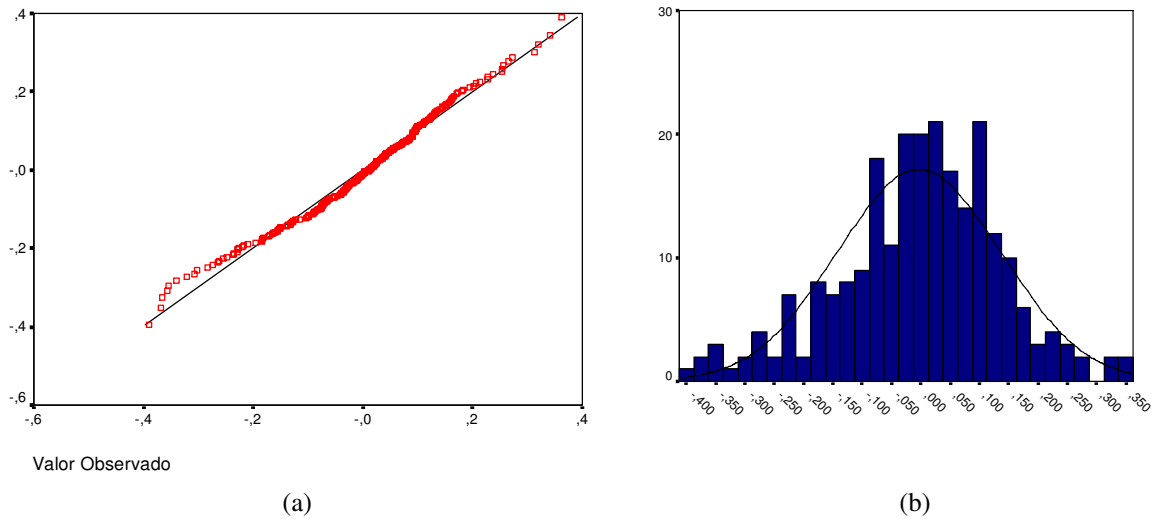


Figura 6.60 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA *feedforward* para NDLPERD

6.3.1.3 Análise da série da Perdigoão – Método RNA EKF

As previsões, para a série NDLPERD, obtidas pela RNA treinada com o filtro de Kalman estendido (EKF) são mostradas na Figura 6.61.

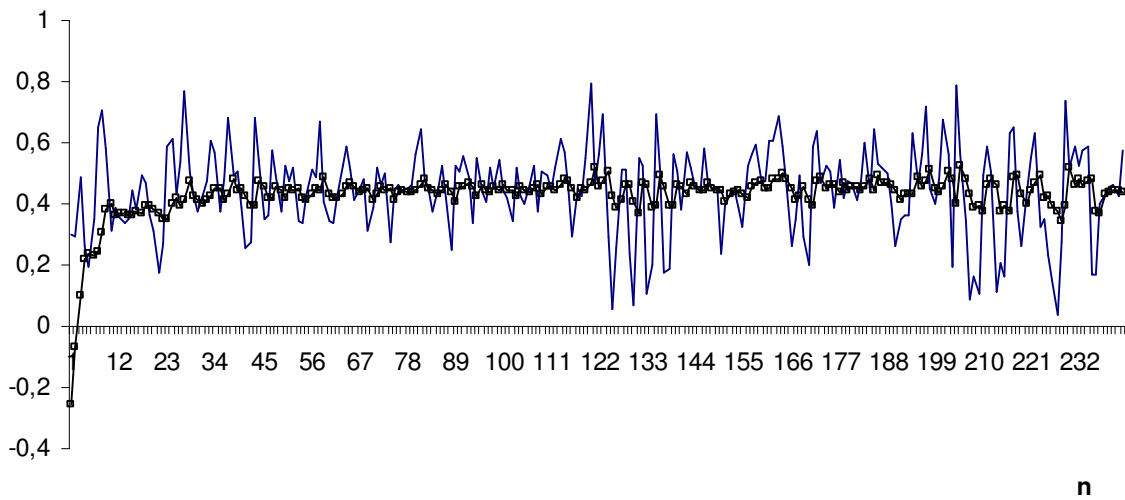


Figura 6.61 Previsão obtida pela RNA EKF para NDLPERD

Na Figura 6.62a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman estendido e na Figura 6.62b o histograma desses resíduos.

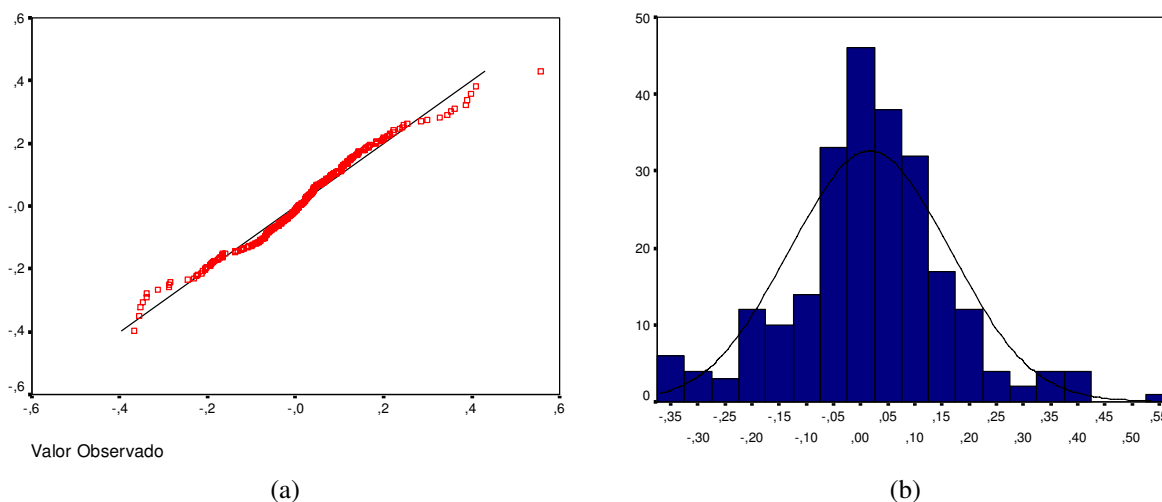


Figura 6.62 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLPERD

6.3.1.4 Análise da série da Perdigoão – Método RNA UKF

As previsões obtidas pela RNA treinada com o filtro de Kalman *unscented* são mostradas na Figura 6.63.

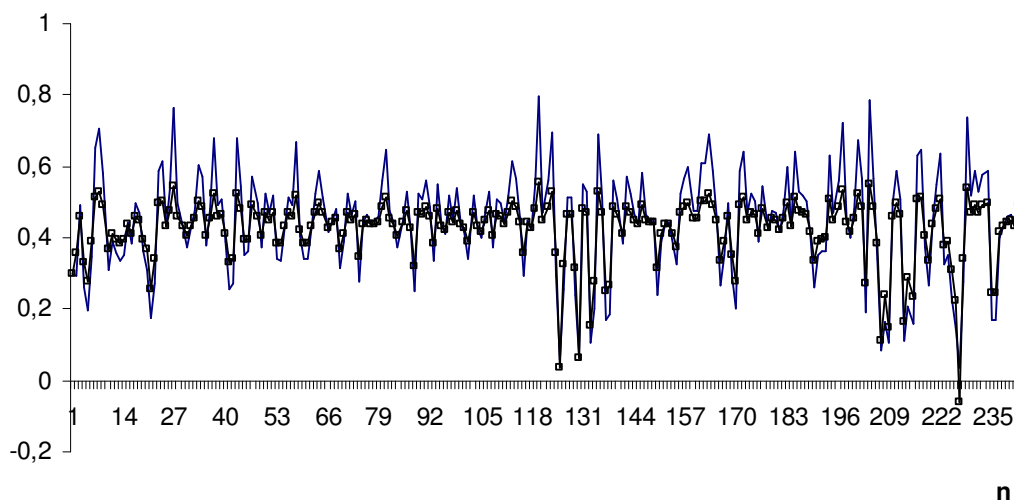


Figura 6.63 Previsão obtida pela RNA UKF para NDLPERD

Na Figura 6.64a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman *unscented* e na Figura 6.64b o histograma desses resíduos.

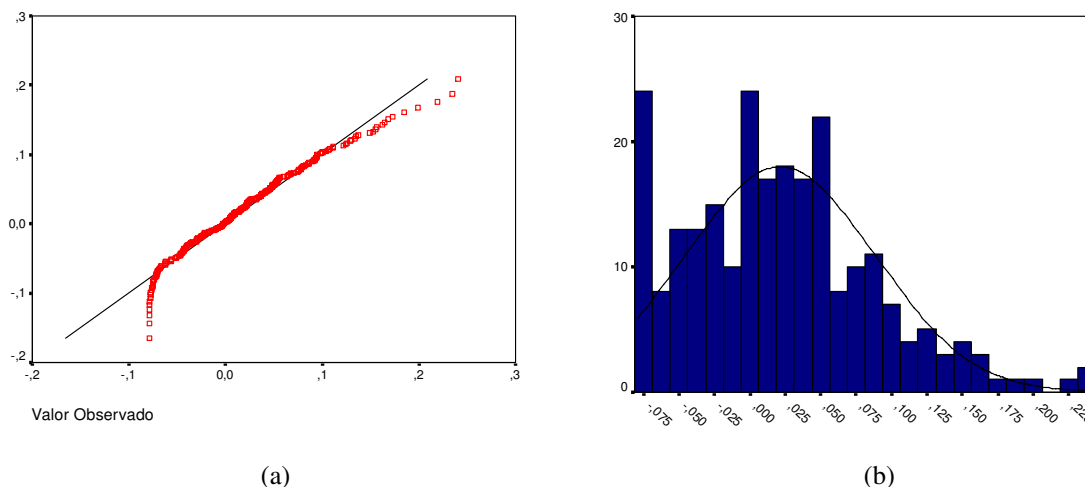


Figura 6.64 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLPED

As estatísticas de previsão obtidas pelos modelos ARIMA-GARCH, RNA *feedforward*, RNA EKF e RNA UKF são apresentadas na Tabela 6.30 e na Figura 6.65.

Método	TIC	RMSE
AR(1)-GARCH(1,1)	0,150648	0,137459
RNA 10-1-1	0,152813	0,140041
RNA EKF	0,164558	0,148415
RNA UKF	0,078030	0,070216

Tabela 6.30 Estatísticas de previsão para o método de ARIMA-GARCH e de RNA

Pode-se verificar que as melhores previsões foram obtidas pelo método que empregou redes neurais treinadas com o filtro de Kalman *unscented*.

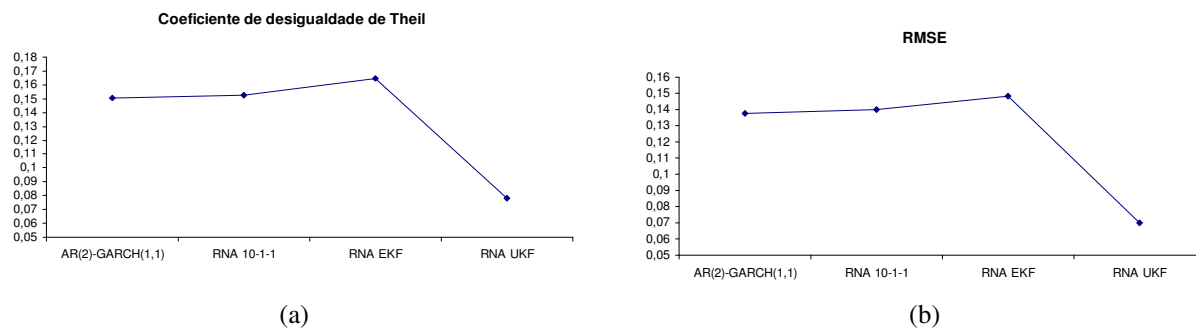


Figura 6.65 Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA

6.3.2 Análise da série da Sadia

A seguir são apresentados os resultados obtidos para a série da Sadia.

6.3.2.1 Análise da série da Sadia – Método ARIMA-GARCH

Na Figura 6.66a é apresentada a série original do preço de fechamento da Sadia, na Figura 6.66b é apresentada a série normalizada da primeira diferença do logaritmo do preço, NDLSADI.

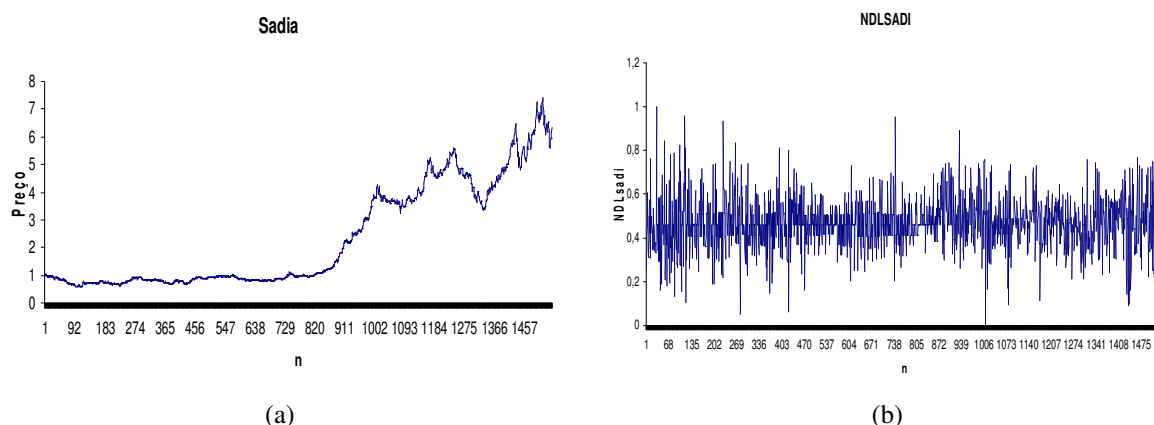


Figura 6.66 (a) Série no nível do preço do Bradesco, (b) Série transformada NDLSADI

O resultado do teste ADF, para a série original, é apresentado na Tabela 6.31.

Estatística do teste ADF para série original	0,60901	valores críticos 1%	-3,43439
		valores críticos 5%	-2,86321
		valores críticos 10%	-2,56771

Tabela 6.31 Teste ADF para série original da Sadia

Não é rejeitada a hipótese de raiz unitária para a série no nível pelo teste ADF. O teste de presença de uma raiz unitária para a série transformada apresentou os seguintes resultados:

Estatística do teste ADF para série transformada	-28,6619	valores críticos 1%	-3,43439
		valores críticos 5%	-2,86321
		valores críticos 10%	-2,56771

Tabela 6.32 Teste ADF para série NDLSADI

Na Figura 6.67 é apresentada a estatística de Jarque-Bera, que rejeita fortemente a hipótese de distribuição normal.

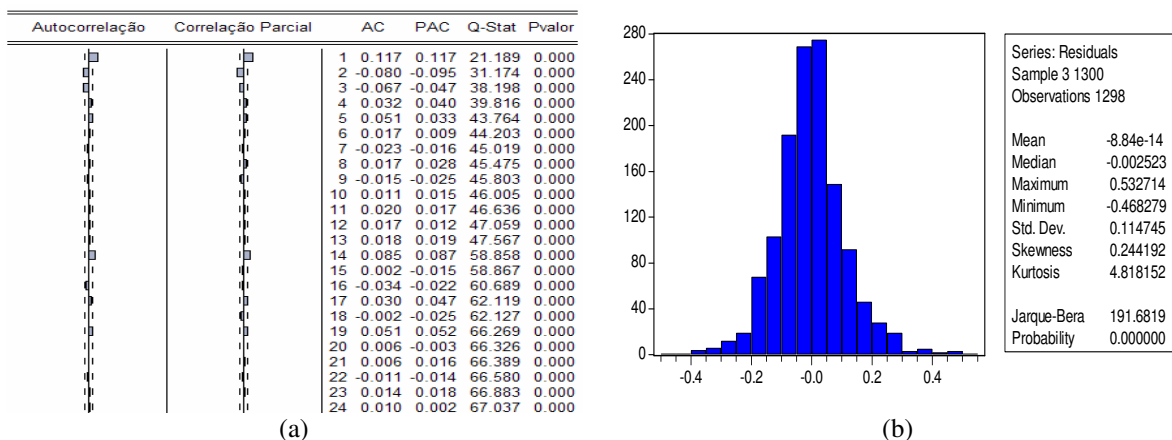


Figura 6.67 (a) Correlograma da série NDLSADI, (b) Estatísticas dos resíduos

Os resultados para o teste BDS são apresentados na Tabela 6.33.

BDS				
Dimensão	Estatística	Erro-padrão	Estatística-z	p-valor
2	0,018003	0,002128	8,458460	0,0000
3	0,031549	0,003380	9,335218	0,0000
4	0,042207	0,004021	10,49710	0,0000
5	0,046904	0,004187	11,20235	0,0000
6	0,047280	0,004034	11,72010	0,0000

Tabela 6.33 Resultados obtidos pelo teste BDS nas respectivas dimensões.

O melhor modelo ajustado para a série é um AR(2)-GARCH(1,1). Os parâmetros estimados do modelo para a série são mostrados na Tabela 6.34.

Parâmetros	Coefficiente	Erro-padrão	Estatística-z	p-valor
ϕ_0	0,467754	0,002857	163,7015	0,0000
ϕ_1	0,095151	0,027697	3,435433	0,0006
ϕ_2	-0,073470	0,028653	-2,564098	0,0103
α_0	0,000562	0,000153	3,682027	0,0002
α_1	0,104281	0,015441	6,753738	0,0000
β_1	0,856652	0,021548	39,75636	0,0000

Tabela 6.34 Parâmetros estimados para a série NDLSADI

O resultado do teste Ljung-Box para a série NDLSADI é apresentado na Figura 6.68.

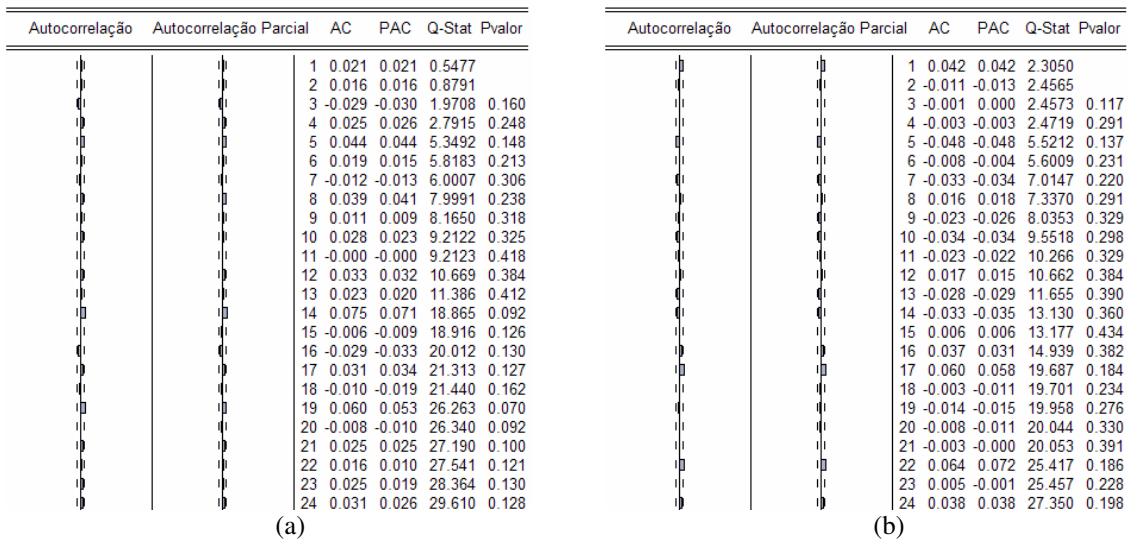


Figura 6.68 (a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado

As previsões realizadas utilizando o modelo ARIMA-GARCH são mostradas na Figura 6.69. A linha contínua indica o valor da série NDLSADI e a linha central com marcador quadrado indica as previsões.

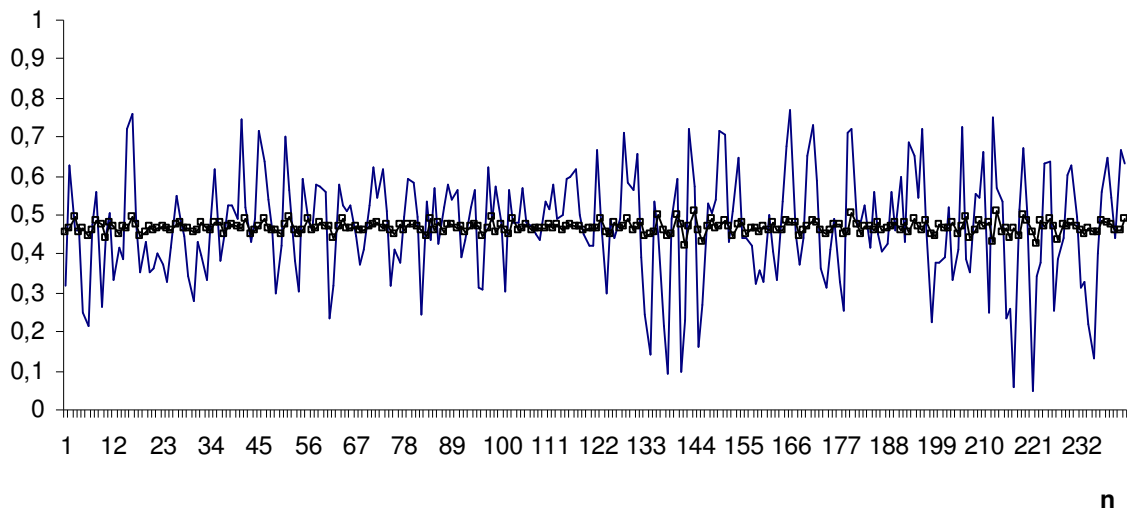


Figura 6.69 Previsão obtida pelo modelo ARIMA-GARCH para NDLSADI

Na Figura 6.70a é mostrada a previsão da volatilidade e na Figura 6.70b a série original e o intervalo de previsão para a série NDLSADI.

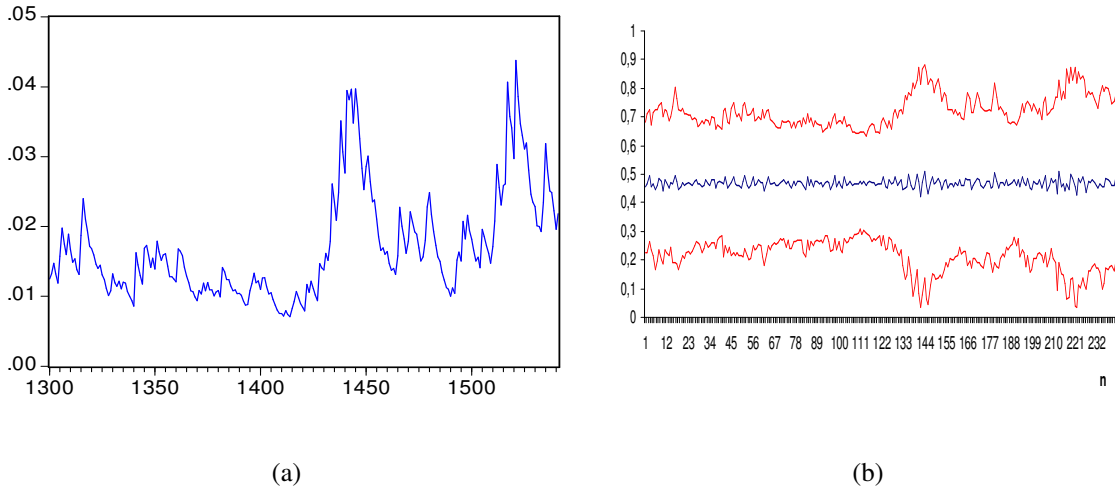


Figura 6.70 (a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLSADI

6.3.2.2 Análise da série da Sadia – Método RNA *feedforward*

Para a série NDLSADI os resultados obtidos pelas diferentes topologias de RNA são apresentados na Tabela 6.35.

Topologia	FIC	FIP	TIC	RMSE
RNA 10-1-1	133	0	0,144893	0,138782
RNA 10-2-1	146	0	0,149009	0,143148
RNA 10-3-1	146	0	0,152123	0,145382
RNA 10-4-1	94	0	0,154393	0,151044
RNA 10-5-1	109	1	0,170448	0,16436
RNA 10-8-1	79	1	0,195074	0,194734

Tabela 6.35 Estatísticas para diferentes topologias de RNA para a série

Na Figura 6.71 são apresentadas as previsões obtidas pela RNA 10-1-1 para a série NDLSADI.

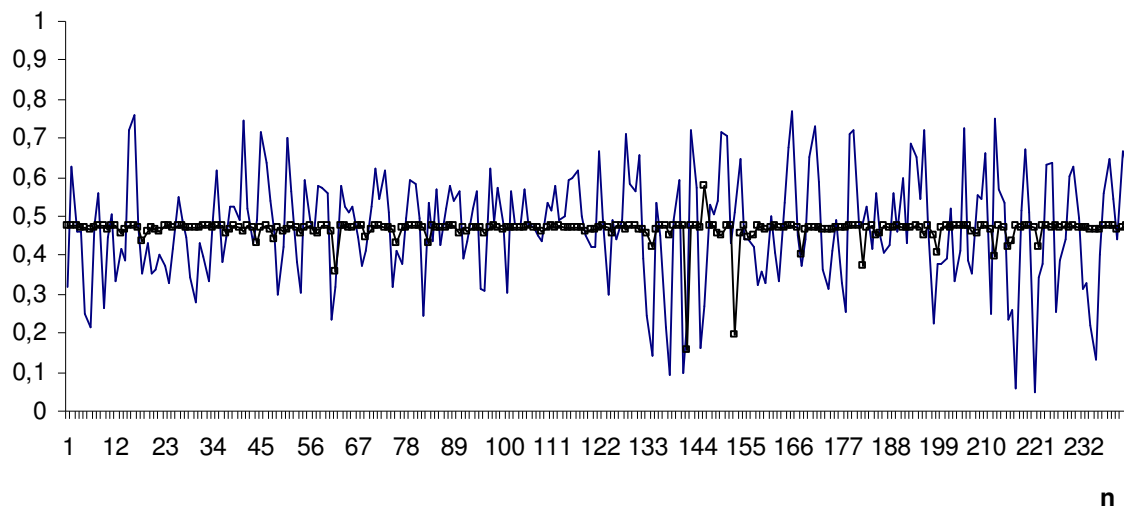


Figura 6.71 Previsão obtida pela RNA *feedforward* para NDLSADI

Na Figura 6.72a é mostrado o intervalo de confiança para as previsões obtidas pela RNA 10-1-1 e na Figura 6.72b o intervalo de predição para a série NDLSADI.

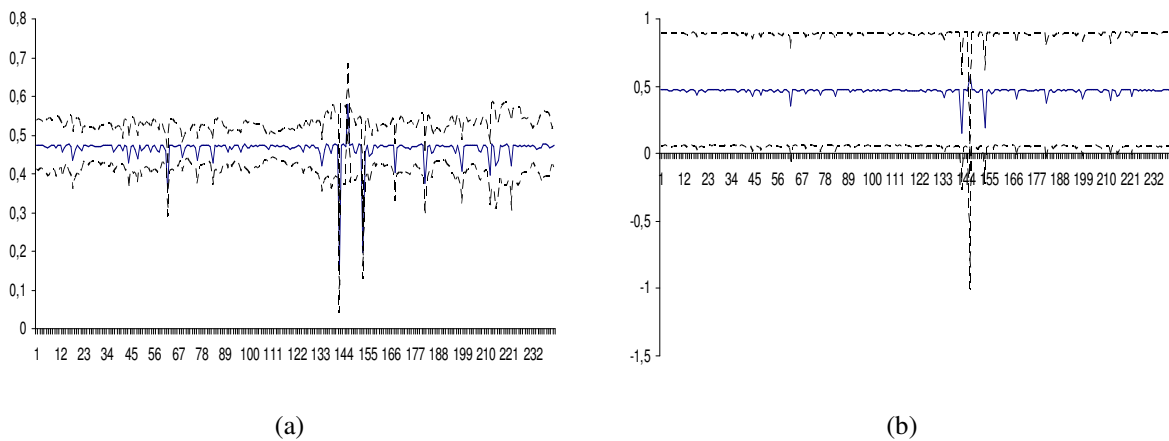
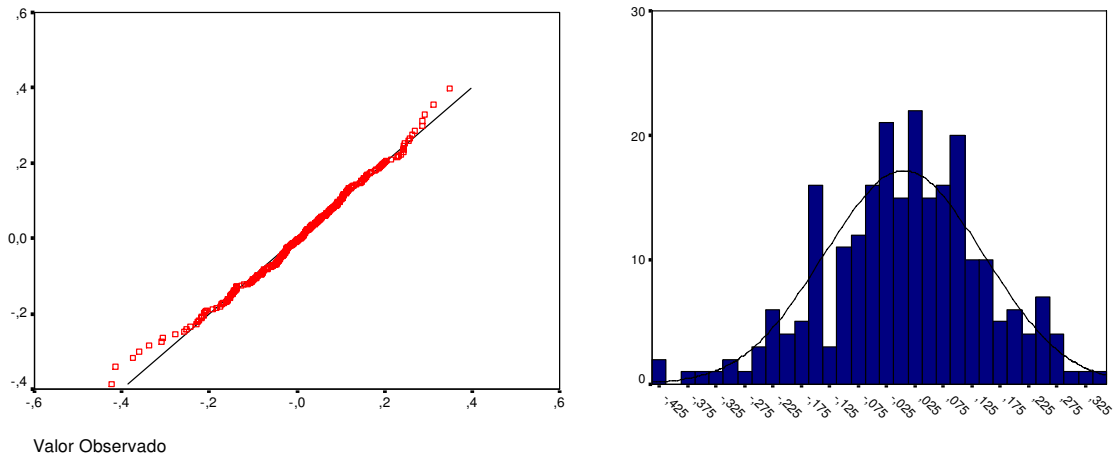


Figura 6.72 (a) Intervalo de confiança, (b) Intervalo de predição para NDLSADI

Na Figura 6.73a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA 10-1-1 e na Figura 6.73b o histograma desses resíduos.



(a) (b)
Figura 6.73 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA *feedforward* para NDLSADI

6.3.2.3 Análise da série da Sadia – Método RNA EKF

As previsões, para a série NDLSADI, obtidas pela RNA treinada com o filtro de Kalman estendido (EKF) são mostradas na Figura 6.74.

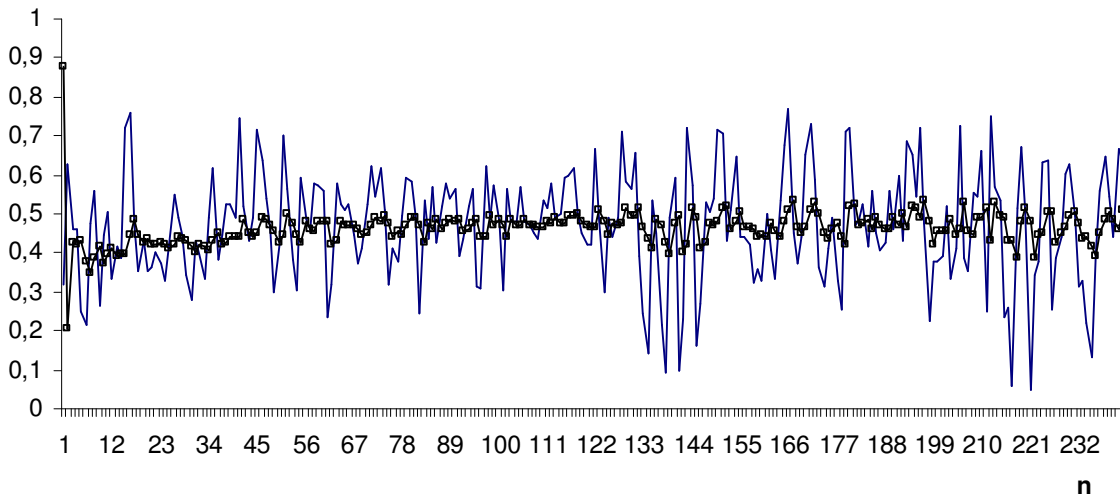
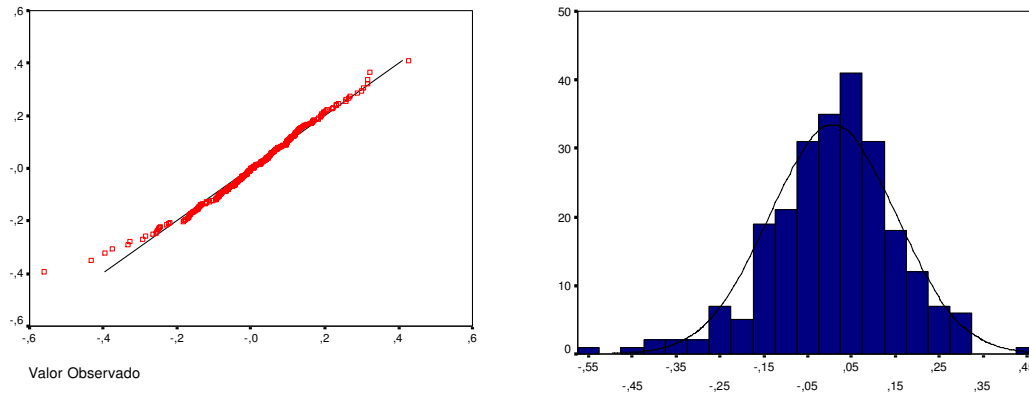


Figura 6.74 Previsão obtida pela RNA EKF para NDLSADI

Na Figura 6.75a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman estendido e na Figura 6.75b o histograma desses resíduos.



(a) (b)
Figura 6.75 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLSADI

6.3.2.4 Análise da série da Sadia – Método RNA UKF

As previsões obtidas pela RNA treinada com o filtro de Kalman *unscented* são mostradas na Figura 6.76.

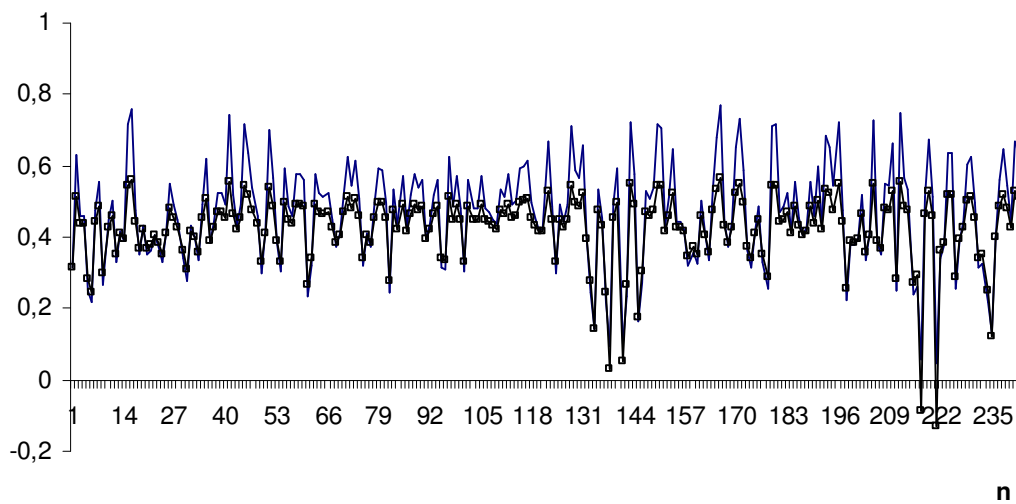


Figura 6.76 Previsão obtida pela RNA UKF para NDLSADI

Na Figura 6.77a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman *unscented* e na Figura 6.77b o histograma desses resíduos.

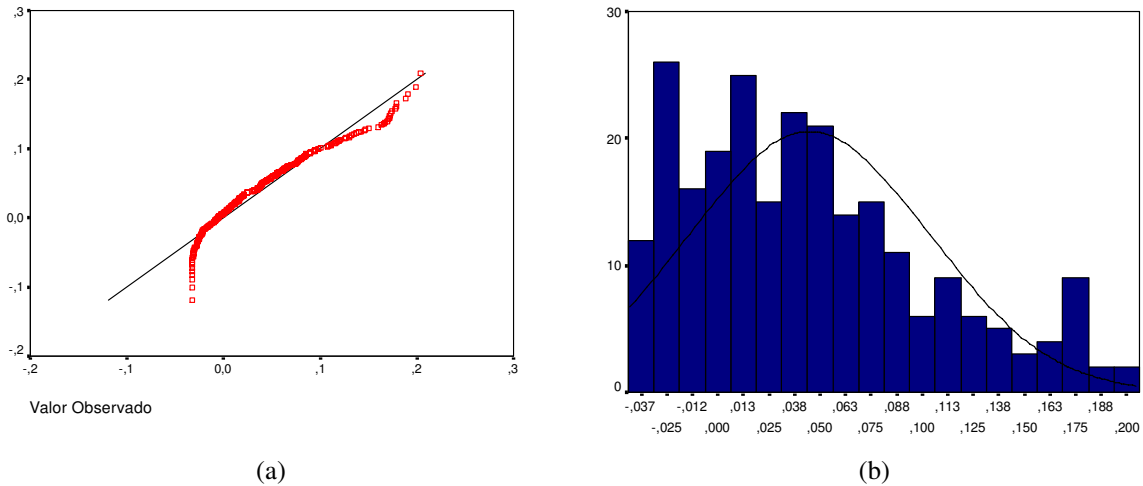


Figura 6.77 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLSADI

As estatísticas de previsão obtidas pelos modelos ARIMA-GARCH, RNA *feedforward*, RNA EKF e RNA UKF são apresentadas na Tabela 6.36 e na Figura 6.78.

Método	TIC	RMSE
AR(2)-GARCH(1,1)	0,141499	0,135700
RNA 10-1-1	0,144893	0,138782
RNA EKF	0,150599	0,143939
RNA UKF	0,079431	0,073696

Tabela 6.36 Estatísticas de previsão para o método de ARIMA-GARCH e de RNA

Pode-se verificar que as melhores previsões foram obtidas pelo método que empregou redes neurais treinadas com o filtro de Kalman *unscented*.

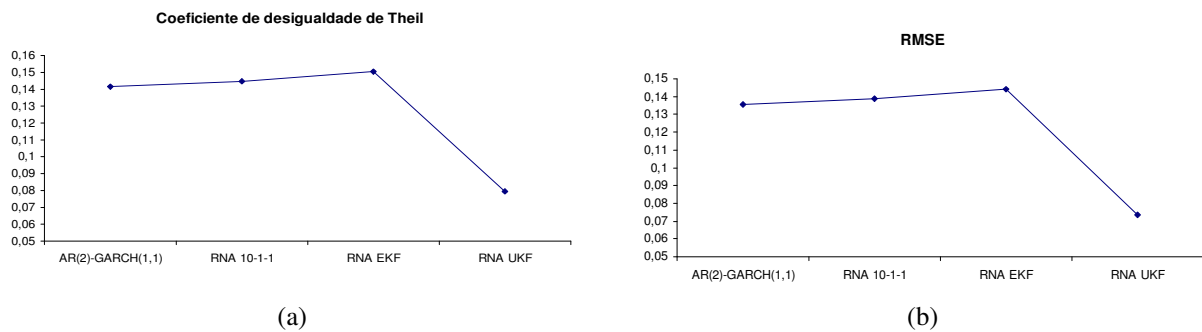


Figura 6.78 Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA

6.3.3 Análise da série da Soja

A seguir são apresentados os resultados obtidos para a série da Soja.

6.3.3.1 Análise da série da Soja – Método ARIMA-GARCH

Na Figura 6.79a é apresentada a série original do preço de fechamento da Soja, na Figura 6.79b é apresentada a série normalizada da primeira diferença do logaritmo do preço, NDLSOJA.

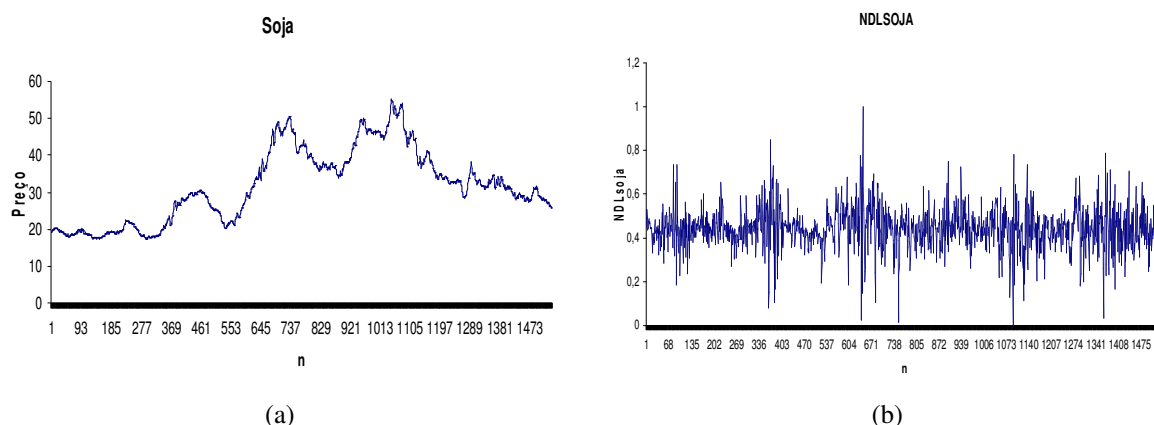


Figura 6.79 (a) Série no nível do preço do Bradesco, (b) Série transformada NDLSOJA

O resultado do teste ADF, para a série original, é apresentado na Tabela 6.37.

Estatística do teste ADF para série original		valores críticos 1%	-3,43439
	-1,48437	valores críticos 5%	-2,86321
		valores críticos 10%	-2,56771

Tabela 6.37 Teste ADF para série original da Soja

Não é rejeitada a hipótese de raiz unitária para a série no nível pelo teste ADF. O teste de presença de uma raiz unitária para a série transformada apresentou os seguintes resultados:

Estatística do teste ADF para série transformada		valores críticos 1%	-3,43439
	-29,2673	valores críticos 5%	-2,86321
		valores críticos 10%	-2,56771

Tabela 6.38 Teste ADF para série NDLSOJA

Na Figura 6.80 é apresentada a estatística de Jarque-Bera, que rejeita fortemente a hipótese de distribuição normal.

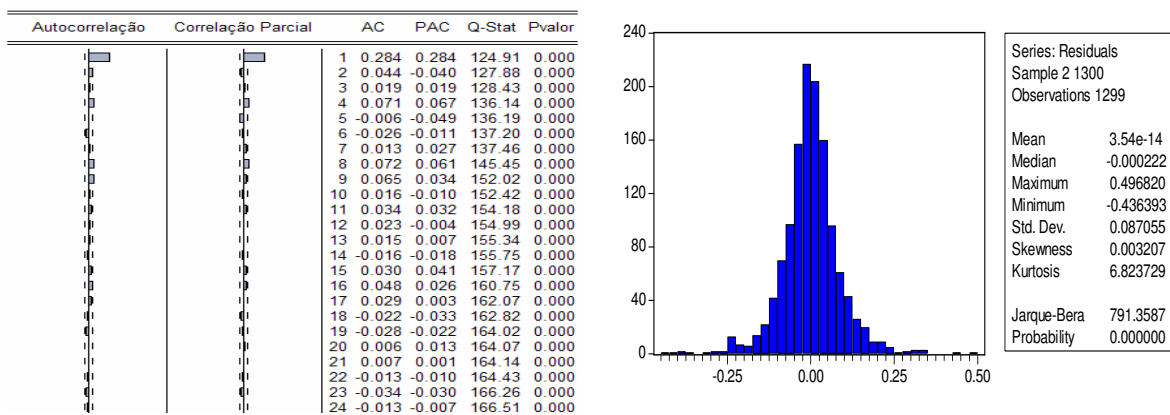


Figura 6.80 (a) Correlograma da série NDLSOJA, (b) Estatísticas dos resíduos

Os resultados para o teste BDS são apresentados na Tabela 6.39.

BDS				
Dimensão	Estatística	Erro-padrão	Estatística-z	p-valor
2	0,026691	0,002309	11,56045	0,0000
3	0,050396	0,003663	13,75628	0,0000
4	0,064790	0,004356	14,87377	0,0000
5	0,072188	0,004534	15,92260	0,0000
6	0,076479	0,004366	17,51681	0,0000

Tabela 6.39 Resultados obtidos pelo teste BDS nas respectivas dimensões.

O melhor modelo ajustado para a série é um AR(1)-GARCH(1,1). Os parâmetros estimados do modelo para a série são mostrados na Tabela 6.40.

Parâmetros	Coefficiente	Erro-padrão	Estatística-z	p-valor
ϕ_0	0,440591	0,003062	143,9029	0,0000
ϕ_1	0,322893	0,029698	10,87238	0,0000
α_0	0,000248	4,86E-05	5,104793	0,0000
α_1	0,107116	0,013544	7,908650	0,0000
β_1	0,863172	0,015222	56,70457	0,0000

Tabela 6.40 Parâmetros estimados para a série NDLSOJA

O resultado do teste Ljung-Box para a série NDLSOJA é apresentado na Figura 6.81.

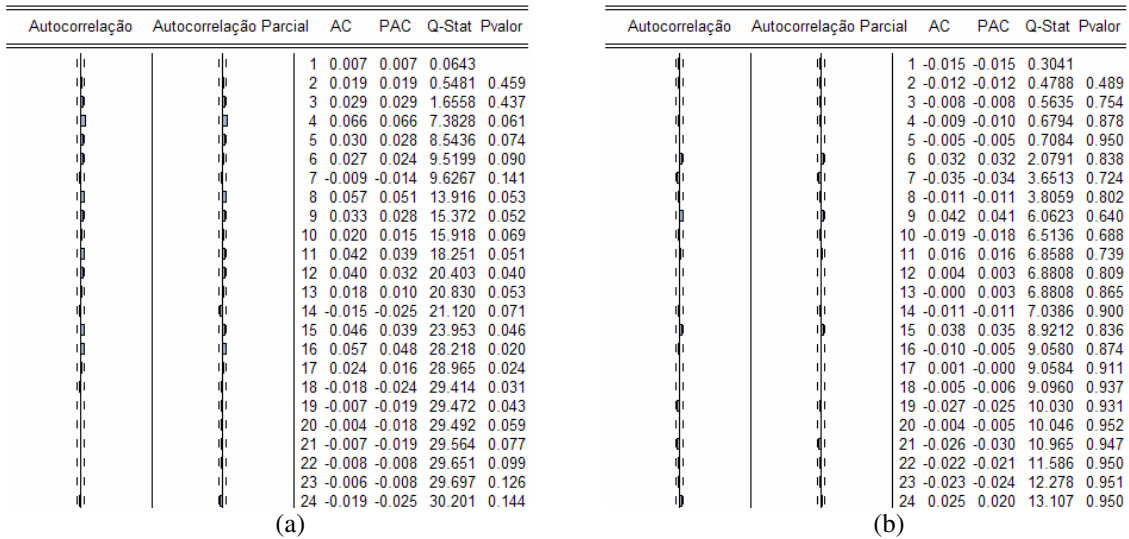


Figura 6.81 (a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado

As previsões realizadas utilizando o modelo ARIMA-GARCH são mostradas na Figura 6.82. A linha contínua indica o valor da série NDLSOJA e a linha central com marcador quadrado indica as previsões.

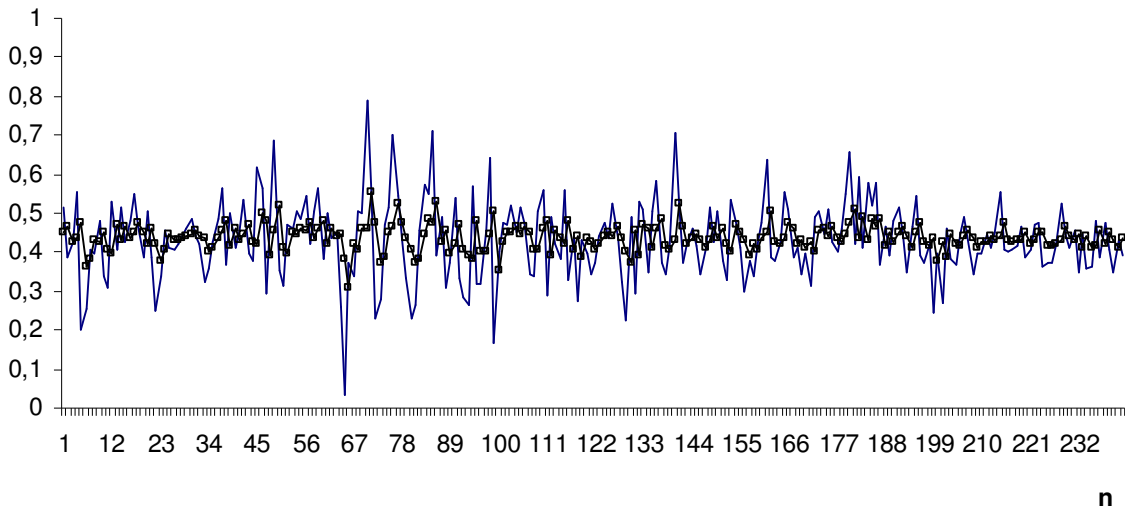


Figura 6.82 Previsão obtida pelo modelo ARIMA-GARCH para NDLSOJA

Na Figura 6.83a é mostrada a previsão da volatilidade e na Figura 6.83b a série original e o intervalo de previsão para a série NDLSOJA.

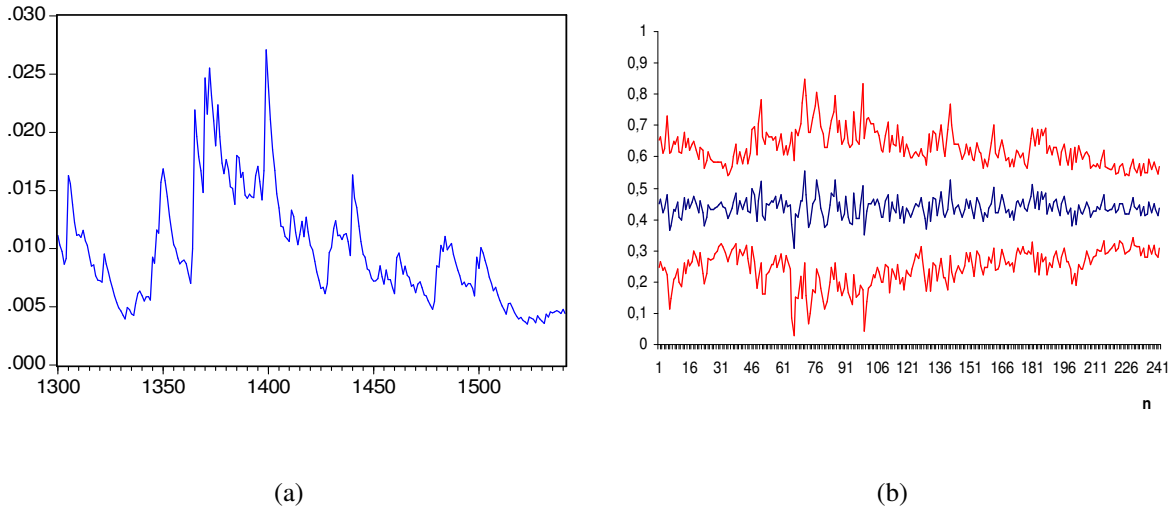


Figura 6.83 (a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLSOJA

6.3.3.2 Análise da série da Soja – Método RNA *feedforward*

Para a série NDLSOJA os resultados obtidos pelas diferentes topologias de RNA são apresentados na Tabela 6.41.

Topologia	FIC	FIP	TIC	RMSE
RNA 10-1-1	100	0	0,113623	0,10056
RNA 10-2-1	100	0	0,114457	0,101523
RNA 10-3-1	101	0	0,118568	0,10518
RNA 10-4-1	96	0	0,117766	0,104133
RNA 10-5-1	65	0	0,116789	0,103834
RNA 10-8-1	68	1	0,19517	0,176874

Tabela 6.41 Estatísticas para diferentes topologias de RNA para a série

Na Figura 6.84 são apresentadas as previsões obtidas pela RNA 10-1-1 para a série NDLSOJA.

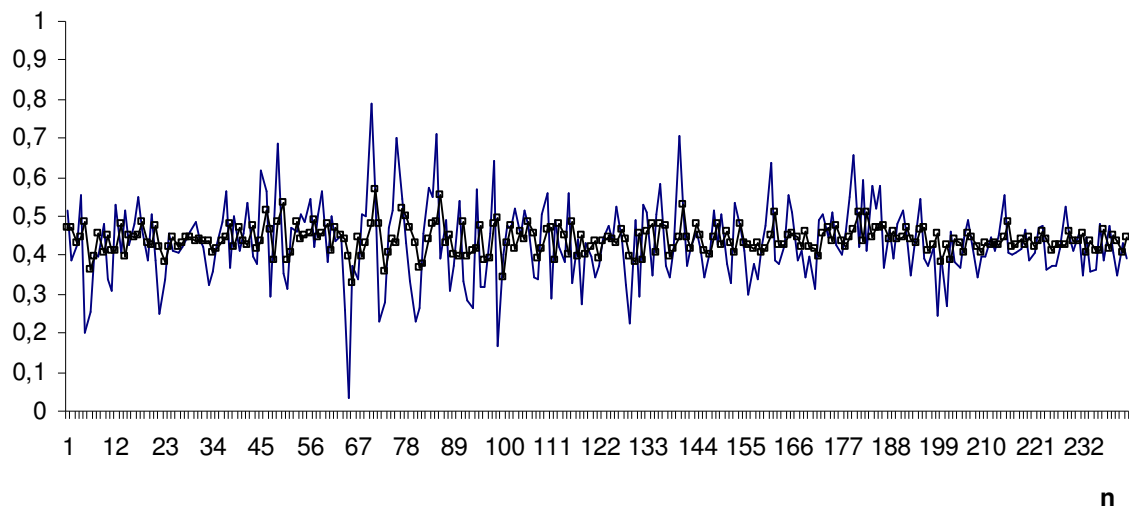


Figura 6.84 Previsão obtida pela RNA *feedforward* para NDLSOJA

Na Figura 6.85a é mostrado o intervalo de confiança para as previsões obtidas pela RNA 10-1-1 e na Figura 6.85b o intervalo de predição para a série NDLSOJA.

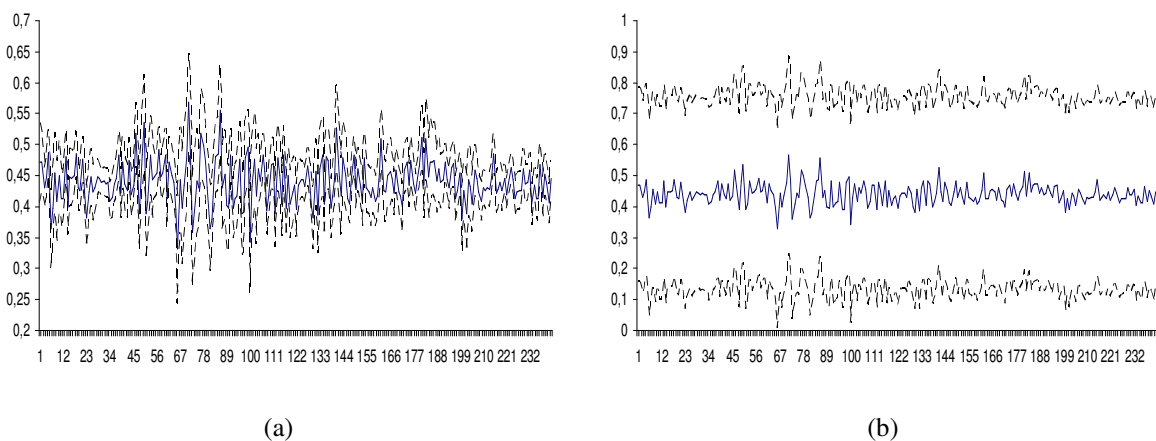


Figura 6.85 (a) Intervalo de confiança, (b) Intervalo de predição para NDLSOJA

Na Figura 6.86a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA 10-1-1 e na Figura 6.86b o histograma desses resíduos.

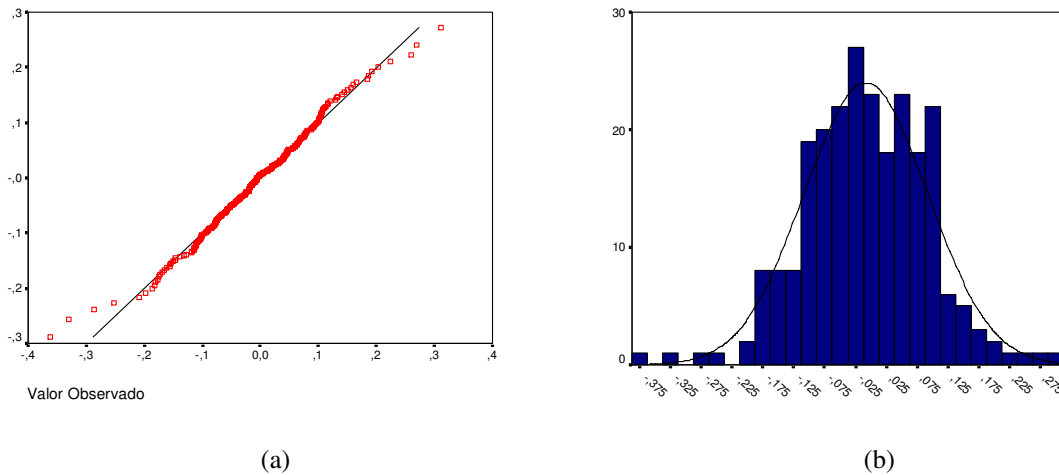


Figura 6.86 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA *feedforward* para NDLSOJA

6.3.3.3 Análise da série da Soja – Método RNA EKF

As previsões, para a série NDLSOJA, obtidas pela RNA treinada com o filtro de Kalman estendido (EKF) são mostradas na Figura 6.87.

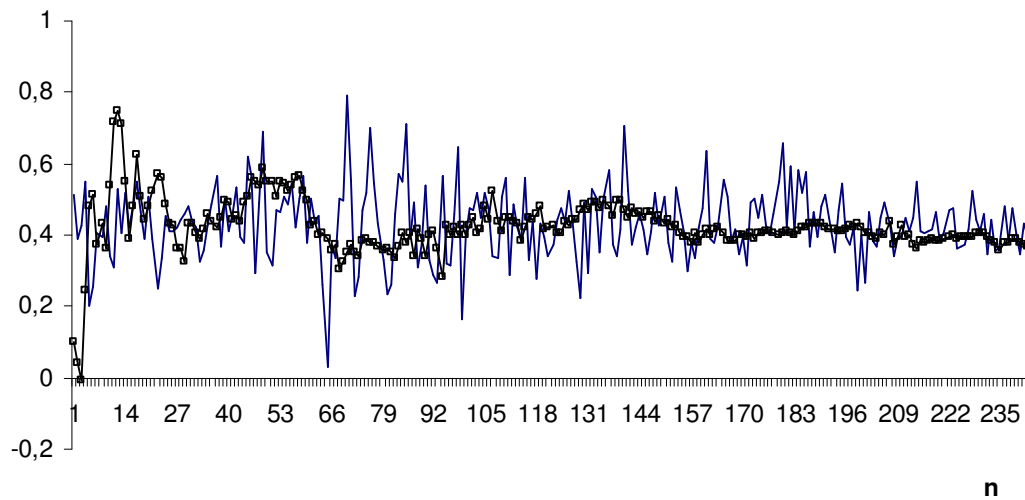


Figura 6.87 Previsão obtida pela RNA EKF para NDLSOJA

Na Figura 6.88a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman estendido e na Figura 6.88b o histograma desses resíduos.

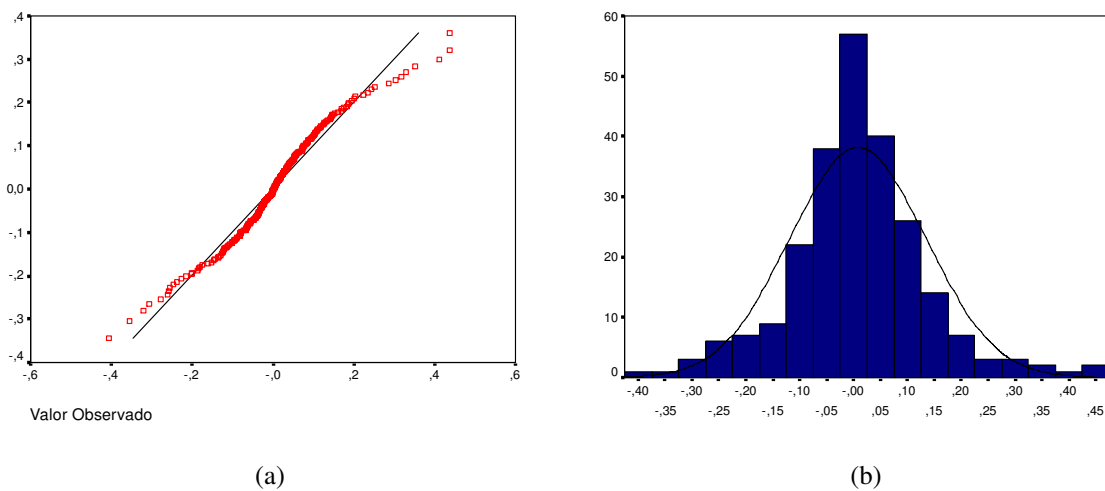


Figura 6.88 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLSOJA

6.3.3.4 Análise da série da Soja – Método RNA UKF

As previsões obtidas pela RNA treinada com o filtro de Kalman unscented são mostradas na Figura 6.89.

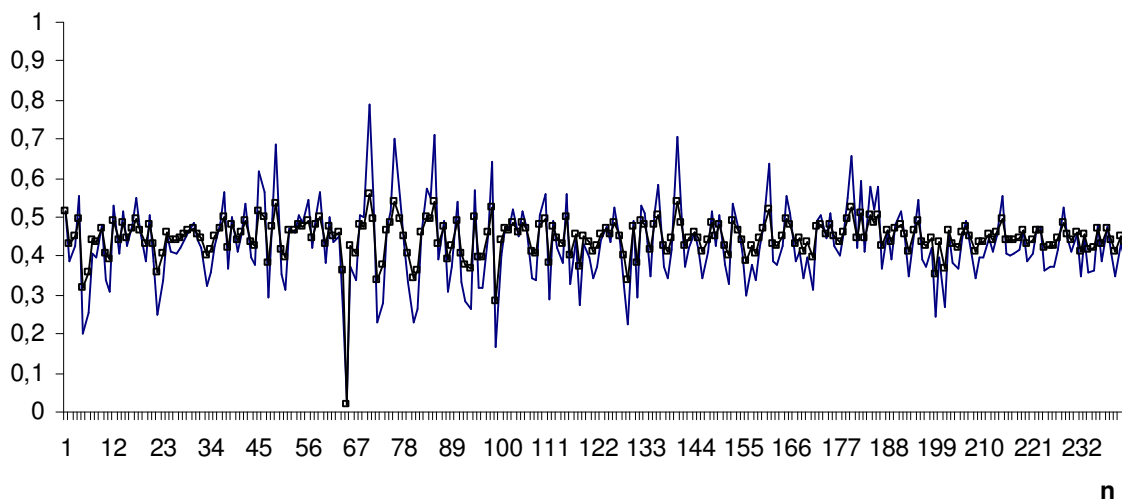


Figura 6.89 Previsão obtida pela RNA UKF para NDLSOJA

Na Figura 6.90a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman *unscented* e na Figura 6.90b o histograma desses resíduos.

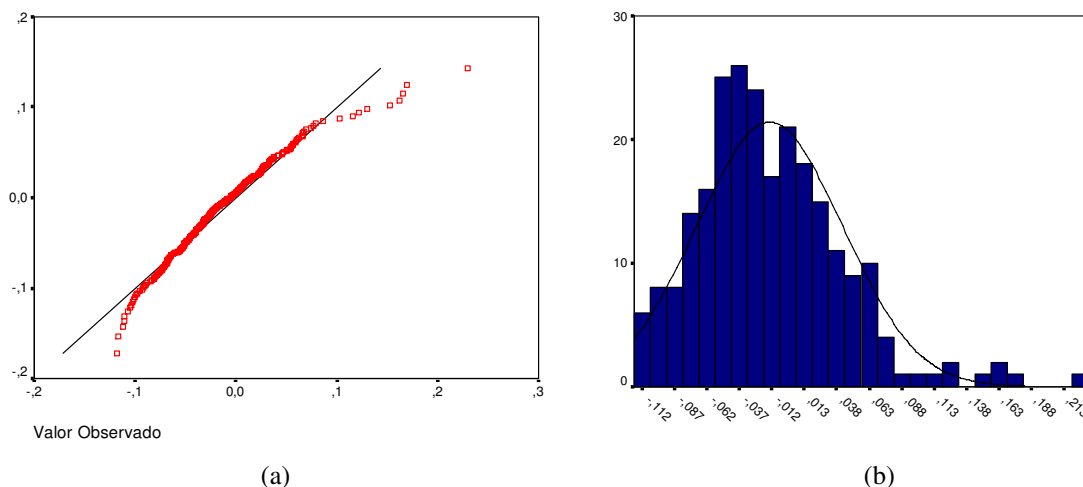


Figura 6.90 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLSOJA

As estatísticas de previsão obtidas pelos modelos ARIMA-GARCH, RNA *feedforward*, RNA EKF e RNA UKF são apresentadas na Tabela 6.42 e na Figura 6.91.

Método	TIC	RMSE
AR(1)-GARCH(1,1)	0,113194	0,099915
RNA 10-1-1	0,113623	0,10056
RNA EKF	0,144380	0,126283
RNA UKF	0,064885	0,0579267

Tabela 6.42 Estatísticas de previsão para o método de ARIMA-GARCH e de RNA

Pode-se verificar que as melhores previsões foram obtidas pelo método que empregou redes neurais treinadas com o filtro de Kalman *unscented*.

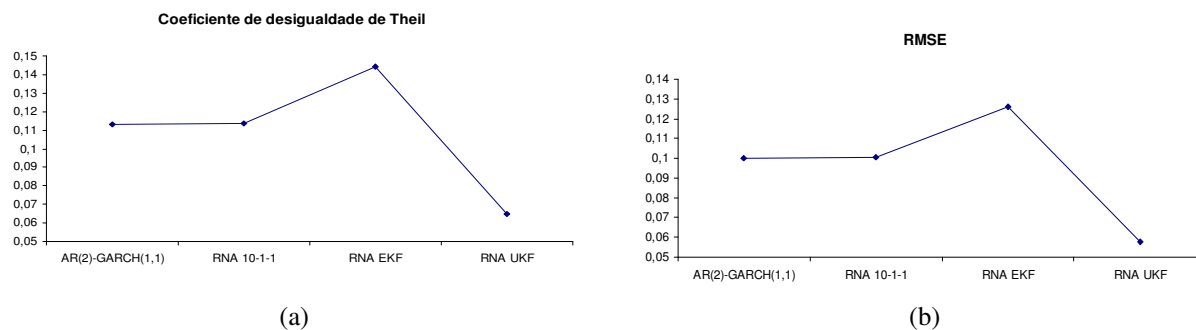


Figura 6.91 Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA

6.3.4 Análise da série do Açúcar

A seguir são apresentados os resultados obtidos para a série do Açúcar.

6.3.4.1 Análise da série do Açúcar – Método ARIMA-GARCH

Na Figura 6.92a é apresentada a série original do preço de fechamento do Açúcar, na Figura 6.92b é apresentada a série normalizada da primeira diferença do logaritmo do preço, NDLCANA.

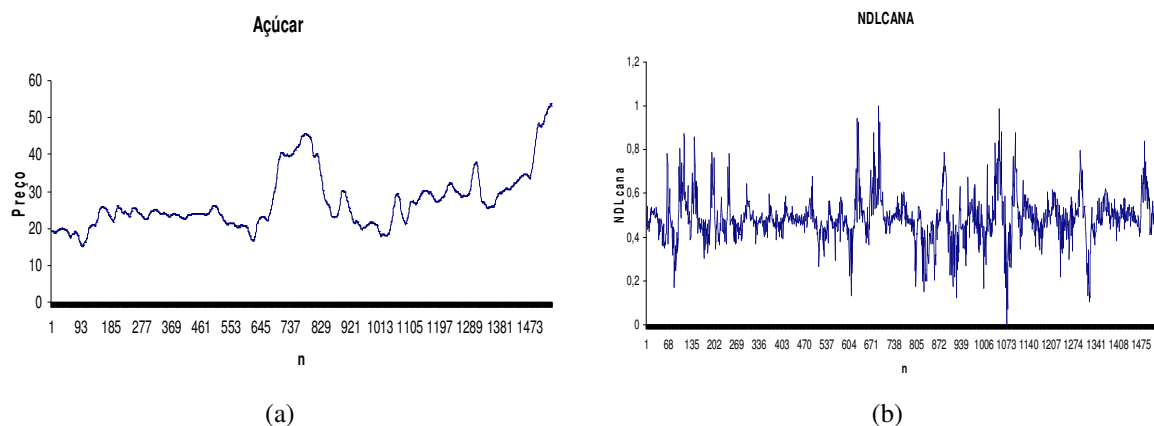


Figura 6.92 (a) Série no nível do preço do Bradesco, (b) Série transformada NDLCANA

O resultado do teste ADF, para a série original, é apresentado na Tabela 6.43.

Estatística do teste ADF para série original		valores críticos 1%	-3,43439
	-1,98074	valores críticos 5%	-2,86321
		valores críticos 10%	-2,56771

Tabela 6.43 Teste ADF para série original do Açúcar

Não é rejeitada a hipótese de raiz unitária para a série no nível pelo teste ADF. O teste de presença de uma raiz unitária para a série transformada apresentou os seguintes resultados:

Estatística do teste ADF para série transformada		valores críticos 1%	-3,43439
	-9,20293	valores críticos 5%	-2,86321
		valores críticos 10%	-2,56771

Tabela 6.44 Teste ADF para série NDLCANA

Na Figura 6.93 é apresentada a estatística de Jarque-Bera, que rejeita fortemente a hipótese de distribuição normal.

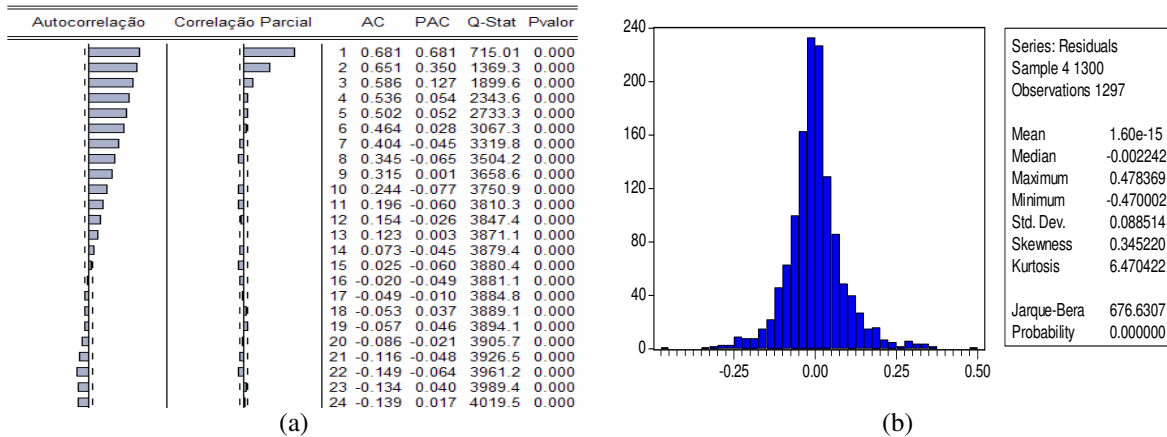


Figura 6.93 (a) Correlograma da série NDLCANA, (b) Estatísticas dos resíduos

Os resultados para o teste BDS são apresentados na Tabela 6.45.

BDS				
Dimensão	Estatística	Erro-padrão	Estatística-z	p-valor
2	0,077008	0,002794	27,55834	0,0000
3	0,140875	0,004442	31,71736	0,0000
4	0,182521	0,005292	34,48987	0,0000
5	0,206403	0,005520	37,39231	0,0000
6	0,218401	0,005328	40,99084	0,0000

Tabela 6.45 Resultados obtidos pelo teste BDS nas respectivas dimensões.

O melhor modelo ajustado para a série é um ARMA(1,1)-GARCH(1,2). Os parâmetros estimados do modelo para a série são mostrados na Tabela 6.46.

Parâmetros	Coefficiente	Erro-padrão	Estatística-z	p-valor
ϕ_0	0,479819	0,011600	41,36228	0,0000
ϕ_1	0,933476	0,014003	66,66069	0,0000
θ_1	-0,542762	0,033284	-16,30678	0,0000
α_0	0,000129	2,42E-05	5,344430	0,0000
α_1	0,191833	0,024317	7,888991	0,0000
β_1	0,422729	0,133919	3,156605	0,0016
β_2	0,383313	0,118277	3,240796	0,0012

Tabela 6.46 Parâmetros estimados para a série NDLCANA

O resultado do teste Ljung-Box para a série NDLCANA é apresentado na Figura 6.94.

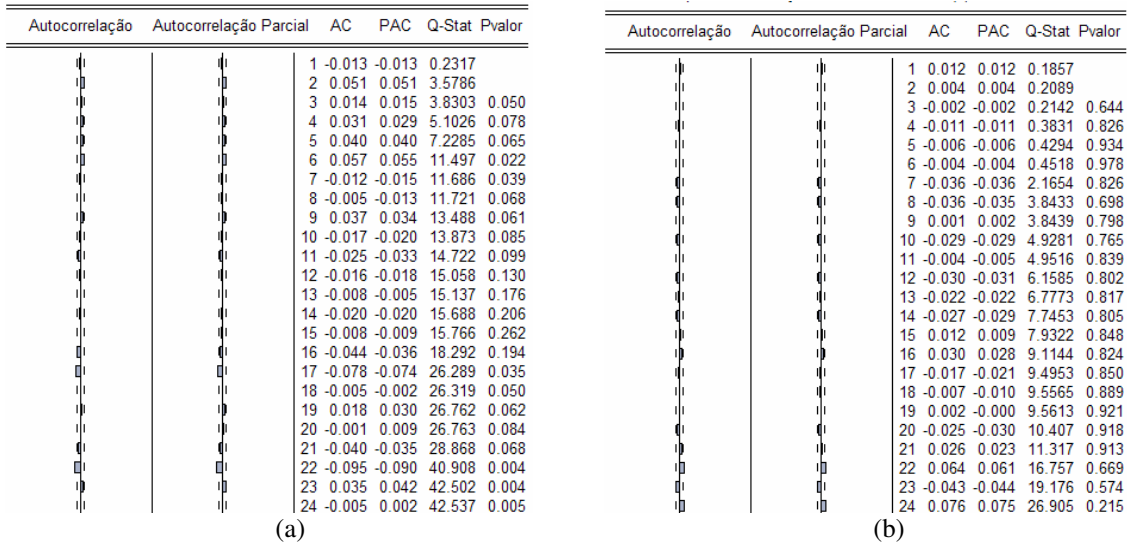


Figura 6.94 (a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado

As previsões realizadas utilizando o modelo ARIMA-GARCH são mostradas na Figura 6.95. A linha contínua indica o valor da série NDLCANA e a linha central com marcador quadrado indica as previsões.

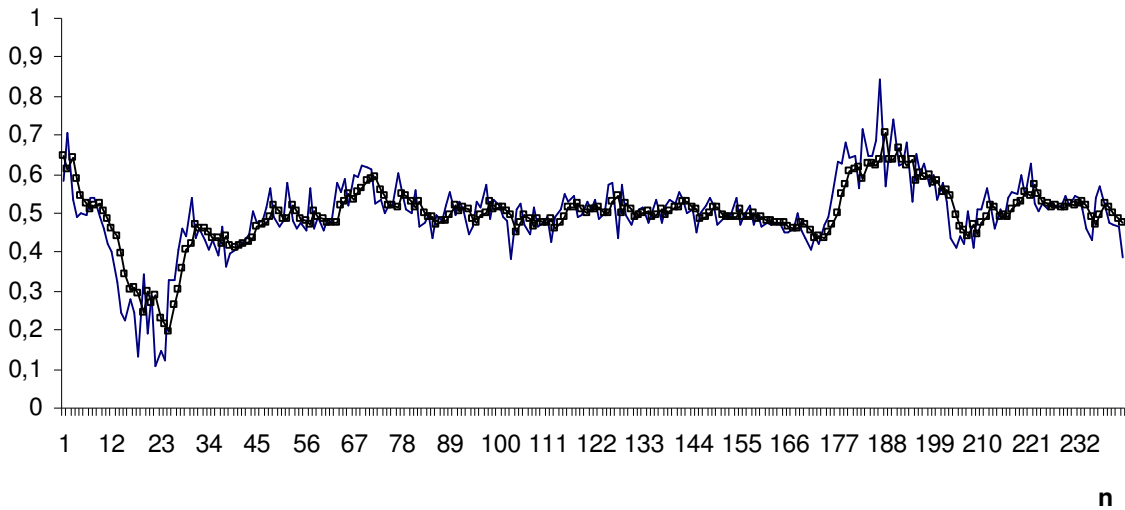


Figura 6.95 Previsão obtida pelo modelo ARIMA-GARCH para NDLCANA

Na Figura 6.96a é mostrada a previsão da volatilidade e na Figura 6.96b a série original e o intervalo de previsão para a série NDLCANA.

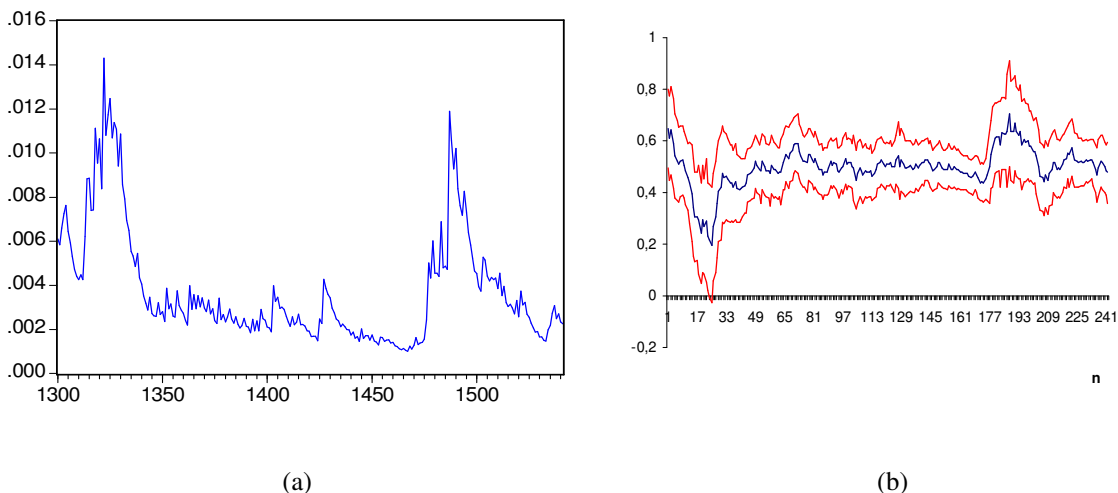


Figura 6.96 (a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLCANA

6.3.4.2 Análise da série do Açúcar – Método RNA *feedforward*

Para a série NDLCANA os resultados obtidos pelas diferentes topologias de RNA são apresentados na Tabela 6.47.

Topologia	FIC	FIP	TIC	RMSE
RNA 10-1-1	10	1	0,0561481	0,0568204
RNA 10-2-1	13	0	0,0549061	0,0556192
RNA 10-3-1	15	0	0,0564448	0,0572777
RNA 10-4-1	10	0	0,0569026	0,0576001
RNA 10-5-1	5	0	0,0558808	0,056489
RNA 10-8-1	16	0	0,0583576	0,0592224

Tabela 6.47 Estatísticas para diferentes topologias de RNA para a série

Na Figura 6.97 são apresentadas as previsões obtidas pela RNA 10-2-1 para a série NDLCANA.

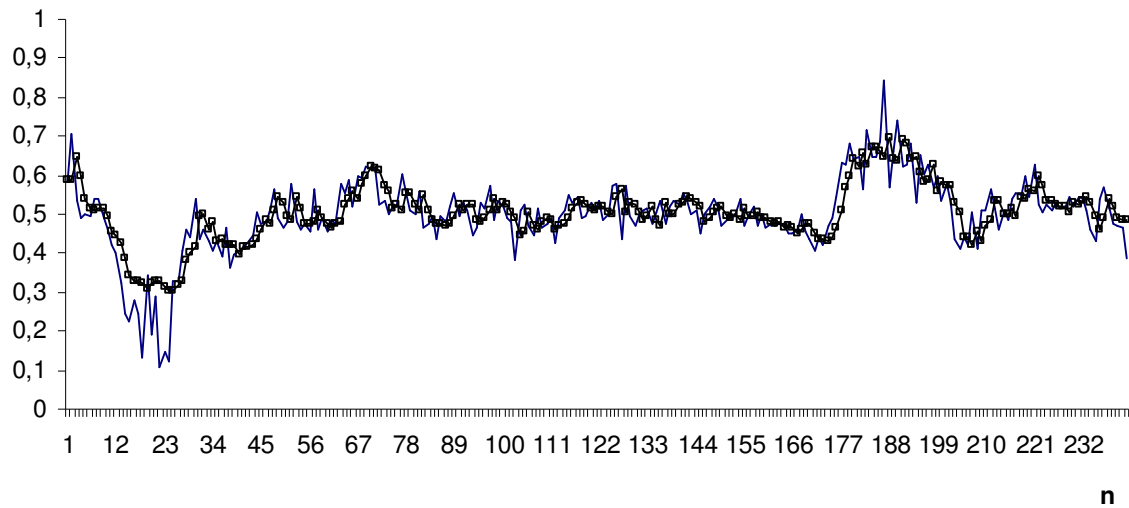


Figura 6.97 Previsão obtida pela RNA *feedforward* para NDLCANA

Na Figura 6.98a é mostrado o intervalo de confiança para as previsões obtidas pela RNA 10-2-1 e na Figura 6.98b o intervalo de predição para a série NDLCANA.

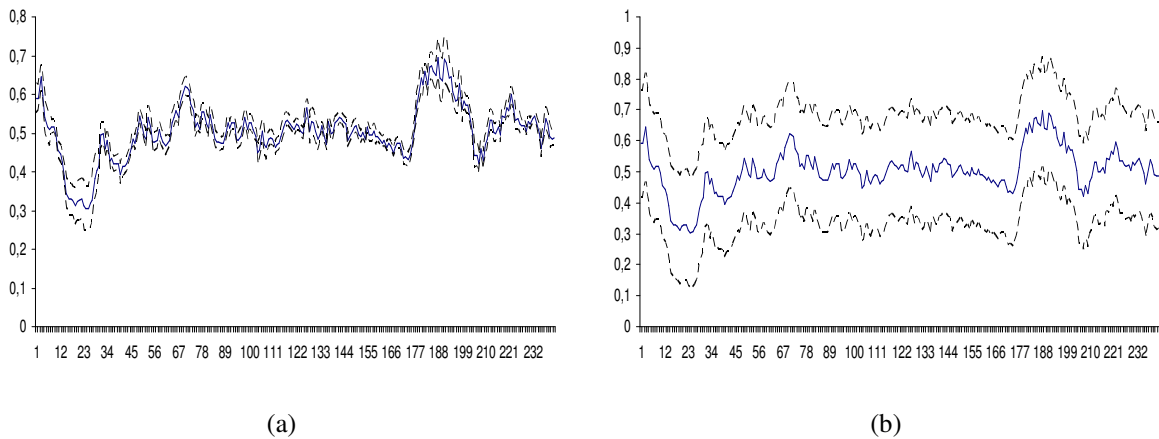
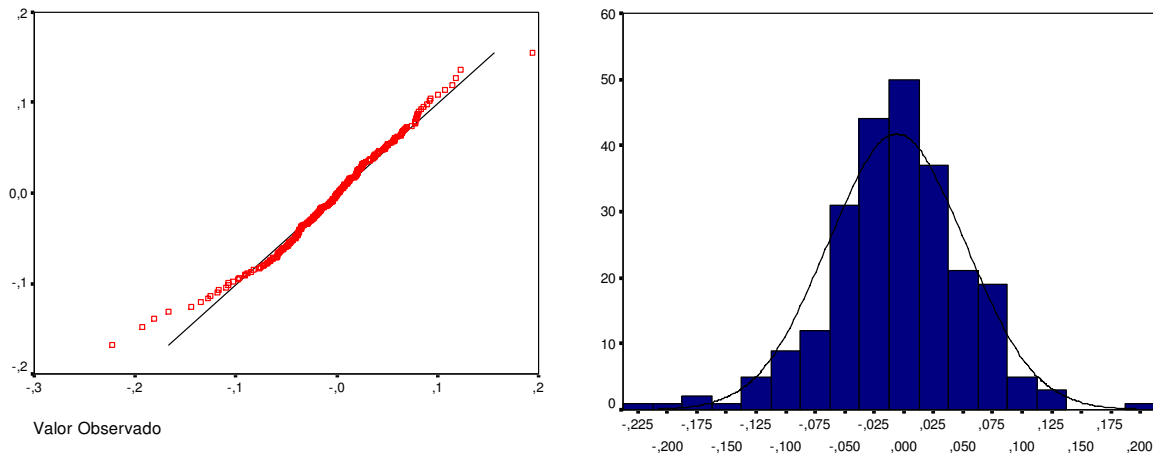


Figura 6.98 (a) Intervalo de confiança, (b) Intervalo de predição para NDLCANA

Na Figura 6.99a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA 10-2-1 e na Figura 6.99b o histograma desses resíduos.



(a) (b)
Figura 6.99 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA *feedforward* para NDLCANA

6.3.4.3 Análise da série do Açúcar – Método RNA EKF

As previsões, para a série NDLCANA, obtidas pela RNA treinada com o filtro de Kalman estendido (EKF) são mostradas na Figura 6.100.

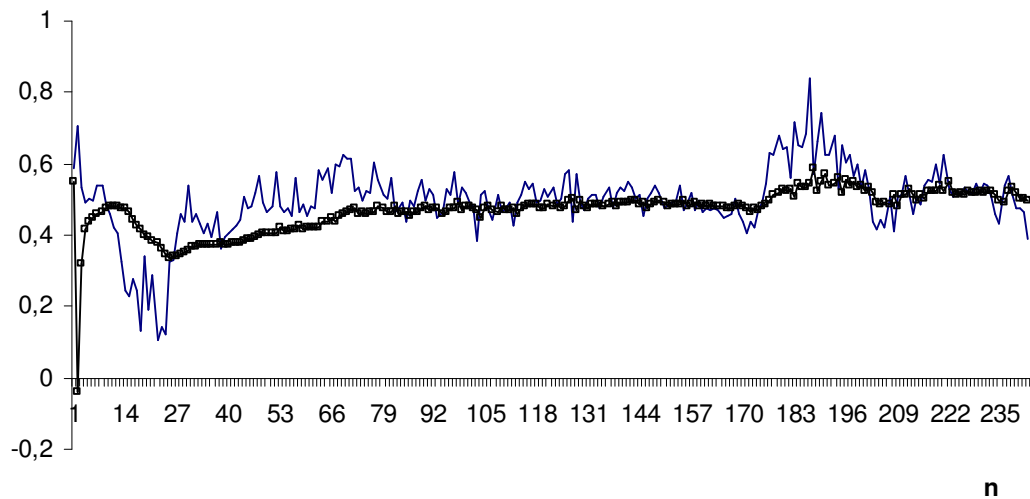


Figura 6.100 Previsão obtida pela RNA EKF para NDLCANA

Na Figura 6.101a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman estendido e na Figura 6.101b o histograma desses resíduos.

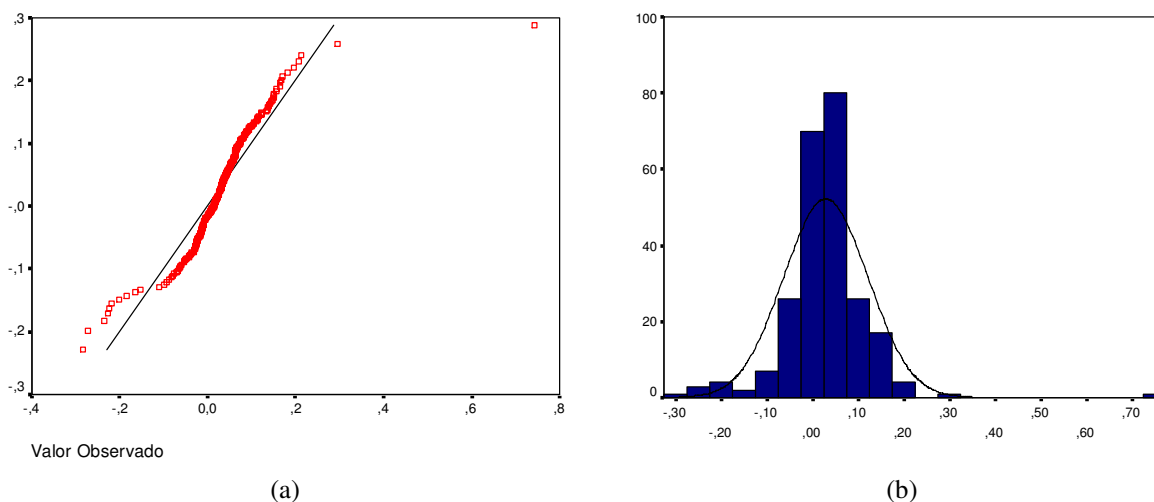


Figura 6.101 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLCANA

6.3.4.4 Análise da série do Açúcar – Método RNA UKF

As previsões obtidas pela RNA treinada com o filtro de Kalman unscented são mostradas na Figura 6.102.

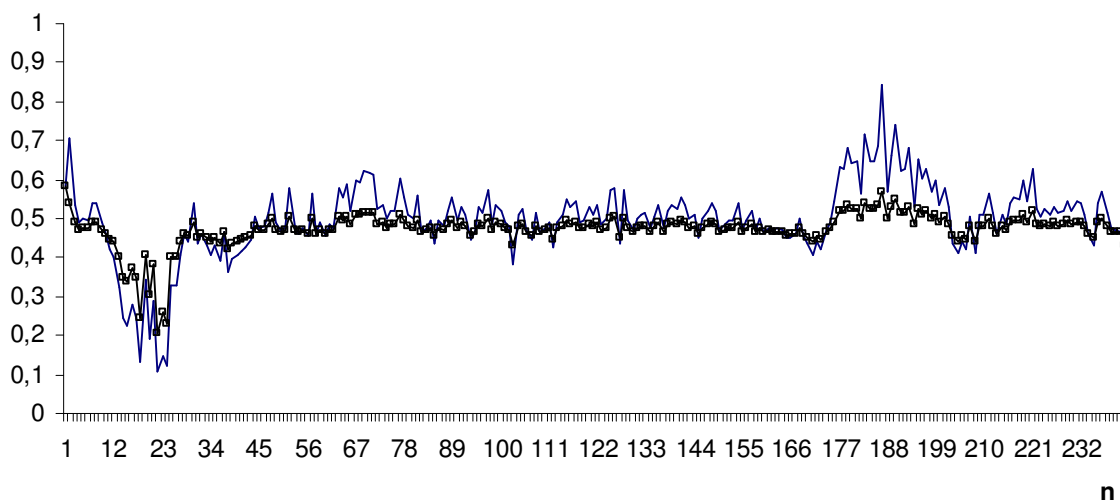


Figura 6.102 Previsão obtida pela RNA UKF para NDLCANA

Na Figura 6.103a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman *unscented* e na Figura 6.103b o histograma desses resíduos.

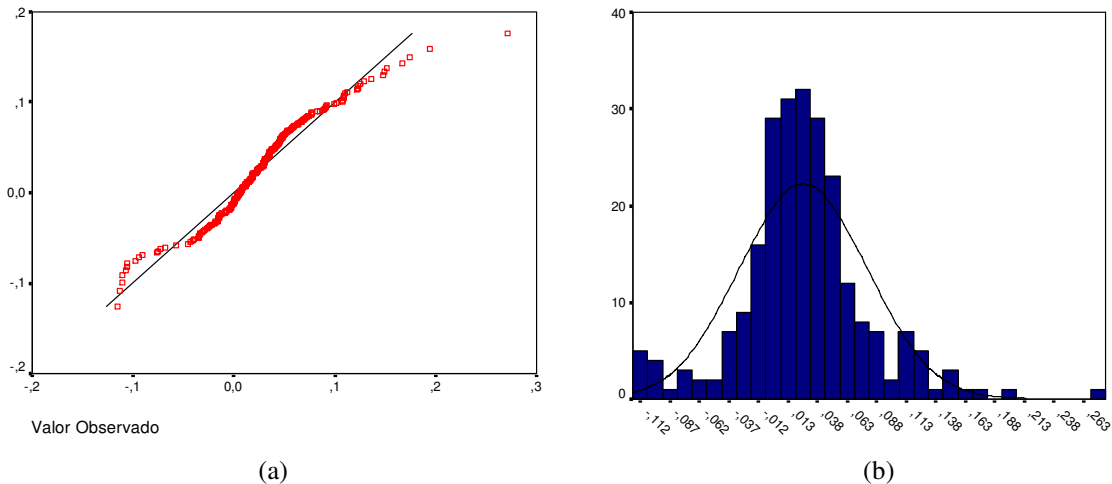


Figura 6.103 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLCANA

As estatísticas de previsão obtidas pelos modelos ARIMA-GARCH, RNA *feedforward*, RNA EKF e RNA UKF são apresentadas na Tabela 6.48 e na Figura 6.104.

Método	TIC	RMSE
ARMA(1,1)-GARCH(1,2)	0,054873	0,055313
RNA 10-2-1	0,113623	0,100560
RNA EKF	0,144380	0,126283
RNA UKF	0,053993	0,053920

Tabela 6.48 Estatísticas de previsão para o método de ARIMA-GARCH e de RNA

Pode-se verificar que as melhores previsões foram obtidas pelo método que empregou redes neurais treinadas com o filtro de Kalman *unscented*.

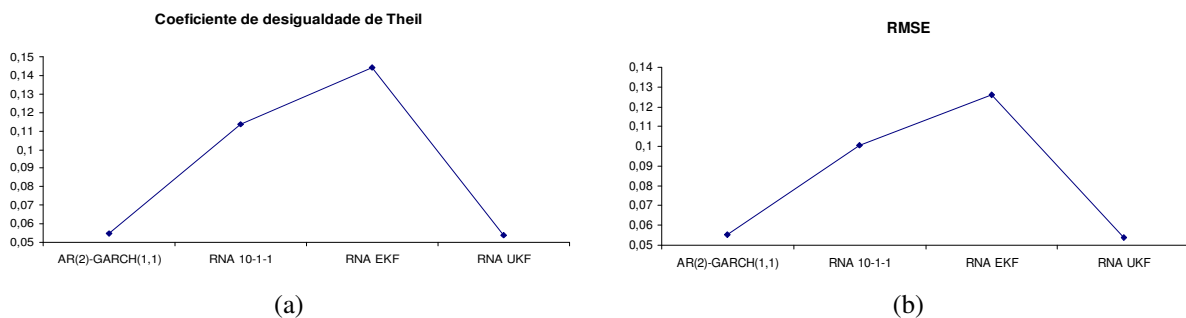


Figura 6.104 Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b)

RMSE

6.4 Setor Industrial – Marcopolo, Petrobrás, Embraer, Ripasa, Souza Cruz e Gerdau

No caso do setor industrial as séries analisadas seguirão a seqüência: Marcopolo, Petrobrás, Embraer, Ripasa, Souza Cruz e a Gerdau.

6.4.1 Análise da série da Marcopolo

A seguir são apresentados os resultados obtidos para a série da Marcopolo.

6.4.1.1 Análise da série da Marcopolo – Método ARIMA-GARCH

Na Figura 6.105a é apresentada a série original do preço de fechamento da Marcopolo, na Figura 6.105b é apresentada a série normalizada da primeira diferença do logaritmo do preço, NDLMARC.

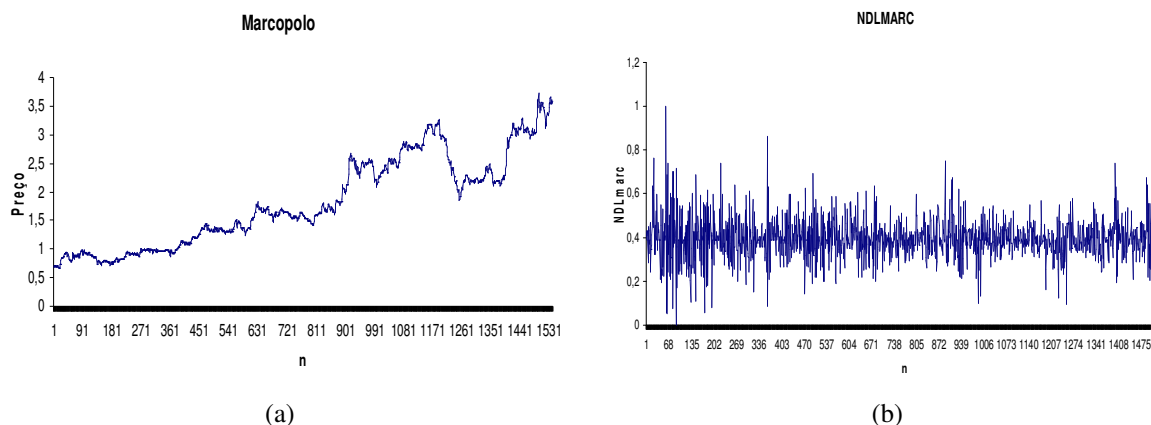


Figura 6.105 (a) Série no nível do preço do Bradesco, (b) Série transformada NDLMARC

O resultado do teste ADF, para a série original, é apresentado na Tabela 6.49.

Estatística do teste ADF para série original	valores críticos 1%	-3,43439
-0,13742	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771

Tabela 6.49 Teste ADF para série original da Marcopolo

Não é rejeitada a hipótese de raiz unitária para a série no nível pelo teste ADF. O teste de presença de uma raiz unitária para a série transformada apresentou os seguintes resultados:

Estatística do teste ADF para série transformada	valores críticos 1%	-3,43439
-44,279	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771

Tabela 6.50 Teste ADF para série NDLMARC

Na Figura 6.106 é apresentada a estatística de Jarque-Bera, que rejeita fortemente a hipótese de distribuição normal.

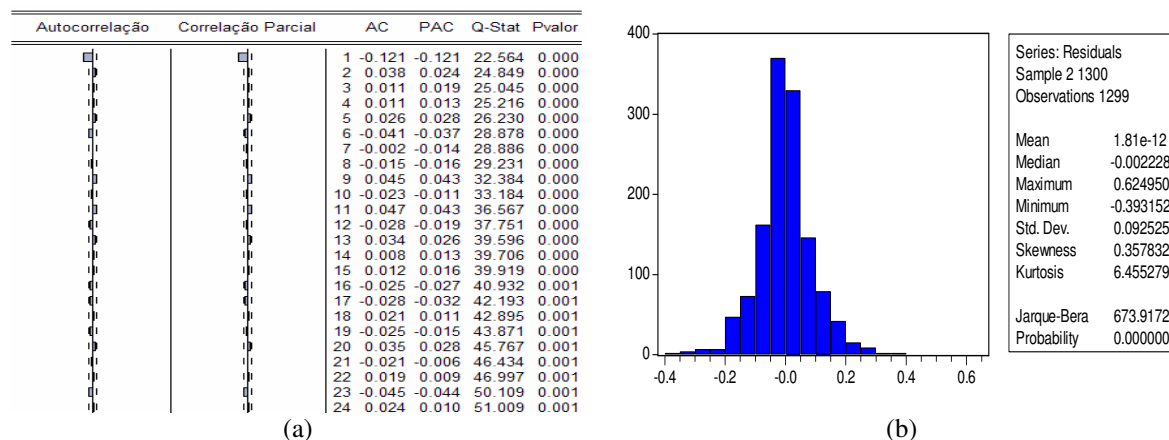


Figura 6.106 (a) Correlograma da série NDLMARC, (b) Estatísticas dos resíduos

Os resultados para o teste BDS são apresentados na Tabela 6.51.

BDS				
Dimensão	Estatística	Erro-padrão	Estatística-z	p-valor
2	0,024105	0,002352	10,24946	0,0000
3	0,045905	0,003729	12,30942	0,0000
4	0,059606	0,004431	13,45071	0,0000
5	0,065103	0,004609	14,12419	0,0000
6	0,065512	0,004436	14,76765	0,0000

Tabela 6.51 Resultados obtidos pelo teste BDS nas respectivas dimensões.

O melhor modelo ajustado para a série é um AR(1)-GARCH(1,3). Os parâmetros estimados do modelo para a série são mostrados na Tabela 6.52.

Parâmetros	Coefficiente	Erro-padrão	Estatística-z	p-valor
ϕ_0	0,391540	0,002055	190,5000	0,0000
ϕ_1	-0,049904	0,030386	-1,642313	0,1005
α_0	0,001052	0,000168	6,276204	0,0000
α_1	0,289674	0,032377	8,946866	0,0000
β_1	0,175690	0,083923	2,093476	0,0363
β_2	0,273404	0,077426	3,531157	0,0004
β_3	0,163260	0,069493	2,349295	0,0188

Tabela 6.52 Parâmetros estimados para a série NDLMARC

O resultado do teste Ljung-Box para a série NDLMARC é apresentado na Figura 6.107.

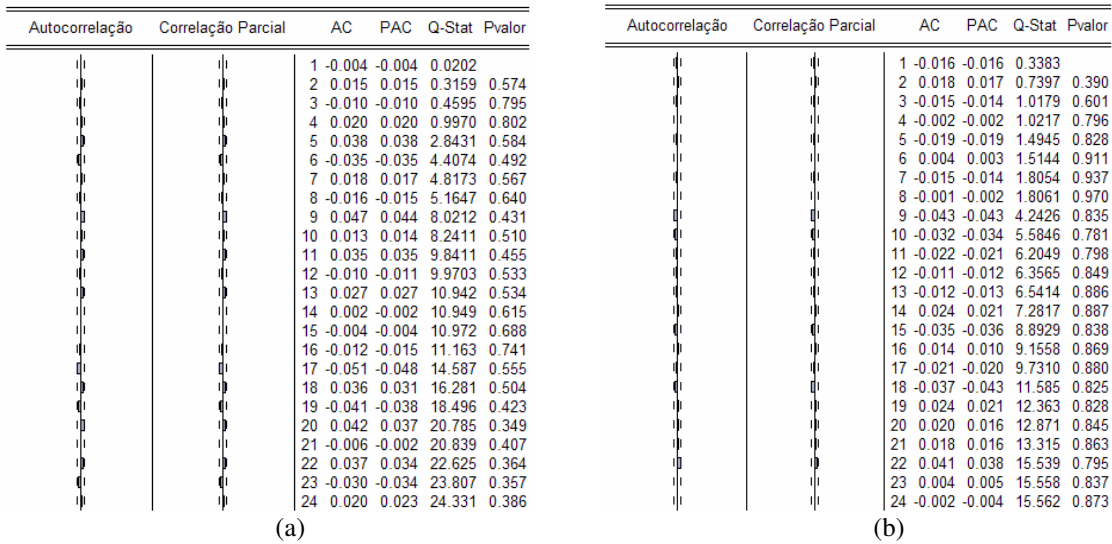


Figura 6.107 (a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado

As previsões realizadas utilizando o modelo ARIMA-GARCH são mostradas na Figura 6.108. A linha contínua indica o valor da série NDLMARC e a linha central com marcador quadrado indica as previsões.

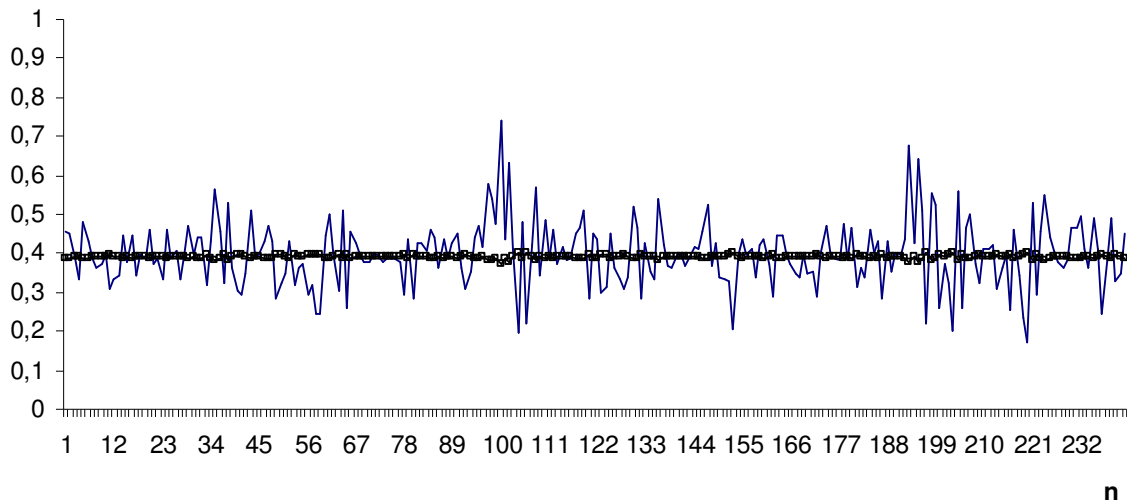


Figura 6.108 Previsão obtida pelo modelo ARIMA-GARCH para NDLMARC

Na Figura 6.109a é mostrada a previsão da volatilidade e na Figura 6.109b a série original e o intervalo de previsão para a série NDLMARC.

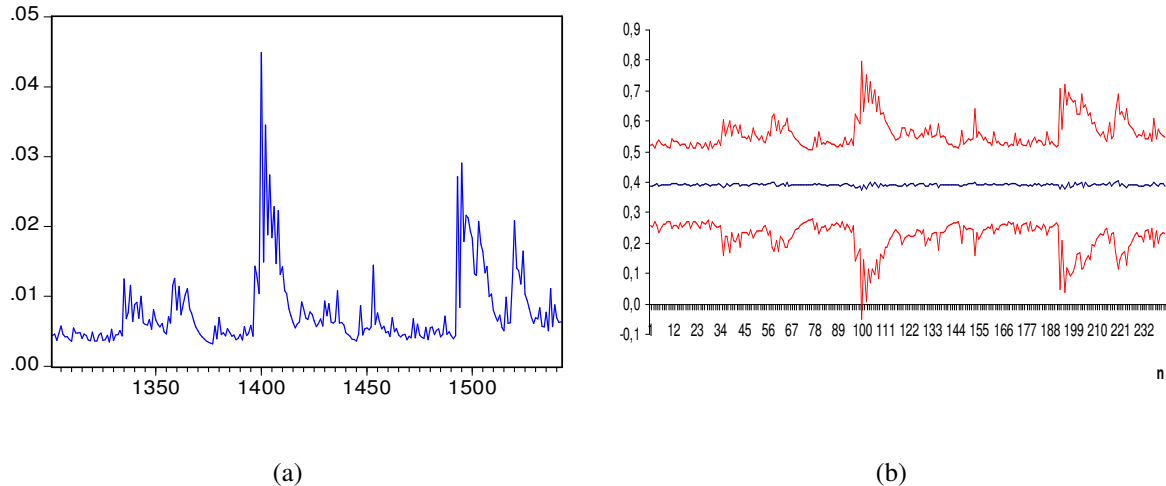


Figura 6.109 (a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLMARC

6.4.1.2 Análise da série da Marcopolo – Método RNA *feedforward*

Para a série NDLMARC os resultados obtidos pelas diferentes topologias de RNA são apresentados na Tabela 6.53.

Topologia	FIC	FIP	TIC	RMSE
RNA 10-1-1	136	2	0,102471	0,0820804
RNA 10-2-1	138	2	0,103583	0,0830482
RNA 10-3-1	139	0	0,101771	0,0813925
RNA 10-4-1	145	0	0,103455	0,08257
RNA 10-5-1	122	0	0,105083	0,0843011
RNA 10-8-1	140	0	0,102898	0,0822796

Tabela 6.53 Estatísticas para diferentes topologias de RNA para a série

A Figura 6.110 apresenta as previsões obtidas pela RNA 10-3-1 para a série NDLMARC.

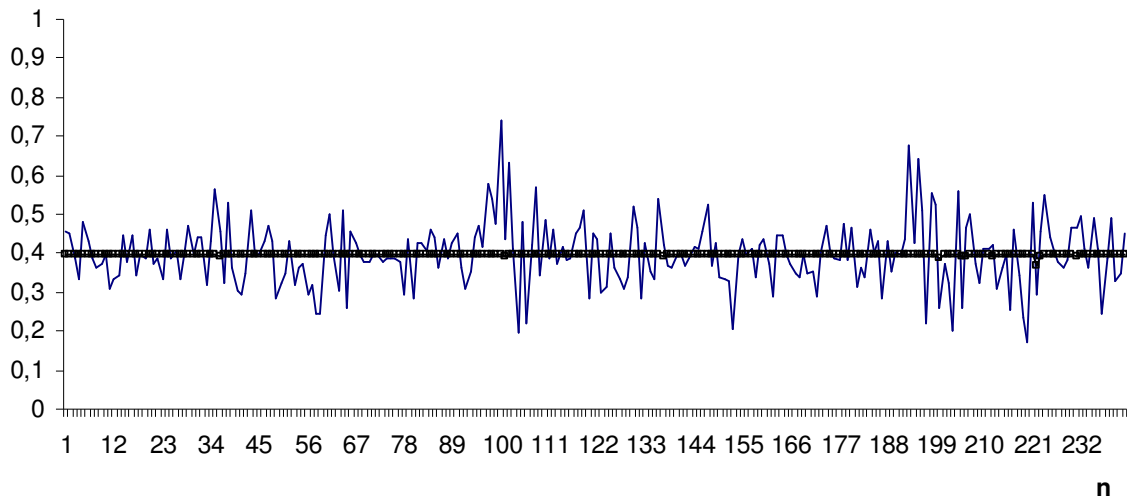


Figura 6.110 Previsão obtida pela RNA *feedforward* para NDLMARC

Na Figura 6.111a é mostrado o intervalo de confiança para as previsões obtidas pela RNA 10-3-1 e na Figura 6.111b o intervalo de predição para a série NDLMARC.

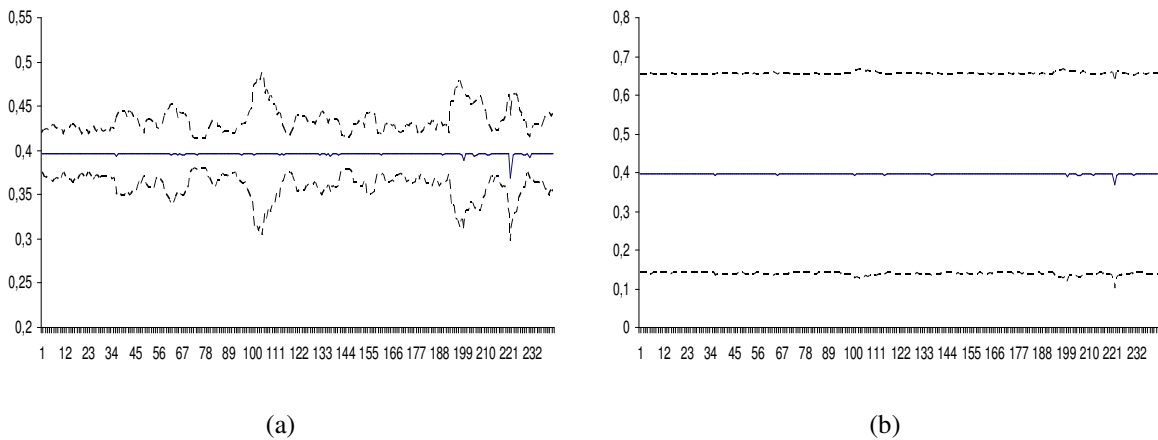


Figura 6.111 (a) Intervalo de confiança, (b) Intervalo de predição para NDLMARC

Na Figura 6.112a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA 10-3-1 e na Figura 6.112b o histograma desses resíduos.

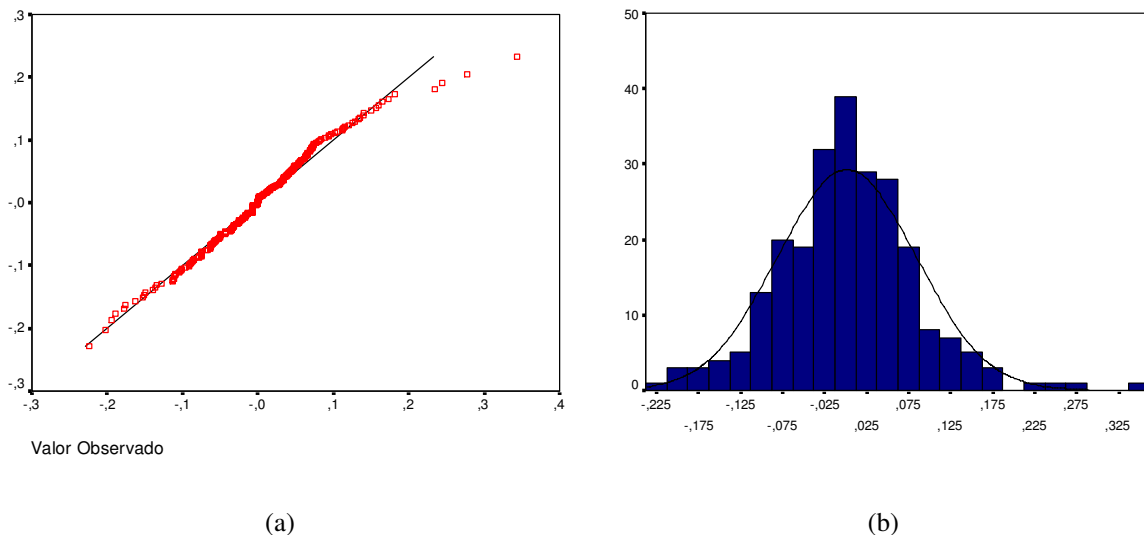


Figura 6.112 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA *feedforward* para NDLMARC

6.4.1.3 Análise da série da Marcopolo – Método RNA EKF

As previsões, para a série NDLMARC, obtidas pela RNA treinada com o filtro de Kalman estendido (EKF) são mostradas na Figura 6.113.

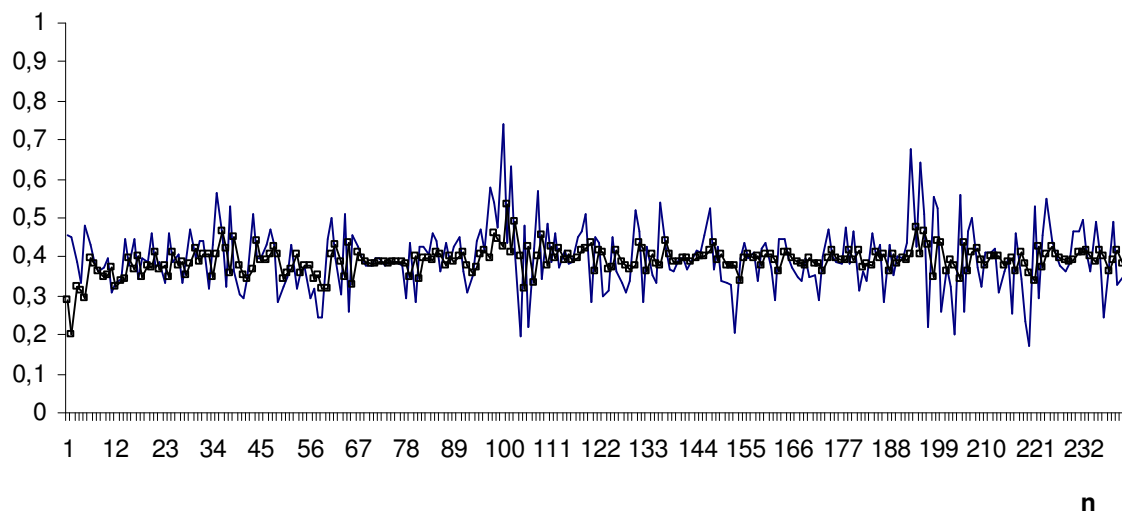


Figura 6.113 Previsão obtida pela RNA EKF para NDLMARC

Na Figura 6.114a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman estendido e na Figura 6.114b o histograma desses resíduos.

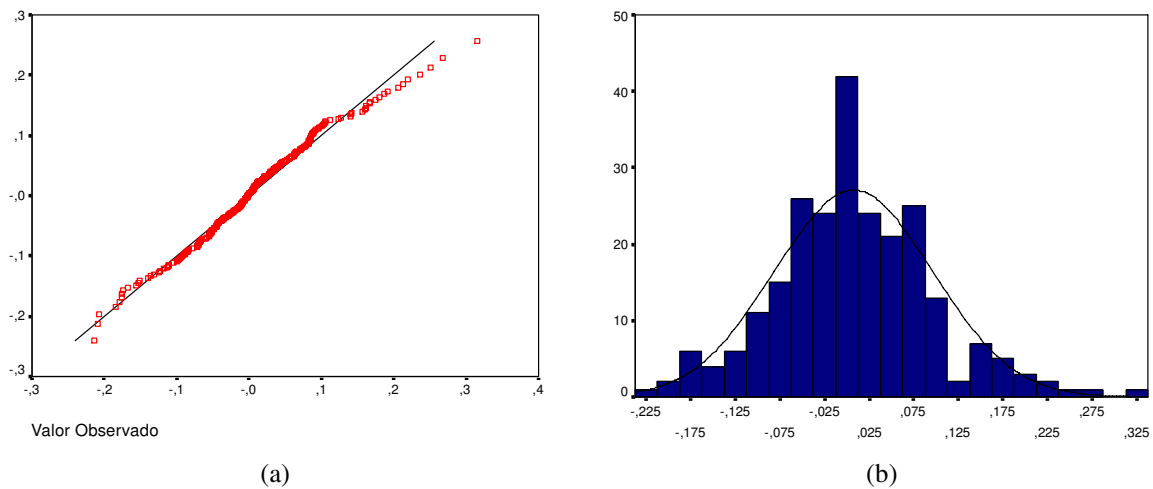


Figura 6.114 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLMARC

6.4.1.4 Análise da série da Marcopolo – Método RNA UKF

As previsões obtidas pela RNA treinada com o filtro de Kalman *unscented* são mostradas na Figura 6.115.

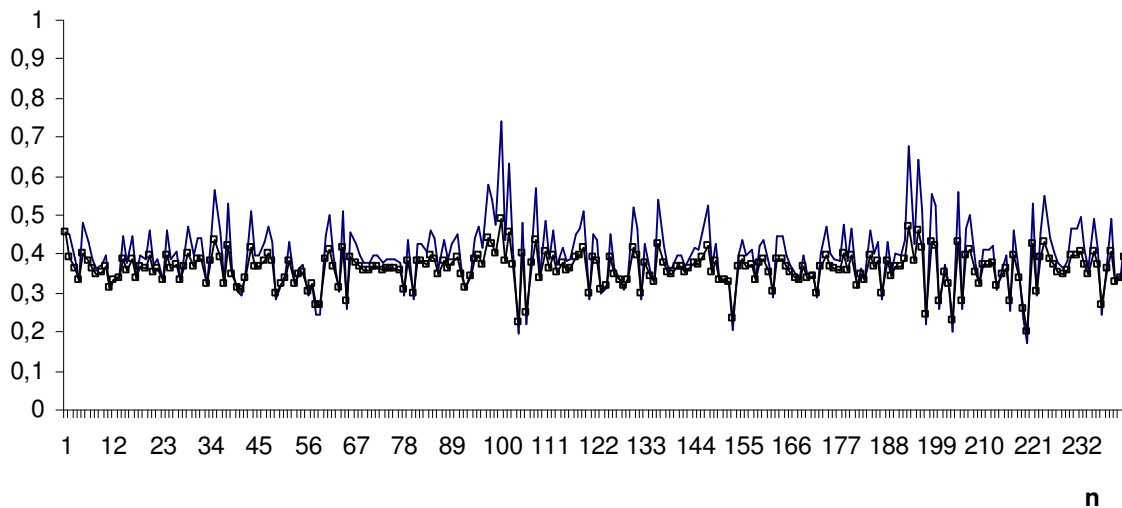


Figura 6.115 Previsão obtida pela RNA UKF para NDLMARC

Na Figura 6.116a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman *unscented* e na Figura 6.116b o histograma desses resíduos.

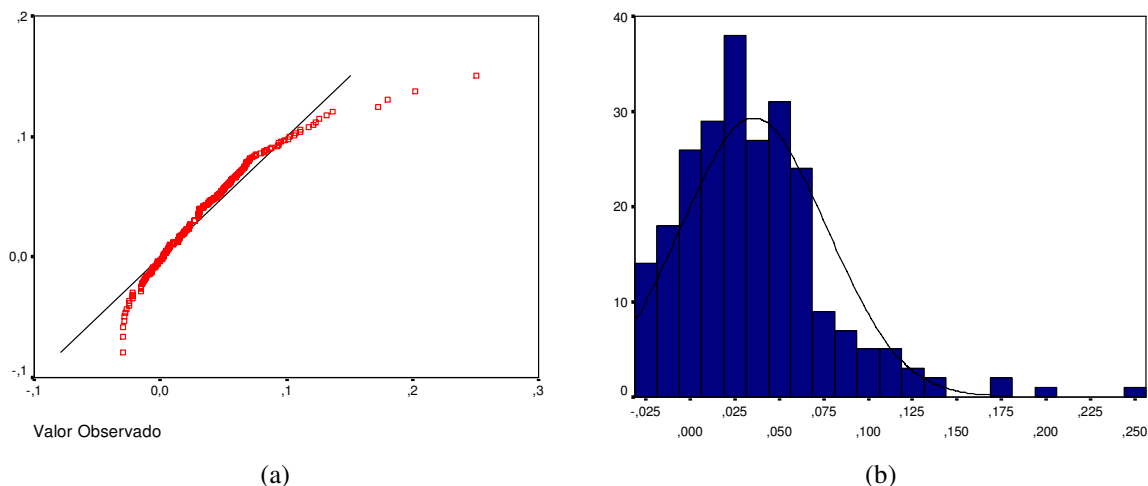


Figura 6.116 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLMARC

As estatísticas de previsão obtidas pelos modelos ARIMA-GARCH, RNA *feedforward*, RNA EKF e RNA UKF são apresentadas na Tabela 6.54 e na Figura 6.117.

Método	TIC	RMSE
AR(1)-GARCH(1,3)	0,103986	0,082921
RNA 10-3-1	0,102471	0,082080
RNA EKF	0,111634	0,089109
RNA UKF	0,070419	0,054314

Tabela 6.54 Estatísticas de previsão para o método de ARIMA-GARCH e de RNA

Pode-se verificar que as melhores previsões foram obtidas pelo método que empregou redes neurais treinadas com o filtro de Kalman *unscented*.

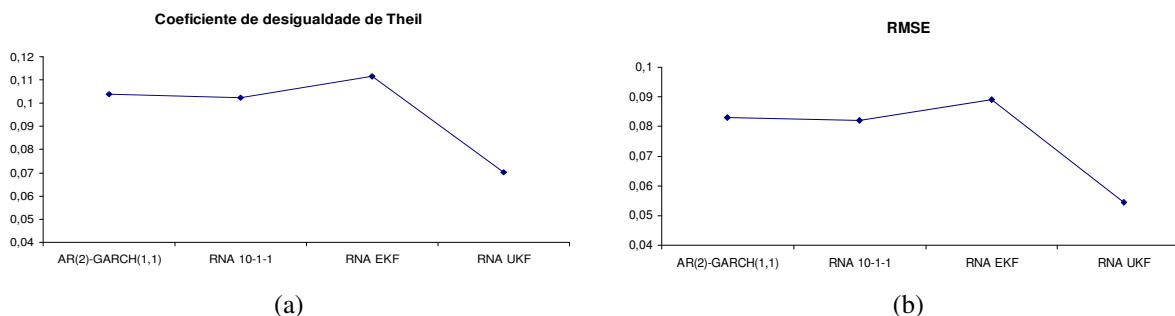


Figura 6.117 Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b) RMSE

6.4.2 Análise da série da Petrobras

A seguir são apresentados os resultados obtidos para a série da Petrobras.

6.4.2.1 Análise da série da Petrobras– Método ARIMA-GARCH

Na Figura 6.118a é apresentada a série original do preço de fechamento da Petrobras, na Figura 6.118b é apresentada a série normalizada da primeira diferença do logaritmo do preço, NDLPETR.

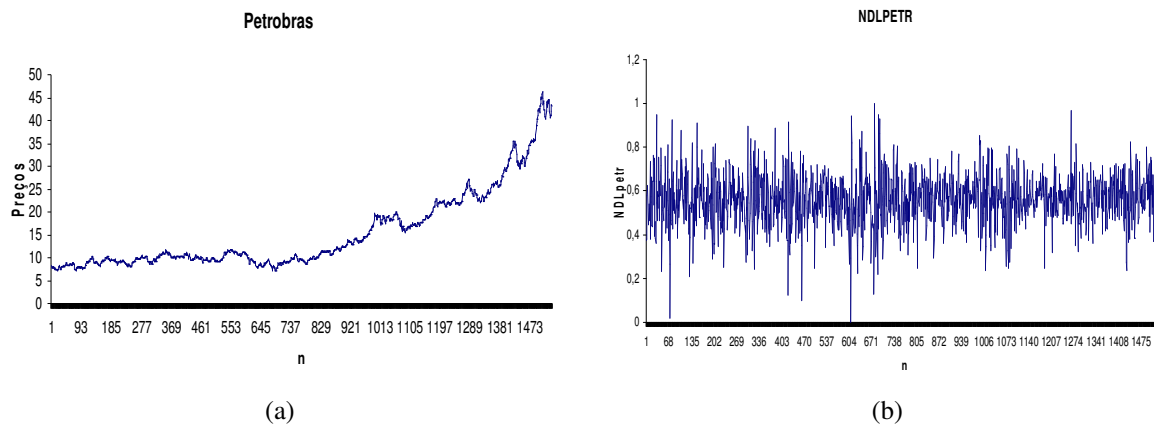


Figura 6.118 (a) Série no nível do preço do Bradesco, (b) Série transformada NDLPETR

O resultado do teste ADF, para a série original, é apresentado na Tabela 6.55.

Estatística do teste ADF para série original	valores críticos 1%	-3,43439
2,07273	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771

Tabela 6.55 Teste ADF para série original da Petrobras

Não é rejeitada a hipótese de raiz unitária para a série no nível pelo teste ADF. O teste de presença de uma raiz unitária para a série transformada apresentou os seguintes resultados:

Estatística do teste ADF para série transformada	valores críticos 1%	-3,43439
-27,4447	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771

Tabela 6.56 Teste ADF para série NDLPETR

Na Figura 6.119 é apresentada a estatística de Jarque-Bera, que rejeita fortemente a hipótese de distribuição normal.

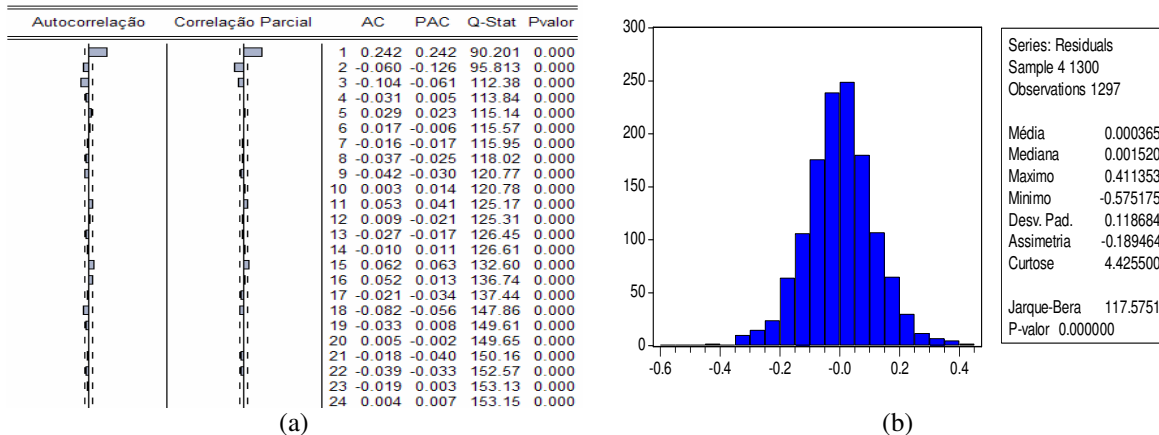


Figura 6.119 (a) Correlograma da série NDLPETR, (b) Estatísticas dos resíduos

Os resultados para o teste BDS são apresentados na Tabela 6.57.

BDS				
Dimensão	Estatística	Erro-padrão	Estatística-z	p-valor
2	0,016847	0,001937	8,696789	0,0000
3	0,026464	0,003065	8,632803	0,0000
4	0,032478	0,003634	8,936365	0,0000
5	0,035915	0,003771	9,523393	0,0000
6	0,037347	0,003621	10,31510	0,0000

Tabela 6.57 Resultados obtidos pelo teste BDS nas respectivas dimensões.

O melhor modelo ajustado para a série é um ARMA([2],2)-GARCH(1,1). Os parâmetros estimados do modelo para a série são mostrados na Tabela 6.58.

Parâmetros	Coefficiente	Erro-padrão	Estatística-z	p-valor
ϕ_0	0,562631	0,003536	159,1287	0,0000
ϕ_2	-0,225349	0,105865	-2,128656	0,0333
θ_1	0,248369	0,030252	8,210041	0,0000
θ_2	0,207638	0,103979	1,996921	0,0458
α_0	0,001173	0,000329	3,565910	0,0004
α_1	0,095697	0,018074	5,294652	0,0000
β_1	0,816016	0,039406	20,70802	0,0000

Tabela 6.58 Parâmetros estimados para a série NDLPETR

O resultado do teste Ljung-Box para a série NDLPETR é apresentado na Figura 6.120.

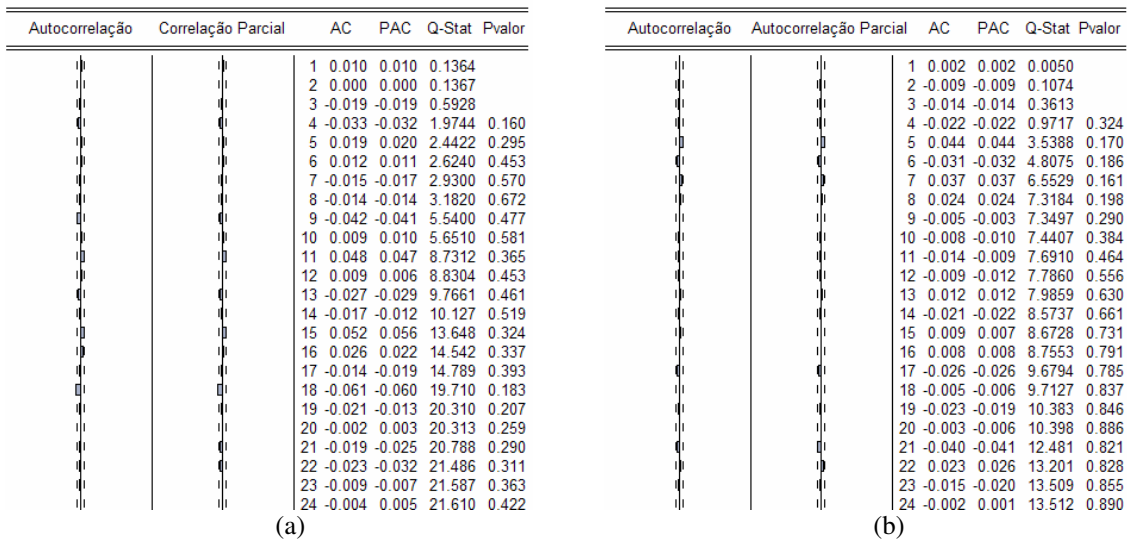


Figura 6.120 (a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado

As previsões realizadas utilizando o modelo ARIMA-GARCH são mostradas na Figura 6.121. A linha contínua indica o valor da série NDLPETR e a linha central com marcador quadrado indica as previsões.

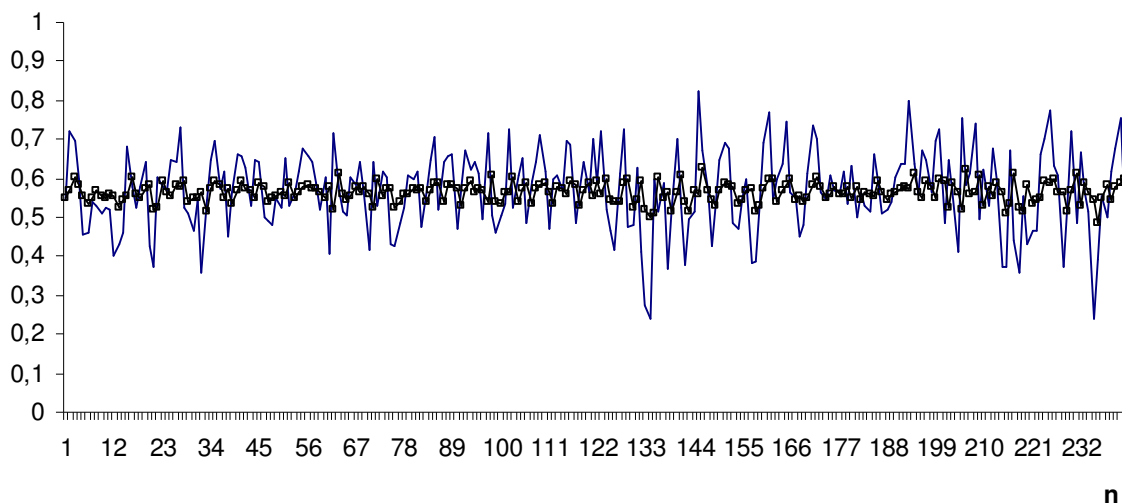


Figura 6.121 Previsão obtida pelo modelo ARIMA-GARCH para NDLPETR

Na Figura 6.122a é mostrada a previsão da volatilidade e na Figura 6.122b a série original e o intervalo de previsão para a série NDLPETR.

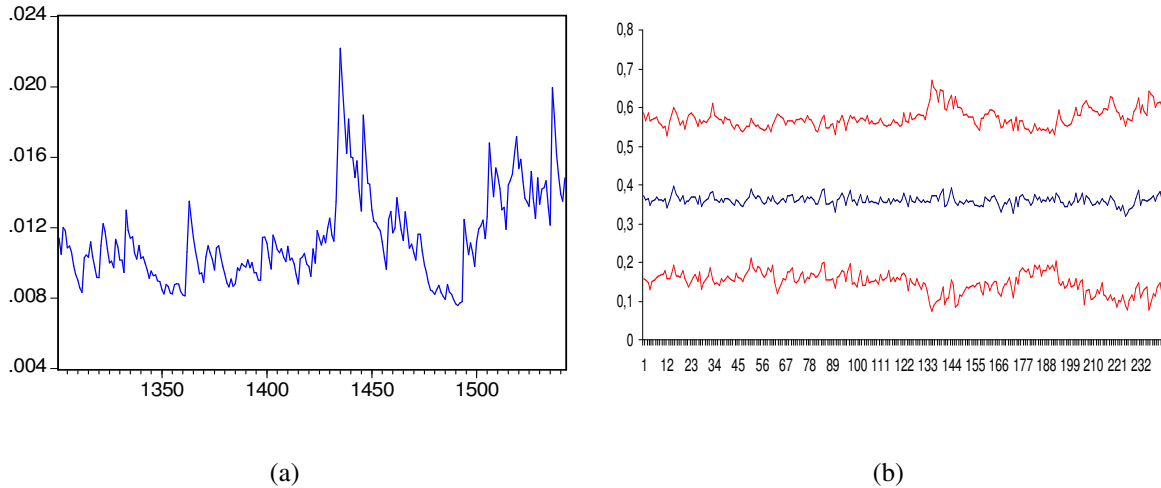


Figura 6.122 (a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLPETR

6.4.2.2 Análise da série da Petrobras– Método RNA *feedforward*

Para a série NDLPETR os resultados obtidos pelas diferentes topologias de RNA são apresentados na Tabela 6.59.

Topologia	FIC	FIP	TIC	RMSE
RNA 10-1-1	128	0	0,0873976	0,0994528
RNA 10-2-1	128	0	0,0872705	0,0992968
RNA 10-3-1	127	0	0,0904073	0,102675
RNA 10-4-1	120	0	0,0883269	0,100884
RNA 10-5-1	113	0	0,0895795	0,102175
RNA 10-8-1	123	0	0,0926664	0,105745

Tabela 6.59 Estatísticas para diferentes topologias de RNA para a série

Na Figura 6.123 são apresentadas as previsões obtidas pela RNA 10-2-1 para a série NDLPETR.

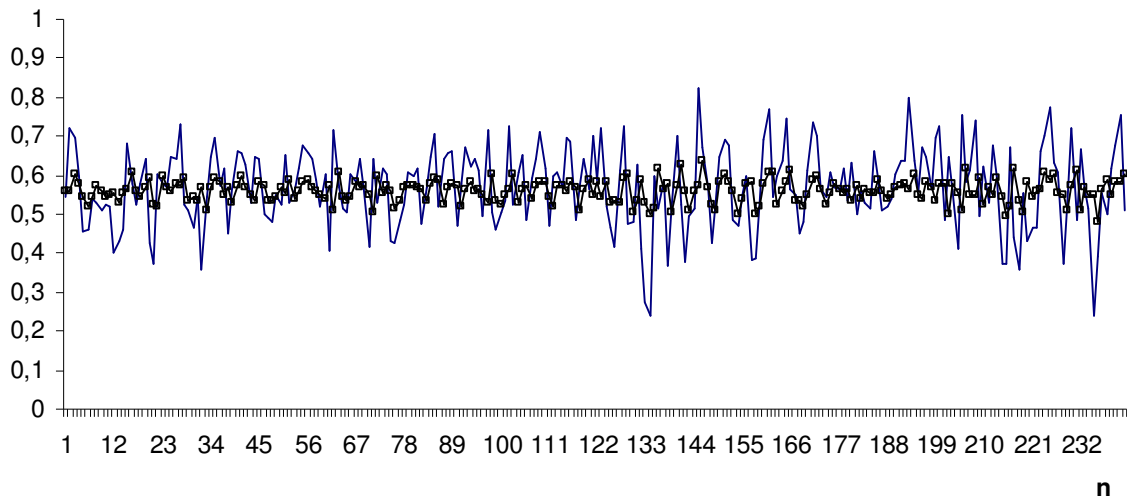


Figura 6.123 Previsão obtida pela RNA *feedforward* para NDLPETR

Na Figura 6.124a é mostrado o intervalo de confiança para as previsões obtidas pela RNA 10-2-1 e na Figura 6.124b o intervalo de predição para a série NDLPETR.

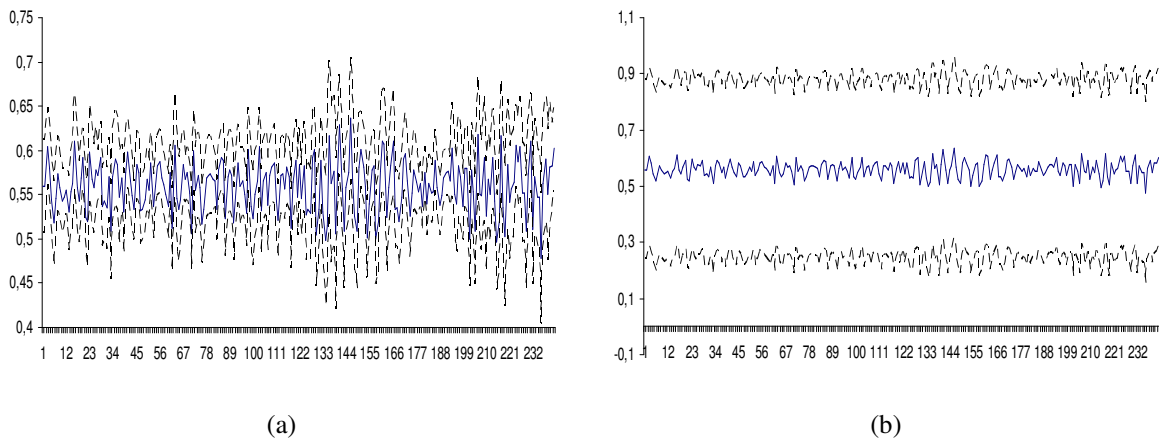


Figura 6.124 (a) Intervalo de confiança, (b) Intervalo de predição para NDLPETR

Na Figura 6.125a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA 10-2-1 e na Figura 6.125b o histograma desses resíduos.

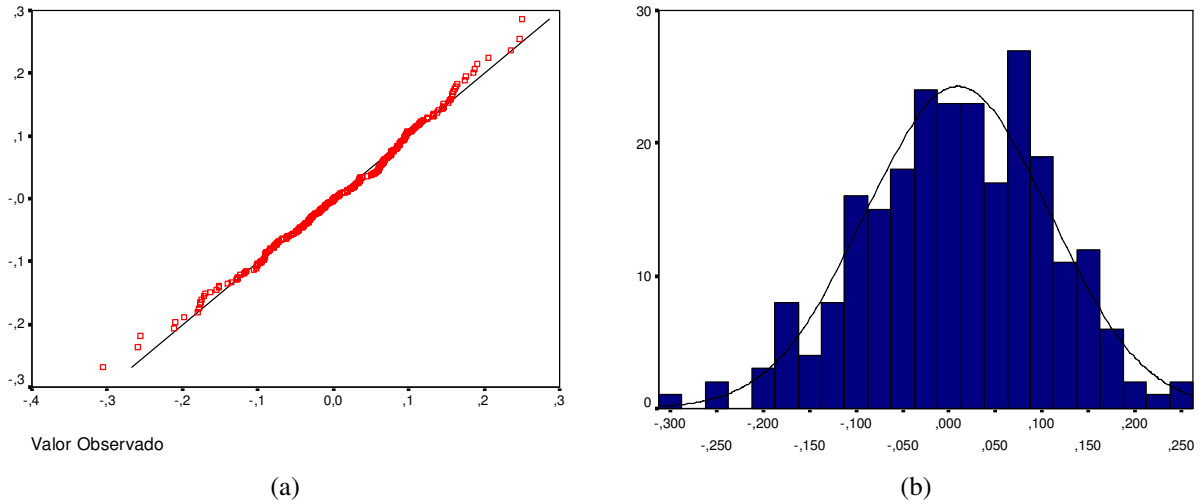


Figura 6.125 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA *feedforward* para NDLPETR

6.4.2.3 Análise da série da Petrobras– Método RNA EKF

As previsões, para a série NDLPETR, obtidas pela RNA treinada com o filtro de Kalman estendido (EKF) são mostradas na Figura 6.126.

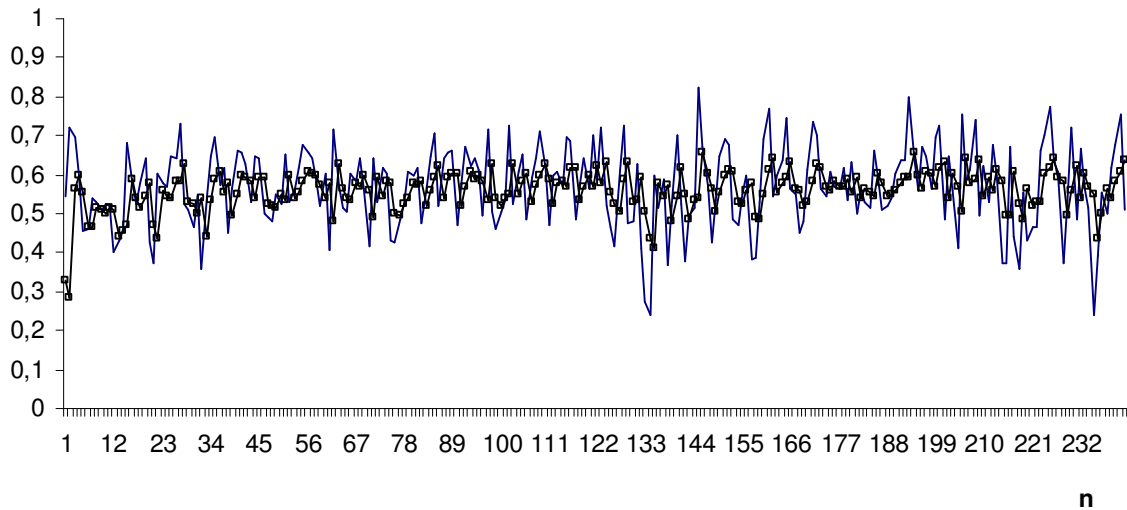


Figura 6.126 Previsão obtida pela RNA EKF para NDLPETR

Na Figura 6.127a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman estendido e na Figura 6.127b o histograma desses resíduos.

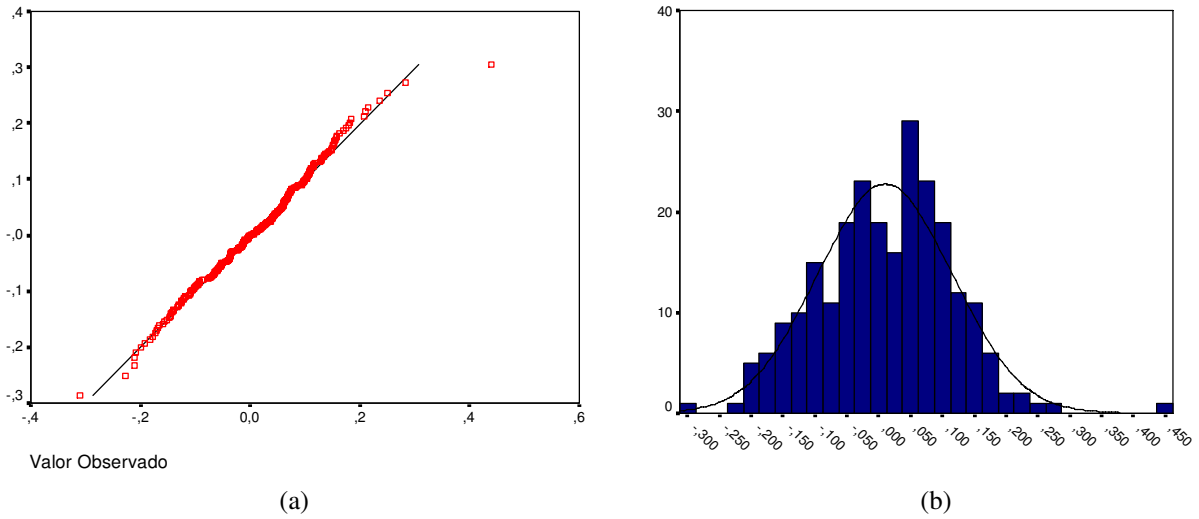


Figura 6.127 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLPETR

6.4.2.4 Análise da série da Petrobras– Método RNA UKF

As previsões obtidas pela RNA treinada com o filtro de Kalman *unscented* são mostradas na Figura 6.128.

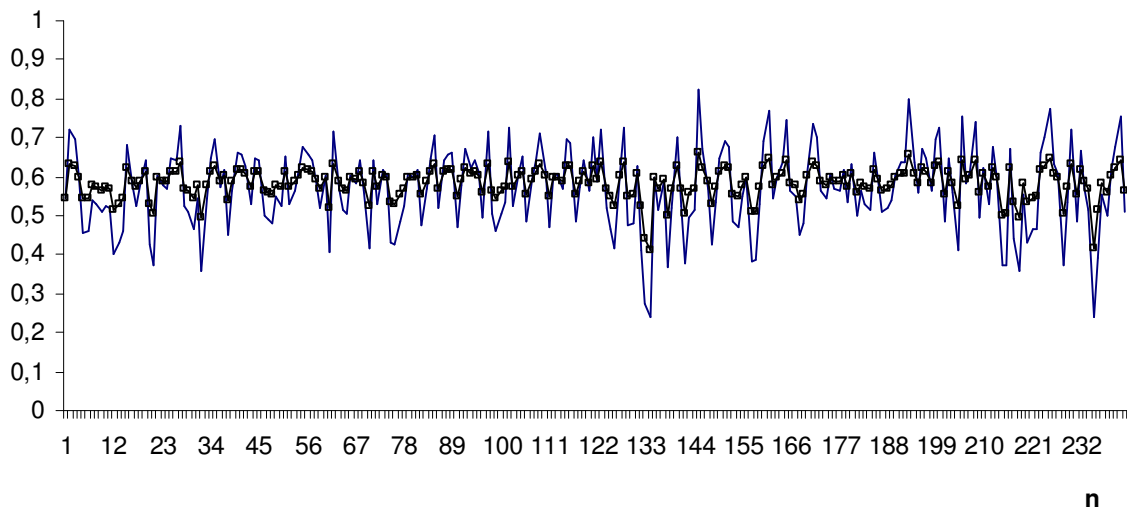


Figura 6.128 Previsão obtida pela RNA UKF para NDLPETR

Na Figura 6.129a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman *unscented* e na Figura 6.129b o histograma desses resíduos.

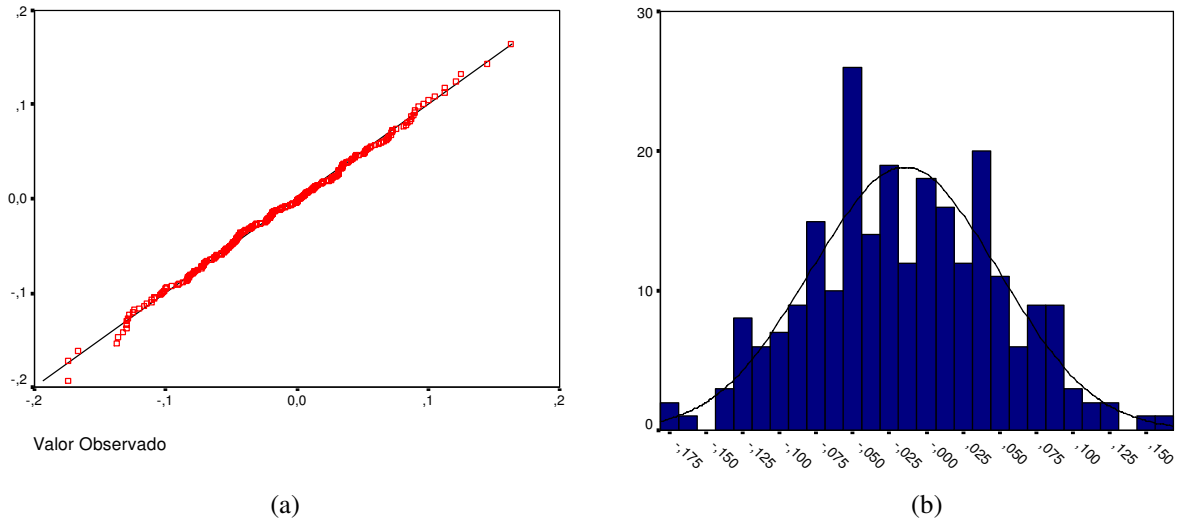


Figura 6.129 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLPETR

As estatísticas de previsão obtidas pelos modelos ARIMA-GARCH, RNA *feedforward*, RNA EKF e RNA UKF são apresentadas na Tabela 6.60 e na Figura 6.130.

Método	TIC	RMSE
ARMA([2],2)-GARCH(1,1)	0,086353	0,098616
RNA 10-2-1	0,087271	0,099297
RNA EKF	0,093159	0,106081
RNA UKF	0,056308	0,065454

Tabela 6.60 Estatísticas de previsão para o método de ARIMA-GARCH e de RNA

Pode-se verificar que as melhores previsões foram obtidas pelo método que empregou redes neurais treinadas com o filtro de Kalman *unscented*.

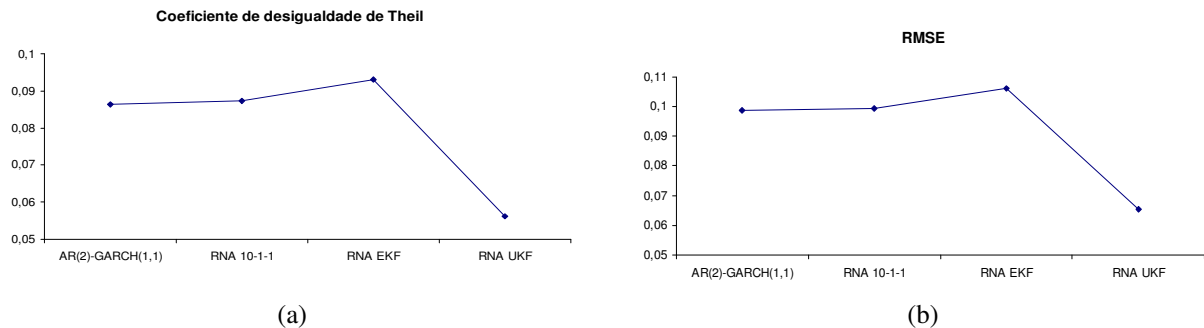


Figura 6.130 Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b) RMSE

6.4.3 Análise da série da Embraer

A seguir são apresentados os resultados obtidos para a série da Embraer.

6.4.3.1 Análise da série da Embraer – Método ARIMA-GARCH

Na Figura 6.131a é apresentada a série original do preço de fechamento da Embraer, na Figura 6.131b é apresenta a série normalizada da primeira diferença do logaritmo do preço, NDLEMBR.

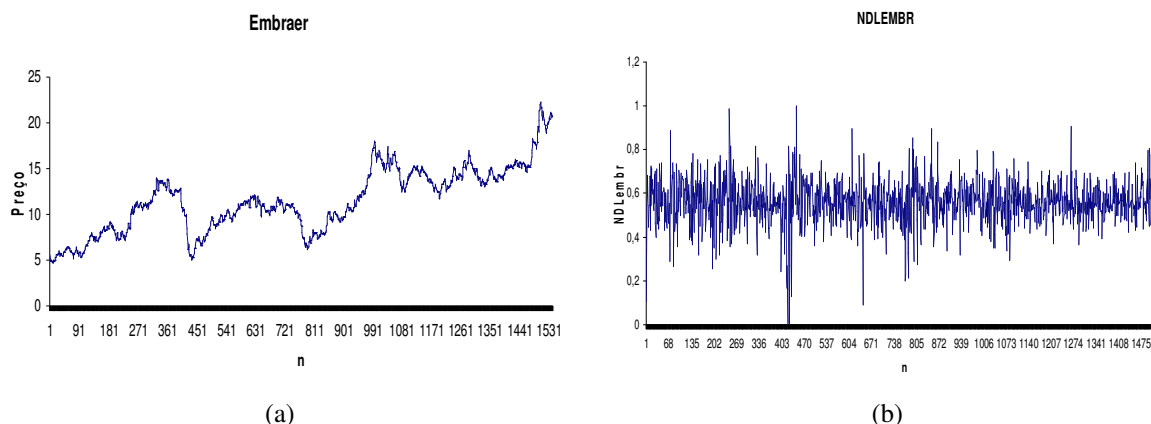


Figura 6.131 (a) Série no nível do preço do Bradesco, (b) Série transformada NDLEMBR

O resultado do teste ADF, para a série original, é apresentado na Tabela 6.61.

Estatística do teste ADF para série original	valores críticos 1%	-3,43439
-1,10833	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771

Tabela 6.61 Teste ADF para série original da Embraer

Não é rejeitada a hipótese de raiz unitária para a série no nível pelo teste ADF. O teste de presença de uma raiz unitária para a série transformada apresentou os seguintes resultados:

Estatística do teste ADF para série transformada	valores críticos 1%	-3,43439
-27,5092	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771

Tabela 6.62 Teste ADF para série NDLEMBR

Na Figura 6.132 é apresentada a estatística de Jarque-Bera, que rejeita fortemente a hipótese de distribuição normal.

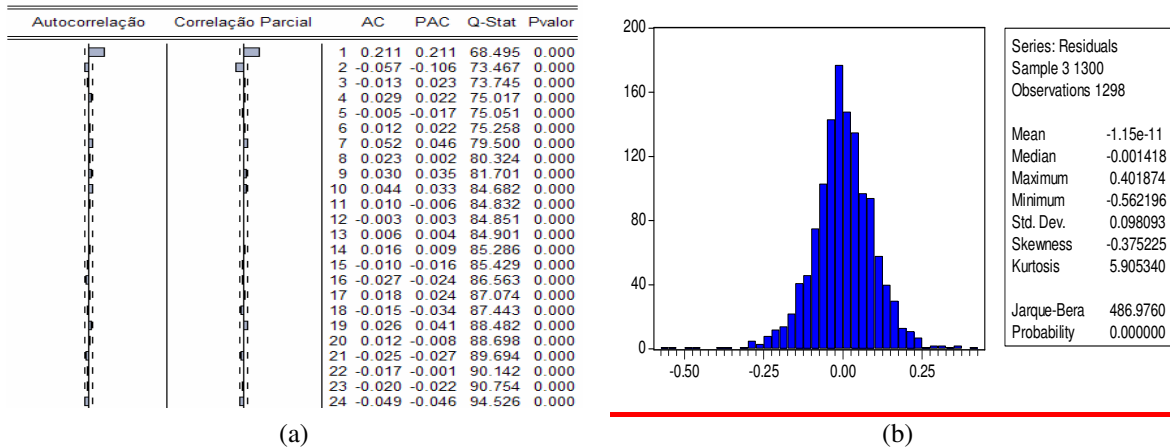


Figura 6.132 (a) Correlograma da série NDLEMBR, (b) Estatísticas dos resíduos

Os resultados para o teste BDS são apresentados na Tabela 6.63.

BDS				
Dimensão	Estatística	Erro-padrão	Estatística-z	p-valor
2	0,019002	0,002102	9,040253	0,0000
3	0,032013	0,003341	9,581217	0,0000
4	0,039726	0,003980	9,982406	0,0000
5	0,043678	0,004149	10,52848	0,0000
6	0,043916	0,004001	10,97516	0,0000

Tabela 6.63 Resultados obtidos pelo teste BDS nas respectivas dimensões.

O melhor modelo ajustado para a série é um AR(2)-GARCH(1,1). Os parâmetros estimados do modelo para a série são mostrados na Tabela 6.64.

Parâmetros	Coefficiente	Erro-padrão	Estatística-z	p-valor
ϕ_0	0,560709	0,002855	196,4279	0,0000
ϕ_1	0,205746	0,027666	7,436648	0,0000
ϕ_2	-0,089933	0,031368	-2,866991	0,0041
α_0	0,000713	0,000148	4,799765	0,0000
α_1	0,099315	0,016139	6,153566	0,0000
β_1	0,825302	0,025350	32,55611	0,0000

Tabela 6.64 Parâmetros estimados para a série NDLEMBR

O resultado do teste Ljung-Box para a série NDLEMBR é apresentado na Figura 6.133.

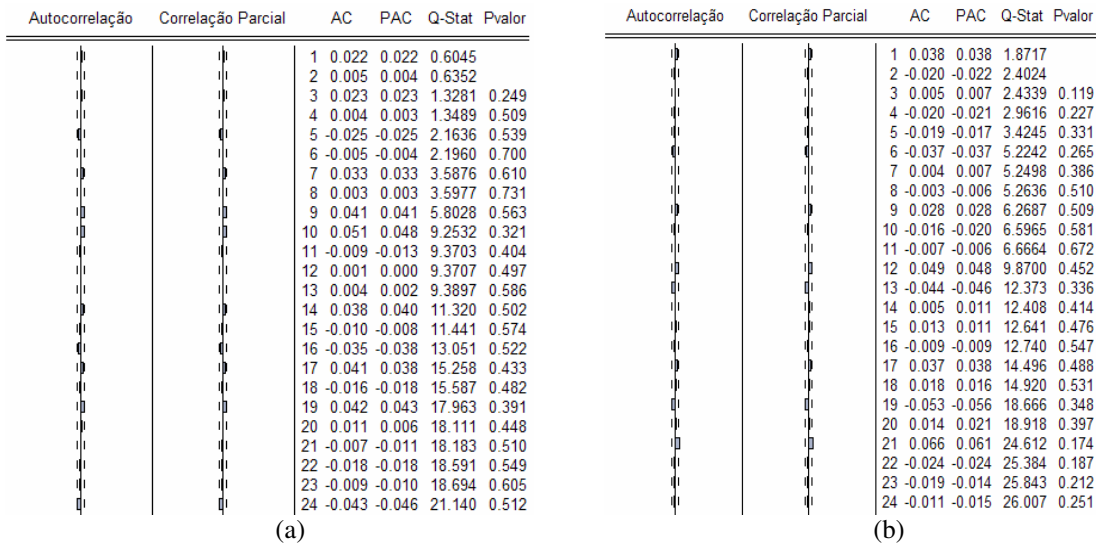


Figura 6.133 (a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado

As previsões realizadas utilizando o modelo ARIMA-GARCH são mostradas na Figura 6.134. A linha contínua indica o valor da série NDLEMBR e a linha central com marcador quadrado indica as previsões.

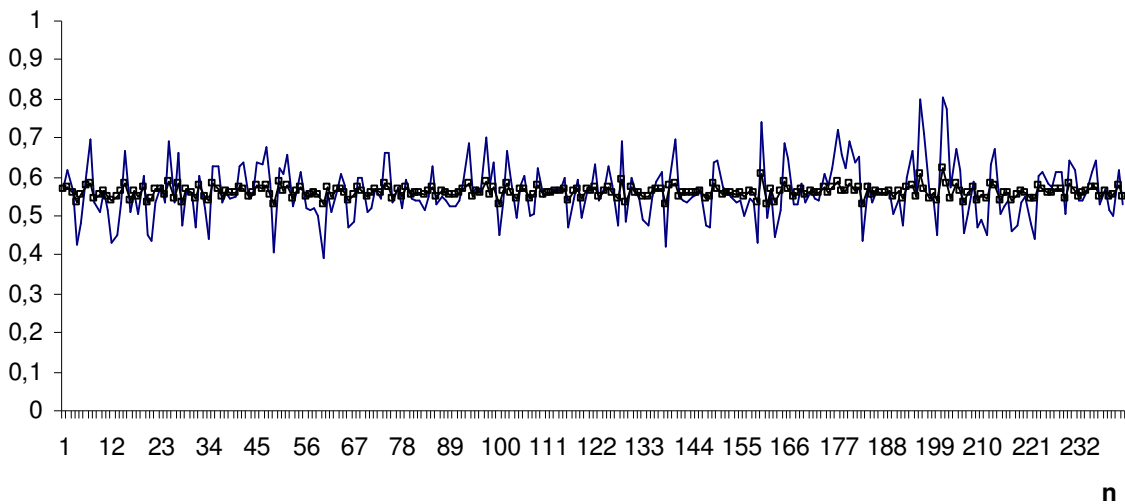


Figura 6.134 Previsão obtida pelo modelo ARIMA-GARCH para NDLEMBR

Na Figura 6.135a é mostrada a previsão da volatilidade e na Figura 6.135b a série original e o intervalo de previsão para a série NDLEMBR.

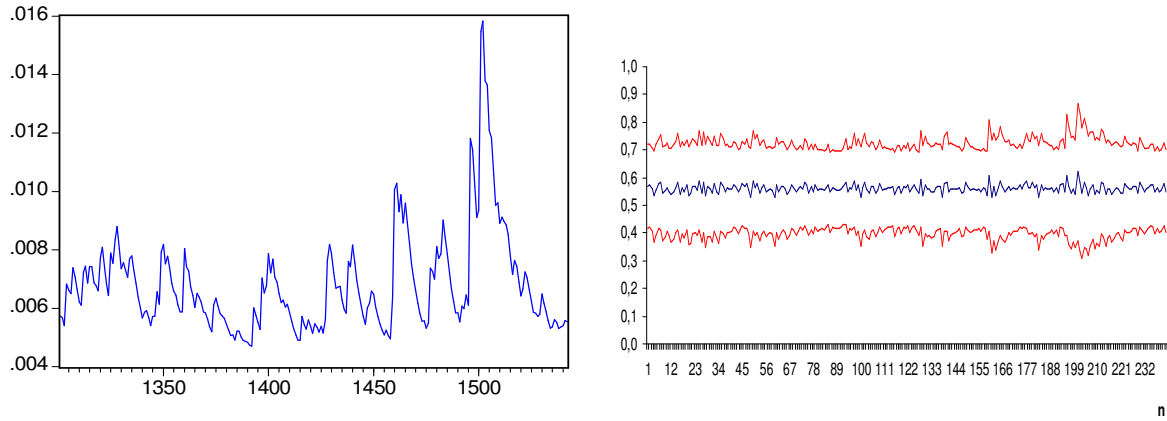


Figura 6.135 (a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLEMBR

6.4.3.2 Análise da série da Embraer – Método RNA *feedforward*

Para a série NDLEMBR os resultados obtidos pelas diferentes topologias de RNA são apresentados na Tabela 6.65.

Topologia	FIC	FIP	TIC	RMSE
RNA 10-1-1	120	0	0,06225	0,0702507
RNA 10-2-1	120	0	0,0622938	0,0703796
RNA 10-3-1	125	0	0,0612281	0,0691871
RNA 10-4-1	113	0	0,0635915	0,0716501
RNA 10-5-1	117	0	0,0619443	0,0696899
RNA 10-8-1	114	0	0,063537	0,0716325

Tabela 6.65 Estatísticas para diferentes topologias de RNA para a série

Na Figura 6.136 são apresentadas as previsões obtidas pela RNA 10-3-1 para a série NDLEMBR.

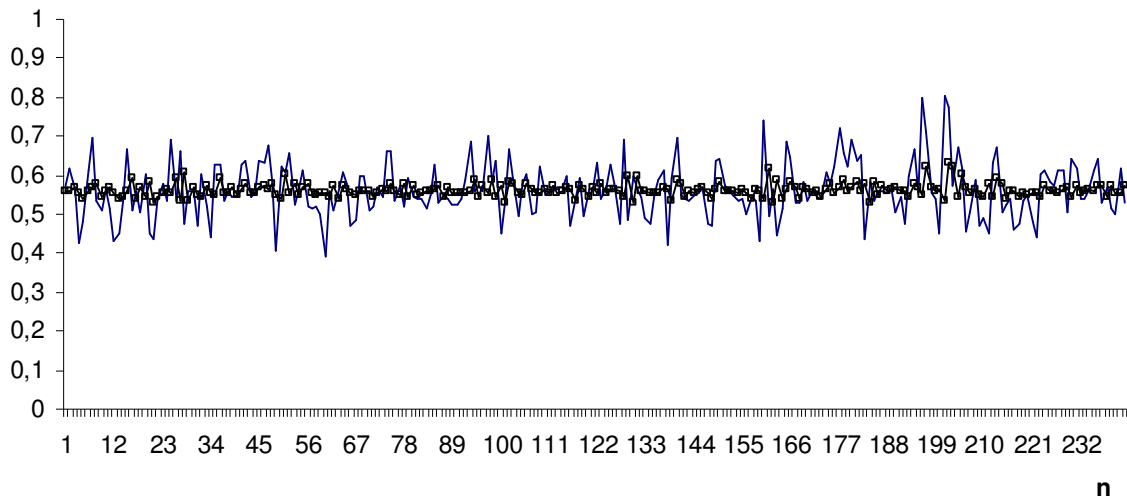


Figura 6.136 Previsão obtida pela RNA *feedforward* para NDLEMBR

Na Figura 6.137a é mostrado o intervalo de confiança para as previsões obtidas pela RNA 10-3-1 e na Figura 6.137b o intervalo de predição para a série NDLEMBR.

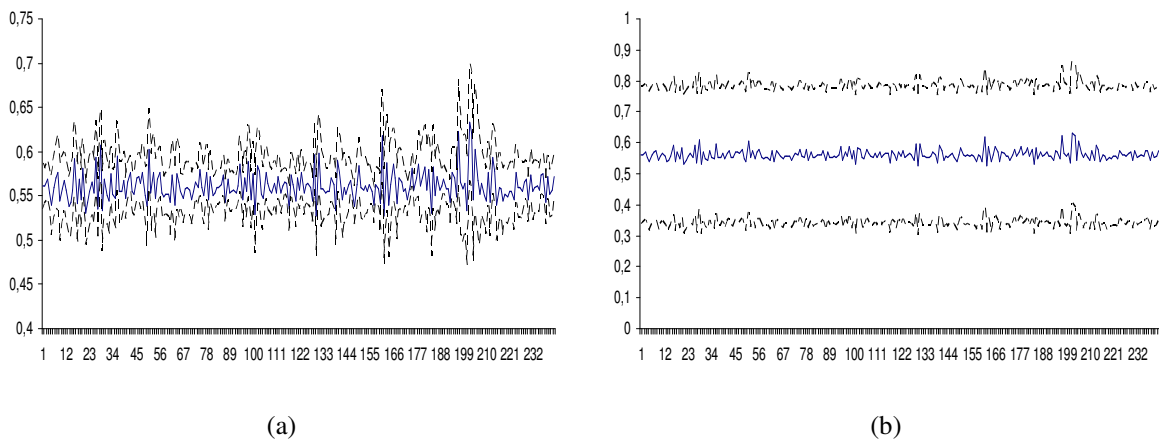


Figura 6.137 (a) Intervalo de confiança, (b) Intervalo de predição para NDLEMBR

Na Figura 6.138a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA 10-3-1 e na Figura 6.138b o histograma desses resíduos.

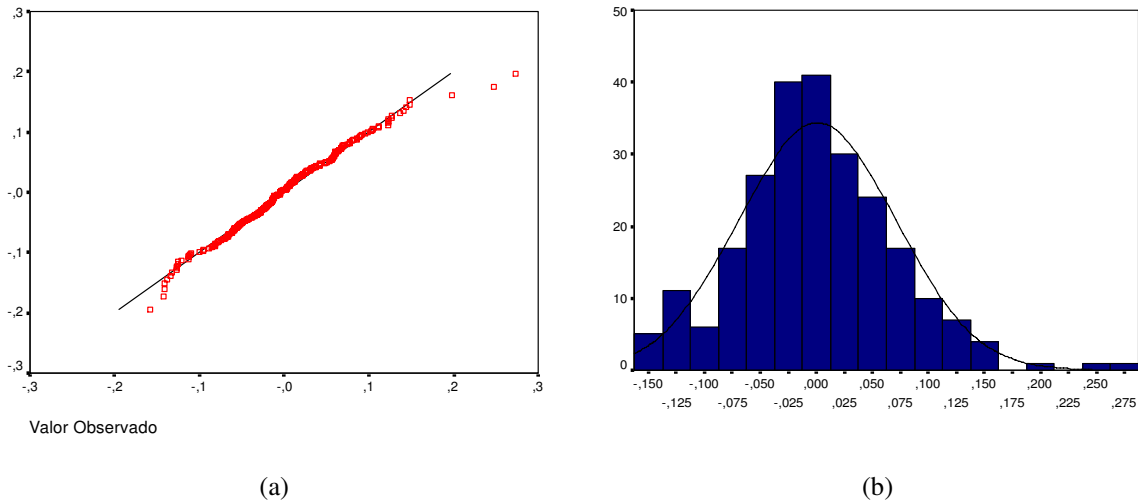


Figura 6.138 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA *feedforward* para NDLEMBR

6.4.3.3 Análise da série da Embraer – Método RNA EKF

As previsões, para a série NDLEMBR, obtidas pela RNA treinada com o filtro de Kalman estendido (EKF) são mostradas na Figura 6.139.

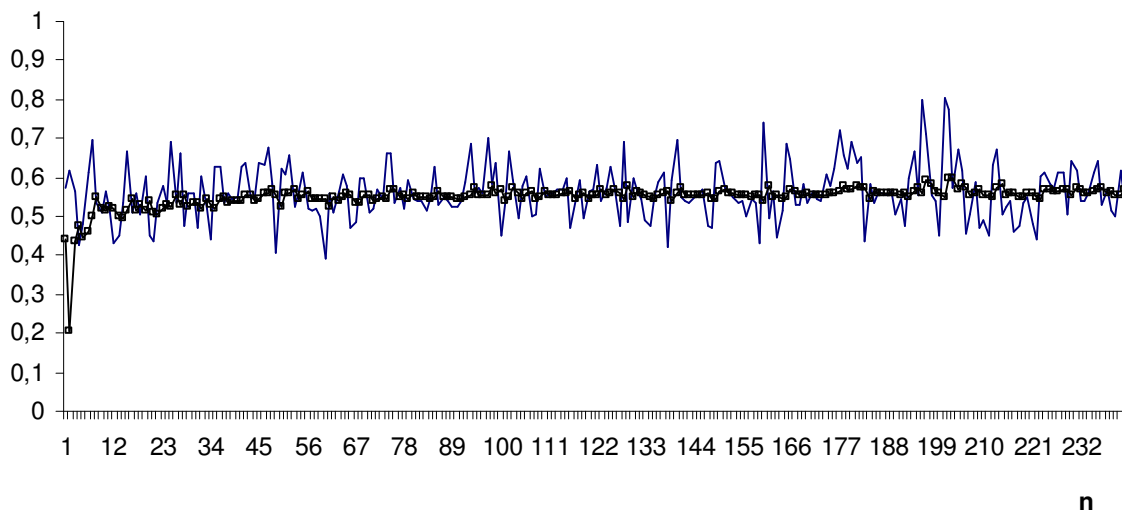


Figura 6.139 Previsão obtida pela RNA EKF para NDLEMBR

Na Figura 6.140a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman estendido e na Figura 6.140b o histograma desses resíduos.

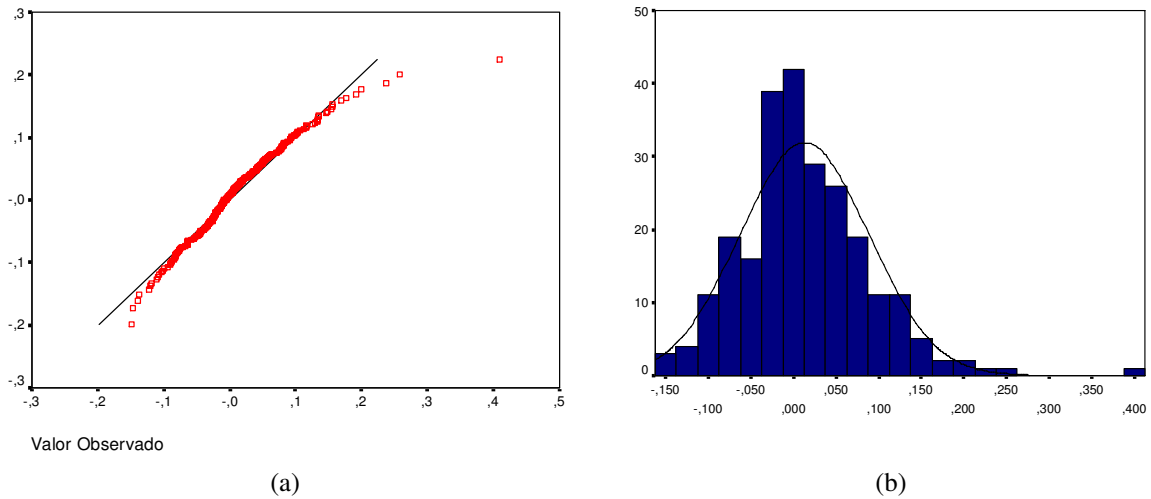


Figura 6.140 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLEMBR

6.4.3.4 Análise da série da Embraer – Método RNA UKF

As previsões obtidas pela RNA treinada com o filtro de Kalman unscented são mostradas na Figura 6.141.

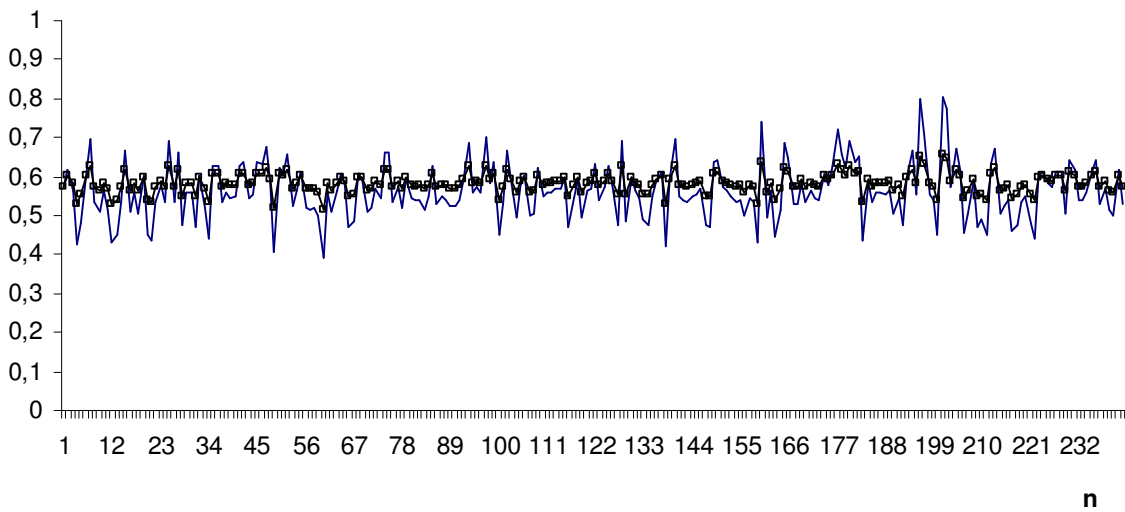


Figura 6.141 Previsão obtida pela RNA UKF para NDLEMBR

Na Figura 6.142a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman *unscented* e na Figura 6.142b o histograma desses resíduos.

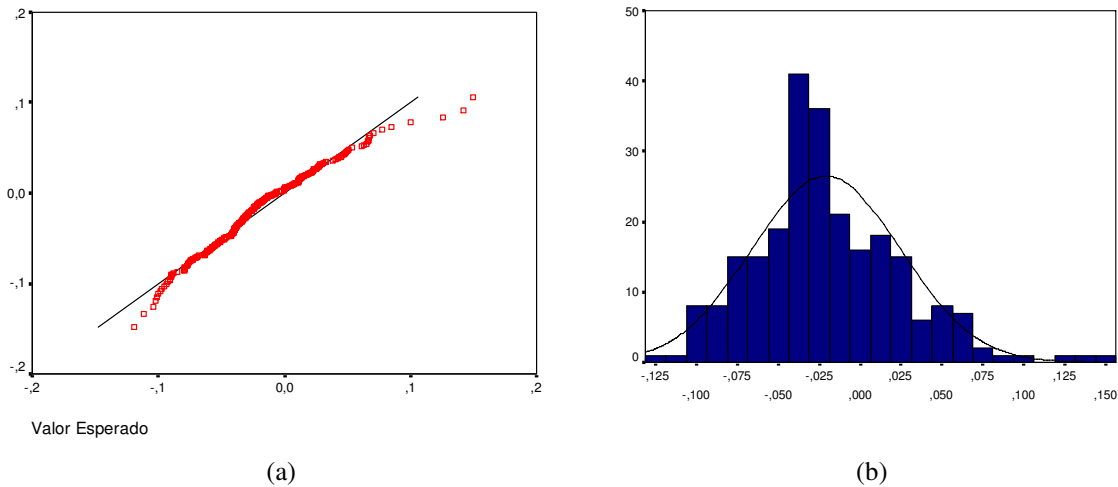


Figura 6.142 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLEMBR

As estatísticas de previsão obtidas pelos modelos ARIMA-GARCH, RNA *feedforward*, RNA EKF e RNA UKF são apresentadas na Tabela 6.66 e na Figura 6.143.

Método	TIC	RMSE
AR(2)-GARCH(1,1)	0,061654	0,069521
RNA 10-3-1	0,061228	0,069187
RNA EKF	0,083861	0,942240
RNA UKF	0,043441	0,049959

Tabela 6.66 Estatísticas de previsão para o método de ARIMA-GARCH e de RNA

Pode-se verificar que as melhores previsões foram obtidas pelo método que empregou redes neurais treinadas com o filtro de Kalman *unscented*.

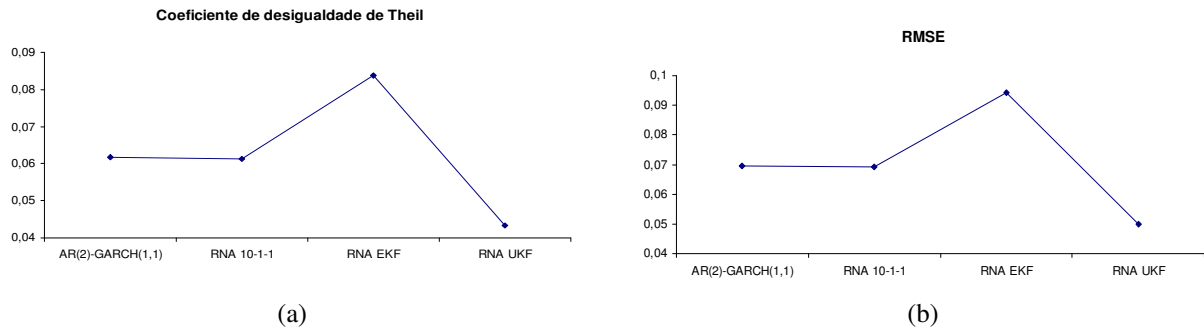


Figura 6.143 Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b) RMSE

6.4.4 Análise da série da Ripasa

A seguir são apresentados os resultados obtidos para a série da Ripasa.

6.4.4.1 Análise da série da Ripasa – Método ARIMA-GARCH

Na Figura 6.144a é apresentada a série original do preço de fechamento da Ripasa, na Figura 6.144b é apresentada a série normalizada da primeira diferença do logaritmo do preço, NDLRIPA.

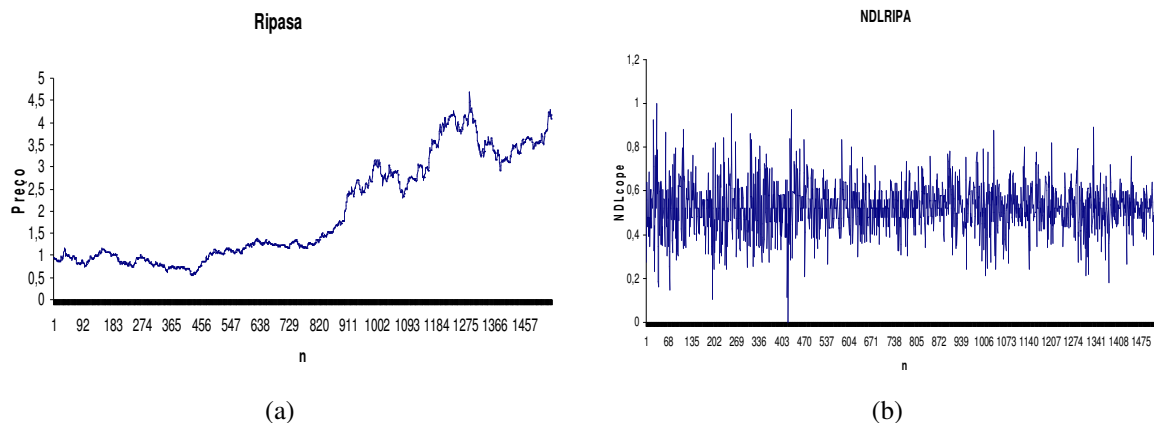


Figura 6.144 (a) Série no nível do preço do Bradesco, (b) Série transformada NDLRIPA

O resultado do teste ADF, para a série original, é apresentado na Tabela 6.67.

Estatística do teste ADF para série original	valores críticos 1%	-3,43439
0,029212	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771

Tabela 6.67 Teste ADF para série original da Ripasa

Não é rejeitada a hipótese de raiz unitária para a série no nível pelo teste ADF. O teste de presença de uma raiz unitária para a série transformada apresentou os seguintes resultados:

Estatística do teste ADF para série transformada	valores críticos 1%	-3,43439
-35,49343	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771

Tabela 6.68 Teste ADF para série NDLRIPA

Na Figura 6.145 é apresentada a estatística de Jarque-Bera, que rejeita fortemente a hipótese de distribuição normal.

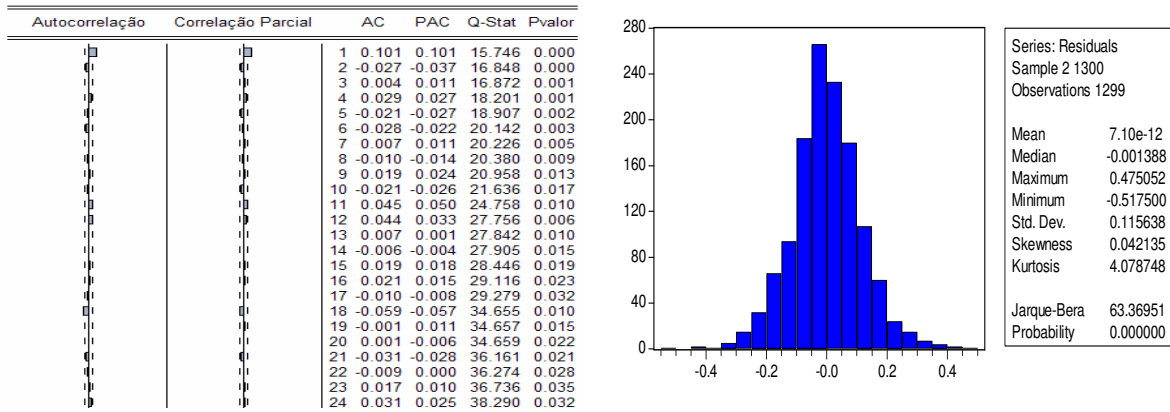


Figura 6.146 (a) Correlograma da série NDLRIPA, (b) Estatísticas dos resíduos

Os resultados para o teste BDS são apresentados na Tabela 6.69.

BDS				
Dimensão	Estatística	Erro-padrão	Estatística-z	p-valor
2	0,011725	0,002052	5,714177	0,0000
3	0,026605	0,003248	8,191594	0,0000
4	0,037826	0,003852	9,820544	0,0000
5	0,043236	0,003998	10,81466	0,0000
6	0,045432	0,003839	11,83304	0,0000

Tabela 6.69 Resultados obtidos pelo teste BDS nas respectivas dimensões.

O melhor modelo ajustado para a série é um AR(1)-GARCH(1,1). Os parâmetros estimados do modelo para a série são mostrados na Tabela 6.70.

Parâmetros	Coefficiente	Erro-padrão	Estatística-z	p-valor
ϕ_0	0,526551	0,003319	158,6668	0,0000
ϕ_1	0,124693	0,029453	4,233621	0,0000
α_0	0,000758	0,000227	3,336120	0,0008
α_1	0,127002	0,020133	6,308306	0,0000
β_1	0,820159	0,029686	27,62774	0,0000

Tabela 6.70 Parâmetros estimados para a série NDLRIPA

O resultado do teste Ljung-Box para a série NDLRIPA é apresentado na Figura 6.146.

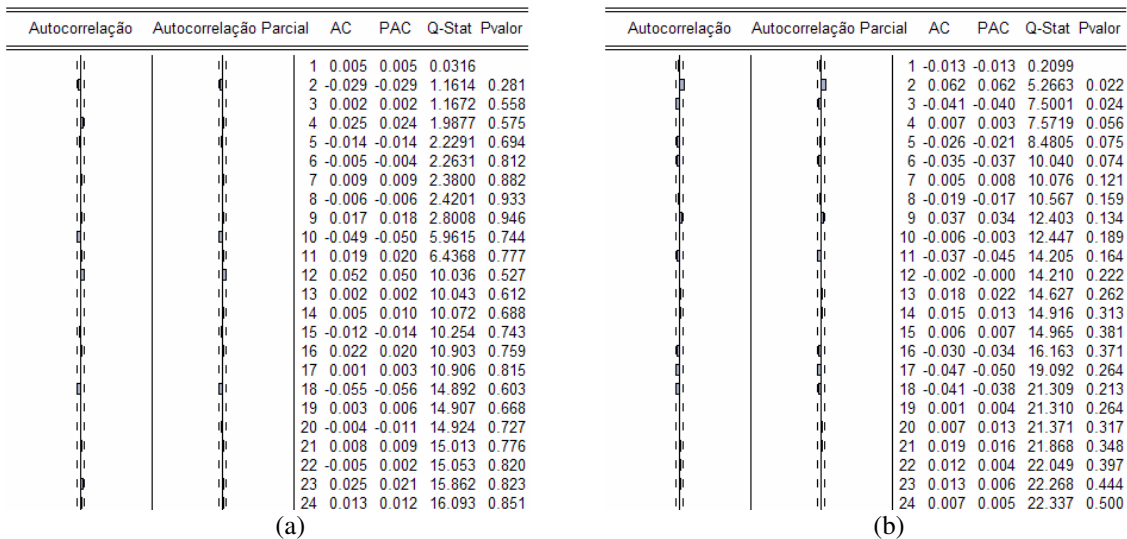


Figura 6.146 (a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado

As previsões realizadas utilizando o modelo ARIMA-GARCH são mostradas na Figura 6.147. A linha contínua indica o valor da série NDLRIPA e a linha central com marcador quadrado indica as previsões.

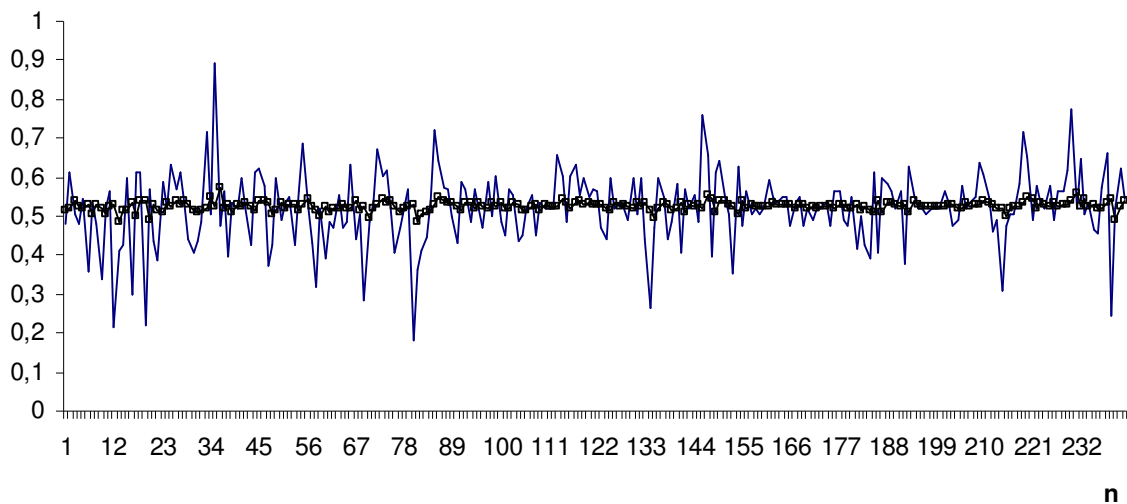


Figura 6.147 Previsão obtida pelo modelo ARIMA-GARCH para NDLRIPA

Na Figura 6.148a é mostrada a previsão da volatilidade e na Figura 6.148b a série original e o intervalo de previsão para a série NDLRIPA.

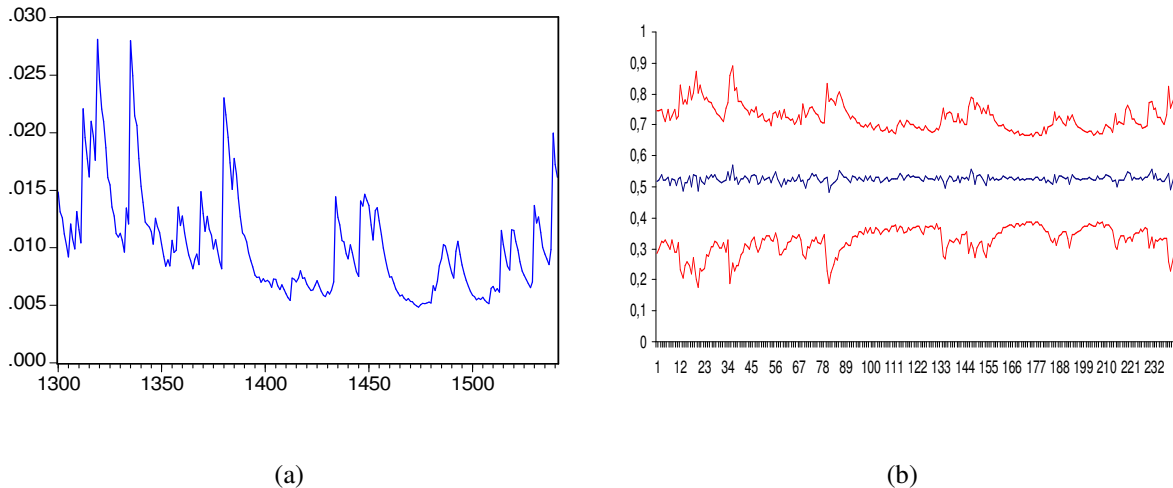


Figura 6.148 (a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLRIPA

6.4.4.2 Análise da série da Ripasa – Método RNA *feedforward*

Para a série NDLRIPA os resultados obtidos pelas diferentes topologias de RNA são apresentados na Tabela 6.71.

Topologia	FIC	FIP	TIC	RMSE
RNA 10-1-1	113	3	0,0876881	0,0925127
RNA 10-2-1	108	0	0,0898482	0,0946801
RNA 10-3-1	104	1	0,0886981	0,0936388
RNA 10-4-1	96	0	0,0911516	0,0958087
RNA 10-5-1	83	0	0,0910662	0,0965057
RNA 10-8-1	92	0	0,100409	0,106718

Tabela 6.71 Estatísticas para diferentes topologias de RNA para a série

Na Figura 6.149 são apresentadas as previsões obtidas pela RNA 10-1-1 para a série NDLRIPA.

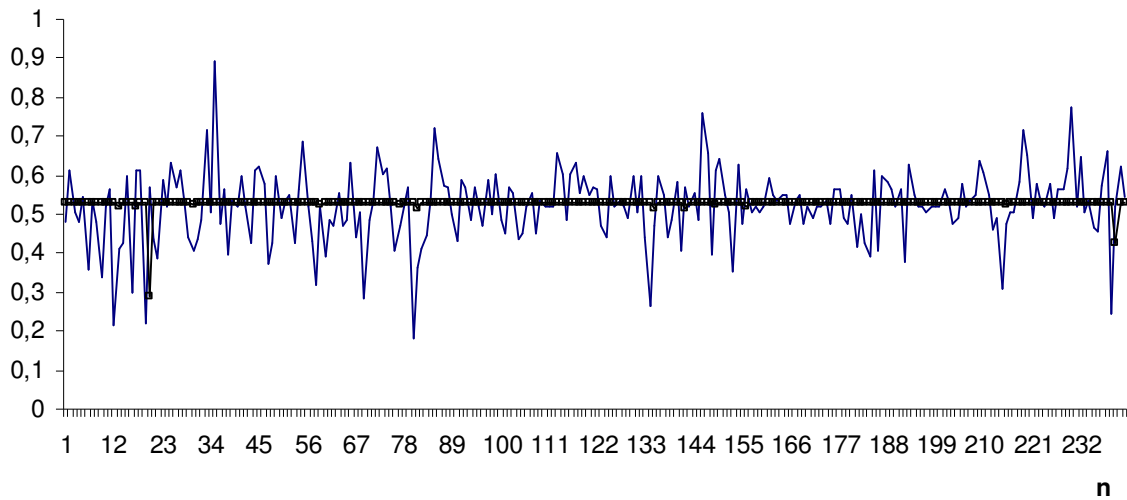


Figura 6.149 Previsão obtida pela RNA *feedforward* para NDLRIPA

Na Figura 6.150a é mostrado o intervalo de confiança para as previsões obtidas pela RNA 10-1-1 e na Figura 6.150b o intervalo de predição para a série NDLRIPA.

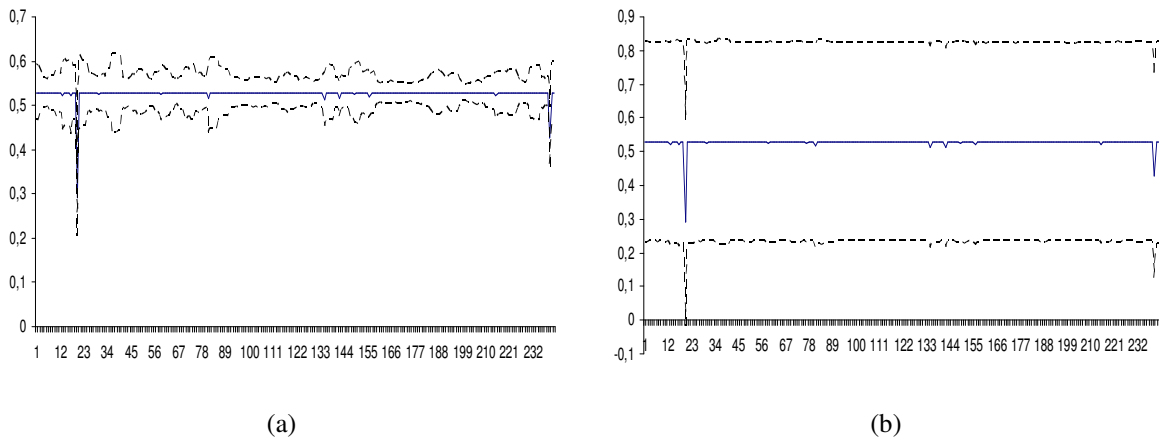


Figura 6.150 (a) Intervalo de confiança, (b) Intervalo de predição para NDLRIPA

Na Figura 6.151a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA 10-1-1 e na Figura 6.151b o histograma desses resíduos.

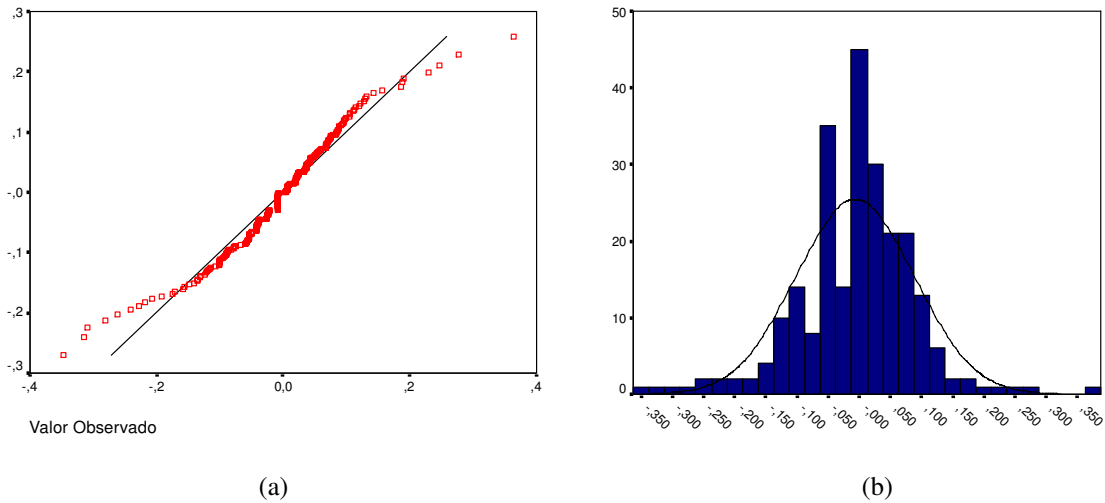


Figura 6.151 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA *feedforward* para NDLRIPA

6.4.4.3 Análise da série da Ripasa – Método RNA EKF

As previsões, para a série NDLRIPA, obtidas pela RNA treinada com o filtro de Kalman estendido (EKF) são mostradas na Figura 6.152.

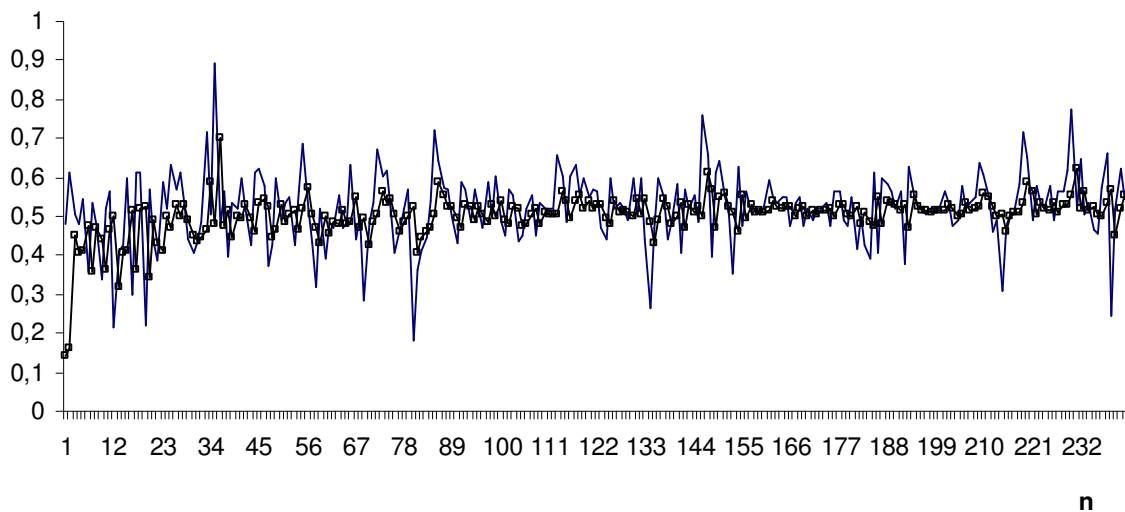


Figura 6.152 Previsão obtida pela RNA EKF para NDLRIPA

Na Figura 6.153a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman estendido e na Figura 6.153b o histograma desses resíduos.

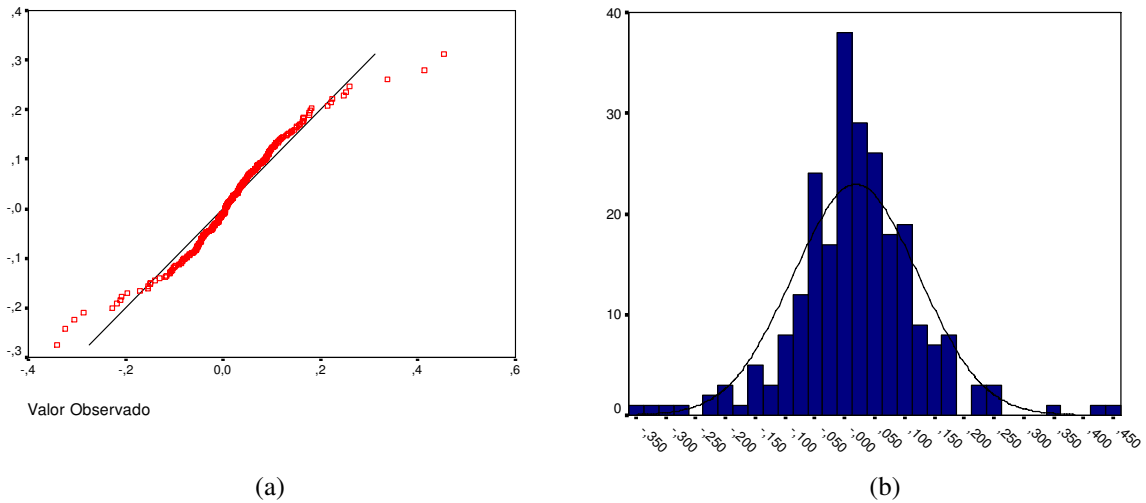


Figura 6.153 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLRIPA

6.4.4.4 Análise da série da Ripasa – Método RNA UKF

As previsões obtidas pela RNA treinada com o filtro de Kalman unscented são mostradas na Figura 6.154.

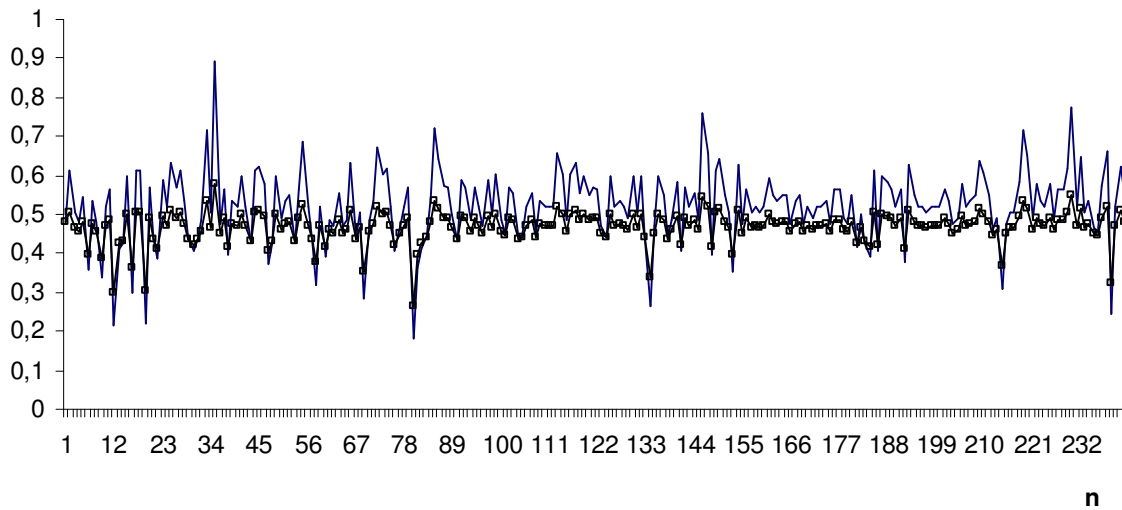


Figura 6.154 Previsão obtida pela RNA UKF para NDLRIPA

Na Figura 6.155a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman *unscented* e na Figura 6.155b o histograma desses resíduos.

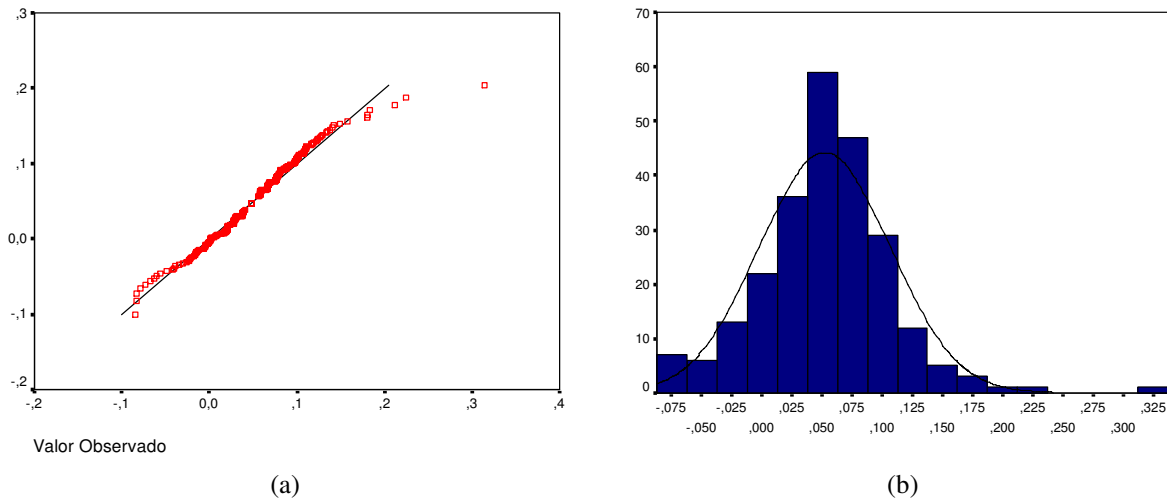


Figura 6.155 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLRIPA

As estatísticas de previsão obtidas pelos modelos ARIMA-GARCH, RNA *feedforward*, RNA EKF e RNA UKF são apresentadas na Tabela 6.72 e na Figura 6.156.

Método	TIC	RMSE
AR(1)-GARCH(1,1)	0,087977	0,092753
RNA 10-1-1	0,087688	0,092513
RNA EKF	0,103092	0,106514
RNA UKF	0,075393	0,075269

Tabela 6.72 Estatísticas de previsão para o método de ARIMA-GARCH e de RNA

Pode-se verificar que as melhores previsões foram obtidas pelo método que empregou redes neurais treinadas com o filtro de Kalman *unscented*.

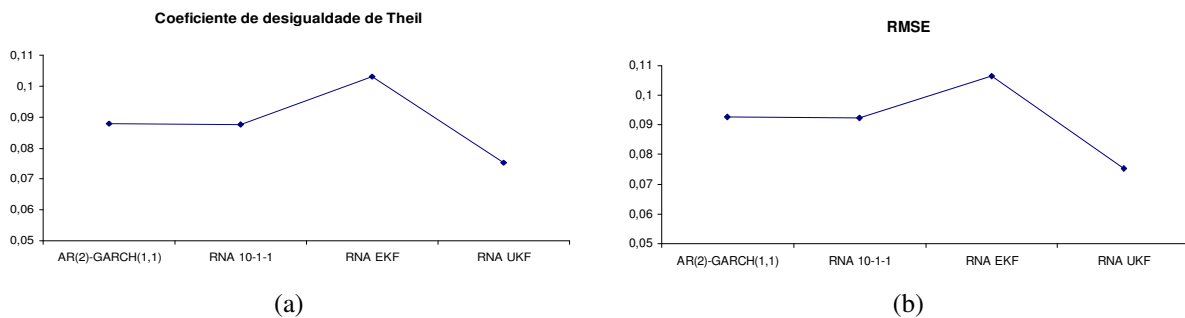


Figura 6.156 Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b) RMSE

6.4.5 Análise da série da Souza Cruz

A seguir são apresentados os resultados obtidos para a série da Souza Cruz.

6.4.5.1 Análise da série da Souza Cruz – Método ARIMA-GARCH

Na Figura 6.157a é apresentada a série original do preço de fechamento da Souza Cruz, na Figura 6.157b é apresentada a série normalizada da primeira diferença do logaritmo do preço, NDLSOUZ.

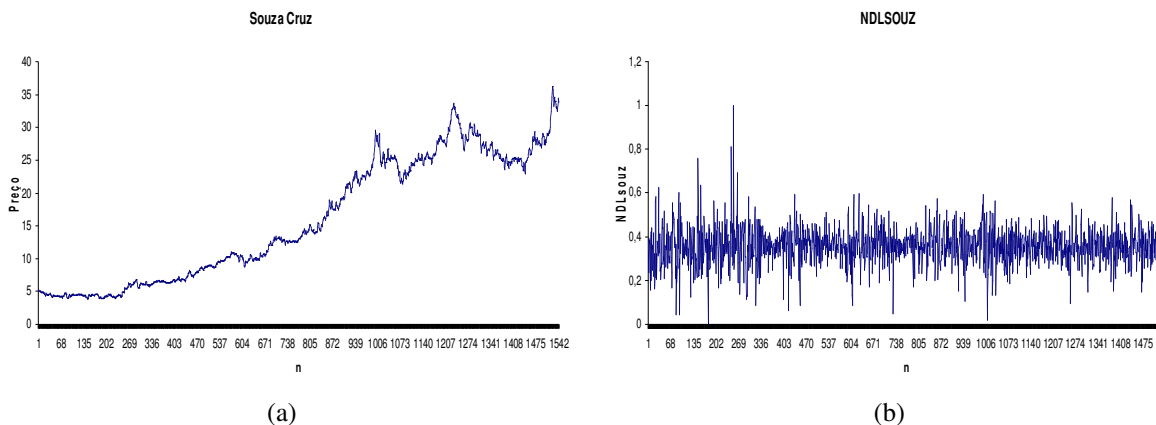


Figura 6.157 (a) Série no nível do preço do Bradesco, (b) Série transformada NDLSOUZ

O resultado do teste ADF, para a série original, é apresentado na Tabela 6.73.

Estatística do teste ADF para série original	valores críticos 1%	-3,43439
-0,020157	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771

Tabela 6.73 Teste ADF para série original da Souza Cruz

Não é rejeitada a hipótese de raiz unitária para a série no nível pelo teste ADF. O teste de presença de uma raiz unitária para a série transformada apresentou os seguintes resultados:

Estatística do teste ADF para série transformada	valores críticos 1%	-3,43439
-28,33388	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771

Tabela 6.74 Teste ADF para série NDLSOUZ

Na Figura 6.158 é apresentada a estatística de Jarque-Bera, que rejeita fortemente a hipótese de distribuição normal.

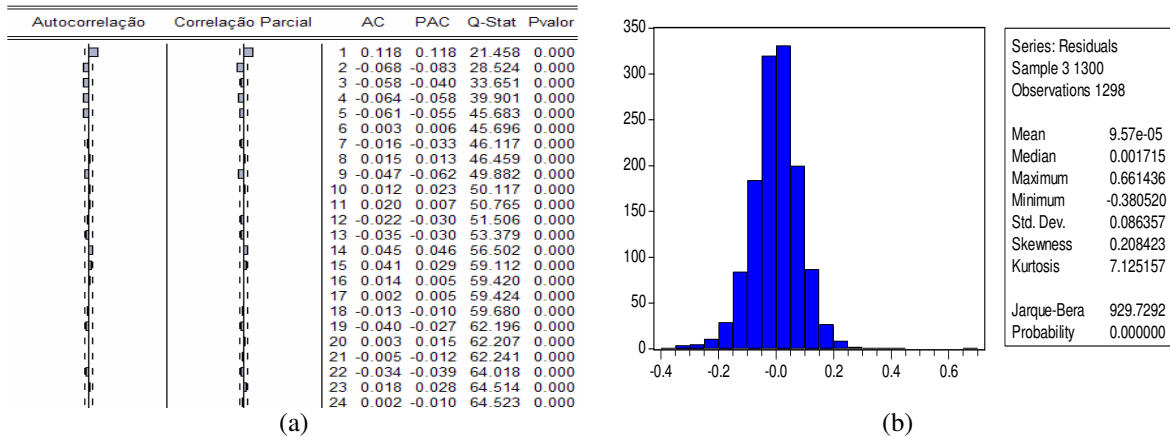


Figura 6.158 (a) Correlograma da série NDLSOUZ, (b) Estatísticas dos resíduos

Os resultados para o teste BDS são apresentados na Tabela 6.75.

BDS				
Dimensão	Estatística	Erro-padrão	Estatística-z	p-valor
2	0,010723	0,002013	5,326083	0,0000
3	0,023794	0,003187	7,465817	0,0000
4	0,030220	0,003780	7,994324	0,0000
5	0,035183	0,003924	8,965320	0,0000
6	0,037537	0,003769	9,958500	0,0000

Tabela 6.75 Resultados obtidos pelo teste BDS nas respectivas dimensões.

O melhor modelo ajustado para a série é um ARMA(2,1)-GARCH(1,1). Os parâmetros estimados do modelo para a série são mostrados na Tabela 6.76.

Parâmetros	Coefficiente	Erro-padrão	Estatística-z	p-valor
ϕ_0	0,359110	0,001678	214,0390	0,0000
ϕ_1	0,879950	0,073728	11,93503	0,0000
ϕ_2	-0,147924	0,031135	-4,751017	0,0000
θ_1	-0,788884	0,070592	-11,17528	0,0000
α_0	0,000240	6,50E-05	3,689005	0,0002
α_1	0,087287	0,012852	6,791852	0,0000
β_1	0,883718	0,017728	49,84941	0,0000

Tabela 6.76 Parâmetros estimados para a série NDLSOUZ

O resultado do teste Ljung-Box para a série NDLSOUZ é apresentado na Figura 6.159.

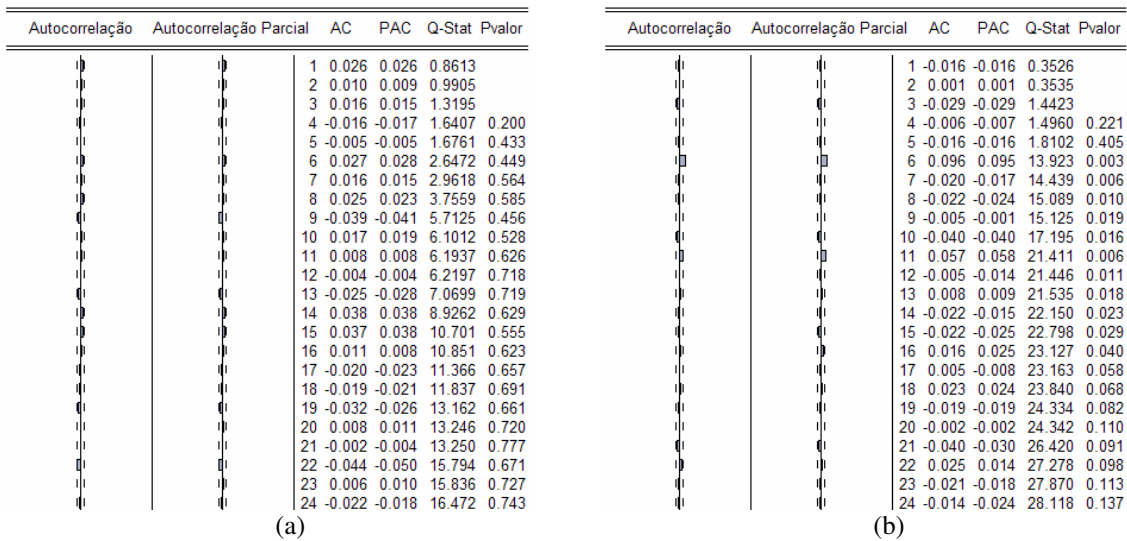


Figura 6.159 (a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado

As previsões realizadas utilizando o modelo ARIMA-GARCH são mostradas na Figura 6.160. A linha contínua indica o valor da série NDLSOUZ e a linha central com marcador quadrado indica as previsões.

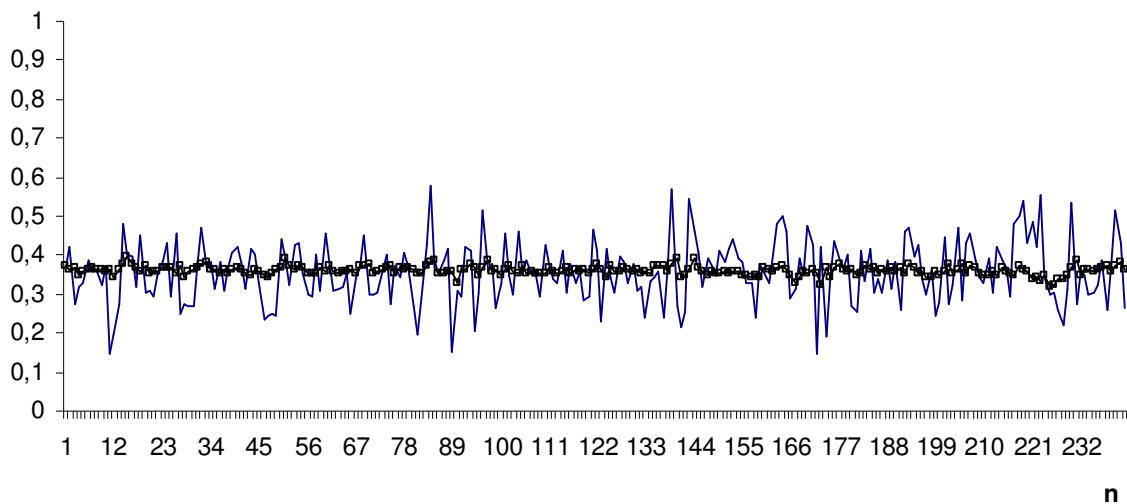


Figura 6.160 Previsão obtida pelo modelo ARIMA-GARCH para NDLSOUZ

Na Figura 6.161a é mostrada a previsão da volatilidade e na Figura 6.161b a série original e o intervalo de previsão para a série NDLSOUZ.

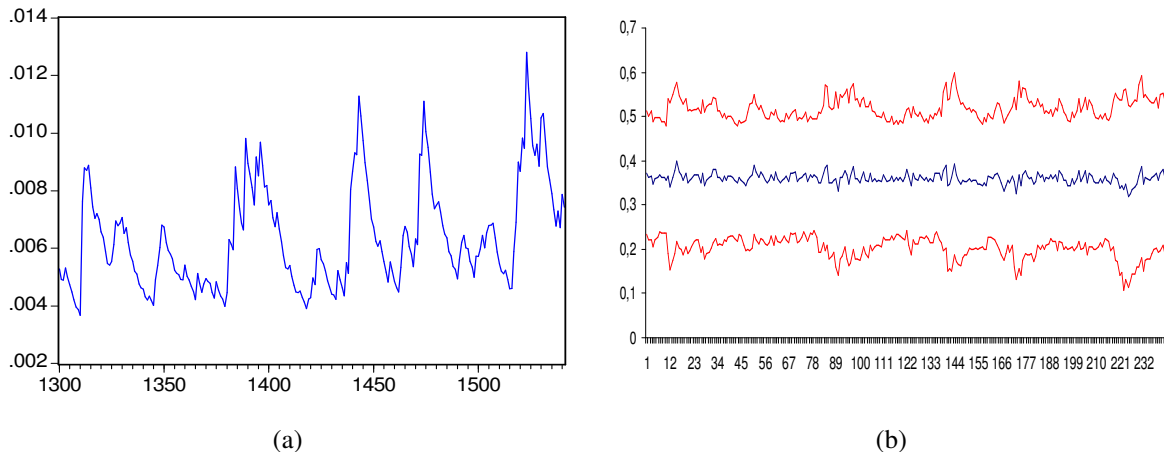


Figura 6.161 (a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLSOUZ

6.4.5.2 Análise da série da Souza Cruz – Método RNA *feedforward*

Para a série NDLSOUZ os resultados obtidos pelas diferentes topologias de RNA são apresentados na Tabela 6.77.

Topologia	FIC	FIP	TIC	RMSE
RNA 10-1-1	146	0	0,106247	0,0770627
RNA 10-2-1	148	0	0,108844	0,0787424
RNA 10-3-1	143	0	0,107033	0,0774686
RNA 10-4-1	146	0	0,111665	0,0812594
RNA 10-5-1	139	1	0,135643	0,0979015
RNA 10-8-1	134	0	0,115426	0,0836699

Tabela 6.77 Estatísticas para diferentes topologias de RNA para a série

Na Figura 6.162 são apresentadas as previsões obtidas pela RNA 10-1-1 para a série NDLSOUZ.

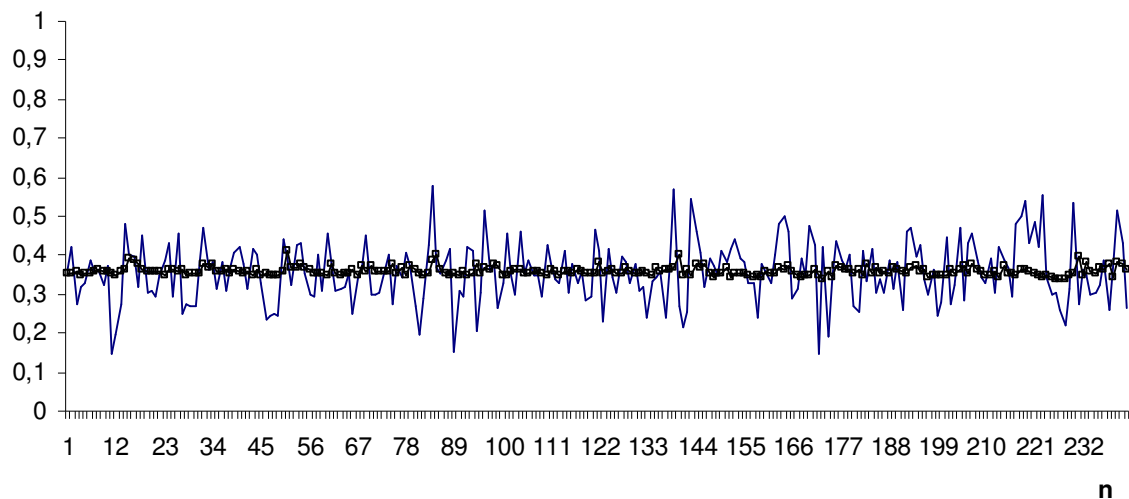


Figura 6.162 Previsão obtida pela RNA *feedforward* para NDLSOUZ

Na Figura 6.163a é mostrado o intervalo de confiança para as previsões obtidas pela RNA 10-1-1 e na Figura 6.163b o intervalo de predição para a série NDLSOUZ.

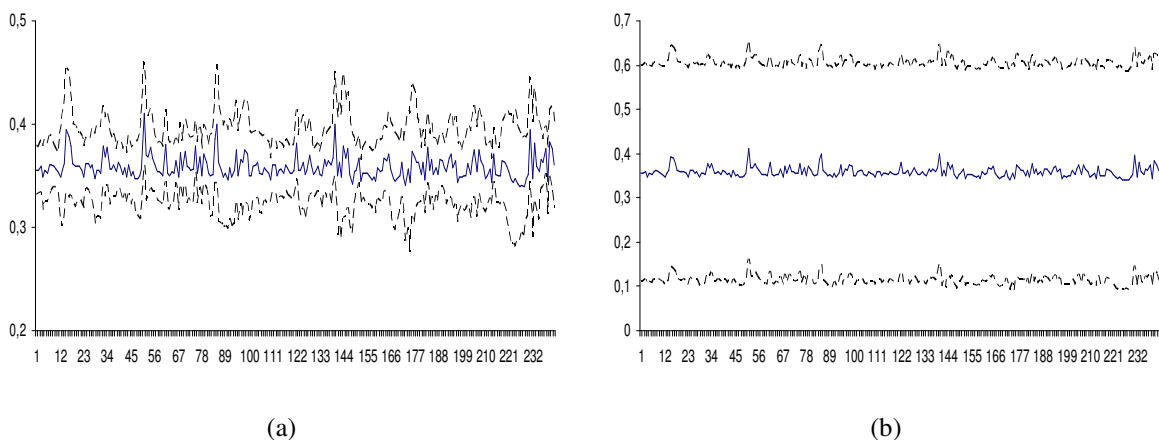


Figura 6.163 (a) Intervalo de confiança, (b) Intervalo de predição para NDLSOUZ

Na Figura 6.164a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA 10-1-1 e na Figura 6.164b o histograma desses resíduos.

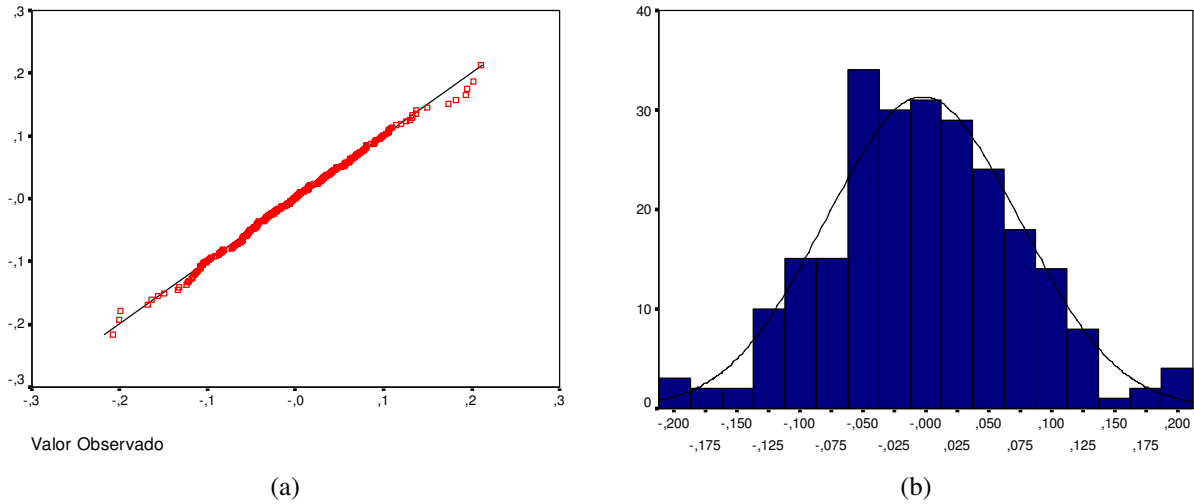


Figura 6.164 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA *feedforward* para NDLSOUZ

6.4.5.3 Análise da série da Souza Cruz – Método RNA EKF

As previsões, para a série NDLSOUZ, obtidas pela RNA treinada com o filtro de Kalman estendido (EKF) são mostradas na Figura 6.165.

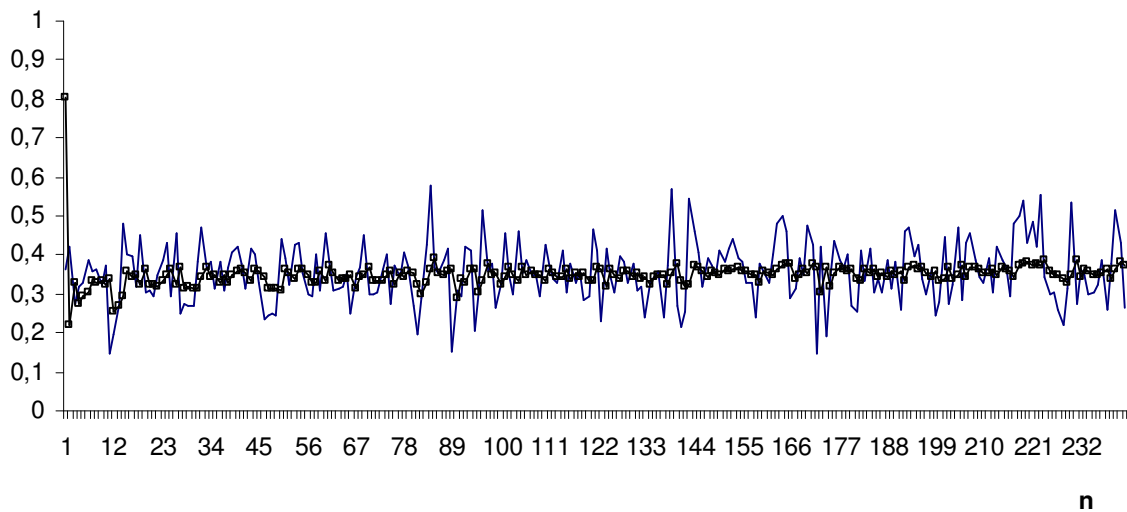


Figura 6.165 Previsão obtida pela RNA EKF para NDLSOUZ

Na Figura 6.166a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman estendido e na Figura 6.166b o histograma desses resíduos.

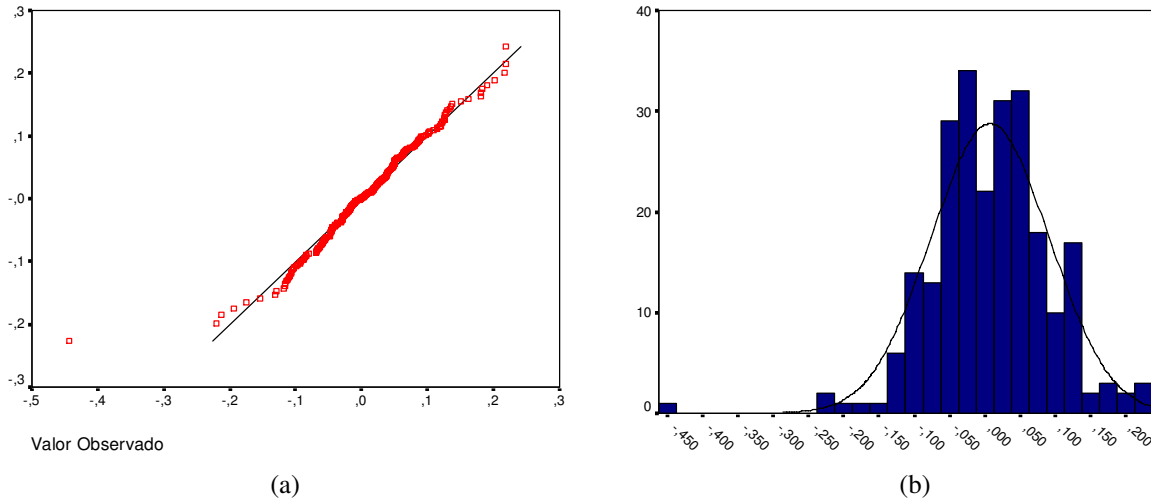


Figura 6.166 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLSOUZ

6.4.5.4 Análise da série da Souza Cruz – Método RNA UKF

As previsões obtidas pela RNA treinada com o filtro de Kalman unscented são mostradas na Figura 6.167.

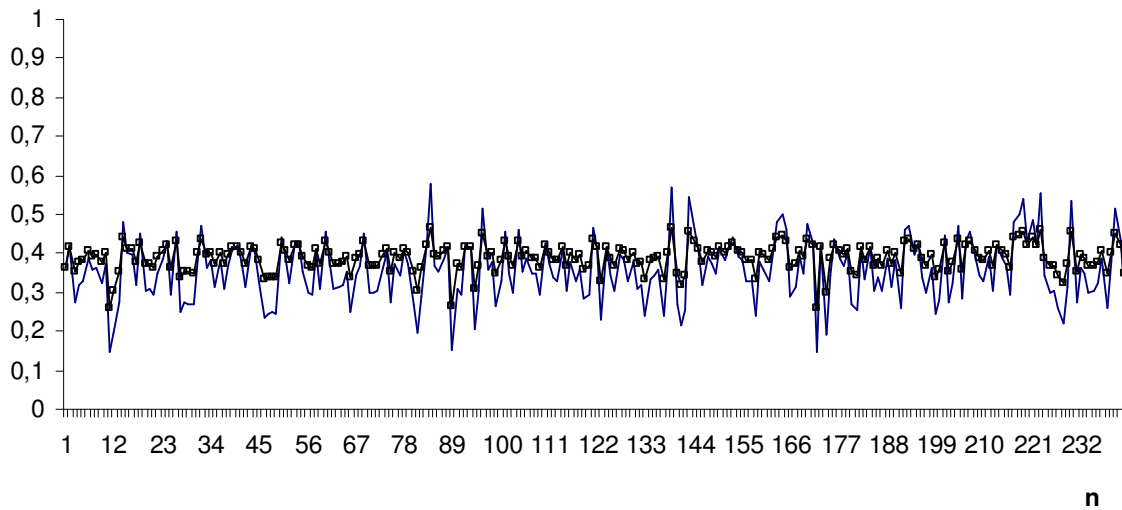


Figura 6.167 Previsão obtida pela RNA UKF para NDLSOUZ

Na Figura 6.168a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman *unscented* e na Figura 6.168b o histograma desses resíduos.

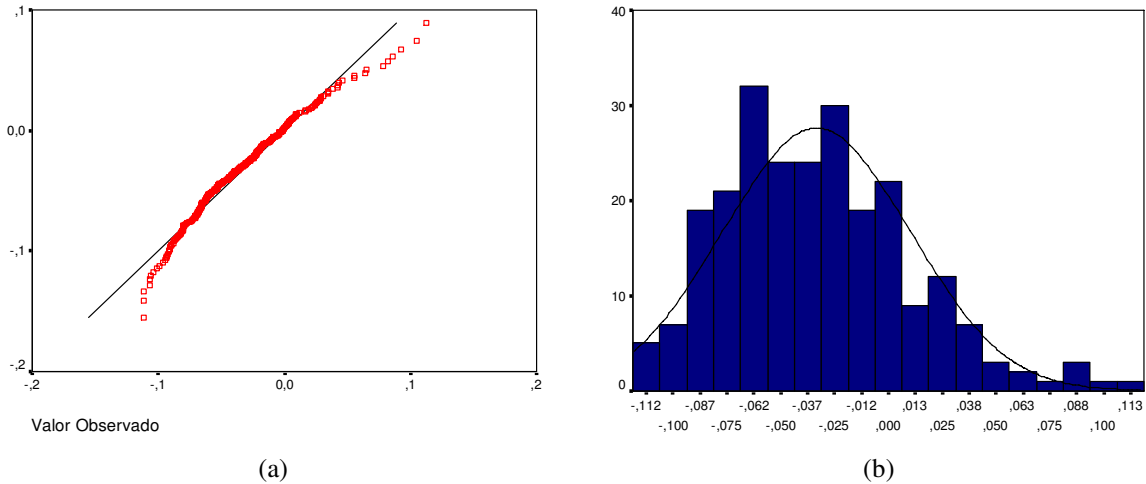


Figura 6.168 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLSOUZ

As estatísticas de previsão obtidas pelos modelos ARIMA-GARCH, RNA *feedforward*, RNA EKF e RNA UKF são apresentadas na Tabela 6.78 e na Figura 6.169.

Método	TIC	RMSE
ARMA(2,1)-GARCH(1,1)	0,105425	0,076396
RNA 10-1-1	0,106247	0,077063
RNA EKF	0,117432	0,083913
RNA UKF	0,072650	0,054875

Tabela 6.78 Estatísticas de previsão para o método de ARIMA-GARCH e de RNA

Pode-se verificar que as melhores previsões foram obtidas pelo método que empregou redes neurais treinadas com o filtro de Kalman *unscented*.

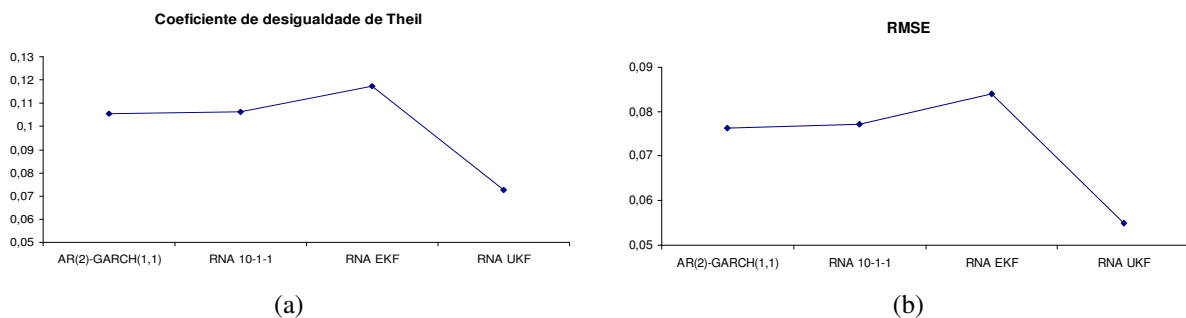


Figura 6.169 Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b)

RMSE

6.4.6 Análise da série da Gerdau

A seguir são apresentados os resultados obtidos para a série da Gerdau.

6.4.6.1 Análise da série da Gerdau – Método ARIMA-GARCH

Na Figura 6.170a é apresentada a série original do preço de fechamento da Gerdau, na Figura 6.170b é apresentada a série normalizada da primeira diferença do logaritmo do preço, NDLGERD.

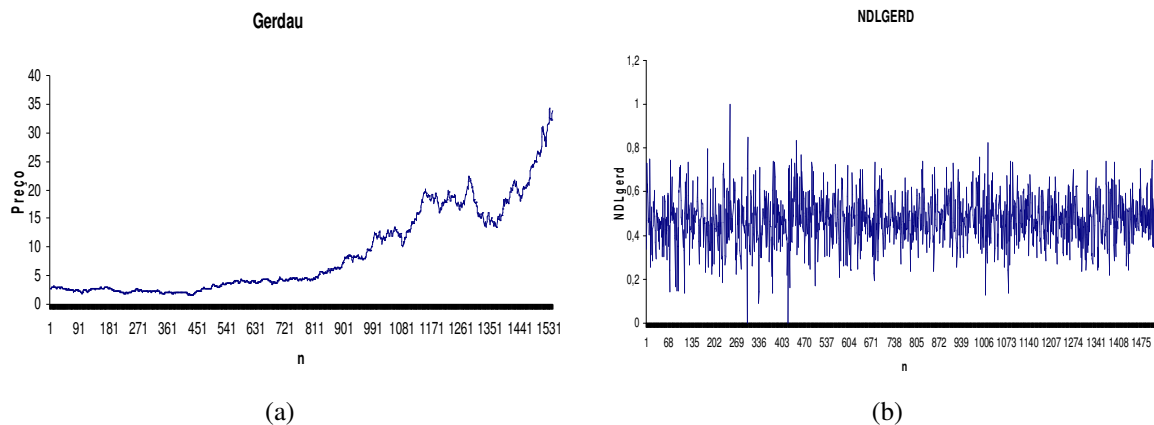


Figura 6.170 (a) Série no nível do preço do Bradesco, (b) Série transformada NDLGERD

O resultado do teste ADF, para a série original, é apresentado na Tabela 6.79.

Estatística do teste ADF para série original	valores críticos 1%	-3,43439
2,5839	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771

Tabela 6.79 Teste ADF para série original da Gerdau

Não é rejeitada a hipótese de raiz unitária para a série no nível pelo teste ADF. O teste de presença de uma raiz unitária para a série transformada apresentou os seguintes resultados:

Estatística do teste ADF para série transformada	valores críticos 1%	-3,43439
-26,2791	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771

Tabela 6.80 Teste ADF para série NDLGERD

Na Figura 6.171 é apresentada a estatística de Jarque-Bera, que rejeita fortemente a hipótese de distribuição normal.

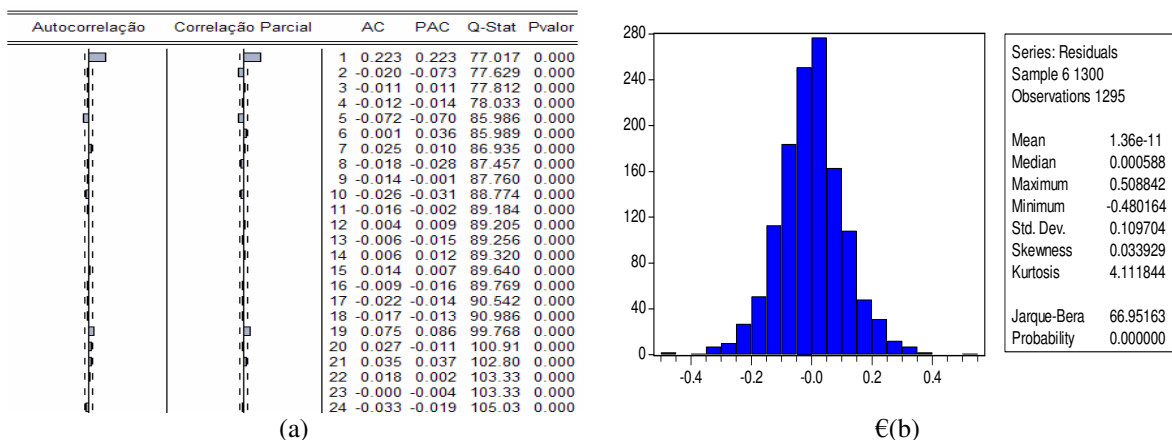


Figura 6.171 (a) Correlograma da série NDLGERD, (b) Estatísticas dos resíduos

Os resultados para o teste BDS são apresentados na Tabela 6.81.

BDS				
Dimensão	Estatística	Erro-padrão	Estatística-z	p-valor
2	0,012370	0,001894	6,530277	0,0000
3	0,020900	0,002997	6,974642	0,0000
4	0,026464	0,003552	7,451446	0,0000
5	0,029548	0,003684	8,020877	0,0000
6	0,029726	0,003535	8,407964	0,0000

Tabela 6.81 Resultados obtidos pelo teste BDS nas respectivas dimensões.

O melhor modelo ajustado para a série é um AR([1,5])-GARCH(1,2). Os parâmetros estimados do modelo para a série são mostrados na Tabela 6.82.

Parâmetros	Coefficiente	Erro-padrão	Estatística-z	p-valor
ϕ_0	0,471706	0,003567	132,2595	0,0000
ϕ_1	0,223667	0,028531	7,839307	0,0000
ϕ_5	-0,085569	0,027208	-3,145014	0,0017
α_0	0,001530	0,000440	3,473241	0,0005
α_1	0,063380	0,020474	3,095603	0,0020
β_1	1,236971	0,229076	5,399829	0,0000
β_2	0,427018	0,190692	-2,239307	0,0251

Tabela 6.82 Parâmetros estimados para a série NDLGERD

O resultado do teste Ljung-Box para a série NDLGERD é apresentado na Figura 6.172.

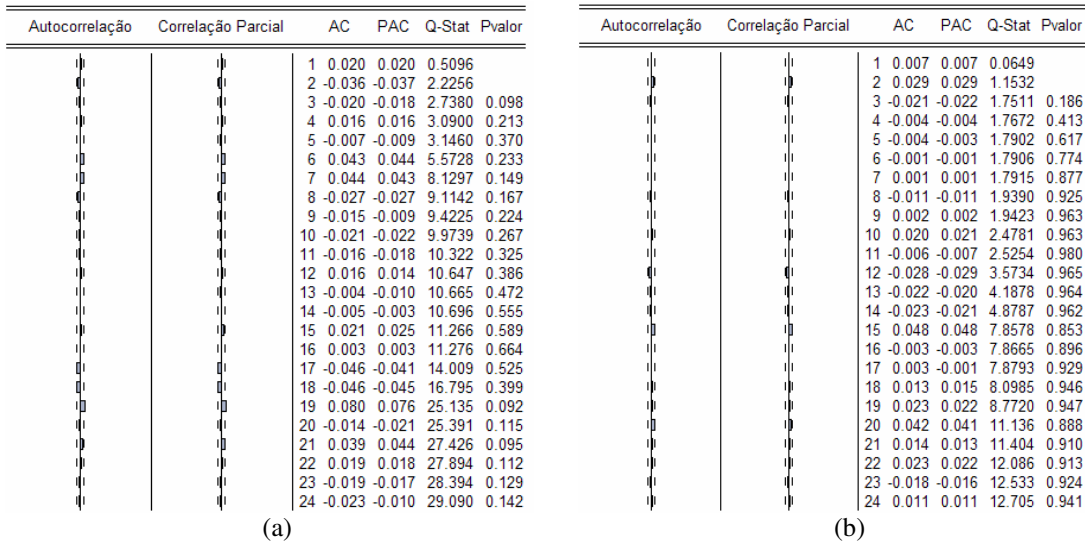


Figura 6.172 (a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado

As previsões realizadas utilizando o modelo ARIMA-GARCH são mostradas na Figura 6.173. A linha contínua indica o valor da série NDLGERD e a linha central com marcador quadrado indica as previsões.

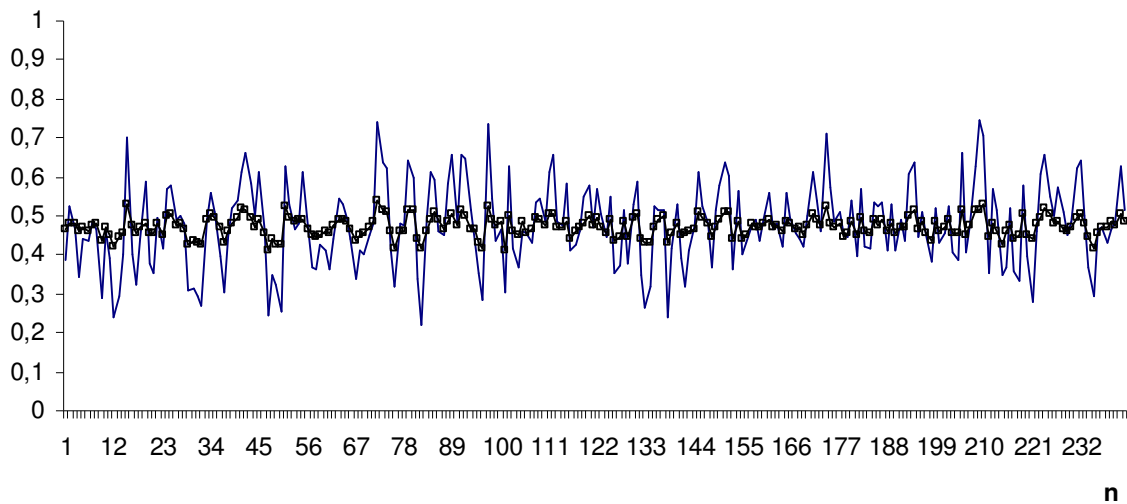


Figura 6.173 Previsão obtida pelo modelo ARIMA-GARCH para NDLGERD

Na Figura 6.174a é mostrada a previsão da volatilidade e na Figura 6.174b a série original e o intervalo de previsão para a série NDLGERD.

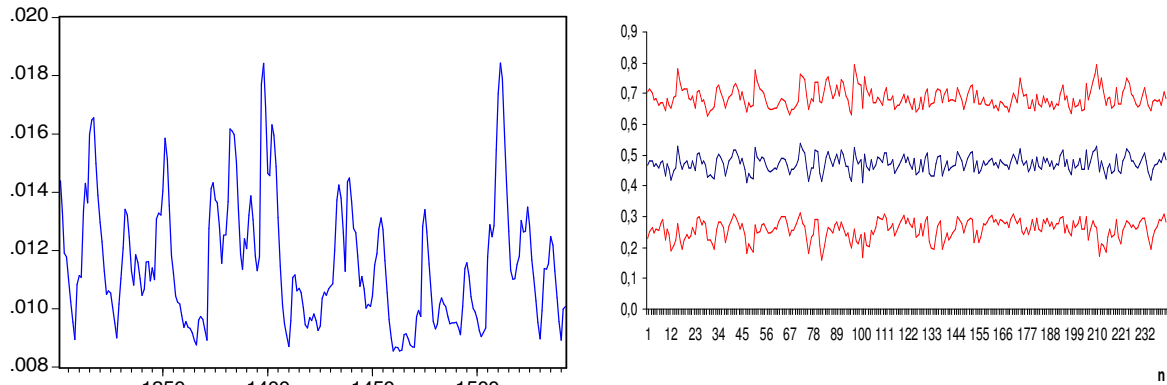


Figura 6.174 (a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLGERD

6.4.6.2 Análise da série da Gerdau – Método RNA *feedforward*

Para a série NDLGERD os resultados obtidos pelas diferentes topologias de RNA são apresentados na Tabela 6.83.

Topologia	FIC	FIP	TIC	RMSE
RNA 10-1-1	105	0	0,106974	0,102849
RNA 10-2-1	105	0	0,105365	0,101338
RNA 10-3-1	104	0	0,106219	0,10207
RNA 10-4-1	105	0	0,108299	0,104256
RNA 10-5-1	103	0	0,107115	0,102865
RNA 10-8-1	101	0	0,109942	0,105815

Tabela 6.83 Estatísticas para diferentes topologias de RNA para a série

Na Figura 6.175 são apresentadas as previsões obtidas pela RNA 10-2-1 para a série NDLGERD.

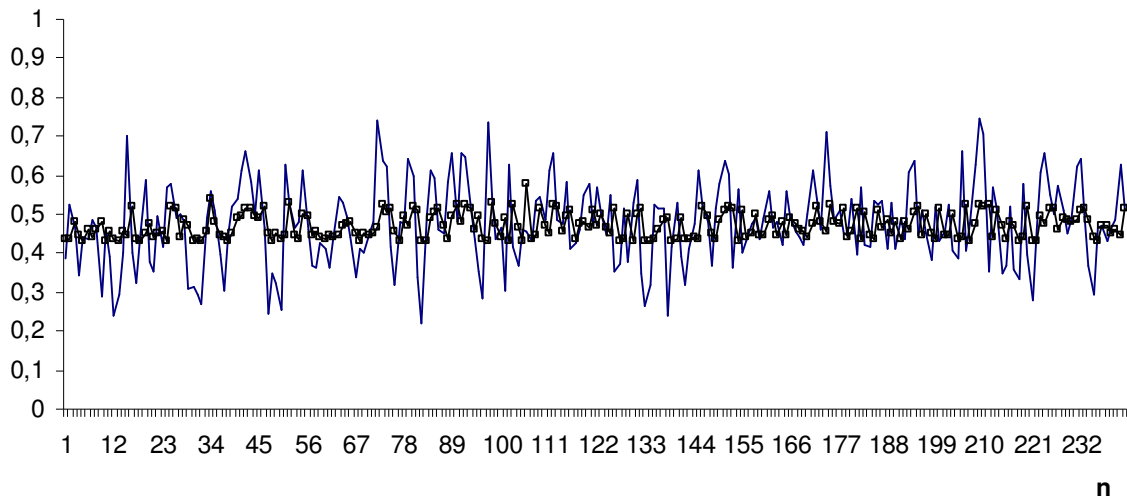
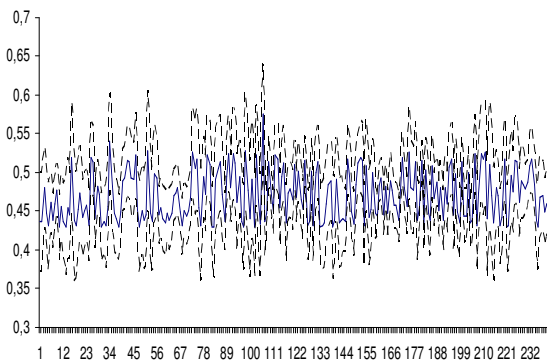
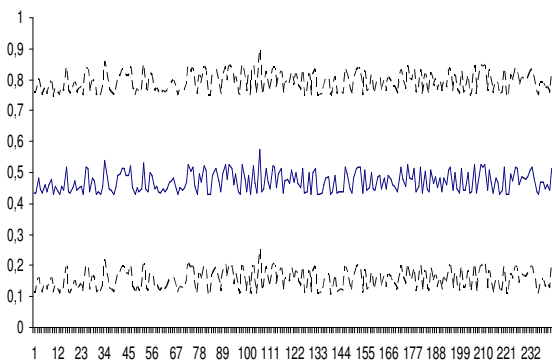


Figura 6.175 Previsão obtida pela RNA *feedforward* para NDLGERD

Na Figura 6.176a é mostrado o intervalo de confiança para as previsões obtidas pela RNA 10-2-1 e na Figura 6.176b o intervalo de predição para a série NDLGERD.



(a)



(b)

Figura 6.176 (a) Intervalo de confiança, (b) Intervalo de predição para NDLGERD

Na Figura 6.177a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA 10-2-1 e na Figura 6.177b o histograma desses resíduos.

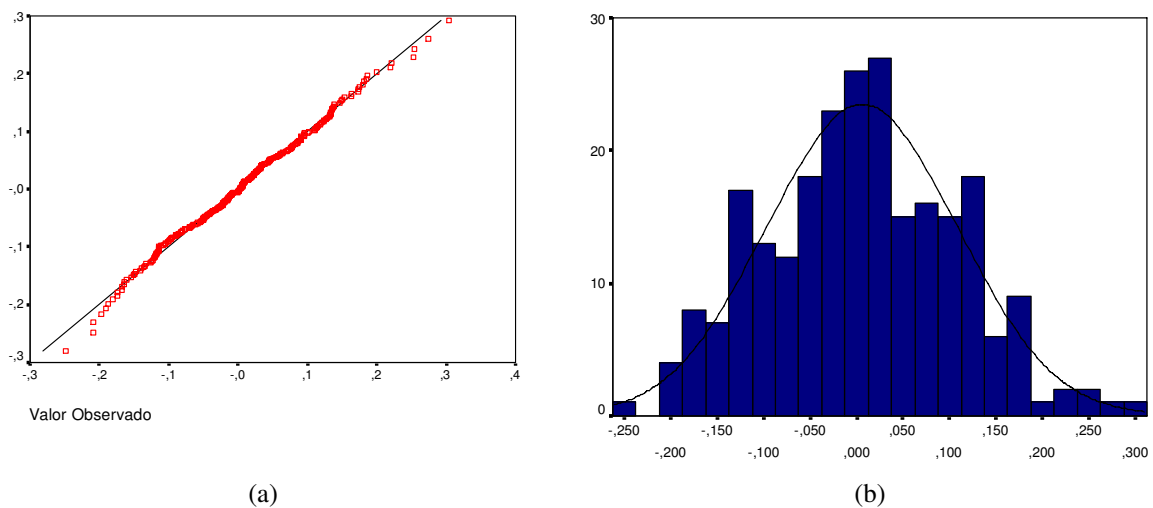


Figura 6.177 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA *feedforward* para NDLGERD

6.4.6.3 Análise da série da Gerdau – Método RNA EKF

As previsões, para a série NDLGERD, obtidas pela RNA treinada com o filtro de Kalman estendido (EKF) são mostradas na Figura 6.178.

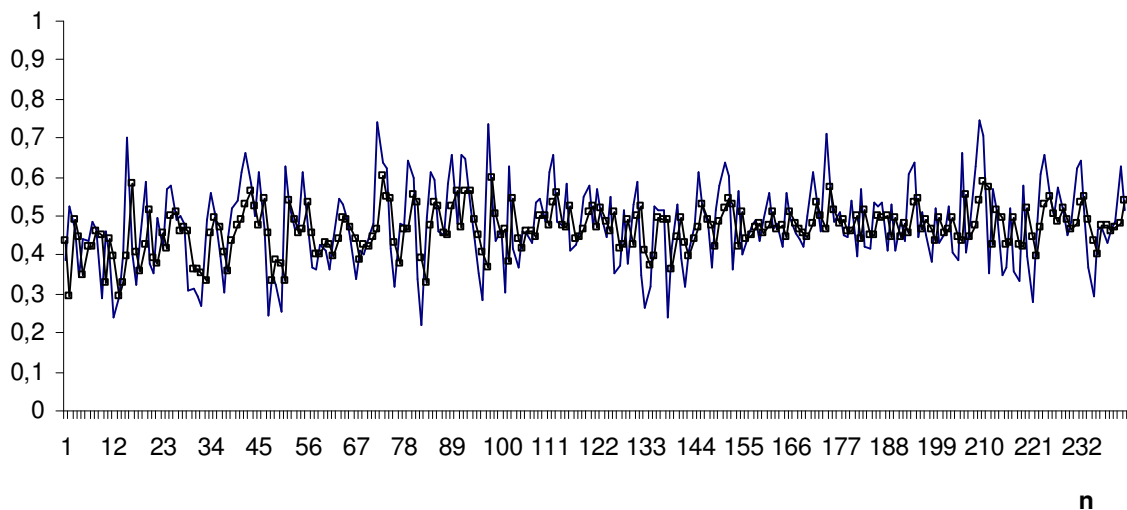


Figura 6.178 Previsão obtida pela RNA EKF para NDLGERD

Na Figura 6.179a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman estendido e na Figura 6.179b o histograma desses resíduos.

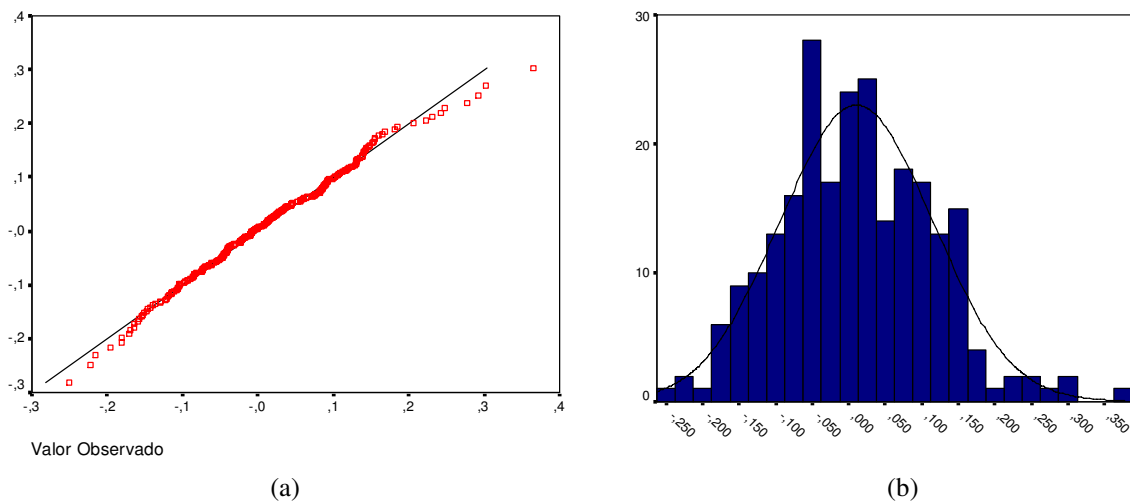


Figura 6.179 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para ND LGERD

6.4.6.4 Análise da série da Gerdau – Método RNA UKF

As previsões obtidas pela RNA treinada com o filtro de Kalman *unscented* são mostradas na Figura 6.180.

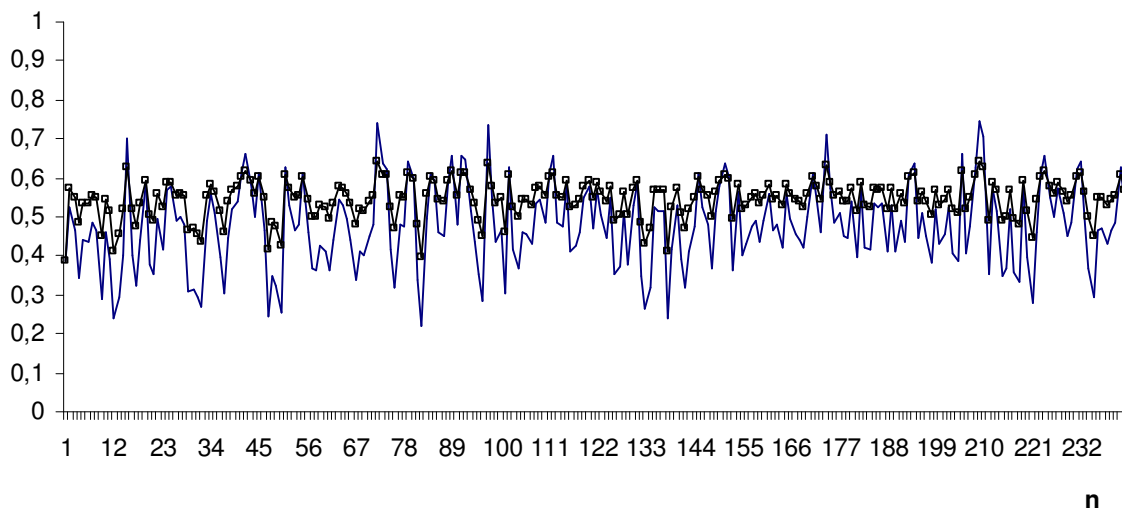


Figura 6.180 Previsão obtida pela RNA UKF para ND LGERD

Na Figura 6.181a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman *unscented* e na Figura 6.181b o histograma desses resíduos.

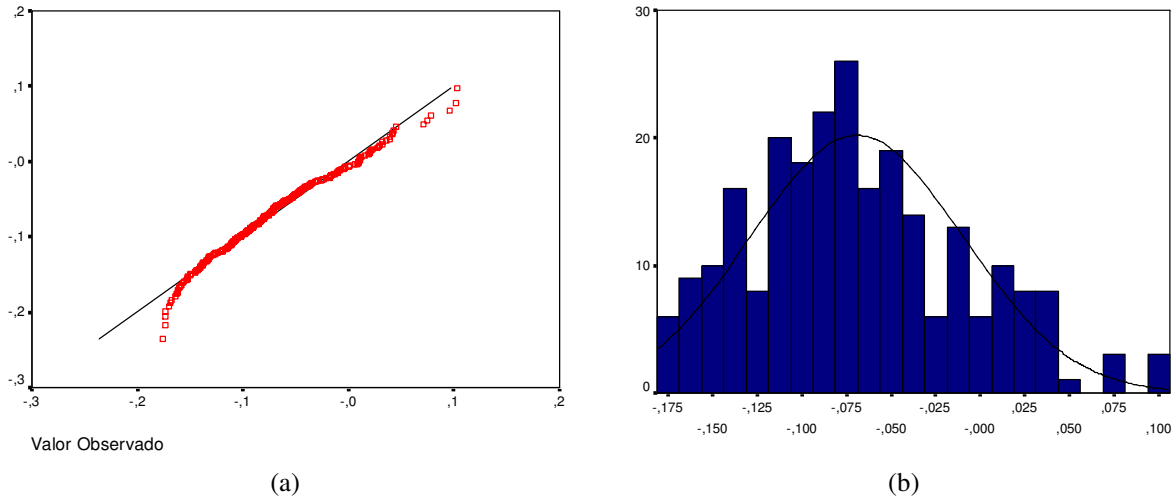


Figura 6.181 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLOGGER

As estatísticas de previsão obtidas pelos modelos ARIMA-GARCH, RNA *feedforward*, RNA EKF e RNA UKF são apresentadas na Tabela 6.84 e na Figura 6.182.

Método	TIC	RMSE
AR([1,5])-GARCH(1,2)	0,105962	0,101888
RNA 10-2-1	0,105365	0,101338
RNA EKF	0,109806	0,105104
RNA UKF	0,088050	0,091181

Tabela 6.84 Estatísticas de previsão para o método de ARIMA-GARCH e de RNA

Pode-se verificar que as melhores previsões foram obtidas pelo método que empregou redes neurais treinadas com o filtro de Kalman *unscented*.

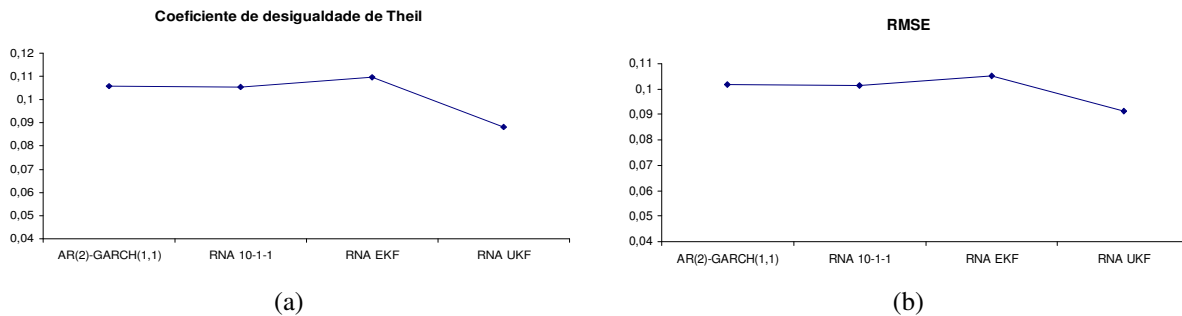


Figura 6.182 Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b) RMSE

6.5 Setor de Serviços – Pão de Açúcar, Eletropaulo, Eletrobras, Brasil Telecom, Cesp e Lojas Americanas

No caso do setor de serviços as séries analisadas seguirão a seqüência: Pão de Açúcar, Eletropaulo, Eletrobras, Brasil Telecom, Cesp e Lojas Americanas.

6.5.1 Análise da série do Pão de Açúcar

A seguir são apresentados os resultados obtidos para a série da Pão de Açúcar.

6.5.1.1 Análise da série do Pão de Açúcar – Método ARIMA-GARCH

Na Figura 6.183a é apresentada a série original do preço de fechamento do Pão de Açúcar, na Figura 6.183b é apresentada a série normalizada da primeira diferença do logaritmo do preço, NDLPAOD.

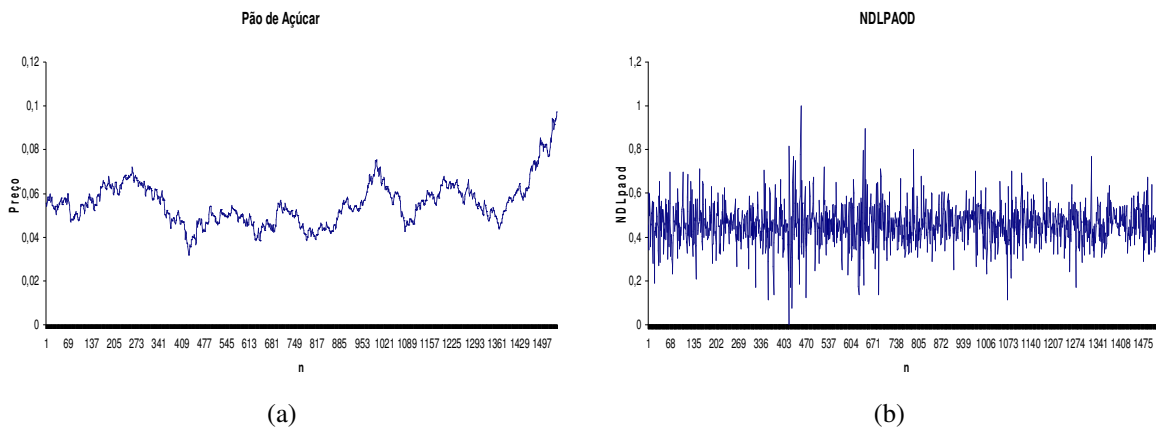


Figura 6.183 (a) Série no nível do preço do Pão de Açúcar, (b) Série transformada NDLPAOD

O resultado do teste ADF, para a série original, é apresentado na Tabela 6.85.

Estatística do teste ADF para série original		valores críticos 1%	-3,43439
	-1,12288	valores críticos 5%	-2,86321
		valores críticos 10%	-2,56771

Tabela 6.85 Teste ADF para série original do Pão de Açúcar

Não é rejeitada a hipótese de raiz unitária para a série no nível pelo teste ADF. O teste de presença de uma raiz unitária para a série transformada apresentou os seguintes resultados:

Estatística do teste ADF para série transformada	valores críticos 1%	-3,43439
-32,3008	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771

Tabela 6.86 Teste ADF para série NDLPAD

Na Figura 6.184 é apresentada a estatística de Jarque-Bera, que rejeita fortemente a hipótese de distribuição normal.

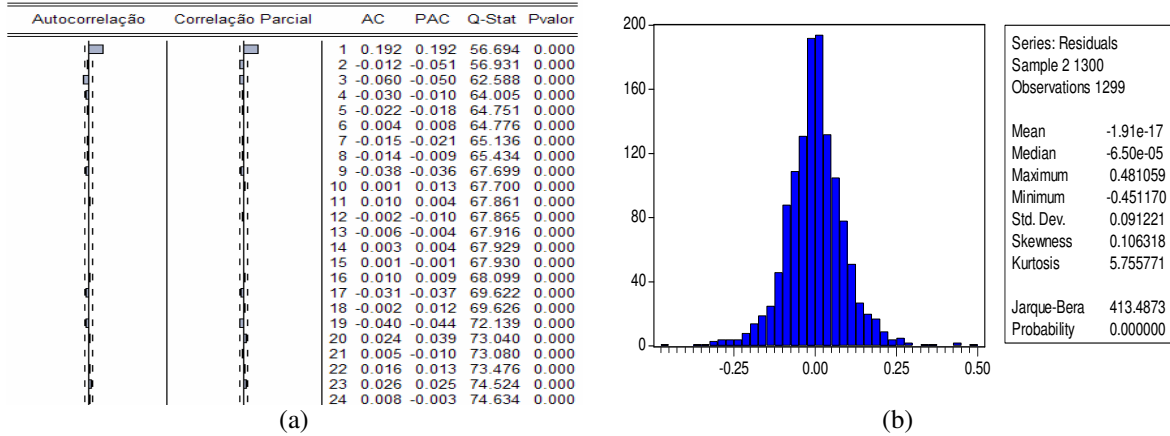


Figura 6.184 (a) Correlograma da série NDLPAD, (b) Estatísticas dos resíduos

Os resultados para o teste BDS são apresentados na Tabela 6.87.

BDS				
Dimensão	Estatística	Erro-padrão	Estatística-z	p-valor
2	0,018311	0,002144	8,541553	0,0000
3	0,034349	0,003396	10,11477	0,0000
4	0,046301	0,004031	11,48712	0,0000
5	0,050719	0,004187	12,11207	0,0000
6	0,052324	0,004025	12,99924	0,0000

Tabela 6.87 Resultados obtidos pelo teste BDS nas respectivas dimensões.

O melhor modelo ajustado para a série é um AR(1)-GARCH(1,1). Os parâmetros estimados do modelo para a série são mostrados na Tabela 6.88.

Parâmetros	Coefficiente	Erro-padrão	Estatística-z	p-valor
ϕ_0	0,457490	0,002958	154,6595	0,0000
ϕ_1	0,207472	0,026685	7,774861	0,0000
α_0	0,000243	7,77E-05	3,123960	0,0018
α_1	0,047141	0,008710	5,412514	0,0000
β_1	0,923120	0,016001	57,69272	0,0000

Tabela 6.88 Parâmetros estimados para a série NDLPAD

O resultado do teste Ljung-Box para a série NDLPAD é apresentado na Figura 6.185.

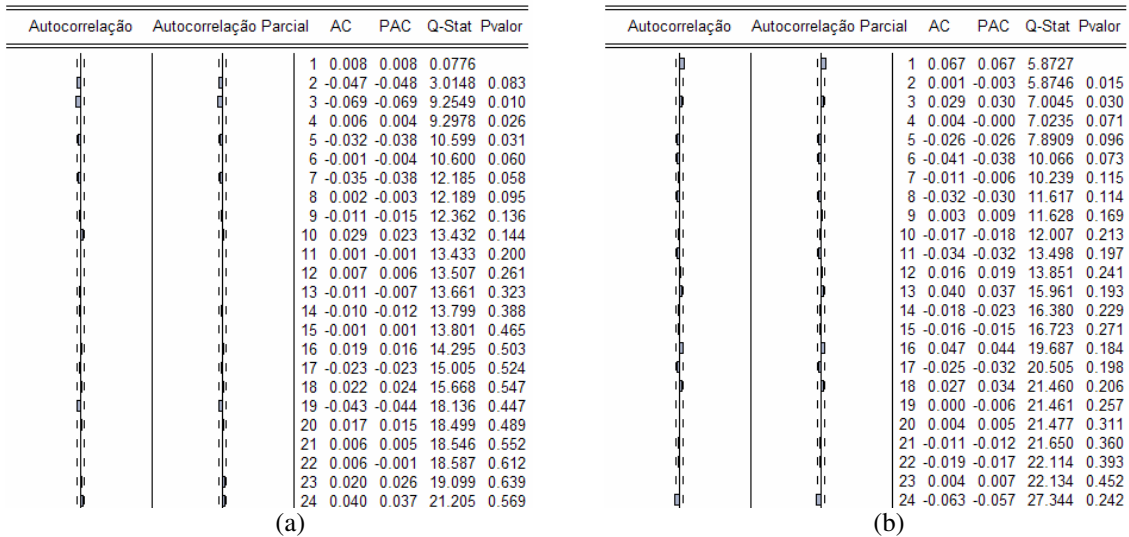


Figura 6.185 (a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado

As previsões realizadas utilizando o modelo ARIMA-GARCH são mostradas na Figura 6.186. A linha contínua indica o valor da série NDLPAD e a linha central com marcador quadrado indica as previsões.

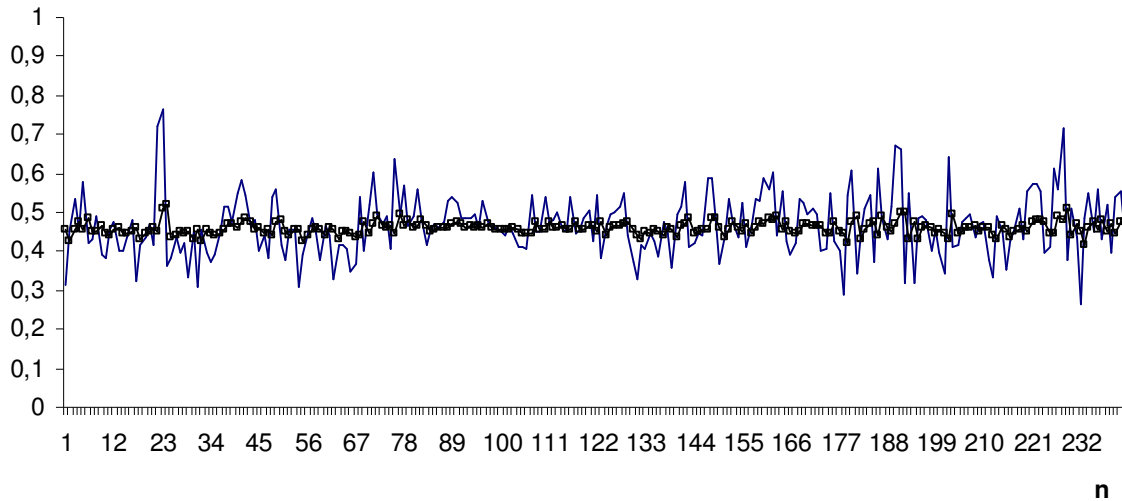


Figura 6.186 Previsão obtida pelo modelo ARIMA-GARCH para NDLPAD

Na Figura 6.187a é mostrada a previsão da volatilidade e na Figura 6.187b a série original e o intervalo de previsão para a série NDLPAD.

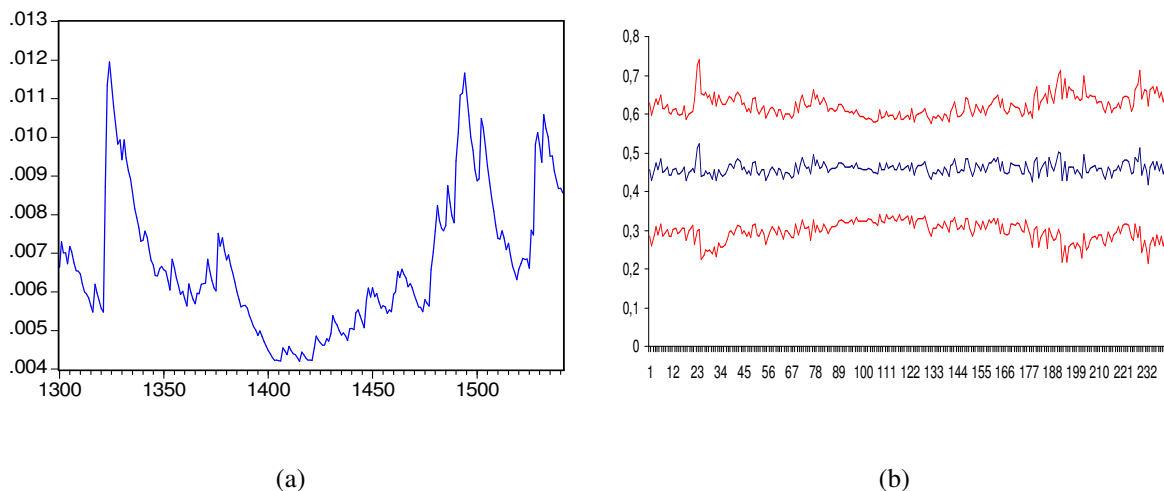


Figura 6.187 (a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLPAD

6.5.1.2 Análise da série do Pão de Açúcar – Método RNA *feedforward*

Para a série NDLPAD os resultados obtidos pelas diferentes topologias de RNA são apresentados na Tabela 6.89.

Topologia	FIC	FIP	TIC	RMSE
RNA 10-1-1	106	1	0,0835172	0,0777186
RNA 10-2-1	114	1	0,084331	0,0785562
RNA 10-3-1	103	0	0,0862342	0,0803553
RNA 10-4-1	113	0	0,0847044	0,0787518
RNA 10-5-1	99	0	0,0862845	0,0803278
RNA 10-8-1	90	0	0,0889694	0,0828117

Tabela 6.89 Estatísticas para diferentes topologias de RNA para a série

Na Figura 6.188 são apresentadas as previsões obtidas pela RNA 10-1-1 para a série NDLPAD.

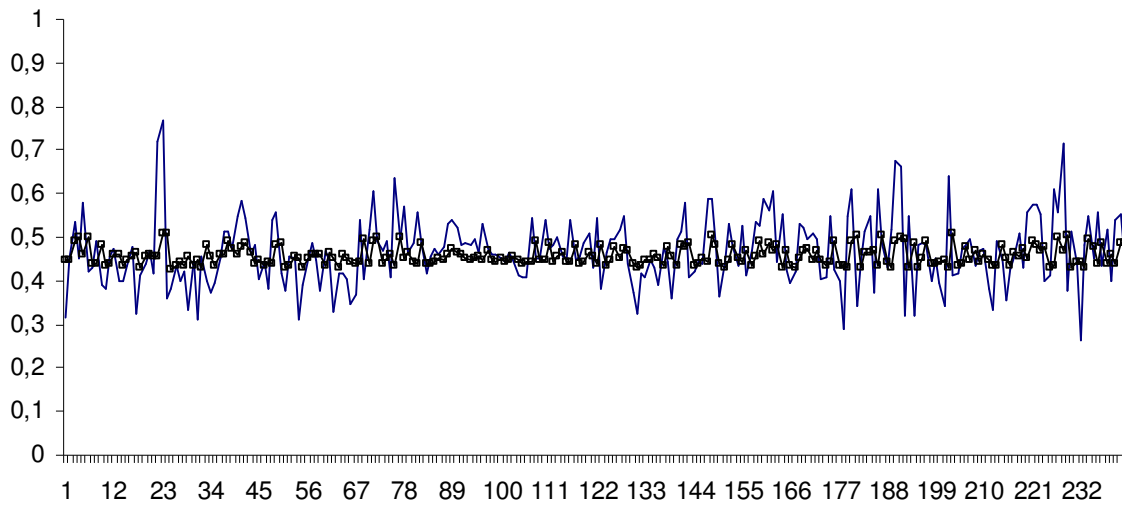
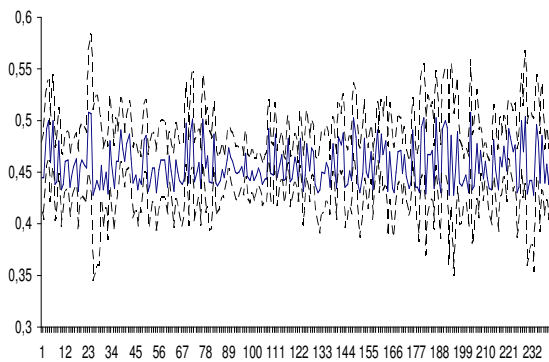
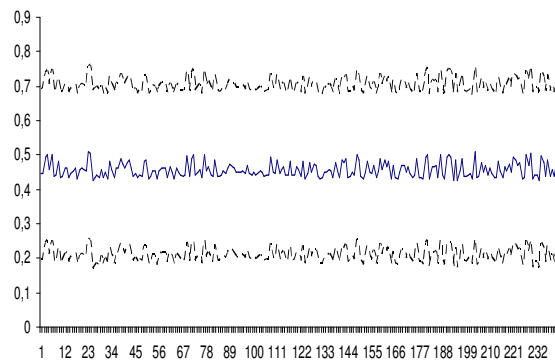


Figura 6.188 Previsão obtida pela RNA *feedforward* para NDLPAOD

Na Figura 6.189a é mostrado o intervalo de confiança para as previsões obtidas pela RNA 10-1-1 e na Figura 6.189b o intervalo de predição para a série NDLPAOD.



(a)



(b)

Figura 6.189 (a) Intervalo de confiança, (b) Intervalo de predição para NDLPAOD

Na Figura 6.190a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA 10-1-1 e na Figura 6.190b o histograma desses resíduos.

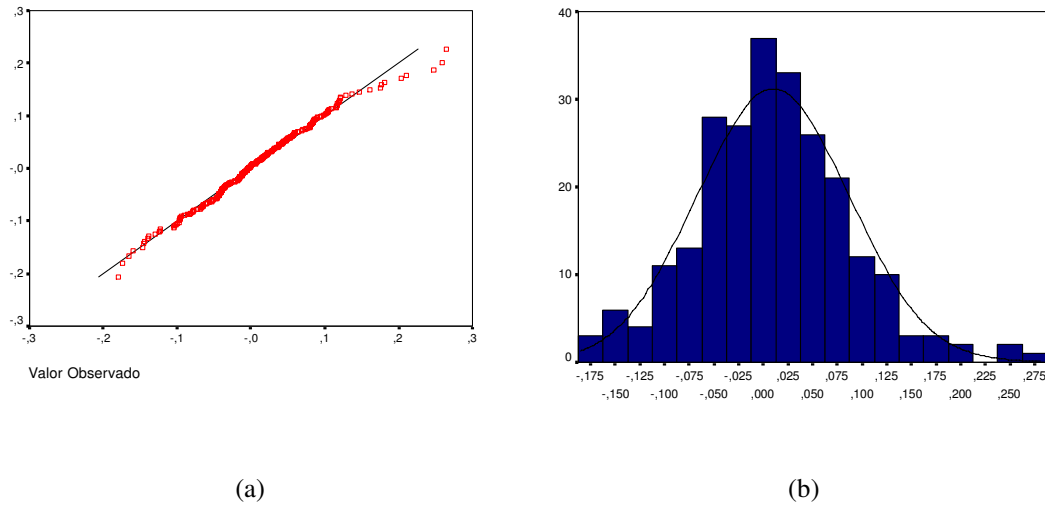


Figura 6.190 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA *feedforward* para NDLPAD

6.5.1.3 Análise da série do Pão de Açúcar – Método RNA EKF

As previsões, para a série NDLPAD, obtidas pela RNA treinada com o filtro de Kalman estendido (EKF) são mostradas na Figura 6.191.

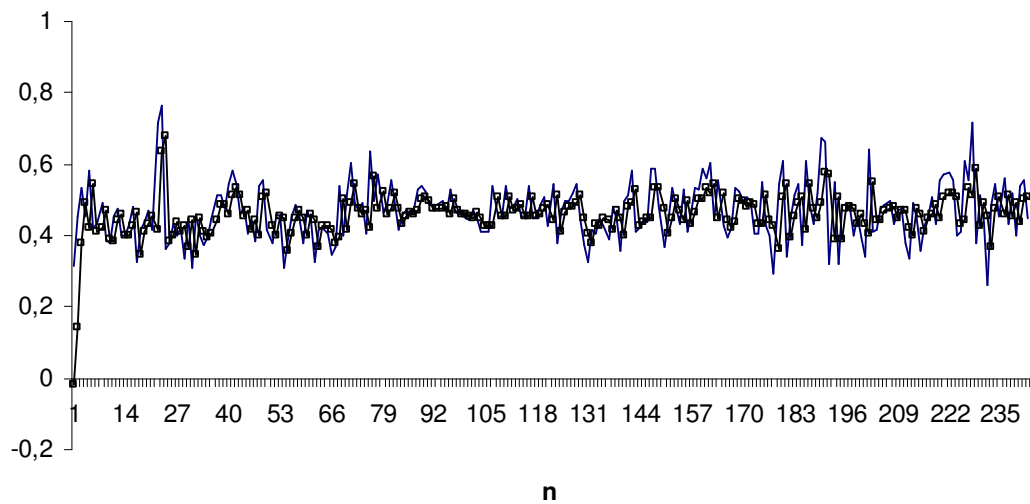


Figura 6.191 Previsão obtida pela RNA EKF para NDLPAD

Na Figura 6.192a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman estendido e na Figura 6.192b o histograma desses resíduos.

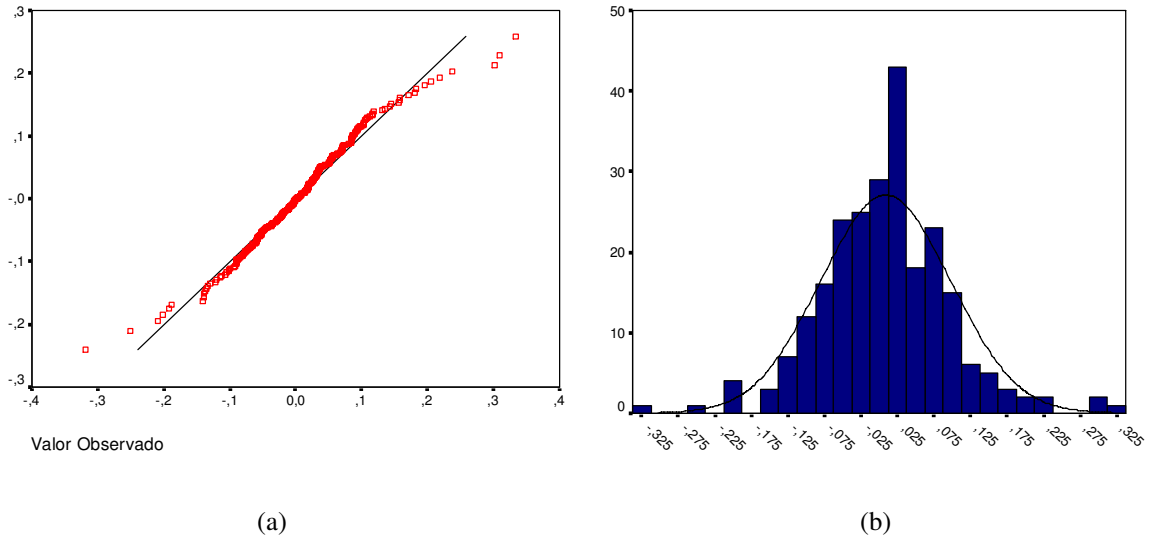


Figura 6.192 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLPAD

6.5.1.4 Análise da série do Pão de Açúcar – Método RNA UKF

As previsões obtidas pela RNA treinada com o filtro de Kalman unscented são mostradas na Figura 6.193.

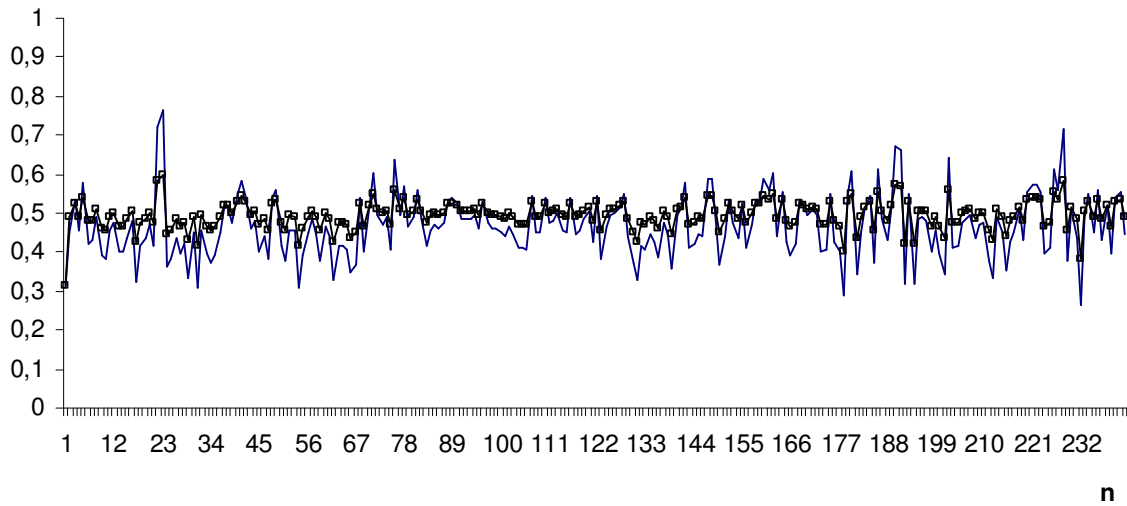


Figura 6.193 Previsão obtida pela RNA UKF para NDLPAD

Na Figura 6.194a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman *unscented* e na Figura 6.194b o histograma desses resíduos.

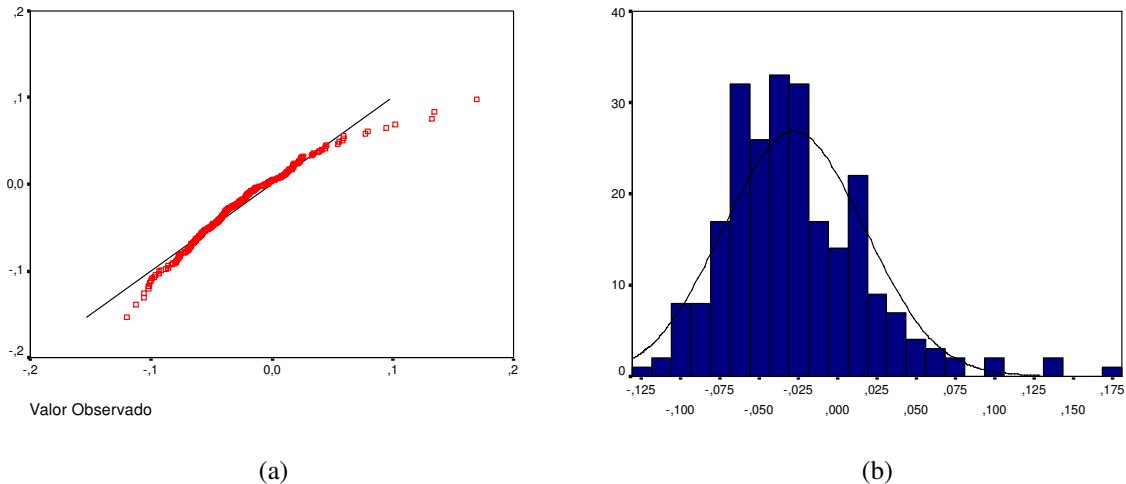


Figura 6.194 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLPAD

As estatísticas de previsão obtidas pelos modelos ARIMA-GARCH, RNA *feedforward*, RNA EKF e RNA UKF são apresentadas na Tabela 6.90 e na Figura 6.195.

Método	TIC	RMSE
AR(1)-GARCH(1,1)	0,082931	0,077371
RNA 10-1-1	0,083517	0,077719
RNA EKF	0,095404	0,089211
RNA UKF	0,054492	0,052816

Tabela 6.90 Estatísticas de previsão para o método de ARIMA-GARCH e de RNA

Pode-se verificar que as melhores previsões foram obtidas pelo método que empregou redes neurais treinadas com o filtro de Kalman *unscented*.

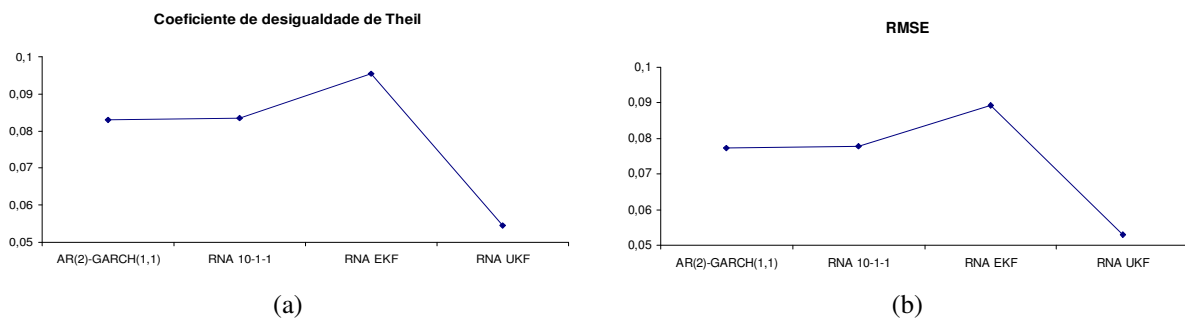


Figura 6.195 Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b)

RMSE

6.5.2 Análise da série da Eletropaulo

A seguir são apresentados os resultados obtidos para a série da Eletropaulo.

6.5.2.1 Análise da série da Eletropaulo – Método ARIMA-GARCH

Na Figura 6.196a é apresentada a série original do preço de fechamento da Eletropaulo, na Figura 6.196b é apresentada a série normalizada da primeira diferença do logaritmo do preço, NDLELET.

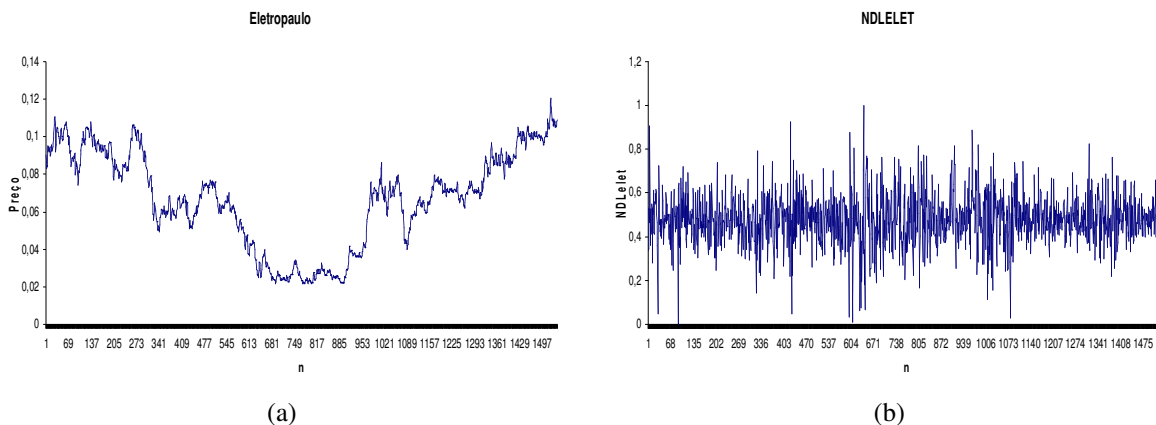


Figura 6.196 (a) Série no nível do preço do Bradesco, (b) Série transformada NDLELET

O resultado do teste ADF, para a série original, é apresentado na Tabela 6.91.

Estatística do teste ADF para série original	valores críticos 1%	-3,43439
-1,34233	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771

Tabela 6.91 Teste ADF para série original da Eletropaulo

Não é rejeitada a hipótese de raiz unitária para a série no nível pelo teste ADF. O teste de presença de uma raiz unitária para a série transformada apresentou os seguintes resultados:

Estatística do teste ADF para série transformada	valores críticos 1%	-3,43439
-32,1299	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771

Tabela 6.92 Teste ADF para série NDLELET

Na Figura 6.197 é apresentada a estatística de Jarque-Bera, que rejeita fortemente a hipótese de distribuição normal.

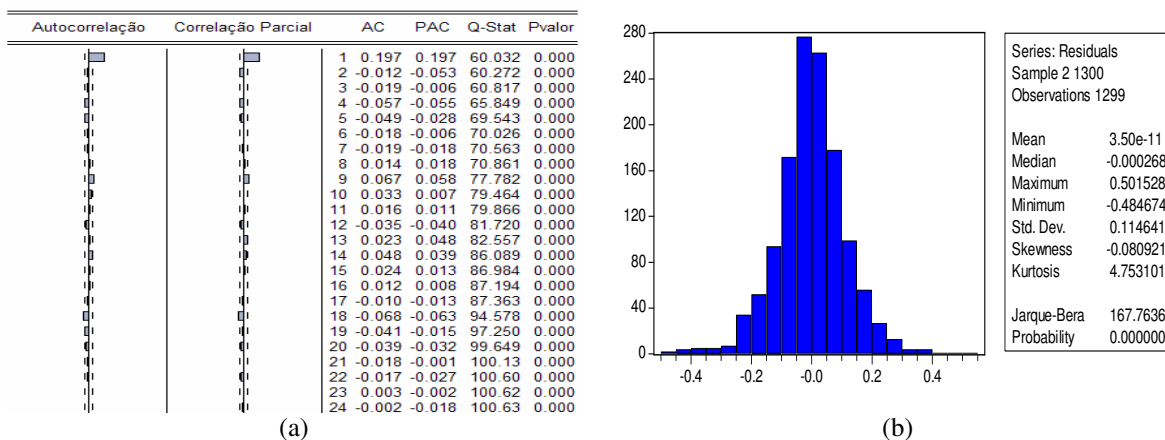


Figura 6.197 (a) Correlograma da série NDLELET, (b) Estatísticas dos resíduos

Os resultados para o teste BDS são apresentados na Tabela 6.93.

BDS				
Dimensão	Estatística	Erro-padrão	Estatística-z	p-valor
2	0,016015	0,002119	7,558092	0,0000
3	0,027794	0,003360	8,271023	0,0000
4	0,037410	0,003993	9,368623	0,0000
5	0,044242	0,004153	10,65270	0,0000
6	0,046243	0,003997	11,57040	0,0000

Tabela 6.93 Resultados obtidos pelo teste BDS nas respectivas dimensões.

O melhor modelo ajustado para a série é um AR(1)-GARCH(1,1). Os parâmetros estimados do modelo para a série são mostrados na Tabela 6.94.

Parâmetros	Coefficiente	Erro-padrão	Estatística-z	p-valor
ϕ_0	0,479898	0,003537	135,6610	0,0000
ϕ_1	0,182134	0,028272	6,442219	0,0000
α_0	0,000546	0,000168	3,259575	0,0011
α_1	0,080313	0,012129	6,621634	0,0000
β_1	0,877888	0,021345	41,12861	0,0000

Tabela 6.94 Parâmetros estimados para a série NDLELET

O resultado do teste Ljung-Box para a série NDLELET é apresentado na Figura 6.198.

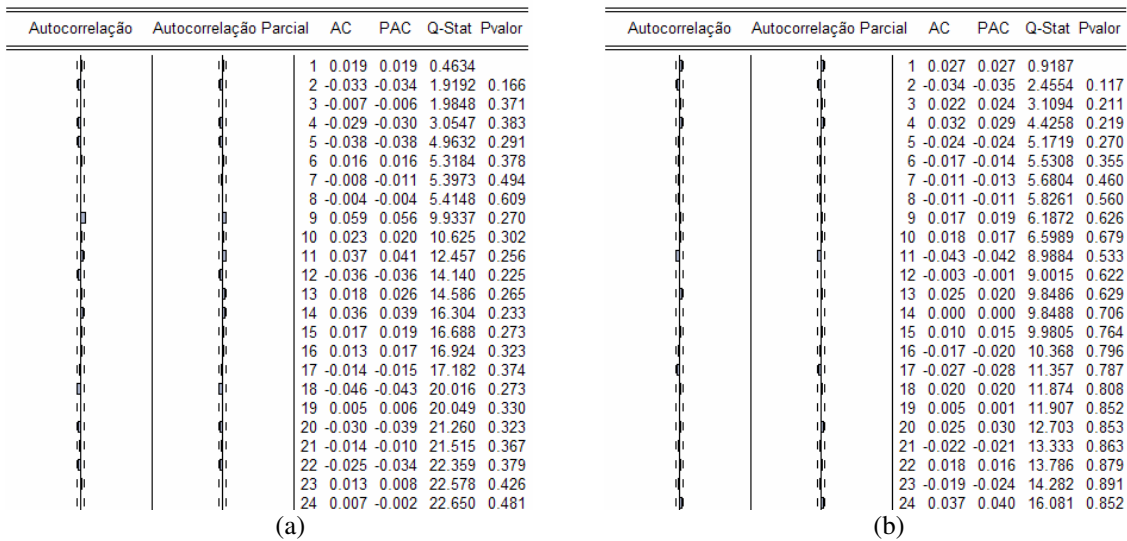


Figura 6.198 (a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado

As previsões realizadas utilizando o modelo ARIMA-GARCH são mostradas na Figura 6.199. A linha contínua indica o valor da série NDLELET e a linha central com marcador quadrado indica as previsões.

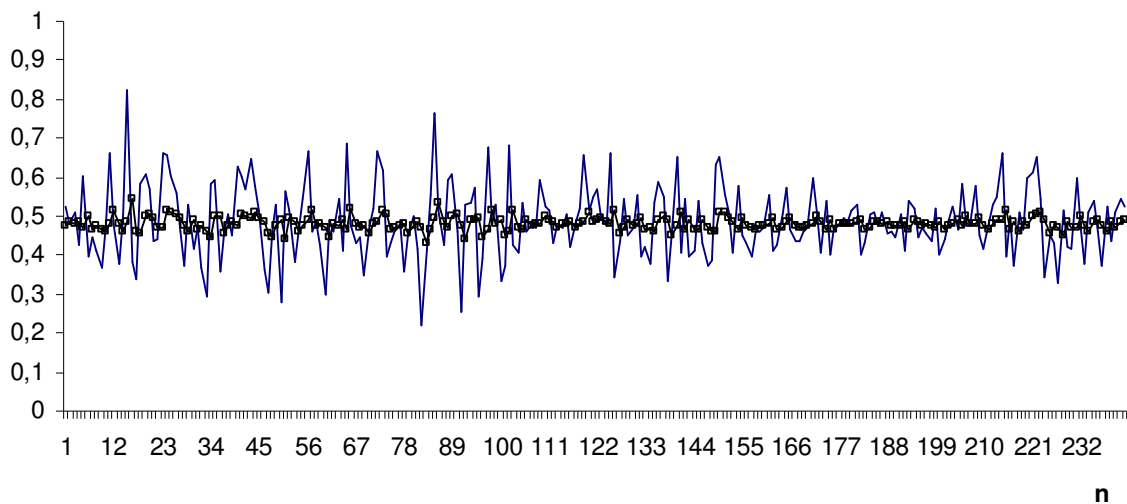


Figura 6.199 Previsão obtida pelo modelo ARIMA-GARCH para NDLELET

Na Figura 6.200a é mostrada a previsão da volatilidade e na Figura 6.200b a série original e o intervalo de previsão para a série NDLELET.

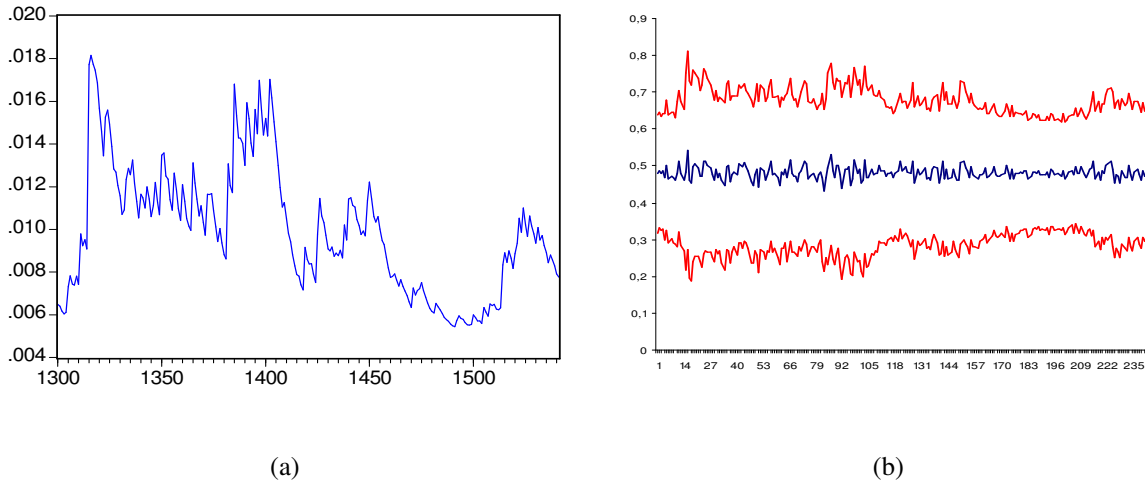


Figura 6.200 (a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLELET

6.5.2.2 Análise da série da Eletropaulo – Método RNA *feedforward*

Para a série NDLELET os resultados obtidos pelas diferentes topologias de RNA são apresentados na Tabela 6.95.

Topologia	FIC	FIP	TIC	RMSE
RNA 10-1-1	120	0	0,0938675	0,0915504
RNA 10-2-1	116	0	0,0937925	0,091657
RNA 10-3-1	127	0	0,094495	0,0918954
RNA 10-4-1	117	0	0,0997144	0,0971812
RNA 10-5-1	117	0	0,098843	0,0961249
RNA 10-8-1	87	0	0,0962183	0,0935042

Tabela 6.95 Estatísticas para diferentes topologias de RNA para a série

Nesse caso foi escolhida a RNA 10-2-1.

Na Figura 6.201 são apresentadas as previsões obtidas pela RNA 10-2-1 para a série NDLELET.

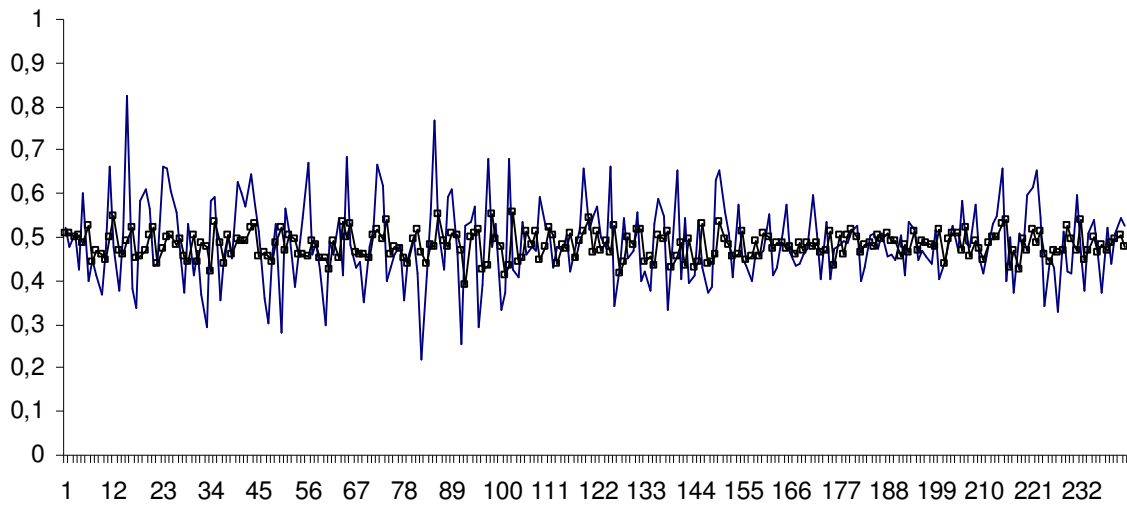


Figura 6.201 Previsão obtida pela RNA *feedforward* para NDLELET

Na Figura 6.202a é mostrado o intervalo de confiança para as previsões obtidas pela RNA 10-2-1 e na Figura 6.202b o intervalo de predição para a série NDLELET.

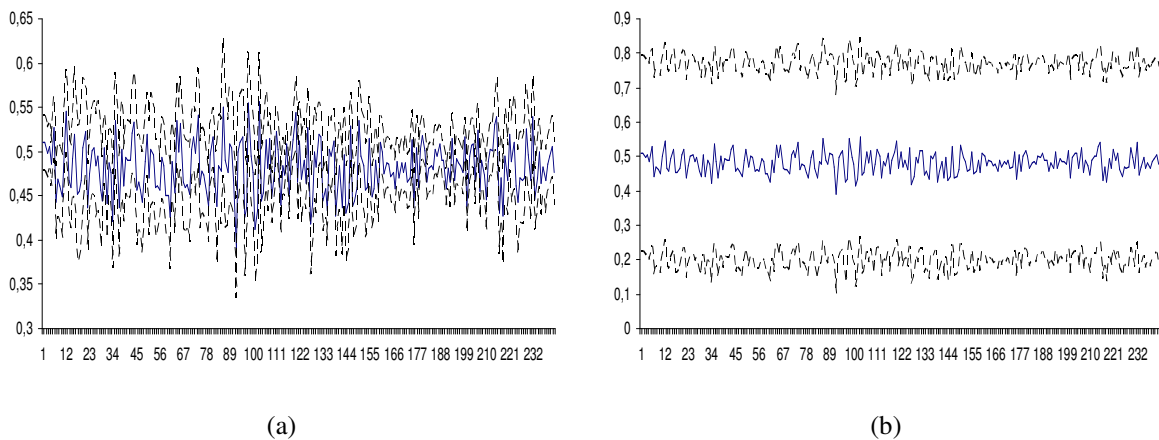


Figura 6.202 (a) Intervalo de confiança, (b) Intervalo de predição para NDLELET

Na Figura 6.203a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA 10-2-1 e na Figura 6.203b o histograma desses resíduos.

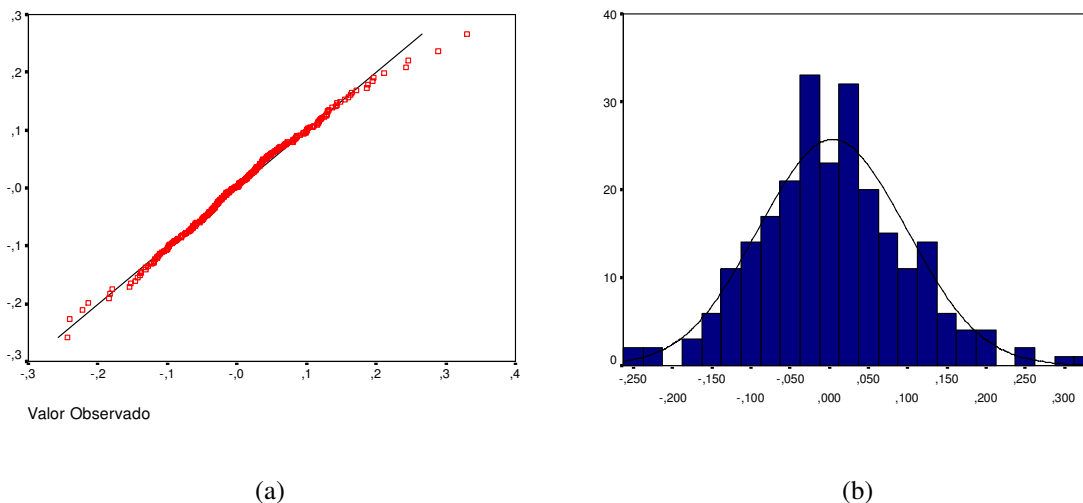


Figura 6.203 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA *feedforward* para NDLELET

6.5.2.3 Análise da série da Eletropaulo – Método RNA EKF

As previsões, para a série NDLELET, obtidas pela RNA treinada com o filtro de Kalman estendido (EKF) são mostradas na Figura 6.204.

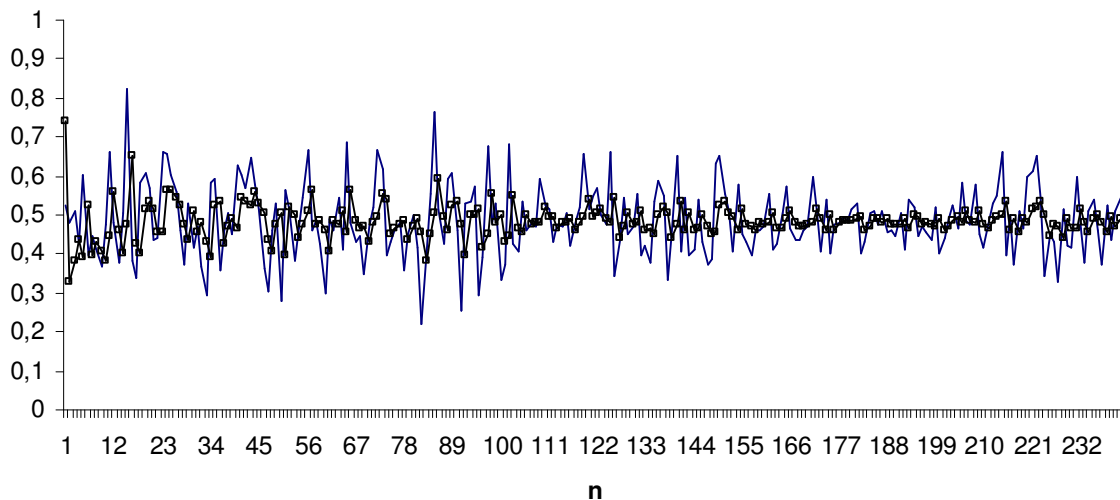
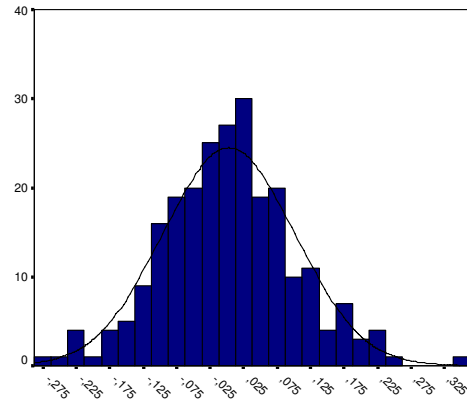
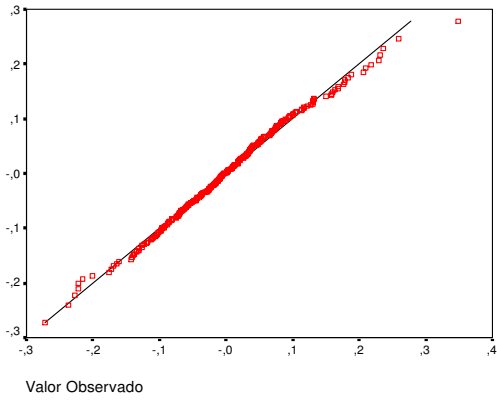


Figura 6.204 Previsão obtida pela RNA EKF para NDLELET

Na Figura 6.205a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman estendido e na Figura 6.205b o histograma desses resíduos.



(a)

(b)

Figura 6.205 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLELET

6.5.2.4 Análise da série da Eletropaulo – Método RNA UKF

As previsões obtidas pela RNA treinada com o filtro de Kalman *unscented* são mostradas na Figura 6.206.

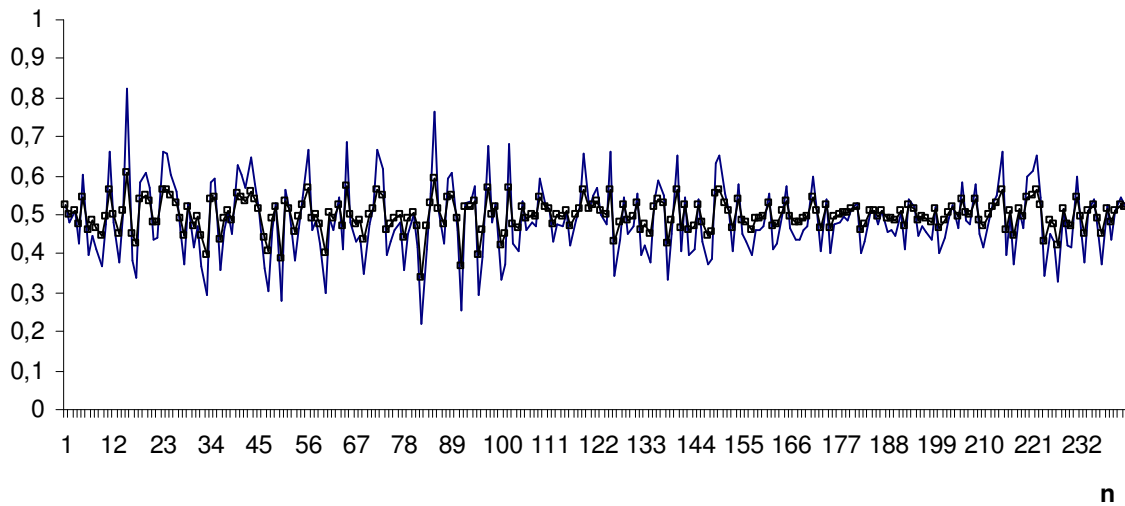


Figura 6.206 Previsão obtida pela RNA UKF para NDLELET

Na Figura 6.207a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman *unscented* e na Figura 6.207b o histograma desses resíduos.

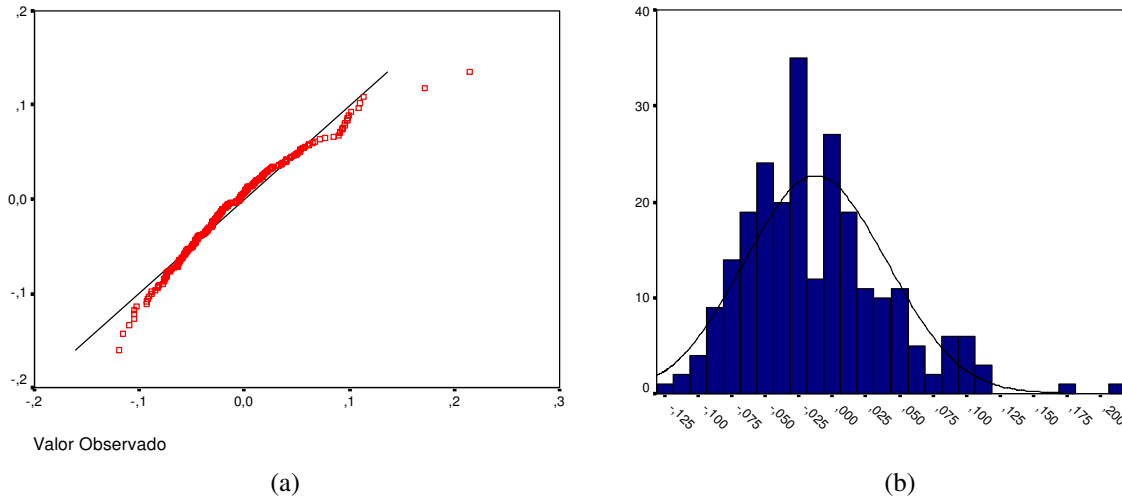


Figura 6.207 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLELET

As estatísticas de previsão obtidas pelos modelos ARIMA-GARCH, RNA *feedforward*, RNA EKF e RNA UKF são apresentadas na Tabela 6.96 e na Figura 6.208.

Método	TIC	RMSE
AR(1)-GARCH(1,1)	0,093865	0,091593
RNA 10-2-1	0,093793	0,091657
RNA EKF	0,100200	0,098143
RNA UKF	0,054597	0,054301

Tabela 6.96 Estatísticas de previsão para o método de ARIMA-GARCH e de RNA

Pode-se verificar que as melhores previsões foram obtidas pelo método que empregou redes neurais treinadas com o filtro de Kalman *unscented*.

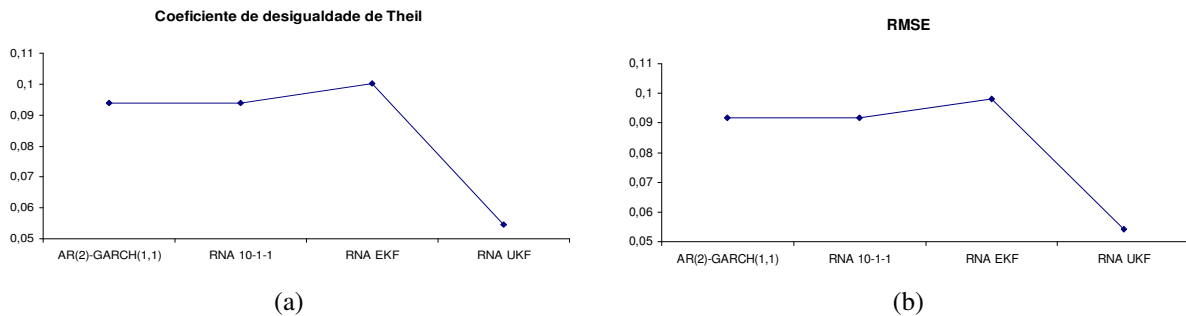


Figura 6.208 Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b) RMSE

6.5.3 Análise da série da Eletrobras

A seguir são apresentados os resultados obtidos para a série da Eletrobras.

6.5.3.1 Análise da série da Eletrobras – Método ARIMA-GARCH

Na Figura 6.209a é apresentada a série original do preço de fechamento da Eletrobras, na Figura 6.209b é apresentada a série normalizada da primeira diferença do logaritmo do preço, NDLELBR.

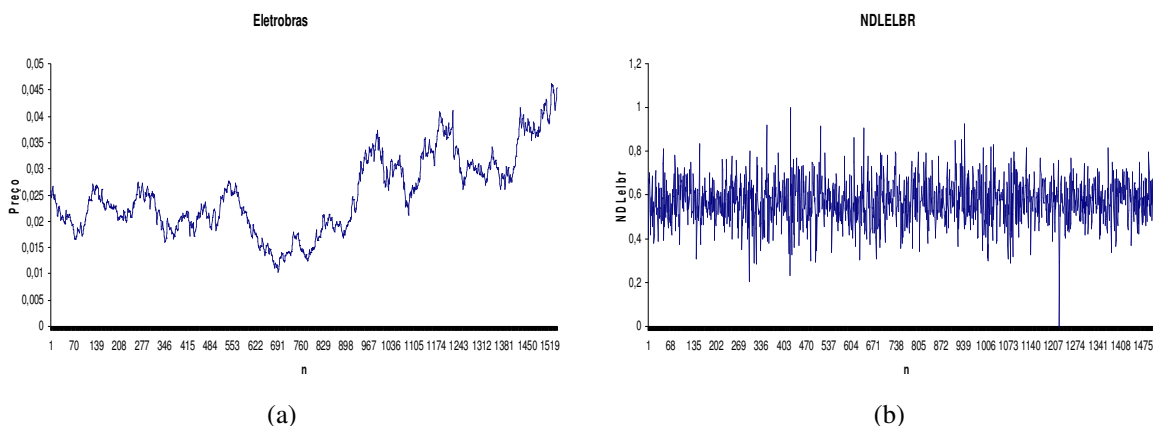


Figura 6.209 (a) Série no nível do preço do Bradesco, (b) Série transformada NDLELBR

O resultado do teste ADF, para a série original, é apresentado na Tabela 6.97.

Estatística do teste ADF para série original	-1,13314	valores críticos 1%	-3,43439
		valores críticos 5%	-2,86321
		valores críticos 10%	-2,56771

Tabela 6.97 Teste ADF para série original da Eletrobras

Não é rejeitada a hipótese de raiz unitária para a série no nível pelo teste ADF. O teste de presença de uma raiz unitária para a série transformada apresentou os seguintes resultados:

Estatística do teste ADF para série transformada	-26,6516	valores críticos 1%	-3,43439
		valores críticos 5%	-2,86321
		valores críticos 10%	-2,56771

Tabela 6.98 Teste ADF para série NDLELBR

Na Figura 6.210 é apresentada a estatística de Jarque-Bera, que rejeita fortemente a hipótese de distribuição normal.

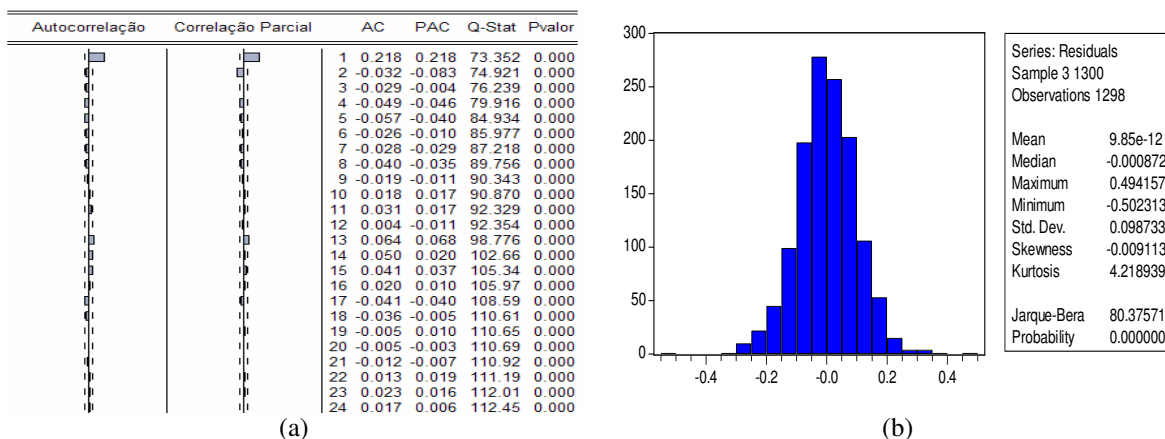


Figura 6.210 (a) Correlograma da série NDLELBR, (b) Estatísticas dos resíduos

Os resultados para o teste BDS são apresentados na Tabela 6.99.

BDS				
Dimensão	Estatística	Erro-padrão	Estatística-z	p-valor
2	0,012534	0,001814	6,910698	0,0000
3	0,018995	0,002871	6,617120	0,0000
4	0,021510	0,003403	6,320082	0,0000
5	0,022170	0,003532	6,277582	0,0000
6	0,022176	0,003391	6,540506	0,0000

Tabela 6.99 Resultados obtidos pelo teste BDS nas respectivas dimensões.

O melhor modelo ajustado para a série é um AR(2)-GARCH(1,1). Os parâmetros estimados do modelo para a série são mostrados na Tabela 6.100.

Parâmetros	Coefficiente	Erro-padrão	Estatística-z	p-valor
ϕ_0	0,572616	0,003170	180,6157	0,0000
ϕ_1	0,252360	0,029249	8,627887	0,0000
ϕ_2	-0,087730	0,030854	-2,843418	0,0045
α_0	0,002080	0,000725	2,869515	0,0041
α_1	0,100547	0,018394	5,466323	0,0000
β_1	0,686305	0,083184	8,250424	0,0000

Tabela 6.100 Parâmetros estimados para a série NDLELBR

O resultado do teste Ljung-Box para a série NDLELBR é apresentado na Figura 6.211.

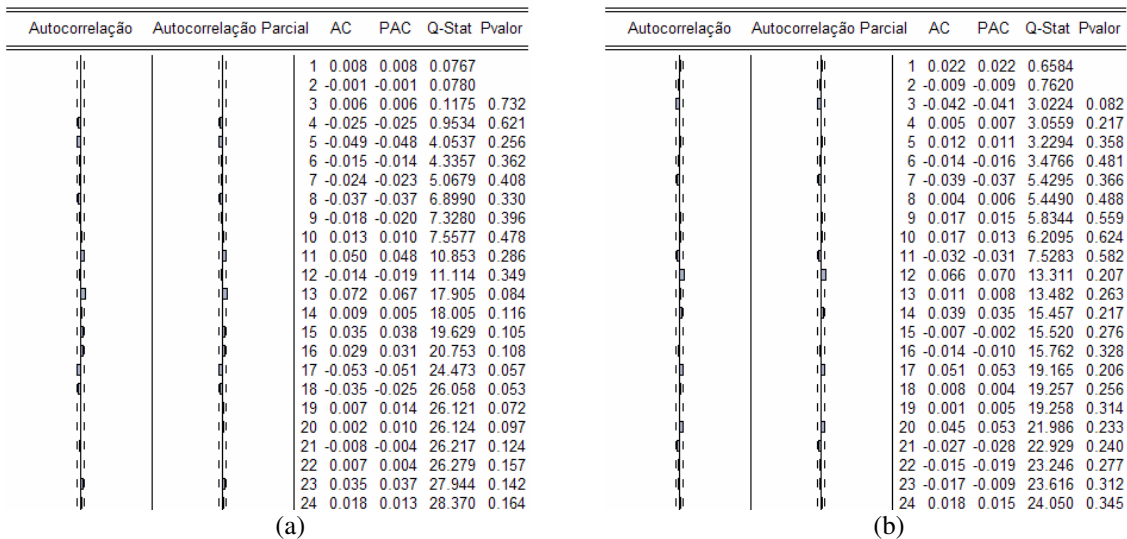


Figura 6.211 (a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado

As previsões realizadas utilizando o modelo ARIMA-GARCH são mostradas na Figura 6.212. A linha contínua indica o valor da série NDLELBR e a linha central com marcador quadrado indica as previsões.

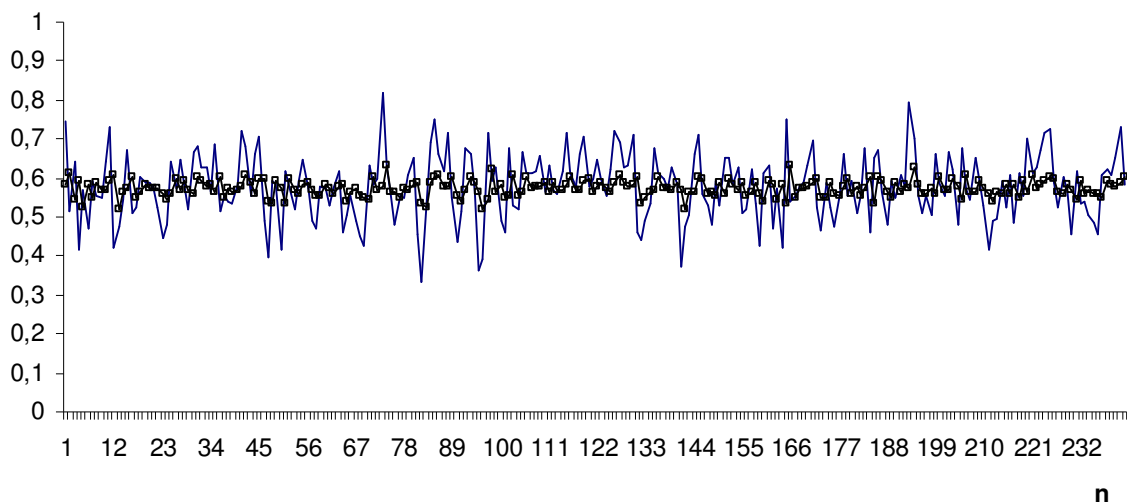


Figura 6.212 Previsão obtida pelo modelo ARIMA-GARCH para NDLELBR

Na Figura 6.213a é mostrada a previsão da volatilidade e na Figura 6.213b a série original e o intervalo de previsão para a série NDLELBR.

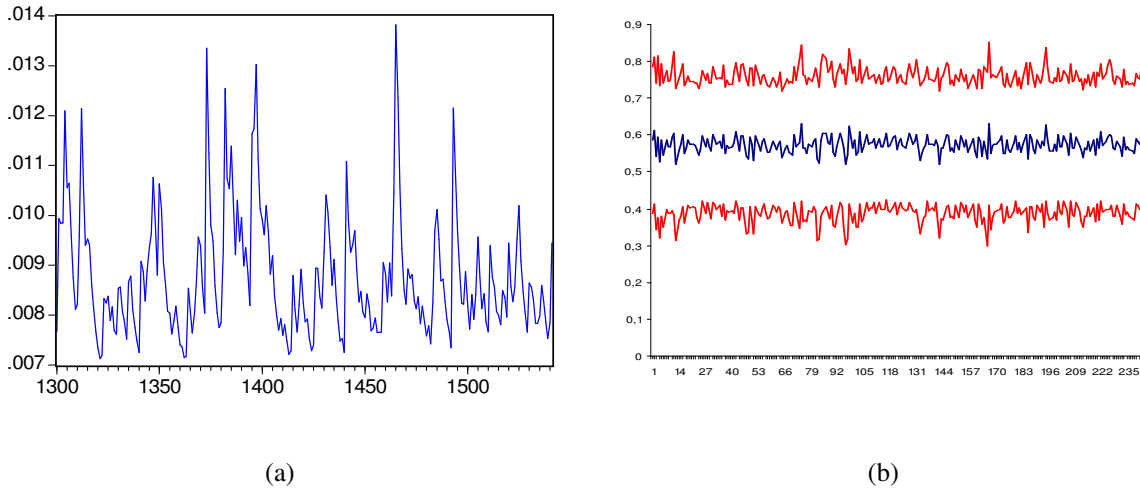


Figura 6.213 (a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLELBR

6.5.3.2 Análise da série da Eletrobras – Método RNA *feedforward*

Para a série NDLELBR os resultados obtidos pelas diferentes topologias de RNA são apresentados na Tabela 6.101.

Topologia	FIC	FIP	TIC	RMSE
RNA 10-1-1	130	0	0,0711881	0,0822516
RNA 10-2-1	109	0	0,0729902	0,0842609
RNA 10-3-1	107	0	0,070557	0,0815157
RNA 10-4-1	110	0	0,0712598	0,0821884
RNA 10-5-1	109	0	0,0734707	0,0849946
RNA 10-8-1	100	0	0,0763056	0,0881415

Tabela 6.101 Estatísticas para diferentes topologias de RNA para a série

A Figura 6.214 apresenta as previsões obtidas pela RNA 10-3-1 para a série NDLELBR.

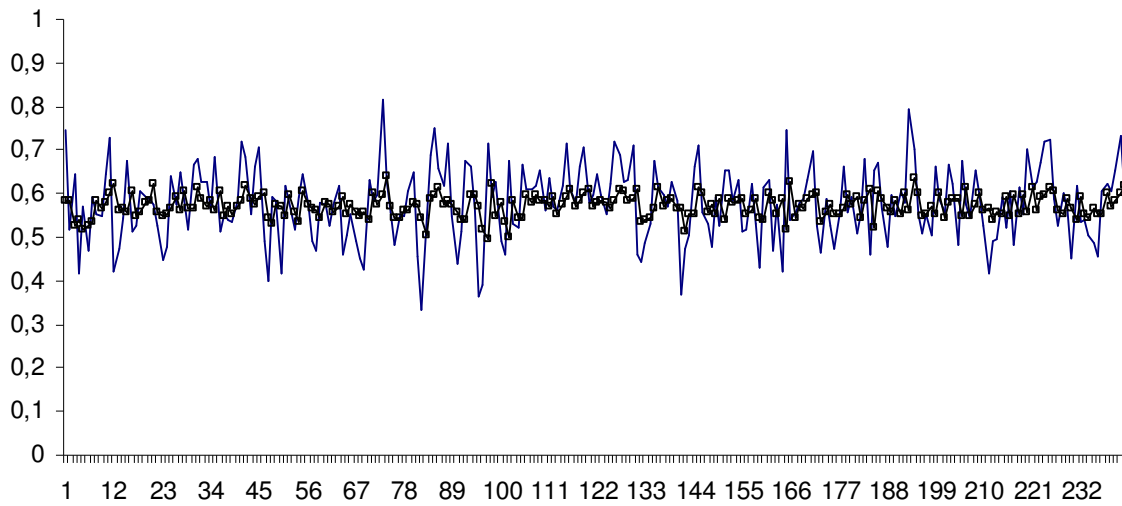
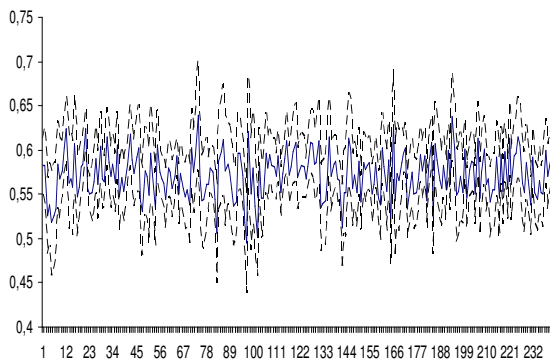
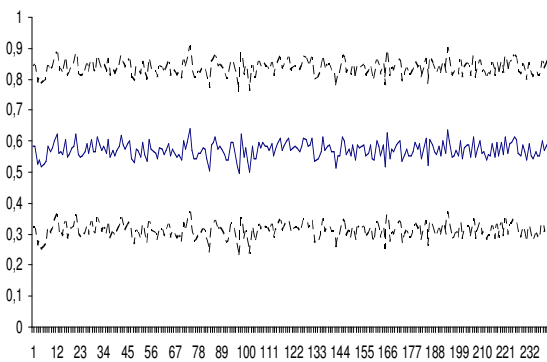


Figura 6.214 Previsão obtida pela RNA *feedforward* para NDLELBR

Na Figura 6.215a é mostrado o intervalo de confiança para as previsões obtidas pela RNA 10-3-1 e na Figura 6.215b o intervalo de predição para a série NDLELBR.



(a)



(b)

Figura 6.215 (a) Intervalo de confiança, (b) Intervalo de predição para NDLELBR

Na Figura 6.216a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA 10-3-1 e na Figura 6.216b o histograma desses resíduos.

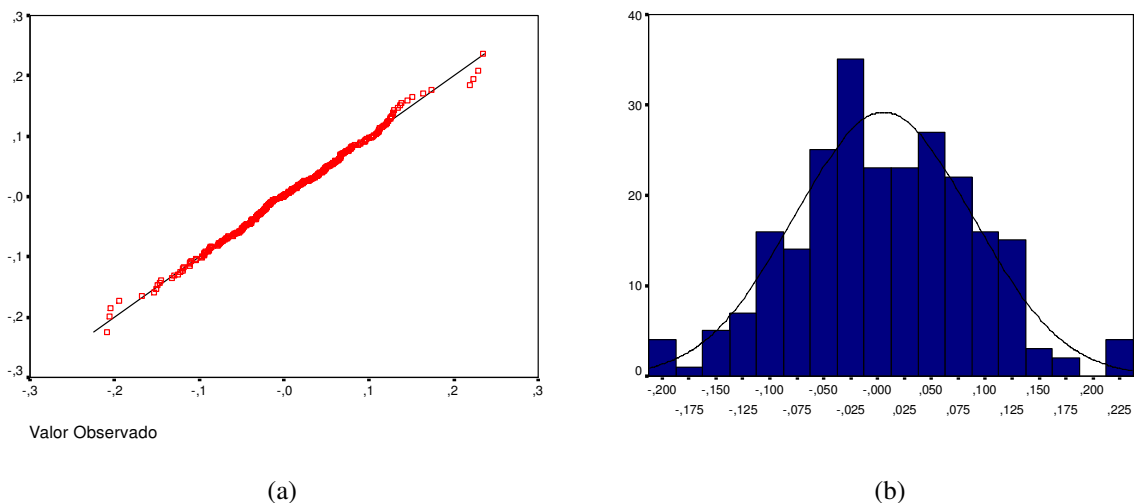


Figura 6.216 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA *feedforward* para NDLELBR

6.5.3.3 Análise da série da Eletrobras – Método RNA EKF

As previsões, para a série NDLELBR, obtidas pela RNA treinada com o filtro de Kalman estendido (EKF) são mostradas na Figura 6.217.

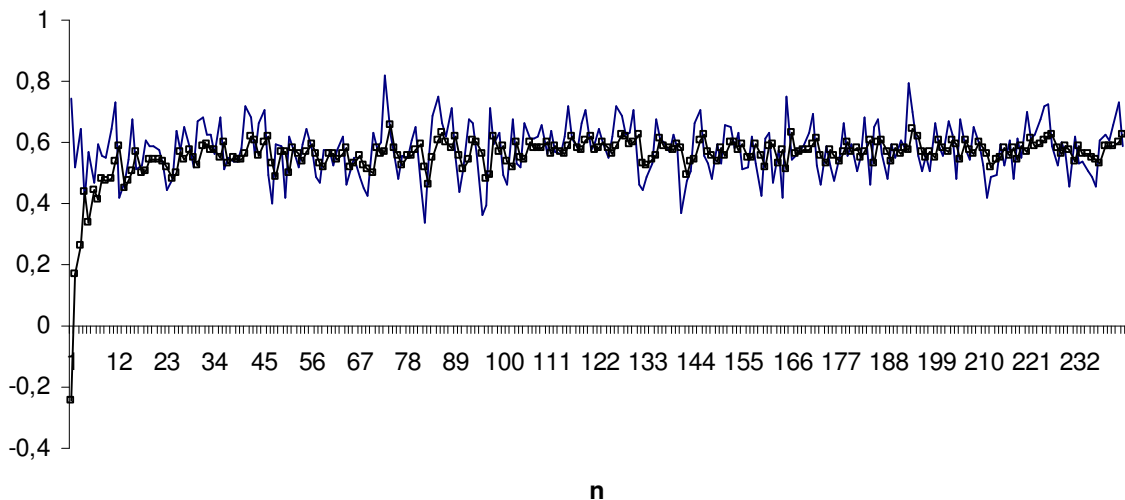


Figura 6.217 Previsão obtida pela RNA EKF para NDLELBR

Na Figura 6.218a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman estendido e na Figura 6.218b o histograma desses resíduos.

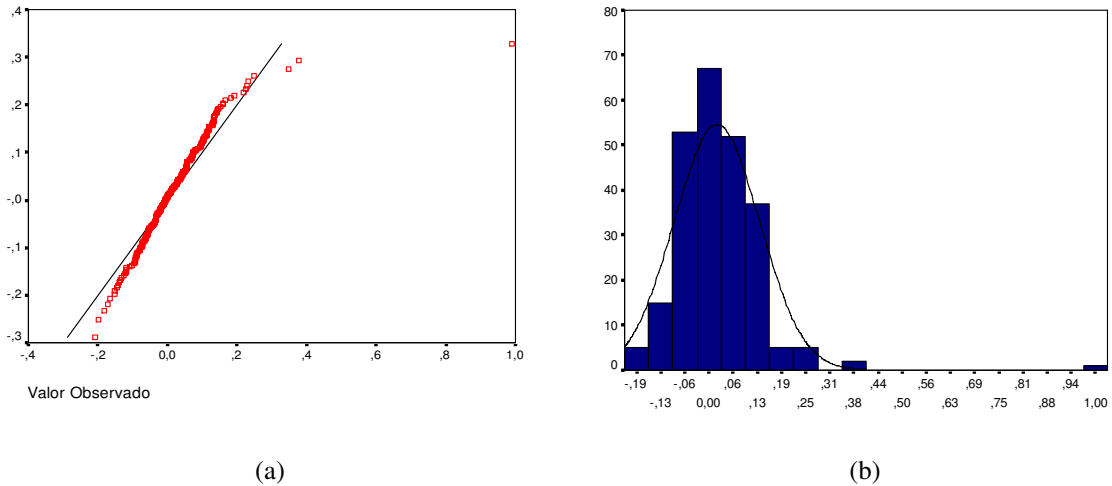


Figura 6.218 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLELBR

6.5.3.4 Análise da série da Eletrobras – Método RNA UKF

As previsões obtidas pela RNA treinada com o filtro de Kalman *unscented* são mostradas na Figura 6.219.

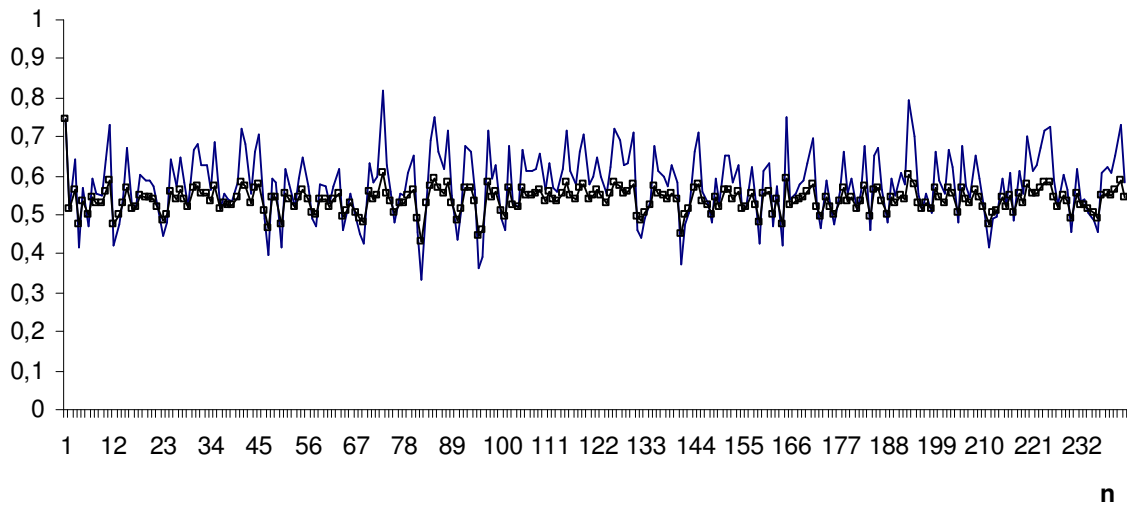
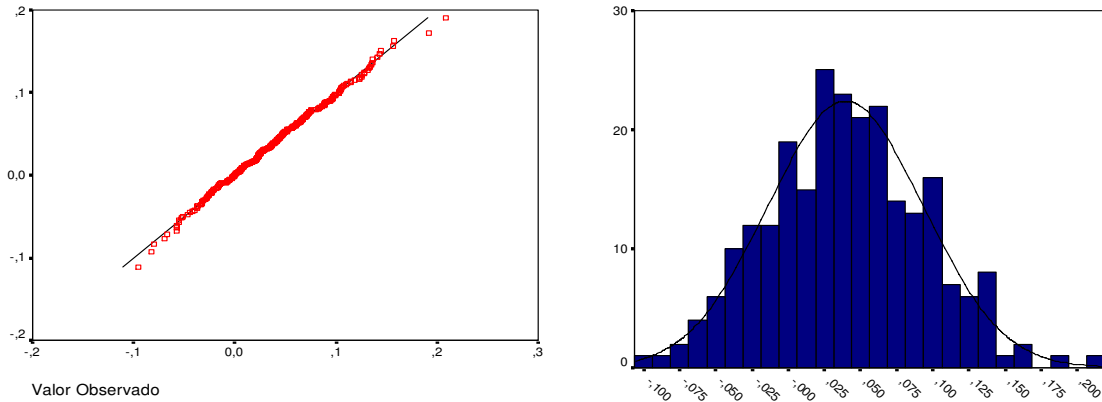


Figura 6.219 Previsão obtida pela RNA UKF para NDLELBR

Na Figura 6.220a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman *unscented* e na Figura 6.220b o histograma desses resíduos.



(a)

(b)

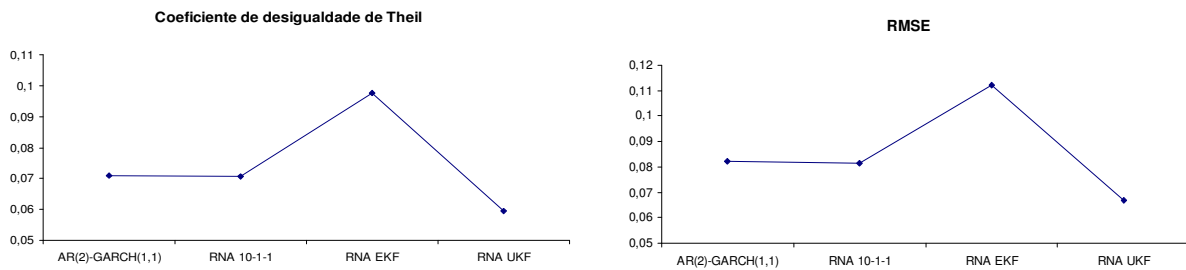
Figura 6.220 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLELBR

As estatísticas de previsão obtidas pelos modelos ARIMA-GARCH, RNA *feedforward*, RNA EKF e RNA UKF são apresentadas na Tabela 6.102 e na Figura 6.221.

Método	TIC	RMSE
AR(2)-GARCH(1,1)	0,070875	0,082059
RNA 10-3-1	0,070557	0,081516
RNA EKF	0,097718	0,111974
RNA UKF	0,059481	0,066792

Tabela 6.102 Estatísticas de previsão para o método de ARIMA-GARCH e de RNA

Pode-se verificar que as melhores previsões foram obtidas pelo método que empregou redes neurais treinadas com o filtro de Kalman *unscented*.



(a)

(b)

Figura 6.221 Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b)

RMSE

6.5.4 Análise da série da Brasil Telecom

A seguir são apresentados os resultados obtidos para a série da Brasil Telecom.

6.5.4.1 Análise da série da Brasil Telecom – Método ARIMA-GARCH

Na Figura 6.222 é apresentada a série original do preço de fechamento da Brasil Telecom, na Figura 6.222b é apresentada a série normalizada da primeira diferença do logaritmo do preço, NDLBRAS.

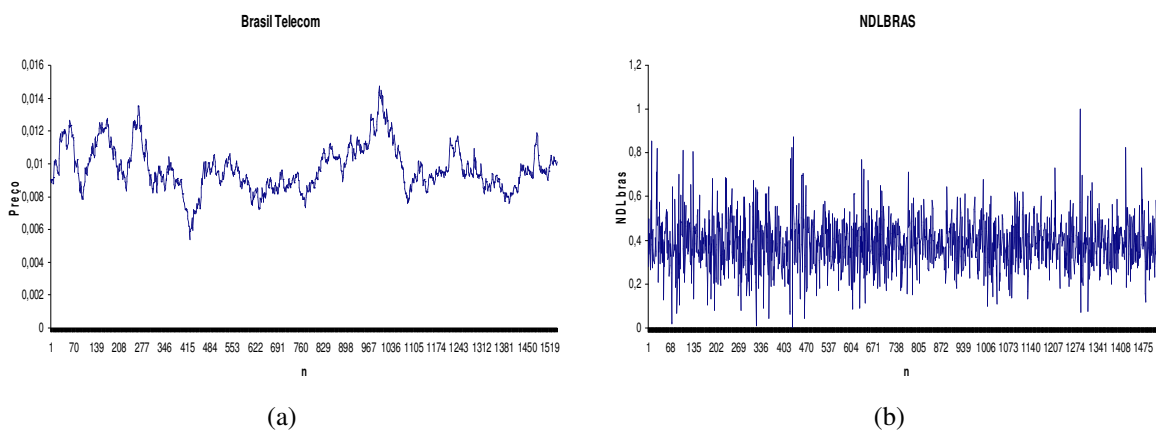


Figura 6.222 (a) Série no nível do preço do Bradesco, (b) Série transformada NDLBRAS

O resultado do teste ADF, para a série original, é apresentado na Tabela 6.103.

Estatística do teste ADF para série original		valores críticos 1%	-3,43439
	-3,81189	valores críticos 5%	-2,86321
		valores críticos 10%	-2,56771

Tabela 6.103 Teste ADF para série original da Brasil Telecom

Não é rejeitada a hipótese de raiz unitária para a série no nível pelo teste ADF. O teste de presença de uma raiz unitária para a série transformada apresentou os seguintes resultados:

Estatística do teste ADF para série transformada		valores críticos 1%	-3,43439
	-28,16046	valores críticos 5%	-2,86321
		valores críticos 10%	-2,56771

Tabela 6.104 Teste ADF para série NDLBRAS

Na Figura 6.223 é apresentada a estatística de Jarque-Bera, que rejeita fortemente a hipótese de distribuição normal.

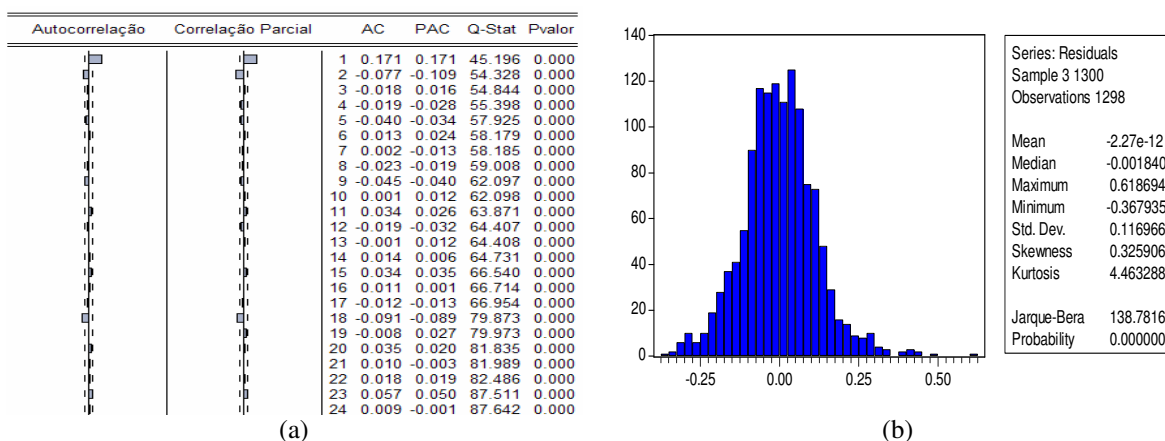


Figura 6.223 (a) Correlograma da série NDLBRAS, (b) Estatísticas dos resíduos

Os resultados para o teste BDS são apresentados na Tabela 6.105.

BDS				
Dimensão	Estatística	Erro-padrão	Estatística-z	p-valor
2	0,008259	0,001957	4,220164	0,0000
3	0,016136	0,003097	5,210015	0,0000
4	0,021663	0,003672	5,898998	0,0000
5	0,024450	0,003811	6,415727	0,0000
6	0,024963	0,003659	6,822369	0,0000

Tabela 6.105 Resultados obtidos pelo teste BDS nas respectivas dimensões.

O melhor modelo ajustado para a série é um AR(2)-GARCH(1,1). Os parâmetros estimados do modelo para a série são mostrados na Tabela 6.106.

Parâmetros	Coefficiente	Erro-padrão	Estatística-z	p-valor
ϕ_0	0,388152	0,003510	110,5770	0,0000
ϕ_1	0,204859	0,029253	7,003084	0,0000
ϕ_2	-0,093952	0,027331	-3,437566	0,0006
α_0	0,000411	0,000170	2,422375	0,0154
α_1	0,038960	0,008605	4,527686	0,0000
β_1	0,930990	0,018791	49,54395	0,0000

Tabela 6.106 Parâmetros estimados para a série NDLBRAS

O resultado do teste Ljung-Box para a série NDLBRAS é apresentado na Figura 6.224.

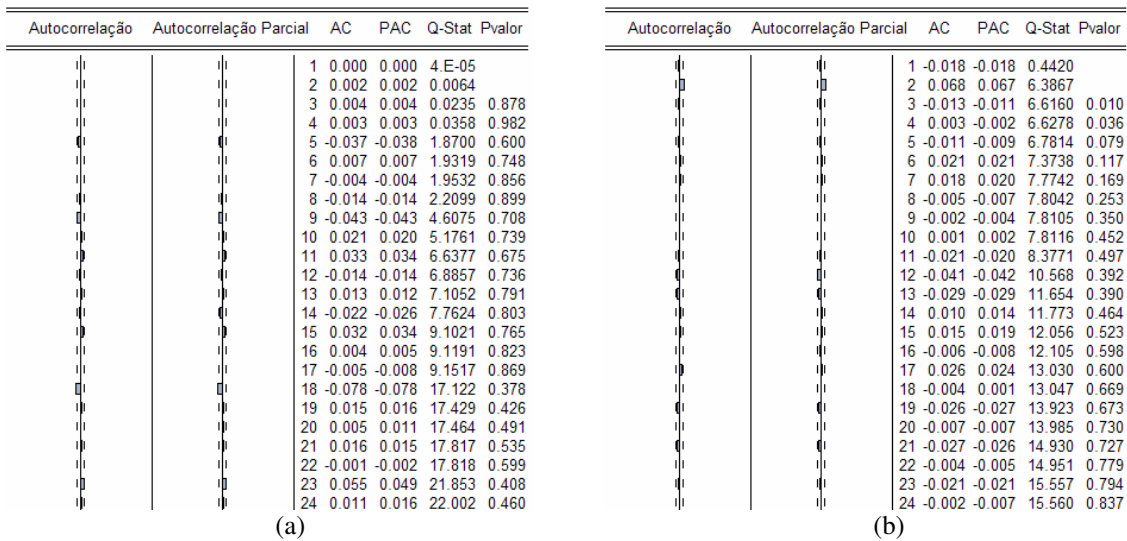


Figura 6.224 (a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado

As previsões realizadas utilizando o modelo ARIMA-GARCH são mostradas na Figura 6.225. A linha contínua indica o valor da série NDLBRAS e a linha central com marcador quadrado indica as previsões.

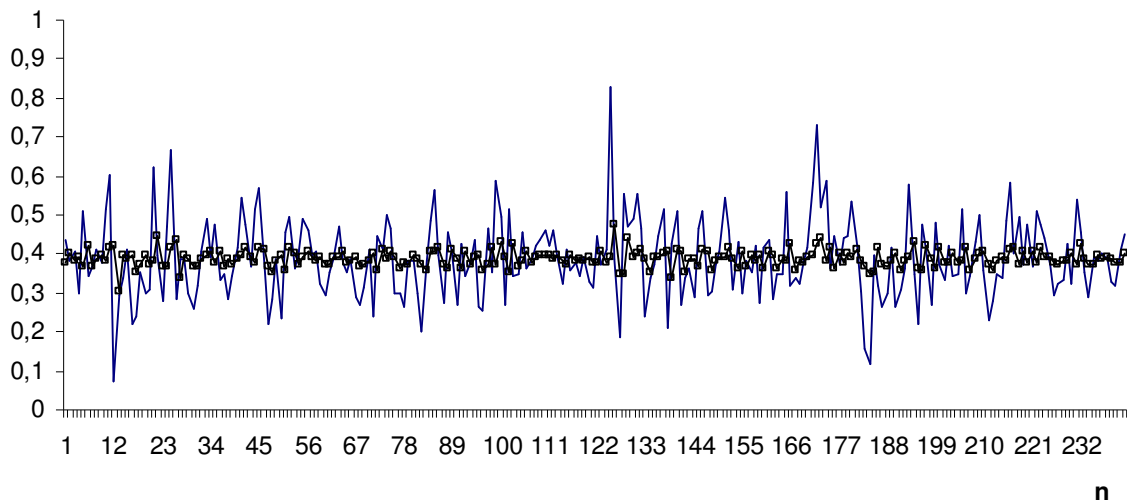


Figura 6.225 Previsão obtida pelo modelo ARIMA-GARCH para NDLBRAS

Na Figura 6.226a é mostrada a previsão da volatilidade e na Figura 6.226b a série original e o intervalo de previsão para a série NDLBRAS.

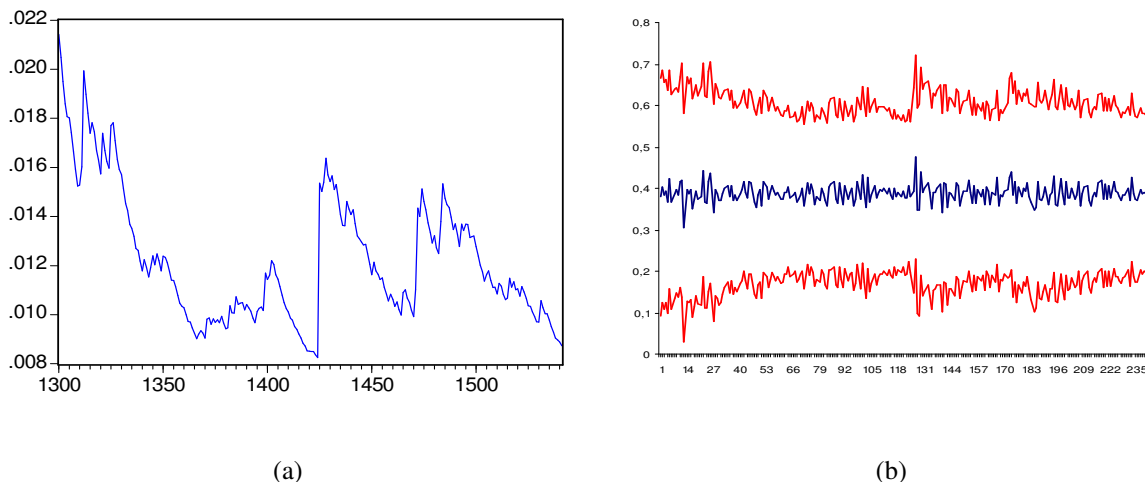


Figura 6.226 (a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLBRAS

6.5.4.2 Análise da série da Brasil Telecom– Método RNA *feedforward*

Para a série NDLBRAS os resultados obtidos pelas diferentes topologias de RNA são apresentados na Tabela 6.107.

Topologia	FIC	FIP	TIC	RMSE
RNA 10-1-1	129	1	0,124492	0,0984932
RNA 10-2-1	128	1	0,125268	0,0988723
RNA 10-3-1	125	1	0,126988	0,100029
RNA 10-4-1	108	0	0,129103	0,102378
RNA 10-5-1	107	1	0,130489	0,102739
RNA 10-8-1	95	0	0,135591	0,106871

Tabela 6.107 Estatísticas para diferentes topologias de RNA para a série

Na Figura 6.227 são apresentadas as previsões obtidas pela RNA 10-1-1 para a série NDLBRAS.

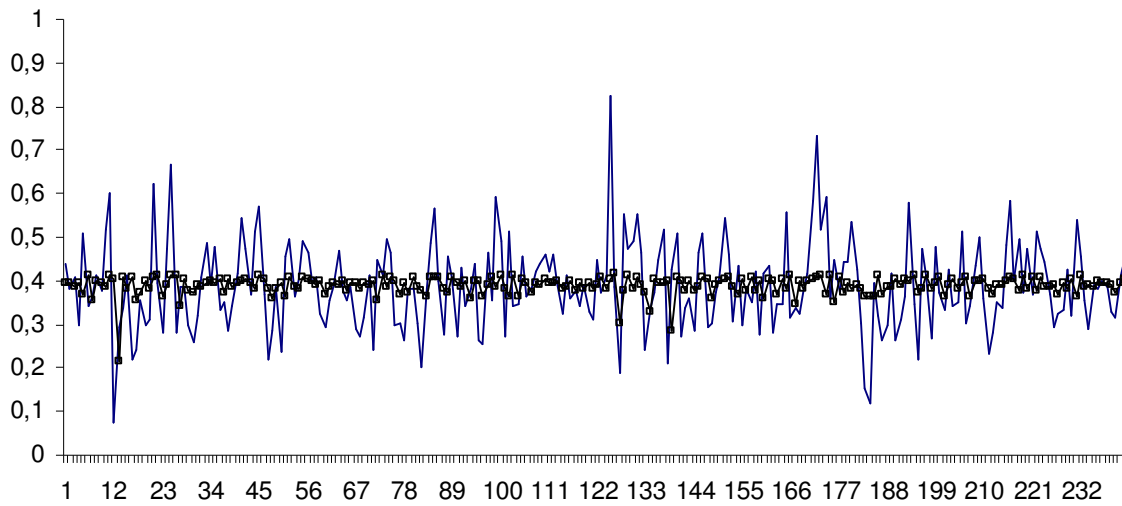


Figura 6.227 Previsão obtida pela RNA *feedforward* para NDLBRAS

Na Figura 6.228a é mostrado o intervalo de confiança para as previsões obtidas pela RNA 10-1-1 e na Figura 6.228b o intervalo de predição para a série NDLBRAS.

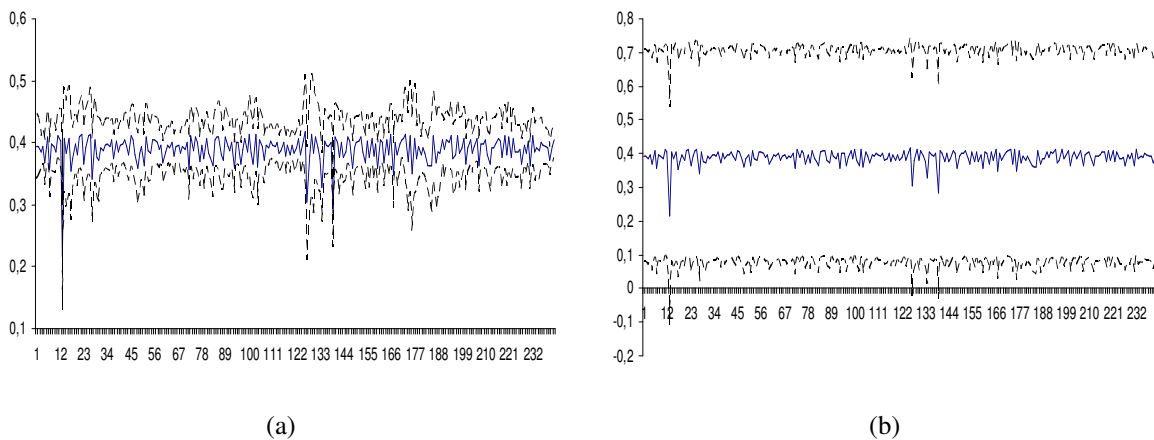


Figura 6.228 (a) Intervalo de confiança, (b) Intervalo de predição para NDLBRAS

Na Figura 6.229a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA 10-1-1 e na Figura 6.229b o histograma desses resíduos.

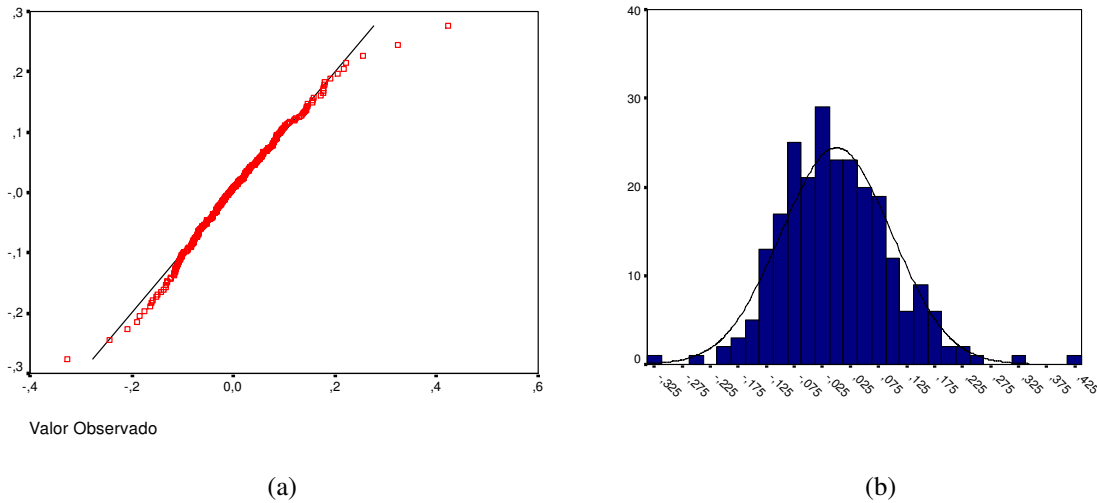


Figura 6.229 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA *feedforward* para NDLBRAS

6.5.4.3 Análise da série da Brasil Telecom – Método RNA EKF

As previsões, para a série NDLBRAS, obtidas pela RNA treinada com o filtro de Kalman estendido (EKF) são mostradas na Figura 6.230.

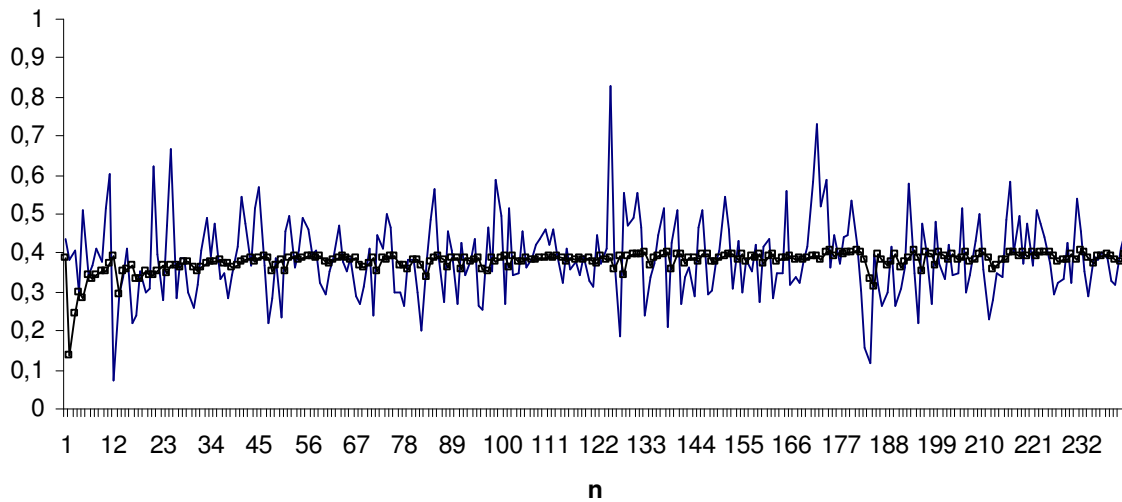


Figura 6.230 Previsão obtida pela RNA EKF para NDLBRAS

Na Figura 6.231a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman estendido e na Figura 6.231b o histograma desses resíduos.

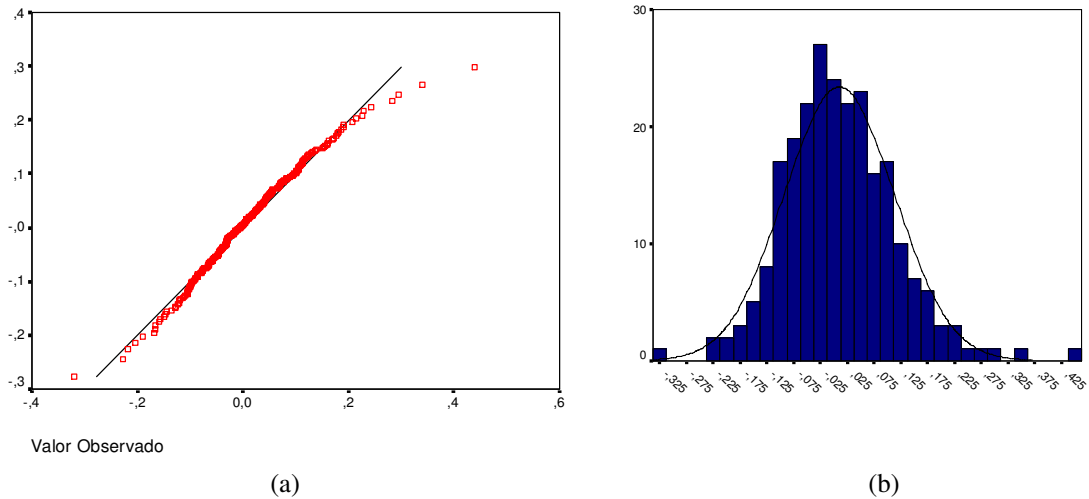


Figura 6.231 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLBRAS

6.5.4.4 Análise da série da Brasil Telecom – Método RNA UKF

As previsões obtidas pela RNA treinada com o filtro de Kalman unscented são mostradas na Figura 6.232.

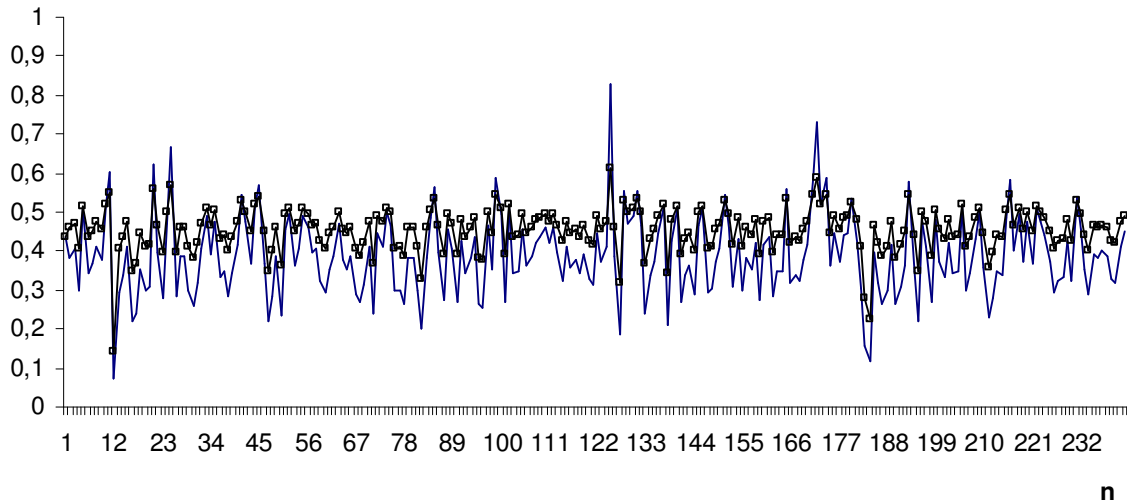


Figura 6.232 Previsão obtida pela RNA UKF para NDLBRAS

Na Figura 6.233a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman *unscented* e na Figura 6.233b o histograma desses resíduos.

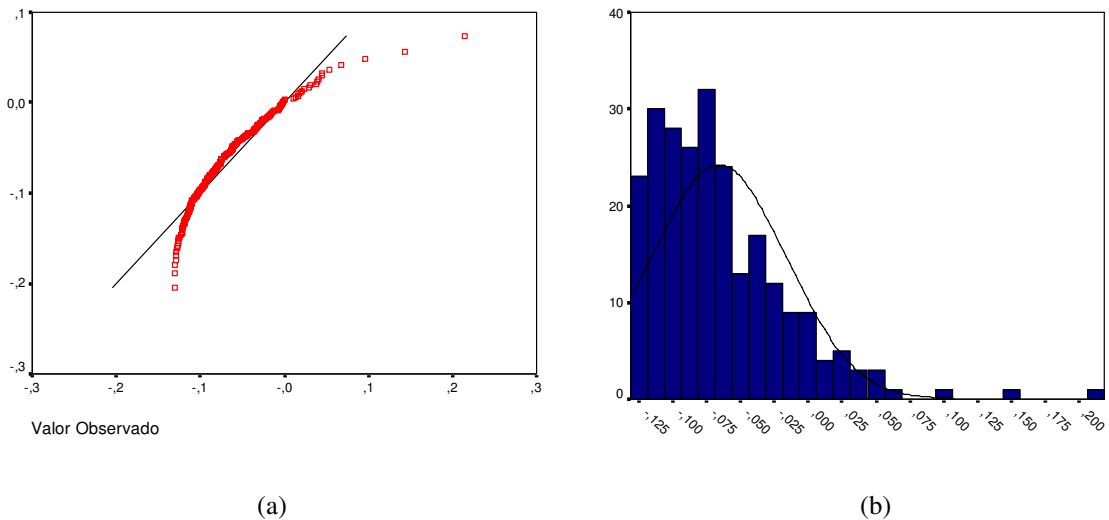


Figura 6.233 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLBRAS

As estatísticas de previsão obtidas pelos modelos ARIMA-GARCH, RNA *feedforward*, RNA EKF e RNA UKF são apresentadas na Tabela 6.108 e na Figura 6.234.

Método	TIC	RMSE
AR(2)-GARCH(1,1)	0,125325	0,099069
RNA 10-1-1	0,124492	0,0984932
RNA EKF	0,132413	0,103412
RNA UKF	0,096001	0,082556

Tabela 6.108 Estatísticas de previsão para o método de ARIMA-GARCH e de RNA

Pode-se verificar que as melhores previsões foram obtidas pelo método que empregou redes neurais treinadas com o filtro de Kalman *unscented*.

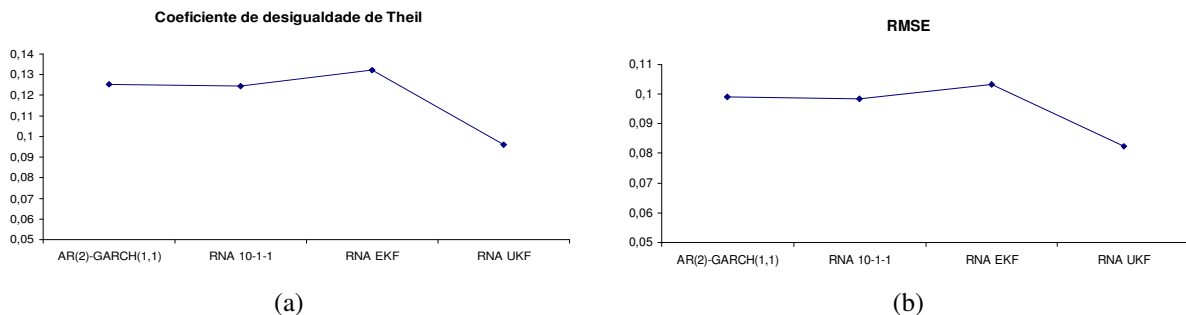


Figura 6.234 Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b) RMSE

6.5.5 Análise da série da Cesp

A seguir são apresentados os resultados obtidos para a série da Cesp.

6.5.5.1 Análise da série da Cesp– Método ARIMA-GARCH

Na Figura 6.235a é apresentada a série original do preço de fechamento da Cesp, na Figura 6.235b é apresentada a série normalizada da primeira diferença do logaritmo do preço, NDLCESP.

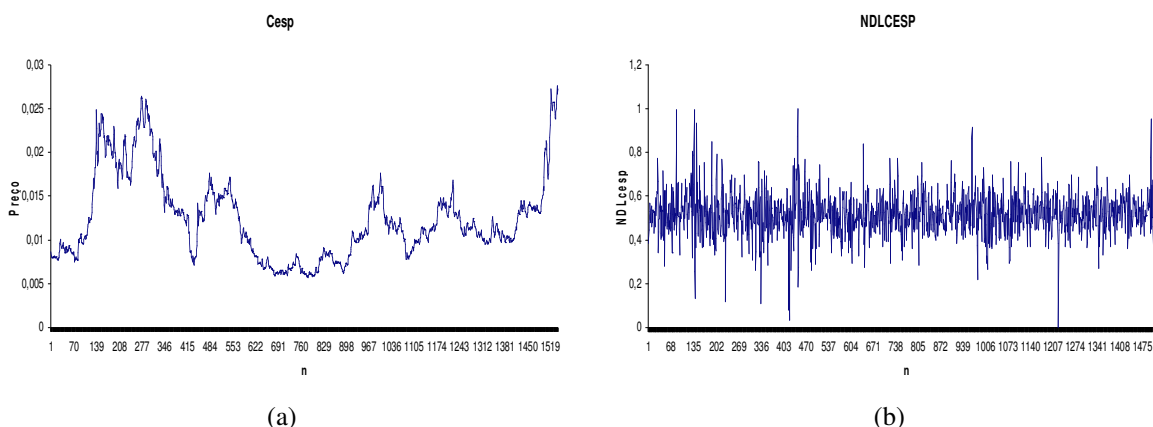


Figura 6.235 (a) Série no nível do preço do Bradesco, (b) Série transformada NDLCESP

O resultado do teste ADF, para a série original, é apresentado na Tabela 6.109.

Estatística do teste ADF para série original	valores críticos 1%	-3,43439
-1.368193	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771

Tabela 6.109 Teste ADF para série original da Cesp

Não é rejeitada a hipótese de raiz unitária para a série no nível pelo teste ADF. O teste de presença de uma raiz unitária para a série transformada apresentou os seguintes resultados:

Estatística do teste ADF para série transformada	valores críticos 1%	-3,43439
-26.92157	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771

Tabela 6.110 Teste ADF para série NDLCESP

Na Figura 6.236 é apresentada a estatística de Jarque-Bera, que rejeita fortemente a hipótese de distribuição normal.

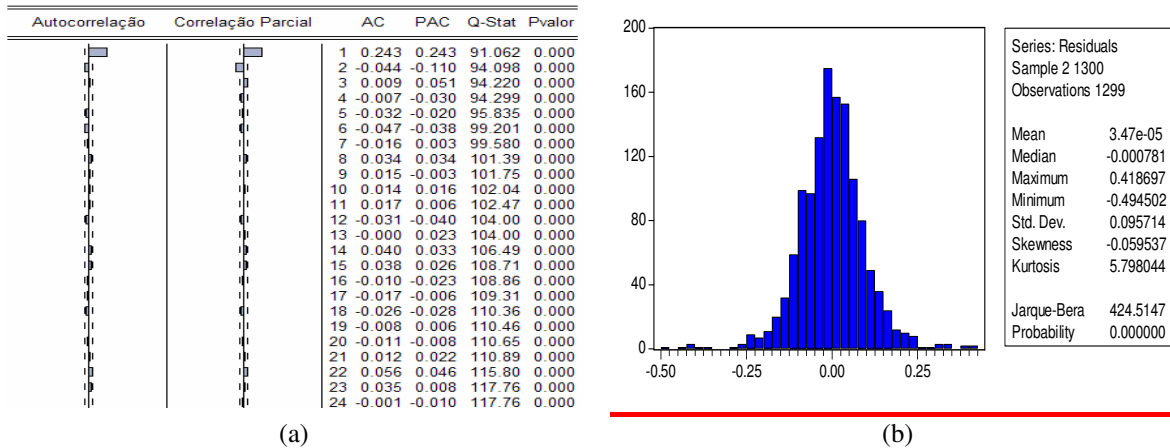


Figura 6.236 (a) Correlograma da série NDLCESP, (b) Estatísticas dos resíduos

Os resultados para o teste BDS são apresentados na Tabela 6.111.

BDS

Dimensão	Estatística	Erro-padrão	Estatística-z	p-valor
2	0,022797	0,002098	10,86603	0,0000
3	0,038759	0,003336	11,61997	0,0000
4	0,048205	0,003973	12,13189	0,0000
5	0,052348	0,004143	12,63567	0,0000
6	0,052093	0,003997	13,03440	0,0000

Tabela 6.111 Resultados obtidos pelo teste BDS nas respectivas dimensões.

O melhor modelo ajustado para a série é um AR(2)-GARCH(1,1). Os parâmetros estimados do modelo para a série são mostrados na Tabela 6.112.

Parâmetros	Coefficiente	Erro-padrão	Estatística-z	p-valor
ϕ_0	0,518540	0,003033	170,9659	0,0000
ϕ_1	0,290302	0,032055	9,056289	0,0000
ϕ_2	-0,115583	0,030625	-3,774078	0,0002
α_0	0,001792	0,000311	5,766656	0,0000
α_1	0,173883	0,026825	6,482167	0,0000
β_1	0,629868	0,049415	12,74647	0,0000

Tabela 6.112 Parâmetros estimados para a série NDLCESP

O resultado do teste Ljung-Box para a série NDLCESP é apresentado na Figura 6.237.

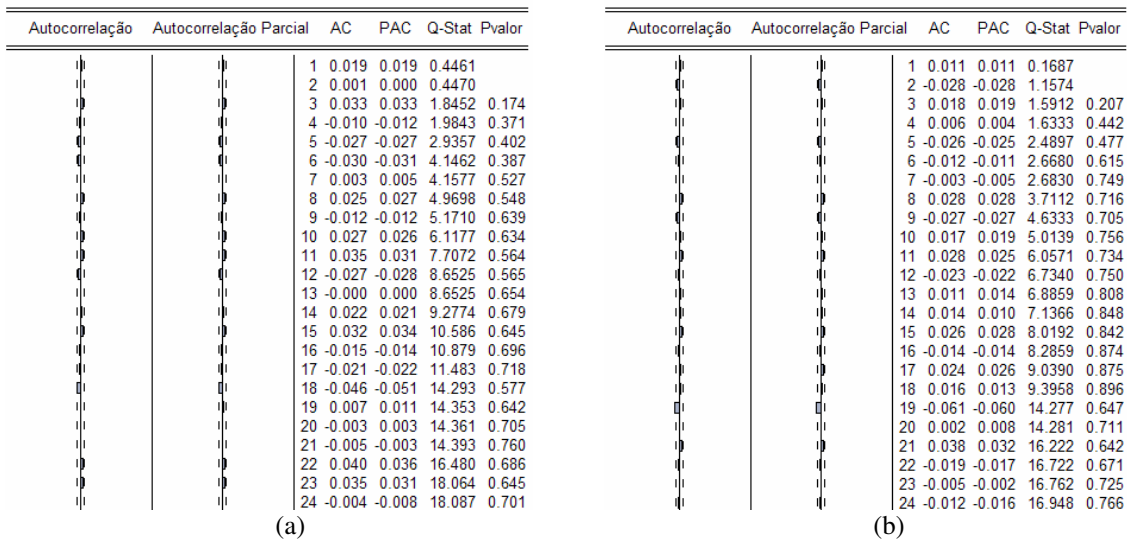


Figura 6.237 (a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado

As previsões realizadas utilizando o modelo ARIMA-GARCH são mostradas na Figura 6.238. A linha contínua indica o valor da série NDLCESP e a linha central com marcador quadrado indica as previsões.

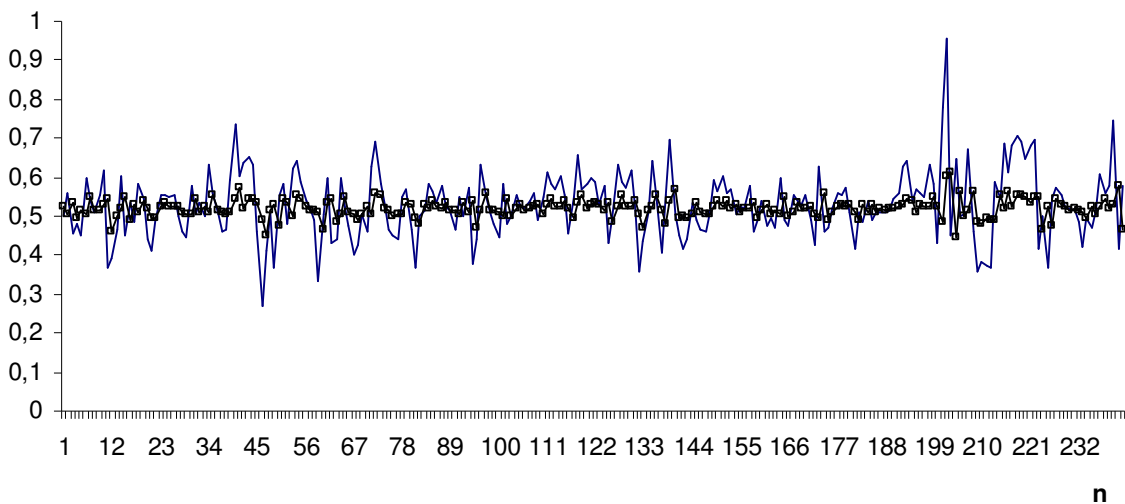


Figura 6.238 Previsão obtida pelo modelo ARIMA-GARCH para NDLCESP

Na Figura 6.239a é mostrada a previsão da volatilidade e na Figura 6.239b a série original e o intervalo de previsão para a série NDLCESP.

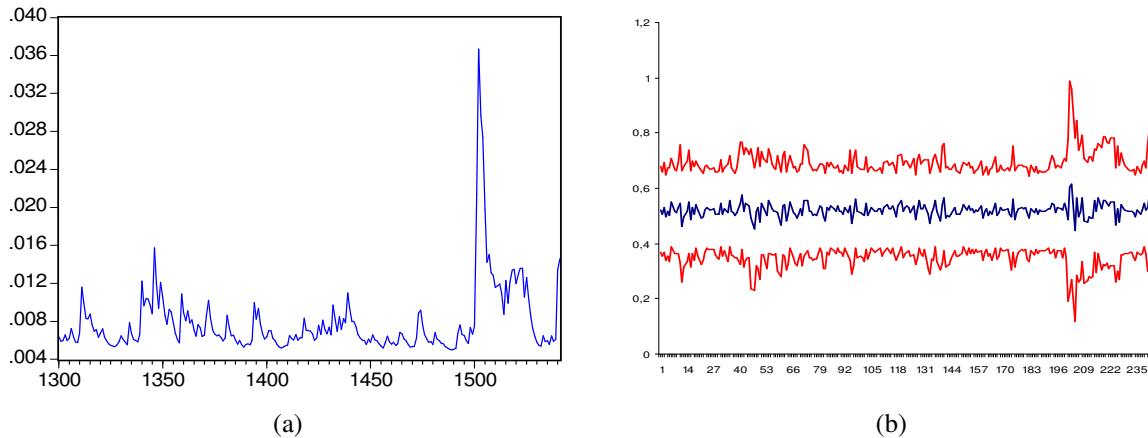


Figura 6.239 (a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLCESP

6.5.5.2 Análise da série da Cesp – Método RNA *feedforward*

Para a série NDLCESP os resultados obtidos pelas diferentes topologias de RNA são apresentados na Tabela 6.113.

Topologia	FIC	FIP	TIC	RMSE
RNA 10-1-1	119	1	0,0762292	0,0805912
RNA 10-2-1	134	1	0,0802943	0,0848267
RNA 10-3-1	100	1	0,0765816	0,0811428
RNA 10-4-1	115	0	0,0762528	0,0808497
RNA 10-5-1	102	1	0,0784227	0,0831358
RNA 10-8-1	100	0	0,0812149	0,0860298

Tabela 6.113 Estatísticas para diferentes topologias de RNA para a série

Na Figura 6.240 são apresentadas as previsões obtidas pela RNA 10-1-1 para a série NDLCESP.

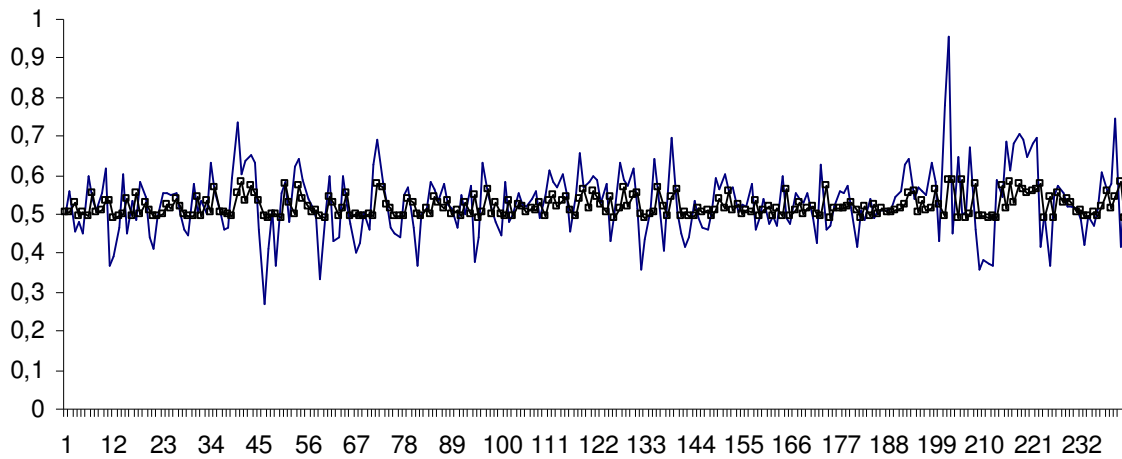
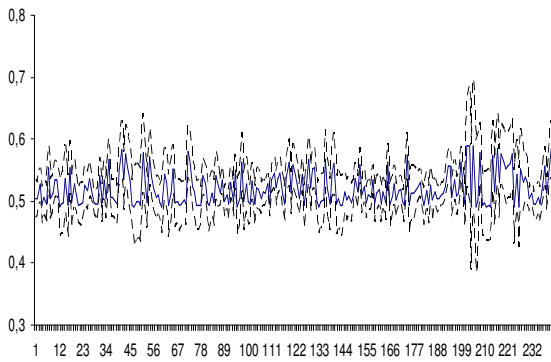
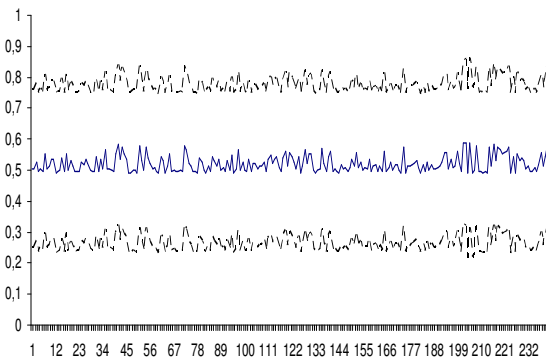


Figura 6.240 Previsão obtida pela RNA *feedforward* para NDLCESP

Na Figura 6.241a é mostrado o intervalo de confiança para as previsões obtidas pela RNA 10-1-1 e na Figura 6.241b o intervalo de predição para a série NDLCESP.



(a)



(b)

Figura 6.241 (a) Intervalo de confiança, (b) Intervalo de predição para NDLCESP

Na Figura 6.242a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA 10-1-1 e na Figura 6.242b o histograma desses resíduos.

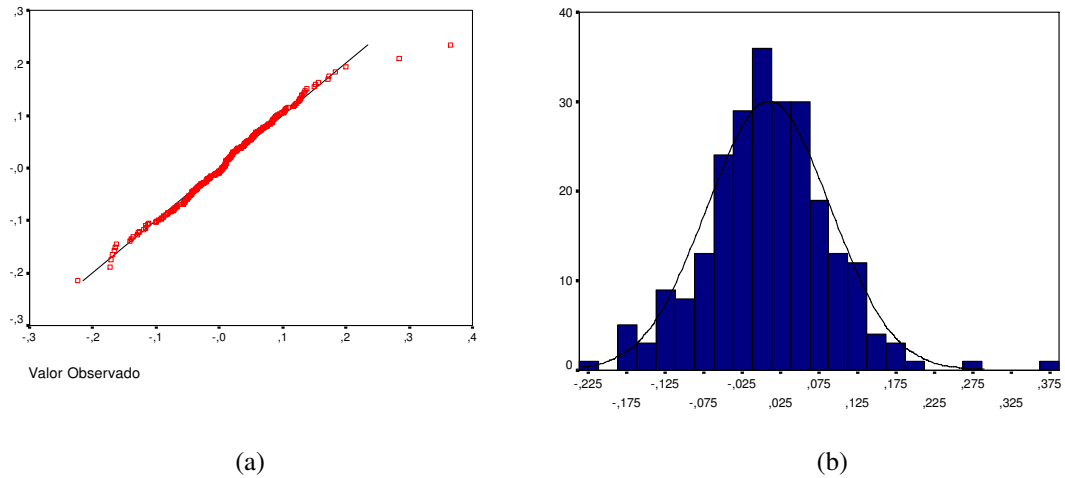


Figura 6.242 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA *feedforward* para NDLCESP

6.5.5.3 Análise da série da Cesp – Método RNA EKF

As previsões, para a série NDLCESP, obtidas pela RNA treinada com o filtro de Kalman estendido (EKF) são mostradas na Figura 6.243.

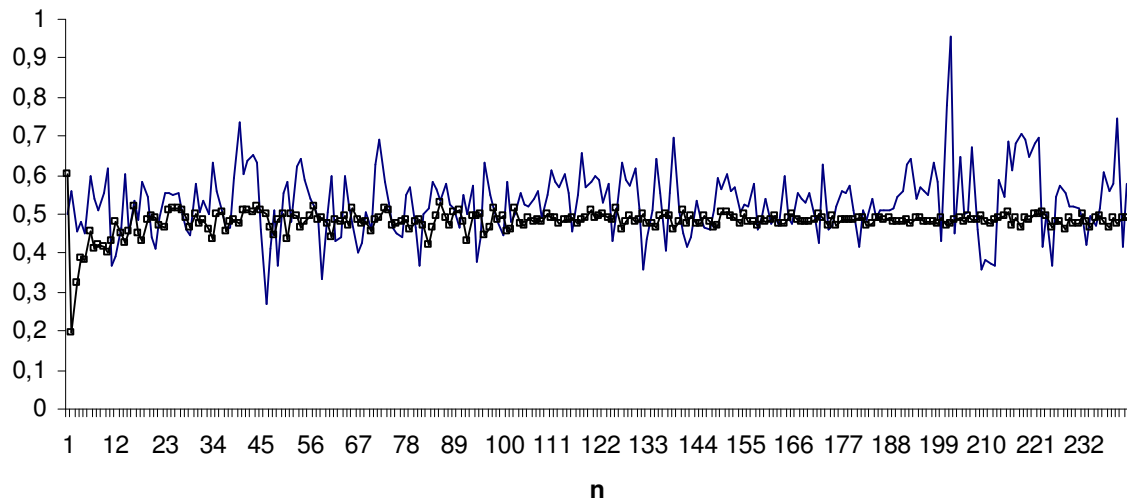


Figura 6.243 Previsão obtida pela RNA EKF para NDLCESP

Na Figura 6.244a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman estendido e na Figura 6.244b o histograma desses resíduos.

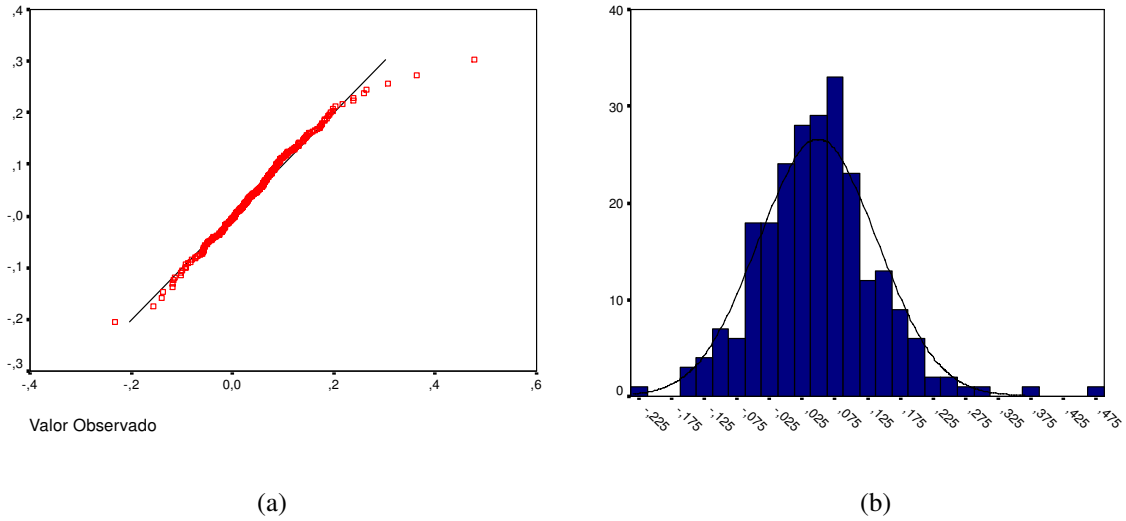


Figura 6.244 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLCESP

6.5.5.4 Análise da série da Cesp – Método RNA UKF

As previsões obtidas pela RNA treinada com o filtro de Kalman unscented são mostradas na Figura 6.245.

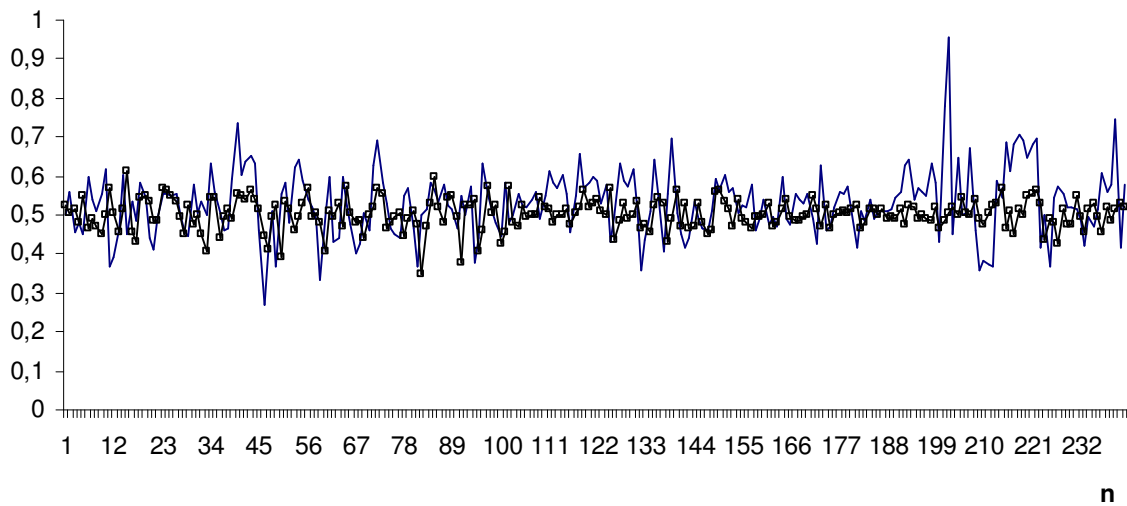


Figura 6.245 Previsão obtida pela RNA UKF para NDLCESP

Na Figura 6.246a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman *unscented* e na Figura 6.246b o histograma desses resíduos.

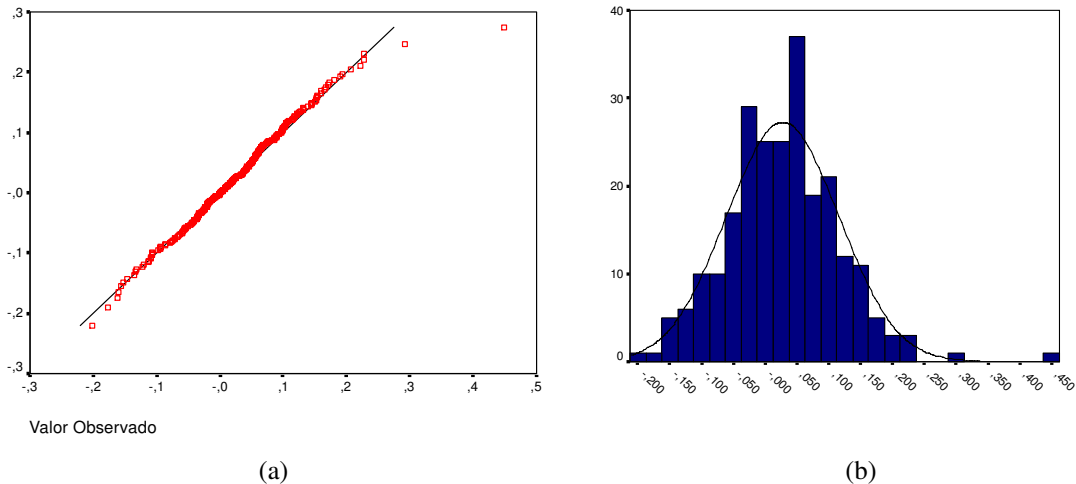


Figura 6.246 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLCESP

As estatísticas de previsão obtidas pelos modelos ARIMA-GARCH, RNA *feedforward*, RNA EKF e RNA UKF são apresentadas na Tabela 6.114 e na Figura 6.247.

Método	TIC	RMSE
AR(2)-GARCH(1,1)	0,076660	0,081037
RNA 10-1-1	0,076229	0,080591
RNA EKF	0,098415	0,096417
RNA UKF	0,055934	0,055821

Tabela 6.114 Estatísticas de previsão para o método de ARIMA-GARCH e de RNA

Pode-se verificar que as melhores previsões foram obtidas pelo método que empregou redes neurais treinadas com o filtro de Kalman *unscented*.

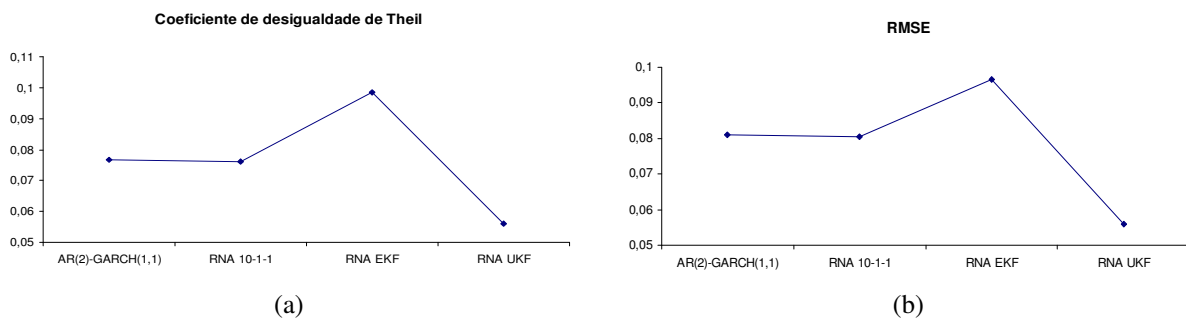


Figura 6.247 Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b)

RMSE

6.5.6 Análise da série da Lojas Americanas

A seguir são apresentados os resultados obtidos para a série da Lojas Americanas.

6.5.6.1 Análise da série da Lojas Americanas– Método ARIMA-GARCH

Na Figura 6.248a é apresentada a série original do preço de fechamento da Lojas Americanas, na Figura 6.248b é apresentada a série normalizada da primeira diferença do logaritmo do preço, NDLLOJA.

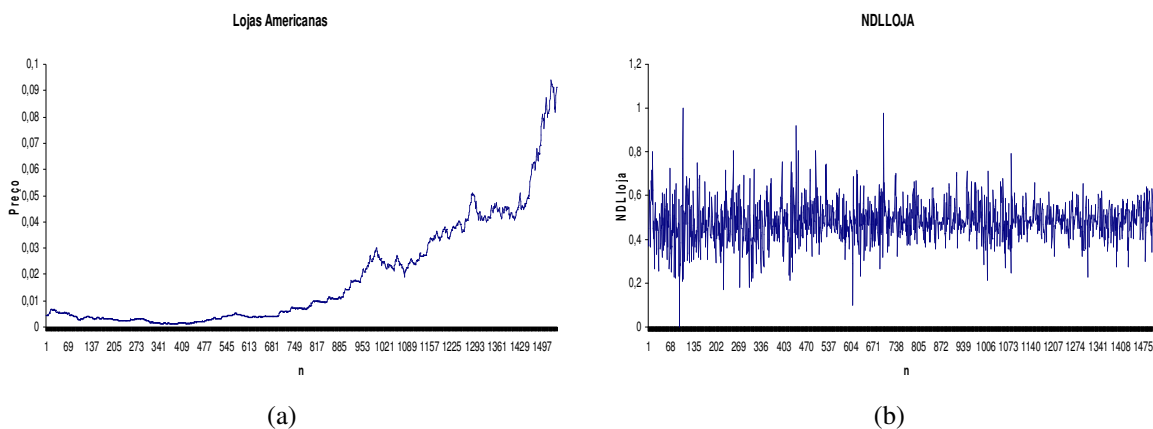


Figura 6.248 (a) Série no nível do preço da Lojas Americanas, (b) Série transformada NDLLOJA

O resultado do teste ADF, para a série original, é apresentado na Tabela 6.115.

Estatística do teste ADF para série original	valores críticos 1%	-3,43439
	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771
3,49532		

Tabela 6.115 Teste ADF para série original da Lojas Americanas

Não é rejeitada a hipótese de raiz unitária para a série no nível pelo teste ADF. O teste de presença de uma raiz unitária para a série transformada apresentou os seguintes resultados:

Estatística do teste ADF para série transformada	valores críticos 1%	-3,43439
	valores críticos 5%	-2,86321
	valores críticos 10%	-2,56771
-34,3720		

Tabela 6.116 Teste ADF para série NDLLOJA

Na Figura 6.249 é apresentada a estatística de Jarque-Bera, que rejeita fortemente a hipótese de distribuição normal.

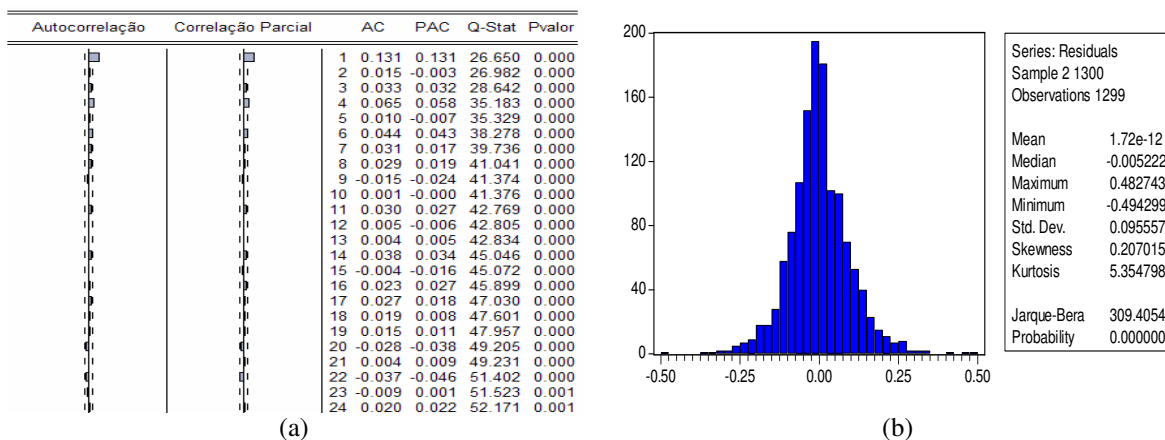


Figura 6.249 (a) Correlograma da série NDLLLOJA, (b) Estatísticas dos resíduos

Os resultados para o teste BDS são apresentados na Tabela 6.117.

BDS				
Dimensão	Estatística	Erro-padrão	Estatística-z	p-valor
2	0,018311	0,002144	8,541553	0,0000
3	0,034349	0,003396	10,11477	0,0000
4	0,046301	0,004031	11,48712	0,0000
5	0,050719	0,004187	12,11207	0,0000
6	0,052324	0,004025	12,99924	0,0000

Tabela 6.117 Resultados obtidos pelo teste BDS nas respectivas dimensões.

O melhor modelo ajustado para a série é um AR(1)-GARCH(1,1). Os parâmetros estimados do modelo para a série são mostrados na Tabela 6.118.

Parâmetros	Coefficiente	Erro-padrão	Estatística-z	p-valor
ϕ_0	0,483798	0,002890	167,3901	0,0000
ϕ_1	0,143090	0,029431	4,861816	0,0000
α_0	0,000542	0,000114	4,765216	0,0000
α_1	0,153613	0,018005	8,531701	0,0000
β_1	0,793986	0,023577	33,67619	0,0000

Tabela 6.118 Parâmetros estimados para a série NDLLLOJA

O resultado do teste Ljung-Box para a série NDLLLOJA é apresentado na Figura 6.250.

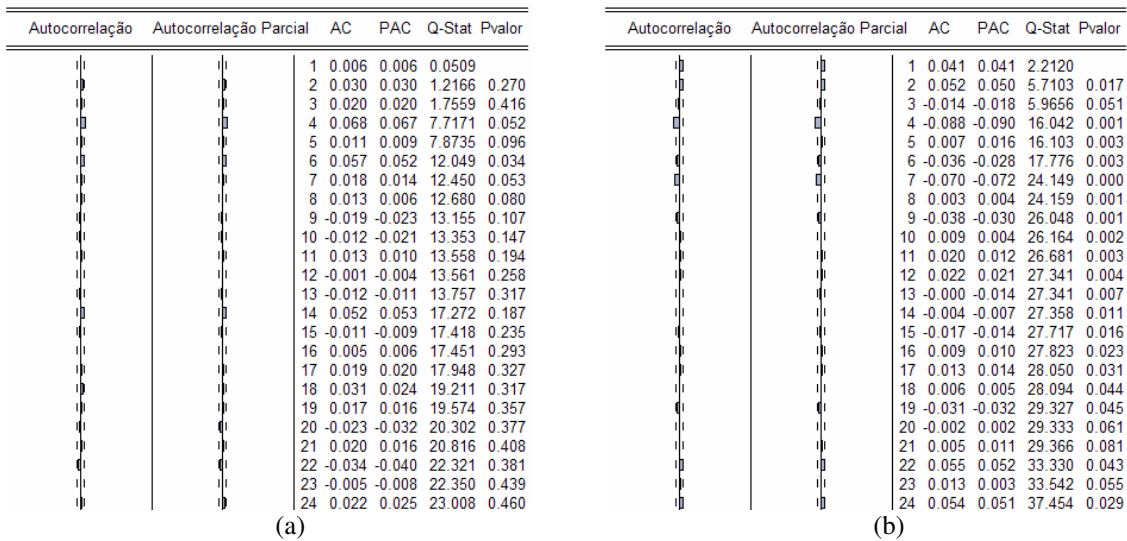


Figura 6.250 (a) Correlograma dos resíduos, (b) Correlograma dos resíduos ao quadrado

As previsões realizadas utilizando o modelo ARIMA-GARCH são mostradas na Figura 6.251. A linha contínua indica o valor da série NDLLOJA e a linha central com marcador quadrado indica as previsões.

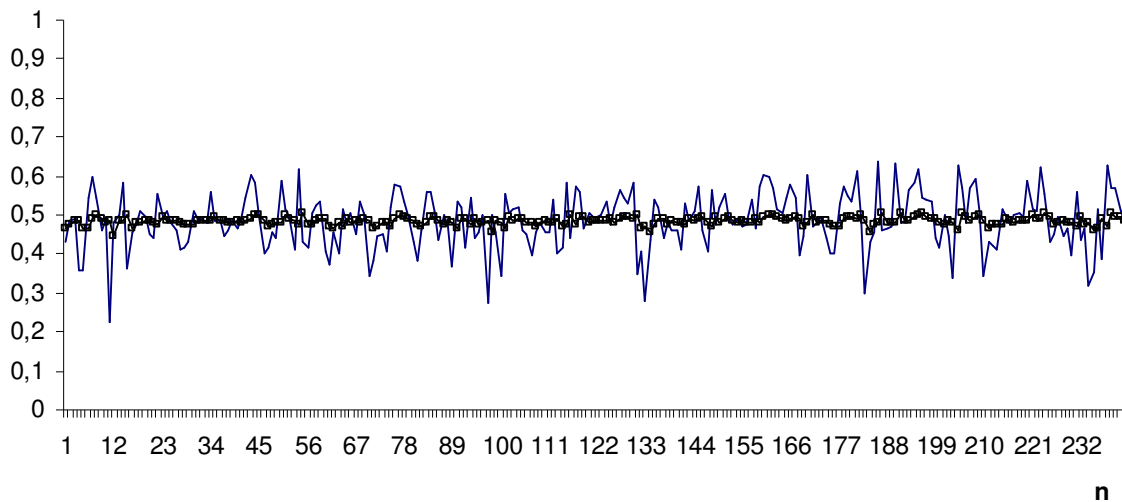


Figura 6.251 Previsão obtida pelo modelo ARIMA-GARCH para NDLLOJA

Na Figura 6.252a é mostrada a previsão da volatilidade e na Figura 6.252b a série original e o intervalo de previsão para a série NDLLOJA.

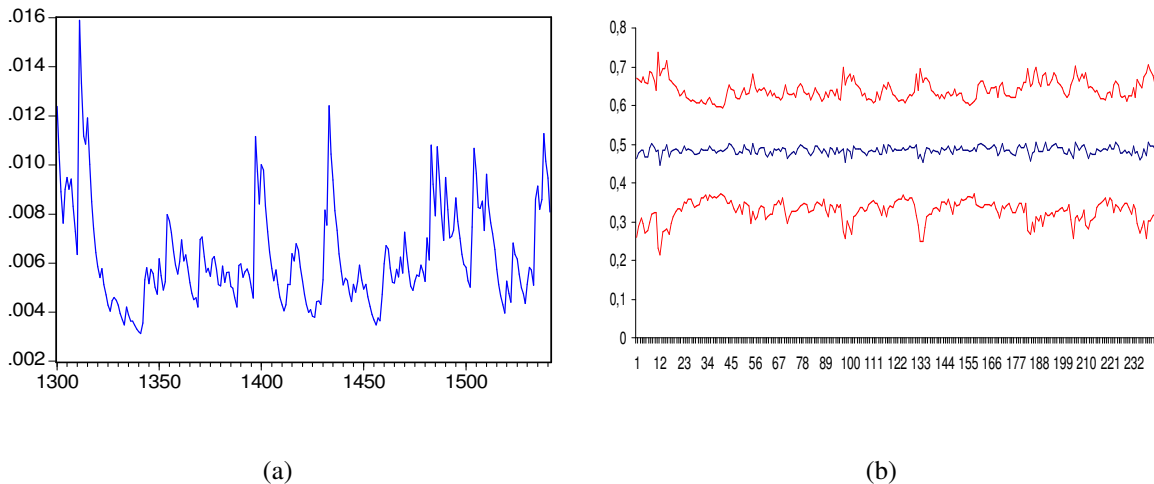


Figura 6.252 (a) Previsão da Volatilidade, (b) Intervalo de predição obtida pelo modelo ARIMA-GARCH para NDLLOJA

6.5.6.2 Análise da série da Lojas Americanas – Método RNA *feedforward*

Para a série NDLLOJA os resultados obtidos pelas diferentes topologias de RNA são apresentados na Tabela 6.119.

Topologia	FIC	FIP	TIC	RMSE
RNA 10-1-1	132	1	0,0727541	0,0706928
RNA 10-2-1	132	1	0,0727211	0,0703685
RNA 10-3-1	128	0	0,0729165	0,0706693
RNA 10-4-1	121	0	0,0731999	0,0712223
RNA 10-5-1	121	0	0,0764064	0,0743788
RNA 10-8-1	126	0	0,076748	0,0746166

Tabela 6.119 Estatísticas para diferentes topologias de RNA para a série

Na Figura 6.253 são apresentadas as previsões obtidas pela RNA 10-2-1 para a série NDLLOJA.

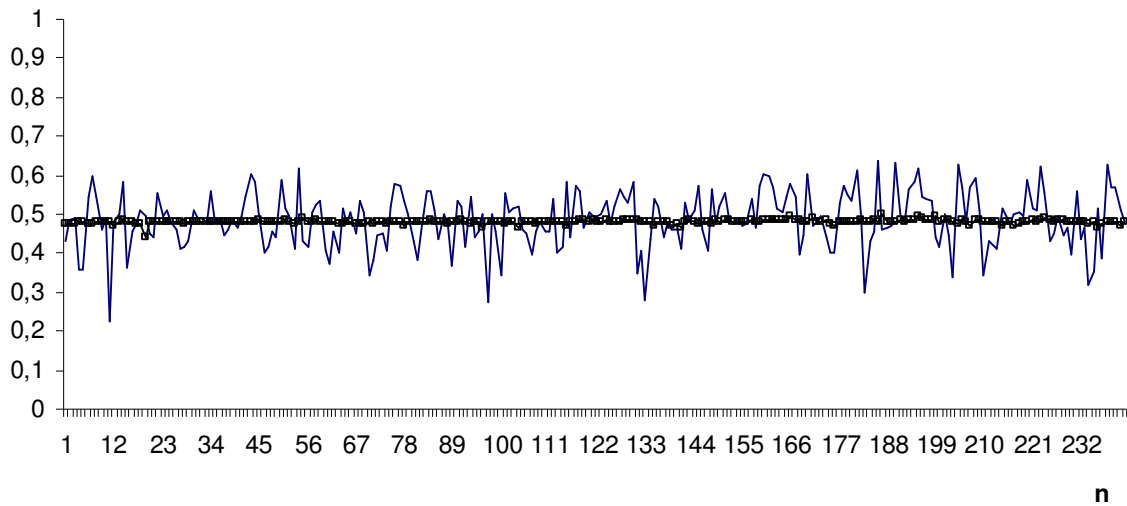


Figura 6.253 Previsão obtida pela RNA *feedforward* para NDLLOJA

Na Figura 6.254a é mostrado o intervalo de confiança para as previsões obtidas pela RNA 10-2-1 e na Figura 6.254b o intervalo de predição para a série NDLLOJA.

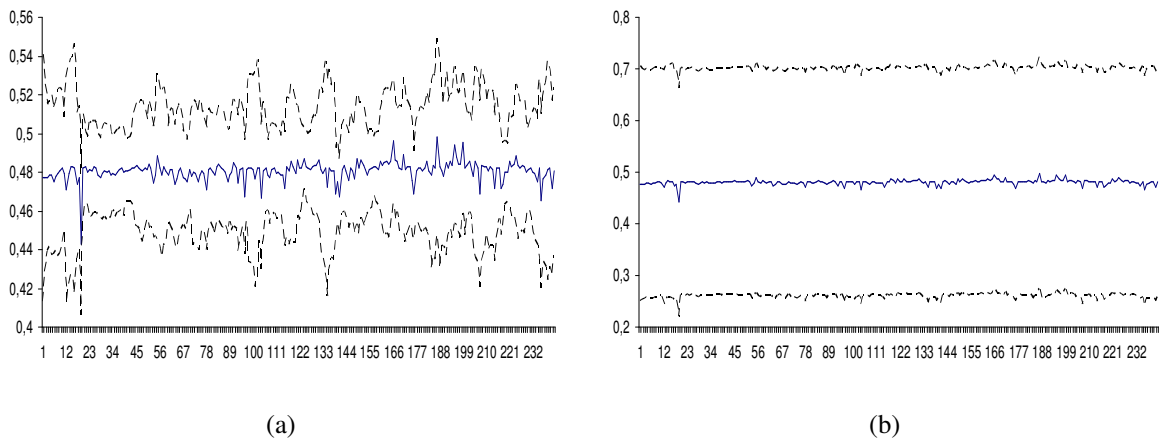


Figura 6.254 (a) Intervalo de confiança, (b) Intervalo de predição para NDLLOJA

Na Figura 6.255a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA 10-2-1 e na Figura 6.255b o histograma desses resíduos.

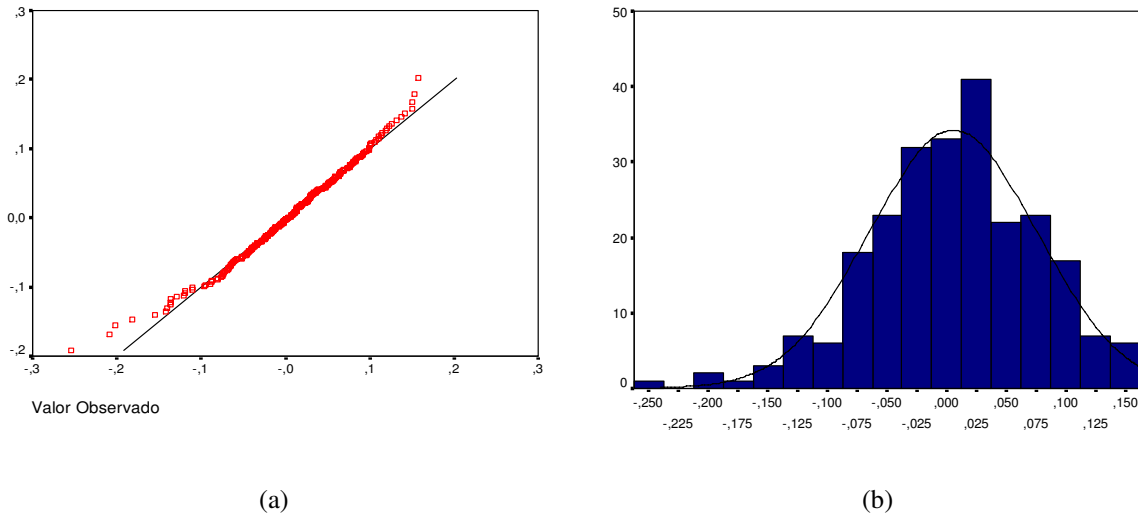


Figura 6.255 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA *feedforward* para NDLLOJA

6.5.6.3 Análise da série da Lojas Americanas – Método RNA EKF

As previsões, para a série NDLLOJA, obtidas pela RNA treinada com o filtro de Kalman estendido (EKF) são mostradas na Figura 6.256.

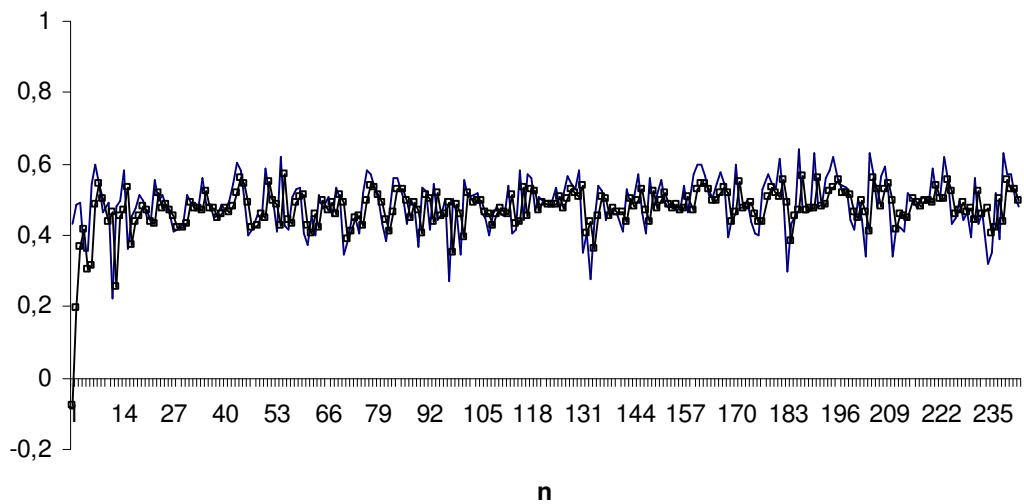


Figura 6.256 Previsão obtida pela RNA EKF para NDLLOJA

Na Figura 6.257a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman estendido e na Figura 6.257b o histograma desses resíduos.

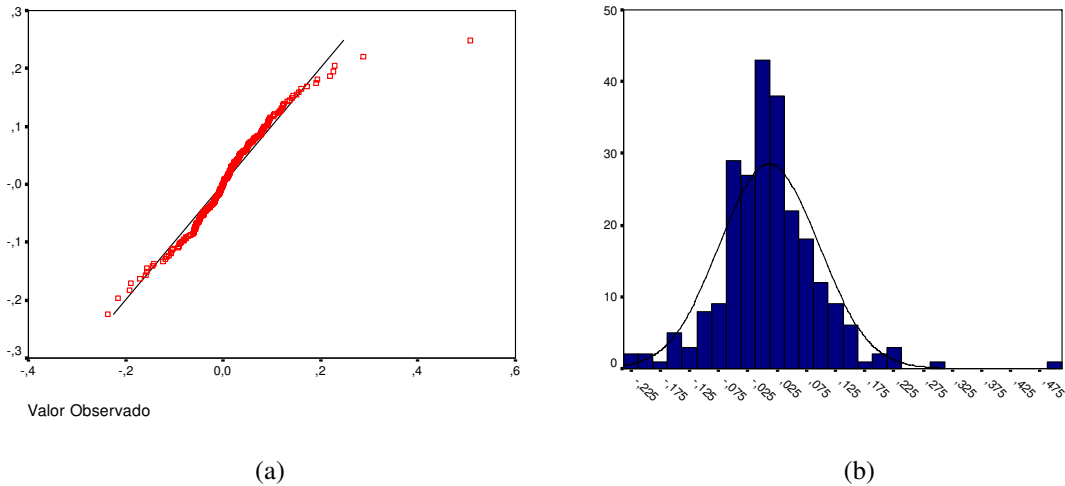


Figura 6.257 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA EKF para NDLLLOJA

6.5.6.4 Análise da série da Lojas Americanas – Método RNA UKF

As previsões obtidas pela RNA treinada com o filtro de Kalman *unscented* são mostradas na Figura 6.258.

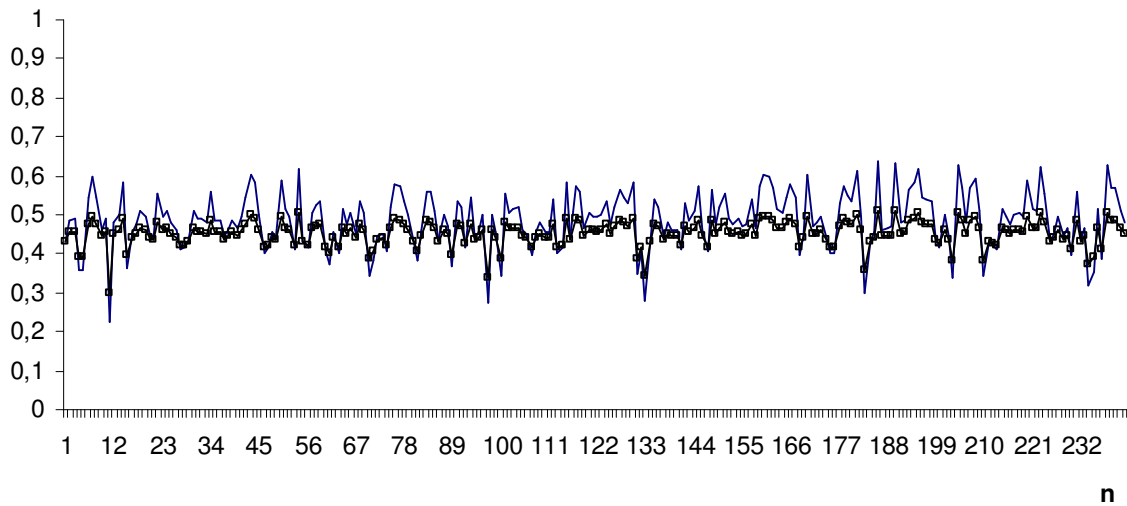


Figura 6.258 Previsão obtida pela RNA UKF para NDLLLOJA

Na Figura 6.259a são apresentados o Q-Q plot dos resíduos da previsão obtidos pela RNA treinada com o filtro de Kalman *unscented* e na Figura 6.259b o histograma desses resíduos.

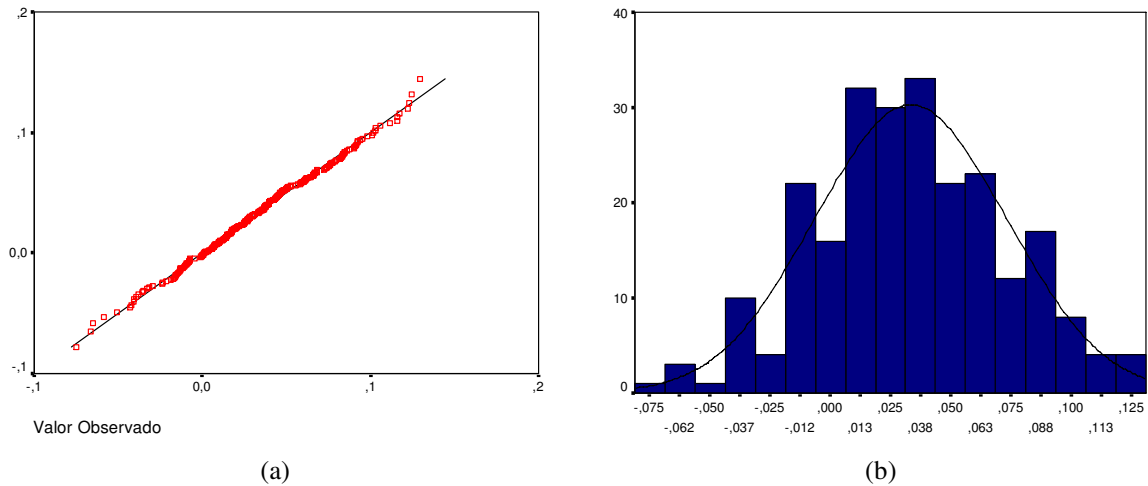


Figura 6.259 (a) Normal Q-Q Plot, (b) Histograma dos resíduos da previsão da RNA UKF para NDLOJA

As estatísticas de previsão obtidas pelos modelos ARIMA-GARCH, RNA *feedforward*, RNA EKF e RNA UKF são apresentadas na Tabela 6.120 e na Figura 6.260.

Método	TIC	RMSE
AR(1)-GARCH(1,1)	0,070846	0,069101
RNA 10-2-1	0,072721	0,070369
RNA EKF	0,087742	0,085072
RNA UKF	0,054908	0,051889

Tabela 6.120 Estatísticas de previsão para o método de ARIMA-GARCH e de RNA

Pode-se verificar que as melhores previsões foram obtidas pelo método que empregou redes neurais treinadas com o filtro de Kalman *unscented*.

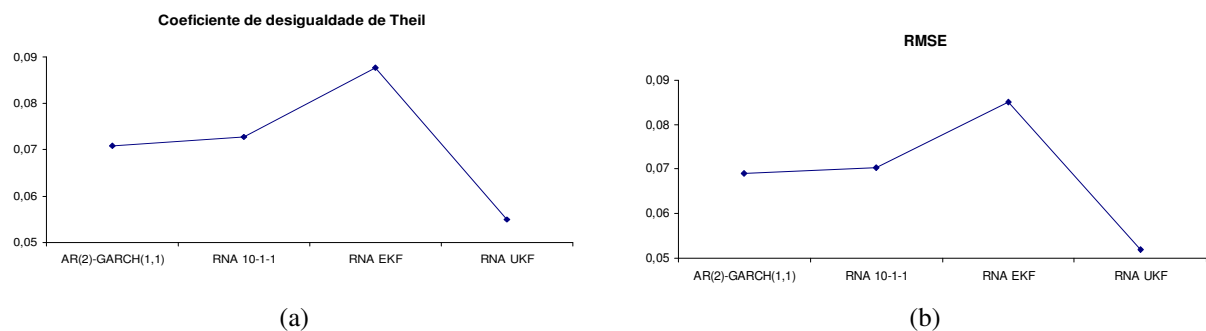


Figura 6.260 Estatísticas de previsão obtidas pelo método ARIMA-GARCH e por RNA: (a) TIC e (b)

RMSE

6.6 Síntese dos Resultados Obtidos

Os resultados obtidos dos testes ADF, para todas as séries, indicaram a rejeição da hipótese de presença de raiz unitária para as séries transformadas. Depois da estimação dos modelos ARIMA verificou-se que os resíduos resultantes destes ajustes apresentaram distribuições leptocúrticas e a estatística de Jarque-Bera rejeitou a hipótese de distribuição normal para esses resíduos. A seguir, a realização do teste BDS, para verificar a presença de dependência não linear, mostrou para todas as séries que pelo menos um dos cinco níveis descritivos do teste (p-valor) foi menor do que o nível de significância de 5%, o que justifica o ajuste de um modelo GARCH. O teste ARCH-LM para os resíduos dos modelos ajustados para as séries NDL mostraram que há efeito ARCH nos resíduos. Após o ajuste dos modelos ARIMA-GARCH foi realizado o teste de Ljung-Box, para todas as séries dos quatro setores, verificando-se que os resíduos obtidos são RB. Finalmente, foram realizadas as previsões utilizando os modelos ARIMA-GARCH e obtidas as estatísticas de ajuste TIC (coeficiente de desigualdade de Theil) e RMSE (raiz quadrada do erro quadrado médio). A previsão da volatilidade e os intervalos de previsão obtidos pelos modelos ARIMA-GARCH para as séries foram apresentados no final de cada análise.

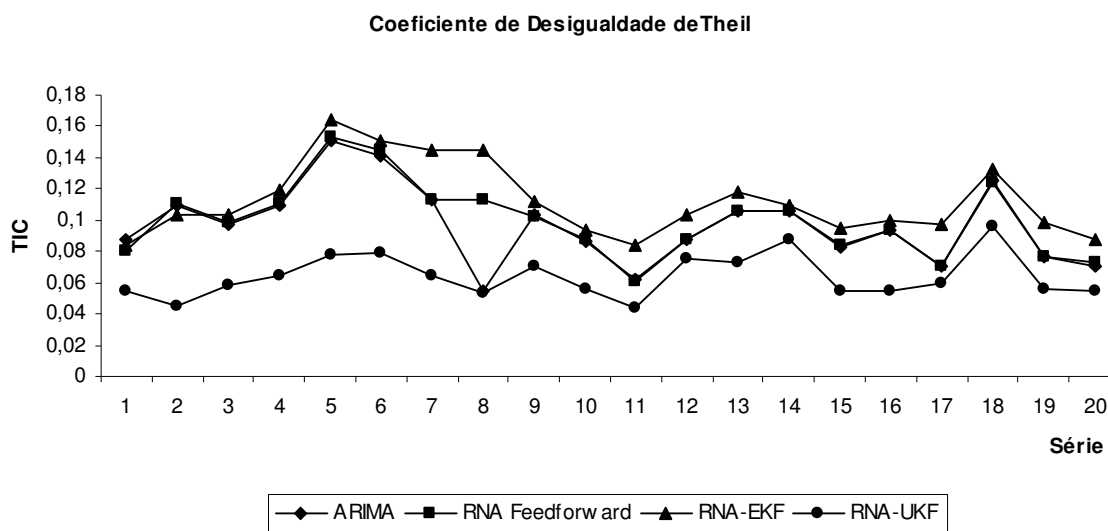


Figura 6.261 Resultados das estatísticas TIC para cada uma das vinte séries analisadas

No caso das redes neurais alimentadas adiante, para se escolher as melhores topologias, foram utilizados os critérios TIC, RMSE, FIC e o FIP, sendo que os dois últimos foram

utilizados como critérios de desempate. Depois de obtidas as previsões das RNA *feedforward*, foram construídos os intervalos de confiança e de predição.

As previsões das RNA-EKF apresentaram os maiores valores de TIC e de RMSE quando comparados com aqueles das outras técnicas. Enquanto que as previsões das RNA-UKF apresentaram os melhores desempenhos.

Nas Figuras-6.261 e 6.262 podem ser vistos os resultados das estatísticas TIC e RMSE para as vinte séries. Nota-se que as RNA-UKF obtiveram os menores valores dessas estatísticas, o que caracteriza melhores resultados de previsão.

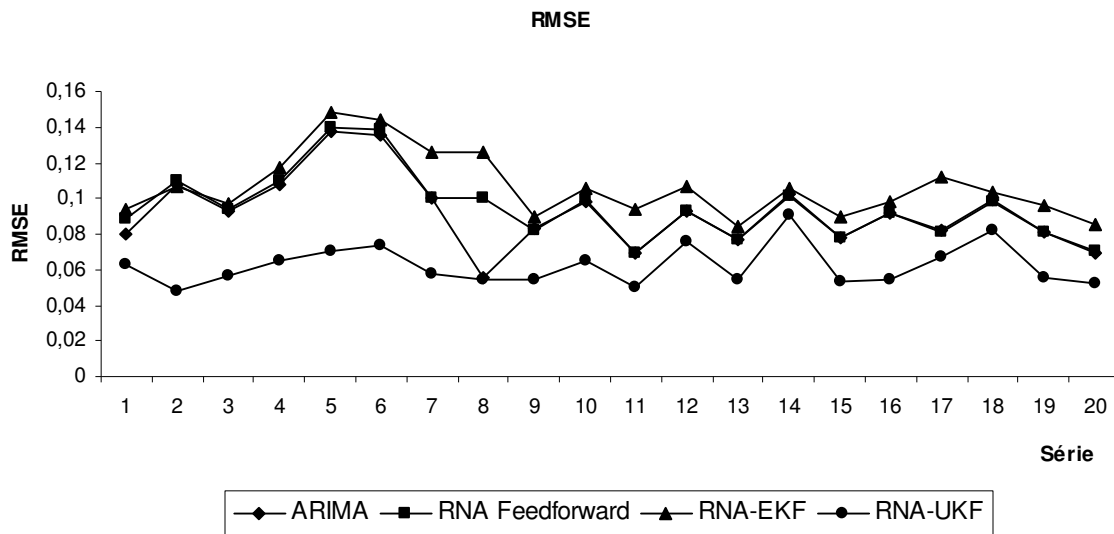


Figura 6.262 Resultados das estatísticas RMSE para cada uma das vinte séries analisadas

Capítulo 7

Conclusões

As redes neurais artificiais têm sido utilizadas em finanças e econometria para auxiliar especialistas em processos de tomada de decisão. Tanto os modelos tradicionais construídos a partir de técnicas estatísticas quanto a tecnologia de redes neurais apresentam limitações quando são usados para modelar o mercado financeiro e eventos econômicos, devido a volatilidade e imprevisibilidade de determinados eventos nesses setores. Porém, desde o início do século passado pesquisadores têm buscado compreender e modelar os mercados, muitas vezes com teorias incapazes de lidar com eventos radicais e imprevisibilidades, mas que servem para ampliar a compreensão do mundo real. Neste contexto, surgem as redes neurais artificiais como uma tecnologia para modelar os mercados e trazer novas alternativas para aplicações reais.

Neste trabalho foi apresentada a metodologia de construção de intervalos de previsão e de confiança para RNA *feedforward*. A construção de intervalos de predição e de confiança para RNA foi realizada através da técnica de regressão multivariada. O intervalo de previsão no método ARIMA-GARCH mostra que existe uma maior influência da volatilidade nos limites superior e inferior do intervalo quando comparado com a RNA *feedforward*. Os intervalos gerados pela RNA *feedforward* gerados para todas as séries analisadas mostraram-se mais conservadores (maior amplitude) do que aqueles fornecidos pelo método ARIMA-GARCH.

Com os resultados obtidos, pode-se concluir que a determinação de intervalos de predição e de confiança é uma ferramenta auxiliar para determinar a topologia da RNA utilizada na predição da série. O método apresentado pode ser estendido para aplicações de RNA com mais de uma camada intermediária, mais de uma saída e diferentes formas de entrada.

Os parâmetros de séries temporais que apresentam heterocedasticidade condicional autorregressiva generalizada influenciam no desempenho da predição de redes neurais *feedforward*. Os principais resultados obtidos das simulações foram:

- i) A saída da função de ativação do tipo sigmóide gera maior concentração de valores próximos de zero e um a medida que cada um dos parâmetros do modelo AR-GARCH aumenta e os outros dois permanecem menores que 0,2.
- ii) Ocorre uma maior concentração de resultados próximos do valor máximo da derivada da função de ativação para modelos AR-GARCH. Mantendo dois dos parâmetros fixos e aumentando o outro ocorre uma migração de valores para próximo de zero, sendo que essa dinâmica é mais intensa em séries em que o parâmetro da volatilidade aumenta.
- iii) A medida que o parâmetro da estacionariedade aumenta, mantendo os outros dois parâmetros fixos, as previsões da RNA melhoram, independente da topologia. No caso da RNA com maior número de neurônios na camada intermediária 10-9-1 verificou-se uma leve melhora de previsão quando o parâmetro relacionando a estacionariedade varia entre 0,1 e 0,8 mantendo-se os demais parâmetros inferiores a 0,2.
- iv) Todas as topologias apresentaram melhores previsões para séries em que o parâmetro ligado a estacionariedade estava dentro do maior intervalo de simulação [0,7;0,8].

O estudo concentrou-se no uso de RNA do tipo *feedforward* de três camadas treinada com algoritmo de Levenberg-Marquardt. Para essa configuração específica pode-se concluir que é importante realizar a estimação do modelo ARIMA-GARCH antes da seleção da topologia da rede neural, pois os valores dos parâmetros influenciam diretamente nos resultados das previsões.

Foram utilizados quatro métodos para realizar previsões de um passo a frente para séries temporais: ARIMA-GARCH, RNA *feedforward*, RNA treinada com o filtro de Kalman estendido e o *unscented*. O filtro de Kalman é um estimador ótimo, permite a inferência de parâmetros de interesse de observações que muitas vezes são indiretas, imprecisas e incertas. O algoritmo para implementação tanto do EKF quanto do UKF é recursivo, de tal forma que novas medidas podem ser processadas a medida que são inseridas no modelo.

Dos resultados obtidos verifica-se que a rede neural treinada com o filtro de Kalman *unscented* apresentou melhores resultados de previsão que a RN treinada com o EKF, RNA *feedforward* e o modelo ARIMA-GARCH. Embora o UKF não necessite do cálculo da matriz Jacobiana, uma limitação da implementação do UKF é a necessidade de escolher os três parâmetros da transformação *unscented* (α , β e κ). A seleção ótima depende do problema e ainda não tem uma metodologia para escolha de tais parâmetros. O uso do UKF

em redes neurais é uma evolução para estimação de parâmetros quando comparado ao EKF, a vantagem do UKF não é óbvia. O UKF fornece uma estimativa mais precisa através de uma aproximação direta da esperança da Hessiana e a taxa de convergência pode ser diferente do EKF. No caso das séries analisadas neste trabalho, os critérios RMSE e TIC foram menores em todos os resultados obtidos pela RNA-UKF.

Algumas sugestões para trabalhos futuros podem ser geradas a partir dos resultados obtidos nesta pesquisa. A primeira delas seria a construção de intervalos de previsão para redes neurais que utilizam o filtro de Kalman. Os intervalos de confiança e previsão para as RNA *feedforward* não apresentam bons resultados para a volatilidade, essa limitação deveria ser melhorada para que estes intervalos possam ser comparados ao intervalos gerados pelos modelos GARCH . Este trabalho considerou a análise de séries univariadas, uma extensão desta pesquisa seria aplicar RNA-UKF em modelos multivariados. Como os parâmetros da transformação *unscented* dependem do problema que se tem em mãos, simulações de Monte Carlo poderiam ser realizadas para mostrar a influência destes parâmetros nas previsões e verificar se diferentes topologias de RNA teriam resultados diferentes.

8 Referências Bibliográficas

ABU-MOSTAFA, Y. S., ATIYA, A. F., MAGDON-ISMAIL, M, e WHITE, H. Introduction to the special issue on neural networks in financial engineering. *IEEE Transactions on Neural Networks* 12: 653–655, 2001.

ALMEIDA, R. O. M. Três ensaios com aplicações de redes neurais em séries financeiras, FEA - Fac Economia, Administração e Contabilidade, TESE Dat.Def. 04.06., 2004

ANDERS, W. Garch forecasting performance under different distribution assumptions, **Journal of Forecasting**, 2006.

ANDERSEN, T. J. e WILAMOWSKI, B.M. “A. Modified Regression Algorithm for Fast One Layer Neural Network Training”, *World Congress of Neural Networks*, vol. 1, p. 687-690, 1995.

AZOFF, E. M. **Neural network time series forecasting of financial markets**, John Wiley & Sons Ltd., 1994.

AZZOF, E. M. Reducing Error in Neural Network Time Series Forecasting. **Neural Computing and Applications**, vol. 1, pp. 240-247, 1993.

BALKIN, S. D. e ORD, J. K. Automatic neural network modeling for univariate time series. **International Journal of Forecasting** 16: 509–515, 2000.

BARUCCI, E. e RENO, R. On measuring volatility and the GARCH forecasting performance, Vol.12, **Journal of International Financial Markets**, Institutions and Money, 2002.

BATTITI, R., “First- and second-order methods for learning: between steepest descent and Newton’s method”, **Neural Computation**, vol. 4, no. 2, p. 141-166, 1992.

BERA, A. K. e HIGGINS, M. L. ARCH Models: Properties, Estimation and Testing, **Journal of Economic Surveys**, Vol. 7 No. 4, p.305-362, 1993.

BISHOP, C. M. **Neural Networks for Pattern Recognition**, Oxford University Press, 1995.

BOLLERSLEV, T., ENGLE, R. F., E NELSON, D. B. ARCH models, **Handbook of Econometrics**, 63, 37-60, 1994.

BOLLERSLEV, T., Generalized Autoregressive Conditional Heteroskedasticity, **Journal of Econometrics**, 31, pp. 307-327, 1986.

BOX, G. e JENKINS, G. **Time series analysis: Forecasting and control**, San Francisco: Holden-Day, 1970.

BOX, G. e PIERCE, D.A. Distribution of residual autocorrelations in autoregressive integrated moving average time series models. **Journal of the American Statistical Association** 65, p. 1509-1526, 1970.

BOX, G. E. P. E JENKINS, G. M. **Time series analysis: forecasting and control**, San Francisco: Holden Day, 1976.

BOX, G.E.P., JENKINS, G.M., e REINSEL, G.C., **Time Series Analysis: Forecasting and Control**, Terceira Edição, Prentice Hall, 1994.

BROCK, W. A., DECHERT, W. D. e SCHEINKMAN, J. A. A Test for Independence Based on the Correlation Dimension, Department of Economics, University of Wisconsin at Madison, University of Houston, and University of Chicago, 1987.

BROCK, W., DECHERT, W., SCHEINKMAN, J., e LEBARON, B. A Test for Independence Based on the Correlation Dimension, **Econometric Reviews**, 15, p.197-235, 1996.

CAMPBELL, J. Y., LO, A. W. e MACKINLAY, A. C., **The Econometrics of Financial Markets**, 1997.

CASDAGLI, M. e EUBANK, S. **Nonlinear Modeling and Forecasting**, Santa Fe Institute Proceedings vol. XII, Addison-Wesley, 1992.

CASDAGLI, M. Nonlinear Prediction of chaotic time series. *Physica D*, vol. 35, pp. 335-356, 1989.

CHAKRABORTY, K. Forecasting the behaviour of Multivariate Time Series Using Neural Networks. **Neural Networks**, vol. 5, pp. 961-970, 1992.

CHARALAMBOUS, C., “Conjugate gradient algorithm for efficient training of artificial neural networks”, *IEEE Proceedings*, vol. 139, no. 3, pp. 301-310, 1992.

CHEN, X., RACINE, J. e SWANSON, N. R. Semiparametric ARX neural-network models with an application to forecasting inflation, *IEEE Transactions on Neural Networks* 12, 2001.

CHENG, B. E TITTERINGTON, D. M., **Neural Networks: A Preview from Statistical Perspective**. Statistical Science, 1994.

CLEMENTS, M. P. e HENDRY, D. F. **Forecasting Non-stationary Economic Time Series**. MIT Press: Cambridge, MA, 1999.

DE VAUX, R. D., SCHUMI, J., SCHWEINSBERG, J. e UNGAR, L. H. Prediction Intervals for Neural Networks via Nonlinear Regression, **Technometrics**, Vol. 40, No. 4, Novembro, 1998.

DELGADO, M., PEGALAJAR, M. C., CUÉLLAR, M. P. Memetic evolutionary training for recurrent neural networks: an application to time-series prediction, *Expert Systems*. Oxford, Vol.23, Num. 2, 2006.

DIEBOLD, F. X. E MARIANO, R. S. Comparing predictive accuracy, **Journal of Business and Economic Statistics**, 13, 253-263, 1995.

DREYFUS, G. **Neural Networks**, Springer, 2005.

DUCH, W. e JANKOWSKI, N. New neural transfer functions. *Journal of Applied Mathematics and Computer Science*, 7(3):639–658, 1997.

DUCH, W. e JANKOWSKI, N. Survey of neural transfer functions. **Neural Computing Surveys**, 2:163–212, 1999.

DUCH, W. e JANKOWSKI, N. Taxonomy of neural transfer functions. In Shun-Ichi Amari, C. Lee Giles, Marco Gori, and Vincenzo Piuri, editors, *International Joint Conference on Neural Networks*, vol. III, pages 477–484, Computer Society and IEEE, 2000.

DUDA, R. e HART, P. **Pattern Classification and Scene Analysis**. Segunda edição, John Wiley, New York, 1993.

DUFFIE, D., **Dynamic Asset Pricing Theory**, Princeton, 2001.

ELMAN, J. L. Finding structure in time (CRL Technical Report 8801). La Jolla: University of California, San Diego, Center for Research in Language, 1988.

ELMAN, J. L. Finding structure in time, **Cognitive Science**, vol. 14, p. 179-211, 1990.

ENDERS, W. **Applied Econometric Time Series**, Wiley, 2004.

ENGLE, R. F. Autoregressive conditional heteroskedasticity with estimates of the variance of U. K. inflation, *Econometrica* 50, p. 987-1008, 1982.

ERSOY, K. E HONG, D. Parallel, self-organizing, hierarchical neural networks, p. 167-178, 1990.

FELDKAMP, L. A., PROKHOROV, D. V., e FELDKAMP, T. M. Conditioned adaptive behavior from a fixed neural network. *Proceedings of the 11th Yale Workshop on Adaptive and Learning Systems* p. 78-83, 2001.

FERNANDES, M., PREUMONT, P.Y. The size of the BDS test on GARCH standardized residuals. Rio de Janeiro: Fundação Getulio Vargas, 2002.

FERRARI, P. e GALVES, A. Acoplamentos e processos estocásticos. IMPA, Rio de Janeiro, Brasil. <http://www.ime.usp.br/~Pablo>, 1997.

FINE, T. L. **Feedforward Neural Network Methodology**. Springer: New York, 1999.

FISHMAN, M. B., BARR, D. S. e LOICK, W. J. Using Neural Nets in Market Analysis. Technical Analysis of Stocks and Commodities, vol. 9, no. 4, 1991.

FORSGREN, A. E KLING, R. An Implementation of Recurrent Neural Networks for Prediction and Control of Nonlinear Dynamic Systems, março, 2003.

FRANSES, P. H. **Time series models for business and economic forecasting**, Cambridge University Press, 1998.

FREITAS, J. F. G. Bayesian Methods for Neural Networks. Trinity College. University of Cambridge. 1999.

GALINDO-FLORES, J. Statistical and Machine Learning Frameworks for Economics: Analysis of Error Curves and Applications to Derivatives Pricing and Credit Risk Assessment, Harvard University, 1998.

GATELY, E. **Neural Networks for Financial Forecasting** , John Wiley & Sons, 1997.

GHIASSI, M., SAIDANE, H., ZIMBRA, D. K. A dynamic artificial neural network model for forecasting time series events, Amsterdam, Vol. 21, Num. 2, 2005.

GOURIÉROUX , C. **Time Series and Dynamic Models**, Cambridge Univ Press, 1997.

HAGAN, M. T. e MENHAJ, M., “Training feedforward networks with the Marquardt algorithm”, IEEE Transactions on Neural Networks, vol. 5, no. 6, pp. 989-993, 1994.

HAMILTON, J. D. **Time series analysis**, Princeton: Princeton University Press, 1994.

HAWLEY, D. D., JOHNSON, J. D. e RAINA, D. Artificial Neural Systems: A New Tool for Financial Decision-Making. Finan Analy J, 1990.

HAYKIN, S. **Adaptive Filter Theory**, Prentice Hall, 1996.

HAYKIN, S. **Redes neurais: princípios e prática**, Bookman, Porto Alegre, 2001.

HERTZ, J. A. **Introduction to the theory of neural computation**, Addison-Wesley, 1991.

HSIEH, D. A. Testing for nonlinear dependence in daily foreign exchange rates, **Journal of Business**, 62(3), p. 339–368, 1989.

HWANG, J.T.G. e DING, A. A. Prediction intervals for artificial neural networks. **Journal of the American Statistical Association** 92: 109–125, 1997.

HWANG, J.T.G. e DING, A. A. Prediction intervals for artificial neural networks. **Journal of the American Statistical Association** 92: p. 109–125, 1997.

IMAN, R. L. E CONOVER, W. J. The use of the rank transform in regression, **Technometrics**, vol. 21, no. 4, novembro 1979.

JASIC,T., WOOD, D. The profitability of daily stock market indices trades based on neural network predictions: Case study for the S&P 500, the DAX, the TOPIX and the FTSE in the period 1965-1999, vol. 14, Num. 4; p. 285, 2004.

JHEE, W. C. E LEE, J. K. Performance of neural networks in managerial forecasting, 1993.

JOHNSON, R. A. e WICHERN, D. W. **Applied Multivariate Statistical Analysis**, Prentice Hall, 2002.

JOHNSTON, J. e J. DINARDO, **Econometric Methods**, Quarta Edição. New York, McGraw-Hill.

JORDAN, M. I. Attractor dynamics and parallelism in a connectionist sequential machine, 1986.

JULIER ,S. e UHLMANN, J. The scaled unscented transformation. Em Proc. of the IEEE American Control Conference, 2002.

JULIER, S. J., UHLMANN, J. K. The Scaled Unscented Transformation, Proceedings of the IEEE American Control Conference, 8-10 May, 2002.

JULIER, S. J., UHLMANN, J. K. A general method for approximating nonlinear transformations of probability distributions, Technical Report, University of Oxford, 1996.

JULIER, S. J., UHLMANN, J. K. A new extension of the Kalman filter to nonlinear systems, Proceedings of AeroSense, 1997.

JULIER, S. J., UHLMANN, J. K. e DURRANT-WHITE, H. A new approach for filtering nonlinear systems, em Proceedings of the American Control Conference, p. 1628-1632, 1995.

KIANI, K. M. Detecting Business Cycle Asymmetries Using Artificial Neural Networks and Time Series Models, Computational Economics. Dordrecht, Vol. 26, Num. 1, 2005.

- KNELLER, G. F. **A Ciência como atividade humana**, Edusp, 1978.
- KNUTH, D. E. Seminumerical Algorithms, vol. 2 of The Art of Computer Programming, second edn, Addison-Wesley, 1981.
- KUAN, C. M., e WHITE, H. Artificial Neural Networks: An Econometric Perspective. **Econometric Reviews** 13, p. 1-91, 1994.
- LANG, K. J. E HINTON, G. E., The development of the time-delay neural network architecture for speech recognition, Technical Report, Carnegie-Mellon University, Pittsburgh, PA, 1988.
- LAPEDES, A. e FARBER, R. Nonlinear Signal Processing Using Neural Networks, Prediction and System Modelling. Los Alamos Report LA-UR-87-2662, Los Alamos National Laboratory, 1987.
- LEVENBERG, K., A method for the solution of certain problems in least squares. **Applied Mathematics** vol. 2, pp. 164-168, 1944.
- LIMA, F. G. Um método de análise e previsão de sucessões cronológicas unidimensionais lineares e não lineares, FEA - Fac Economia, Administração e Contabilidade, 2004.
- LIPPMAN, R. P. An introduction to computing with neural nets, IEEE ASSP Magazine, vol.4, pp. 4-22, 1987.
- MADDALA, G. S., e KIM, I. **Unit roots, cointegration and structural change**, Cambridge, 2003.
- MARCONI, M. A. e LAKATOS, E. M. **Metodologia Científica**, Atlas, 2007.
- MARQUARDT, D., An algorithm for least-squares estimation of nonlinear parameters, **Journal of Applied Mathematics** vol. 11, pp. 431-441, 1963.
- MCLEOD, A. e LI, W. Diagnostic checking of ARMA time series models using squared residuals autocorrelations. **Journal of Time Series Analysis**, 4, p. 269–273, 1983.
- MEDEIROS M., TERÄSVIRTA, T. e RECH, G. Building Neural Network Models for Time Series: A Statistical Approach, **Journal of Forecasting** 25, p. 49–75, 2006.
- MORETTIN, P. A. Séries temporais em finanças, IMCA: PUC, 2002.
- MORETTIN, P. A., TOLOI, C. M. C., **Análise de Séries Temporais**, Edgard Blucher, 2004.
-
- _____. **Análise de Séries Temporais**, São Paulo : Edgard Blucher, 2006.

MOZER, M. C. A focused back-propagation algorithm for temporal pattern recognition, University of Toronto, 1988.

NETER, J., KUTNER, M. H., NACHTSHEIM, C. J. **Applied linear regression models**, Irwin, 1996.

NIEDERREITER, H. Random Number Generation and Quasi-Monte Carlo Methods. CBMS-NSF Regional Conference Series in Applied Mathematics 63. SIAM, 1992.

OLIVEIRA, M. A. e FÁVERO, L. P. L. Uma breve descrição de algumas técnicas para análise de séries temporais: séries de Fourier, wavelets, arima, modelos estruturais para séries de tempo e redes neurais. In: VI Semead - Seminários em Administração FEA-USP, 2003, São Paulo. VI Semead - Seminários em Administração FEA-USP, 2003.

OLIVEIRA, M. A. e SIQUEIRA, J. O. Treinamento de Redes Neurais Utilizando Variantes do Filtro de Kalman (EKF e UKF) para Predição de Séries Heterocedásticas, *Simpósio Brasileiro de Redes Neurais*, SBRN, São Luis-MA, 2004.

OLIVEIRA, M. A. e SIQUEIRA, J. O. Um estudo comparativo quanto à eficiência de previsão de séries temporais utilizando redes neurais artificiais recorrentes (RTRL) e processos ARIMA-GARCH, XXVIII EnANPAD, 2004.

OLIVEIRA, M. A. , BERGMANN, D. R., FÁVERO, L. P. L. Previsão de retornos de ações de empresas dos setores financeiro e de alimentos por meio de redes neurais e modelos ARIMA-GARCH. X SEMEAD Seminários em Administração FEA-USP, 2007a.

OLIVEIRA, M. A., FÁVERO, L. P. L. , FAVERO, P. B. Análise multivariada para a avaliação do comportamento de consumo no Município de São Paulo. X SEMEAD Seminários em Administração FEA-USP, 2007b.

OLIVEIRA, M. A., MONTINI, A. A., BERGMANN, D. R. Previsão de retornos de ações de empresas dos setores financeiro, alimentos, industrial e de serviços por meio de redes neurais e modelos ARIMA-GARCH. EnAnpad, 2007c.

_____. Construção de Intervalos de Predição para Redes Neurais via Regressão Multivariada e sua Aplicação em Séries Heterocedásticas. FACEF Pesquisa, 2007d.

_____. Calculating conditional beta using nonparametric estimation of the utility function. Third Brazilian Conference on Statistical Modelling in Insurance and Finance, Maresias. Proceedings of the Third Brazilian Conference on Statistical Modelling in Insurance and Finance, 2007e.

_____. Um Teste do Modelo CAPM no Mercado de Capitais Brasileiro Via GMM, Congresso UFSC de Controladoria e Finanças, 2007f.

OLIVEIRA, M. A., MONTINI, A. A., FÁVERO, L. P. L. Previsão de retornos de ações de empresas dos setores de serviços e industrial por meio de redes neurais do tipo feedforward e dos modelos ARIMA-GARCH, Colóquio de Séries Temporais. 65 anos do Prof. Pedro Alberto Morettin, 2007g.

OLIVEIRA, M. A., MONTINI, A. A., CASTRO, F. H. F. Cálculo do VaR por meio dos modelos ARIMA-GARCH para os retornos de empresas brasileiras dos setores financeiros, de alimentos, industrial e de serviços, Colóquio de Séries Temporais. 65 anos do Prof. Pedro Alberto Morettin, 2007h.

OLIVEIRA, M. A., MONTINI, A. A. A influência dos parâmetros de modelos ARIMA-GARCH na predição de redes neurais feedforward. 12a Escola de Séries Temporais e Econometria, 2007, Gramado. 12a Escola de Séries Temporais e Econometria, 2007.

PATEL, G. S. Modeling Linear dynamics with extended Kalman filter trained recurrent multilayer perceptrons, McMaster University, 2000.

PATEL, G. S. Modeling Linear dynamics with extended Kalman filter trained recurrent multilayer perceptrons, McMaster University, 2000.

PATEL, G. S. Modeling Linear dynamics with extended Kalman filter trained recurrent multilayer perceptrons, McMaster University, 2000.

PEARLMAN, J. G. An algorithm for the exact likelihood of a high-order autoregressive-moving average process. **Biometrika** 67 , 1980.

PEREIRA, B. B. E RODRIGUES, C. V. S. Redes Neurais em Estatística, 13°. SINAPE, Caxambu, 1998.

PUSKORIOUS, G. V. E FELDKAMP, L. A. Decoupled extended Kalman filter training of feedforward layered networks, Proceeding of the International Joint Conference on Neural Networks, vol.1, pp. 771-777, Seattle, WA 1991.

PUSKORIOUS, G. V. E FELDKAMP, L. A. Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrente networks, IEEE Transactions on Neural Networks, vol.5, p.279-297, 1994.

RECH, G. Forecasting with artificial neural network models. Working Paper Series in Economics and Finance 491, Stockholm School of Economics, 2002.

REFENES, A. N. Currency Exchange Rate Prediction and Neural Network Design Strategies. Neural Computing and Applications, vol.1, pp. 46-58, 1993.

REFENES, A. N., BURGESS, A.N. E BENTZ, Y. Neural networks in financial engineering: a study in methodology, IEEE vol.8, no. 6, Novembro, 1997.

RIPLEY, B. D., Neural Networks and Related Methods of Classification, **Journal of the Royal Statistical Society**, Series B, 56, 3, pp. 409-456, 1994.

RIPLEY, B. D., **Pattern Recognition and Neural Networks**. Cambridge Press, 1996.

RIPLEY, B. D., **Statistical Aspects of Neural Networks**. Chapman e Hall, 1993.

RIVALS, I. E PERSONNAZ, L., Construction of confidence intervals for neural networks based on least square estimation, *Neural Networks*, vol. 13, p. 463-484, 2000.

ROJAS , R. **Neural Networks: A Systematic Introduction**, Springer, 1996.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain, **Psychological Review**, vol. 65, pp. 386-408, 1958.

ROSS, S. M. **Simulation**, Academic Press, 2006.

RUMELHART, D. E. e McCLELLAND, J. L. **Parallel Distributed Processing: Explorations in the Microstructure of Cognition**, MIT Press, 1986.

RUMELHART, D. E., HINTON, G. E. e WILLIAMS, R. J, “Learning internal representations by error propagation”, In *Parallel Distributed Processing*, vol 1, pp. 318-362. Cambridge, MA: MIT Press, 1986a.

RUMELHART, D. E., HINTON, G. E. e WILLIAMS, R. J, “Learning representations of back-propagating errors”, **Nature**, vol. 323, pp. 533-536, 1986b.

SCARBOROUGH, D. J., SOMERS, M. J., **Neural Networks in Organizational Research: Applying Pattern Recognition to the Analysis of Organizational Behavior**, 2006.

SEBER, G. A. F. e WILD, G. J. **Nonlinear Regression**. New York: John Wiley & Sons, 1989.

SHAH, S. e PALMIERI, F.. “MEKA - A fast, local algorithm for training feedforward neural networks”, *Proceedings of International Joint Conference on Neural Networks*, San Diego, CA, 3, p. 41-46, 1990.

SINGHAL, S. E WU, L. Training feed-forward networks with extended Kalman filter, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, p. 1187-1190, 1989.

SURATGAR, A. A., TAVAKOLI, M. B. e HOSEINABADI, A. Modified Levenberg-Marquardt Method for Neural Networks Training, *Transactions on Engineering, Computing and Technology*, 2005.

TALEB, N. N. **Dynamic Hedging: Managing Vanilla and Exotic Options**, John Wiley & Sons, New York, 1997.

TAYLOR, J. W. A Quantile Regression Neural Network Approach to Estimating the Conditional Density of Multiperiod Returns, Saïd Business School, University of Oxford, **Journal of Forecasting**, Vol. 19, pp. 299-311, 2000.

TERÄSVIRTA, T. e GRANGER, C. Modeling Nonlinear Economic Relationships, Oxford University Press, 1993.

TERÄSVIRTA, T., VAN DIJK, D. e MEDEIROS M. Linear models, smooth transition autoregressions, and neural networks for forecasting macroeconomic time series: a re-examination. **International Journal of Forecasting**, 2005.

TERÄSVIRTA, T., VAN DIJK, D. e MEDEIROS M. Linear models, smooth transition autoregressions, and neural networks for forecasting macroeconomic time series: a re-examination. **International Journal of Forecasting**, 2005.

TODOROVIC´, B., STANKOVIC´, M. TODOROVIC´-ZARKULA, S. Structurally adaptive RBF network in non-stationary time series prediction, In Proc. IEEE AS-SPCC, p. 224-229, 2000.

TODOROVIC´, B., STANKOVIC´, M., MORAGA, C. Extended Kalman Filter trained Recurrent Radial Basis Function Network in Nonlinear System Identification, Proc. do ICANN 2002, Spain, LNCS 2415, p. 819-824, Springer, 2002.

VAN DIJK, D. Neural networks in finance - gaining predictive edge in the market, **International Journal of Forecasting**, vol. 22, Num. 2, 2006.

WAIBEL, A., HANAZAWA, T., HINTON, G., SHIKANO, K. E LANG, K. J. Phoneme recognition using time-delay neural networks, IEEE Transactions on Acoustics, Speech, and Signal Processing, 1989.

WAN, E. A. Temporal backpropagation for FIR neural networks, IEEE International Joint Conference on Neural Networks, vol. I, pp. 575-580, San Diego, CA, 1990.

WAN, E. A. Time series prediction by using a connectionist network with internal delay lines, 1994.

WAN, E. e MERWE, R. van der, Efficient Derivative-Free Kalman Filters for Online Learning, em Proceedings of the European Symposium on Artificial Neural Networks (ESANN), 2001b.

WAN, E. e MERWE, R. van der, The Square-Root Unscented Kalman Filter for State and Parameter Estimation, Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, IEEE, 2001a.

WARNER, B. e MISRA, M. Understanding Neural Networks as Statistical Tools, The American Statistician, vol. 50, n^o: 4, novembro, 1996.

WERBOS, P. J. "Back-propagation: Past and future", Proceeding of International Conference on Neural Networks, San Diego, CA, 1, p. 343-354, 1988.

WERBOS, P. J. Backpropagation through time: what it does and how to do it, Proceedings of the IEEE, vol. 78, p. 1550-1560, 1990.

WILAMOWSKI, B. M., IPLIKCI, S., KAYNAK, O. e EFE, M. O. An Algorithm for Fast Convergence in Training Neural Networks, IEEE, 2001.

WILLIAMS, R. J. E ZIPSER, D. Gradient-based learning algorithms for recurrent networks and their computational complexity, 1995.

WILLIAMS, R. J. Training recurrent networks using the extended Kalman filter, Proceeding of the International Joint Conference on Neural Networks, vol. 4, p. 241-246, Baltimore, 1992.

Apêndice

Apêndice-A

A matriz $\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}'$ é conhecida como matriz *hat* (chapéu) e possui as seguintes propriedades:

1. $[\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}']' = [\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}']$ (simetria)
2. $[\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}'] \cdot [\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}'] = \mathbf{I}-2\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}'+\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}'\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}'$
 $= [\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}']$ (idempotência)
3. $\mathbf{J}'[\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}'] = \mathbf{J}'-\mathbf{J}' = 0$

Continuando a determinação do valor esperado da soma quadrática dos resíduos da página 4 temos:

$$\begin{aligned}
 &\approx \frac{1}{2}E\left[\mathbf{k}'(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')'(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\mathbf{k}\right] \\
 &\approx \frac{1}{2}E\left[\mathbf{k}'(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}'-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}'+\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}'\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\mathbf{k}\right] \\
 &\approx \frac{1}{2}E\left[\mathbf{k}'(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}'-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}'+\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\mathbf{k}\right] \approx \frac{1}{2}E\left[\mathbf{k}'(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\mathbf{k}\right] \\
 &\approx \frac{1}{2}E\left[(\boldsymbol{\varepsilon}^*+\mathbf{J}\mathbf{w}^*)'(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')(\boldsymbol{\varepsilon}^*+\mathbf{J}\mathbf{w}^*)\right] \approx \frac{1}{2}E\left[(\boldsymbol{\varepsilon}^*(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')+\mathbf{w}^*\mathbf{J}'(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}'))(\boldsymbol{\varepsilon}^*+\mathbf{J}\mathbf{w}^*)\right] \\
 &\approx \frac{1}{2}E\left[\boldsymbol{\varepsilon}^{*\prime}(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\boldsymbol{\varepsilon}^*+\mathbf{w}^{*\prime}\mathbf{J}'(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\boldsymbol{\varepsilon}^*+\boldsymbol{\varepsilon}^{*\prime}(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\mathbf{J}\mathbf{w}^*+\mathbf{w}^{*\prime}\mathbf{J}'(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\mathbf{J}\mathbf{w}^*\right] \\
 &\approx \frac{1}{2}E\left[\boldsymbol{\varepsilon}^{*\prime}(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\boldsymbol{\varepsilon}^*+\mathbf{w}^{*\prime}\mathbf{J}'(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\boldsymbol{\varepsilon}^*+\boldsymbol{\varepsilon}^{*\prime}(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\mathbf{J}\mathbf{w}^*+(\mathbf{w}^{*\prime}\mathbf{J}')\mathbf{J}\mathbf{w}^*-\mathbf{w}^{*\prime}\mathbf{J}'\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}'\mathbf{J}\mathbf{w}^*\right] \\
 &\approx \frac{1}{2}E\left[\boldsymbol{\varepsilon}^{*\prime}(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\boldsymbol{\varepsilon}^*+\mathbf{w}^{*\prime}\mathbf{J}'(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\boldsymbol{\varepsilon}^*+\boldsymbol{\varepsilon}^{*\prime}(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\mathbf{J}\mathbf{w}^*+\mathbf{w}^{*\prime}\mathbf{J}'\mathbf{J}\mathbf{w}^*-\mathbf{w}^{*\prime}\mathbf{J}'\mathbf{J}\mathbf{w}^*\right] \\
 &\approx \frac{1}{2}E\left[\boldsymbol{\varepsilon}^{*\prime}(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\boldsymbol{\varepsilon}^*+\mathbf{w}^{*\prime}\mathbf{J}'(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\boldsymbol{\varepsilon}^*+\boldsymbol{\varepsilon}^{*\prime}(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\mathbf{J}\mathbf{w}^*\right] \\
 &\approx \frac{1}{2}E\left[\boldsymbol{\varepsilon}^{*\prime}(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\boldsymbol{\varepsilon}^*+\mathbf{w}^{*\prime}\mathbf{J}'(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\boldsymbol{\varepsilon}^*+\boldsymbol{\varepsilon}^{*\prime}(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\mathbf{J}\mathbf{w}^*\right] \\
 &\approx \frac{1}{2}\left[E\left[\boldsymbol{\varepsilon}^{*\prime}(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\boldsymbol{\varepsilon}^*\right]+E\left[\mathbf{w}^{*\prime}\mathbf{J}'(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\boldsymbol{\varepsilon}^*\right]+E\left[\boldsymbol{\varepsilon}^{*\prime}(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\mathbf{J}\mathbf{w}^*\right]\right] \\
 &\approx E\left[\frac{1}{2}\boldsymbol{\varepsilon}^{*\prime}(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\boldsymbol{\varepsilon}^*\right]+\frac{1}{2}\mathbf{w}^{*\prime}\mathbf{J}'(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')E[\boldsymbol{\varepsilon}^*]+\frac{1}{2}(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\mathbf{J}\mathbf{w}^*E[\boldsymbol{\varepsilon}^*]
 \end{aligned}$$

Como o erro é considerado pequeno, multiplicar por 1/2 o erro ao quadrado não afetará de maneira significativa o resultado, dessa forma suprimimos esse valor do restante da demonstração.

$$\begin{aligned}
 &\approx E\left[\boldsymbol{\varepsilon}^{*\prime}(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\boldsymbol{\varepsilon}^*\right] \approx E\left[\text{tr}(\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}')\boldsymbol{\varepsilon}^*\boldsymbol{\varepsilon}^{*\prime}\right] \approx \text{tr}\left[\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}'\right]E[\boldsymbol{\varepsilon}^*\boldsymbol{\varepsilon}^{*\prime}] \\
 &\approx \text{tr}\left[\mathbf{I}-\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}'\right]\sigma^2 \approx \text{tr}[\mathbf{I}]\sigma^2-\text{tr}\left[\mathbf{J}(\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}'\right]\sigma^2 \approx \text{tr}\left[\mathbf{I}_{n \times n}\right]\sigma^2-\text{tr}\left[\mathbf{I}_{(r+1) \times (r+1)}\right]\sigma^2
 \end{aligned}$$

$$E[SQR] \approx n\sigma^2 - \sigma^2(r+1)$$

$$E[SQR] \approx \sigma^2(n-r-1)$$

Dessa forma um estimador assintótico não-viesado para σ^2 é então dado por: $s^2 = \frac{SQR}{(n-r-1)}$

Apêndice-B

(* Carregar todos os pacotes que serao necessarios para realizar as operacoes*)

```
Needs["TimeSeries`TimeSeries`"]
<<NeuralNetworks`
<<Graphics`MultipleListPlot`
<<Statistics`LinearRegression`
<<Statistics`ANOVA`
<<Graphics`Legend`
<<Statistics`ContinuousDistributions`
<<Graphics`Colors`
```

(* PRIMEIRA PARTE
CONSTRUCAO DO MODELO AR-GARCH E DA REDE NEURAL QUE FAZ A
PREDICAO DESSE MODELO *)

```
ClearAll[ModeloARGARCH];
ClearAll[ModeloARGARCHestima];
ClearAll[ModeloARGARCHvalida];
ClearAll[ys];
ClearAll[model1];
ye = {};
yv = {};
Previsao = {};
flatPrevisao2 = {};
```

```
ModeloARGARCH = {};
```

```
ModeloARGARCHestima = ModeloARGARCH[[Range[1299]]];
ModeloARGARCHvalida = ModeloARGARCH[[Range[1300,1541]]];
```

```
ModeloARGARCHestima;
For[i=1,i<1300,i++,AppendTo[ye,{ModeloARGARCHestima[[i]]}]];
ye;
```

```
ModeloARGARCHvalida;
For[i=1,i<243,i++,AppendTo[yv,{ModeloARGARCHvalida[[i]]}]];
yv;
```

```
ListPlot[Flatten[ye], PlotJoined→ True, Frame→ True, PlotRange→ {-3,3}];
ListPlot[Flatten[yv], PlotJoined→ True, Frame→ True, PlotRange→ {-3,3}];
```

```
Dimensions[ye]
Dimensions[yv]
```

```
{model1,fitrecord}=NeuralARFit[ye,10,FeedForwardNet,{3}];
```

```
NetInformation[model1];
```

```
Previsoes;
```

```
Previsao= NetPredict[yv,model1];
```

```
NetComparePlot[yv,model1,PredictHorizon→1,PlotLegend→{"Verdadeiro","Simulado"},PlotStyle→{Hue[.6],Hue[.8]}]
```

```
NetComparePlot[yv,model1,PredictHorizon→1,PlotLegend→{"Verdadeiro","Simulado"}]
```

```
yv;
```

```
Previsao;
```

```
(* Definicao das variaveis *)
```

```
dados={};dadosaux={};errors =
```

```
{ };n=100;p=6;z={};zaux={};zzero={};zinter={};zinteraux={};IP1aux={};IP2aux={};IP1={};IP2={};zinterauxconf={};zinterconf={};IC1auxconf={};IC2auxconf={};IC1conf={};IC2conf={};SortModeloARGARCHvalida={};
```

```
SortSimulaRede={};SortIP1={};SortIP2={};SortIC1conf={};SortIC2conf={};FIC1={};
```

```
DentroIC1={};FIC2={};DentroIC2={};ForaIP1={};DentroIP1={};ForaIP2={};DentroIP2={};
```

```
e={};errospred={};flaterrospred={};SortPredicaoEviews={};SortICsuperiorEviews={};
```

```
SortICinferiorEviews={};
```

```
For[t=1300, t<1542, t++,
```

```
  dadosaux= {ModeloARGARCH[t-5], ModeloARGARCH[t-4], ModeloARGARCH[t-3], ModeloARGARCH[t-2],  
  ModeloARGARCH[t-1], ModeloARGARCH[t]}; AppendTo[dados, dadosaux];
```

```
{yhat=Regress[dados, {1, j1, j2, j3, j4, j5}, {j1, j2, j3, j4, j5}, RegressionReport→{FitResiduals,  
  SinglePredictionCITable, ParameterConfidenceRegion}];
```

```
errors=FitResiduals/.yhat;
```

```
{observed, predicted, se, ci}=Transpose[(SinglePredictionCITable/.yhat)[[1]]];
```

```
ListPlot[Transpose[{predicted, errors}]];
```

```
yhat=Fit[dados, {1, j1, j2, j3, j4, j5}, {j1, j2, j3, j4, j5};
```

$$SQR = \sum_{i=1}^{242} errors[[i]]^2;$$


```

MultipleListPlot[ModeloARGARCHvalida,Flatten[Previsao],Flatten[IC1conf],Flatten[IC2c
onf],
PlotStyle→{Hue[0.5],Hue[0.6],Hue[0.9],Hue[0.9]},
PlotLegend→{"Serie Original","Previsao","IC-Inferior","IC-Superior"},
PlotJoined→True,Frame→True,SymbolShape→None]

```

(* QUARTA PARTE
PLOTAGEM DOS INTERVALOS DE CONFIANCA E PREDICAO *)

```

SortModeloARGARCHvalida=Sort[ModeloARGARCHvalida];
SortSimulaRede=Sort[Flatten[Previsao]];
SortIP1=Sort[Flatten[IP1]];
SortIP2=Sort[Flatten[IP2]];
SortIC1conf=Sort[Flatten[IC1conf]];
SortIC2conf=Sort[Flatten[IC2conf]];

```

```

MultipleListPlot[SortModeloARGARCHvalida,SortSimulaRede,SortIP1,SortIP2,
PlotLegend→{"Serie Original","Previsao","IPred-Inferior","IPred-Superior"},
PlotStyle→{GrayLevel[0],Dashing[ {.03}],Dashing[ {.01}],Dashing[ {.01}]},
PlotJoined→True,Frame→True,SymbolShape→None,LegendSize → {1.5,0.3},
LegendPosition→{-0.7,-0.95}]

```

```

MultipleListPlot[SortModeloARGARCHvalida,SortSimulaRede,SortIC1conf,SortIC2conf,
PlotLegend→{"Serie Original","Previsao","ICConf-Inferior","ICConf-Superior"},
PlotStyle→{GrayLevel[0],Dashing[ {.03}],Dashing[ {.01}],Dashing[ {.01}]},
PlotJoined→True,Frame→True,SymbolShape→None,LegendSize → {1.5,0.3},
LegendPosition→{-0.7,-0.95}]

```

(* QUINTA PARTE
ESTATÍSTICAS *)

(* Valores fora do IC *)

```
For[u=1,u<243,u++,
```

```

If[SortModeloARGARCHvalida[[u]]>SortIC1conf[[u]],AppendTo[FIC1,SortIC1conf[[u]],
AppendTo[DentroIC1,SortIC1conf[[u]]]];

```

```

If[SortModeloARGARCHvalida[[u]]<SortIC2conf[[u]],AppendTo[FIC2,SortIC2conf[[u]],
AppendTo[DentroIC2,SortIC2conf[[u]]]];

```

```

FIC1;
FIC2;

```

(* Valores fora do IP *)

```
For[c=1,c<243,c++,
```

```

If[SortModeloARGARCHvalida[[c]]>SortIP1[[c]],AppendTo[ForaIP1,SortIP1[[c]],
AppendTo[DentIP1,SortIP1[[c]]]];
If[SortModeloARGARCHvalida[[c]]<SortIP2[[c]],AppendTo[ForaIP2,SortIP2[[c]],
AppendTo[DentIP2,SortIP2[[c]]]];
ForaIP1;
ForaIP2;

SaiudoIC1
Dimensions[FIC1]

SaiudoIC2
Dimensions[FIC2]

TotalsaiudoIC=Dimensions[FIC1]+Dimensions[FIC2]

SaiudoIntervaloP1
Dimensions[ForaIP1]

SaiudoIntervaloP2
Dimensions[ForaIP2]

TotalsaiudoIntervaloP=Dimensions[ForaIP1]+Dimensions[ForaIP2]

For[h=1,h<243,h++,
e=SortSimulaRede[[h]]-SortModeloARGARCHvalida[[h]];
AppendTo[errospred,e]];
flaterrospred = Flatten[errospred];
flatSortModeloARGARCHvalida= Flatten[SortModeloARGARCHvalida];
flatSortSimulaRede=Flatten[SortSimulaRede];

```


(*

(* Calculo do RMSE *)

$$\text{RMSErede} = \sqrt{\frac{\sum_{i=1}^{242} \text{errospred}[i] * \text{errospred}[i]}{242}}$$

(* Calculo do TIC *)

$$\text{TIC} = \sqrt{\frac{\sum_{i=1}^{242} \text{errospred}[i] * \text{errospred}[i]}{242}} \left/ \left(\sqrt{\frac{\sum_{i=1}^{242} \text{flatSortSimulaRede}[i] * \text{flatSortSimulaRede}[i]}{242}} + \sqrt{\left(\sum_{i=1}^{242} \frac{1}{242} (\text{flatSortModeloARGARCHvalida}[i] * \text{flatSortModeloARGARCHvalida}[i]) \right)} \right) \right.$$

*)

(* Histograma dos erros de predicao *)

```
Histogram[errospred, Frame -> True, BarStyle -> {GrayLevel[0.4]},  
PlotLegend -> {"Histograma do Erro de Prediçao"}, AxesLabel -> {"Erro de Predicao", "Qtd"}]
```

```
MultipleListPlot[SortModeloARGARCHvalida,  
PlotStyle -> {GrayLevel[0], Dashing[ {.03}], Dashing[ {.01}], Dashing[ {.01}]},  
PlotLegend -> {"Serie Original", "Previsao", "IPred-Inferior", "IPred-Superior"},  
PlotJoined -> True, Frame -> True, SymbolShape -> None, LegendSize -> {1.5, 0.3},  
LegendPosition -> {-0.7, -0.95}]
```

```
Previsao // TableForm
```

```
ColumnForm[IC1conf]
```

```
ColumnForm[IC2conf]
```

```
ColumnForm[IP1]
```

```
ColumnForm[IP2]
```

```

For[i = 1, i < 243, i++,
  AppendTo[flatPrevisao2, Previsao[[i]] - ModeloARGARCHvalida[[i]]];

```

$$RMSErede = \sqrt{\sum_{i=1}^{242} \frac{\text{flatPrevisao2}[[i]] * \text{flatPrevisao2}[[i]]}{242}};$$

$$\begin{aligned}
TIC = & \left(\sqrt{\left(\frac{1}{242} \left((\text{Flatten}[\text{Previsao}] - \text{Flatten}[\text{ModeloARGARCHvalida}]) \cdot (\text{Flatten}[\text{Previsao}] - \right. \right. \right. \\
& \left. \left. \left. \text{Flatten}[\text{ModeloARGARCHvalida}]) \right) \right) \right) / \left(\sqrt{\frac{\text{Flatten}[\text{Previsao}] \cdot \text{Flatten}[\text{Previsao}]}{242}} + \right. \\
& \left. \sqrt{\left(\frac{1}{242} (\text{Flatten}[\text{ModeloARGARCHvalida}] \cdot \text{Flatten}[\text{ModeloARGARCHvalida}]) \right) \right) };
\end{aligned}$$

```

RMSErede
TIC
TimeUsed[]
Null

```

Apêndice-C

Programa MATLAB para o algoritmo RNA-EKF

```
% ALGORITMO UTILIZANDO FILTRO DE KALMAN ESTENDIDO (EKF) PARA TREINAMENTO
DE REDES NEURAIS
% DO TIPO FEEDFORWARD - MULTI LAYER PERCEPTRON (FEEDFORWARD-MLP)
% Versao 0.0 (zero)
clear;
echo off;
% DEFINICAO DOS PARAMETROS UTILIZADOS NO ALGORITMO:
% =====
N ;                % NUMERO DE INSTANTES DE TEMPO.
t ;                % TEMPO.
x0;                % ENTRADA INICIAL.
x ;                % VETOR DE ENTRADA.
y ;                % VETOR DE SAIDA.
x(1,1) ;          % PRIMEIRA ENTRADA.
R_Atual;           % VARIANCIA DO ERRO DE MEDIDA.
Q_Atual;           % VARIANCIA DO ERRO DO PROCESSO.
NeuronHid;         % NEURONIOS NA CAMADA ESCONDIDA.
NeuronSaida;       % NEURONIOS NA CAMADA DE SAIDA.
VarInicial;        % VARIANCIA INICIAL PARA OS PESOS.
COVKalmanR;        % COVARIANCIA DOS ERROS DA EQUACAO DE MEDIDA.
COVKalmanQ;        % COVARIANCIA DOS ERROS DA EQUACAO DO PROCESSO.
KalmanP;           % COVARIANCIA DOS PESOS INICIAIS.
% DEFINICAO DOS ERROS DE MEDIDA E DO PROCESSO:
% =====
v = randn(N,1);
w = randn(N,1);
% GERACAO DOS DADOS DE ENTRADA E DE SAIDA:
% =====
J = [];
for t=2:N,
    x(t,1) = J(t-1,1) ;
    y(t,1) = J(t,1) ;
end;
fprintf('\n')
fprintf('Treinamento de uma RNA Feedforward utilizando o EKF')
fprintf('\n')
% DESEMPENHO DO EKF EM TRENIAR A RN FEEDFORWARD-MLP:
% =====
Crono=clock;
[Previsao_EKF,Pesos,PesosR,PR,Inovacoes] =
RN_MLP_EKF(x',y',NeuronHid,NeuronSaida, COVKalmanR, COVKalmanQ,
COVKalmanP,VarInicial,N);
e = Previsao_EKF' - J;
% DURACAO DOS CALCULOS
Duracao_Calculos=etime(clock,Crono);
fprintf('Duracao da previsao utilizando o EKF = %d
seconds.\n',Duracao_Calculos)
fprintf('\n')
```

```

fprintf('RELATORIO : ESTATISTICAS DE AJUSTE DO MODELO')
fprintf('\n')
% CALCULO DO RMSE
rmse = norm(e)/sqrt(N);
fprintf('RMSE produzido pelo EKF      = %d',rmse)
fprintf('\n')
% CALCULO DO MAE
mae = mad(e);
fprintf('MAE produzido pelo EKF      = %d',mae)
fprintf('\n')
% CALCULO DO MAPE
% CALCULO DO TIC
tic = (norm(e)/sqrt(N))/((norm(Previsao_EKF)/sqrt(N))+ (norm(J)/sqrt(N)));
fprintf('TIC produzido pelo EKF      = %d',tic)
fprintf('\n')
% APRESENTACAO DOS RESULTADOS:
% =====
figure(5)
clf;
plot(1:length(x),J,'g',1:length(x),Previsao_EKF,'b')
legend('Valor Verdadeiro','Previsao usando o EKF');
ylabel('Previsao um passo a frente','fontsize',10)
xlabel('Tempo','fontsize',10)

```