

UNIVERSIDADE FEDERAL DO MARANHÃO - UFMA  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA - CCET  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE - PPGEE

**SELMO EDUARDO RODRIGUES JÚNIOR**

**METODOLOGIA EVOLUTIVA PARA PREVISÃO INTELIGENTE DE SÉRIES  
TEMPORAIS SAZONAIS BASEADA EM ESPAÇO DE ESTADOS  
NÃO-OBSERVÁVEIS**

São Luís - MA

2017



**SELMO EDUARDO RODRIGUES JÚNIOR**

**METODOLOGIA EVOLUTIVA PARA PREVISÃO INTELIGENTE DE SÉRIES  
TEMPORAIS SAZONAIS BASEADA EM ESPAÇO DE ESTADOS  
NÃO-OBSERVÁVEIS**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão como requisito para obtenção do título de mestre em engenharia de eletricidade na área de concentração de Automação e Controle.

Orientador: Prof. Dr. Ginalber Luiz de Oliveira Serra

São Luís - MA

2017

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).  
Núcleo Integrado de Bibliotecas/UFMA

Rodrigues Júnior, Selmo Eduardo.

Metodologia Evolutiva para Previsão Inteligente de Séries Temporais Sazonais Baseada em Espaço de Estados Não-Observáveis / Selmo Eduardo Rodrigues Júnior. - 2017.  
134 p.

Orientador(a): Ginalber Luiz de Oliveira Serra.

Dissertação (Mestrado) - Programa de Pós-graduação em Engenharia de Eletricidade/ccet, Universidade Federal do Maranhão, São Luís, 2017.

1. Análise Espectral Singular. 2. Componentes Não-Observáveis. 3. Modelos no Espaço de Estados. 4. Previsão de Séries Temporais Sazonais. 5. Rede Neuro-Fuzzy Takagi-Sugeno Evolutiva. I. Luiz de Oliveira Serra, Ginalber. II. Título.

**SELMO EDUARDO RODRIGUES JÚNIOR**

**METODOLOGIA EVOLUTIVA PARA PREVISÃO INTELIGENTE DE SÉRIES  
TEMPORAIS SAZONAIS BASEADA EM ESPAÇO DE ESTADOS  
NÃO-OBSERVÁVEIS**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão como requisito para obtenção do título de mestre em engenharia de eletricidade na área de concentração de Automação e Controle.

Trabalho aprovado em São Luís - MA, 26 de janeiro de 2017.

---

**Prof. Dr. Ginalber Luiz de Oliveira Serra**  
Orientador

---

**Prof. Dr. Omar Andres Carmona Cortes**  
Primeiro Membro da Banca

---

**Prof. Dr. Francisco das Chagas de Souza**  
Segundo Membro da Banca



## **AGRADECIMENTOS**

Agradeço a Deus por ter me iluminado, me guiado por esse caminho cheio de desafios e por ter me oferecido todas as condições, saúde e força para superar as dificuldades.

Agradeço muito ao professor Dr. Ginalber Luiz de Oliveira Serra pela orientação, ensino, confiança e todo o suporte na elaboração deste trabalho.

Aos meus pais e a todos os meus familiares que sempre torceram e oraram por mim e que me ensinaram os princípios e valores que levarei comigo para sempre. Agradeço também a meu avô que não está mais entre nós e, mesmo com suas limitações, me deu forças para ser melhor sempre.

A minha namorada Emilly Mota por todo seu companheirismo, carinho, amizade e amor que foram essenciais para o desenvolvimento dessa pesquisa.

A Universidade Federal do Maranhão e a CAPES pelo apoio estrutural e financeiro para que essa pesquisa fosse desenvolvida.

Ao Instituto Federal do Maranhão e a todas as pessoas com que convivi ao longo desses 2 anos, sendo que a comunhão com os amigos e o aprendizado em grupo foram experiências enriquecedoras na minha formação. Agradeço aos companheiros de estudo Adriano, Anderson Pablo, Arnaldo, Danúbia, Edson, Jéssica, Luís Miguel, Hugo, Orlando e Washington por todo o apoio que me deram.

Ao professor José Pinheiro de Moura por ter me aconselhado a fazer o mestrado, sendo muito importante para que eu tomasse essa decisão.

Agradeço a todos que, diretamente ou indiretamente, fizeram parte da minha formação acadêmica. O meu muito obrigado a vocês.





*"Entrega teu caminho ao Senhor;  
confia Nele, e Ele o fará."  
(Salmos 37:5)*



## RESUMO

Esse trabalho propõe uma nova metodologia para modelagem baseada em uma Rede *Neuro-Fuzzy* Takagi-Sugeno (RNF-TS) evolutiva para a previsão de séries temporais sazonais. A RNF-TS considera as componentes não-observáveis extraídas a partir da série para evoluir, ou seja, adaptar e ajustar sua estrutura, sendo que a quantidade de regras *fuzzy* dessa rede pode aumentar ou ser reduzida conforme o comportamento das componentes. O método utilizado para extrair as componentes é uma versão recursiva desenvolvida nessa pesquisa baseada na técnica de Análise Espectral Singular (AES). A metodologia proposta tem como princípio dividir para conquistar, isto é, dividir um problema em subproblemas mais fáceis de lidar, realizando a previsão separadamente de cada componente já que apresentam comportamentos dinâmicos mais simples de prever. As proposições do consequente das regras *fuzzy* são modelos lineares no espaço de estados, sendo que os estados são os próprios dados das componentes não-observáveis. Quando há observações disponíveis da série temporal, o estágio de treinamento da RNF-TS é realizado, ou seja, a RNF-TS evolui sua estrutura e adapta seus parâmetros para realizar o mapeamento entre os dados das componentes e a amostra disponível da série temporal original. Caso contrário, se essa observação não está disponível, a rede aciona o estágio de previsão, mantendo sua estrutura fixa e usando os estados dos consequentes das regras *fuzzy* para realimentar os dados das componentes para a RNF-TS. A RNF-TS foi avaliada e comparada com outras técnicas recentes e tradicionais para previsão de séries temporais sazonais, obtendo resultados competitivos e vantajosos em relação a outras pesquisas. Este trabalho apresenta também um estudo de caso da metodologia proposta para detecção em tempo-real de anomalias baseada em dados de eletrocardiogramas de um paciente.

**Palavras-chaves:** Previsão de Séries Temporais Sazonais, Componentes Não-Observáveis, Modelos no Espaço de Estados, Rede *Neuro-Fuzzy* Takagi-Sugeno Evolutiva, Análise Espectral Singular.



## ABSTRACT

This paper proposes a new methodology for modelling based on an evolving Neuro-Fuzzy Network Takagi-Sugeno (NFN-TS) for seasonal time series forecasting. The NFN-TS use the unobservable components extracted from the time series to evolve, i.e., to adapt and to adjust its structure, where the number of fuzzy rules of this network can increase or reduced according the components behavior. The method used to extract the components is a recursive version developed in this paper based on the Spectral Singular Analysis (SSA) technique. The proposed methodology has the principle divide to conquer, i.e., it divides a problem into easier subproblems, forecasting separately each component because they present dynamic behaviors that are simpler to forecast. The consequent propositions of fuzzy rules are linear state space models, where the states are the unobservable components data. When there are available observations from the time series, the training stage of NFN-TS is performed, i.e., the NFN-TS evolves its structure and adapts its parameters to carry out the mapping between the components data and the available sample of original time series. On the other hand, if this observation is not available, the network considers the forecasting stage, keeping its structure fixed and using the states of consequent fuzzy rules to feedback the components data to NFN-TS. The NFN-TS was evaluated and compared with other recent and traditional techniques for forecasting seasonal time series, obtaining competitive and advantageous results in relation to other papers. This paper also presents a case study of proposed methodology for real-time detection of anomalies based on a patient's electrocardiogram data.

**Key-words:** Seasonal Time Series Forecasting, Unobservable Components, Space State Model, Evolving Neuro-Fuzzy Takagi-Sugeno, Singular Spectral Analysis.



## LISTA DE ILUSTRAÇÕES

Figura 1 – Ruído Branco. . . . .	41
Figura 2 – Função de Autocorrelação Ideal do Ruído Branco. . . . .	42
Figura 3 – Função de Autocorrelação Amostral do Ruído Branco. . . . .	42
Figura 4 – Série Temporal da Taxa de Produção de Leite. . . . .	44
Figura 5 – Função de Autocorrelação Amostral da Série de Taxa de Produção de Leite. . . . .	44
Figura 6 – Série Temporal Mackey-Glass. . . . .	46
Figura 7 – Abrangência dos Modelos Puramente Adaptativos e dos Modelos Evolutivos. . . . .	63
Figura 8 – Princípio de Criação e Remoção de <i>Clusters</i> no Agrupamento Evolutivo. . . . .	67
Figura 9 – Espaço de Dados Agrupado por <i>Clusters</i> Sobrepostos. . . . .	68
Figura 10 – Formato dos <i>Clusters</i> Quanto a Distância de Mahalanobis (Elipse) e Euclidiana (Círculo). . . . .	70
Figura 11 – Estrutura da Rede <i>Neuro-Fuzzy</i> Takagi-Sugeno Proposta. . . . .	79
Figura 12 – Ciclo de Execução do Algoritmo AESR. . . . .	82
Figura 13 – Quantidade de Regras <i>Fuzzy</i> ou <i>Clusters</i> e Resultados da RNF-TS para a Série Temporal de <b>Livros</b> . . . . .	103
Figura 14 – Quantidade de Regras <i>Fuzzy</i> ou <i>Clusters</i> e Resultados da RNF-TS para a Série Temporal de <i>Roupas</i> . . . . .	104
Figura 15 – Quantidade de Regras <i>Fuzzy</i> ou <i>Clusters</i> e Resultados da RNF-TS para a Série Temporal de <b>Móveis</b> . . . . .	104
Figura 16 – Quantidade de Regras <i>Fuzzy</i> ou <i>Clusters</i> e Resultados da RNF-TS para a Série Temporal de <b>Hardware</b> . . . . .	105
Figura 17 – Resultados de Treinamento e Previsão da RNF-TS para a Série Temporal <b>Papel</b> . . . . .	106
Figura 18 – Componente Não-Observável $c_k^1$ da Série Temporal <b>Papel</b> . . . . .	106
Figura 19 – Componente Não-Observável $c_k^2$ da Série Temporal <b>Papel</b> . . . . .	107
Figura 20 – Componente Não-Observável $c_k^3$ da Série Temporal <b>Papel</b> . . . . .	107
Figura 21 – Quantidade de Regras <i>Fuzzy</i> ou <i>Clusters</i> para a Série Temporal <b>Papel</b> . . . . .	108
Figura 22 – Densidade do Vetor de Dados Padronizado $\hat{\mathbf{d}}_k$ para Avaliação da Condição A (Equação 4.17). . . . .	108
Figura 23 – Comportamento do Raio de um <i>Cluster</i> Específico Gerado em $k = 18$ . . . . .	109
Figura 24 – Utilidade de um <i>Cluster</i> Particular Gerado em $k = 18$ . . . . .	110
Figura 25 – Suporte de um <i>Cluster</i> Particular Gerado em $k = 18$ . . . . .	110
Figura 26 – Grau de Ativação Normalizado de um <i>Cluster</i> Particular Gerado em $k = 18$ . . . . .	111
Figura 27 – Eletrocardiograma de um Paciente. . . . .	112
Figura 28 – Quantidade de Regras <i>Fuzzy</i> ou <i>Clusters</i> para a Série <b>Eletrocardiograma</b> . . . . .	112
Figura 29 – Componente Não-Observável $c_k^1$ da Série <b>Eletrocardiograma</b> . . . . .	113

Figura 30 – Componente Não-Observável $c_k^2$ da Série <b>Eletrocardiograma</b> . . . . .	113
Figura 31 – Componente Não-Observável $c_k^3$ da Série <b>Eletrocardiograma</b> . . . . .	114
Figura 32 – Componente Não-Observável $c_k^4$ da Série <b>Eletrocardiograma</b> . . . . .	114



## LISTA DE TABELAS

Tabela 1	–	Comparação das Metodologias de Acordo com o Critério SMAPE (Melhores Resultados em Negrito). . . . .	100
Tabela 2	–	Comparação das Metodologias de Acordo com o Critério MASE (Melhores Resultados em Negrito). . . . .	100
Tabela 3	–	Valores dos Parâmetros da RNF-TS para cada Série Temporal e Ambos Horizontes de Previsão. . . . .	100
Tabela 4	–	Comparação dos Resultados para as Séries Temporais Usando o RMSE, com Melhores Resultados em Negrito (T - Estágio de Treinamento e P - Estágio de Previsão). . . . .	102
Tabela 5	–	Valores dos Parâmetros da RNF-TS para cada Série Temporal no Experimento 2.	103



## LISTA DE ABREVIATURAS E SIGLAS

ADANN	<i>Automatic Design of Artificial Neural Networks</i> - Projeto Automático de Redes Neurais Artificiais.
AES	Análise Espectral Singular.
AESR	Análise Espectral Singular Recursiva.
ANN	<i>Artificial Neural Networks</i> - Redes Neurais Artificiais.
AR	<i>Autoregressive Model</i> - Modelo Autorregressivo.
ARCH	<i>Autoregressive Conditional Heteroskedasticity Model</i> - Modelo de Heteroscedasticidade Condicional Autorregressivo.
ARIMA	<i>Autoregressive Integrated Moving Average</i> - Modelo Autorregressivo Integrado de Médias Móveis.
ARMA	<i>Autoregressive Moving Average</i> - Modelo Autorregressivo de Médias Móveis.
DENFIS	<i>Dynamic Evolving Neuro-Fuzzy Inference System</i> - Sistema de Inferência Neuro-Fuzzy Evolutivo Dinâmico
DSDT	<i>Deseasonalized and Detrended</i> - Sem Sazonalidade e Sem Tendência.
eFPT	<i>evolving Fuzzy Pattern Tree</i> - Árvores de Padrões Fuzzy Evolutivas.
eFT	<i>evolving Fuzzy Linear Regression Trees</i> - Árvores de Regressão Lineares Fuzzy Evolutivas).
eMG	<i>Multivariable Gaussian Evolving Fuzzy System</i> - Sistema Fuzzy Evolutivo Multivariável Gaussiano.
EQM	Erro Quadrático Médio.
eT2FIS	<i>evolving Type-2 Neural Fuzzy Inference System</i> - Sistema de Inferência Neuro-Fuzzy Tipo-2 Evolutivo.
eTS	<i>evolving Takagi-Sugeno</i> - Takagi-Sugeno evolutivo.
eTS-LS-SVM	<i>evolving Takagi-Sugeno Least Square Support Vector Machine</i> - Máquina de Vetores de Suporte por Mínimos Quadrados Takagi-Sugeno Evolutiva.
FANN	<i>Fuzzy Artificial Neural Networks</i> - Redes Neurais Artificiais Fuzzy.

FLEXFIS	<i>Flexible Fuzzy Inference Systems</i> - Sistemas de Inferência <i>Fuzzy</i> Flexíveis.
FSVM	<i>Fuzzy Support Vector Machine</i> - Máquina de Vetores de Suporte <i>Fuzzy</i> .
HW	Holt-Winters.
KDE	<i>Kernel Density Estimation</i> - Estimação de Densidade Kernel.
L-FWRLS	<i>Local Fuzzy Weighted Recursive Least Squares</i> - Mínimos Quadrados Recursivo Ponderados <i>Fuzzy</i> Local.
MA	<i>Moving Average</i> - Modelo de Médias Móveis.
MASE	<i>Mean Absolute Scaled Error</i> - Erro Médio Absoluto.
NFN-TS	<i>Neuro-Fuzzy Network Takagi-Sugeno</i> .
RSSA	<i>Recursive Singular Spectral Analysis</i> .
RBF-AR	<i>Radial Basis Function Based on Autoregressive Model</i> - Rede Neural de Função de Base Radial Baseado em Modelos Autorregressivos.
RDE	<i>Recursive Density Estimation</i> - Estimação Recursiva da Densidade.
RMSE	<i>Root Mean Square Error</i> - Raiz Quadrada do Erro Quadrático Médio.
RNF-TS	Rede <i>Neuro-Fuzzy</i> Takagi-Sugeno.
SAFIS	<i>Sequential Adaptive Fuzzy Inference System</i> - Sistema de Inferência <i>Fuzzy</i> Adaptativo e Sequencial.
SARIMA	<i>Seasonal Autoregressive Integrated Moving Average</i> - Modelo Autorregressivo Integrado de Médias Móveis Sazonal.
SMAPE	<i>Symmetric Mean Absolute Percentage Error</i> - Erro Médio Absoluto Percentual Simétrico.
SSA	<i>Singular Spectral Analysis</i> .
STAR	<i>Smooth Transition Autoregression Model</i> - Modelo Autorregressivo de Transição Suave.
STL	<i>Seasonal and Trend Decomposition using Loess</i> - Decomposição da Tendência e da Sazonalidade usando <i>Loess</i>
SVD	<i>Singular Value Decomposition</i> - Decomposição em Valores Singulares.
SVM	<i>Support Vector Machine</i> - Máquina de Vetores de Suporte.

TDNN      *Time Delay Neural Network* - Rede Neural com Atraso de Tempo.  
xTS      *eXtended Takagi-Sugeno* - Takagi-Sugeno estendido.



## LISTA DE SÍMBOLOS

### Escalares:

$A$	Coefficiente de suavização de Holt-Winters para o nível.
$b_k$	Elemento para o cálculo da densidade $D_k(\hat{\mathbf{d}}_k)$ .
$c_k^j$	Valor para a componente não-observável $j$ no instante $k$ , para $j = 1, \dots, n$ .
$\tilde{c}_k^j$	Previsão para a componente não-observável $j$ no instante $k$ , para $j = 1, \dots, n$ .
$C$	Coefficiente de suavização de Holt-Winters para a tendência.
$d$	Quantidade de autovalores não-nulos do AES e AESR.
$D$	Coefficiente de suavização de Holt-Winters para a sazonalidade.
$D_k$	Valor para a densidade no instante $k$ .
$E^G$	Função custo global.
$E^L$	Função custo local.
$f_X$	Função densidade de probabilidades.
$F_X$	Função distribuição de probabilidades.
$g_k^j$	Elemento para o cálculo da densidade $D_k(\hat{\mathbf{d}}_k)$ , para $j = 1, \dots, n + 1$ .
$h$	Horizonte de previsão.
$i$	Índice para identificação das regras <i>fuzzy</i> ou <i>clusters</i> .
$I$	Grupo genérico de índices do AES e do AESR.
$j$	Índice para identificação das componentes não-observáveis.
$k$	Instante de tempo discreto.
$K$	Quantidade de vetores $\mathbf{m}_q$ do AES.
$K_k$	Quantidade de vetores $\mathbf{m}_q$ do AESR no instante $k$ .
$L$	Dimensão dos vetores $\mathbf{m}_q$ do AES e AESR.
$n$	Quantidade de componentes não-observáveis.

$N$	Quantidade total de amostras de uma série temporal ou dados usados no treinamento da RNF-TS.
$\tilde{p}_k^q$	Último elemento do autovetor $\mathbf{p}_k^l$ , para $l = 1, \dots, d$ .
$p_X$	Função massa de probabilidades.
$P(A)$	Probabilidade de um evento $A$ .
$q$	Índice genérico.
$R$	Quantidade de regras <i>fuzzy</i> ou <i>clusters</i> .
$s$	Período sazonal.
$s_k$	Amostra da componente não-observável sazonal no instante $k$ .
$S_k^i$	Suporte do <i>cluster</i> $i$ no instante $k$ .
$t$	Instante de tempo contínuo.
$t_k$	Componente não-observável tendência no instante $k$ .
$tc_k$	Amostra da componente não-observável de tendência-cíclica no instante $k$ .
$T$	Período amostral.
$T_{i*}$	Instante $k$ que o <i>cluster</i> $i$ foi gerado
$w_k$	Observação de uma componente ruidosa ou ruído branco no instante $k$ .
$W$	Quantidade de amostras da série temporal para o intervalo inicial do AESR.
$\bar{y}$	Média amostral da série temporal.
$y_k$	Observação de uma série temporal discreta no instante $k$ .
$y_t$	Observação de uma série temporal contínua no instante $t$ .
$\tilde{y}_k$	Previsão da série temporal realizada pela RNF-TS no instante $k$ .
$z_k$	Componente não-observável nível no instante $k$ .
$\alpha$	Coefficiente usado de forma genérica.
$\beta$	Coefficiente usado de forma genérica.
$\gamma_y(h)$	Função de autocovariância com avanço $h$ .
$\hat{\gamma}_y(h)$	Função de autocovariância amostral com avanço $h$ .



$\zeta$	Constante de aprendizado para o cálculo do raio dos <i>clusters</i> .
$\theta_k^q$	Variáveis da etapa de SVD do método AESR, para $q = 1, \dots, d$ .
$\theta_k^I$	Variáveis da etapa de agrupamento do método AESR, para $I$ um grupo de índices.
$\lambda_k^i$	Grau de ativação normalizado do <i>cluster</i> $i$ no instante $k$ .
$\xi^2$	Distância de uma amostra dos dados para o ponto focal de um <i>cluster</i> .
$\mu_i^j$	Função de pertinência gaussiana no eixo $j$ e para o <i>cluster</i> $i$ .
$\bar{\mu}$	Média ou valor esperado da série temporal.
$\rho_y(h)$	Função de autocorrelação com avanço $h$ .
$\hat{\rho}_y(h)$	Função de autocorrelação amostral com avanço $h$ .
$\rho^q$	Autovalores da matriz de covariância $\mathbf{S}$ do método AES, para $q = 1, \dots, d$ .
$\rho_k^q$	Autovalores da matriz de covariância $\mathbf{S}_k$ do método AESR, para $q = 1, \dots, d$ .
$\sigma_y^2$	Variância da série temporal.
$\sigma_i^j(k)$	Raio ou zona de influência do <i>cluster</i> $i$ , no eixo $j$ e no instante $k$ .
$\tau_k^i$	Grau de ativação do <i>cluster</i> $i$ no instante $k$ .
$\Phi$	Intervalo inicial com $W$ amostras da série temporal do AESR.
$\Omega_k^i$	Utilidade do <i>cluster</i> $i$ no instante $k$ .
$\bar{\Omega}_k$	Média das utilidades dos <i>clusters</i> no instante $k$ .
$\tilde{\Omega}_k$	Desvio padrão das utilidades dos <i>clusters</i> no instante $k$ .

### **Vetores:**

$\mathbf{a}_k$	Ruído branco multivariável no instante $k$ descorrelacionado com $\mathbf{w}_k$ .
$\mathbf{d}_k$	Vetor de dados no instante $k$ .
$\hat{\mathbf{d}}_k$	Vetor de dados padronizado no instante $k$ .
$\bar{\mathbf{d}}_k$	Vetor média dos dados de $\mathbf{d}_k$ .
$\mathbf{f}_{i*}$	Ponto focal de um <i>cluster</i> $i$ .

$\mathbf{m}_q$	Vetor com amostras da série temporal do método AES e AESR.
$\mathbf{p}^q$	Autovetores da matriz de covariância $\mathbf{S}$ do método AES, para $q = 1, \dots, d$ .
$\mathbf{p}_k^q$	Autovetores da matriz de covariância $\mathbf{S}_k$ do método AESR, para $q = 1, \dots, d$ .
$\mathbf{q}_k$	Vetor de atualização do AESR no instante $k$ .
$\hat{\mathbf{u}}_{k-s}$	Vetor de componentes não-observáveis do último período sazonal padronizado.
$\mathbf{w}_k$	Ruído branco multivariável no instante $k$ .
$\hat{\mathbf{x}}_k$	Vetor de estados padronizado para treinamento da RNF-TS.
$\hat{\mathbf{x}}'_k$	Vetor de estados global padronizado gerado como saída da RNF-TS.
$\hat{\mathbf{x}}_k^i$	Vetor de estados padronizado gerado como saída de cada submodelo $i$ da RNF-TS.
$\tilde{\mathbf{x}}'_k$	Vetor de estados global gerado como saída da RNF-TS na escala original.
$\mathbf{y}_k$	Série temporal multivariável no instante $k$ .
$\boldsymbol{\delta}_k$	Desvio padrão dos elementos de $\mathbf{d}_k$ .

### Matrizes:

$\mathbf{A}_k$	Matriz de estado.
$\mathbf{A}_k^i$	Matriz de estado do consequente da regra <i>fuzzy</i> $i$ da RNF-TS.
$\mathbf{B}_k^i$	Matriz de entrada do consequente da regra <i>fuzzy</i> $i$ da RNF-TS.
$\mathbf{C}_k$	Matriz de observação.
$\mathbf{G}_k^i$	Matriz de ganho do L-FWRLS da regra <i>fuzzy</i> $i$ .
$\mathbf{M}$	Matriz trajetória do AES e AESR.
$\mathbf{Q}_k$	Matriz de atualização do AESR no instante $k$ .
$\mathbf{P}_k^i$	Matriz de covariância do L-FWRLS da regra <i>fuzzy</i> $i$ .
$\mathbf{S}$	Matriz de covariância do AES.
$\mathbf{S}_k$	Matriz de covariância do AESR no instante $k$ .
$\boldsymbol{\Delta}^i(\mathbf{k})$	Matriz de elementos de $\mathbf{A}_k^i$ e $\mathbf{B}_k^i$ do L-FWRLS da regra <i>fuzzy</i> $i$ .
$\boldsymbol{\Sigma}$	Matriz de covariância entre a amostra dos dados e o ponto focal do <i>cluster</i> .

## SUMÁRIO

<b>1</b>	<b>CONSIDERAÇÕES INICIAIS</b> . . . . .	<b>29</b>
<b>1.1</b>	<b>Introdução</b> . . . . .	<b>29</b>
<b>1.2</b>	<b>Objetivos da Pesquisa</b> . . . . .	<b>32</b>
1.2.1	Objetivo Geral . . . . .	32
1.2.2	Objetivos Específicos . . . . .	32
<b>1.3</b>	<b>Descrição da Metodologia e Principais Contribuições</b> . . . . .	<b>33</b>
<b>1.4</b>	<b>Trabalhos Publicados</b> . . . . .	<b>34</b>
<b>1.5</b>	<b>Organização da Dissertação</b> . . . . .	<b>35</b>
<b>2</b>	<b>SÉRIES TEMPORAIS</b> . . . . .	<b>37</b>
<b>2.1</b>	<b>Definição de Séries Temporais</b> . . . . .	<b>37</b>
<b>2.2</b>	<b>Classificação de Séries Temporais</b> . . . . .	<b>38</b>
2.2.1	Estacionariedade . . . . .	38
2.2.2	Linearidade . . . . .	41
2.2.3	Sazonalidade . . . . .	43
2.2.4	Séries Temporais Univariáveis e Multivariáveis . . . . .	44
2.2.5	Séries Temporais Caóticas . . . . .	45
<b>2.3</b>	<b>Aspectos sobre Modelagem Matemática de Séries Temporais</b> . . . . .	<b>45</b>
2.3.1	Definição de Modelo e Modelagem . . . . .	45
2.3.2	Categorias da Modelagem Matemática . . . . .	46
2.3.3	Procedimento da Modelagem Matemática . . . . .	46
2.3.4	Modelos Regressivos . . . . .	47
2.3.4.1	<i>Modelos AR</i> . . . . .	48
2.3.4.2	<i>Modelos MA</i> . . . . .	48
2.3.4.3	<i>Modelos ARMA</i> . . . . .	48
2.3.4.4	<i>Modelos ARIMA</i> . . . . .	49
2.3.5	Modelos no Domínio do Tempo . . . . .	49
2.3.5.1	<i>Modelos de Função de Transferência</i> . . . . .	49
2.3.5.2	<i>Modelos no Espaço de Estados</i> . . . . .	50
2.3.6	Modelos no Domínio da Frequência . . . . .	51
2.3.7	Modelos Não-Lineares . . . . .	51
<b>2.4</b>	<b>Aspectos sobre Previsão de Séries Temporais</b> . . . . .	<b>52</b>
2.4.1	Descrição e Objetivos da Previsão Qualitativa . . . . .	53
2.4.2	Procedimento Geral para Previsão Quantitativa . . . . .	54
<b>2.5</b>	<b>Aspectos sobre Decomposição de Séries Temporais</b> . . . . .	<b>56</b>
2.5.1	Métodos de Suavização Exponencial . . . . .	56

2.5.2	Análise Espectral Singular - AES . . . . .	58
2.5.2.1	<i>Etapa de Incorporação</i> . . . . .	59
2.5.2.2	<i>Etapa de Decomposição em Valores Singulares</i> . . . . .	60
2.5.2.3	<i>Etapa de Agrupamento</i> . . . . .	60
2.5.2.4	<i>Etapa de Média Diagonal</i> . . . . .	61
<b>3</b>	<b>SISTEMAS EVOLUTIVOS</b> . . . . .	<b>63</b>
<b>3.1</b>	<b>Definição de Sistemas Evolutivos</b> . . . . .	<b>63</b>
<b>3.2</b>	<b>Procedimentos Funcionais dos Sistemas Evolutivos</b> . . . . .	<b>66</b>
3.2.1	Particionamento do Espaço de Dados por Agrupamento Evolutivo . . . . .	66
3.2.2	Padronização e Normalização dos Dados . . . . .	71
3.2.3	Monitoramento da Qualidade dos <i>Clusters</i> . . . . .	73
3.2.4	Estimação dos Parâmetros dos Submodelos . . . . .	74
<b>3.3</b>	<b>Aspectos do Estado da Arte para Sistemas Evolutivos</b> . . . . .	<b>76</b>
<b>4</b>	<b>METODOLOGIA EVOLUTIVA PROPOSTA PARA PREVISÃO DE SÉRIES TEMPORAIS</b> . . . . .	<b>79</b>
<b>4.1</b>	<b>Descrição Geral da Metodologia Proposta</b> . . . . .	<b>79</b>
<b>4.2</b>	<b>Estágio de Treinamento</b> . . . . .	<b>81</b>
4.2.1	Camada de Decomposição da Série Temporal pelo Método AESR . . . . .	81
4.2.1.1	<i>Condições Iniciais do Algoritmo AESR</i> . . . . .	82
4.2.1.2	<i>Atualização Recursiva da Matriz de Covariância</i> . . . . .	84
4.2.1.3	<i>Decomposição em Valores Singulares - SVD</i> . . . . .	84
4.2.1.4	<i>Agrupamento do AESR</i> . . . . .	85
4.2.1.5	<i>Determinação das Componentes Não-observáveis</i> . . . . .	85
4.2.1.6	<i>Algoritmos da Camada de Decomposição da Série Temporal</i> . . . . .	86
4.2.2	Camada de Agrupamento Evolutivo . . . . .	86
4.2.2.1	<i>Algoritmo da Camada de Agrupamento Evolutivo</i> . . . . .	90
4.2.3	Camada de Agregação . . . . .	90
4.2.3.1	<i>Algoritmo da Camada de Agregação</i> . . . . .	91
4.2.4	Camada de Estimação das Matrizes de Estado . . . . .	91
4.2.4.1	<i>Algoritmo da Camada de Estimação das Matrizes de Estado</i> . . . . .	92
4.2.5	Camada de Inferência da RNF-TS . . . . .	92
4.2.5.1	<i>Algoritmo da Camada de Inferência</i> . . . . .	93
4.2.6	Camada de Reconstrução . . . . .	93
4.2.6.1	<i>Algoritmo da Camada de Reconstrução</i> . . . . .	94
4.2.7	Remoção de <i>Clusters</i> com Baixa Qualidade . . . . .	94
4.2.8	Algoritmo do Estágio de Treinamento . . . . .	95
<b>4.3</b>	<b>Estágio de Previsão</b> . . . . .	<b>95</b>

<b>5</b>	<b>RESULTADOS EXPERIMENTAIS . . . . .</b>	<b>97</b>
<b>5.1</b>	<b>Considerações Iniciais sobre os Experimentos . . . . .</b>	<b>97</b>
<b>5.2</b>	<b>Experimento 1 - Desempenho da RNF-TS para Horizontes de Previsão Distintos . . . . .</b>	<b>97</b>
<b>5.3</b>	<b>Experimento 2 - Desempenho da RNF-TS nos Estágios de Treinamento e Previsão . . . . .</b>	<b>101</b>
<b>5.4</b>	<b>Avaliação das Características da RNF-TS para a Série Temporal de Ven- das de Papel . . . . .</b>	<b>105</b>
<b>5.5</b>	<b>Estudo de Caso: Detecção de Anomalias Baseada em Dados de Eletro- cardiogramas . . . . .</b>	<b>111</b>
<b>6</b>	<b>CONSIDERAÇÕES FINAIS . . . . .</b>	<b>115</b>
<b>6.1</b>	<b>Conclusões . . . . .</b>	<b>115</b>
<b>6.2</b>	<b>Trabalhos Futuros . . . . .</b>	<b>116</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>117</b>
	<b>APÊNDICE A – FUNDAMENTOS SOBRE SISTEMAS FUZZY . . . . .</b>	<b>125</b>
<b>A.1</b>	<b>Conjuntos <i>Fuzzy</i> . . . . .</b>	<b>125</b>
<b>A.2</b>	<b>Operações com Conjuntos <i>Fuzzy</i> . . . . .</b>	<b>126</b>
A.2.1	Complemento <i>Fuzzy</i> . . . . .	126
A.2.2	União <i>Fuzzy</i> . . . . .	127
A.2.3	Intersecção <i>Fuzzy</i> . . . . .	127
<b>A.3</b>	<b>Variáveis Linguísticas e Regras <i>Fuzzy</i> . . . . .</b>	<b>128</b>
<b>A.4</b>	<b>Sistemas <i>Fuzzy</i> Takagi-Sugeno . . . . .</b>	<b>129</b>
	<b>APÊNDICE B – PRINCÍPIOS DA TEORIA DA PROBABILIDADE . . . . .</b>	<b>131</b>



# 1 CONSIDERAÇÕES INICIAIS

## 1.1 Introdução

Uma série temporal pode ser definida como qualquer conjunto de dados que estão ordenados no tempo. Devido a essa relevância sequencial dos dados, as observações vizinhas são geralmente dependentes entre si; logo, o objetivo principal dos analistas de séries temporais é analisar e modelar esta dependência. As séries temporais existem nos mais diversos campos do conhecimento, como economia, medicina, engenharia, física e geografia. Ao trabalharem com dados com essas características, os analistas de séries temporais podem estar interessados em diversas finalidades, entre elas estão a previsão de valores futuros para a série e a busca por padrões em seus dados (BROCKWELL; DAVIS, 2002) (HAMILTON, 1994) (YOUNG, 2014).

A previsão de séries temporais é um tema relevante não apenas para a comunidade científica em diversas áreas, como também para organizações privadas e governamentais, devido a aplicação de seus resultados no processo de tomada de decisão por parte dos especialistas. Dessa forma, a previsão permite que sejam realizadas ações preventivas, possibilitando um planejamento de estratégias antecipadas. O principal objetivo das técnicas de previsão de séries temporais é identificar padrões no conjunto de dados tal que um modelo matemático seja desenvolvido para reconstruir os futuros padrões temporais (BROCKWELL; DAVIS, 2002) (MONTGONOMERY; JENNINGS; KULAHCI, 2015).

Além da previsão, os especialistas podem também estar interessados na identificação de padrões e periodicidades relevantes nos dados. Esses padrões que geralmente são ocultos aos olhos dos analistas são chamados de componentes não-observáveis. As componentes apresentam características importantes que ajudam a entender o comportamento geral da série, podendo ser muito úteis para a construção de modelos matemáticos e para auxiliar nos resultados da previsão, pois a extração dessas componentes e a remoção de ruído em uma fase de pré-processamento podem melhorar bastante o desempenho da previsão (ABDOLLAHZADE et al., 2015). Existem diversos métodos que têm como finalidade extrair essas componentes não-observáveis, sendo que os tipos de padrões e a quantidade deles dependem da técnica aplicada para esse fim. Exemplos de tais componentes são a tendência, a sazonalidade, o padrão cíclico, o nível, os impulsos, entre outros tipos de comportamentos (HARVEY; KOOPMAN, 2009). Em particular, essa pesquisa se interessa por séries temporais que apresentam comportamento sazonal, ou seja, padrões que ocorrem regularmente em períodos de tempo fixos, podendo ser semanais, mensais ou anuais, por exemplo.

Atualmente, há um grande volume de dados brutos em diversos formatos e uma elevada velocidade em que eles são gerados. Dessa forma, há uma grande necessidade de processar uma quantidade crescente de informações de forma eficiente para a extração de conhecimentos

úteis à modelagem de sistemas complexos. Além disso, as informações geralmente são bastante complexas, não-lineares e não-estacionárias. Logo, pesquisadores são desafiados a desenvolver algoritmos que possam lidar com grande quantidade de informações, trabalhar com amostras instantâneas dos dados de maneira *online* e em tempo real, adaptar-se às mudanças do ambiente, ser computacionalmente eficiente e preservar a transparência do modelo (ANGELOV, 2013). Esses desafios também estão inseridos no domínio das séries temporais.

Visando atender a esses requisitos atuais quando se trabalha com dados, inclusive séries temporais, recentemente surgiu o paradigma de sistemas evolutivos. Esse paradigma é baseado no conceito de modificação da estrutura do sistema, ou seja, o sistema é capaz de se adaptar às mudanças do ambiente modificando sua arquitetura interna e ajustando seus parâmetros. Para operar dessa maneira, os sistemas evolutivos têm interesse em algoritmos que trabalham recursivamente. Além de evoluir sua estrutura, esses sistemas são capazes de extrair conhecimento a partir de amostras de dados a cada instante, ou seja, eles monitoram o ambiente desconhecido a todo momento. Geralmente, esses sistemas são baseados em sistemas puramente *fuzzy*, redes neurais, redes *neuro-fuzzy* e árvores de decisão. São usados também conceitos de aprendizagem de máquinas e reconhecimento de padrões como ferramentas de treinamento para o modelo selecionado (ANGELOV, 2013) (LUGHOFER, 2011) (KASABOV, 2007).

Tradicionalmente, os modelos regressivos como os ARIMA (*Autoregressive Integrated Moving Average*, Modelo Autorregressivo Integrado de Médias Móveis) formam uma categoria muito utilizada para trabalhar com séries temporais. No entanto, esses métodos exigem muitas restrições para serem adotados, como por exemplo, a série precisa ser estacionária e linear, entre outras exigências que limitam sua aplicação em dados reais (BROCKWELL; DAVIS, 2002). No mundo real, as séries temporais são não-estacionárias, apresentam incertezas em seu comportamento e muitas possuem características sazonais e tendenciosas (HAMILTON, 1994). Devido a essas complexidades, metodologias de previsão de séries temporais baseadas em inteligência computacional têm sido amplamente consideradas pela comunidade científica, por exemplo, redes neurais, sistemas *fuzzy* e redes *neuro-fuzzy* (GAN et al., 2014).

O desenvolvimento das redes neurais artificiais se fundamentou na estrutura biológica do cérebro de seres vivos, sendo aplicadas no processamento de informação, aprendizado e aproximação de funções. Entre os principais tipos de redes neurais, pode-se citar as redes neurais com função de base radial, as redes neurais recorrentes e as redes neurais com múltiplas camadas (KOCADAGLI; ASIKGIL, 2014). Muitas pesquisas usam redes neurais para trabalharem com séries temporais, por exemplo, em Leung (2001), uma rede neural com função de base radial foi desenvolvida para previsão de séries temporais caóticas e ruidosas. Em Cui, Shi e Wang (2015), os autores propõem um modelo inovador denominado de Rede Neural Dinâmica Quântica de Rotação Complexa para previsão de séries temporais. Por sua vez, Yan (2012) propôs uma rede neural de regressão generalizada para modelagem efetiva de uma série temporal na área de economia.



Outra técnica que vem sendo bastante utilizada para modelar séries temporais é a rede *neuro-fuzzy*, que une as aptidões de aprendizado e de processamento paralelo das redes neurais e a capacidade de representação do conhecimento por parte dos sistemas *fuzzy* (ANGELOV, 2013). As redes *neuro-fuzzy* apresentam alto desempenho e são adequadas para identificação e previsão de processos não-lineares complexos, que é o caso de séries temporais reais, de acordo com a pesquisa de Zhang, Chung e Wai-Lun (2008). Em Hilas (2012), um sistema *neuro-fuzzy* Takagi-Sugeno modificado é apresentado, sendo que suas proposições consequentes das regras *fuzzy* são redes neurais com recorrência interna que representam a dinâmica do sistema global. Essa metodologia é aplicada para séries temporais no domínio das telecomunicações.

As redes *neuro-fuzzy* com submodelos lineares locais correspondem a uma classe especial de redes *neuro-fuzzy* que têm sido muito utilizada nos últimos anos para previsão de séries temporais. Essa categoria de rede *neuro-fuzzy* se baseia na estratégia dividir para conquistar, em que um problema complexo pode ser dividido em diversos problemas mais simples de lidar (ANGELOV, 2013). A contribuição de cada submodelo linear para a saída global do sistema é indicada por uma ponderação denominada grau de ativação de cada regra *fuzzy*. Em outras palavras, cada regra *fuzzy* possui um submodelo associado. Essa estratégia tem apresentado resultados promissores. Por exemplo, os autores Abdollahzade et al. (2010) empregaram uma rede *neuro-fuzzy* com submodelos lineares locais para efetuar a previsão do preço da eletricidade em diversos locais. Há uma discussão aprofundada sobre esse princípio em Angelov (2013). A metodologia proposta nessa dissertação também segue esse princípio do dividir para conquistar em combinação com uma estrutura *neuro-fuzzy*.

Vem sendo uma tendência atual utilizar métodos híbridos que abordam várias técnicas de inteligência computacional, conforme comentado em Palit e Popovic (2005). Em Abdollahzade et al. (2015), um método híbrido para previsão de séries temporais não-lineares e caóticas baseado em um modelo *neuro-fuzzy* linear local e no método de Análise Espectral Singular otimizado pelo algoritmo de enxame de partículas é desenvolvido. Esse método opera de forma *offline* para efetuar a previsão. Um sistema evolucionário híbrido composto por filtro de suavização exponencial simples, modelos lineares ARIMA e modelo de Máquina de Vetores de Suporte (SVM, do inglês *Support Vector Machine*) é aplicado também para prever séries temporais em Oliveira e Ludermir (2016).

Algumas metodologias recentes foram desenvolvidas voltadas para as séries temporais sazonais. Em Stepnicka et al. (2013), os autores discutem sobre diferentes métodos de inteligência computacional e sobre o potencial de suas combinações para prever séries temporais sazonais múltiplos passos a frente, como por exemplo, redes neurais artificiais evolucionárias, máquinas de vetores de suporte e regras *fuzzy*. Já em Gan et al. (2014), um modelo autorregressivo quase-linear, que pertence a classe de modelos com coeficientes variantes determinados por redes neurais radiais, é aplicado para previsão de séries temporais sazonais.

Várias pesquisas envolvendo previsão de séries temporais estão considerando concei-

tos de aprendizagem de máquina em suas propostas, dentre eles, destaca-se o método não-supervisionado conhecido como agrupamento, que também segue o princípio do dividir para conquistar. Um modelo baseado no método de agrupamento *fuzzy c-means* e granulação que não exigem que os dados tenham a mesma dimensionalidade é proposto em Wang, Pedrycz e Liu (2015) para previsão a longo prazo de séries temporais. Em Wu e Lee (2015), uma estratégia de modelagem local e a investigação da sua eficácia para três abordagens de aprendizado populares, que são uma rede neural, uma rede *neuro-fuzzy* adaptativa e uma máquina de vetores de suporte de mínimos quadrados, são analisadas e aplicadas para a previsão de séries temporais. Uma técnica de aprendizado extremo melhorada baseada no agrupamento para previsão sequencial *online* de séries temporais multivariadas é apresentada em Wang e Han (2015).

Existem também pesquisas direcionadas à área de séries temporais utilizando modelos evolutivos. Em Birek, Petrovic e Boylan (2014), um algoritmo baseado em um sistema *fuzzy* Takagi-Sugeno evolutivo é aplicado a um problema de previsão de vazamento de líquidos, cujos dados estão em formato de séries temporais. Um preditor baseado em uma rede *neuro-fuzzy* evolutiva é proposto em Li, Wang e Ismail (2014) para extrair informações representativas de conjuntos de dados multidimensionais para uma previsão mais precisa dos estados de um sistema. Por sua vez, Komijani et al. (2012) apresentam o algoritmo eTS-LS-SVM (*evolving Takagi-Sugeno Least Square Support Vector Machine*) aplicado para previsão de séries temporais. A principal contribuição desse trabalho é que ele aborda modelos locais não-lineares baseados em máquinas de vetores de suporte por mínimos quadrados como proposição consequente das regras *fuzzy*.

A metodologia proposta nessa dissertação investiga a previsão de séries temporais sazonais de forma evolutiva, contudo, diferenciando-se das demais por utilizar os dados das componentes não-observáveis extraídas da série para adaptar a estrutura da rede *neuro-fuzzy* e prever separadamente cada componente, finalizando com a obtenção da previsão da série original.

## 1.2 Objetivos da Pesquisa

### 1.2.1 Objetivo Geral

Desenvolver uma metodologia evolutiva baseada em uma rede *neuro-fuzzy* Takagi-Sugeno para previsão de observações múltiplos passos a frente e para identificação de componentes não-observáveis de séries temporais sazonais.

### 1.2.2 Objetivos Específicos

Os seguintes objetivos específicos são considerados nessa dissertação:

- Projetar e implementar uma metodologia de previsão que seja adaptativa, evolutiva e que

trabalha de forma *online*;

- Desenvolver uma versão recursiva do método de Análise Espectral Singular que extraia as componentes não-observáveis visando integrá-la a estrutura da rede *neuro-fuzzy*;
- Representar as componentes não-observáveis como estados, já que as proposições consequentes das regras *fuzzy* são compostas por modelos no espaço de estados;
- Comparar a metodologia proposta com os resultados de outras pesquisas científicas;
- Aplicar em um estudo de caso envolvendo dados de eletrocardiogramas.

### 1.3 Descrição da Metodologia e Principais Contribuições

A rede *neuro-fuzzy* Takagi-Sugeno apresenta características evolutiva, adaptativa e *online*, sendo aplicada para realizar a previsão e decomposição dos padrões de séries temporais sazonais. Essa rede *neuro-fuzzy* tem característica evolutiva porque é capaz de expandir ou simplificar sua estrutura, isto é, aumentar ou diminuir a quantidade de regras *fuzzy* para se ajustar ao comportamento dos dados. Já a característica adaptativa da rede *neuro-fuzzy* está relacionada ao ajuste de parâmetros de suas equações matemáticas relacionadas. De acordo com Angelov (2013), todo sistema evolutivo é adaptativo, contudo não são todos os sistemas adaptativos que são evolutivos. Além disso, a rede *neuro-fuzzy* é *online* porque considera os dados chegando a todo instante, amostra por amostra, para realizar a previsão e ajustar sua estrutura.

Essa rede *neuro-fuzzy* segue o princípio do dividir para conquistar que vem amplamente sendo considerado em vários trabalhos, isto é, subdivide-se um problema complexo em vários subproblemas mais simples. À vista disso, essa metodologia decompõe a série temporal complexa em várias componentes não-observáveis que são mais simples de lidar.

Essas componentes são identificadas por uma abordagem também proposta nessa pesquisa baseada no método de Análise Espectral Singular (AES, do inglês *Singular Spectrum Analysis* - SSA). Essa nova abordagem foi nomeada de *Recursive SSA* (RSSA) ou AES Recursiva (AESR). Ela foi desenvolvida porque originalmente o método AES é executado em batelada, considerando todo o conjunto de dados da série temporal. Para que a extração das componentes e a previsão seja executada recursivamente, essa versão alternativa foi proposta.

A rede *neuro-fuzzy* usa esses dados das componentes para adaptar e evoluir sua estrutura de regras *fuzzy* (adicionando ou removendo) para prever futuras observações da série temporal. A previsão é primeiramente realizada para cada componente não-observável separadamente. Por conseguinte, os resultados da previsão de cada componente não-observável são reunidos para compor a previsão da série temporal original.

A proposição consequente das regras *fuzzy* são modelos no espaço de estados, cujos estados são formados pelas próprias componentes extraídas. As componentes não-observáveis

extraídas pelo método AESR são linearmente independentes, portanto, podem ser consideradas como estados. Essa característica dos consequentes das regras *fuzzy* tratados como modelos no espaço de estados é uma importante contribuição dessa dissertação. Esse tipo de modelo foi utilizado porque eles representam bem o comportamento dinâmico das componentes não-observáveis e é essencial para obter o comportamento das componentes mesmo quando não há dados disponíveis da série temporal para aplicar o método AESR. Cada regra *fuzzy* possui sua própria equação de estado relacionada.

Sendo assim, as principais contribuições desse trabalho são listadas a seguir:

- Metodologia de previsão evolutiva de séries temporais sazonais múltiplos passos a frente;
- Uso dos dados das componentes não-observáveis para previsão da série temporal;
- Previsão de cada componente não-observável separadamente e, em seguida, de dados da série temporal original;
- As proposições consequentes das regras *fuzzy* são modelos no espaço de estados;
- Proposta do método AESR;
- Inclusão de uma nova condição para avaliação de qualidade dos *clusters* (Condição *C* na Equação 4.33);
- Comparação com outras metodologias para previsão de séries temporais sazonais.

#### 1.4 Trabalhos Publicados

As publicações produzidas no decorrer desta pesquisa são apresentadas nessa seção. Dois artigos em congressos foram publicados e, além disso, um artigo para periódico já foi submetido e está em processo de análise por parte dos avaliadores, conforme segue:

- RODRIGUES JÚNIOR, S. E.; SERRA, G. L. O. *An Evolving Algorithm Based on Unobservable Components Neuro-Fuzzy Model For Time Series Forecasting*. IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS 2016). Natal, Brazil.
- RODRIGUES JÚNIOR, S. E.; SERRA, G. L. O. *Modelagem Neuro-Fuzzy Evolutiva no Espaço de Estados Baseada em Componentes Não-Observáveis para Previsão de Séries Temporais Sazonais*. XXI Congresso Brasileiro de Automática (CBA 2016). Vitória, Brasil.
- RODRIGUES JÚNIOR, S. E.; SERRA, G. L. O. *A Novel Intelligent Approach for State Space Evolving Forecasting of Seasonal Time Series*. Engineering Applications of Artificial Intelligence (Submetido), 2017.

## 1.5 Organização da Dissertação

A organização dessa dissertação será descrita logo a seguir. O Capítulo 1 apresenta as considerações iniciais sobre a pesquisa, as motivações, os objetivos, as justificativas, as contribuições e os trabalhos publicados.

O Capítulo 2 apresenta os conceitos fundamentais sobre séries temporais, como a definição e a classificação. Em seguida, esse capítulo comenta alguns aspectos sobre modelos tradicionais utilizados para representar séries. Questões e métodos para a previsão de séries temporais e para decomposição de suas componentes não-observáveis, ambos objetivos da metodologia proposta, são descritos também.

O terceiro capítulo comenta sobre os principais fundamentos da teoria de sistemas evolutivos. Por conseguinte, o estado-da-arte dessa área do conhecimento é descrita, destacando-se a evolução desse campo de pesquisa que continua com muitas questões em aberto. Por fim, o princípio de funcionamento dos sistemas evolutivos é apresentado.

A metodologia evolutiva para previsão de séries temporais é descrita no Capítulo 4. Em primeiro lugar, comentários gerais são apresentados sobre a estrutura da rede *neuro-fuzzy* evolutiva em camadas e sobre a sua operação. As seções seguintes descrevem a execução e os algoritmos referentes a cada camada e a cada estágio de operação.

Os resultados experimentais para avaliação do desempenho da metodologia proposta são apresentados no Capítulo 5. São realizados dois experimentos, cujos resultados são comparados com outras pesquisas e com métodos recentes e tradicionais. Outros aspectos sobre as características principais da metodologia proposta são detalhados também. Por fim, esse capítulo apresenta um estudo de caso da metodologia proposta envolvendo a detecção em tempo-real de anomalias baseada em dados de eletrocardiogramas de um paciente.

O Capítulo 6 traz as considerações finais, discussões sobre as principais contribuições e sugestões para trabalhos futuros.

Além desses capítulos, a dissertação também possui alguns apêndices que dão suporte teórico ao leitor. O Apêndice A exibe alguns fundamentos necessários sobre sistemas *fuzzy*, focando na categoria Takagi-Sugeno; enquanto que o Apêndice B comenta alguns aspectos básicos sobre a teoria da probabilidade.



## 2 SÉRIES TEMPORAIS

### 2.1 Definição de Séries Temporais

A análise de séries temporais apresenta vasta aplicação nas mais variadas áreas de conhecimento. Na economia, pode-se encontrar séries temporais relacionadas com o mercado de ações por exemplo, onde os investidores as analisam para planejar e avaliar suas estratégias. Outro exemplo diz respeito às séries meteorológicas que são consideradas principalmente para previsão de variáveis climáticas. Cientistas sociais estudam e avaliam séries temporais como taxas de natalidade e índices de desemprego. Na engenharia, séries temporais são bastante aplicadas no monitoramento industrial, detecção de falhas e na estimação de sinais diversos. Além disso, as séries podem ser utilizadas em técnicas de reconhecimento de padrões, de aprendizagem de máquinas e para mineração de dados (*data mining*) (PALIT; POPOVIC, 2005) (ABDOLLAHZADE et al., 2015).

Brockwell e Davis (2002) definem uma série temporal como um conjunto de observações  $y_t$  ordenadas no tempo. A ordem temporal dos dados da séries é muito importante e deve ser respeitada em qualquer tipo de análise. É possível gerar séries por meio de alguma função matemática, tal como:

$$y_t = \sin(2\pi t), \quad t \in [0, \infty), \quad (2.1)$$

sendo que esta apresenta um comportamento determinístico. Já uma série temporal que é descrita através de termos de distribuição de probabilidades e constituída por variáveis aleatórias é caracterizada como estocástica. Séries temporais podem ser classificadas também em contínuas ou discretas (BROCKWELL; DAVIS, 2002). A Equação 2.1 traz uma série contínua, pois suas observações são obtidas continuamente no intervalo de tempo  $t \in [0, \infty)$ . Por outro lado, séries temporais discretas são denotadas por  $\{y_1, y_2, \dots\}$ , em que  $y_t$  é a observação correspondente ao instante  $t = kT$ , para  $k = 1, 2, \dots$  e  $T$  o período de amostragem. Todas as séries temporais abordadas neste trabalho são discretas.

Ao analisarem uma série temporal, especialistas ou pesquisadores visam diferentes objetivos. Dentre os principais, pode-se destacar (BROCKWELL; DAVIS, 2002):

- A investigação do mecanismo gerador da série, identificando um modelo apropriado;
- A descrição do comportamento da série através de gráficos;
- A previsão de valores futuros para a série temporal, sendo a curto, a médio ou a longo prazo;

- A identificação de padrões relevantes ou componentes não-observáveis nos dados da série temporal.

A metodologia desenvolvida nessa pesquisa visa dois desses objetivos simultaneamente, realizando a previsão de dados para as séries temporais e a identificação de suas componentes não-observáveis.

Existem dois enfoques principais para a análise de séries temporais: a análise no domínio do tempo e a análise no domínio da frequência (PALIT; POPOVIC, 2005) (BROCKWELL; DAVIS, 2002). Segundo esses autores, a análise no domínio temporal é principalmente baseada no uso da função de covariância das séries temporais, descrevendo a relação entre as observações em diferentes instantes. Já o segundo enfoque analisa a frequência de eventos que ocorrem em determinados períodos de tempo, baseando-se na função de densidade espectral e na análise de Fourier. Ambas abordagens não são excludentes entre si, pelo contrário, são complementares na tarefa de explorar a natureza da série e construir um modelo apropriado.

## 2.2 Classificação de Séries Temporais

As séries temporais podem ser classificadas em:

- Estacionárias ou não-estacionárias;
- Lineares ou não-lineares;
- Sazonais ou não-sazonais;
- Univariáveis ou multivariáveis;
- Caóticas;
- Entre outras classificações.

Uma série temporal pode pertencer a uma ou mais dessas categorias apresentadas. A seguir, cada uma dessas propriedades são descritas.

### 2.2.1 Estacionariedade

Uma série temporal é dita estacionária se suas propriedades estatísticas do primeiro e segundo momento permanecem as mesmas ao longo de suas observações, ou seja, ela se desenvolve aleatoriamente em torno de uma média constante e com covariância independente



do tempo (BROCKWELL; DAVIS, 2002) (PALIT; POPOVIC, 2005) (HAMILTON, 1994). Em outros termos,  $y_k$  é estacionária no sentido amplo se:

$$\begin{aligned} \text{i) } E[y_k] &= \mu, \quad k = 1, 2, 3, \dots \text{ é independente de } k; \\ \text{ii) } \gamma_y(p, q) &= Cov(y_p, y_q) = E[(y_p - \mu)(y_q - \mu)] \text{ é independente de } k \text{ para todo } p \text{ e } q. \end{aligned} \quad (2.2)$$

sendo  $E[y_k]$  a esperança matemática que é igual a média  $\mu$ ,  $\gamma_y$  é a função de autocovariância da série temporal e os valores  $p$  e  $q$  são instantes específicos de  $k = 1, 2, \dots$ .

Sendo assim, a estacionariedade da série  $\{y_k, k = 1, 2, \dots\}$  indica que  $(y_1, y_2, \dots, y_n)$  e  $(y_{1+h}, y_{2+h}, \dots, y_{n+h})$  possuem a mesma distribuição de probabilidade para todo  $h$ . Entretanto, a maior parte das séries temporais que se encontra na prática apresenta alguma forma de não-estacionariedade, desobedecendo as condições apresentadas acima. De acordo com Aguirre (2015), o conceito de estacionariedade está fortemente relacionado ao de invariância temporal de modelos de sistemas, que significa a manutenção do comportamento dinâmico do sistema ao longo do tempo.

Para analisar se uma série temporal  $y_k$  é estacionária, as propriedades das funções de autocovariância e de autocorrelação são bastante úteis. Ambas fornecem diferentes medidas do grau de dependência entre os dados da série temporal. De forma específica, a função de autocovariância pode ser definida como uma medida do grau de interdependência numérica entre realizações de variáveis aleatórias (BROCKWELL; DAVIS, 2002). A função de autocovariância de  $y_k$  com avanço  $h$  é denotada por:

$$\gamma_y(h) = Cov(y_{k+h}, y_k), \quad (2.3)$$

sendo que para  $h = 0$ , essa equação retorna o próprio valor da variância que, por definição, é:

$$\sigma_y^2 = \gamma_y(0) = Cov(y_k, y_k). \quad (2.4)$$

Caso as realizações da série  $y_{k+h}, y_k$  forem totalmente independentes entre si, a função de autocovariância  $\gamma_y(h)$  será nula. Já a função de autocorrelação traz uma medida que indica o quanto o valor de uma realização da variável aleatória é influenciada por outras observações. Seu valor está normalizado entre  $-1$  e  $1$ . O valor  $1$  indica que há uma correlação perfeita entre as observações consideradas, sendo que o crescimento de uma acarreta no crescimento da outra. Enquanto que o valor  $-1$  expressa que existe uma correlação invertida perfeita, em que o crescimento de uma está relacionado com o decréscimo da outra. O valor nulo denota que não existe correlação entre ambos valores da série temporal. A função de autocorrelação de  $y_k$

com avanço  $h$  é calculada conforme a seguir (BROCKWELL; DAVIS, 2002):

$$\rho_y(h) = \frac{\gamma_y(h)}{\gamma_y(0)} = \text{Cor}(y_{k+h}, y_k). \quad (2.5)$$

Ambas relações precisam de um modelo disponível da série temporal. Porém, em aplicações práticas geralmente não há esse modelo disponível, apenas os dados da série. À vista disso, são realizadas estimativas dessas funções conhecidas como função de autocovariância amostral e função de autocorrelação amostral, respectivamente (BROCKWELL; DAVIS, 2002) (MONTGONOMERY; JENNINGS; KULAHCI, 2015). Além de auxiliar na identificação da estacionariedade, essas ferramentas são muito úteis ainda para seleção de modelos apropriados para séries temporais. Seja então  $y_1, y_2, \dots, y_n$  observações de uma série. Conforme Brockwell e Davis (2002), se a média amostral da série é representada por:

$$\bar{y} = \frac{1}{n} \sum_{k=1}^n y_k. \quad (2.6)$$

então, a função de autocovariância amostral é:

$$\hat{\gamma}_y(h) = \frac{1}{n} \sum_{k=1}^{n-|h|} (y_{k+|h|} - \bar{y})(y_k - \bar{y}), \quad -n < h < n. \quad (2.7)$$

e a função de autocorrelação amostral é calculada por:

$$\hat{\rho}_y(h) = \frac{\hat{\gamma}_y(h)}{\hat{\gamma}_y(0)}, \quad -n < h < n. \quad (2.8)$$

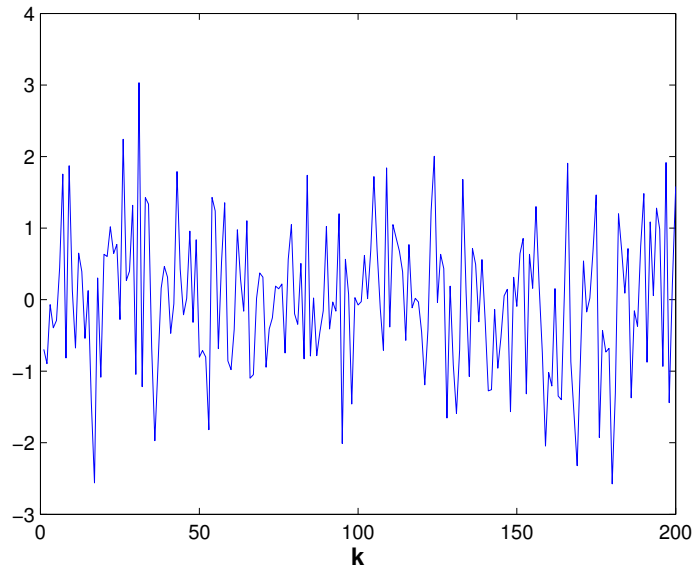
Como exemplo da aplicação desses conceitos, seja um ruído branco  $w_k$  (Figura 1), que é tipicamente uma série temporal estacionária, com distribuição de probabilidades normal de média 0 e variância igual a 1, ou seja,  $N \sim (0, 1)$ . Por definição, o ruído branco é composto por variáveis aleatórias independentes e não-correlacionadas quando consideradas em instantes diferentes. Em termos matemáticos, isso significa que o ruído branco apresenta as seguintes propriedades (BROCKWELL; DAVIS, 2002):

$$\gamma_w(h) = \begin{cases} \sigma_w^2, & \text{se } h = 0; \\ 0, & \text{se } h \neq 0; \end{cases} \quad (2.9)$$

$$\rho_w(h) = \begin{cases} 1, & \text{se } h = 0; \\ 0, & \text{se } h \neq 0. \end{cases}$$

Tendo em vista essas definições, a função de autocorrelação ideal do ruído branco, por exemplo, para  $h = 0, 1, \dots, 40$  é representada na Figura 2. Percebe-se que a correlação entre os dados do ruído branco é nula, exceto para  $h = 0$ . No entanto, um sinal ruído branco

Figura 1 – Ruído Branco.



Fonte: (BROCKWELL; DAVIS, 2002).

exatamente com essas características e com esse formato para a função de autocorrelação não é encontrado na prática. Dessa forma, aplica-se a função de autocorrelação amostral para obter uma aproximação do comportamento da Figura 2. A função de autocorrelação amostral calculada a partir dos dados disponíveis do ruído branco é exibida na Figura 3. Para indicar a estacionariedade, aproximadamente 95% dos valores de autocorrelação amostral devem se situar entre os limites de  $\pm 1.96/N$ , para  $N$  sendo a quantidade de amostras disponíveis da série temporal. Obedecendo esses limites, a função de autocorrelação amostral indicará o mesmo comportamento que a função de autocorrelação ideal (BROCKWELL; DAVIS, 2002).

### 2.2.2 Linearidade

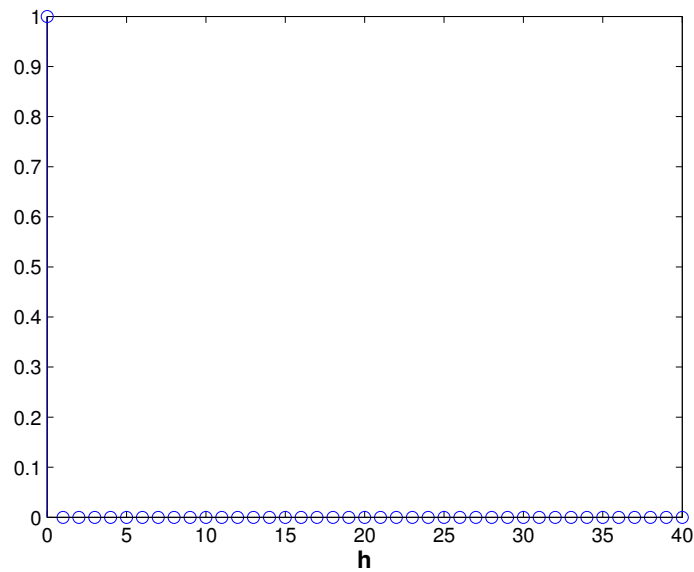
Conforme Palit e Popovic (2005), uma série temporal  $y_k$  é linear se ela possui a seguinte representação:

$$y_k = \sum_{j=-\infty}^{\infty} \beta_j w_{k-j}, \text{ para } \sum_{j=-\infty}^{\infty} |\beta_j| < \infty. \quad (2.10)$$

onde  $w_k$  é um ruído branco com média nula e variância unitária e  $\beta_j$  é uma sequência de constantes.

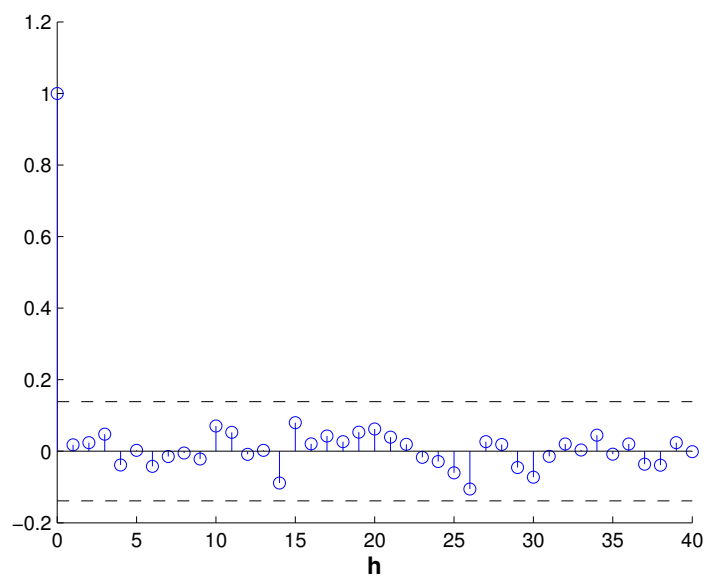
Se uma série é linear, então ela pode ser representada por uma função linear entre o valor presente e os valores passados. As séries temporais lineares podem ser geradas por processos estacionários de segunda ordem, que geralmente são lineares ou podem ser transformados em processos lineares usando a técnica de decomposição de Wold, descrita em Brockwell e Davis (2002). Exemplos de representações lineares muito aplicadas são os modelos AR, MA e ARMA,

Figura 2 – Função de Autocorrelação Ideal do Ruído Branco.



Fonte: Autor.

Figura 3 – Função de Autocorrelação Amostral do Ruído Branco.



Fonte: Autor.

que serão comentados na Seção 2.3.4. Entretanto, a maioria das séries temporais encontradas na prática são não-lineares, fazendo com que esses modelos acima não sejam adequados. Modelos para séries não-lineares serão comentados na Seção 2.3.7.

### 2.2.3 Sazonalidade

As séries temporais sazonais têm como principal característica exibir um comportamento periódico, podendo ser anual, mensal ou semanal, por exemplo. O período sazonal ( $s$ ), ou seja, a frequência de repetição do comportamento, é fixo e deve ser determinado pelo especialista. Esse padrão sazonal é encontrado em muitas séries temporais reais, sendo comum na economia, indústria e meteorologia. Uma série que possui sazonalidade como consequência também é não-estacionária (PALIT; POPOVIC, 2005).

Existem muitas abordagens para lidar com séries temporais sazonais, sendo que a mais tradicional é o procedimento de ajuste sazonal, que decompõe a série em diversas componentes ou padrões como tendência, sazonalidade, padrão cíclico e componentes ruidosas. Esse procedimento pode ser aditivo ou multiplicativo, dependendo da série temporal (YOUNG, 2014):

$$\begin{aligned} y_k &= w_k + s_k + tc_k \text{ (aditivo),} \\ y_k &= w_k \times s_k \times tc_k \text{ (multiplicativo),} \end{aligned} \tag{2.11}$$

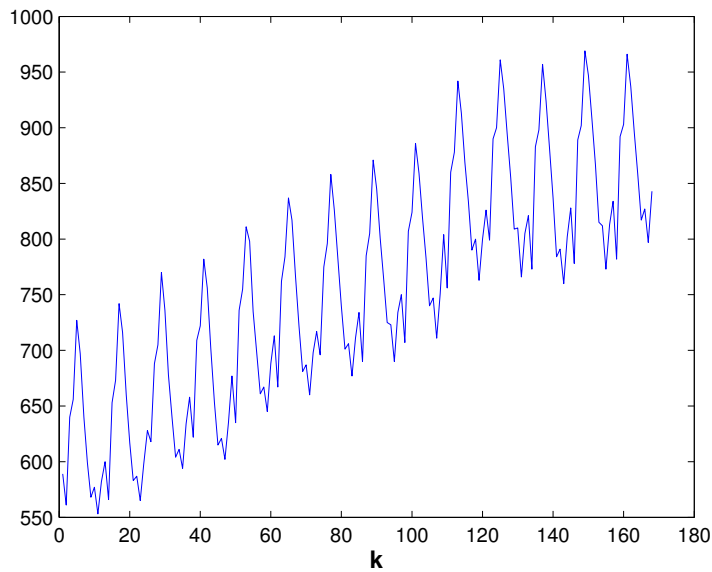
onde  $w_k$  é uma componente ruidosa,  $s_k$  representa a sazonalidade,  $tc_k$  é uma junção entre o padrão cíclico e a tendência, pois geralmente é conveniente fazer essa combinação na prática, segundo Young (2014). É importante não confundir o padrão sazonal com a componente cíclica. De forma semelhante ao sazonal, o padrão cíclico exibe um comportamento que aumenta e reduz a amplitude da série, entretanto, essas flutuações não ocorrem em um período fixo (HYNDMAN; ATHANASOPOULOS, 2014).

Outra abordagem muito conhecida e aplicada para trabalhar com séries sazonais é o modelo SARIMA, uma versão especial do ARIMA tradicional, comentado na Seção 2.3.4.4 (BROCKWELL; DAVIS, 2002) (PALIT; POPOVIC, 2005). Atualmente, pesquisas envolvendo séries sazonais vêm sendo direcionadas à área de inteligência computacional, principalmente baseadas em redes neurais e sistemas *fuzzy*. A metodologia de previsão desenvolvida nesse trabalho lida com séries temporais que apresentam comportamento sazonal.

Para comprovar a sazonalidade e verificar qual o período sazonal de uma série, o especialista pode analisar o comportamento da função de autocorrelação amostral a partir dos dados. Por exemplo, seja a série que possui dados mensais sobre a taxa de produção de leite em determinada região representada na Figura 4.

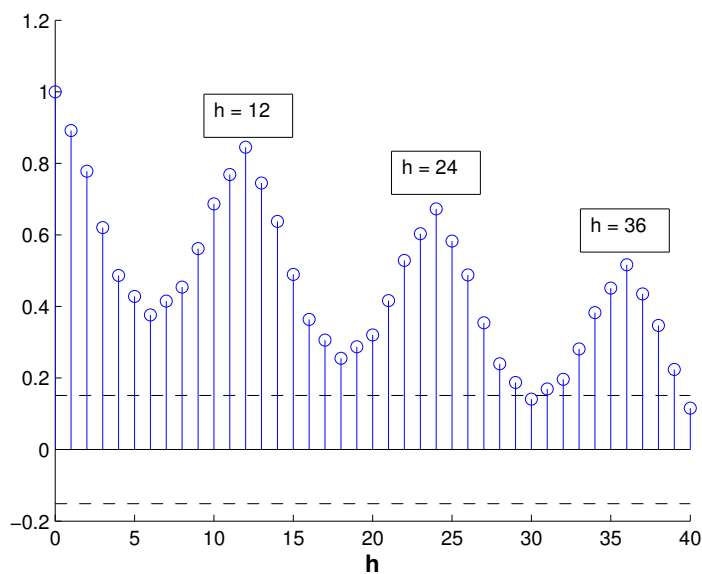
A sua função de autocorrelação amostral é exibida na Figura 5, na qual é possível observar o comportamento periódico a cada  $h = 12$  atrasos no tempo. Devido a esse comportamento, chega-se a conclusão que o período sazonal dessa série é  $s = 12$ . Percebe-se ainda que esta série não é estacionária, já que ultrapassa os limites representados pelas linhas tracejadas horizontais.

Figura 4 – Série Temporal da Taxa de Produção de Leite.



Fonte: (BROCKWELL; DAVIS, 2002).

Figura 5 – Função de Autocorrelação Amostral da Série de Taxa de Produção de Leite.



Fonte: Autor.

#### 2.2.4 Séries Temporais Univariáveis e Multivariáveis

Séries temporais univariáveis possuem dados que se referem a apenas uma variável física ou a somente um sinal dependente do tempo. As séries representadas nas Figuras 1 e 4 são exemplos dessa categoria de séries. Por outro lado, séries temporais multivariáveis são compostas por observações de dois ou mais processos simultâneos, ou seja, os dados desses processos são coletados no mesmo instante de tempo discreto. Logo, as observações em cada instante e para

cada um dos  $n$  processos simultâneos são representados por um vetor, conforme a seguir (PALIT; POPOVIC, 2005):

$$\mathbf{y}_k = [y_k^1 \ y_k^2 \ \cdots \ y_k^n]^T, \quad (2.12)$$

Na prática, as séries temporais complexas geralmente são multivariáveis, sendo que diversos processos ou séries apresentam dependências e relações entre si (PALIT; POPOVIC, 2005). A metodologia proposta considera a previsão apenas de séries temporais univariáveis, sendo que as multivariáveis serão tratadas em trabalhos futuros.

### 2.2.5 Séries Temporais Caóticas

As séries temporais caóticas são caracterizadas por observações que são não-periódicas, ou seja, são representadas por valores que se repetem aleatoriamente sem manter uma periodicidade definida. Outra característica importante desse tipo de série é sua grande sensibilidade à alteração de suas condições iniciais, sendo que uma variação pequena nessas condições faz com que o sistema se desenvolva de forma imprevisível (LIU, 2010).

O exemplo mais famoso de uma série temporal caótica é a Mackey-Glass, que é gerada usando a seguinte equação diferencial (ABDOLLAHZADE et al., 2015):

$$\frac{dy(t)}{dt} = \frac{\alpha y(t - \tau)}{1 + y^{10}(t - \tau)} + \beta y(t) \quad (2.13)$$

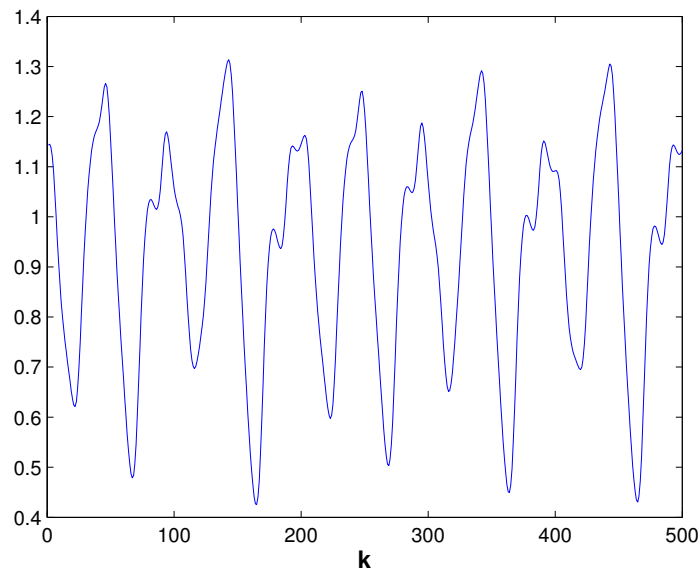
onde esse processo será caótico quando  $\tau > 16.8$ . Na Figura 6 é exibida a Mackey-Glass para as seguintes condições iniciais e parâmetros:  $y(0) = 1.2$ ,  $\alpha = 0.2$ ,  $\beta = 0.1$  e  $\tau = 17$ .

## 2.3 Aspectos sobre Modelagem Matemática de Séries Temporais

### 2.3.1 Definição de Modelo e Modelagem

Os termos modelo e modelagem podem ser interpretados de diferentes formas conforme a área do conhecimento, já que são conceitos multidisciplinares. Um modelo em geral pode ser definido como uma idealização simplificada de um sistema que possui maior complexidade, reproduzindo algumas de suas características de interesse (WALTER; PRONZATO, 1997). Aguirre (2015) cita que o próprio raciocínio do ser humano aplicado nas tarefas cotidianas se baseia em modelos mentais desenvolvidos ao longo do tempo. No entanto, essa pesquisa se restringe a modelos matemáticos. Dessa forma, a modelagem matemática é a área do conhecimento que estuda maneiras de construir e implementar modelos matemáticos de sistemas experimentais, modelos estes que são representados por relações matemáticas (AGUIRRE, 2015).

Figura 6 – Série Temporal Mackey-Glass.



Fonte: (ABDOLLAHZADE et al., 2015).

### 2.3.2 Categorias da Modelagem Matemática

De acordo com Aguirre (2015), há várias formas de classificar técnicas de modelagem, sendo que uma delas agrupa os métodos em três categorias: modelagem caixa branca, modelagem caixa preta e modelagem caixa cinza. Para realizar a modelagem caixa branca, é preciso ter um profundo conhecimento a respeito do sistema a ser modelado, isto é, das leis físicas, dos fenômenos envolvidos e das relações matemáticas que os descrevem. Entretanto, nem sempre é viável utilizar esse tipo de modelagem, pois é necessário grande conhecimento e recursos de tempo podem ser escassos em determinadas situações.

Já a modelagem caixa preta, também referenciada como identificação de sistemas (LJUNG, 1987), estuda formas alternativas de se obter o modelo matemático do sistema. Essa abordagem é aplicada quando não há nenhum conhecimento prévio a respeito das leis físicas que regulam o sistema, somente dados experimentais estão disponíveis. Dessa forma, esses modelos são construídos a partir das relações causa e efeitos desses entre as variáveis de entrada e saída presentes nesse conjunto de dados.

Por sua vez, a modelagem caixa cinza busca combinar as vantagens dos procedimentos de identificação caixa-preta e caixa-branca. Nesse caso, tanto dados de entrada e saída obtidos no sistema, quanto informação auxiliar são usados na identificação

### 2.3.3 Procedimento da Modelagem Matemática

Segundo Walter e Pronzato (1997), o procedimento para construção de um modelo matemático consiste em cinco etapas:



1. Coleta de dados;
2. Escolha da estrutura do modelo;
3. Definição do critério de qualidade;
4. Obtenção dos valores ótimos para os parâmetros;
5. Validação dos resultados.

sendo que essas etapas não necessariamente são executadas nessa ordem.

A etapa de coleta de dados consiste na obtenção dos dados de entrada e saída, efetuando-se testes no sistema de forma a extrair a informação dinâmica do sistema. Preocupações nessa fase são a escolha dos sinais de excitação e a seleção do tempo de amostragem (AGUIRRE, 2015).

A escolha da estrutura do modelo matemático é denominado de caracterização. Essa etapa é crítica, já que a intuição do especialista é muito importante. De acordo com Palit e Popovic (2005), os principais modelos utilizados para séries temporais são:

- Modelos regressivos;
- Modelos no domínio do tempo;
- Modelos no domínio da frequência;
- Modelos não-lineares.

sendo que estes serão apresentados nas próximas subseções.

Realizada a caracterização do modelo, a estimação de valores ótimos para os modelos precisa ser efetuada. Nessa etapa, precisa-se escolher o algoritmo a ser aplicado com esse objetivo. Exemplos de métodos que visam estimar os parâmetros são os mínimos quadrados e o filtro de Kalman (WALTER; PRONZATO, 1997).

Por fim, a validação visa verificar se o modelo atende os requisitos de desempenho exigidos. Além disso, é interessante também comparar diversos modelos candidatos nessa fase. Sendo assim, a validação é bem subjetiva e depende da aplicação pretendida para o modelo e a quantidade de informação disponível a respeito do sistema original (AGUIRRE, 2015).

#### 2.3.4 Modelos Regressivos

Os modelos regressivos, muito utilizados na estatística e engenharia, focam na relação entre uma variável dependente e uma ou mais variáveis independentes. Logo, esses modelos são

analisados para entender o comportamento da variável dependente enquanto as variáveis independentes são modificadas. Além dessas variáveis, esses modelos também possuem parâmetros que precisam ser estimados, aplicando-se, por exemplo, o método dos mínimos quadrados. No contexto de séries temporais, os modelos de regressão mais aplicados, segundo Palit e Popovic (2005), são os Autorregressivos (*Autoregressive - AR*), Médias Móveis (*Moving Average - MA*), os Autorregressivos de Médias Móveis (*Autoregressive Moving Average - ARMA*) e os Autorregressivos Integrados de Médias Móveis (*Autoregressive Integrated Moving Average - ARIMA*). Um grande desafio que envolve os modelos regressivos é a seleção de qual modelo é adequado e a determinação de sua ordem. A partir do formato e comportamento das funções de autocovariância e de autocorrelação é possível inferir qual modelo seria mais apropriado para determinada série temporal. Já para designar a ordem desses modelos, alguns critérios tais como Critério de Akaike e Critério de Informação Bayesiana podem ser empregados (YOUNG, 2014).

#### 2.3.4.1 Modelos AR

Os modelos autorregressivos expressam o valor da série temporal no instante atual em função de seus valores para instantes anteriores. Essa categoria de modelo regressivo é válida apenas para séries lineares e estacionárias. Um modelo  $AR(n_y)$  é representado conforme a Equação 2.14 a seguir (BROCKWELL; DAVIS, 2002) (PALIT; POPOVIC, 2005):

$$y_k = \alpha_1 y_{k-1} + \alpha_2 y_{k-2} + \dots + \alpha_{n_y} y_{k-n_y} + w_k, \quad (2.14)$$

onde os valores  $\alpha_1$  a  $\alpha_{n_y}$  são os parâmetros a serem estimados,  $w_k$  é um ruído geralmente branco e  $n_y$  é a ordem do modelo.

#### 2.3.4.2 Modelos MA

Outra abordagem para modelagem de séries temporais são os modelos de  $MA(n_w)$ , estruturados conforme a Equação 2.15 (PALIT; POPOVIC, 2005):

$$y_k = w_k + \beta_1 w_{k-1} + \beta_2 w_{k-2} + \dots + \beta_{n_w} w_{k-n_w}, \quad (2.15)$$

onde  $n_w$  é a ordem do modelo e as constantes  $\beta_1$  a  $\beta_{n_w}$  são os parâmetros a serem estimados. Esses modelos podem ser utilizados para séries temporais univariáveis, estacionárias e lineares.

#### 2.3.4.3 Modelos ARMA

A combinação dos modelos  $AR(n_y)$  e  $MA(n_w)$  formam o modelo  $ARMA(n_y, n_w)$ , o mais utilizado para modelar séries temporais estacionárias (AGUIRRE, 2015). Matematicamente,

um modelo  $ARMA(n_y, n_w)$  possui a estrutura apresentada na Equação 2.16 (PALIT; POPOVIC, 2005):

$$y_k = \alpha_1 y_{k-1} + \alpha_2 y_{k-2} + \dots + \alpha_{n_y} y_{k-n_y} + w_k + \beta_1 w_{k-1} + \beta_2 w_{k-2} + \dots + \beta_{n_w} w_{k-n_w} \quad (2.16)$$

sendo que os parâmetros  $\alpha_1$  a  $\alpha_{n_y}$  estão relacionados às observações da série temporal e os parâmetros  $\beta_1$  a  $\beta_{n_w}$  com as observações do ruído branco.

#### 2.3.4.4 Modelos ARIMA

Um modelo  $ARIMA(n_y, n_d, n_w)$  é o caso geral dessa classe de modelos regressivos, sendo que os outros apresentados acima podem ser deduzidos a partir desse modelo. Dessa forma, o modelo  $ARIMA(n_y, 0, 0)$  equivale a um  $AR(n_y)$ , o  $ARIMA(0, 0, n_w)$  é igual ao  $MA(n_w)$  e o  $ARIMA(n_y, 0, n_w)$  é o mesmo modelo  $ARMA(n_y, n_w)$ . O modelo  $ARIMA$  é apropriado para aplicações que envolvem séries temporais não-estacionárias, tornando-as estacionárias por meio do operador de diferenças (MONTGONOMERY; JENNINGS; KULAHCI, 2015). O operador de diferenças, que corresponde a parte integrativa do  $ARIMA$ , é definido matematicamente conforme a seguir (AGUIRRE, 2015):

$$\begin{aligned} \Delta y_k &= y_k - y_{k-1} && \text{Primeira diferença;} \\ \Delta^2 y_k &= \Delta[\Delta y_k] = \Delta y_k - \Delta y_{k-1} && \text{Segunda diferença;} \\ \Delta^{n_d} y_k &= \Delta[\Delta^{n_d-1} y_k] && \text{Caso geral,} \end{aligned} \quad (2.17)$$

sendo que  $n_d$  é a ordem do operador de diferenças. Geralmente, a primeira ou a segunda diferença é suficiente para transformar uma série não-estacionária em estacionária. Por conveniência, a Equação 2.18 exhibe um modelo  $ARIMA(1,1,1)$ :

$$\begin{aligned} \Delta y_k &= \alpha \Delta y_{k-1} + w_k + \beta w_{k-1}, \\ y_k &= \alpha y_k + (1 - \alpha) y_{k-1} + w_k + \beta w_{k-1}. \end{aligned} \quad (2.18)$$

#### 2.3.5 Modelos no Domínio do Tempo

De acordo com Palit e Popovic (2005), dois tipos de modelos no domínio do tempo são de grande importância na modelagem de séries temporais:

- Modelos de Função de Transferência;
- Modelos no Espaço de Estados.

##### 2.3.5.1 Modelos de Função de Transferência

Os modelos desse tipo são considerados uma extensão dos modelos de regressão comentados, sendo representados por funções de transferência que modelam a dinâmica da série

temporal. A construção de modelos de função de transferência se baseia em dados experimentais de séries temporais de entrada e saída. A título de exemplo, seja o modelo ARMA( $n_y, n_w$ ) da Equação 2.16. De forma alternativa, esse modelo pode ser representado conforme a Equação 2.19, considerando  $z^{-1}$  como operador de atraso de forma que  $y_k z^{-1} = y_{k-1}$  (AGUIRRE, 2015):

$$\begin{aligned} A(z^{-1})y_k &= B(z^{-1})w_k, \text{ onde} \\ A(z^{-1}) &= 1 - \alpha_1 z^{-1} - \dots - \alpha_{n_y} z^{-n_y} \text{ e} \\ B(z^{-1}) &= 1 + \beta_1 z^{-1} + \dots + \beta_{n_w} z^{-n_w} \end{aligned} \quad (2.19)$$

Portanto, a função de transferência desse sistema é denotada por:

$$\frac{y_k}{w_k} = \frac{B(z^{-1})}{A(z^{-1})}. \quad (2.20)$$

### 2.3.5.2 Modelos no Espaço de Estados

Os modelos no espaço de estados fornecem uma maneira prática e compacta para modelar e analisar séries temporais multivariáveis, consistindo de duas equações. A primeira é conhecida como equação de observação, que é denotada pela Equação 2.21 (BROCKWELL; DAVIS, 2002) (HAMILTON, 1994):

$$\begin{aligned} \mathbf{y}_k &= \mathbf{C}_k \mathbf{x}_k + \mathbf{w}_k, \\ \text{para } \mathbf{y}_k &\in \mathbb{R}^{m \times 1}, \mathbf{C}_k \in \mathbb{R}^{m \times n}, \mathbf{x}_k \in \mathbb{R}^{n \times 1} \text{ e } \mathbf{w}_k \in \mathbb{R}^{m \times 1}, \end{aligned} \quad (2.21)$$

sendo que  $\mathbf{y}_k$  é a série multivariável,  $\mathbf{C}_k$  é a matriz de observação, os estados são representados por  $\mathbf{x}_k$  e  $\mathbf{w}_k$  é uma sequência de ruído branco. A segunda equação, comumente chamada de equação de estado, determinar o valor de  $\mathbf{x}_{k+1}$  da seguinte forma:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}_k \mathbf{x}_k + \mathbf{a}_k \\ \text{para } \mathbf{A}_k &\in \mathbb{R}^{n \times n}, \mathbf{x}_k \in \mathbb{R}^{n \times 1} \text{ e } \mathbf{a}_k \in \mathbb{R}^{n \times 1}, \end{aligned} \quad (2.22)$$

onde  $\mathbf{A}_k$  é a matriz de estado e  $\mathbf{a}_k$  é uma sequência de ruído branco não-correlacionada com  $\mathbf{w}_k$ .

É possível encontrar uma representação no espaço de estados para variados tipos de séries temporais, sendo que nem os estados  $\mathbf{x}_k$  nem a série  $\mathbf{y}_k$  precisam necessariamente serem estacionários. Uma das grandes vantagens da apresentação no espaço de estados é que tanto os estados e o comportamento de  $\mathbf{y}_k$  são facilmente obtidos de acordo com as equações acima, sendo muito úteis para representação da dinâmica de modelos multivariáveis.

A metodologia proposta nessa pesquisa utiliza a representação de espaço de estados para modelar o comportamento das componentes não-observáveis (Seção 4.2.4).

### 2.3.6 Modelos no Domínio da Frequência

Uma série temporal pode ser representada no domínio da frequência, sendo constituída por um conjunto de senos e cossenos com diferentes frequências. Essa abordagem consiste de uma representação de Fourier dada pela Equação 2.23 (YOUNG, 2014):

$$y_k = \sum_{k=1}^T a_k \sin(2\pi f_k) + b_k \cos(2\pi f_k) \quad (2.23)$$

onde a frequência é  $f_k = k/T$ , para  $T$  o período. Os termos  $a_k$  e  $b_k$  são calculados da seguinte forma:

$$\begin{aligned} a_k &= \frac{2}{T} \sum_{k=1}^T \cos(2\pi f_k), \\ b_k &= \frac{2}{T} \sum_{k=1}^T \sin(2\pi f_k). \end{aligned} \quad (2.24)$$

Uma ferramenta utilizada para representar visualmente a série temporal no domínio da frequência é conhecida como periodograma, computado pela Equação 2.25 (BROCKWELL; DAVIS, 2002):

$$I(f_k) = \frac{T}{2}(a_k^2 + b_k^2). \quad (2.25)$$

Essa técnica é aplicada para produzir o espectro ou a função de densidade espectral. A função de densidade espectral é interpretada como a transformada de Fourier da função de autocorrelação, portanto, ela disponibiliza informações semelhantes às encontradas nas autocorrelações (BROCKWELL; DAVIS, 2002) (PALIT; POPOVIC, 2005).

### 2.3.7 Modelos Não-Lineares

As observações de séries temporais não-lineares podem conter vários padrões de diferentes formatos e intensidades. Na prática, é muito comum em primeiro lugar modelar uma série por meio de representações lineares (AGUIRRE, 2015). Se esse modelo não for adequado para esta série, então um modelo não-linear precisa ser considerado. Sendo assim, o especialista se depara com o desafio de selecionar o modelo não-linear apropriado. Exemplos de modelos não-lineares tradicionais são (PALIT; POPOVIC, 2005):

- Modelos STAR (*Smooth Transition Autoregression Model*);
- Modelos ARCH (*Autoregressive Conditional Heteroskedasticity Model*);
- Modelos bilineares;
- Modelos pelas Cadeias de Markov.

Além desses modelos tradicionais, métodos baseados em inteligência computacional vêm sendo amplamente utilizados para séries temporais não-lineares. É possível citar as redes neurais, a lógica *fuzzy*, as redes *Wavelets*, a computação evolucionária com os algoritmos genéticos, a nuvem de partículas e a programação genética. Conforme Palit e Popovic (2005), é uma tendência utilizar abordagens híbridas ao trabalhar com séries, aproveitando as vantagens de variadas metodologias. Um exemplo dessas técnicas híbridas é a rede *neuro-fuzzy* evolutiva aplicada dessa pesquisa.

## 2.4 Aspectos sobre Previsão de Séries Temporais

Após a obtenção de um modelo para determinada série temporal, ele pode ser aplicado para prever valores futuros usando um método apropriado de acordo com a série temporal. Dado um conjunto de amostras  $y_1, y_2, \dots, y_n$  de uma série temporal, a previsão visa obter os dados  $y_{n+1}, y_{n+2}, \dots$  (HAMILTON, 1994). Ao selecionar um método de previsão, o analista da série temporal deve levar em consideração alguns critérios e condições, como a exatidão desejada, quais recursos computacionais estão disponíveis, quantos passos a frente deseja-se prever, quantos dados estão disponíveis, entre outras questões. Primeiramente, para qualquer método de previsão, é recomendável visualizar graficamente os dados da série temporal, pois muitas informações úteis podem ser obtidas a partir de um exame visual da série, auxiliando na escolha do método de previsão (PALIT; POPOVIC, 2005).

Problemas de previsão são frequentemente classificados em curto, médio ou longo prazo. Segundo Montgomery, Jennings e Kulahci (2015), as previsões de curto e médio prazo são geralmente utilizadas em atividades que vão desde a gestão de operações até o orçamento e seleção de novos projetos de pesquisa e desenvolvimento. Já as previsões a longo prazo estão relacionadas com o planejamento estratégico de uma empresa, por exemplo. Como os dados históricos de séries temporais geralmente apresentam padrões ocultos em seu comportamento, os métodos de previsão buscam reconstruir esses padrões para realizar a previsão de séries temporais (BROCKWELL; DAVIS, 2002).

Dois conceitos muito importantes no problema de previsão são o horizonte de previsão  $h$  e o intervalo de previsão. O primeiro corresponde a quantidade de passos a frente que se deseja prever dados. Geralmente, o valor do horizonte é definido de acordo com a natureza do problema e dos objetivos em questão. Por sua vez, o intervalo de previsão é a frequência com que são preparadas novas previsões (MONTGONOMERY; JENNINGS; KULAHCI, 2015). Como exemplo, deseja-se realizar a previsão de uma série temporal, com horizonte igual a 3 passos a frente e intervalo de previsão igual a 50. Isso significa que a cada 50 amostras, uma previsão de 3 passos a frente é realizada.

De acordo com Montgomery, Jennings e Kulahci (2015), há dois grandes tipos de técnicas de previsão: os métodos qualitativos e os métodos quantitativos. Seus conceitos serão apresentados nas Subseções 2.4.1 e 2.4.2.

### 2.4.1 Descrição e Objetivos da Previsão Qualitativa

As técnicas de previsão qualitativas são de natureza subjetiva, ou seja, consideram sobretudo o julgamento subjetivo por parte dos especialistas, embora uma análise superficial dos dados possa ser realizada. Essas previsões qualitativas são frequentemente utilizadas em situações em que há pouca ou nenhuma informação histórica disponível (CHAMBERS; MULLICK; SMITH, 1971). Um exemplo apresentado em Montgomery, Jennings e Kulahci (2015) seria a introdução de um novo produto em uma empresa, para o qual não há dados históricos relevantes. Dessa maneira, a empresa pode considerar opiniões de potenciais clientes, de especialistas de vendas e de *marketing*, e análise dos concorrentes para prever subjetivamente as vendas desse produto durante a fase de introdução. Os métodos para previsão qualitativa variam em relação ao custo e complexidade. Eles podem ser usados separadamente, mas são geralmente integrados com outros métodos qualitativos ou com métodos quantitativos (MAKRIDAKIS; WHEELWRIGHT; HYNDMAN, 1998).

A técnica de previsão qualitativa mais amplamente utilizada é o Método *Delphi*, desenvolvido pela *RAND Corporation* (DALKEY, 1967). Este método considera um conjunto de especialistas que estão fisicamente separados para evitar que suas opiniões sejam afetadas por outros indivíduos. Cada um deve responder um questionário contendo perguntas sobre o domínio do problema. Depois da aplicação desse primeiro questionário, as próximas perguntas são submetidas a cada especialista juntamente com as informações sobre as opiniões de todos os outros, permitindo que eles revejam suas previsões observando as considerações de todo o grupo. Após a realização desse procedimento sucessivas vezes, espera-se que as opiniões dos palestrantes convirjam para um consenso (KAYO; SECURATO, 1997). É importante destacar que a escolha dos especialistas que participarão desse processo é uma das principais questões no método Delphi, já que a qualidade da previsão depende desses envolvidos (DIETZ, 1987). Essa seleção visa formar um conjunto equilibrado, contando com especialistas em universidades, institutos de pesquisa e indústrias.

Outro método para previsão qualitativa é o de Pesquisa de Intenções, que avalia planos, metas e expectativas de indivíduos que são entrevistados sobre o futuro de uma variável ou evento considerado. Esse método analisa padrões que podem descrever as preferências dos consumidores e a probabilidade que existe de eles comprarem determinado produto (CHAMBERS; MULLICK; SMITH, 1971). Os resultados dessa abordagem podem ser influenciados por três tipos de erros: erro de amostragem, erro das respostas e erro devido a falta de respostas. O primeiro tipo, o erro de amostragem, ocorre quando a amostra não é representativa. Já o erro das respostas acontece quando o indivíduo entrevistado não compreende as perguntas da pesquisa. Por fim, o erro devido a falta de respostas acontece quando o entrevistado não é encontrado ou se nega a responder o questionário (ARMSTRONG, 2001).

É possível citar ainda o método do Jogo de Representação, que é aplicado quando há conflitos que envolvem grandes mudanças entre os especialistas e pouca informação existe sobre

eventos similares no passado. Entende-se por situações de conflito aquelas em que duas ou mais partes têm objetivos opostos, diferentes estratégias, ou competem por um dado recurso (ARMSTRONG, 2001). Aplicações bem sucedidas do Jogo de Representação têm sido consideradas em diversas áreas, sendo muito aplicado por empresas para previsões das reações dos concorrentes devido a mudanças de estratégia da organização ou a modificações de cenários de mercado (ARMSTRONG, 2001).

#### 2.4.2 Procedimento Geral para Previsão Quantitativa

As técnicas de previsão quantitativas usam um modelo de previsão e consideram um grande conjunto de informações históricas da série temporal. Esse modelo relaciona os padrões nos dados e expressa uma relação estatística entre os valores anteriores e os valores atuais da variável, sendo aplicado em seguida para projetar os dados no futuro. Segundo Goodwin (2002), os resultados dos métodos quantitativos podem ser refinados pelos qualitativos ao considerar o conhecimento subjetivo do especialista. Em Montgonomery, Jennings e Kulahci (2015), encontra-se uma interessante metodologia para previsão constituída por uma sequência de atividades listadas a seguir:

1. Definição do problema;
2. Coleta dos dados;
3. Análise dos dados;
4. Seleção do modelo;
5. Validação do modelo;
6. Implementação do modelo de previsão;
7. Monitoramento do desempenho do modelo de previsão.

A primeira atividade é a definição do problema, que engloba a compreensão de como os resultados da previsão serão aplicados em determinado problema. Nessa etapa, define-se o horizonte de previsão, o intervalo de previsão e qual nível de exatidão é desejável para adequação ao problema de previsão (MONTGONOMERY; JENNINGS; KULAHCI, 2015).

A coleta de dados consiste em obter informações históricas relevantes para realização da previsão, pois nem todos os dados são úteis para o problema em questão. Todavia, muitas vezes é necessário lidar com valores ausentes de algumas amostras da série, *outliers* (valores discrepantes) ou outros problemas relacionados aos dados coletados. À vista disso, a terceira atividade, a análise dos dados, deve ser aplicada antes de selecionar modelos candidatos para previsão, visando resolver esses empecilhos citados. Sendo assim, é importante inspecionar visualmente os dados e verificar se há padrões reconhecíveis como tendências, sazonalidades e



outras componentes cíclicas (BROCKWELL; DAVIS, 2002). Essa visualização é importante pois proporciona uma intuição ao analista sobre quais modelos serão adequados de acordo com o comportamento da série temporal. Nessa fase, também devem ser calculados e avaliados as médias, o desvio padrão, as funções de autocorrelações, entre outras medidas numéricas e estatísticas.

A etapa de seleção do modelo consiste em escolher modelos candidatos para previsão da série e, após a escolha, estimar os parâmetros do modelo selecionado, utilizando, por exemplo, o método dos mínimos quadrados. Há vários tipos de modelos de previsão de uso geral. Tradicionalmente, são considerados os modelos regressivos apresentados na Seção 2.3 e os modelos de suavização exponencial. Os modelos regressivos utilizam um algoritmo matemático complexo com termos autorregressivos e de média móvel para identificar a forma do modelo matemático mais adequado para a série temporal (ARCHER, 1980). A partir dos dados disponíveis, esses modelos visam capturar o comportamento da correlação seriada ou autocorrelação entre os valores da série temporal, e com base nesse comportamento, realizar previsões futuras (MAKRIDAKIS; WHEELWRIGHT; HYNDMAN, 1998). Já os modelos de suavização exponencial se caracterizam por decompor uma série temporal em componentes não-observáveis, suavizar seus valores passados e recompor as componentes para previsão dos dados (TRATAR; STRMCNIK, 2016) (SZMIT; SZMIT, 2012). Além de serem aplicados para a previsão de séries temporais, também são considerados para o objetivo de decomposição de componentes não-observáveis, de acordo com a Seção 2.5.1.

Por sua vez, a validação do modelo visa avaliar o modelo de previsão para verificar se este cumpre os requisitos exigidos e se as magnitudes dos erros não ultrapassem os limites impostos na definição do problema (MONTGONOMERY; JENNINGS; KULAHCI, 2015). É importante destacar que os erros cometidos no processo de treinamento e ajuste do modelo são bem menores que os erros cometidos na previsão e na validação. Uma abordagem amplamente aplicada para validação de modelos é a divisão dos dados disponíveis em dois segmentos, um para sintonizar o modelo e outro para efetivamente realizar a simulação da previsão sem ajustes. Essa técnica fornece uma orientação útil sobre como o modelo desenvolvido irá atuar quando exposto a novos dados da série, sendo que seus resultados podem ser utilizados para discriminar modelos de previsão concorrentes (AGUIRRE, 2015).

A implementação do modelo de previsão envolve sua efetiva utilização pelo usuário. Nessa etapa, é importante garantir que esse usuário entenda como aplicar o modelo e como gerar previsões úteis. Por fim, a última atividade é o monitoramento do desempenho do modelo de previsão, que deve ser contínuo. Um modelo que funcionou bem em uma situação pode deteriorar seu desempenho no futuro, já os dados e as condições mudam ao longo do tempo. Normalmente, essa deterioração do desempenho resulta em erros de previsão maiores ou mais sistemáticos. Logo, monitoramento de erros de previsão é uma parte essencial do bom projeto de sistema de previsão (BROCKWELL; DAVIS, 2002) (MONTGONOMERY; JENNINGS; KULAHCI,

2015).

## 2.5 Aspectos sobre Decomposição de Séries Temporais

Outro importante objetivo dos especialistas ao estudarem séries temporais é a decomposição de padrões relevantes em seus dados. Esses padrões ocultos da série também podem ser chamados de componentes não-observáveis. As componentes não-observáveis são bem mais simples que a série original, sendo muito útil para que os especialistas analisem e entendam seu comportamento geral (HARVEY; KOOPMAN, 2009). A extração dessas componentes da série temporal se encaixa nos objetivos da metodologia de previsão proposta nessa pesquisa, que se baseia na divisão de um problema complexo em vários subproblemas mais simples de lidar.

Séries complexas podem exibir componentes ou padrões de diversas naturezas. A quantidade e o tipo de componentes não-observáveis identificadas dependem do método usado para extraí-las, das características da série temporal e dos objetivos dos especialistas (BROCKWELL; DAVIS, 2002). Alguns exemplos de componentes não-observáveis tradicionais são:

- Tendência;
- Sazonalidade;
- Padrão Cíclico;
- Impulso;
- Degrau.

Muitas abordagens podem ser usadas para realizar essa tarefa de caracterização ou decomposição. É possível citar o procedimento X-11, que é baseado na estimação iterativa das componentes usando médias móveis. Este método é capaz de identificar a tendência, a sazonalidade e uma componente irregular a partir dos dados originais da série temporal. Outro método é a Decomposição da Tendência e da Sazonalidade usando *Loess*, do inglês *Seasonal and Trend Decomposition using Loess* (STL). Esse último extrai apenas a tendência e a sazonalidade da série original (DENG, 2014).

Para um melhor entendimento sobre a decomposição de séries temporais, dois tipos de métodos com essa finalidade serão descritos: os métodos de Suavização Exponencial e os métodos de Análise Espectral Singular.

### 2.5.1 Métodos de Suavização Exponencial

Os métodos de suavização exponencial formam uma grande classe de técnicas para a modelagem e previsão de séries temporais. As equações que compõem essa classe de métodos são ponderadas exponencialmente, visando produzir ajustes nas variações aleatórias dos dados

da série temporal. Essas técnicas de suavização exponencial usam apenas informações da própria série, têm grande popularidade e são ferramentas muito empregadas devido a sua simplicidade, eficiência computacional e a sua razoável precisão. Esses métodos lidam com a decomposição das séries temporais nas componentes de tendência, nível e sazonalidade, assim como um termo aleatório (BROCKWELL; DAVIS, 2002) (MONTGONOMERY; JENNINGS; KULAHCI, 2015). O nível é o valor médio da observação no período (valor observado retirando a sazonalidade e a componente aleatória), a tendência é a diferença sequencial de dois níveis consecutivos e a sazonalidade é um evento que se repete com uma periodicidade constante (MIRANDA, 2009). As abordagens de suavização exponencial se diferenciam uma das outras de acordo com as componentes de que tratam seus modelos.

Uma das técnicas mais utilizadas dessa classe é o método de Holt-Winters (HW) ou Suavização Exponencial Tripla, que utiliza as três componentes acima citadas e o termo aleatório (SZMIT; SZMIT, 2012). Quando se lida com séries temporais com um padrão de comportamento complexo e que claramente possuem períodos sazonais e tendências, o método de HW é adequado. Existem dois procedimentos principais que dependem das características da série para o uso do método de HW: o multiplicativo e o aditivo. Ambos os procedimentos apresentam três equações com constantes de suavização diferentes, associadas a cada uma das componentes da série temporal: nível, tendência e sazonalidade (TRATAR; STRMCNIK, 2016) (SZMIT; SZMIT, 2012).

O método de HW multiplicativo é apropriado na modelagem de dados quando a amplitude do ciclo sazonal varia com o passar do tempo. Sendo assim, seja uma série temporal  $y_k$  com período sazonal  $s$ . Segundo Tratar e Strmcnik (2016) e Szmit e Szmit (2012), o modelo multiplicativo para obter uma aproximação  $\hat{y}_k$  da série é expressado por:

$$\hat{y}_k = z_k s_k + t_k + w_k, \quad k = 1, \dots, N, \quad (2.26)$$

na qual  $z_k$  é o nível,  $s_k$  é a sazonalidade,  $t_k$  é a tendência,  $w_k$  é uma componente aleatória e  $N$  é a quantidade de dados da série temporal considerada. Para esse procedimento, as três equações de suavização são dadas por:

$$\begin{aligned} \hat{z}_k &= A \left( \frac{y_k}{\hat{s}_{k-s}} \right) + (1-A)(\hat{z}_{k-1} + \hat{t}_{k-1}), \quad 0 < A < 1, \quad k = s+1, \dots, N; \\ \hat{t}_k &= C(\hat{z}_k - \hat{z}_{k-1}) + (1-C)\hat{t}_{k-1}, \quad 0 < C < 1, \quad k = s+1, \dots, N; \\ \hat{s}_k &= D \left( \frac{y_k}{\hat{z}_k} \right) + (1-D)\hat{s}_{k-s}, \quad 0 < D < 1, \quad k = s+1, \dots, N, \end{aligned} \quad (2.27)$$

sendo que representam respectivamente o nível, a tendência e a sazonalidade, assim como suas constantes de suavização associadas  $A$ ,  $C$  e  $D$ .

Em contrapartida, a abordagem aditiva é usada para séries temporais cuja amplitude do ciclo sazonal é aproximadamente constante. Seja novamente a série temporal  $y_k$  com período sazonal  $s$ . De acordo com Tratar e Strmcnik (2016), o modelo aditivo é representado pela seguinte relação:

$$\hat{y}_k = z_k + s_k + t_k + a_k, \quad k = 1, \dots, N. \quad (2.28)$$

De forma similar ao procedimento anterior, a abordagem aditiva é composta pelas equações:

$$\begin{aligned} \hat{z}_k &= A(y_k - \hat{s}_{k-s}) + (1-A)(\hat{z}_{k-1} + \hat{t}_{k-1}), \quad 0 < A < 1, \quad k = s+1, \dots, N; \\ \hat{t}_k &= C(\hat{z}_k - \hat{z}_{k-1}) + (1-C)\hat{t}_{k-1}, \quad 0 < C < 1, \quad k = s+1, \dots, N; \\ \hat{s}_k &= D(y_k - \hat{z}_k) + (1-D)\hat{s}_{k-s}, \quad 0 < D < 1, \quad k = s+1, \dots, N. \end{aligned} \quad (2.29)$$

Os valores iniciais para as equações das componentes nível, tendência e sazonalidade em ambos os procedimentos apresentados podem ser calculados pelas seguintes fórmulas:

$$\begin{aligned} \hat{s}_j &= \frac{y_j}{\frac{1}{s} \sum_{i=1}^s y_i}, \quad j = 1, 2, \dots, s; \\ \hat{z}_s &= \frac{1}{s} \sum_{i=1}^s y_i; \\ \hat{t}_s &= 0. \end{aligned} \quad (2.30)$$

Existem dificuldades em determinar os valores apropriados para as constantes de suavização e de estudar as propriedades estatísticas como a média e a variância de previsão (BROCKWELL; DAVIS, 2002) (TRATAR; STRMCNIK, 2016) (SZMIT; SZMIT, 2012). A determinação das constantes de suavização  $A$ ,  $C$  e  $D$  é realizada de modo a tornar mínimo o erro quadrático médio dos erros de ajustamento, ou seja:

$$EQM = \frac{1}{N} \sum_{t=1}^N (y_k - \hat{y}_k)^2 \quad (2.31)$$

em que  $\hat{y}_k$  é a série temporal gerada pela Equação 2.26 ou Equação 2.28.

### 2.5.2 Análise Espectral Singular - AES

O método conhecido como Análise Espectral Singular (AES), do inglês *Singular Spectral Analysis* (SSA), é uma técnica recente para análise e decomposição de séries temporais complexas

com variadas características (ABDOLLAHZADE et al., 2015). Este método se caracteriza por ser em batelada e por incorporar conceitos clássicos sobre análise de séries temporais, estatística multivariável, sistemas dinâmicos e processamento de sinais. O AES não é considerado um método estatístico, porque seu algoritmo não faz nenhuma suposição estatística prévia a respeito dos dados da série temporal. O objetivo do AES é decompor a série temporal original em uma soma de componentes interpretáveis, tal como tendências, sazonalidades e outros tipos de padrões (GOLYANDINA; NEKRUTKIN; ZHIGLJAVSKY, 2001).

Segundo Golyandina, Nekrutkin e Zhigljavsky (2001), o algoritmo básico do AES é organizado da seguinte forma:

1. Estágio de Decomposição:

- Etapa de Incorporação;
- Etapa de Decomposição em Valores Singulares.

2. Estágio de Reconstrução:

- Etapa de Agrupamento;
- Etapa de Média Diagonal.

### 2.5.2.1 Etapa de Incorporação

O objetivo dessa etapa é mapear uma série temporal unidimensional ou vetorial para uma representação multidimensional ou matricial (GOLYANDINA; NEKRUTKIN; ZHIGLJAVSKY, 2001). Seja uma série temporal com  $N$  amostras e com pelo menos um valor não-nulo constituída pelas observações  $y_1, y_2, \dots, y_N$ . Seja também um parâmetro inteiro  $L$  a ser determinado pelo especialista, conhecido como dimensão dos vetores  $\mathbf{m}_q$ , para  $1 < L < \frac{N}{2}$ .

Sendo assim, formam-se  $K$  vetores com dimensão  $L$ , para  $K = N - L + 1$ . Logo:

$$\mathbf{m}_q = (y_q, \dots, y_{q+L-1})^T, \quad 1 \leq q \leq K. \quad (2.32)$$

Para mapeamento dos dados de forma multidimensional, esses vetores obtidos são usados para desenvolver a matriz trajetória da série temporal, representada por:

$$\mathbf{M} = [\mathbf{m}_1 : \dots : \mathbf{m}_K] = \begin{bmatrix} y_1 & y_2 & y_3 & \cdots & y_K \\ y_2 & y_3 & y_4 & \cdots & y_{K+1} \\ y_3 & y_4 & y_5 & \cdots & y_{K+2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ y_L & y_{L+1} & y_{L+2} & \cdots & y_N \end{bmatrix}_{L \times K} \quad (2.33)$$

A matriz trajetória pode ser caracterizada como uma matriz Hankel<sup>1</sup>, já que os elementos ao longo das diagonais que ascendem da esquerda para a direita são iguais.

### 2.5.2.2 Etapa de Decomposição em Valores Singulares

Essa etapa visa realizar a Decomposição em Valores Singulares (*Singular Value Decomposition* - SVD) da matriz trajetória calculada anteriormente (ABDOLLAHZADE et al., 2015) (GOLYANDINA; NEKRUTKIN; ZHIGLJAVSKY, 2001). Seja  $\mathbf{S} = \mathbf{M}\mathbf{M}^T$  conhecida como matriz de covariância do AES. Os autovalores de  $\mathbf{S}$  são denotados em ordem decrescente de magnitude por:

$$\rho^1 \geq \rho^2 \geq \rho^3 \geq \dots \geq \rho^L \geq 0, \quad (2.34)$$

O sistema ortonormal de autovetores da matriz  $\mathbf{S}$ , também conhecidos como vetores singulares à esquerda, relacionados a cada um dos autovalores citados acima é representado por  $\mathbf{p}^1, \mathbf{p}^2, \mathbf{p}^3, \dots, \mathbf{p}^L$ . Define-se também o parâmetro  $d = \max \{q, \text{tal que } \rho^q > 0\}$ , que indica a quantidade de autovalores não-nulos. Considerando que:

$$\mathbf{v}_q = \frac{\mathbf{M}^T \mathbf{p}^q}{\sqrt{\rho^q}}, \quad q = 1, \dots, d. \quad (2.35)$$

então, a SVD da matriz trajetória pode ser representada por:

$$\begin{aligned} \mathbf{M} &= \mathbf{M}_1 + \mathbf{M}_2 + \dots + \mathbf{M}_d, \text{ para} \\ \mathbf{M}_q &= \sqrt{\rho^q} \mathbf{p}^q \mathbf{v}_q^T, \quad q = 1, \dots, d. \end{aligned} \quad (2.36)$$

em que as matrizes  $\mathbf{M}_q$  possuem posto<sup>2</sup> igual a 1, ou seja, são matrizes elementares. Percebe-se que a soma dessas matrizes elementares é a própria matriz trajetória.

### 2.5.2.3 Etapa de Agrupamento

Após a expansão realizada na Equação 2.36, a etapa de agrupamento é iniciada, cuja função é particionar o conjunto de índices  $\{1, \dots, d\}$  relacionados com cada matriz  $\mathbf{M}_q$  em  $n$  subconjuntos disjuntos denotados por  $I_1, \dots, I_n$ . Esse valor  $n$  corresponde a quantidade de componentes não-observáveis que serão extraídas da série temporal original e deve ser informado pelo especialista (GOLYANDINA; NEKRUTKIN; ZHIGLJAVSKY, 2001).

<sup>1</sup> Não confundir com matriz simétrica que precisa ser necessariamente quadrada ( $\mathbf{A} = \mathbf{A}^T$ ) (CHEN, 1999).

<sup>2</sup> O posto ou *rank* de uma matriz  $\mathbf{A}$  quadrada ou retangular é definido como o número de colunas e linhas linearmente independentes de  $\mathbf{A}$  (CHEN, 1999).

Vamos considerar um subconjunto genérico  $I$  contendo um número de  $p$  índices, isto é,  $I = \{i_1, \dots, i_p\}$ . Sendo assim, a matriz resultante  $\mathbf{M}_I$  associada a este subconjunto genérico é definida por:

$$\mathbf{M}_I = \mathbf{M}_{i_1} + \dots + \mathbf{M}_{i_p}. \quad (2.37)$$

Portanto, realizando esse procedimento para as matrizes resultantes de cada subconjunto  $I = I_1, \dots, I_n$ , obtém-se:

$$\mathbf{M} = \mathbf{M}_{I_1} + \dots + \mathbf{M}_{I_n}. \quad (2.38)$$

#### 2.5.2.4 Etapa de Média Diagonal

Por fim, essa etapa transforma cada uma das  $n$  matrizes resultantes da Equação 2.38 nos dados das componentes não-observáveis em formato unidimensional (GOLYANDINA; NEKRUTKIN; ZHIGLJAVSKY, 2001). Seja  $\mathbf{M}_I$  uma dessas matrizes resultantes, de ordem  $L \times K$  e com elementos  $m_{ij}$ , para  $1 \leq i \leq L$  e para  $1 \leq j \leq K$ . A componente não-observável  $c_k$  relacionada com a matriz resultante genérica  $\mathbf{M}_I$  é obtida a partir da seguinte formulação, de acordo com Abdollahzade et al. (2015):

$$c_k = \begin{cases} \frac{1}{k} \sum_{q=1}^k m_{q,k-q+1} & \text{para } i_1 \leq k \leq L, \\ \frac{1}{L} \sum_{q=1}^L m_{q,k-q+q} & \text{para } L < k \leq K, \\ \frac{1}{N-k+1} \sum_{q=k-K+1}^{N-K+1} m_{q,k-q+1} & \text{para } K < k < N. \end{cases} \quad (2.39)$$



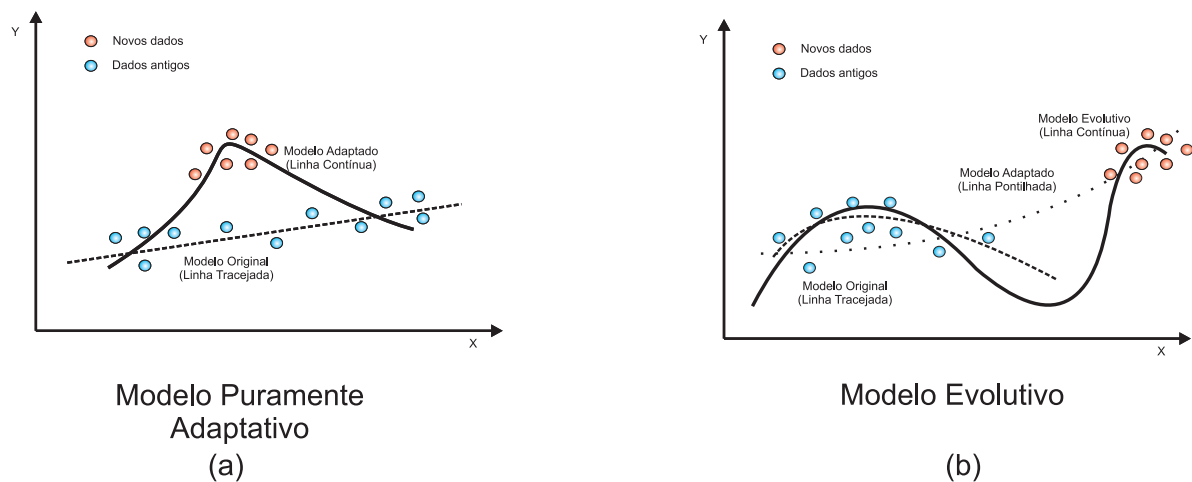


### 3 SISTEMAS EVOLUTIVOS

#### 3.1 Definição de Sistemas Evolutivos

O paradigma de sistemas evolutivos, pioneiramente proposto nas pesquisas de Angelov (2002), Kasabov (2002) e Angelov e Kasabov (2005), envolve modelos de sistemas que possuem a capacidade de expandir ou simplificar sua estrutura para se adaptar às alterações do ambiente ou mesmo a mudanças internas. Sendo assim, esses sistemas evolutivos não são apenas adaptativos, cujos parâmetros se ajustam aos dados, como também evoluem suas funcionalidades e atualizam sua inteligência conforme o comportamento destes. Lughofer (2011) define modelos evolutivos como aqueles direcionados aos dados que se adaptam automaticamente, estendendo e evoluindo dinamicamente sua estrutura em operação baseando-se nas amostras dos dados que chegam.

Figura 7 – Abrangência dos Modelos Puramente Adaptativos e dos Modelos Evolutivos.



Fonte: Adaptado de Lughofer (2011).

Há casos em que somente a adaptação dos parâmetros é suficiente para que o modelo já existente incorpore os novos dados. Entretanto, existem outras situações que apenas a adaptação não é apropriada para que as novas informações sejam consideradas pelo modelo, exigindo-se que este ajuste sua estrutura para representar bem os dados do ambiente. Esses conceitos supracitados podem ser visualizados na Figura 7. Para a Figura 7.(a) que representa um sistema puramente adaptativo, tem-se inicialmente um modelo denotado pela linha tracejada que representa o conjunto de dados antigos (azul). Quando novas amostras chegam em uma nova região (vermelho), o modelo adaptado (linha contínua) se ajusta para conseguir representá-los; entretanto, deixa de abranger alguns dados antigos como pode ser observado. Por outro lado, a Figura 7.(b) destaca a vantagem dos modelos evolutivos em relação aos puramente adaptativos. Enquanto os adaptativos podem não ser suficientes em algumas situações para abranger todo o conjunto de dados, os evolutivos são capazes de modificar sua estrutura visando representar completamente o espaço

de dados. Segundo (ANGELOV, 2013), todo sistema evolutivo é adaptativo, porém, nem todo sistema adaptativo é evolutivo.

A fim de adquirir conhecimento a partir dos dados do ambiente, os sistemas evolutivos utilizam o conceito de Aprendizado Autônomo (*Autonomous Learning*) (ANGELOV, 2013). Na maioria das situações práticas, os dados disponíveis de ambientes, de plantas ou de séries temporais são imprevisíveis, não-lineares e não-estacionários. Dessa forma, os sistemas evolutivos e autônomos devem possuir a capacidade de aprender rapidamente e extrair conhecimento desses dados coletados por sensores em tempo real ou no formato de séries temporais, minimizando a interferência do especialista nesse processo. Além disso, o novo conhecimento adquirido deve ser comparado com a inteligência apresentada pelo sistema, que avalia se essa nova informação é válida para atualizar o modelo do sistema evolutivo (ANGELOV, 2013) (LUGHOFER, 2011) (KASABOV, 2007).

Devido ao processo de globalização, há uma quantidade crescente de fluxo de dados brutos altamente dinâmico em todas as áreas; porém, existe também a necessidade de extrair conhecimentos úteis a partir desse grandioso volume de dados e a exigência de uma resposta rápida. À vista disso, é conveniente que os sistemas evolutivos operem de forma *online*. Em outras palavras, somente são necessários a amostra dos dados atual e limitadas informações agregadas para que o sistema cumpra seus objetivos e evolua conforme alterações do ambiente. Para operar dessa maneira, as metodologias baseadas em sistemas evolutivos têm interesse em algoritmos que trabalham recursivamente, o que é apropriado para sistemas *online* (ANGELOV, 2013).

Em contraste ao conceito de sistemas *online*, sistemas *offline* exigem toda a informação histórica para ser projetado. Logo, quando são necessários reajustes no sistema devido a novas exigências do ambiente ou requisitos de usuários, o sistema deverá ser reprojetoado considerando os novos dados somados ao conjunto completo de observações históricas, exigindo mais recursos computacionais, financeiros e de tempo. Grande parte dos sistemas industriais hoje em dia possui modelos que são projetados de forma batelada, utilizando um conjunto de dados históricos que não cobrem todas as possibilidades de operações e ocorrências. Logo, os sistemas evolutivos podem resolver esses obstáculos, sendo que o aprendizado *online* a partir dos dados é realizado de forma incremental e constante, portanto, proporcionando flexibilidade, precisão e segurança aos sistemas (KASABOV, 2007).

Frequentemente, o conceito de sistemas *online* é confundido com o de operação em tempo real. Um sistema de tempo real pode ser definido como qualquer sistema de processamento de informação que deve responder a um estímulo de entrada gerado externamente dentro de um período de tempo finito e específico (KOPETZ, 2011). Logo, os sistemas em tempo real precisam operar suficientemente rápido para atender a uma exigência do problema em questão, podendo ser *online* ou mesmo *offline*. Em situações em que não é preciso fornecer uma resposta rapidamente, o sistema pode utilizar todo o conjunto de dados históricos disponível para se

atualizar e mesmo assim responder a tempo aos requisitos, portanto, operando em tempo real nestas condições (ANGELOV, 2013).

Outrossim, é importante diferenciar sistemas evolutivos de sistemas evolucionários, termos erroneamente tratados como sinônimos em alguns casos. Os sistemas evolucionários são geralmente aplicados em problemas de otimização, baseando-se na genética, envolvendo conceitos como mutação, cruzamento e reprodução. O principal exemplo de técnica com essa classificação são os algoritmos genéticos (GOLDBERG, 1989). Por sua vez, os sistemas evolutivos levam em questão o aprendizado humano e a adaptação individual do conhecimento de forma constante quando em contato a novos estímulos e informações. Logo, sistemas evolutivos e sistemas evolucionários são utilizados para diferentes finalidades.

Para representar o conhecimento de forma aproximada, os sistemas evolutivos geralmente contam com uma estrutura baseada em modelos *fuzzy*, redes neurais ou redes *neuro-fuzzy* (ANGELOV, 2013). Em especial, os sistemas *fuzzy* representam o conhecimento humano no formato de regras SE-ENTÃO, como pode ser visto no Apêndice A. Assim sendo, os sistemas *fuzzy* evolutivos apresentam a habilidade de aumentar e diminuir a quantidade de regras *fuzzy* em sua base de conhecimento de acordo com uma medida de qualidade, além de adaptar alguns de seus parâmetros, tais como as funções de pertinência (LUGHOFER, 2011). Lughofer (2011) lista algumas aplicações interessantes de sistemas evolutivos:

- Previsão, estimação e filtragem;
- Classificação;
- Controle;
- Atualizar e estender modelos já existentes utilizando novas amostras;
- Desenvolver modelos a partir de grandes volumes de dados que não podem ser tratados em batelada devido a sua elevada carga computacional;
- Adicionar interação ao sistemas, fazendo com que especialistas adicionem conhecimento ao sistema de forma *online*;
- Modelagem de tarefas cognitivas do cérebro humano;
- Robôs inteligentes que precisam aprender e explorar ambientes desconhecidos.

Portanto, verifica-se que esse campo de pesquisa é relativamente recente e emergente, fornecendo maior inteligência, capacidade de aprendizado em ambientes complexos, mais automação, segurança e desempenho, além de proporcionar maior independência da interferência humana. Na próxima seção, os princípios de funcionamento dos sistemas evolutivos serão descritos.

### 3.2 Procedimentos Funcionais dos Sistemas Evolutivos

Na maioria das aplicações, a estrutura dos sistema, como redes neurais, sistemas *fuzzy* ou modelos polinomiais, é fixa e baseada no conhecimento *a priori* a respeito do problema em questão, sendo que esse conhecimento *a priori* é geralmente definido pelos especialistas. Sistemas *fuzzy*, por exemplo, tinham sua estrutura definida apenas com base no conhecimento dos especialistas nas primeiras pesquisas envolvendo essa teoria (WANG, 1997). A determinação da estrutura do sistema depende do método aplicado em sua modelagem. No caso de equações diferenciais, a quantidade de entradas e saídas, e o tipo de equação diferencial são aspectos a serem considerados. Já no caso de funções de transferência, a escolha da estrutura se resume basicamente à escolha de pólos<sup>1</sup> e zeros<sup>2</sup>, bem com a determinação do atraso puro de tempo. Por sua vez, a indicação da estrutura dos modelos no espaço de estados tem relação principalmente com a ordem das matrizes e sobre a natureza dos estados (LJUNG, 1987). Para rede neurais, a estrutura pode ser definida em camadas, memória, quantidade de entradas e saídas, entre outros aspectos. No caso de sistemas *fuzzy*, a determinação da estrutura envolve, por exemplo, a quantidade de regras, o número de entradas e saídas, a característica das funções de pertinência, o tipo de antecedentes e consequentes, a natureza dos conectivos (E, OU), o mecanismo de inferência (Mamdani ou Takagi-Sugeno) e os operadores de defuzzificação (centro de gravidade, vencedor leva tudo, etc.) (ANGELOV, 2013).

A partir desse momento, os aspectos comentados nessa seção estão relacionados a sistemas *fuzzy* e *neuro-fuzzy*, já que a metodologia proposta nessa pesquisa utiliza os conceitos relacionados a essa estrutura.

#### 3.2.1 Particionamento do Espaço de Dados por Agrupamento Evolutivo

Uma alternativa para identificar a estrutura dos sistemas é através do particionamento do espaço de dados em regiões locais (WANG, 1997). Sendo assim, essa abordagem de particionamento divide um problema complexo em vários subproblemas locais, se adequando ao princípio dessa pesquisa: dividir para conquistar.

O espaço de dados completo com dimensão  $n$  pode ser particionado em hiper-caixas ortogonais considerando os valores linguísticos e as funções de pertinência do sistemas *fuzzy* ou *neuro-fuzzy*. No entanto, essa forma de particionar o espaço é muito dispendiosa e pode ser intratável na prática quando houver muitas entradas e valores linguísticos, o que é denominado de maldição da dimensionalidade (WANG, 1997) (ANGELOV, 2013) (LUGHOFER, 2011). A quantidade de regras nesse caso cresce exponencialmente e é calculada por:

$$R = p^n \quad (3.1)$$

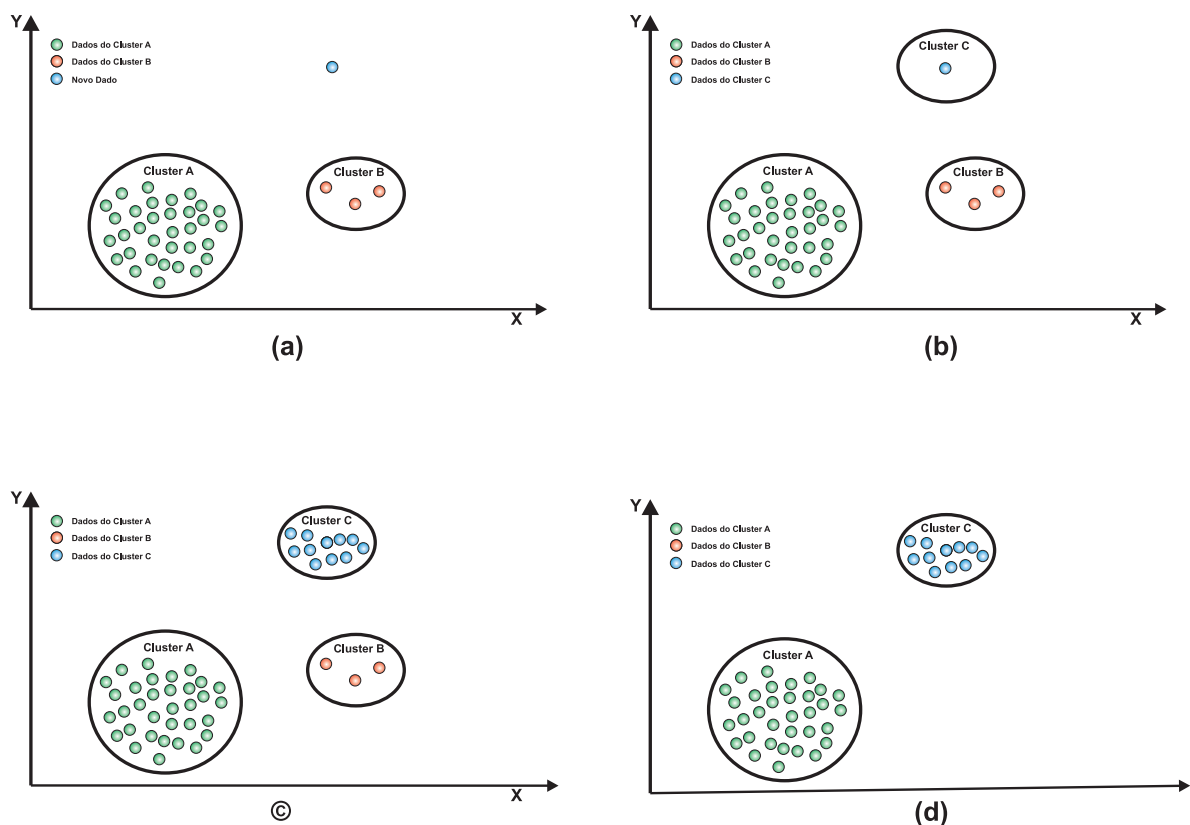
<sup>1</sup> Raízes do polinômio denominador da função de transferência (CHEN, 1999).

<sup>2</sup> Raízes do polinômio numerador da função de transferência (CHEN, 1999).

sendo que  $R$  é quantidade de regras *fuzzy*,  $p$  é a quantidade de termos linguísticos e  $n$  é quantidade de entradas.

De forma alternativa, o espaço de dados pode ser particionado através do agrupamento ou *clustering*, técnica muito conhecida na área de aprendizagem de máquinas e mineração de dados (AGGARWAL; REDDY, 2014) (KHALEGHI et al., 2016). Ela se caracteriza por ser uma técnica de aprendizado não-supervisionado a partir dos dados, isto é, descobre por si só e sem intervenção humana relações, padrões, regularidades ou categorias nos dados que lhe vão sendo apresentados. O principal objetivo do agrupamento é encontrar a melhor maneira de subdividir o espaço de dados em regiões ou grupos (*clusters*) que possuem dados com características similares. Por outro lado, diferentes grupos devem apresentar dados com características bem diferentes entre si. Cada grupo ou *cluster* possui um submodelo associado que gera uma saída local que contribui para a resposta global do sistema. Cada submodelo é gerado para modelar o comportamento dos dados pertencentes ao grupo que está relacionado e geralmente tem característica linear (AGGARWAL; REDDY, 2014) (ANGELOV, 2013).

Figura 8 – Princípio de Criação e Remoção de *Clusters* no Agrupamento Evolutivo.



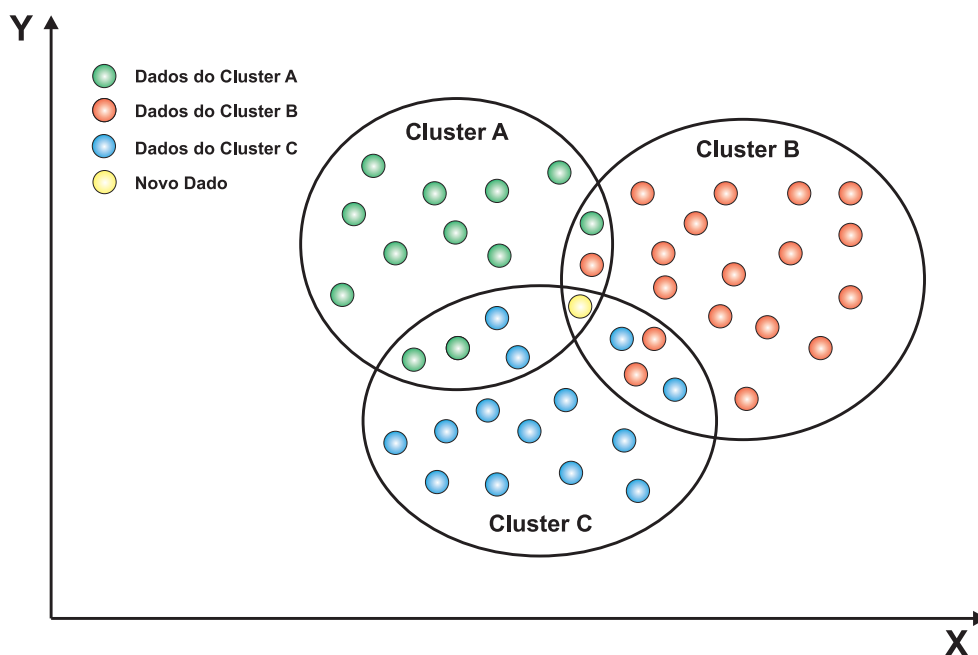
Fonte: Adaptado de Rocha e Serra (2016).

No agrupamento evolutivo, a quantidade desses *clusters* pode variar, sendo possível aumentar ou diminuir conforme o comportamento dos dados que chegam a cada instante. A Figura 8 esclarece os princípios da geração e remoção de *clusters*. Inicialmente, considera-se que há dois *clusters* no espaço bidimensional que englobam os dados na Figura 8.(a). Nessa

figura, um novo dado chega e ainda não está agregado a nenhum *cluster*, pois está presente em uma região inexplorada pelo agrupamento. Já na Figura 8.(b), foi realizada uma avaliação para verificar se esse novo dado tem representatividade suficiente para que seja gerado um novo *cluster*. Como esse dado está em uma região inexplorada, ele satisfaz esse requisito, sendo necessária a criação de um *cluster*. Por sua vez, a Figura 8.(c) mostra que novos dados foram agregados ao *cluster* C gerado recentemente. Por fim, o algoritmo de agrupamento evolutivo monitora a qualidade dos *clusters* existentes. Note que o *cluster* B possui poucos dados associados, sendo assim, ele foi removido do espaço de dados devido a sua baixa qualidade, conforme Figura 8.(d).

As técnicas de agrupamento tradicionais visam formar *clusters* que são mutuamente exclusivos (AGGARWAL; REDDY, 2014) (KHALEGHI et al., 2016). Contudo, no contexto dos sistemas evolutivos, a saída global do sistema em questão é calculada por uma soma ponderada das saídas locais de cada submodelo ou grupo, sendo que esses grupos podem estar se sobrepondo entre si. Devido a essa característica, a teoria de sistemas *fuzzy* descreve de forma apropriada a transição dos dados de uma região para outra, através da medida fornecida pelos graus de pertinência (ANGELOV, 2013) (ANGELOV, 2010) (LUGHOFER, 2011). Por exemplo, seja um espaço de dados representado na Figura 9 onde foram gerados 3 *clusters*: A, B e C. Sendo assim, uma mesma informação (em amarelo) pode pertencer a região A com grau de pertinência  $\mu_A$  e, ao mesmo tempo, a região B com  $\mu_B$  e a região C com  $\mu_C$ . Esse grau de pertinência é determinado de acordo com a proximidade com o centro do *cluster*, podendo variar seu valor no intervalo de 0 a 1. Quanto mais próximo do centro o dado estiver, maior será o grau de pertinência em relação a esse grupo. Outra característica a ser destacada é que a soma dessas pertinências deve ser unitária. Logo, para o exemplo citado acima, a soma de  $\mu_A + \mu_B + \mu_C$  é igual a 1.

Figura 9 – Espaço de Dados Agrupado por *Clusters* Sobrepostos.



Fonte: Autor.

Um das principais questões envolvendo o agrupamento é a determinação do número apropriado de *clusters* no espaço de dados. As técnicas mais conhecidas na literatura, como *k-means*, *fuzzy c-means* e Gustafson-Kessel, consideram um número que é prefixado pelo especialista antes do início do procedimento de agrupamento (BABUSKA, 1998). Essa indicação por parte do especialista não é interessante para os sistemas evolutivos que buscam ser o mais independente o possível dos especialistas. As técnicas de agrupamento para sistemas evolutivos devem ser capazes então de determinar a quantidade apropriada de *clusters* automaticamente a partir do comportamento dos dados que chegam a cada instante e usar o mínimo o possível de conhecimento por parte do especialista para definir restrições (ANGELOV, 2013) (ANGELOV, 2010).

Os objetivos fundamentais das técnicas de agrupamento são a determinação dos pontos focais ou centros dos *clusters* e a definição das fronteiras que limitam seu alcance no espaço. Os pontos focais não necessariamente são os centros geométricos dos *clusters*. Em relação a forma dos *clusters*, esta pode ser um hiper-cubo, uma hiper-esfera, uma hiper-elipse, entre outros formatos, sendo que as denominadas medidas de proximidade entre os dados determinam o formatos desses grupos (AGGARWAL; REDDY, 2014) (KHALEGHI et al., 2016). Os principais tipos de medidas de proximidade usadas são a Euclidiana e a Mahalonobis (ANGELOV, 2013). A distância Euclidiana entre um vetor de dados  $\mathbf{d}_k$  no espaço com  $n$  dimensões e o centro de um *cluster*  $i$  ( $\mathbf{f}_{i*}$ ) é definida por:

$$\xi^2 = \|\mathbf{d}_k - \mathbf{f}_{i*}\|^2 = (\mathbf{d}_k - \mathbf{f}_{i*})(\mathbf{d}_k - \mathbf{f}_{i*})^T. \quad (3.2)$$

sendo  $\mathbf{d}_k \in \mathbb{R}^{1 \times n}$  e  $\mathbf{f}_{i*} \in \mathbb{R}^{1 \times n}$ . Por sua vez, a distância de Mahalonobis entre esses mesmos dados é definida por:

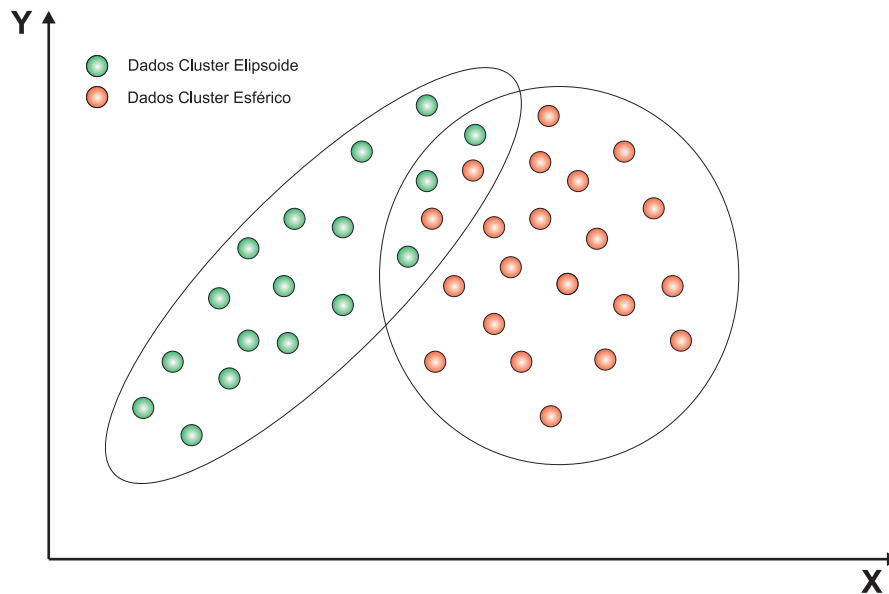
$$\xi^2 = (\mathbf{d}_k - \mathbf{f}_{i*})\mathbf{\Sigma}^{-1}(\mathbf{f}_{i*} - \mathbf{d}_k)^T, \quad (3.3)$$

onde  $\mathbf{\Sigma}$  é a matriz de covariância entre os dados desses vetores. A distância Euclidiana gera *clusters* esféricos, já a distância de Mahalonobis produz *clusters* com formato de elipse, pois diferencia as entradas do espaço de acordo com uma ponderação efetuada pelo inverso da matriz de covariância (Figura 10). Cabe ao especialista definir qual formato de grupo considerar de acordo com o propósito do sistema sendo projetado (ANGELOV, 2013).

Uma importante questão a ser levada em conta no agrupamento é a cobertura completa de todo o espaço de dados. Isso evita situações em que não há saídas locais factíveis para dados de entrada que não pertencem a nenhum *cluster*. Nesse caso, portanto, os submodelos não foram gerados para lidar com essas informações. Além disso, a cobertura completa do espaço evita problemas computacionais como divisões por números muito pequenos (ANGELOV, 2013) (KASABOV, 2007).

Para representar adequadamente a complexidade da interação entre as variáveis, todo o

Figura 10 – Formato dos Clusters Quanto a Distância de Mahalanobis (Elipse) e Euclidiana (Círculo).



Fonte: Adaptado de Angelov (2013).

espaço de entrada e de saída deve ser utilizado no agrupamento. Isso porque o objetivo do sistema é identificar através da técnica de agrupamento o mapeamento não-linear entre as variáveis de entrada e saída para formular as regras de um sistema *fuzzy* evolutivo. Caso o agrupamento seja realizado separadamente para entradas e saídas, o modelo resultante não representará esse mapeamento de forma correta (KASABOV, 2007).

Segundo Angelov (2013), o agrupamento evolutivo deve seguir as seguintes etapas para particionar o espaço de dados:

- Verificar o potencial descritivo de cada amostra de dados que chega;
  - Caso essa medida seja elevada, esses dados serão adotados como centros de novos *clusters*;
  - Caso contrário, esses dados são associados a algum grupo já existente.
- Se uma área ainda não coberta tiver dados significativos, gerar um *cluster* para cobrir essa região;
- Os *clusters* não devem:
  - Estar próximos de forma exagerada, onde um engloba o outro. Isso gera saídas redundantes;
  - Descrever dados muito antigos;
  - Apresentar baixa utilidade.



O algoritmo evolutivo deve ter bastante cautela com os chamados *outliers*, que são valores esporádicos e que podem fazer com que o sistema apresente erros. Sendo assim, uma amostra antes de ser um ponto focal é analisada para verificar se outros dados se aproximam dela para que seja necessária a criação de um novo *cluster*, evitando a geração destes para esses *outliers*. Se por ventura for criado um *cluster* para esses *outliers*, as medidas de qualidade vão remover esses *clusters* que não estão sendo utilizados, isto é, com baixa qualidade (ANGELOV, 2010).

A zona de influência ou raio dos *clusters* em direção a cada eixo pode ser atualizado recursivamente. Sua variação afeta bastante as saídas de cada submodelo, influenciando também nos resultados globais do sistema evolutivo. De acordo com Angelov (2013), a equação da atualização do raio  $\sigma_i^j(k)$  é:

$$\sigma_i^j(k) = \sqrt{\zeta[\sigma_i^j(k-1)]^2 + (1-\zeta)\frac{1}{S_k^i}(\hat{d}_k^j - \hat{f}_{i*}^j)} \quad (3.4)$$

sendo que  $\zeta$  é uma constante de aprendizado,  $S_k^i$  é o suporte<sup>3</sup> ou a quantidade de amostras do *cluster*  $i$ , e  $\hat{d}_k^j$  são os dados que chegaram no instante  $k$ .

### 3.2.2 Padronização e Normalização dos Dados

A normalização e a padronização são de suma importância a fim de que as entradas e saídas de diferentes grandezas e magnitudes sejam transformadas para que seus valores possam ser comparáveis. Ambas as técnicas são bem conhecidas para aplicação em batelada, sendo que todos os dados estão a disposição. Contudo, para sistemas evolutivos, o fluxo de dados chega a cada instante, sendo necessárias alternativas *online* para esses métodos (LUGHOFER, 2011) (KASABOV, 2007) (DUDA; HART; STORK, 2000).

Conforme debatido, os dados que chegam a todo instante geralmente são não-estacionários, com média e variância que se modificam ao longo do tempo. A técnica de padronização usa essas medidas estatísticas dos dados, assim, a atualização recursiva da média é (ANGELOV, 2010):

$$\bar{\mathbf{d}}_k = \frac{k-1}{k}\bar{\mathbf{d}}_{k-1} + \frac{1}{k}\mathbf{d}_k, \quad \bar{\mathbf{d}}_1 = \mathbf{d}_1, \quad (3.5)$$

onde  $\mathbf{d}_k = [\mathbf{x}^T \ \mathbf{y}^T]$  representa os dados das  $n$  entradas e das  $m$  saídas, para  $\mathbf{x}^T = [x_1 \ x_2 \ \dots \ x_n]$  e  $\mathbf{y}^T = [y_1 \ y_2 \ \dots \ y_m]$ . O desvio padrão pode ser atualizado conforme a seguir:

$$\delta_k^2 = \frac{k-1}{k}\delta_{k-1}^2 + \frac{1}{k}\|\mathbf{d}_k - \bar{\mathbf{d}}_k\|^2, \quad \delta_1^2 = 1. \quad (3.6)$$

<sup>3</sup> Conceito da teoria sobre os métodos de agrupamento que indica a quantidade de amostras que estão associadas a um *cluster* (ANGELOV, 2010)

A padronização dessa amostra é baseada na subtração da média e pelo desvio padrão calculados acima, conforme a seguir (DUDA; HART; STORK, 2000):

$$\hat{\mathbf{d}}_{\mathbf{k}} = \frac{\mathbf{d}_{\mathbf{k}} - \bar{\mathbf{d}}_{\mathbf{k}}}{\delta_{\mathbf{k}}}, \quad (3.7)$$

onde  $\hat{\mathbf{d}}_{\mathbf{k}}$  representa a amostra padronizada. Para retornar a escala original, a seguinte transformação é necessária (ANGELOV, 2010):

$$\mathbf{d}_{\mathbf{k}} = \hat{\mathbf{d}}_{\mathbf{k}}\delta_{\mathbf{k}} + \bar{\mathbf{d}}_{\mathbf{k}}. \quad (3.8)$$

Por sua vez, a normalização é baseada na atualização dos mínimos e máximos por variável de entrada quando uma amostra dos dados excede um desses limites. O procedimento de normalização apenas inicia a partir da terceira amostra  $\mathbf{d}_{\mathbf{k}}$  ( $k = 3, 4, \dots$ ), pois as duas primeiras são consideradas como os limites necessários para iniciar o método (DUDA; HART; STORK, 2000). Os limites inferior e superior são definidos por:

$$\begin{aligned} \min &= \min_{q=1}^{k-1}(d_q^j) \\ \max &= \max_{q=1}^{k-1}(d_q^j). \end{aligned} \quad (3.9)$$

Em primeiro lugar, para cada entrada em  $\mathbf{d}_{\mathbf{k}}$  ( $d_k^1, d_k^2, \dots, d_k^n$ ) verifica-se caso a amostra ultrapasse os limites conforme a expressão a seguir (DUDA; HART; STORK, 2000):

$$\min \leq d_k^j \leq \max, \text{ para } j = 1, 2, \dots, n. \quad (3.10)$$

Se a Equação 3.10 for satisfeita, então a amostra será normalizada de acordo com a seguinte relação (DUDA; HART; STORK, 2000):

$$\hat{d}_k^j = \frac{d_k^j - \min}{\max - d_k^j}, \text{ para } j = 1, 2, \dots, n. \quad (3.11)$$

para  $\hat{d}_k^j$  sendo o valor normalizado para a  $j$ -ésima entrada. Caso a Equação 3.10 não seja satisfeita, então os limites devem atualizados de acordo com o dado que os excedeu. Para obter de volta o dado na escala original, basta utilizar a mesma equação acima, reformulando-a da seguinte forma:

$$d_k^j = \frac{\hat{d}_k^j \max + \min}{\hat{d}_k^j + 1}, \text{ para } j = 1, 2, \dots, n. \quad (3.12)$$

A padronização sempre atualiza a média e a variância para toda amostra de dados que chega, já a normalização apenas atualiza suas equações quando os novos dados ultrapassam um

dos limites definidos. No entanto, Duda, Hart e Stork (2000) recomendam o uso da padronização para sistemas evolutivos, pois a média e variância acumulam recursivamente as informações de todo o conjunto de dados e seus valores não mudam abruptamente.

### 3.2.3 Monitoramento da Qualidade dos *Clusters*

Afim de simplificar a estrutura do sistema evolutivo, os *clusters* que foram gerados no agrupamento podem ser monitorados quanto a sua qualidade. Logo, se eles apresentarem baixa qualidade, podem ser eliminados. O primeiro tipo de parâmetro de qualidade que será apresentado é a idade de um *cluster* ou submodelo (ANGELOV, 2010) (ANGELOV, 2013). A idade está relacionada com o tempo em que as amostras estão associadas com um *cluster*  $i$ . Sua fórmula é:

$$idade_k^i = k - \frac{\sum_{q=1}^{S_k^i} T_q}{S_k^i}, \text{ para } i = 1, 2, \dots, R, \quad (3.13)$$

sendo que  $S_k^i$  é o suporte do *cluster*  $i$  no instante  $k$  e  $T_q$  é o instante em que cada amostra foi inserida no grupo. O valor da idade pode variar de 0 até o instante  $k$  atual. Um *cluster* é dito antigo quando este não possui amostras associadas recentemente, em contrapartida, os *clusters* jovens são formados por dados recentes. O usuário pode optar por remover esses *clusters* antigos, sendo uma sugestão dada por Angelov (2013) a seguinte regra:

$$\begin{aligned} \text{SE } idade^i > \overline{idade^i} + \delta_{idade^i} \\ \text{ENTÃO } R = R - 1, \end{aligned} \quad (3.14)$$

para  $\overline{idade^i}$  sendo a média da variável  $idade^i$  considerando cada *cluster* e  $\delta_{idade^i}$  denota o desvio padrão da  $idade^i$ , com a possibilidade de ambos valores serem calculados recursivamente.

Outra medida muito comum para simplificação da estrutura dos sistemas é a utilidade dos *clusters* (ANGELOV, 2010) (ANGELOV, 2013). A utilidade  $\Omega_k^i$  indica o quão vem sendo utilizado um submodelo associado ao *cluster*  $i$ . Sua equação é denotada por:

$$\Omega_k^i = \frac{\sum_{q=1}^k \lambda_q}{k - T_{i*}} \quad (3.15)$$

em que  $T_{i*}$  corresponde ao instante em o *cluster*  $i$  foi gerado e  $\lambda_q$  é o grau de ativação ou ponderação do submodelo. Essa medida acumula essas ponderações para avaliar as contribuições dos submodelos nas respostas globais do sistema. Para remover *clusters* com baixa utilidade,

pode-se utilizar a seguinte regra segundo (ANGELOV, 2013):

$$\begin{aligned} & \text{SE } (\Omega^i < \varepsilon) \\ & \text{ENTÃO } (R = R - 1), \end{aligned} \quad (3.16)$$

sendo que  $\varepsilon$  geralmente é definido entre 0.03 e 0.1.

O método usado para monitoramento de *clusters* varia de acordo com os objetivos dos usuários e de cada aplicação. Os projetistas dos sistemas podem criar novas medidas de qualidade e regras para remover *clusters*, dessa forma, essa etapa é bem subjetiva.

### 3.2.4 Estimação dos Parâmetros dos Submodelos

Uma vez definidos os *clusters* para determinado instante, os sistemas evolutivos precisam estimar os parâmetros dos submodelos, efetuando o aprendizado supervisionado, já que é preciso ter disponível o valor da saída de cada submodelo ou saída global. Portanto, o aprendizado a partir dos dados para sistemas *fuzzy* evolutivos consiste na identificação da estrutura, no aprendizado não-supervisionado dos antecedentes e no aprendizado supervisionado dos parâmetros dos consequentes. Para um conjunto de submodelos, o objetivo é encontrar os parâmetros que minimizem os erros de previsão, classificação, controle ou de estimação, isto é, dependendo da aplicação (ANGELOV, 2010) (ANGELOV, 2013) (LUGHOFER, 2011) (KASABOV, 2007). Esses submodelos podem ter diversas formas de acordo com o problema, como funções polinomiais ou no formato de conjuntos *fuzzy*, por exemplo, de forma gaussiana, triangular ou trapezoidal. Contudo, sem perda de generalidade, vamos assumir que os submodelos são representados por equações lineares.

Seja  $\mathbf{y}_k$  um vetor com  $m$  saídas de um sistema no instante  $k$  dado pela seguinte expressão:

$$\mathbf{y}_k = \Psi_k^T \boldsymbol{\theta}_k \quad (3.17)$$

onde  $\mathbf{y}_k = [y_k^1 \ y_k^2 \ \dots \ y_k^m]^T$ ,  $\Psi_k = [\lambda_1 \mathbf{x}_k^T \ \lambda_2 \mathbf{x}_k^T \ \dots \ \lambda_R \mathbf{x}_k^T]^T$  representam as ponderações dadas ao vetor com  $n$  entradas denotado por  $\mathbf{x}_k = [1 \ x_k^1 \ x_k^2 \ \dots \ x_k^n]^T$ ,  $\boldsymbol{\theta}_k = [(\gamma_{k1})^T \ (\gamma_{k2})^T \ \dots \ (\gamma_{kR})^T]^T$  corresponde ao vetor de parâmetros da base de regras e  $\gamma_{ki}$  denota a matriz que contém os parâmetros dos consequentes relacionados com os  $R$  submodelos lineares, ou seja:

$$\gamma_i = \begin{bmatrix} a_i^{01} & \dots & a_i^{0m} \\ \dots & \dots & \dots \\ a_i^{n1} & \dots & a_i^{nm} \end{bmatrix}, \text{ para } i = 1, 2, \dots, R. \quad (3.18)$$

Para sistemas *offline*, técnicas para estimar seus parâmetros são facilmente aplicáveis, como o método dos mínimos quadrados. Porém, sistemas evolutivos não possuem estrutura fixa, sendo que seus submodelos podem ser removidos e novos submodelos podem ser gerados. Sendo assim, a aplicação direta dos mínimos quadrados convencional não é válida (ANGELOV, 2010)

(ANGELOV, 2013). A estimação dos parâmetros dos submodelos pode ser realizada de forma global ou local. Para determinados pares de vetores entrada e saída  $(\mathbf{x}_q^T, \mathbf{y}_q)$ , para  $q = 1, 2, \dots, k$ , a função custo global que precisa ser minimizada é representada por:

$$E^G = \sum_{q=1}^k (\mathbf{y}_q - \Psi_q^T \boldsymbol{\theta}_q)^2 \quad (3.19)$$

ou de forma matricial:

$$E^G = (\mathbf{Y} - \Psi^T \boldsymbol{\Theta})^T (\mathbf{Y} - \Psi^T \boldsymbol{\Theta}) \quad (3.20)$$

Dessa forma, a expressão que obtém os valores dos parâmetros tal que a função custo global seja mínima é:

$$\boldsymbol{\Theta} = (\Psi^T \Psi)^{-1} \Psi^T \mathbf{Y}. \quad (3.21)$$

A minimização das Equações 3.19 e 3.20 garantem o melhor desempenho global para os sistema, no entanto, não garantem o melhor desempenho para os submodelos locais e apresentam um menor nível de transparência para a base de regras do sistema. O procedimento visa obter a melhor aproximação global para os dados, enquanto o local visa alcançar a melhor aproximação dos dados referentes a cada submodelo definido por um *cluster* (YEN; WANG; GILLESPIE, 1998). A função custo local para um submodelo  $i = 1, 2, \dots, R$  que deve ser minimizada é ponderada e denotada por:

$$E_i^L = \sum_{q=1}^k \lambda_q^i (\mathbf{x}_q) (\mathbf{y}_q - \mathbf{x}_q^T \boldsymbol{\gamma}_q^i)^2 \quad (3.22)$$

É importante destacar que o critério local é dado para cada uma das  $R$  regras. Para obtermos o critério global a partir da equação acima, fazemos:

$$E^L = \sum_{i=1}^R E_i^L. \quad (3.23)$$

Representando o critério local de forma matricial:

$$E_i^L = (\mathbf{Y} - \mathbf{X}^T \Gamma_i)^T \Lambda^i (\mathbf{Y} - \mathbf{X}^T \Gamma_i), \text{ para } i = 1, 2, \dots, R, \quad (3.24)$$

sendo que a matriz  $\Lambda^i$  é diagonal e conta com as ponderações  $\lambda^i$  dos modelos,  $\mathbf{X}^T$  é a matriz com as entradas e  $\Gamma_i$  é uma matriz composta por valores instantâneos de  $\mathbf{x}$  e  $\boldsymbol{\gamma}_i$ .

Em geral, o conjunto de parâmetros que satisfazem o critério global difere bastante dos parâmetros encontrados localmente. Em relação a questão evolutiva, a mudança da estrutura

afeta muito menos o procedimento local. As dimensões das matrizes de covariância de cada submodelo local não são afetadas quando novos submodelos são gerados. Isto é, para cada submodelo, há uma função custo com sua própria matriz de covariância. Por outro lado, ao adicionar ou remover submodelos, a matriz de covariância global precisa ter suas dimensões modificadas, já que existe apenas uma matriz e esta precisa agregar ou retirar informações dos submodelos gerados ou removidos (ANGELOV, 2010) (ANGELOV, 2013).

Sendo assim, o procedimento local é mais apropriado na maioria dos casos envolvendo sistemas evolutivos, já que são mais simples, possuem matrizes de covariância com dimensões menores, são mais estáveis e mais robustos (YEN; WANG; GILLESPIE, 1998).

### 3.3 Aspectos do Estado da Arte para Sistemas Evolutivos

Conforme já citado acima, os conceitos pioneiros sobre sistemas evolutivos foram introduzidos nas pesquisas de Angelov (2002), de Kasabov (2002) e de Angelov e Kasabov (2005). Frequentemente, os sistemas evolutivos são estruturados com modelos *fuzzy*, redes neurais e redes *neuro-fuzzy*.

Grande parte dos sistemas *fuzzy* nessa área de pesquisa é baseada em regras do tipo Takagi-Sugeno apresentadas no Apêndice A. A quantidade de regras para esse tipo de sistema pode aumentar ou diminuir, utilizando conceitos de aprendizado não-supervisionado para formular os antecedentes das regras *fuzzy*. Em relação aos consequentes, geralmente utilizam-se algoritmos baseados em mínimos quadrados para estimar seus parâmetros. Em Angelov e Filev (2004), o modelo *online* que utiliza esses conceitos acima chamado de *evolving Takagi-Sugeno* (eTS) foi proposto. Sua natureza adaptativa combinada com a transparência da representação por regras *fuzzy* Takagi-Sugeno são características tornam este modelo muito utilizado nesse campo de pesquisa para a modelagem *online* de processos complexos, dessa maneira, muitas abordagens baseadas nessa versão têm sido propostas.

A principal limitação do eTS é que ele não é capaz de remover regras que se tornam irrelevantes durante a aprendizagem. Para superar essa desvantagem, Angelov e Filev (2005) propuseram uma versão do eTS denominada de SimpLeTS que apresenta essa capacidade de simplificar a base de regras *fuzzy*. Essa simplificação se fundamenta na representatividade das regras, isto é, se o número de amostras dos dados associadas a uma regra for menor que um determinado percentual dos dados, essa regra é ignorada e removida.

Outra versão do eTS foi desenvolvida na pesquisa de Angelov e Zhou (2006), sendo denominada de xTS (*eXtended Takagi-Sugeno*). Essa nova abordagem traz como novidades a atualização recursiva do raio de cada *cluster* e a introdução do conceito de idade dos *clusters*. Já na pesquisa de Angelov (2010), uma versão do eTS original robusta, flexível e autônoma com o nome de eTS+ foi introduzida, cujos os parâmetros do antecedente e a estrutura da base de regras são atualizados usando critérios como idade, utilidade, densidade local e zona de influência. As

grandes vantagens do eTS+ é que seu algoritmo pode inicializar sem haver nenhuma regra *fuzzy* e não são necessários parâmetros especificados pelo usuário. A metodologia desenvolvida nesse trabalho se baseia no eTS+.

O algoritmo FLEXFIS (*Flexible Fuzzy Inference System* - Sistema de Inferência *Fuzzy* Flexível), descrito em Lughofer (2008), considera um algoritmo de agrupamento recursivo e realiza a estimação de parâmetros dos consequentes através dos mínimos quadrados recursivos ponderados. A principal aplicação desse algoritmo é na pesquisa sobre o controle e previsão de emissões de poluentes de motores de automóveis de combustão interna (LUGHOFER et al., 2011). Uma versão do FLEXFIS apresentada em Lughofer (2012) é denominada de FLEXFIS++, cujas novidades são a redução da complexidade na adição e exclusão de regras *fuzzy* e sua maior interpretabilidade.

Em Rong et al. (2006), apresentou-se o algoritmo SAFIS (*Sequential Adaptive Fuzzy Inference System* - Sistema de Inferência *Fuzzy* Adaptativo e Sequencial), que utiliza um método baseado em uma rede neural com função de base radial para construir a base de regras *fuzzy*, sendo possível remover ou adicionar regras. Para atualizar os parâmetros do consequente, uma modificação do Filtro de Kalman é aplicada.

Na pesquisa de Lima et al. (2010), foi apresentado o conceito de aprendizado participativo, em que o conhecimento já adquirido pelo sistema é aplicado durante o aprendizado. Dessa forma, uma conexão deve ser realizada no aprendizado participativo entre as informações coletadas a todo instante e o conhecimento atual, sendo que as informações relevantes devem atualizar o conhecimento do sistema no formato de regras *fuzzy*. A técnica de aprendizagem participativa é aplicada na atualização dos antecedentes das regras e o algoritmo de mínimos quadrados recursivos é usado na atualização dos parâmetros dos consequentes. Baseando-se no aprendizado participativo, um sistema *fuzzy* evolutivo com o nome de eMG (*Multivariable Gaussian Evolving Fuzzy System* - Sistema *Fuzzy* Evolutivo Multivariável Gaussiano) foi apresentado no trabalho de Lemos, Caminhas e Gomide (2011b). No eMG, cada *cluster* é representado por uma função Gaussiana do tipo multivariável, sendo que os parâmetros dos *clusters* são atualizados recursivamente a cada instante.

Além de modelos *fuzzy*, alguns sistemas evolutivos têm como base outras técnicas de modelagem. Em Lemos, Caminhas e Gomide (2011a), considerou-se uma árvore de regressão linear recursiva no desenvolvimento da técnica eFT (*evolving Fuzzy Linear Regression Trees* - Árvores de Regressão Lineares *Fuzzy* Evolutivas). A topologia da eFT é atualizada recursivamente por meio de testes estatísticos, substituindo as folhas por sub-árvores que melhoram a qualidade do modelo. O algoritmo de aprendizado do eFT verifica se a combinação ponderada dos modelos de regressão linear locais apresentam desempenho superior ao do modelo linear global inicial. Além disso, o eFT captura mudanças nas informações de entrada e fornece modelos linguisticamente interpretáveis. Outra abordagem evolutiva que considera árvores de decisão é proposta em Shaker, Senge e Hulermeier (2013) com o nome de eFPT (*evolving Fuzzy Pattern Tree* - Árvores de

Padrões *Fuzzy* Evolutivas). A ideia principal dessa metodologia é manter um conjunto de árvores que podem substituir o modelo atual caso o seu desempenho seja inferior ao de uma dessas árvores alternativas. Portanto, o desempenho dessas árvores deve ser monitorado continuamente pela eFPT.

As redes *neuro-fuzzy* também são bastante exploradas como modelo para sistemas evolutivos. A primeira pesquisa envolvendo redes *neuro-fuzzy* evolutivas encontra-se no trabalho de Kasabov e Song (1999). Esse artigo apresenta duas redes *neuro-fuzzy*, sendo uma com regras do tipo Mamdani e a outra do tipo Takagi-Sugeno. Além disso, o antecedente das regras é obtido por meio da técnica tradicional de agrupamento *fuzzy C-means*. Baseando-se nessa teoria, Kasabov e Song (2002) desenvolveram um novo modelo *neuro-fuzzy* denominado de DENFIS (*Dynamic Evolving Neuro-Fuzzy Inference System* - Sistema de Inferência Neuro-Fuzzy Dinâmico Evolutivo), muito conhecido na área de sistemas evolutivos para aprendizado *online* e *offline*. Para adaptar sua estrutura composta por base de regras, o DENFIS usa um método de agrupamento evolutivo tanto para versão *online* quanto para a *offline*. Os parâmetros do consequente das regras são atualizados recursivamente pelo método dos mínimos quadrados.

Uma rede *neuro-fuzzy* com característica recorrente é proposta em Lin, Chang e Lin (2013). A estrutura recorrente dessa rede é constituída por laços de repetição externos e por uma realimentação interna que influenciam no grau de ativação de suas regras. Essas regras são geradas pelo método de agrupamento evolutivo e os parâmetros do consequente são atualizados por um algoritmo baseado no Filtro de Kalman. O algoritmo do gradiente descendente é aplicado para ajustar os antecedentes das regras e os parâmetros de recorrência da rede.

Outrossim, uma rede *neuro-fuzzy* evolutiva que utiliza o modelo *fuzzy* Tipo-2 é proposta em Tung, Quek e Guan (2013), com o desenvolvimento do método eT2FIS (*evolving Type-2 Neural Fuzzy Inference System* - Sistema de Inferência Neuro-Fuzzy Tipo-2 Evolutivo). Esse método é aplicado em um problema de identificação *online* para um sistema variante no tempo, no monitoramento online dos preços de ações financeiras e na modelagem da densidade do fluxo de tráfego de uma estrada. Outra pesquisa que considera os fundamentos sobre os modelos *fuzzy* Tipo-2 está detalhada em Lin, C. e Lin (2014). As proposições antecedentes das regras *fuzzy* são determinadas usando conjuntos *fuzzy* do Tipo-2 e o consequente é do tipo Takagi-Sugeno. Nesse rede, utiliza-se um algoritmo de agrupamento *online* e *fuzzy* do Tipo-2 que é atualizado pelo algoritmo do gradiente descendente, enquanto que os parâmetros dos consequentes das regras são adaptados através do filtro de Kalman. Em Rocha e Serra (2016), propõe-se um algoritmo de identificação *online* para a modelagem *neuro-fuzzy* evolutiva baseada em variáveis instrumentais aplicada a sistemas dinâmicos em ambientes ruidosos. Os autores analisam o desempenho do algoritmo proposto com base na modelagem em caixa-preta de dados de um helicóptero.

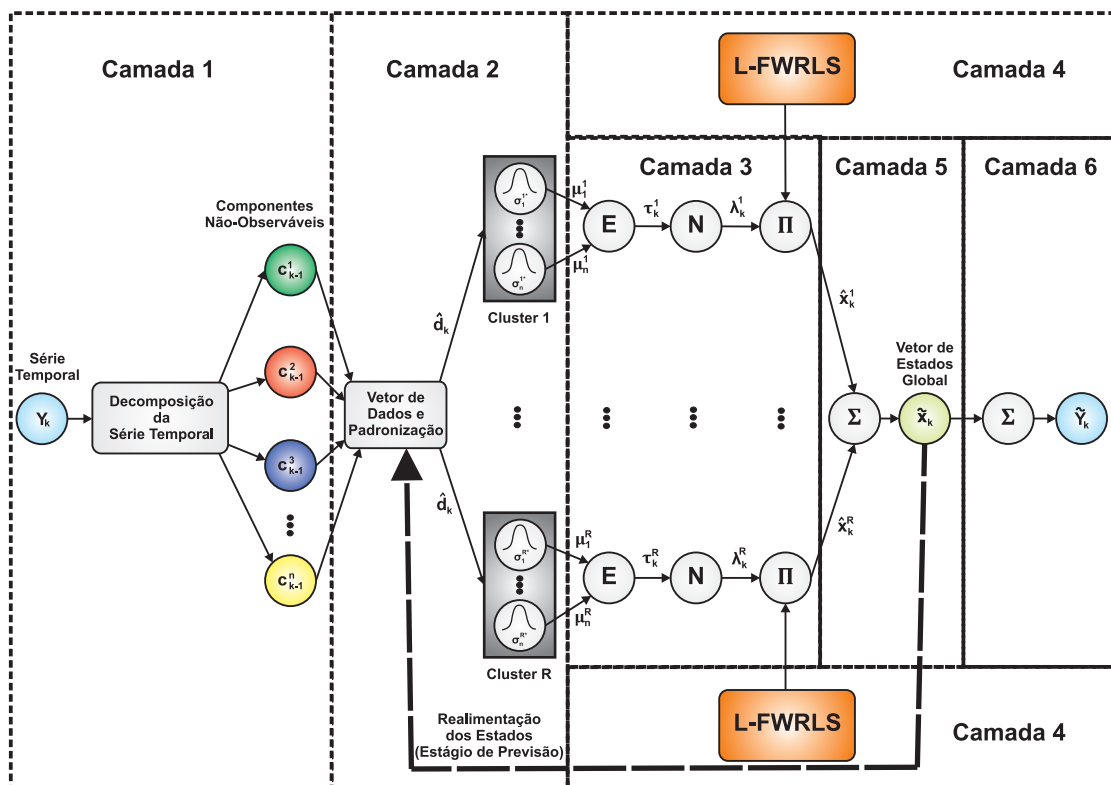


## 4 METODOLOGIA EVOLUTIVA PROPOSTA PARA PREVISÃO DE SÉRIES TEMPORAIS

### 4.1 Descrição Geral da Metodologia Proposta

A Figura 11 representa a arquitetura da metodologia desenvolvida nessa pesquisa para previsão de séries temporais sazonais. A metodologia proposta está estruturada como uma Rede *Neuro-Fuzzy* Takagi-Sugeno (RNF-TS). Essa RNF-TS é capaz de prever observações de uma série temporal um passo a frente sucessivas vezes. Portanto, diz-se que a RNF-TS realiza a previsão quantitativa múltiplos passos a frente. Por exemplo, se o objetivo é prever dados seis passos a frente, a RNF-TS proposta então irá executar seis previsões de um passo a frente. Além disso, a RNF-TS visa identificar e decompor as componentes não-observáveis da série temporal considerada. Os próprios dados dessas componentes extraídas da série são utilizados para realizar a previsão da série temporal. Sendo assim, essa abordagem considera a estratégia de dividir para conquistar, que decompõe uma série temporal complexa em um conjunto de padrões mais fáceis de lidar.

Figura 11 – Estrutura da Rede *Neuro-Fuzzy* Takagi-Sugeno Proposta.



Fonte: Autor.

Outrossim, a RNF-TS também usa os dados dessas componentes para adaptar e evoluir sua estrutura. Portanto, a metodologia proposta se caracteriza por ser adaptativa, evolutiva e *on-*

*line*. Ela é adaptativa porque seus parâmetros são atualizados a cada instante. A sua característica evolutiva advém da mudança da estrutura do modelo no formato de regras *fuzzy*, cuja quantidade pode aumentar ou diminuir a cada instante, dependendo do comportamento das componentes não-observáveis. Por fim, a RNF-TS opera de forma *online*, não sendo necessário valer-se de toda a informação histórica da série temporal para evoluir a sua estrutura e realizar a previsão.

As proposições consequentes das regras *fuzzy* da RNF-TS são constituídas por modelos no espaço de estados, sendo que os estados correspondem às próprias componentes não-observáveis extraídas. Essa abordagem no espaço de estados é apropriada para representar o comportamento dinâmico das componentes, pois os consequentes da RNF-TS devem operar de forma multivariável, conforme será analisado na Seção 4.2.4. Diante disso, a RNF-TS realiza a previsão de cada componente separadamente e, posteriormente, reconstrói a série temporal no espaço original, o que mais uma vez segue o princípio do dividir para conquistar que orienta essa pesquisa. A RNF-TS desenvolvida nesse trabalho subdivide-se em 6 camadas:

- **Camada 1:** Decomposição da Série Temporal;
- **Camada 2:** Agrupamento Evolutivo;
- **Camada 3:** Agregação;
- **Camada 4:** Estimação das Matrizes de Estado;
- **Camada 5:** Inferência da RNF-TS;
- **Camada 6:** Reconstrução da Série Temporal.

Essas camadas realizam suas funções específicas para cada amostra  $y_k$  da série temporal. Em outras palavras, a RNF-TS executa de forma cíclica, voltando a camada inicial de decomposição que aguarda a nova amostra da série após o fim do ciclo que ocorre na Camada 6. Antes de iniciar a execução da RNF-TS, o usuário deve estabelecer a quantidade de passos a frente que deseja prever, ou seja, o horizonte de previsão  $h$ .

A RNF-TS possui dois estágios de operação: o estágio de treinamento e o estágio de previsão. No primeiro estágio, a observação  $y_k$  da série temporal está disponível e é conhecida. Logo, a RNF-TS é treinada e evolui para realizar o mapeamento entre as componentes não-observáveis no instante  $k - 1$  e a previsão um passo a frente no instante  $k$ . Por outro lado, quando não há amostras da série temporal disponíveis, o estágio de previsão é acionado, no qual é possível prever observações da série com múltiplos passos a frente. No estágio de previsão, a RNF-TS não evolui sua estrutura, mantendo-a fixa.

De forma geral, quando uma amostra  $y_k$  é recebida pela RNF-TS no estágio de treinamento, as componentes não-observáveis da série temporal são identificadas na Camada de

Decomposição. Na segunda camada, a técnica de agrupamento é considerada para extrair conhecimento a partir dos dados dessas componentes, gerando grupos ou *clusters* no espaço que estão diretamente relacionados com as regras *fuzzy* da RNF-TS. A quantidade de regras *fuzzy* ou *clusters* pode crescer ou ser reduzida, de acordo com o comportamento dos dados considerados. Na Camada de Agregação, o grau de influência de cada regra *fuzzy* na saída da RNF-TS, isto é, na previsão da série, é determinado. Conforme citado, as proposições consequentes das regras *fuzzy* nessa metodologia proposta são modelos no espaço de estados. As matrizes dessas equações de estado são estimadas pela quarta camada, a de Estimação das Matrizes de Estado. A previsão um passo a frente de cada componente não-observável separadamente é calculada na quinta camada. Por fim, a sexta e última camada reconstrói a série temporal original, usando os previsões das componentes na etapa anterior para obter a previsão um passo a frente. A descrição do estágio de treinamento é apresentada na Seção 4.2.

Sob outra perspectiva, a RNF-TS no estágio de previsão não evolui nem atualiza os consequentes das regras *fuzzy*, ou seja, mantém sua estrutura fixa. Dessa maneira, algumas das camadas não são executadas. A Camada de Decomposição não pode executar porque não há  $y_k$  disponível para extrair as componentes não-observáveis. Para obter os valores das componentes que são essenciais para a operação da metodologia, os estados provenientes das equações de estado são realimentados para que a RNF-TS prossiga sua execução. Além da Camada de Decomposição, a Camada de Estimação das Matrizes de Estado também não opera, já que não há dados disponíveis da série para atualizar os parâmetros dos consequentes. A Seção 4.3 apresenta com mais detalhes essa modificação na operação da RNF-TS.

## 4.2 Estágio de Treinamento

### 4.2.1 Camada de Decomposição da Série Temporal pelo Método AESR

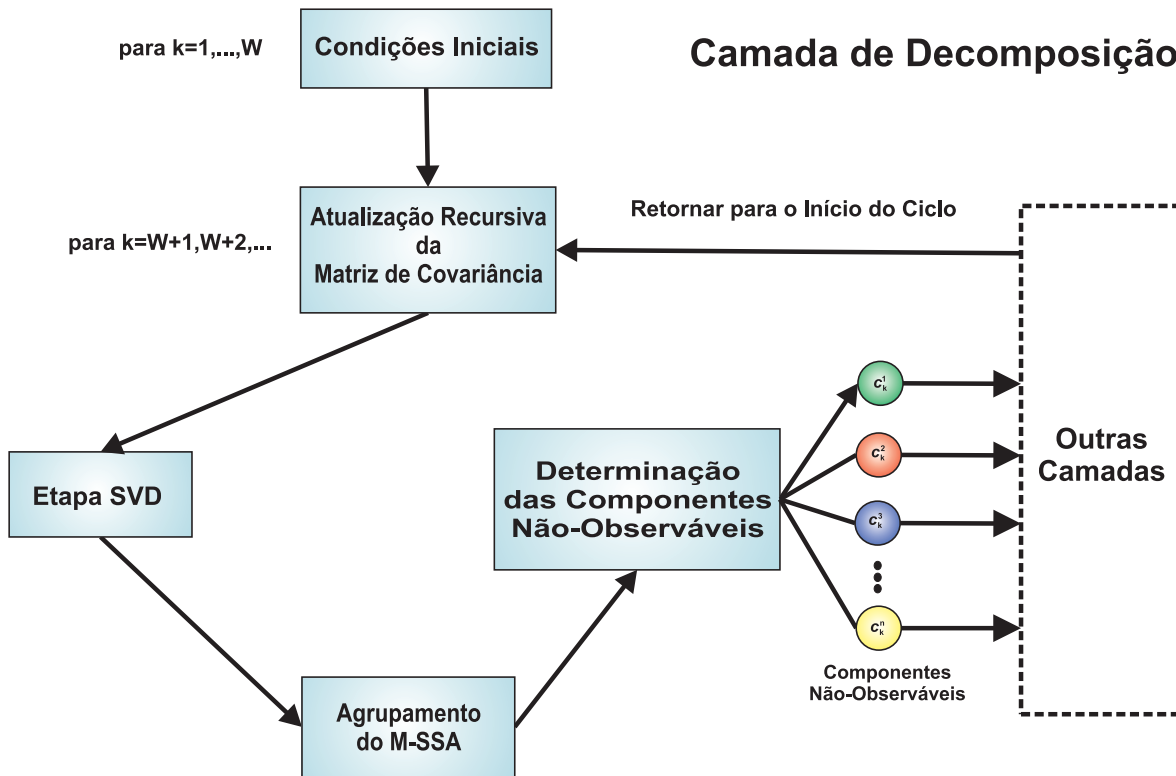
O principal propósito dessa camada é extrair as componentes não-observáveis ou padrões ocultos contidos nos dados da série temporal. Esses padrões apresentam comportamento mais simples e previsível que a série original, além de serem bastante úteis para análise por parte dos especialistas (BROCKWELL; DAVIS, 2002).

Como já apresentado, vários métodos podem ser aplicados para realizar a decomposição de séries temporais. Na metodologia desenvolvida nessa pesquisa, uma abordagem baseada no método AES, apresentado na Seção 2.5.2, é proposta para decomposição nessa camada. A grande vantagem do AES é que ele não está restrito a componentes pré-determinadas, como tendência e sazonalidade. Sendo assim, as componentes são determinadas somente a partir dos dados da série temporal. Além disso, o usuário da RNF-TS proposta pode selecionar a quantidade de componentes que deseja identificar, diferente de outras abordagens clássicas.

O algoritmo do AES tradicional se caracteriza por executar em batelada, levando em conta todos os dados da série temporal. Porém, a RNF-TS requer uma solução recursiva para

a camada de decomposição. Dessa maneira, o algoritmo Análise Espectral Singular Recursivo (AESR), do inglês *Recursive Singular Spectral Analysis* (RSSA) é proposto nesse trabalho para realizar uma atualização recursiva da matriz de covariância e extrair os dados das componentes não-observáveis da série temporal em cada instante  $k$ . A operação dessa abordagem alternativa do AES é resumida na Figura 12, além de ser descrita detalhadamente nas subseções a seguir.

Figura 12 – Ciclo de Execução do Algoritmo AESR.



Fonte: Autor.

#### 4.2.1.1 Condições Iniciais do Algoritmo AESR

Inicialmente, o usuário da RNF-TS proposta precisa informar valores para três parâmetros, que são:

- $W$ : quantidade de amostras da série temporal para as condições iniciais;
- $L$ : dimensão dos vetores  $\mathbf{m}_q$  (Equação 4.2), para  $1 < L < \frac{W}{2}$ ;
- $n$ : número de componentes não-observáveis extraídas.

Esses parâmetros são fatores críticos que influenciam no desempenho geral da RNF-TS e precisam ser apropriadamente selecionados. Antes de efetivamente iniciar o algoritmo AESR e a evolução da RNF-TS, um procedimento prévio em batelada é necessário. Contudo, diferente do AES original que inclui todos os dados históricos da série temporal, são necessárias apenas

$W$  de suas amostras nessa fase. Logo, seja  $y_k$  a observação no instante  $k$  de uma série temporal. Um intervalo denotado por  $\Phi$  com  $W$  amostras da série temporal é representado por:

$$\Phi = \{y_k \in \mathbb{R} | k = 1, 2, \dots, W\}. \quad (4.1)$$

Em seguida, é gerada uma quantidade  $K_W$  de vetores denominados  $\mathbf{m}_q$  com dimensão  $L$ , utilizando os dados no intervalo  $\Phi$  definido acima na Equação 4.1:

$$\begin{aligned} \mathbf{m}_q &= [y_q \ \dots \ y_{q+L-1}]^T, \text{ para } \mathbf{m}_q \in \mathbb{R}^{L \times 1}, \\ &\text{sendo} \\ K_W &= W - L + 1 \text{ e } q = 1, \dots, K_W. \end{aligned} \quad (4.2)$$

onde  $K_W$  está associado com a última amostra da série temporal contida no intervalo  $\Phi$ , ou seja,  $y_W$ . Essa operação acima é análoga a Equação 2.32 do AES original. Genericamente, o termo  $K_k$  está associado com a observação da série no instante  $k$ . O valor de  $K_k$  é incrementado para todo  $k$  nas etapas posteriores do AESR.

Conforme apresentado anteriormente sobre o algoritmo AES original, a matriz trajetória  $\mathbf{M}$  deve ser formulada de acordo com a Equação 2.33 exibida. Entretanto, ela não considera todos os dados da série temporal no procedimento do AESR. Neste caso, a matriz  $\mathbf{M}$  possui dados apenas das observações contidas no intervalo  $\Phi$ . Portanto:

$$\mathbf{M} = [\mathbf{m}_1 \ \mathbf{m}_2 \ \dots \ \mathbf{m}_{K_W}], \ \mathbf{M} \in \mathbb{R}^{L \times K_W}, \quad (4.3)$$

em outros termos, a matriz trajetória é representada por:

$$\mathbf{M} = \begin{bmatrix} y_1 & y_2 & \dots & y_{K_W} \\ y_2 & y_3 & \dots & y_{K_W+1} \\ \vdots & \vdots & \vdots & \vdots \\ y_L & y_{L+1} & \dots & y_W \end{bmatrix}_{L \times K_W} \quad (4.4)$$

A matriz de covariância inicial, considerando a matriz trajetória  $\mathbf{M}$ , é calculada por:

$$\mathbf{S}_W = \mathbf{M}\mathbf{M}^T, \ \mathbf{S}_W \in \mathbb{R}^{L \times L}. \quad (4.5)$$

A função dessa fase é fornecer os dados necessários para iniciar o AESR (Figura 12). As etapas descritas a seguir, responsáveis pela extração das componentes não-observáveis, são repetidas para todos os instantes  $k$  subsequentes ( $k = W + 1, W + 2, \dots$ ).

#### 4.2.1.2 Atualização Recursiva da Matriz de Covariância

Para cada observação  $y_k$  da série que chega para a RNF-TS nos instantes  $k = W + 1, W + 2, \dots$ , o valor  $K_k$  é incrementado por:

$$K_k = K_{k-1} + 1 \quad \text{ou} \quad K_k = k - L + 1, \quad (4.6)$$

sendo que ambas operações produzem o mesmo resultado.

No método AES tradicional, todos os dados históricos da série temporal estão contidos na matriz trajetória  $\mathbf{M}$ . Com a chegada de novas amostras e a variação de  $K_k$  para cada instante, essa matriz precisaria ter sua dimensão modificada para incorporar os novos dados, conforme as suas dimensões na Equação 4.4. Essa condição não é interessante computacionalmente, já que seria construída uma matriz com dimensões que se elevariam a cada instante. Por esse motivo, a matriz trajetória no AESR tem a função apenas de inicializar a matriz de covariância da Equação 4.5, sendo que a informação histórica da série é armazenada de forma incremental nessa matriz de covariância  $\mathbf{S}_k$ , para  $k = W + 1, W + 2, \dots$ . A dimensão de  $\mathbf{S}_k$  é fixa ( $L \times L$ ) e, além disso, essa matriz é atualizada recursivamente para cada observação da série que chega. Com essa finalidade, o seguinte vetor de atualização é definido:

$$\mathbf{q}_k = [y_{K_k} \ y_{K_k+1} \ \dots \ y_k]^T, \quad \mathbf{q}_k \in \mathbb{R}^{L \times 1}. \quad (4.7)$$

Portanto:

$$\begin{aligned} \mathbf{S}_k &= \mathbf{S}_{k-1} + \mathbf{Q}_k, \quad \mathbf{S}_k \in \mathbb{R}^{L \times L}, \\ &\text{onde} \\ \mathbf{Q}_k &= \mathbf{q}_k \mathbf{q}_k^T, \quad \mathbf{Q}_k \in \mathbb{R}^{L \times L}, \end{aligned} \quad (4.8)$$

onde  $\mathbf{Q}_k$  é conhecida como matriz de atualização.

Com essa operação, evita-se armazenar toda a informação histórica da série temporal, sendo exigidos somente os dados necessários para formular o vetor de atualização  $\mathbf{q}_k$  (Equação 4.7).

#### 4.2.1.3 Decomposição em Valores Singulares - SVD

A etapa SVD nessa abordagem proposta inicia após a obtenção da matriz de covariância  $\mathbf{S}_k$  calculada na Equação (4.8). O procedimento dessa etapa é bem semelhante ao considerado no algoritmo AES tradicional. Porém, em vez de calcularmos o SVD da matriz trajetória, o objetivo nessa etapa do AESR é obter o resultado da decomposição em valores singulares somente para a amostra  $y_k$  disponível da série temporal.

Os autovalores apresentados na Equação 2.34 anteriormente passam a ser calculados em cada instante para  $y_k$ , ou seja:

$$\rho_k^1 \geq \rho_k^2 \geq \rho_k^3 \geq \dots \geq \rho_k^L \geq 0. \quad (4.9)$$

Da mesma maneira, o sistema de autovetores ortonormais associados a cada autovalor acima também são determinados para todo instante  $k$ , passando a ser representados por  $\mathbf{p}_k^1, \mathbf{p}_k^2, \mathbf{p}_k^3, \dots, \mathbf{p}_k^L$ . Assim como no AES tradicional, define-se  $d$  como a quantidade de autovalores não-nulos da Equação 4.9. Levando em consideração essas definições, a SVD da observação  $y_k$  é representada por:

$$y_k = \theta_k^1 + \theta_k^2 + \dots + \theta_k^d, \text{ para} \quad (4.10)$$

$$\theta_k^l = \tilde{p}_k^l \mathbf{q}_k^T \mathbf{p}_k^l, \quad q = 1, \dots, d.$$

onde  $\tilde{p}_k^l$  é o elemento na posição  $L$  do autovetor  $\mathbf{p}_k^l$ , ou seja, o elemento correspondente a sua última posição. Esse valor é necessário para calcular o SVD somente da observação  $y_k$  da série temporal.

#### 4.2.1.4 Agrupamento do AESR

A etapa de agrupamento do AESR segue os mesmos passos do AES tradicional, ambos com o objetivo de distribuir o conjunto de índices  $\{1, \dots, d\}$  em  $n$  subconjuntos disjuntos. O valor de  $n$  informado pelo usuário corresponde a quantidade de componentes não-observáveis extraídas da série temporal.

Portanto, os  $n$  grupos de índices  $I_1, I_2, \dots, I_n$  são formados considerando os termos  $\theta_k^q$  ( $q = 1, \dots, d$ ) na Equação 4.10, resultando na seguinte expressão:

$$y_k = \theta_k^{I_1} + \theta_k^{I_2} + \dots + \theta_k^{I_n}. \quad (4.11)$$

#### 4.2.1.5 Determinação das Componentes Não-observáveis

Por fim, a última etapa converte cada grupo gerado na Equação 4.11 em uma observação no instante  $k$  de uma componente não-observável. Logo, esse passo pode ser escrito como:

$$\begin{aligned} c_k^j &= \theta_k^{I_j}, \text{ para } j = 1, \dots, n, \\ y_k &= c_k^1 + c_k^2 + \dots + c_k^n, \end{aligned} \quad (4.12)$$

onde  $c_k^j$  para  $j = 1, \dots, n$  representa a componente não-observável  $j$  extraída da amostra da série temporal  $y_k$ . Observe que a soma dessas componentes é igual a série original. Outra característica importante das componentes não-observáveis é que elas são linearmente independentes entre si (GOLYANDINA; NEKRUTKIN; ZHIGLJAVSKY, 2001).

#### 4.2.1.6 Algoritmos da Camada de Decomposição da Série Temporal

Os algoritmos a seguir exibem o procedimento realizado nessa primeira camada. O Algoritmo 1 traz a execução das condições iniciais do AESR.

---

##### Algoritmo 1 Condições Iniciais do AESR

---

- 1: Indicar valores para  $W$ ,  $n$ , e  $L$ ;
  - 2: Formulação do intervalo inicial  $\Phi$  (Equação 4.1);
  - 3: Calcular  $K_W = W - L + 1$ ;
  - 4: **para**  $k \leftarrow 1$  **até**  $W$  **faça**
  - 5:     Obter os vetores  $\mathbf{m}_q$  (Equação 4.2);
  - 6: **fim para**
  - 7: Construir a Matriz Trajetória  $\mathbf{M}$  relacionada aos dados do intervalo (Equação 4.4);
  - 8: Determinar a Matriz de Covariância Inicial  $\mathbf{S}_W$  (Equação 4.5).
- 

O Algoritmo 2, por sua vez, inicia efetivamente a operação de extração das componentes não-observáveis, repetindo-se esse ciclo a cada instante, conforme Figura 12.

---

##### Algoritmo 2 Decomposição da Série Temporal pelo AESR

---

- 1: Calcular o valor de  $K_k$  (Equação 4.6);
  - 2: Formular o vetor de atualização  $\mathbf{q}_k$  (Equação 4.7);
  - 3: Atualizar a Matriz de Covariância  $\mathbf{S}_k$  (Equação 4.8);
  - 4: Obter autovalores e autovetores de  $\mathbf{S}_k$ ;
  - 5: Aplicar o SVD (Equação 4.10);
  - 6: Realizar o Agrupamento do AESR (Equação 4.11);
  - 7: Determinar as componentes não-observáveis  $c_k^j$  (Equação 4.12).
- 

#### 4.2.2 Camada de Agrupamento Evolutivo

O propósito dessa camada é formular as proposições antecedentes das regras *fuzzy* usando o conhecimento extraído a partir dos dados das componentes não-observáveis oriundos da Camada de Decomposição. Nessa camada, a quantidade de regras *fuzzy* pode aumentar ou ser reduzida, modificando a estrutura da RNF-TS. Essa recorrente modificação na estrutura de base de regras da RNF-TS caracteriza o comportamento evolutivo da metodologia proposta.

Essa camada é responsável por construir um vetor de dados  $\mathbf{d}_k$  para todo instante  $k$ . O vetor  $\mathbf{d}_k$  considera as  $n$  componentes não-observáveis da Camada de Decomposição:  $c_{k-1}^1, c_{k-1}^2, \dots, c_{k-1}^n$ . Assim,  $\mathbf{d}_k$  é estruturado da seguinte forma:

$$\mathbf{d}_k = [c_{k-1}^1, c_{k-1}^2, \dots, c_{k-1}^n, y_k], \mathbf{d}_k \in \mathbb{R}^{1 \times n+1}. \quad (4.13)$$

onde  $c_{k-1}^1, c_{k-1}^2, \dots, c_{k-1}^n$ , além de serem as componentes, são consideradas também como entradas do sistema de inferência da RNF-TS e  $y_k$  é a saída desejada usada no estágio de treinamento da rede. A observação  $y_k$  é fundamental no estágio de treinamento para evoluir e



adaptar a estrutura da RNF-TS. É importante notar que a RNF-TS é treinada para utilizar os dados das componentes no instante  $k - 1$  para prever a observação da série temporal original no instante  $k$ .

Após o vetor de dados ser formulado em um instante  $k$  específico, é necessário normalizar ou padronizar  $\mathbf{d}_k$ , porque dados de componentes não-observáveis distintas podem apresentar valores em escalas significativamente diferentes, dificultando a comparação entres eles. A normalização e a padronização são técnicas com o papel de transformar os dados de  $\mathbf{d}_k$  de tal forma que estes possam estar em intervalos com valores possíveis de comparação (ANGELOV, 2010) (ANGELOV, 2013). Neste trabalho, apenas a técnica de padronização é adotada, pois se apresentou como mais conveniente para a metodologia proposta, sendo que a média e o desvio padrão calculados recursivamente acumulam os efeitos de todos os dados a cada instante. A padronização para cada elemento ou coluna de  $\mathbf{d}_k$  ( $d_k^j$ ,  $j = 1, 2, \dots, n + 1$ ) pode ser escrita conforme a seguir:

$$\hat{d}_k^j = \frac{d_k^j - \bar{d}_k^j}{\delta_k^j}, \text{ para } j = 1, 2, \dots, n + 1 \text{ e } k = W + 2, W + 3, \dots, \quad (4.14)$$

sendo que  $\hat{d}_k^j$  corresponde ao valor padronizado do elemento  $d_k^j$  do vetor de dados. A média  $\bar{d}_k^j$  e o desvio padrão  $\delta_k^j$  em relação a  $\mathbf{d}_k$  são atualizados recursivamente por:

$$\begin{aligned} \bar{d}_k^j &= \frac{k-1}{k} \bar{d}_{k-1}^j + \frac{1}{k} d_k^j, \quad j = 1, 2, \dots, n + 1 \text{ e } k = W + 2, W + 3, \dots, \\ \delta_k^j &= \sqrt{\frac{k-1}{k} (\delta_{k-1}^j)^2 + \frac{1}{k} \|\mathbf{d}_k - \bar{\mathbf{d}}_k\|^2}, \quad j = 1, 2, \dots, n + 1 \text{ e } k = W + 2, W + 3, \dots. \end{aligned} \quad (4.15)$$

As condições iniciais para essa técnica de padronização dos dados são:

$$\begin{aligned} \bar{d}_k^j &= 0, \quad j = 1, 2, \dots, n + 1 \text{ e } k = W + 1, \\ \delta_k^j &= 1, \quad j = 1, 2, \dots, n + 1 \text{ e } k = W + 1, \\ \hat{d}_k^j &= 0, \quad j = 1, 2, \dots, n + 1 \text{ e } k = W + 1. \end{aligned} \quad (4.16)$$

onde  $W + 1$  é o primeiro instante após às condições iniciais do método AESR, isto é, corresponde ao início do ciclo do AESR. A partir de agora, todas as variáveis que estão com dados padronizados possuem o acento circunflexo.

Em seguida, a metodologia proposta usa uma técnica de aprendizado não-supervisionado para extrair conhecimento dos vetores de dados padronizados  $\hat{\mathbf{d}}_k$ . Essa obtenção de conhecimento é realizada através de um método de agrupamento (*clustering*) evolutivo baseado no algoritmo *eClustering+*, proposto por Angelov (2010). Esse método recursivo e evolutivo divide o espaço de dados do vetor  $\hat{\mathbf{d}}_k$  em grupos ou *clusters* que podem se sobrepor uns aos outros, sendo que cada um destes *clusters* estão diretamente associados a uma regra *fuzzy* e a um submodelo.

O algoritmo de agrupamento evolutivo nessa camada tem como objetivos:

- Formar *clusters* com elevada capacidade de generalização;
- Determinar a quantidade adequada de *clusters* automaticamente a partir do vetor de dados padronizado  $\hat{\mathbf{d}}_k$ ;
- Simplificar a base de regras *fuzzy* da RNF-TS, removendo *clusters* ou regras redundantes.

O critério utilizado para gerar novos *clusters* é baseado no conceito da Estimação Recursiva da Densidade, do inglês *Recursive Density Estimation* (RDE), que possui como fundamentos alguns aspectos sobre a teoria da probabilidade no Apêndice B (ANGELOV; FILEV, 2003) (ANGELOV, 2010) (ANGELOV, 2013). Um vetor de dados padronizado  $\hat{\mathbf{d}}_k$  que satisfaz a Condição A na Equação 4.17 tem alta capacidade de generalização ou apresenta informações sobre regiões inexploradas no espaço de dados. Portanto, neste caso, os elementos de  $\hat{\mathbf{d}}_k$  irão compor o centro ou ponto focal de um novo *cluster* no espaço  $n$ -dimensional. Porém, se essa condição não for satisfeita, os elementos de  $\hat{\mathbf{d}}_k$  são associados com o *cluster* existente mais próximo. O *cluster* mais próximo ao ponto  $\hat{\mathbf{d}}_k$  é aquele que sua regra *fuzzy* associada apresenta maior grau de ativação, atributo que será comentado na próxima camada. A Condição A é denotada por (ANGELOV, 2010):

$$D_k(\hat{\mathbf{d}}_k) > \max_{i=1}^R D_k(\hat{\mathbf{f}}_{i*}) \text{ OU } D_k(\hat{\mathbf{d}}_k) < \min_{i=1}^R D_k(\hat{\mathbf{f}}_{i*}) \quad (4.17)$$

onde  $R$  é o número de *clusters* ou regras *fuzzy*,  $D_k(\hat{\mathbf{d}}_k)$  é a densidade do vetor  $\hat{\mathbf{d}}_k$ ,  $\hat{\mathbf{f}}_{i*}$  corresponde ao ponto focal do  $i$ -ésimo *cluster*, e  $D_k(\hat{\mathbf{f}}_{i*})$  é a densidade desse ponto focal. O valor da densidade para todo vetor de dados padronizados  $\hat{\mathbf{d}}_k$  é computado de acordo com a seguinte expressão (ANGELOV, 2010):

$$D_k(\hat{\mathbf{d}}_k) = \frac{k-1}{(k-1) \left( \sum_{j=1}^{n+1} (\hat{d}_k^j)^2 + 1 \right) + b_k - 2 \sum_{j=1}^{n+1} \hat{d}_k^j g_k^j}, \quad k = W+2, W+3, \dots$$

$$\text{onde } D_{W+1}(\hat{\mathbf{d}}_{W+1}) = 1 \text{ (condição inicial),} \quad (4.18)$$

$$b_k = b_{k-1} + \sum_{j=1}^{n+1} (\hat{d}_{k-1}^j)^2, \quad b_{W+1} = 0, \text{ para } j = 1, 2, \dots, n+1$$

$$g_k^j = g_{(k-1)}^j + \hat{d}_{(k-1)}^j, \quad g_{W+1}^j = 0, \text{ para } j = 1, 2, \dots, n+1,$$

onde o primeiro vetor de dados padronizado é definido como o ponto focal do primeiro *cluster*, com densidade  $D_{W+1}(\hat{\mathbf{d}}_{W+1}) = 1$ , lembrando que o processo evolutivo se inicia nesse instante  $W+1$ .

Quando  $\hat{\mathbf{d}}_k$  é escolhido como ponto focal de um novo *cluster* pela Condição A ( $\hat{\mathbf{f}}_{i^*} = \hat{\mathbf{d}}_k$ ), seu valor de densidade passa a ser armazenado pela variável  $D_k(\hat{\mathbf{f}}_{i^*})$  e atualizado, enquanto esse *cluster* existir, por outra expressão definida conforme a seguir (ANGELOV, 2010):

$$D_k(\hat{\mathbf{f}}_{i^*}) = \frac{k-1}{k-1 + (k-2) \left( \frac{1}{D_{k-1}(\hat{\mathbf{f}}_{i^*})} - 1 \right) + \sum_{j=1}^{n+1} (\hat{d}_k^j - \hat{d}_{k-1}^j)} \quad (4.19)$$

Essa atualização alternativa da densidade na Equação 4.19 ocorre porque todo vetor de dados a cada instante influencia no valor da densidade de todos os *clusters* já existentes. Finalizado esse procedimento do cálculo da densidade, a seguinte Condição B (Equação 4.20) é avaliada para cada *cluster* ou regra *fuzzy*:

$$\begin{aligned} \text{SE } \mu_i^j(\hat{d}_k^j) > e^{-1} \quad (i = 1, 2, \dots, R \text{ e } j = 1, 2, \dots, n) \\ \text{ENTÃO } R \leftarrow R - 1, \end{aligned} \quad (4.20)$$

para  $\mu_i^j$  a função de pertinência do  $i$ -ésimo *cluster* e  $j$ -ésima entrada ou componente contida em  $\hat{\mathbf{d}}_k$ , com  $j = 1, 2, \dots, n$ . Essa condição é muito importante para evitar a redundância e controlar a elevada sobreposição entre os *clusters* (ANGELOV, 2010) (ANGELOV, 2013). Todos os *clusters* existentes que satisfazem essa condição são substituídos pelo novo *cluster* gerado pela Condição A com informações mais atualizadas, pois se a condição B é verdadeira, então não é preciso ter vários *clusters* que descrevem a mesma região.

A função de pertinência  $\mu_i^j$  é gaussiana e é calculada para cada entrada ou componente no vetor de dados padronizado  $\hat{\mathbf{d}}_k$  de acordo com a seguinte equação:

$$\mu_i^j(\hat{d}_k^j) = e^{-\frac{(\hat{f}_{i^*}^j - \hat{d}_k^j)^2}{2[\sigma_i^j(k)]^2}}, \quad \text{para } i = 1, 2, \dots, R \text{ e } j = 1, 2, \dots, n, \quad (4.21)$$

onde  $\sigma_i^j(k)$  representa a zona de influência ou raio do *cluster*  $i$  no eixo de entrada  $j$  no espaço  $n$ -dimensional para o instante  $k$  (Seção 3.2.1). O valor de  $\sigma_i^j(k)$  é atualizado recursivamente a cada instante  $k$  por:

$$\sigma_i^j(k) = \sqrt{\zeta[\sigma_i^j(k-1)]^2 + (1-\zeta)\frac{1}{S_k^i}(\hat{d}_k^j - \hat{f}_{i^*}^j)}, \quad \text{para } i = 1, 2, \dots, R \text{ e } j = 1, 2, \dots, n+1, \quad (4.22)$$

sendo que  $\zeta$  é uma constante de aprendizado e  $S_k^i$  é o suporte ou número de pontos no espaço  $n$ -dimensional associados ao  $i$ -ésimo *cluster* no instante  $k$ . A constante  $\zeta$  e a condição inicial para os raios dos *clusters* -  $\sigma_i^j(W+1)$  - são fatores críticos que afetam o desempenho geral da RNF-TS, sendo assim, seus valores precisam ser selecionados apropriadamente. O valor da zona de influência  $\sigma_i^j(k)$  em cada eixo do espaço de dados faz com que o *cluster* não tenha uma forma bem definida, pois o seu formato se adapta ao comportamento do vetor de dados  $\hat{\mathbf{d}}_k$ .

Finalmente, as proposições do antecedente das regras *fuzzy* podem ser formuladas usando o ponto focal dos *clusters* identificados pelo algoritmo de agrupamento evolutivo descrito. Logo, os antecedentes possuem a seguinte composição, considerando  $\hat{\mathbf{d}}_k$ :

$$\text{Regra}^i : \mathbf{SE} (\hat{c}_{k-1}^1 \sim \hat{f}_{i*}^1) \mathbf{E} (\hat{c}_{k-1}^2 \sim \hat{f}_{i*}^2) \mathbf{E} (\hat{c}_{k-1}^3 \sim \hat{f}_{i*}^3) \mathbf{E} \cdots \mathbf{E} (\hat{c}_{k-1}^n \sim \hat{f}_{i*}^n) \quad (4.23)$$

onde  $\hat{f}_{i*}^j$  é o ponto focal do cluster  $i$  para a  $j$ -ésima entrada ou componente, com  $j = 1, 2, \dots, n$ . O símbolo  $\sim$  significa *está próximo de*.

#### 4.2.2.1 Algoritmo da Camada de Agrupamento Evolutivo

O algoritmo que representa a execução dessa camada é apresentado a seguir:

---

#### Algoritmo 3 Agrupamento Evolutivo

---

- 1: Formular o vetor de dados  $\mathbf{d}_k$  (Equação 4.13);
  - 2: Padronização de  $\mathbf{d}_k$  (Equações 4.14, 4.15 e 4.16);
  - 3: Calcular a densidade  $D_k(\hat{\mathbf{d}}_k)$  (Equação 4.18);
  - 4: **para**  $i \leftarrow 1$  **até**  $R$  **faça**
  - 5:     Calcular  $D_k(\hat{\mathbf{f}}_{i*})$  (Equação 4.19);
  - 6: **fim para**
  - 7: **se** Condição  $A$  é verdadeira (Equação 4.17) **então**
  - 8:     Gerar um novo *cluster*:  $R = R + 1$  e  $\hat{\mathbf{f}}_{i*} = \hat{\mathbf{d}}_k$ ;
  - 9:     **se** Condição  $B$  é verdadeira (Equação 4.20) **então**
  - 10:         Substituir os *clusters* que satisfazem essa condição;
  - 11:     **fim se**
  - 12: **fim se**
  - 13: **para**  $i \leftarrow 1$  **até**  $R$  **faça**
  - 14:     Atualizar Funções de Pertinência 4.21;
  - 15:     Atualizar Raio dos *Clusters* 4.22;
  - 16: **fim para**
- 

#### 4.2.3 Camada de Agregação

A terceira camada da RNF-TS usa a norma-t do tipo produto para calcular o grau de ativação de cada regra *fuzzy*, considerando as funções de pertinência obtidas na camada anterior. Logo:

$$\tau_k^i = t \left\{ \mu_i^j(\hat{d}_k^j) \right\} = \prod_{j=1}^n \mu_i^j(\hat{d}_k^j), \text{ para } i = 1, 2, \dots, R. \quad (4.24)$$

Os valores  $\tau_k^i$  devem ser normalizados de tal forma que seja possível mensurar a contribuição da resposta local de cada regra na resposta global do sistema. Dessa forma, a soma de

todos os graus de ativação deve ser igual a 1, ou seja:

$$\lambda_k^i = \frac{\tau_k^i}{\sum_{q=1}^R \tau_k^i}. \quad (4.25)$$

para  $\lambda_{ki}$  o grau de ativação normalizado do *cluster*  $i$ .

#### 4.2.3.1 Algoritmo da Camada de Agregação

O Algoritmo 4 traz a execução simples da Camada de Agregação, que visa obter os valores do grau de ativação normalizado  $\lambda_k^i$  para cada regra *fuzzy*:

---

#### Algoritmo 4 Algoritmo da Camada de Agregação

---

- 1: **para**  $i \leftarrow 1$  **até**  $R$  **faça**
  - 2:     Calcular grau de ativação  $\tau_k^i$  (Equação 4.24);
  - 3: **fim para**
  - 4: **para**  $i \leftarrow 1$  **até**  $R$  **faça**
  - 5:     Calcular grau de ativação normalizado  $\lambda_k^i$  (Equação 4.25);
  - 6: **fim para**
- 

#### 4.2.4 Camada de Estimação das Matrizes de Estado

A quarta camada envolve as proposições do consequente das regras *fuzzy*. Nesse trabalho, uma regra *fuzzy* completa da RNF-TS é representada por:

$$\begin{aligned} \text{Regra}^i : \quad & \mathbf{SE} (\hat{c}_{k-1}^1 \sim \hat{f}_{i^*}^1) \mathbf{E} \cdots \mathbf{E} (\hat{c}_{k-1}^n \sim \hat{f}_{i^*}^n) \\ & \text{ENTÃO } \hat{\mathbf{x}}_k = \mathbf{A}_k^i \hat{\mathbf{x}}_{k-1} + \mathbf{B}_k^i \hat{\mathbf{u}}_{k-s}, \end{aligned} \quad (4.26)$$

para  $\mathbf{A}_k^i \in \mathbb{R}^{n \times n}$ ,  $\mathbf{B}_k^i \in \mathbb{R}^{n \times n}$ , e:

$$\begin{aligned} \hat{\mathbf{x}}_k &= [\hat{c}_k^1, \hat{c}_k^2, \dots, \hat{c}_k^n]^T, & \hat{\mathbf{x}}_k &\in \mathbb{R}^{n \times 1}; \\ \hat{\mathbf{x}}_{k-1} &= [\hat{c}_{k-1}^1, \hat{c}_{k-1}^2, \dots, \hat{c}_{k-1}^n]^T, & \hat{\mathbf{x}}_{k-1} &\in \mathbb{R}^{n \times 1}; \\ \hat{\mathbf{u}}_{k-s} &= [\hat{c}_{k-s}^1, \hat{c}_{k-s}^2, \dots, \hat{c}_{k-s}^n]^T, & \hat{\mathbf{u}}_{k-s} &\in \mathbb{R}^{n \times 1}. \end{aligned} \quad (4.27)$$

Em outras palavras, as componentes não-observáveis extraídas são tratadas como os próprios estados e os modelos no espaço de estado representam o comportamento dinâmico dessas componentes. As componentes não-observáveis podem ser consideradas como estados, pois são linearmente independentes entre si quando extraídas pelo método AES, condição necessária para uma variável ser considerada um estado do sistema (CHEN, 1999). O vetor  $\hat{\mathbf{u}}_{k-s}$  considera informações das componentes no instante  $k-s$ , em que  $s$  é o período sazonal da série temporal. O valor de  $s$  pode ser informado pelos especialistas ou estimado a partir da função de autocovariância, conforme apresentado na Seção 2.2.3. Para séries sazonais, a informação nesse instante é muito significativa, pois apresenta um comportamento similar ao instante atual  $k$ .

O método dos Mínimos Quadrados Recursivos Ponderados *Fuzzy Local*, do inglês *Local Fuzzy Weighted Recursive Least Squares* (L-FWRLS) (ANGELOV, 2010), é responsável por estimar as matrizes  $\mathbf{A}_k^i$  e  $\mathbf{B}_k^i$  das proposições do consequente de cada regra *fuzzy* existente, usando o grau de ativação normalizado  $\lambda_k^i$  (Equação 4.25) como fator de ponderação. Essa camada precisa dos dados das componentes e, portanto, da observação da série  $y_k$  para atualizar os elementos das matrizes. Considerando as componentes ou estados obtidos na Camada de Decomposição:

$$\begin{aligned}\hat{\mathbf{x}}_k &= \mathbf{A}_k^i \hat{\mathbf{x}}_{k-1} + \mathbf{B}_k^i \hat{\mathbf{u}}_{k-s}, \\ \hat{\mathbf{x}}_k &= \begin{bmatrix} \mathbf{A}_k^i & \mathbf{B}_k^i \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_{k-1} \\ \hat{\mathbf{u}}_{k-s} \end{bmatrix},\end{aligned}\quad (4.28)$$

$$\hat{\mathbf{x}}_k = \Delta_i^T(k) \mathbf{v}^T.$$

para  $\Delta_i^T(k)$  a matriz que contém os elementos de  $\mathbf{A}_k^i$  e  $\mathbf{B}_k^i$  a serem estimados. Logo, a estimação recursiva para cada regra *fuzzy*  $i = 1, 2, \dots, R$  é dada por:

$$\begin{aligned}\mathbf{G}_k^i &= \frac{\mathbf{P}_{k-1}^i \mathbf{v}^T}{\mathbf{v} \lambda_k^i \mathbf{P}_{k-1}^i \mathbf{v}^T + 1}, \quad \mathbf{G}_k^i \in \mathbb{R}^{2n \times 1}, \\ \Delta_i^T(k) &= \Delta_i^T(k-1) + \lambda_k^i \mathbf{G}_k^i (\hat{\mathbf{x}}_k^T - \mathbf{v} \Delta_i^T(k-1)), \quad \Delta_i^T(k) \in \mathbb{R}^{2n \times n},\end{aligned}\quad (4.29)$$

$$\mathbf{P}_k^i = \mathbf{P}_{k-1}^i - \lambda_k^i \mathbf{G}_k^i \mathbf{v}^T \mathbf{P}_{k-1}^i, \quad \mathbf{P}_k^i \in \mathbb{R}^{2n \times 2n},$$

onde  $\mathbf{G}_k^i$  é a matriz de ganho de cada regra e  $\mathbf{P}_k^i$  é a matriz de covariância da  $i$ -ésima regra *fuzzy*.

#### 4.2.4.1 Algoritmo da Camada de Estimação das Matrizes de Estado

A execução dessa camada é representada no algoritmo a seguir:

---

#### Algoritmo 5 Algoritmo da Camada de Estimação das Matrizes de Estado

---

- 1: **para**  $i \leftarrow 1$  **até**  $R$  **faça**
  - 2:   Obter valores do grau de ativação normalizado (Equação 4.25);
  - 3:   Aplicar o método L-FWRLS (Equação 4.29).
  - 4: **fim para**
- 

#### 4.2.5 Camada de Inferência da RNF-TS

A Camada de Inferência tem a função de gerar como saída os estados globais da RNF-TS denotadas por  $\hat{\mathbf{x}}'_k$ , considerando a contribuição  $\hat{\mathbf{x}}_k^i$  de cada regra *fuzzy*  $i$  através da média ponderada pelo grau de ativação normalizado  $\lambda_k^i$ . Esses termos significam:

- $\hat{\mathbf{x}}_k^i$ : saída padronizada contendo os valores das componentes após a atualização das matrizes das equações de estado para cada regra *fuzzy*  $i$ , ou seja, a resposta do submodelo;
- $\hat{\mathbf{x}}_k'$ : saída padronizada global contendo os valores das componentes após a atualização das matrizes das equações de estado, ou seja, a previsão de cada componente.

Os estados globais são calculados por:

$$\hat{\mathbf{x}}_k' = \sum_{i=1}^R \lambda_k^i \hat{\mathbf{x}}_k^i. \quad (4.30)$$

Percebe-se então que a metodologia proposta realiza a previsão de cada componente separadamente, já que a saída global é composta pelo vetor de estado global  $\hat{\mathbf{x}}_k'$ .

#### 4.2.5.1 Algoritmo da Camada de Inferência

Logo abaixo, o Algoritmo 6 mostra o procedimento realizado na Camada de Inferência para obtenção do vetor de estados global:

---

#### Algoritmo 6 Algoritmo da Camada de Inferência

---

- 1: **para**  $i \leftarrow 1$  **até**  $R$  **faça**
  - 2:     Obter estados locais  $\hat{\mathbf{x}}_k^i = \mathbf{A}_k^i \hat{\mathbf{x}}_{k-1}^i + \mathbf{B}_k^i \hat{\mathbf{u}}_{(k-s)}^i$ ;
  - 3: **fim para**
  - 4: Calcular o estado global  $\hat{\mathbf{x}}_k'$  (Equação 4.30).
- 

#### 4.2.6 Camada de Reconstrução

A última camada usa os estados globais obtidos a partir da camada anterior (Equação 4.30) e reconstrói a série temporal no instante  $k$  ao somar os valores das componentes. No entanto, os dados de  $\hat{\mathbf{x}}_k'$  estão padronizados. Sendo assim, essa camada precisa retornar os dados para escala original, utilizando os valores do desvio padrão e média calculados por meio da Equação 4.15. Logo, os estados globais  $\tilde{\mathbf{x}}_k'$ , agora no espaço original, são:

$$\tilde{x}_k^{j'} = \hat{x}_k^{j'} \delta_k^j + \bar{d}_k^j, \text{ para } j = 1, 2, \dots, n. \quad (4.31)$$

A previsão da série temporal no instante  $k$  no estágio de treinamento da RNF-TS é computada por uma das seguintes expressões:

$$\begin{aligned} \tilde{y}_k &= [1 \ 1 \ \dots \ 1] \tilde{\mathbf{x}}_k', \\ \tilde{y}_k &= \sum_{j=1}^n \tilde{x}_k^{j'}, \\ \tilde{y}_k &= \tilde{c}_k^1 + \tilde{c}_k^2 + \dots + \tilde{c}_k^n. \end{aligned} \quad (4.32)$$

#### 4.2.6.1 Algoritmo da Camada de Reconstrução

O Algoritmo 7 a seguir exhibe o processo da Camada de Reconstrução da metodologia proposta:

---

#### **Algoritmo 7** Algoritmo da Camada de Reconstrução

---

- 1: Transformar os dados do estado global  $\hat{\mathbf{x}}_k^l$  para a escala original (Equação 4.31);
  - 2: Obter o resultado da RNF-TS pela Equação 4.32.
- 

#### 4.2.7 Remoção de *Clusters* com Baixa Qualidade

Durante a execução das camadas da RNF-TS, os *clusters* são monitorados, removendo-se aqueles que apresentam baixa qualidade. A Condição *C* na Equação 4.33 é responsável por realizar a avaliação da qualidade dos *clusters* ou das regras *fuzzy*. Se essa condição é satisfeita, o grau de ativação normalizado do *cluster* *i* será igual a zero ( $\lambda_{ki} = 0$ ), indicando que a saída da regra *fuzzy* não terá influência nos estados globais da RNF-TS (Equação 4.30). A Condição *C* é estruturada conforme a seguir:

$$\begin{aligned} & \mathbf{SE} (\Omega_k^i < \bar{\Omega}_k - \tilde{\Omega}_k) \mathbf{E} (S_k^i < 3) \mathbf{E} (k \geq T_{i^*} + 10) \\ & \mathbf{ENTÃO} \lambda_k^i \leftarrow 0, \end{aligned} \quad (4.33)$$

onde  $S_k^i$  é o suporte do *cluster* *i* e  $T_{i^*}$  representa o instante *k* em que o *cluster* *i* foi gerado. O termo  $\Omega_{ki}$  corresponde a utilidade do *i*-ésimo *cluster* no instante *k*, que é calculada como:

$$\Omega_k^i = \frac{\sum_{q=1}^k \lambda_q^i}{k - T_{i^*}}, \text{ para } i = 1, 2, \dots, R, \quad (4.34)$$

sendo que o numerador é a soma de todos os graus de ativação normalizados do *cluster* *i* desde sua criação.

Os termos  $\bar{\Omega}_k$  e  $\tilde{\Omega}_k$  denotam respectivamente o valor médio e o desvio padrão da utilidade depois que *k* amostras da série temporal são lidas. Ambos valores são calculados respectivamente por:

$$\begin{aligned} \bar{\Omega}_k &= \frac{k-1}{k} \bar{\Omega}_{k-1} + \frac{1}{k} \Omega_k, \quad \bar{\Omega}_{W+1} = 0 \text{ (condição inicial),} \\ \tilde{\Omega}_k &= \sqrt{\frac{k-1}{k} \tilde{\Omega}_{k-1}^2 + \frac{1}{k} (\Omega_k - \bar{\Omega}_k)^2}, \quad \tilde{\Omega}_{W+1} = 0 \text{ (condição inicial).} \end{aligned} \quad (4.35)$$

Após a camada de inferência, os *clusters* que satisfazem a condição *C* são removidos.



#### 4.2.8 Algoritmo do Estágio de Treinamento

Para finalizar essa seção de treinamento, o Algoritmo 8 traz a operação completa desse estágio, considerando  $N$  amostras da série temporal utilizadas para o treinamento da RNF-TS.

---

#### Algoritmo 8 Estágio de Treinamento

---

- 1: Definição dos Parâmetros iniciais:  $W$ ,  $L$ ,  $n$ ,  $\zeta$  e do raio inicial;
  - 2: Condições Iniciais do Método AESR (Algoritmo 1);
  - 3: **para**  $k \leftarrow W + 1$  **até**  $N$  **faça**
  - 4:     Executar o Algoritmo 2 da Camada de Decomposição;
  - 5:     **se** for o primeiro vetor de dados  $\mathbf{d}_k$  **então**
  - 6:         Condições Iniciais da Padronização (Equação 4.16);
  - 7:         Condições Iniciais da Utilidade (Equação 4.35);
  - 8:         Condições Iniciais da Densidade (Equação 4.18);
  - 9:     **senão**
  - 10:         Agrupamento Evolutivo (Algoritmo 3);
  - 11:         Execução da Camada de Agregação (Algoritmo 4);
  - 12:         **se** Condição  $A$  não foi ativada para o vetor  $\hat{\mathbf{d}}_k$  **então**
  - 13:             Associar o ponto  $\hat{\mathbf{d}}_k$  ao *cluster* existente com maior grau de ativação  $\tau_k^i$ ;
  - 14:         **fim se**
  - 15:         Estimação das matrizes de estado (Algoritmo 5);
  - 16:         Atualizar a utilidade 4.34;
  - 17:         Atualizar os parâmetros da utilidade 4.35;
  - 18:         **para**  $i \leftarrow 1$  **até**  $R$  **faça**
  - 19:             **se** Condição  $C$  for satisfeita 4.33 **então**
  - 20:                 Desativar influência da regra *fuzzy* na resposta global ( $\lambda_k^i = 0$ );
  - 21:             **fim se**
  - 22:         **fim para**
  - 23:         Mecanismo de inferência (Algoritmo 6);
  - 24:         Saída geral da RNF-TS (Algoritmo 7);
  - 25:         Remoção dos *clusters* que obedeceram a Condição  $C$ .
  - 26:     **fim se**
  - 27: **fim para**
- 

### 4.3 Estágio de Previsão

Conforme foi mencionado anteriormente, no estágio de treinamento há informação  $y_k$  disponível a partir da série temporal. Então, todo o procedimento descrito anteriormente pelas camadas e resumido pelo Algoritmo 8 é realizado. Por outro lado, quando não há amostras da série temporal disponíveis, o estágio de previsão é acionado. Nesse caso, a RNF-TS não evolui, não monitora os *clusters*, não atualiza os consequentes das regras *fuzzy* e não aplica nenhuma atualização recursiva apresentada nas camadas, ou seja, mantém sua estrutura fixa para prever observações da série. Sendo assim, a Camada de Decomposição e a Camada de Estimação das Matrizes de Estado não são executadas, já que são exigidas observações da série para extrair as componentes não-observáveis e para atualizar as matrizes de estados pelo método L-FWRLS.

Para obter as componentes não-observáveis e construir os vetores de dados  $\mathbf{d}_k$ , o estado global  $\tilde{\mathbf{x}}'_{k-1}$  que a RNF-TS obtém como saída e que contém os dados das componentes no instante anterior é realimentado para a Camada de Agrupamento Evolutivo da RNF-TS. Dessa forma, o vetor de dados no estágio de previsão é:

$$\begin{aligned} \mathbf{d}_k &= [\tilde{\mathbf{x}}'_{k-1} \text{ NULO}], \\ \mathbf{d}_k &= [\tilde{c}_{k-1}^1 \ \tilde{c}_{k-1}^2 \ \cdots \ \tilde{c}_{k-1}^n \ \text{NULO}]. \end{aligned} \quad (4.36)$$

Para descrever a operação do estágio de previsão, o Algoritmo 9 será apresentado logo a seguir, considerando o horizonte de previsão  $h$ .

---

#### Algoritmo 9 Estágio de Previsão

---

- 1: **para**  $k \leftarrow N$  **até**  $N + h$  **faça**
  - 2:   Realimentação dos estados;
  - 3:   Formulação do vetor de dados (Equação 4.36);
  - 4:   Padronização do vetor de dados (Equação 4.14);
  - 5:   Aplicação da Camada de Agregação (Algoritmo 4);
  - 6:   Mecanismo de inferência *fuzzy* (Algoritmo 6);
  - 7:   Saída Geral da RNF-TS (Algoritmo 7).
  - 8: **fim para**
-

## 5 RESULTADOS EXPERIMENTAIS

### 5.1 Considerações Iniciais sobre os Experimentos

Nesse capítulo, a metodologia evolutiva proposta é avaliada e seus resultados são comparados com outras pesquisas. Dois experimentos foram realizados. O Experimento 1 compara os resultados de previsão para dois valores de horizontes de previsão diferentes com as técnicas propostas em Stepnicka et al. (2013). Já o Experimento 2 compara os resultados de treinamento e previsão da RNF-TS com técnicas detalhadas nas pesquisas de Gan et al. (2014), Zhang e Qi (2005) e de Temizel (2005).

A RNF-TS proposta tem alguns parâmetros críticos, cujas variações em seus valores influenciam bastante nos resultados de previsão. São eles:

- $W$ : quantidade de amostras da série temporal para as condições iniciais do AESR;
- $L$ : dimensão dos vetores  $\mathbf{m}_q$  (Equação 4.2);
- $n$ : número de componentes não-observáveis extraídas;
- $\sigma_{W+1}^j$ : raio inicial dos *clusters* para cada  $j$ ,  $j = 1, 2, \dots, n + 1$  (Equação 4.22);
- $\zeta$ : constante de aprendizado (Equação 4.22).

Logo, os valores para esses parâmetros devem ser adequadamente selecionados para cada série temporal. Nessa pesquisa, a escolha dos parâmetros é realizada por tentativa e erro, sendo que o desenvolvimento de uma metodologia para seleção dos parâmetros da RNF-TS é deixada para trabalhos futuros.

### 5.2 Experimento 1 - Desempenho da RNF-TS para Horizontes de Previsão Distintos

As séries temporais utilizadas nesse experimento são compostas por dados reais de diferentes áreas e domínios, o que as tornam interessantes para efetuar a previsão. Todas essas séries são sazonais e mensais, ou seja, com período sazonal  $s = 12$ . Seus dados podem ser obtidos a partir da Biblioteca de Dados de Séries Temporais de Hyndman (*Hyndman's Time Series Data Library*) (HYNDMAN, 2016). A descrição de cada uma dessas séries temporais é listada a seguir:

1. **Passageiros** internacionais de linhas aéreas: totais mensais em milhares, janeiro de 1949 a dezembro de 1960 (144 amostras);

2. Número total mensal de **suínos** abatidos em Victoria, janeiro de 1980 a agosto de 1995 (188 amostras);
3. Vendas de **automóveis** mensais em Quebec, 1960-1968 (108 amostras);
4. Demanda mensal de **gasolina** na cidade de Ontario em milhões de galões, 1960 a 1975 (192 amostras);
5. Produção mensal de **leite**: libras por bovino, janeiro de 1962 a dezembro de 1975 (168 amostras);
6. Vendas da indústria de **papel** para impressão e escrita. Janeiro de 1963 a Dezembro de 1972 (120 amostras);
7. Média mensal de viagens de **ônibus**. Janeiro de 1973 a Junho de 1982 (114 amostras).

A primeira técnica utilizada para comparação de desempenho entre as metodologias discutidas em Stepnicka et al. (2013) é o método tradicional para séries sazonais SARIMA. O modelo SARIMA foi implementado nessa pesquisa de referência através do *software ForecastingPro* (GOODRICH, 2000), uma ferramenta muito utilizada que automaticamente seleciona valores ótimos para os parâmetros desse método. Essa seleção ótima de parâmetros para o SARIMA faz com que seja um desafio superá-lo quanto ao desempenho de previsão.

Outro método apresentado em Stepnicka et al. (2013) é o Projeto Automático de Redes Neurais Artificiais (*Automatic Design of Artificial Neural Networks - ADANN*), que aplica a técnica evolucionária conhecida como Algoritmos Genéticos para construir a estrutura da rede neural. O passo inicial do ADANN é a normalização dos dados da série temporal para que as funções de ativação dos neurônios estejam contidas no intervalo  $[0, 1]$ . Um único neurônio é colocado na camada de saída e previsões múltiplos passos a frente são realizadas usando uma realimentação iterativa de previsões anteriores. O Algoritmo Genético é responsável por buscar a melhor estrutura para a rede neural.

A técnica conhecida como SVM (Máquina de Vetores de Suporte) é uma poderosa ferramenta que se baseia no uso de uma função que transforma variáveis de entrada em um espaço de grande dimensão e, em seguida, encontra o melhor modelo para representar os dados nesse espaço. Os modelos SVM em geral são bastante flexíveis e apresentam a capacidade de aprendizado não-linear. A pesquisa de Stepnicka et al. (2013) considera essa técnica para previsão de múltiplos passos a frente para séries temporais sazonais.

Além das técnicas ADANN e SVM introduzidas acima, a pesquisa de Stepnicka et al. (2013) propõe também a utilização de abordagens de inteligência computacional, mais especificamente a lógica *fuzzy*, sendo que dois novos conceitos híbridos foram introduzidos pelos autores com o nome de *Fuzzy ANN* (FANN) e *Fuzzy SVM* (FSVM).

Todas essas metodologias apresentadas anteriormente visam prever séries temporais sazonais considerando múltiplos passos a frente, ou seja, possuem os mesmos objetivos que a RNF-TS proposta nesta pesquisa. Os mesmos critérios de avaliação utilizados no trabalho de Stepnicka et al. (2013) serão adotados nesse experimento, visando efetuar a comparação entre as técnicas supracitadas e a RNF-TS proposta. Os critérios de avaliação são:

- Erro Médio Absoluto Percentual Simétrico (*Symmetric Mean Absolute Percentage Error - SMAPE*):

$$\text{SMAPE} = \frac{1}{h} \sum_{k=N+1}^{N+h} \frac{|y_k - \tilde{y}_k|}{(|y_k| + |\tilde{y}_k|)/2} \times 100\%; \quad (5.1)$$

- Erro Médio Absoluto (*Mean Absolute Scaled Error - MASE*):

$$\text{MASE} = \frac{1}{h} \sum_{k=N+1}^{N+h} \frac{|y_k - \tilde{y}_k|}{\frac{1}{N-1} \sum_{q=2}^N |y_q - y_{q-1}|}. \quad (5.2)$$

onde  $h$  é o horizonte de previsão,  $N$  é a quantidade de amostras da série para o treinamento do método,  $y_k$  é a observação da série temporal e  $\tilde{y}_k$  é a previsão ou resposta dos métodos analisados. Quanto menor os valores para esses critérios de erros, melhor o desempenho das metodologias avaliadas. Ambos critérios de avaliação são considerados simultaneamente para realizar uma avaliação consistente na comparação entre as metodologias. Além disso, os resultados para diferentes horizontes de previsão também são analisados, a fim de verificar melhor a capacidade de cada método. Sendo assim, dois diferentes horizontes de previsão são adotados:  $h^1 = 12$  e  $h^2 = 24$ .

Os resultados comparativos entre as técnicas de acordo com os critérios de erro SMAPE e MASE são exibidos nas Tabelas 1 e 2, respectivamente, sendo que os melhores estão em negrito. A metodologia proposta apresentou resultados superiores para todas as séries temporais em ambos os horizontes de previsão ( $h^1$  e  $h^2$ ) de acordo com ambos critérios, com exceção da série de **gasolina**, em que a técnica ADANN foi melhor para o horizonte de previsão  $h^2$ . Em relação a série temporal de **papel**, houve um empate técnico entre a RNF-TS e a ADANN para  $h^2$ . Para se ter uma medida de comparação genérica considerando todas as séries, a média aritmética e a mediana são calculadas para cada coluna das tabelas. Portanto, de acordo com essas medidas, a RNF-TS apresentou o melhor desempenho em geral para todas as séries temporais e ambos os horizontes de previsão nesse experimento. No entanto, não é possível generalizar que a metodologia proposta obterá os melhores resultados para qualquer série temporal de qualquer categoria, mas seu potencial de previsão para séries sazonais foi claramente demonstrado nesses resultados.

Adicionalmente, a Tabela 3 exibe os valores para os parâmetros críticos da RNF-TS que forneceram os resultados representados nas Tabelas 1 e 2, considerando ambos horizontes de

Tabela 1 – Comparação das Metodologias de Acordo com o Critério SMAPE (Melhores Resultados em Negrito).

Série	SARIMA		ADANN		SVM		FANN		FSVM		RNF-TS	
	$h^1$	$h^2$	$h^1$	$h^2$	$h^1$	$h^2$	$h^1$	$h^2$	$h^1$	$h^2$	$h^1$	$h^2$
passageiros	6,5	8,0	2,2	2,5	5,0	7,1	2,1	2,5	3,0	3,9	<b>1,9</b>	<b>2,3</b>
suínos	6,1	7,1	7,3	11,5	5,8	7,2	6,7	8,2	7,7	7,8	<b>4,6</b>	<b>6,3</b>
automóveis	7,4	9,1	10,6	9,8	11,0	10,0	12,1	10,0	11,6	10,9	<b>5,4</b>	<b>7,8</b>
gasolina	5,5	6,2	4,1	<b>3,8</b>	6,2	7,0	4,0	5,6	4,8	5,9	<b>2,3</b>	4,0
leite	0,8	0,9	1,4	2,3	1,1	1,3	1,1	1,1	1,0	1,0	<b>0,5</b>	<b>0,9</b>
papel	7,3	9,9	7,1	<b>7,7</b>	6,1	8,3	8,8	8,7	7,5	9,9	<b>5,3</b>	<b>7,7</b>
ônibus	9,0	13,8	3,5	6,0	5,6	7,4	16,1	18,5	14,1	17,7	<b>2,5</b>	<b>3,2</b>
Média	6,1	7,9	5,2	6,2	5,8	6,9	7,3	7,8	7,1	8,2	<b>3,2</b>	<b>4,6</b>
Mediana	6,5	8,0	4,1	6,0	5,8	7,2	6,7	8,2	7,5	7,8	<b>2,5</b>	<b>4,0</b>

Tabela 2 – Comparação das Metodologias de Acordo com o Critério MASE (Melhores Resultados em Negrito).

Série	SARIMA		ADANN		SVM		FANN		FSVM		RNF-TS	
	$h^1$	$h^2$	$h^1$	$h^2$	$h^1$	$h^2$	$h^1$	$h^2$	$h^1$	$h^2$	$h^1$	$h^2$
passageiros	1,24	1,60	0,42	0,51	1,02	1,52	0,39	0,49	0,56	0,80	<b>0,37</b>	<b>0,48</b>
suínos	0,64	0,76	0,74	1,17	0,61	0,76	0,70	0,87	0,80	0,81	<b>0,50</b>	<b>0,66</b>
automóveis	0,54	0,66	0,70	0,68	0,72	0,69	0,79	0,75	0,75	0,74	<b>0,39</b>	<b>0,53</b>
gasolina	1,12	1,27	0,83	<b>0,77</b>	1,24	1,41	0,80	1,15	0,96	1,22	<b>0,45</b>	0,80
leite	0,19	0,21	0,31	0,53	0,25	0,29	0,24	0,24	0,22	0,22	<b>0,12</b>	<b>0,20</b>
papel	0,47	0,69	0,42	<b>0,50</b>	0,42	0,60	0,61	0,62	0,52	0,69	<b>0,38</b>	<b>0,50</b>
ônibus	3,06	4,77	1,15	1,95	1,84	2,42	5,66	6,52	4,91	6,18	<b>0,76</b>	<b>1,00</b>
Média	1,04	1,42	0,65	0,87	0,87	1,10	1,31	1,52	1,25	1,52	<b>0,42</b>	<b>0,60</b>
Mediana	0,64	0,76	0,70	0,68	0,72	0,76	0,70	0,75	0,75	0,80	<b>0,39</b>	<b>0,53</b>

Tabela 3 – Valores dos Parâmetros da RNF-TS para cada Série Temporal e Ambos Horizontes de Previsão.

Série	$W$		$L$		$n$		$\sigma_i^j(W+1)$		$\zeta$		$R$		tempo (s)	
	$h^1$	$h^2$	$h^1$	$h^2$	$h^1$	$h^2$	$h^1$	$h^2$	$h^1$	$h^2$	$h^1$	$h^2$	$h^1$	$h^2$
passageiros	14	19	6	3	6	3	0,14	0,06	0,5	0,5	5	19	0,22	0,36
suínos	21	20	3	6	3	6	0,09	0,14	0,5	0,5	5	11	0,25	0,37
automóveis	13	15	3	6	3	6	0,33	0,34	0,5	0,5	6	4	0,17	0,19
gasolina	13	25	5	5	5	5	0,09	0,179	0,5	0,51	3	9	0,26	0,29
leite	13	17	4	6	4	6	0,05	0,14	0,5	0,5	6	4	0,37	0,24
papel	14	30	3	7	3	7	0,45	0,1	0,5	0,5	3	5	0,17	0,17
ônibus	13	13	3	4	3	4	0,12	0,14	0,5	0,5	6	3	0,19	0,16
Média	14	20	4	5	4	5	0,18	0,16	0,5	0,5	5	8	0,23	0,25

previsão. Outrossim, o número de regras *fuzzy* ou *clusters* gerados no estágio de treinamento ( $R$ ) usado para prever cada série temporal e o tempo de execução em segundos dos experimentos também são mostrados na Tabela 3. Observando a média, note que, para o horizonte  $h^2$ , é necessário em geral uma maior quantidade inicial de amostras  $W$  das séries temporais para o método AESR. Além disso, valores superiores para  $L$  e um número mais elevado de componentes não-observáveis  $n$  são necessários para o horizonte de previsão  $h^2$  em relação a  $h^1$ .

Outra vantagem da RNF-TS desenvolvida é seu tempo de execução. Enquanto as abordagens baseadas em SVM e ADNN executam em dezenas de segundos e dezenas de minutos, respectivamente, a RNF-TS precisa de menos de um segundo devido a sua característica recursiva, considerando plataformas computacionais semelhantes. Para séries temporais mensais,

os requisitos de tempo não são muito exigentes, mas em outras aplicações, o tempo pode ser uma questão-chave. A RNF-TS ainda é capaz de evoluir sua estrutura baseada em regras *fuzzy*, ajustando-se aos dados de entrada para cada instante.

### 5.3 Experimento 2 - Desempenho da RNF-TS nos Estágios de Treinamento e Previsão

Por sua vez, outras quatro séries temporais abordadas no trabalho de Gan et al. (2014), todas com 120 amostras e com período sazonal também definido por  $s = 12$ , são descritas a seguir:

1. Venda de **livros**;
2. Venda de **roupas**;
3. Venda de **móveis**;
4. Venda de **hardware**.

Nesse experimento, a RNF-TS precisa realizar somente a previsão dos últimos doze valores dessas séries, isto é, o horizonte de previsão é  $h = 12$ . Logo, as primeiras 108 amostras dessas séries temporais são usadas para o treinamento da RNF-TS e o restante dos dados é comparado com sua saída para fins de validação. O critério de erro RMSE (Raiz Quadrada do Erro Quadrático Médio - *Root Mean Square Error*) é utilizado para avaliação da metodologia proposta e para comparação com outras técnicas apresentadas em Gan et al. (2014), já que este mesmo critério é aplicada nessa referência. Sua formulação é:

$$\text{RMSE} = \sqrt{\frac{1}{h} \sum_{k=1}^h (y_k - \tilde{y}_k)^2}, \quad (5.3)$$

para  $y_k$  a observação da série e  $\tilde{y}_k$  a previsão resultante dos métodos. Da mesma forma que os critérios do Experimento 1, quanto menor o valor desse índice, melhor o desempenho da técnica considerada.

A RNF-TS proposta foi comparada com outras quatro técnicas e com os resultados apresentados em Gan et al. (2014). Esses autores realizaram a aplicação do método de Rede Neural de Função de Base Radial Baseado em Modelos Autorregressivos (*Radial Basis Function Based on Autoregressive Model*, RBF-AR) para previsão dessas séries sazonais. O modelo RBF-AR é derivado usando um conjunto de redes RBF para aproximar os coeficientes de um modelo AR. Dessa forma, essa técnica é interpretada como um modelo AR linear, em que a evolução do processo é governada por um conjunto de coeficientes AR. Para melhorar o desempenho da previsão do modelo RBF-AR, Gan et al. (2014) propõem o uso de Algoritmos Genéticos e Algoritmos de Otimização baseados no Gradiente para a seleção de valores adequados para os

Tabela 4 – Comparação dos Resultados para as Séries Temporais Usando o RMSE, com Melhores Resultados em Negrito (T - Estágio de Treinamento e P - Estágio de Previsão).

Série	RBR-AR		ANN+DSDT		TDNN	SARIMA	RNF-TS	
	T	P	T	P	P	P	T	P
Livros	62.63(2.03)	96.8(3.87)	37.89	88.74	91.51(10.41)	98.17	<b>20.34</b>	<b>58.06</b>
Roupas	244.89(0.0)	337.72(0.01)	174.07	315.43	372.5(50.66)	519.6	<b>143.8</b>	<b>312.02</b>
Móveis	77.85(0.12)	191.54(1.98)	52.35	<b>99.45</b>	173.1(31.1)	124.44	<b>50.20</b>	110.24
Hardware	40.31(0.3)	23.53(0.28)	26.16	49.17	35.7(6.9)	100.71	<b>25.79</b>	<b>22.80</b>

parâmetros desse modelo, como a seleção de variáveis de entrada adequadas, o número de nós ocultos, as variáveis dependentes do modelo, entre outros.

Por sua vez, a pesquisa de Zhang e Qi (2005) conclui que as Redes Neurais Artificiais (*Artificial Neural Network* - ANN) por si só não são capazes de modelar diretamente séries temporais com tendências e padrões sazonais, sendo necessário um pré-processamento dos dados para fornecer resultados apropriados de previsão. Uma série temporal com tendências e sazonalidade não satisfaz as condições para a aproximação universal, pois o modelo da ANN é limitado para representar essas componentes de acordo com esses autores. Os melhores resultados em Zhang e Qi (2005) foram alcançados quando tanto a tendência e a sazonalidade foram previamente removidas (*Deseasonalized and Detrended* - DSDT), formando o método ANN+DSDT que também será comparado com a RNF-TS proposta nessa dissertação.

A terceira técnica abordada nesse experimento é a Rede Neural com Atraso de Tempo (*Time Delay Neural Network* - TDNN), proposta em Temizel (2005). Para a TDNN, cada camada é conectada à saída da sua camada anterior e, portanto, é capaz de relacionar a entrada atual com valores passados. Por fim, o modelo tradicional SARIMA também será novamente avaliado para efeito de comparação com essas técnicas e a RNF-TS, considerando essas quatro outras séries temporais sazonais.

Na Tabela 4, os resultados das técnicas supracitadas tanto para o treinamento quanto para a previsão dessas séries de acordo com o critério RMSE são exibidos, sendo que os melhores estão em negrito. A partir da Tabela 5, é possível visualizar os valores para os parâmetros da RNF-TS aplicados para que esses resultados fossem alcançados. A quantidade de regras ou *clusters* gerados em cada caso no estágio de treinamento da RNF-TS está listada na coluna *R* dessa última tabela. As colunas dos modelos RBF-AR e TDNN correspondem ao valor médio do critério RMSE para 30 execuções dos algoritmos dos respectivos modelos, sendo que o valor entre parênteses representa o desvio padrão obtido. As demais técnicas, inclusive a RNF-TS, fornecem os mesmos resultados para um mesmo conjunto de dados e parâmetros definidos, não sendo necessário representar suas médias e desvios padrão. No treinamento, a metodologia proposta foi superior em todos os casos.

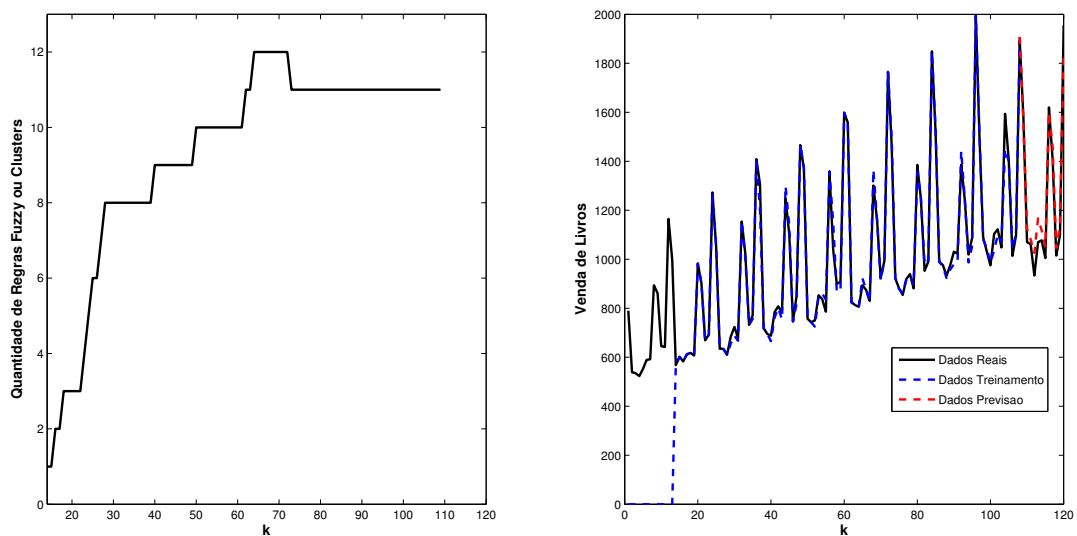
Em relação a previsão, a RNF-TS exibiu resultados melhores segundo o RMSE para as séries de **livros**, de **roupas** e de **hardware**. Porém, para a série de **móveis**, o modelo ANN+DSDT foi levemente superior. Conforme discutido, o modelo ANN+DSDT realiza um



Tabela 5 – Valores dos Parâmetros da RNF-TS para cada Série Temporal no Experimento 2.

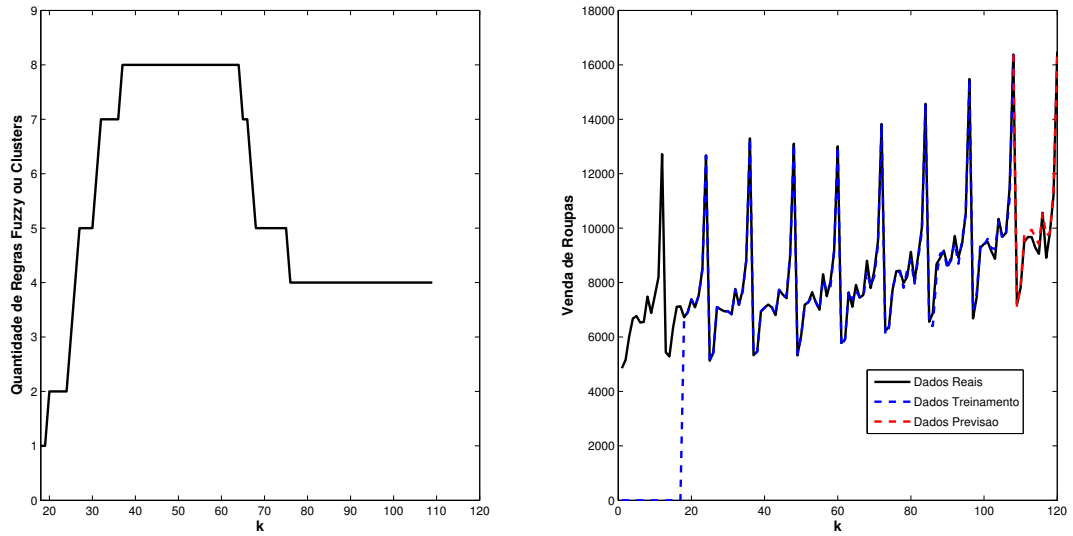
Série	$W$	$L$	$n$	$\sigma_i^j(W+1)$	$\zeta$	$R$	tempo (s)
Livros	13	3	3	0.09	0.5	11	0.38
Roupas	17	7	7	0.156	0.55	4	0.29
Móveis	37	4	4	0.19	0.5	3	0.27
Hardware	18	8	8	0.13	0.5	4	0.26

pré-processamento dos dados que remove a tendência e a sazonalidade. Sendo assim, a RNF-TS apresenta a vantagem de não remover esses padrões específicos, mas sim considerar as componentes não-observáveis e prevê-las separadamente para, em seguida, obter o resultado da previsão da série temporal original.

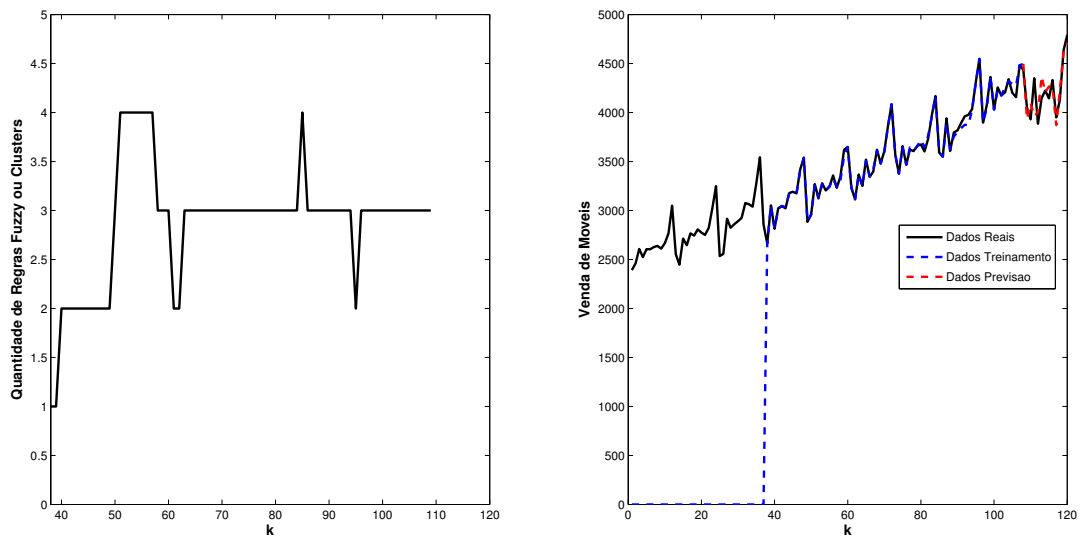
Figura 13 – Quantidade de Regras *Fuzzy* ou *Clusters* e Resultados da RNF-TS para a Série Temporal de **Livros**.

Fonte: Autor.

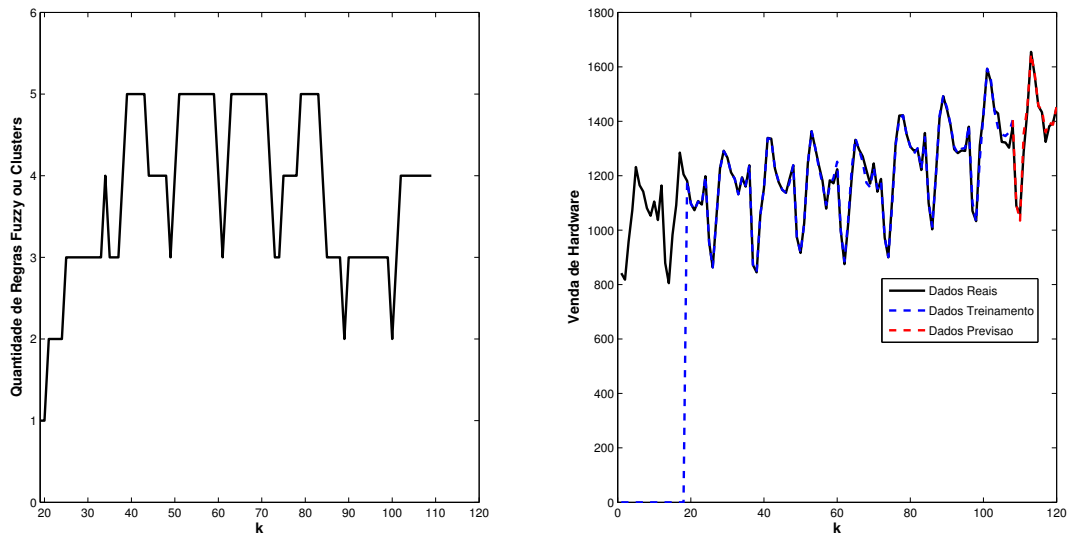
Pode-se observar nas Figura 13, 14, 15 e 16 os resultados do treinamento e da previsão para cada uma das quatro séries temporais do Experimento 2. As figuras à esquerda exibem a mudança da quantidade de *clusters* ou regras *fuzzy* durante o estágio de treinamento da RNF-TS, destacando seu caráter evolutivo. Já nas figuras à direita, em preto, os dados reais das séries são exibidos. A parte inicial em preto corresponde a quantidade de dados iniciais  $W$  para o intervalo da Equação 4.1. A saída do estágio de treinamento e a previsão das 12 últimas amostras geradas pela RNF-TS são mostradas em azul e vermelho, respectivamente.

Figura 14 – Quantidade de Regras *Fuzzy* ou *Clusters* e Resultados da RNF-TS para a Série Temporal de *Roupas*.

Fonte: Autor.

Figura 15 – Quantidade de Regras *Fuzzy* ou *Clusters* e Resultados da RNF-TS para a Série Temporal de *Móveis*.

Fonte: Autor.

Figura 16 – Quantidade de Regras Fuzzy ou Clusters e Resultados da RNF-TS para a Série Temporal de *Hardware*.

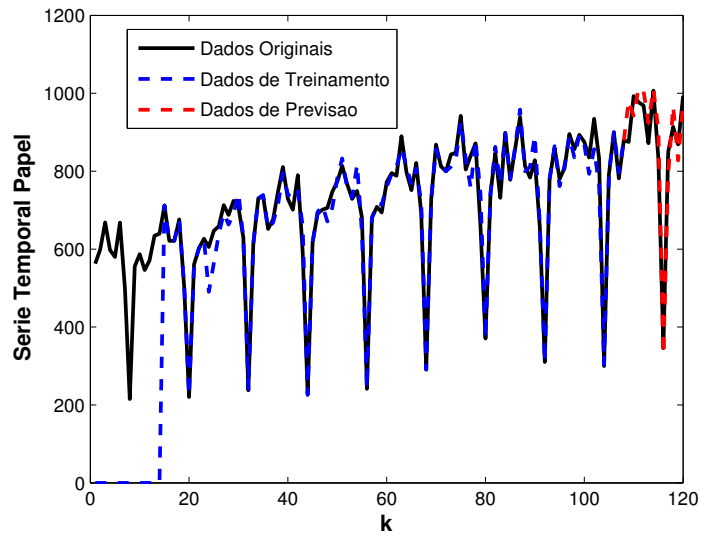
Fonte: Autor.

#### 5.4 Avaliação das Características da RNF-TS para a Série Temporal de Vendas de Papel

Uma avaliação das principais características da metodologia de previsão proposta nesse trabalho é apresentada nessa seção, considerando a série temporal **papel** comentada anteriormente no Experimento 1. Conforme citado, essa série apresenta 120 amostras e seu período sazonal  $s = 12$ . O horizonte de previsão adotado é igual a  $h^1 = 12$ , ou seja, doze passos a frente. Sendo assim, as primeiras 108 amostras da série são usadas no estágio de treinamento para evoluir e adaptar a estrutura da RNF-TS aos dados das componentes não-observáveis extraídas desse conjunto de dados. A outra parte dos dados, constituída por 12 amostras da série, é aplicada para efeito de análise do desempenho e validação da previsão por parte da RNF-TS. Lembrando que os parâmetros selecionados envolvendo a série papel para  $h^1 = 12$  estão representados na Tabela 3.

A Figura 17 mostra os resultados dos estágios de treinamento e previsão para essa série temporal. Da mesma forma que as figuras anteriores, a cor preta indica os dados originais, sendo que a parte inicial corresponde as  $W$  amostras da série no intervalo inicial  $\Phi$  na Equação 4.1. Os resultados no treinamento estão em azul e os da previsão  $h^1 = 12$  passos a frente em vermelho. Note que uma boa aproximação foi alcançada mesmo quando não há dados disponíveis da série temporal para evoluir a RNF-TS.

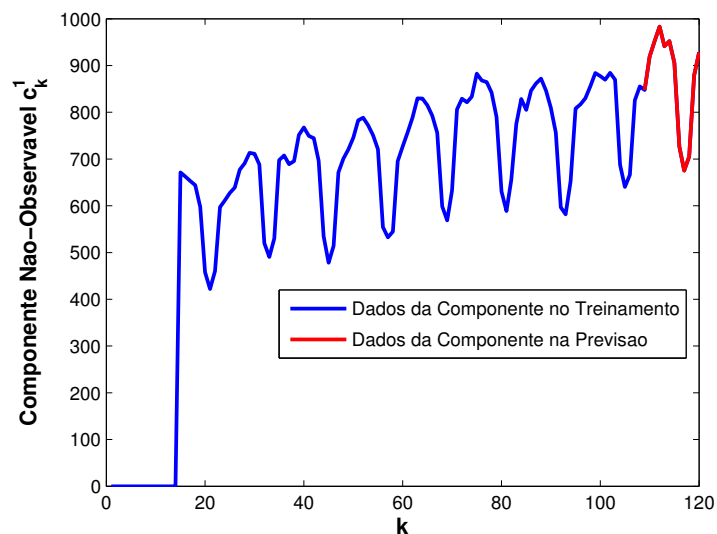
Figura 17 – Resultados de Treinamento e Previsão da RNF-TS para a Série Temporal **Papel**.



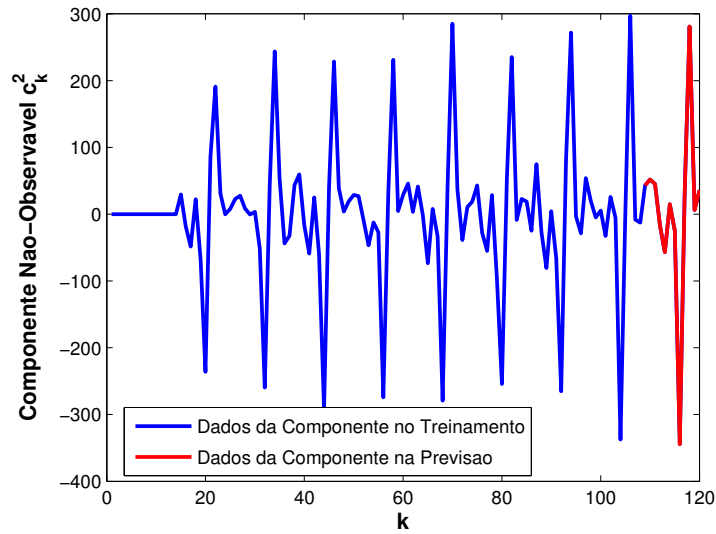
Fonte: Autor.

As Figuras 18, 19 e 20 mostram as três componentes não-observáveis extraídas dessa série temporal, respectivamente:  $c_k^1$ ,  $c_k^2$  e  $c_k^3$ . O valor das componentes ou mesmo estados são exibidos em azul e a cor vermelha corresponde aos estados ou componentes que foram realimentados no estágio de previsão, ou seja, a previsão das componentes. Os modelos no espaço de estados dos consequentes das regras *fuzzy* são capazes de descrever apropriadamente a dinâmica das componentes não-observáveis, obtendo bons resultados na previsão da série original.

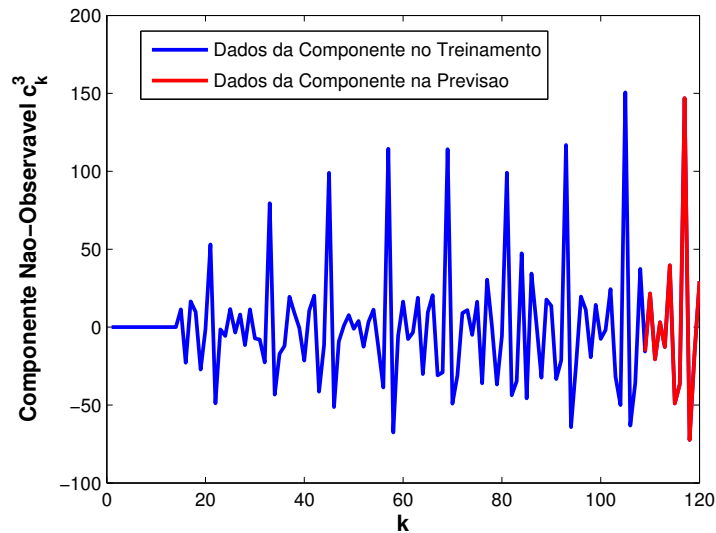
Figura 18 – Componente Não-Observável  $c_k^1$  da Série Temporal **Papel**.



Fonte: Autor.

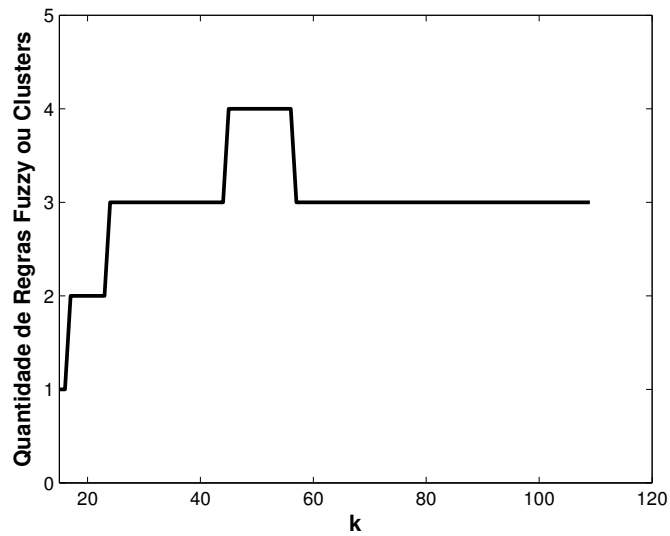
Figura 19 – Componente Não-Observável  $c_k^2$  da Série Temporal **Papel**.

Fonte: Autor.

Figura 20 – Componente Não-Observável  $c_k^3$  da Série Temporal **Papel**.

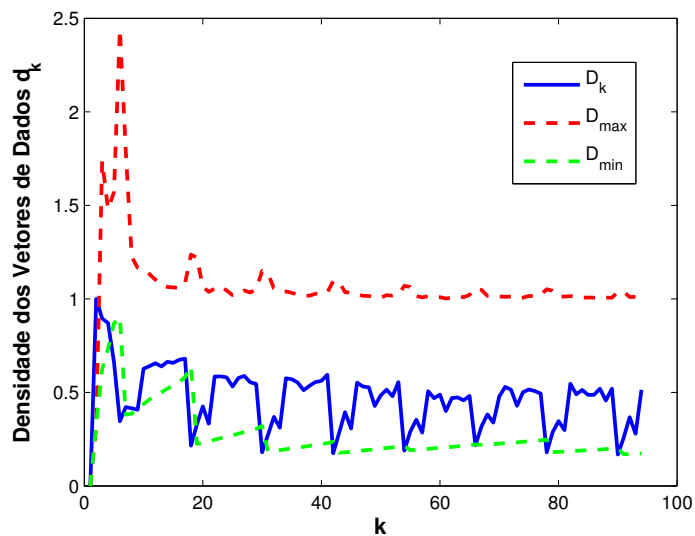
Fonte: Autor.

Devido a característica evolutiva da RNF-TS, sua quantidade de regras *fuzzy* ou *clusters* é alterada, podendo aumentar ou diminuir quando tenta adequar sua estrutura aos dados da série que chegam a todo instante. Assim, a evolução dessas regras ou *clusters* para a série temporal **papel** no estágio de treinamento, no qual ocorre a evolução da RNF-TS, é representada na Figura 21.

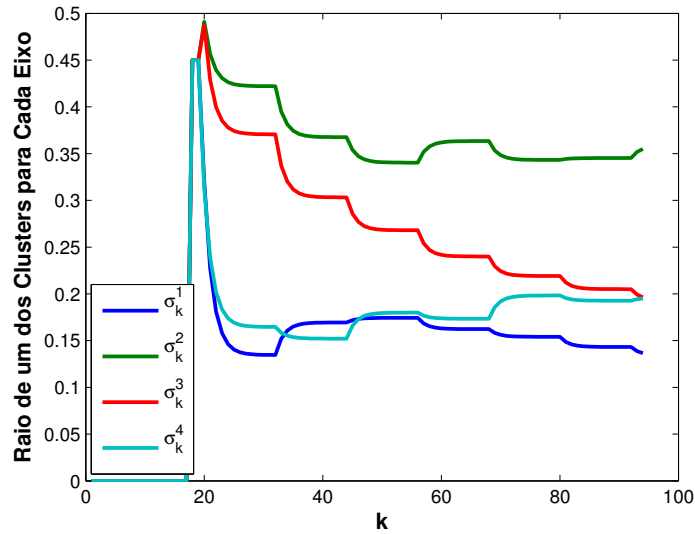
Figura 21 – Quantidade de Regras *Fuzzy* ou *Clusters* para a Série Temporal **Papel**.

Fonte: Autor.

O comportamento da densidade do vetor de dados padronizado  $\hat{\mathbf{d}}_k$  para a série temporal **papel** é exibido na Figura 22. Para cada instante  $k$  em que a densidade de  $\hat{\mathbf{d}}_k$  ( $D_k(\hat{\mathbf{d}}_k)$ ) em azul for além dos limites mínimo (verde) ou máximo (vermelho) representados na Condição A (Equação 4.17), uma nova regra *fuzzy* ou *cluster* é gerado, com ponto focal  $\hat{\mathbf{f}}_{i*} = \hat{\mathbf{d}}_k$ .

Figura 22 – Densidade do Vetor de Dados Padronizado  $\hat{\mathbf{d}}_k$  para Avaliação da Condição A (Equação 4.17).

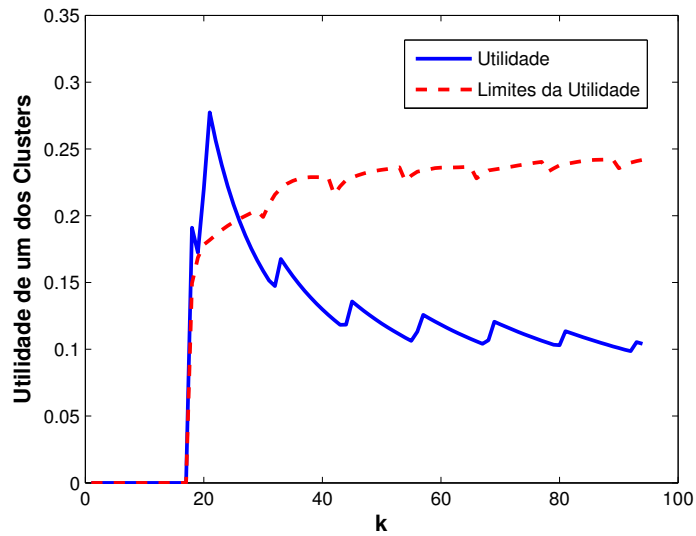
Fonte: Autor.

Figura 23 – Comportamento do Raio de um *Cluster* Específico Gerado em  $k = 18$ .

Fonte: Autor.

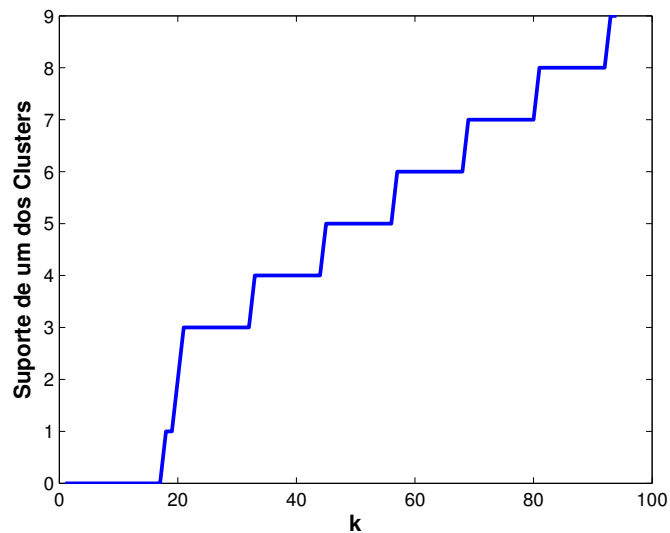
Para analisar algumas características específicas dos *clusters*, vamos considerar um *cluster* particular gerado no instante  $k = 18$  no estágio de treinamento para a série temporal **papel**. Os raios desse *cluster* específico para cada um dos quatro eixos no espaço de dados (três em relação as componentes não-observáveis e uma relacionada a observação  $y_k$  da série utilizada no treinamento) são exibidos na Figura 23. Percebe-se que no instante  $k = 18$  quando o *cluster* é criado, os raios possuem o valor inicial de 0,45, conforme definido na Tabela 3. Nos instantes posteriores, os raios dos *clusters* em cada eixo se adaptam de acordo com os dados que chegam no estágio de treinamento.

A adaptação da medida de utilidade  $\Omega_k^i$  desse mesmo *cluster* pode ser visualizada na Figura 24. Ao longo do tempo, a utilidade do *cluster* decresce, fato relacionado geralmente com *clusters* antigos. Note que a utilidade fica menor que o limite estabelecido em vermelho, sendo assim, caso não haja muitas amostras associadas a esse *cluster*, ele provavelmente será removido através da aplicação da Condição C (Equação 4.33).

Figura 24 – Utilidade de um *Cluster* Particular Gerado em  $k = 18$ .

Fonte: Autor.

Entretanto, pelo comportamento do suporte  $S_k^i$  desse *cluster* (Figura 25), percebe-se que ele não pode ser removido, já que a quantidade de pontos associados a ele é crescente. Dessa maneira, este *cluster* contém informações importantes e possui um submodelo representativo que descreve esse conjunto de dados.

Figura 25 – Suporte de um *Cluster* Particular Gerado em  $k = 18$ .

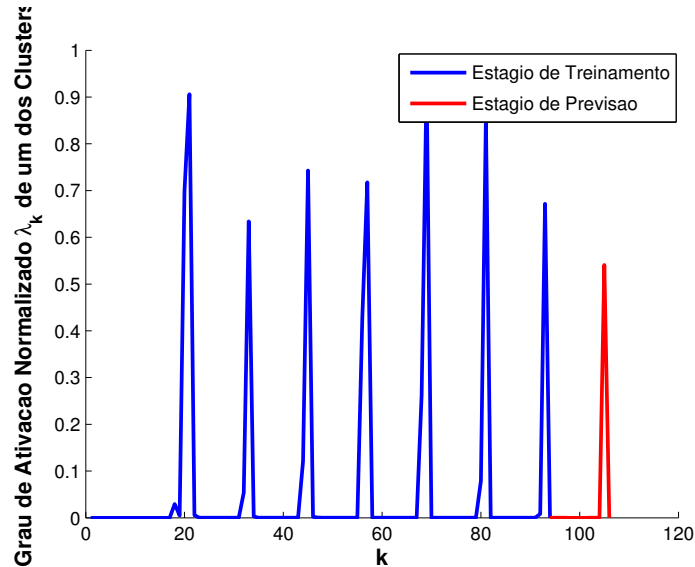
Fonte: Autor.

Por fim, a Figura 26 exibe o grau de ativação normalizado  $\lambda_k^i$  desse mesmo *cluster* ou regra *fuzzy*. Esse valor  $\lambda_k^i$  mostra o grau de influência desse *cluster* particular no resultado da previsão global da RNF-TS. Percebe-se que há um padrão de ativação para  $\lambda_k^i$  e, apesar de um



grande período sem ser ativado reduzindo a utilidade dessa regra *fuzzy*, esse *cluster* é importante para descrever determinado padrão da série temporal de acordo com a figura. Os valores de  $\lambda_k^i$  para o estágio de treinamento e para o estágio de previsão são representados em azul e vermelho, respectivamente.

Figura 26 – Grau de Ativação Normalizado de um *Cluster* Particular Gerado em  $k = 18$ .



Fonte: Autor.

## 5.5 Estudo de Caso: Detecção de Anomalias Baseada em Dados de Eletrocardiogramas

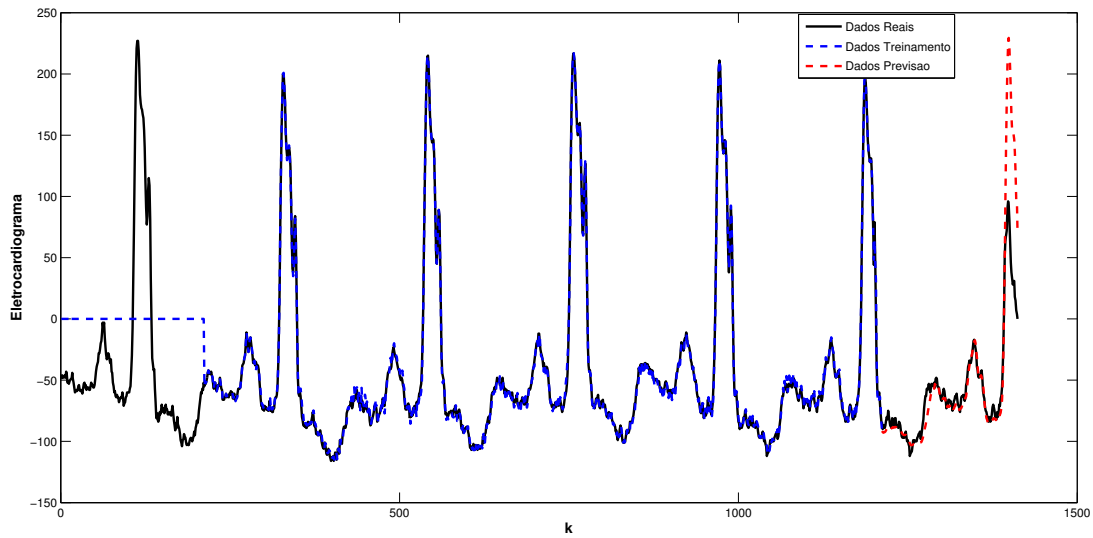
No presente estudo de caso, considera-se uma série temporal com dados de um eletrocardiograma de um paciente com insuficiência cardíaca (BAIM et al., 1986). Um eletrocardiograma é um tipo de exame em que são registradas as variações dos potenciais elétricos geradas pelo coração. Esse exame é indicado para análise de doenças cardíacas, pelo qual o responsável pode visualizar anomalias e arritmias (BAIM et al., 1986).

Dessa forma, o objetivo da RNF-TS nesse estudo de caso é evoluir e se adaptar para que, em seguida, possa monitorar os batimentos do coração em tempo-real. No estágio de treinamento, o comportamento considerado normal do paciente é aprendido e representado pela RNF-TS, enquanto que o estágio de previsão realiza o monitoramento e comparação. Caso a diferença entre os dados de previsão e os dados medidos for muito grande, o erro será considerado alto, fazendo com que os responsáveis sejam alertados sobre o comportamento fora de padrão. Essa série temporal não possui comportamento sazonal bem definido e fixo. Logo, o seu valor foi determinado por inspeção, sendo  $s = 210$ .

Pela Figura 27, nota-se que a RNF-TS representa bem o sinal do eletrocardiograma no estágio de treinamento em azul. Sendo assim, a metodologia proposta reproduz o comportamento considerado normal em vermelho; contudo, os dados reais em preto indicam que a magnitude do

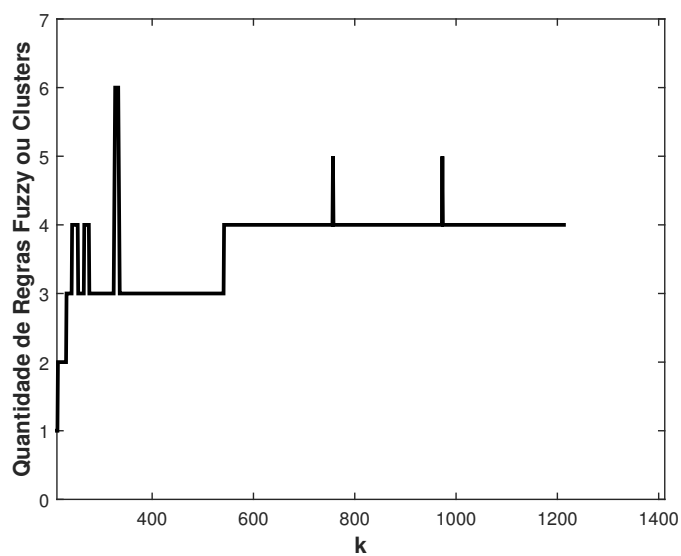
sinal está abaixo do esperado. Portanto, a RNF-TS poderá alertar sobre a anomalia do sinal aos responsáveis pelo paciente nesse caso. O comportamento evolutivo é evidenciado na Figura 28, onde exibe-se a modificação da estrutura manifestada pela quantidade de regras *fuzzy* ou *clusters*. Por fim, as quatro componentes não-observáveis extraídas do eletrocardiograma são exibidas nas Figuras 29 a 32.

Figura 27 – Eletrocardiograma de um Paciente.

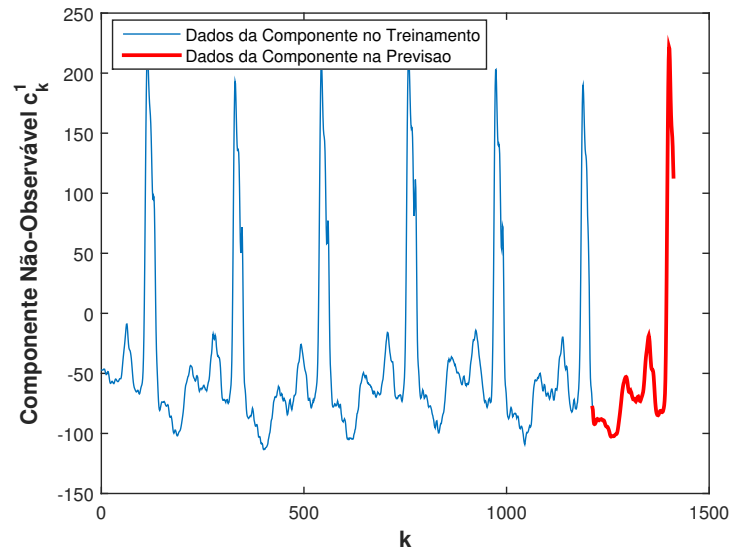


Fonte: Autor.

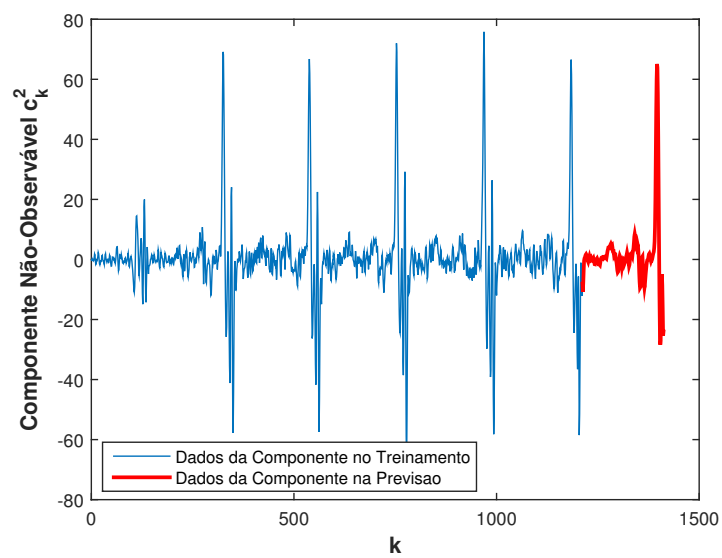
Figura 28 – Quantidade de Regras *Fuzzy* ou *Clusters* para a Série **Eletrocardiograma**.



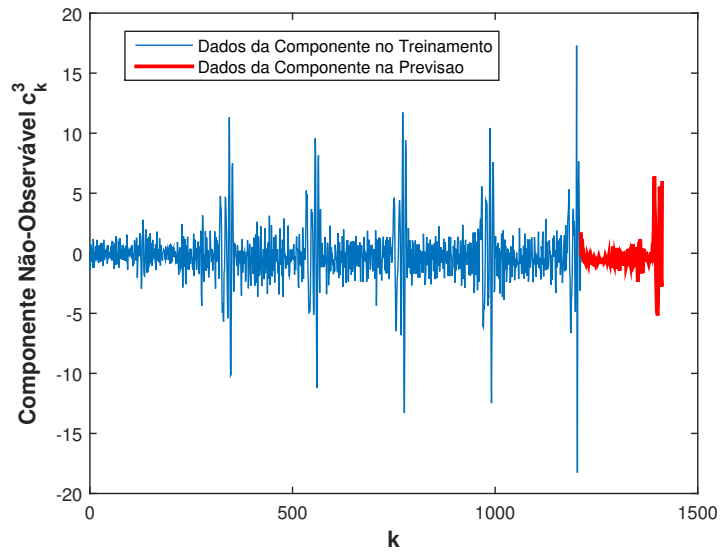
Fonte: Autor.

Figura 29 – Componente Não-Observável  $c_k^1$  da Série **Eletrocardiograma**.

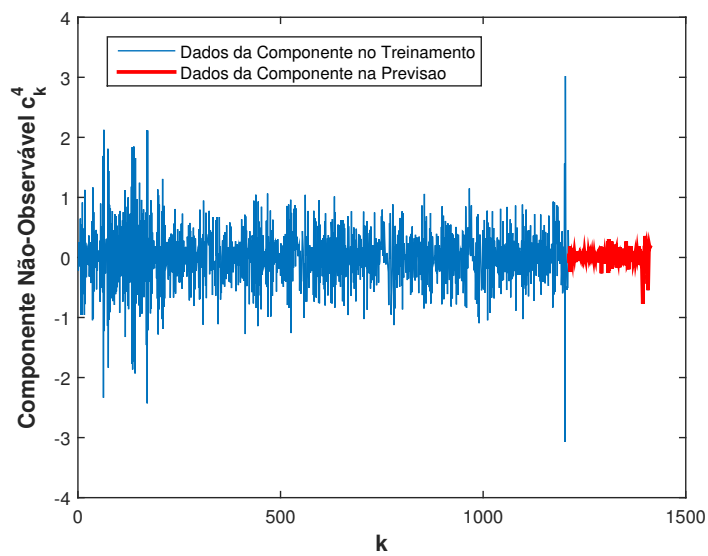
Fonte: Autor.

Figura 30 – Componente Não-Observável  $c_k^2$  da Série **Eletrocardiograma**.

Fonte: Autor.

Figura 31 – Componente Não-Observável  $c_k^3$  da Série **Eletrocardiograma**.

Fonte: Autor.

Figura 32 – Componente Não-Observável  $c_k^4$  da Série **Eletrocardiograma**.

Fonte: Autor.

## 6 CONSIDERAÇÕES FINAIS

### 6.1 Conclusões

Nessa dissertação, uma metodologia evolutiva baseada em uma rede *neuro-fuzzy* Takagi-Sugeno foi proposta para previsão de séries temporais sazonais. Além da previsão, a RNF-TS é capaz também de extrair as componentes não-observáveis, que são os padrões contidos nos dados das séries temporais. Essa rede é capaz de evoluir, ou seja, de aprender e ajustar sua estrutura a partir dos dados e sem interferência do usuário, sendo que a quantidade de regras *fuzzy* pode aumentar ou diminuir. A RNF-TS foi desenvolvida para lidar com os dados que chegam a cada instante, ou seja, realizar seu procedimento de forma *online*.

A RNF-TS apresentou resultados competitivos e vantajosos em relação a outras técnicas desenvolvidas na literatura, de acordo com os critérios de avaliação considerados. A extração das componentes não-observáveis da série original e a previsão separada de cada componente contribuíram bastante para os resultados da metodologia, comprovando a hipótese de que os padrões mais simples podem ser mais facilmente reconstruídos no futuro. Essa previsão separada de cada componente ou padrão é uma característica bem original dessa pesquisa.

Outro aspecto é que o tempo de execução da RNF-TS nos experimentos foi bem pequeno em comparação aos outros métodos comentados nessa pesquisa, que demoram vários minutos para encerrarem a execução de seus algoritmos. Sendo assim, a metodologia proposta nessa pesquisa pode lidar com requisitos de tempo rigorosos.

Ademais, a modelagem no espaço de estados para os consequentes das regras *fuzzy*, uma contribuição inovadora da pesquisa, foi apropriadamente aplicada, já que esse modelo representa a dinâmica das componentes ou estados através das matrizes de estado. No estágio de previsão, os estados são essenciais para a RNF-TS, pois são realimentados para fornecer os dados das componentes necessários para a RNF-TS, já que não há dados disponíveis da série temporal nesse caso. Outrossim, a RNF-TS é capaz de modificar sua operação entre os estágios de treinamento e previsão automaticamente ao identificar se há dados da série temporal disponíveis ou não. Destacou-se o potencial dessa metodologia proposta na apresentação do estudo de caso sobre anomalias em eletrocardiogramas, mostrando que é possível explorar essa abordagem em outros tipos de aplicações.

É importante também destacar a contribuição dessa pesquisa em relação a modificação do método AES tradicional. O AES original considera todo o conjunto de dados da série temporal e realiza um procedimento em batelada. Para extrair as componentes não-observáveis de forma recursiva e lidar com os dados da série temporal que chegam a cada instante, houve a necessidade de adaptar esse método para a camada de decomposição da RNF-TS evolutiva. Sendo assim, a abordagem proposta AESR conseguiu realizar a extração das componentes não-observáveis de

forma apropriada, contribuindo para o bom desempenho da previsão.

Outra contribuição foi a condição de remoção de *clusters* ou regras *fuzzy* utilizando medidas de qualidade. Apesar das medidas de utilidade, suporte e tempo serem conhecidas na área de sistemas evolutivos, essa condição foi formulada especialmente para o problema de previsão de séries temporais sazonais para que *clusters* com informações importantes e periódicas não fossem removidos do espaço de dados, prejudicando o desempenho da RNF-TS.

## 6.2 Trabalhos Futuros

Sugestões de trabalhos futuros são listadas a seguir:

- A RNF-TS depende da seleção *a priori* de valores adequados dos parâmetros para fornecer bons resultados de previsão. Sendo assim, essa metodologia tem potencial de ter um desempenho melhor em comparação aos que foram apresentados nessa pesquisa. Portanto, a investigação e desenvolvimento de uma metodologia para escolha de valores para os parâmetros críticos da RNF-TS é uma meta para trabalhos futuros;
- Expansão da funcionalidade da RNF-TS evolutiva para séries temporais de diversas outras características, podendo ser caóticas e com outros tipos de não-linearidades. Além disso, a exploração dessa metodologia no contexto multivariável é interessante;
- Análise de abordagens alternativas para decomposição de componentes não-observáveis e para agrupamento evolutivo, pois essas áreas de pesquisa estão em constante evolução e crescimento;
- Aplicação em problemas reais na medicina, indústria, identificação de sistemas, entre outras áreas em que a previsão é interessante;
- Inserção do conhecimento do especialista aplicado na previsão do tipo qualitativa no formato de regras *fuzzy* para refinar os resultados da RNF-TS proposta;
- Exploração sobre a extração evolutiva de conhecimentos úteis a partir de grandes volumes de dados denominados *big data*.

## REFERÊNCIAS

ABDOLLAHZADE, M. et al. Locally linear neuro-fuzzy (LLNF) electricity price forecasting in deregulated power markets. *International Journal of Innovative Computing Information and Control*, v. 6, p. 4203–4218, 2010.

ABDOLLAHZADE, M. et al. A new hybrid enhanced local linear neuro-fuzzy model based on the optimized singular spectrum analysis and its application for nonlinear and chaotic time series forecasting. *Information Sciences*, v. 295, p. 107–125, 2015.

AGGARWAL, C. C.; REDDY, C. K. *Data clustering : algorithms and applications*. Boca Raton, FL: CRC Press, 2014. (Chapman & Hall/CRC data mining and knowledge discovery series). ISBN 978-1-466-55821-2.

AGUIRRE, L. A. *Introdução à Identificação de Sistemas - Técnicas Lineares e Não-Lineares: Teoria e Aplicação*. 4. ed. Belo Horizonte: Editora UFMG, 2015.

ANGELOV, P. *Evolving Rule-based Models: A Tool for Design of Flexible Adaptive Systems*. Springer, 2002.

ANGELOV, P. Evolving Takagi-Sugeno Fuzzy Systems From Streaming Data. In: ANGELOV, P.; FILEV, D.; KASABOV, N. (Ed.). *Evolving Intelligent Systems: Methodology and Applications*. Hoboken, New Jersey: IEEE Press Series on Computational Intelligence. Published by John Wiley & Sons, 2010. cap. 2, p. 21–50.

ANGELOV, P. *Autonomous Learning Systems: From Data Streams to Knowledge in Real-Time*. [S.l.]: Wiley, 2013.

ANGELOV, P.; FILEV, D. On-line Design of Takagi-Sugeno Models. *Lecture Notes in Computer Science. Fuzzy Sets and Systems*, v. 2715, p. 576–584, 2003.

ANGELOV, P.; FILEV, D. An approach to online identification of Takagi-Sugeno fuzzy models. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, v. 34, p. 484 – 498, 2004.

ANGELOV, P.; FILEV, D. SimpLeTS: A simplified method for learning evolving Takagi-Sugeno fuzzy models. In *Proceedings of the IEEE International Conference on Fuzzy Systems*, p. 1068 – 1073, 2005.

ANGELOV, P.; KASABOV, N. Evolving Computational Intelligence Systems. International Workshop on Genetic Fuzzy Systems, Granada, Spain, p. 76–82, 2005.

ANGELOV, P.; ZHOU, X. Evolving fuzzy systems from data streams in real-time. In *Proceedings of the International Symposium on Evolving Fuzzy Systems*, p. 29 – 35, 2006.

ARCHER, B. Forecasting Demand: Quantitative and Intuitive Techniques. *International Journal of Tourism Management*, v. 1, p. 5–12, 1980.

ARMSTRONG, J. *Principles of Forecasting: a Handbook for Researchs and Practitioners*. Boston: Kluwer Academic Publishers, 2001.

BABUSKA, R. *Fuzzy Modeling for Control*. New York: Springer Science Business Media, 1998.

- BABUSKA, R. *Fuzzy Algorithms for Control*. New York: Springer Science Business Media, 1999.
- BAIM, D. et al. Survival of patients with severe congestive heart failure treated with oral milrinone. *Journal American College of Cardiology*, v. 7, p. 661–670, 1986.
- BERTSEKAS, D. P.; TSITSIKLIS, J. N. *Introduction to Probability*. 2. ed. [S.l.]: Athena Scientific, 2008.
- BIREK, L.; PETROVIC, D.; BOYLAN, J. Water leakage forecasting: the application of a modified fuzzy evolving algorithm. *Applied Soft Computing*, v. 14, p. 305–315, 2014.
- BROCKWELL, P. J.; DAVIS, R. A. *Introduction to Time Series and Forecasting*. [S.l.]: Springer, 2002. ISBN 9780387953519.
- CHAMBERS, J.; MULLICK, S.; SMITH, D. How to Choose the Right Forecasting Technique. *Harvard Business Review*, v. 49, p. 45–57, 1971.
- CHEN, C.-T. *Linear System Theory and Design*. 3. ed. New York: Oxford University Press, 1999.
- CUI, Y.; SHI, J.; WANG, Z. Complex Rotation Quantum Dynamic Neural Networks (CRQDNN) using Complex Quantum Neuron (CQN): Applications to time series prediction. *Neural Networks*, v. 71, p. 11–26, 2015.
- DALKEY, C. *Delphi Method*. [S.l.]: RAND Corporation, Santa Monica, CA, 1967.
- DENG, C. *Time Series Decomposition Using Singular Spectrum Analysis*. [S.l.]: Department of Mathematics East Tennessee State University, 2014.
- DIETZ, T. Methods for Analyzing Data from Delphi Panels: Some Evidence from Forecasting Study. *Technological Forecasting and Social Change*, v. 31, p. 79–85, 1987.
- DUDA, R.; HART, P.; STORK, D. *Pattern Classification*. 2. ed. Chichester, West Sussex, England: Wiley-Interscience, 2000.
- GAN, M. et al. Seasonal and trend time series forecasting based on a quasi-linear autoregressive models. *Applied Soft Computing*, v. 24, p. 13–18, 2014.
- GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989. ISBN 0201157675.
- GOLYANDINA, N.; NEKRUTKIN, V.; ZHIGLJAVSKY, A. *Analysis of Time Series Structure: SSA and Related Techniques*. New York–London: Chapman & Hall/CRC, 2001.
- GOODRICH, R. The Forecast Pro methodology. *International Journal of Forecasting*, v. 16, p. 533–535, 2000.
- GOODWIN, P. Integrating management judgment and statistical methods to improve short-term forecasts. *Omega*, v. 30, p. 127–135, 2002.
- HAMILTON, J. D. *Time Series Analysis*. New Jersey: Pinceton University Press, 1994.
- HARVEY, A.; KOOPMAN, S. Unobserved components models in economics and finance: The role of the kalman filter in time series econometrics. *IEEE Control Systems Magazine*, v. 29, p. 71–81, 2009.



- HILAS, P. M. and C. A computational intelligence-based forecasting system for telecommunications time series. *Engineering Applications of Artificial Intelligence*, v. 25, p. 200–206, 2012.
- HSU, H. P. *Schaum's Outline of Theory and Problems of Probability, Random Variables, and Random Processes*. [S.l.]: McGraw-Hill, 1997.
- HYNDMAN, R. *Time Series Data Library*. 2016. <<https://datamarket.com/data/list/?q=provider%3Atsdl>>. Acessado em: 19/11/2016.
- HYNDMAN, R. J.; ATHANASOPOULOS, G. *Forecasting: principles and practice*. [S.l.]: OTexts, 2014. ISBN 9780987507105.
- KASABOV, N. Evolving connectionist systems for adaptive learning and knowledge discovery: method, tools, applications. *IEEE International Conference on Intelligent Systems*, v. 1, p. 24–28, 2002.
- KASABOV, N. *Evolving Connectionist Systems: The Knowledge Engineering Approach*. 2. ed. [S.l.]: Springer, 2007.
- KASABOV, N.; SONG, Q. Dynamic evolving fuzzy neural networks with "m-out-of-n" activation nodes for on-line adaptive systems. *Technical report, Department of Information Science.*, University of Otago, Dunedin, New Zealand, 1999.
- KASABOV, N.; SONG, Q. DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *IEEE Transactions on Fuzzy Systems*, v. 10, p. 144 – 154., 2002.
- KAYO, E.; SECURATO, J. *Método Delphi: fundamentos, críticas e vieses*. [S.l.]: Cadernos de Pesquisa em Administração, 1997. "1". 51-61 p.
- KHALEGHI, A. et al. Consistent Algorithms for Clustering Time Series. *Journal of Machine Learning Research*, v. 17, p. 1 – 32., 2016.
- KOCADAGLI, O.; ASIKGIL, B. Nonlinear time series forecasting with Bayesian neural networks. *Expert Systems with Applications*, v. 41, p. 6596–6610, 2014.
- KOMIJANI, M. et al. Introducing evolving Takagi-Sugeno method based on local least squares support vector machine models. *Evolving Systems*, v. 3, p. 81 – 93, 2012.
- KOPETZ, H. *Real-Time Systems: Design Principles for Distributed Embedded Applications*. 2nd. ed. [S.l.]: Springer Publishing Company, Incorporated, 2011. ISBN 1441982361, 9781441982360.
- LEMOS, A.; CAMINHAS, W.; GOMIDE, F. Fuzzy evolving linear regression trees. *Evolving Systems*, v. 2, p. 1 – 14, 2011.
- LEMOS, A.; CAMINHAS, W.; GOMIDE, F. Multivariable gaussian evolving fuzzy modeling system. *IEEE Transactions on Fuzzy Systems*, v. 19, p. 91–104, 2011.
- LEUNG, H. Prediction of noisy chaotic time series using an optimal radial basis function neural network. *IEEE Transactions in Neural Networks*, v. 12, p. 1163–1172, 2001.
- LI, D. Z.; WANG, W.; ISMAIL, F. An evolving fuzzy neural predictor for multi-dimensional system state forecasting. *Neurocomputing*, v. 145, p. 381–391, 2014.

- LIMA, E. et al. Evolving Fuzzy Modeling Using Participatory Learning. In: ANGELOV, P.; FILEV, D.; KASABOV, N. (Ed.). *Evolving Intelligent Systems: Methodology and Applications*. Hoboken, New Jersey: IEEE Press Series on Computational Intelligence. Published by John Wiley & Sons, 2010. cap. 2, p. 67 – 86.
- LIN, Y.; C., J. C.; LIN. A TSK-Type-Based self-evolving compensatory interval Type-2 fuzzy neural network (TSCIT2FNN) and its applications. *IEEE Transactions on Industrial Electronics*, v. 61, p. 447 – 459., 2014.
- LIN, Y.; CHANG, J.; LIN, C. Identification and prediction of dynamic systems using an interactively recurrent self-evolving fuzzy neural network. *IEEE Transactions on Neural Networks and Learning Systems*, v. 24, p. 310 – 321., 2013.
- LIU, Z. Chaotic Time Series Analysis. *Mathematical Problems in Engineering*, 2010.
- LJUNG, L. *System Identification: Theory for the User*. New Jersey, USA: Prentice-Hall, 1987.
- LUGHOFER, E. FLEXFIS: A robust incremental learning approach for evolving Takagi-Sugeno fuzzy models. *IEEE Transactions on Fuzzy Systems*, p. 1393 – 1410, 2008.
- LUGHOFER, E. *Evolving Fuzzy Systems: Methodologies, Advanced Concepts and Applications*. [S.l.]: Springer, 2011.
- LUGHOFER, E. Flexible evolving fuzzy inference systems from data streams (FLEXFIS++). *In Learning in Non-Stationary Environments*, p. 205 – 245, 2012.
- LUGHOFER, E. et al. Identifying static and dynamic prediction models for NO<sub>x</sub> emissions with evolving fuzzy systems. *Applied Soft Computing*, v. 11, p. 2487 – 2500, 2011.
- MAKRIDAKIS, S.; WHEELWRIGHT, S.; HYNDMAN, R. *Forecasting: Methods and Applications*. 3. ed. New York: John Wiley & Sons, 1998.
- MAMDANI, E. Application of fuzzy logic to approximate reasoning using linguistic systems. *Fuzzy Sets and Systems*, v. 26, p. 1182–1191, 1977.
- MONTGOMERY, D. C.; JENNINGS, C. L.; KULAHCI, M. *Introduction to Time Series Analysis and Forecasting*. 2. ed. [S.l.]: Wiley, 2015.
- OLIVEIRA, J. F. L. de; LUDERMIR, T. B. A hybrid evolutionary decomposition system for time series forecasting. *Neurocomputing*, v. 180, p. 27–34, 2016.
- PALIT, A. K.; POPOVIC, D. *Computational Intelligence in Time Series Forecasting: Theory and Engineering Applications (Advances in Industrial Control)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005. ISBN 1852339489.
- PAPOULIS, A.; PILLAI, S. U. *Probability, Random Variables, and Stochastic Processes*. 4. ed. [S.l.]: McGraw-Hill Higher Education, 2002.
- ROCHA, O.; SERRA, G. Adaptive Neuro-Fuzzy Black-Box Modeling Based on Instrumental Variable Evolving Algorithm. *Journal of Control, Automation and Electrical Systems*, 2016.
- RONG, H. et al. Sequential adaptive fuzzy inference system (SAFIS) for non-linear system identification and prediction. *Fuzzy Sets Systems*, v. 157, p. 1260 – 1275, 2006.

- SHAKER, A.; SENGE, R.; HULERMEIER, E. Evolving fuzzy pattern trees for binary classification on data streams. *Information Sciences*, v. 220, p. 34 – 45, 2013.
- STEPNICKA, M. et al. Forecasting seasonal time series with computational intelligence: On recent methods and the potential of their combinations. *Expert Systems with Applications*, v. 40, p. 1981–1992, 2013.
- SZMIT, M.; SZMIT, A. Usage of Modified Holt-Winters Method in the Anomaly Detection of Network Traffic: Case Studies. *Journal of Computer Networks and Communications*, v. 2012, 2012.
- TAKAGI, T.; SUGENO, M. Fuzzy identification of systems and its application to modeling and control. *IEEE Trans. Systems, Man and Cybernetics*, v. 15, p. 116–132, 1985.
- TEMIZEL, M. C. T. A comparative study of autoregressive neural network hybrids. *Neural Networks*, v. 18, p. 781–789, 2005.
- TRATAR, L. F.; STRMCNIK, E. The comparison of Holt-Winters method and Multiple regression method: A case study. *Energy*, v. 109, p. 266–276, 2016.
- TUNG, S.; QUEK, C.; GUAN, C. eT2FIS: An evolving Type-2 neural fuzzy inference system. *Information Sciences*, v. 220, p. 124 – 148., 2013.
- WALTER, E.; PRONZATO, L. *Identification of Parametric Models from Experimental Models*. [S.l.]: Springer, 1997.
- WANG, L. *A Course in Fuzzy Systems and Control*. [S.l.]: Prentice-Hall, 1997.
- WANG, W.; PEDRYCZ, W.; LIU, X. Time series long-term forecasting model based on information granules and fuzzy clustering. *Engineering Applications of Artificial Intelligence*, v. 41, p. 17–24, 2015.
- WANG, X.; HAN, M. Improved extreme learning machine for multivariate time series online sequential prediction. *Engineering Applications of Artificial Intelligence*, v. 40, p. 28–36, 2015.
- WU, S.; LEE, S. Employing local modeling in machine learning based methods for time-series prediction. *Expert Systems with Applications*, v. 42, p. 341–354, 2015.
- YAN, W. Toward automatic time-series forecasting using neural networks. *IEEE Transactions in Neural Networks*, v. 23, p. 1028–1039, 2012.
- YEN, J.; WANG, L.; GILLESPIE, C. Improving the Interpretability of TSK Fuzzy Models by Combining Global and Local Learning. *IEEE Transactions in Fuzzy Systems*, v. 6, p. 530 – 537, 1998.
- YOUNG, P. C. *Recursive Estimation and Time-Series Analysis: An Introduction for the Student and Practitioner*. 2nd. ed. [S.l.]: Springer Publishing Company, 2014. ISBN 3642441297, 9783642441295.
- ZADEH, L. Fuzzy Sets. *Information and Control*, v. 8, p. 338–353, 1965.
- ZHANG, G.; QI, M. Neural network forecasting for seasonal and trend time series. *European Journal of Operational Research*, v. 160, p. 501–514, 2005.

ZHANG, J.; CHUNG, H. S.; WAI-LUN, L. Chaotic time series prediction using a neuro-fuzzy system with time-delay coordinates. *IEEE Transactions on Knowledge and Data Engineering*, v. 20, p. 956–964, 2008.

## **Apêndices**



## APÊNDICE A – FUNDAMENTOS SOBRE SISTEMAS FUZZY

A teoria de sistemas *fuzzy* ou nebulosos foi desenvolvida para modelar o conhecimento no formato de regras SE-ENTÃO. Portanto, diz-se que um sistema *fuzzy* é constituído de uma coleção de regras. De acordo com Wang (1997), a teoria de sistemas *fuzzy* se justifica devido a necessidade de descrever a complexidade do mundo real de forma aproximada e também para representar o conhecimento humano matematicamente, aproveitando essas informações valiosas de especialistas em aplicações práticas na engenharia.

### A.1 Conjuntos Fuzzy

Em primeiro lugar, é importante definir o que são conjuntos *fuzzy*. Diferentemente da teoria de conjuntos tradicional, os elementos podem pertencer parcialmente a um conjunto *fuzzy*. Assim, os conjuntos *fuzzy* classificam os elementos de um dado universo de forma menos rígida que na teoria clássica, admitindo graus parciais de pertinência a um conjunto *fuzzy* (ZADEH, 1965). Logo, a função de pertinência  $\mu$  para um elemento  $x$  pode possuir valores no seguinte intervalo:

$$0 \leq \mu_A(x) \leq 1, \quad (\text{A.1})$$

sendo que  $A$  é um conjunto *fuzzy* no universo de discurso  $U$ . Esse conjunto *fuzzy* também pode ser representado pelo conjunto de pares ordenados a seguir:

$$A = \{(x, \mu_A(x)) | x \in U\}. \quad (\text{A.2})$$

Quando  $U$  está no tempo contínuo ou no discreto, o conjunto  $A$  também é descrito por:

$$A = \int_U \frac{\mu_A(x)}{x} \quad (\text{Contínuo}) \quad A = \sum_U \frac{\mu_A(x)}{x} \quad (\text{Discreto}). \quad (\text{A.3})$$

onde os operadores de integração e somatório significam a coleção de todos os pontos  $x \in U$  com função de pertinência associadas. A soma das funções de pertinência de todas as regras *fuzzy* deve ser igual a 1:

$$\sum_{i=1}^R \mu^i = 1 \quad (\text{A.4})$$

As funções de pertinência são definidas em uma base de dados que juntamente com o conjunto de regras formam a base de conhecimento dos sistemas *fuzzy*. Essas funções podem ser definidas pelo especialista através de seu conhecimento prévio ou formulada automaticamente a

partir dos dados disponíveis de entrada e saída do sistema. As funções de pertinência podem estar representadas de diferentes formas, dependendo do conceito que se deseja representar e do contexto que estão inseridas (ZADEH, 1965) (WANG, 1997) (BABUSKA, 1998) (BABUSKA, 1999). Os tipos mais comuns são:

- triangular;
- trapezoidal;
- gaussiana;
- Cauchy;
- entre outros.

A seguir, algumas características básicas de conjuntos *fuzzy* serão apresentadas, segundo as referências em Zadeh (1965), Wang (1997), Babuska (1998) e Babuska (1999).

**Suporte:** o suporte de um conjunto *fuzzy*  $A$  no universo de discurso  $U$  contém todos os elementos que possuem funções de pertinência não-nulas, ou seja:

$$\text{supp}(A) = \{x \in U \mid \mu_A(x) > 0\}. \quad (\text{A.5})$$

**Altura:** é o maior valor para a função de pertinência. Se esse valor for igual a 1, diz-se que o conjunto *fuzzy* é normal.

**Corte- $\alpha$ :** é um conjunto  $A_\alpha$  que contém todos os elementos em  $U$  que apresentam valores de pertinência em  $A$  maiores ou igual a  $\alpha$ :

$$A_\alpha = \{x \in U \mid \mu_A(x) > \alpha\}. \quad (\text{A.6})$$

## A.2 Operações com Conjuntos *Fuzzy*

### A.2.1 Complemento *Fuzzy*

Qualquer função que realiza o mapeamento  $c : [0, 1] \rightarrow [0, 1]$  e que satisfaz as seguintes condições é uma operação de complemento *fuzzy*. Seja a função a seguir:

$$c[\mu_A(x)] = \mu_{\bar{A}}(x) \quad (\text{A.7})$$

sendo que, para essa função seja complemento, ela deve satisfazer ambas condições abaixo:

- Condição de fronteira:  $c(0) = 1$  e  $c(1) = 0$ ;



- Condição de não-crescimento: para todo  $a, b \in [0, 1]$  sendo funções de pertinência, se  $a < b$  então  $c(a) > c(b)$ .

### A.2.2 União Fuzzy

Qualquer função que realiza o mapeamento  $s : [0, 1] \times [0, 1] \rightarrow [0, 1]$  e que satisfaz as seguintes condições é uma operação de união *fuzzy* ou norma-s. Seja a função representada por:

$$s[\mu_A(x), \mu_B(x)] = \mu_{A \cup B}(x) \quad (\text{A.8})$$

sendo que, para essa função seja uma norma-s, ela deve satisfazer as condições abaixo, para  $a$  e  $b$  funções de pertinência:

- Condição de fronteira:  $s(1, 1) = 1$  e  $s(0, a) = s(a, 0) = a$ ;
- Condição comutativa:  $s(a, b) = s(b, a)$ ;
- Condição de não-decrescimento: se  $a \leq a'$  e  $b \leq b'$  então  $s(a, b) \leq s(a', b')$ ;
- Condição associativa:  $s(s(a, b), c) = s(a, s(b, c))$ .

### A.2.3 Intersecção Fuzzy

Qualquer função que realiza o mapeamento  $t : [0, 1] \times [0, 1] \rightarrow [0, 1]$  e que satisfaz as propriedades a seguir é uma operação de intersecção *fuzzy* ou norma-t. Seja a função representada por:

$$t[\mu_A(x), \mu_B(x)] = \mu_{A \cap B}(x) \quad (\text{A.9})$$

sendo que, para essa função seja uma norma-t, ela deve satisfazer todas as seguintes propriedades, para  $a$  e  $b$  funções de pertinência:

- Condição de fronteira:  $t(0, 0) = 0$  e  $t(a, 1) = t(1, a) = a$ ;
- Condição comutativa:  $t(a, b) = t(b, a)$ ;
- Condição de não-decrescimento: se  $a \leq a'$  e  $b \leq b'$  então  $t(a, b) \leq t(a', b')$ ;
- Condição associativa:  $t(t(a, b), c) = t(a, t(b, c))$ .

### A.3 Variáveis Linguísticas e Regras Fuzzy

Uma variável linguística possui como valores palavras em linguagem natural, sendo que cada uma dessas palavras ou termos linguísticos são descritos por conjuntos *fuzzy* e funções de pertinência. Uma variável linguística é caracterizada pela tupla  $(X, T, U, M)$ , em que (WANG, 1997):

- $X$  é o nome da variável linguística;
- $T$  corresponde ao conjunto de valores linguísticos que  $X$  pode ter;
- $U$  é o universo de discurso;
- $M$  são as regras semânticas que relaciona cada valor linguístico em  $T$  com um conjunto *fuzzy* linguístico em  $U$ .

As regras *fuzzy* podem ser representadas no seguinte formato:

**SE** proposição antecedente **ENTÃO** proposição consequente.

Há dois tipos de proposições tanto antecedente quanto consequente: as proposições atômicas e as compostas. As proposições atômicas são apenas uma simples declaração do tipo  $x$  é  $A$ , sendo que  $x$  é a variável linguística e  $A$  é um valor linguístico de  $x$ , isto é, um conjunto *fuzzy*. Por sua vez, as proposições compostas são constituídas por duas ou mais declarações simples acima citadas, relacionados pelos conectivos **E**, **OU** e **NÃO**, que representam as operações de intersecção (norma-t), união (norma-s) e complemento, respectivamente. As regras para cada tipo de conectivo são descritas a seguir (WANG, 1997):

- Conectivo **E** - se  $x$  e  $y$  são variáveis linguísticas nos domínios  $U$  e  $V$ , e  $A$  e  $B$  são conjuntos *fuzzy* nesses domínios, então a proposição composta:

$$x \text{ é } A \text{ E } y \text{ é } B, \quad (\text{A.10})$$

é interpretada pela relação  $A \cap B$  com função de pertinência e qualquer norma-t pela expressão:

$$\mu_{A \cap B}(x, y) = t[\mu_A(x), \mu_B(y)] \quad (\text{A.11})$$

- Conectivo **OU** - utilizando os mesmos termos apresentados acima, a proposição composta:

$$x \text{ é } A \text{ OU } y \text{ é } B, \quad (\text{A.12})$$

é interpretada pela relação  $A \cup B$  com função de pertinência e qualquer norma-s pela expressão:

$$\mu_{A \cup B}(x, y) = s[\mu_A(x), \mu_B(y)] \quad (\text{A.13})$$

- Conectivo **NÃO** - aplica-se as operações de complemento. Seja a proposição:

$$x \text{ NÃO } \acute{e} A \text{ E } y \acute{e} B, \quad (\text{A.14})$$

ela é representada pela função de pertinência e qualquer norma-t pela expressão:

$$\mu(x, y) = t[c(\mu_A(x)), \mu_B(y)]. \quad (\text{A.15})$$

O mapeamento entre as entradas e saídas é realizado pelo mecanismo de inferência *fuzzy*, que utiliza essa base de conhecimento para gerar os valores de saída. Entretanto, na maioria das aplicações práticas, as entradas e saídas são valores numéricos e não linguísticos. Para lidar com esse obstáculo, os sistemas *fuzzy* podem utilizar interfaces para converter valores numéricos de entrada em valores linguísticos (*fuzzificador*) e vice-versa, de valores linguísticos para saídas em valores numéricos (*defuzzificador*) (BABUSKA, 1998) (WANG, 1997).

#### A.4 Sistemas Fuzzy Takagi-Sugeno

Há diversos tipos de regras *fuzzy*, porém, duas categorias são mais amplamente utilizadas:

- Modelo *Fuzzy* Linguístico (MAMDANI, 1977): antecedente e consequente são formados por variáveis linguísticas e conjuntos *fuzzy*;
- Modelo *Fuzzy* Takagi-Sugeno (TAKAGI; SUGENO, 1985): a proposição consequente da regra é formada por equações matemáticas.

Os sistemas *fuzzy* do tipo Takagi-Sugeno possui uma natureza dual, combinando uma parte antecedente que é linguística com uma parte consequente representada por equações matemáticas geralmente lineares. Segundo Wang (1997), a grande vantagem desses sistemas é que este lida com entradas e saídas de valores reais, condições encontradas na maioria dos problemas práticos de engenharia. No entanto, devido a sua natureza matemática, os consequentes das regras não fornecem uma forma natural de representar o conhecimento humano e, além disso, o sistema *fuzzy* perde um pouco de sua versatilidade. A estrutura dos sistemas desse tipo pode ser descrita por (BABUSKA, 1998):

$$\begin{aligned} \text{Regra}_i : \quad & \text{SE } x^1 \acute{e} A_i^1 \text{ E } x^2 \acute{e} A_i^2 \text{ E } \dots \text{ E } x^n \acute{e} A_i^n \\ & \text{ENTÃO } y_i = f_i(\mathbf{x}), \quad i = 1, 2, \dots, R. \end{aligned} \quad (\text{A.16})$$

onde  $\mathbf{x} = [x^1 \ x^2 \ \dots \ x^n]$  corresponde a variável de entrada no antecedente das regras,  $y_i \in \mathbb{R}$  é a variável de saída do consequente,  $R$  é a quantidade de regras *fuzzy* e  $A_i$  são os conjuntos *fuzzy* dos antecedentes, representados pelas funções de pertinência:

$$\mu_{A_i}(\mathbf{x}) : \mathbb{R}^n \rightarrow [0, 1]. \quad (\text{A.17})$$

As funções do consequente podem ter diversas naturezas matemáticas, contudo, tradicionalmente é representada pelo modelo a seguir:

$$y_i = \mathbf{a}_i^T \mathbf{x} + b_i, \quad (\text{A.18})$$

sendo que  $\mathbf{a}_i^T$  é um vetor de parâmetros relacionados a regra  $i$  e  $b_i$  é um escalar. Quando  $b_i = 0$ , o modelo é dito homogêneo, por outro lado, quando  $\mathbf{a}_i^T \mathbf{x} = 0$ , a equação do consequente é representada somente pelo escalar  $b_i$ , nomeando o modelo de *singleton*.

Antes de inferir o valor da saída, o grau de ativação de cada regra denotado por  $\tau_i$  precisa ser conhecido. Para sistemas com múltiplas entradas representadas pelo vetor  $\mathbf{x}$ , o grau de ativação é igual ao grau de pertinência dessas respectivas entradas, ou seja,  $\tau_i = \mu_{A_i}(\mathbf{x})$ . Esse grau de pertinência é calculado por meio da combinação das funções de pertinência para cada entrada individual, de acordo com os conectivos utilizados nas regras (e, ou, not). Considerando a regra com conectivo **E** da Equação A.16, o grau de ativação pode ser calculado utilizando a seguinte relação (BABUSKA, 1998) (WANG, 1997):

$$\tau_i = t \left\{ \mu_i^j(x^j) \right\} = \prod_{j=1}^n \mu_i^j(x^j), \quad i = 1, 2, \dots, R. \quad (\text{A.19})$$

Sendo assim, no modelo Takagi-Sugeno, a inferência é reduzida a uma simples expressão algébrica denotada por:

$$y = \frac{\sum_{i=1}^R \tau_i y_i}{\sum_{i=1}^R \tau_i}. \quad (\text{A.20})$$

Definindo-se o grau de ativação normalizado de cada regra *fuzzy* por:

$$\lambda_i = \frac{\tau_i}{\sum_{i=1}^R \tau_i}, \quad (\text{A.21})$$

portanto, a saída do modelo Takagi-Sugeno também pode ser expressada pela seguinte equação:

$$y = \left( \sum_{i=1}^R \lambda_i \mathbf{a}_i^T \right) \mathbf{x} + \sum_{i=1}^R \lambda_i b_i \quad (\text{A.22})$$

## APÊNDICE B – PRINCÍPIOS DA TEORIA DA PROBABILIDADE

Em geral, há dois tipos de fenômenos na natureza: os determinísticos e os aleatórios. Os primeiros são aqueles em que os resultados são sempre os mesmos, qualquer que seja o número de ocorrências. Já os fenômenos aleatórios não são previsíveis, gerando diferentes resultados mesmo que as condições iniciais do experimento sejam iguais. Existem processos que são natureza que são puramente aleatórios, como jogos de azar, porém, há muitos outros que não são aleatórios, sendo que esses processos são tratados como aleatórios devido a nossas limitações em entendê-los e descrevê-los. Por exemplo, os fenômenos climáticos não são aleatórios, mas são bastante complexos pois envolvem inúmeras variáveis e condições (ANGELOV, 2013).

De acordo com Papoulis e Pillai (2002), o objetivo da teoria da probabilidade é descrever e prever os eventos aleatórios em termos de probabilidades, isto é, desenvolver um modelo probabilístico para representar matematicamente situações incertas. Se um determinado experimento ocorre  $n$  vezes e um evento  $A$  tem  $n_A$  ocorrências, então, com um alto grau de certeza, a frequência relativa do evento  $A$  é próxima do valor da probabilidade  $P(A)$  para um elevado valor para  $n$ , ou seja, a lei de probabilidade do evento  $A$  é denotada por:

$$P(A) \simeq \frac{n_A}{n}, \quad (\text{B.1})$$

sendo que o valor de  $n$  deve ser suficientemente grande para que esta proposição seja mais confiável.

O problema de probabilidade, segundo Bertsekas e Tsitsiklis (2008), pode ser geralmente dividido em três etapas:

1. A descrição do espaço amostral;
2. A especificação da lei de probabilidade, ou seja, a probabilidade de cada evento;
3. O cálculo das probabilidades e probabilidades condicionais de vários eventos de interesse.

O espaço amostral de um experimento aleatório pode ser definido como o conjunto de todos os seus resultados possíveis. Os subconjuntos desse espaço amostral são os denominados eventos, sendo que o conjunto vazio é o evento impossível, com probabilidade  $P(\emptyset)$  igual a zero. Dois eventos  $A$  e  $B$  são iguais se consistirem nos mesmos elementos, logo,  $P(A) = P(B)$  (HSU, 1997). As operações da teoria dos conjuntos são aplicáveis para esses eventos, tais como reunião, intersecção e complemento. A lei de probabilidade de um evento  $P(A)$  deve satisfazer os seguintes axiomas (BERTSEKAS; TSITSIKLIS, 2008):

1. (Não-negativo)  $P(A) \geq 0$ , para todo evento  $A$ ;

2. (Aditividade) Seja um espaço amostral com vários eventos disjuntos, então a probabilidade da união desses eventos é  $P(A_1 \cup A_2 \cup \dots) = P(A_1) + P(A_2) + \dots$ ;
3. (Normalização) A probabilidade de todo o espaço amostral é igual a 1:  $P(\Omega) = 1$ .

A probabilidade condicional de um evento  $A$ , dado evento  $B$  com  $P(B) > 0$ , é definida por (PAPOULIS; PILLAI, 2002):

$$P(A|B) = \frac{P(A \cap B)}{P(B)}. \quad (\text{B.2})$$

Na prática, é comum usar um número associado a um evento aleatório e calcular a ocorrência desse número. Logo, dá-se o nome de variável aleatória a função que associa a todo evento pertencente a uma partição do espaço amostral um único número real (PAPOULIS; PILLAI, 2002). Para caracterizar uma variável aleatória  $X$ , o comportamento da sua função distribuição de probabilidades  $F_X(x)$  deve ser caracterizado, sendo assim (HSU, 1997):

$$F_X(x) = P(X \leq x), \quad -\infty < x < \infty, \quad 0 \leq F_X(x) \leq 1. \quad (\text{B.3})$$

para  $x$  sendo uma ocorrência da variável aleatória  $X$ .

Uma variável aleatória é dita discreta se o conjunto de seus valores for finito ou infinitamente contável, por sua vez, uma variável aleatória contínua pode assumir infinitos valores não-contáveis. A probabilidade de cada valor numérico que uma variável aleatória discreta pode assumir é analisada através da função massa de probabilidade (BERTSEKAS; TSITSIKLIS, 2008). Essa função é dada por (HSU, 1997):

$$p_X(x) = P(X = x),$$

$$0 \leq p_X(x_k) \leq 1, \quad (\text{B.4})$$

$$\sum_k p_X(x_k) = 1, \quad k = 1, 2, \dots,$$

e está relacionada com a função distribuição de probabilidades da seguinte forma (HSU, 1997):

$$F_X(x) = P(X \leq x) = \sum_{x_k \leq x} p_X(x_k). \quad (\text{B.5})$$

De forma análoga ao conceito de função massa de probabilidade, tem-se a função densidade de probabilidades para variáveis aleatórias contínuas, sendo calculada por (HSU,

1997):

$$f_X(x) = \frac{dF_X(x)}{dx},$$

$$f_X(x) \geq 0, \quad (\text{B.6})$$

$$\int_{-\infty}^{\infty} f_X(x) dx = 1.$$

Alguns parâmetros importantes para ambas funções acima citadas são a média, o momento e a variância (HSU, 1997). A média ou o valor esperado de uma variável aleatória  $X$  é definida por:

$$E(X) = \mu_X = \begin{cases} \sum_k x_k p_X(x_k), & \text{para } X \text{ discreto;} \\ \int_{-\infty}^{\infty} x f_X(x) dx, & \text{para } X \text{ contínuo.} \end{cases} \quad (\text{B.7})$$

O  $n$ -ésimo momento de uma variável aleatória  $X$  é representado por:

$$E(X^n) = \begin{cases} \sum_k x_k^n p_X(x_k), & \text{para } X \text{ discreto;} \\ \int_{-\infty}^{\infty} x^n f_X(x) dx, & \text{para } X \text{ contínuo.} \end{cases} \quad (\text{B.8})$$

sendo que é possível notar que a média é o primeiro momento. Por fim, a variância de  $X$  é definida por:

$$\text{Var}(X) = E[X - E(X)]^2 = \sigma^2 = \begin{cases} \sum_k (x_k - \mu_X)^2 p_X(x_k), & \text{para } X \text{ discreto;} \\ \int_{-\infty}^{\infty} (x - \mu_X)^2 f_X(x) dx, & \text{para } X \text{ contínuo.} \end{cases} \quad (\text{B.9})$$

Frequentemente, são consideradas funções densidade padronizadas e conhecidas para caracterizar variáveis aleatórias por conveniência. O tipo mais comum é a função gaussiana ou normal, denotada por (PAPOULIS; PILLAI, 2002):

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu_X)^2/2\sigma^2}. \quad (\text{B.10})$$

Outro tipo de função densidade especial é a função Cauchy, definida por (PAPOULIS; PILLAI, 2002):

$$f_X(x) = \frac{1}{\pi\gamma \left(1 + \left(\frac{x-x_0}{\gamma}\right)^2\right)}. \quad (\text{B.11})$$

Há ainda inúmeros outros padrões que podem ser encontrados em Papoulis e Pillai (2002), Bertsekas e Tsitsiklis (2008) e Hsu (1997), como as funções de Bernoulli, Binomial, Poisson, Uniforme e Exponencial. Entretanto, as funções densidade de probabilidades de dados reais são geralmente complexas, multimodais e variáveis com o tempo, apresentando comportamento bem diferente dessas funções padronizadas. Surge então a necessidade de estimar essas funções densidade de probabilidade a partir dos dados disponíveis. Um método que possui esse objetivo é a técnica de Estimção de Densidade Kernel (*Kernel Density Estimation* - KDE). Seja um conjunto de dados denotados por  $x_1, x_2, x_3, \dots, x_n$  com função densidade de probabilidades desconhecida. O estimador de densidade Kernel é representado por:

$$\hat{f}_X(x) = \frac{1}{n} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right), \quad (\text{B.12})$$

onde  $K()$  é a função kernel que é simétrica mas não necessariamente positiva e  $h > 0$  é o raio desse kernel. Uma gama de tipos de funções kernel podem ser usadas, como Gaussiana, Cauchy e Epanechnikov, cada qual com sua formulação específica. O método KDE tem apenas um parâmetro a ser determinado, que é o valor de  $h$ , que influencia bastante nos resultados. A principal desvantagem desse método é sua elevada carga computacional e por sua execução *offline*, isto é, é uma abordagem em batelada. Devido a essas limitações, Angelov e Filev (2003) propuseram uma técnica *online* e recursiva para estimação dessa densidade, conhecida como Estimção Recursiva da Densidade, do inglês *Recursive Density Estimation* (RDE).

No ponto de vista de Angelov (2013), a teoria da probabilidade fornece um conjunto de conceitos e procedimentos para descrever processos aleatórios e dados, porém, essa teoria se baseia em condições que muito raramente acontecem na prática, como por exemplo, a natureza aleatória dos eventos, a exigência de  $n \rightarrow \infty$  experimentos, a independência entre os dados e funções distribuição padronizadas. Na situação em que tiver poucos experimentos realizados, isto é, para  $n$  pequeno, as leis da teoria da probabilidade não são aplicáveis. Portanto, o uso dessa teoria deve ser realizado com cautela, verificando se os princípios e conclusões são válidos de acordo com as condições sendo estudadas.