

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 DE MODELAGEM COMPUTACIONAL INTELIGENTE PARA
PREVISÃO DE SÉRIES TEMPORAIS BASEADA EM SISTEMAS EVOLUTIVOS E
ANÁLISE ESPECTRAL SINGULAR RECURSIVA**

São Luís - MA

2021

SELMO EDUARDO RODRIGUES JÚNIOR

**METODOLOGIA DE MODELAGEM COMPUTACIONAL INTELIGENTE PARA
PREVISÃO DE SÉRIES TEMPORAIS BASEADA EM SISTEMAS EVOLUTIVOS E
ANÁLISE ESPECTRAL SINGULAR RECURSIVA**

Tese de Doutorado apresentada à Banca Examinadora como requisito para obtenção do título de Doutor em Engenharia de Eletricidade na área de concentração de Automação e Controle do Programa de Pós-Graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão.

Orientador: Dr. Ginalber Luiz de Oliveira Serra

São Luís - MA

2021

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

Rodrigues Júnior, Selmo Eduardo.

Metodologia de Modelagem Computacional Inteligente para
Previsão de Séries Temporais Baseada Em Sistemas
Evolutivos e Análise Espectral Singular Recursiva / Selmo
Eduardo Rodrigues Júnior. - 2021.

222 p.

Orientador(a): Ginalber Luiz de Oliveira Serra.

Tese (Doutorado) - Programa de Pós-graduação em
Engenharia Elétrica/ccet, Universidade Federal do
Maranhão, São Luís, 2021.

1. Análise Espectral Singular. 2. Componentes Não-
Observáveis. 3. Modelos Evolutivos. 4. Previsão de
Séries Temporais. 5. Redes Neuro-Fuzzy. I. de Oliveira
Serra, Ginalber Luiz. II. Título.

SELMO EDUARDO RODRIGUES JÚNIOR

**METODOLOGIA DE MODELAGEM COMPUTACIONAL INTELIGENTE PARA
PREVISÃO DE SÉRIES TEMPORAIS BASEADA EM SISTEMAS EVOLUTIVOS E
ANÁLISE ESPECTRAL SINGULAR RECURSIVA**

Tese de Doutorado apresentada à Banca Examinadora como requisito para obtenção do título de Doutor em Engenharia de Eletricidade na área de concentração de Automação e Controle do Programa de Pós-Graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão.

Trabalho aprovado em São Luís - MA, 10 de agosto de 2021.

Dr. Ginalber Luiz de Oliveira Serra (IFMA)
Orientador

Dr. Celso José Munaro (UFES)
Primeiro Membro da Banca

Dr. Mateus Giesbrecht (UNICAMP)
Segundo Membro da Banca

Dr. Francisco das Chagas de Souza (UFMA)
Terceiro Membro da Banca

Dr. Anselmo Cardoso de Paiva (UFMA)
Quarto Membro da Banca

*Dedico este trabalho a Deus, à minha esposa
aos meus pais, irmã e a todos meus familiares,
pois sem eles eu não teria capacidade para desenvolver esta tese.*

AGRADECIMENTOS

Agradeço a todos aqueles que, diretamente ou indiretamente, fizeram parte desse trabalho e contribuíram para que esta tese se concretizasse.

Agradeço a Deus por ter me iluminado, me guiado por esse caminho cheio de desafios e por ter me dado força e saúde para superar as adversidades.

Agradeço muito ao professor Dr. Ginalber Luiz de Oliveira Serra pela orientação e por ter acreditado em mim e nas minhas capacidades, sempre trabalhando de forma positiva, me dando forças e muita confiança para 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.

À minha esposa Emilly Mota por todo seu companheirismo, amor e compreensão que foram essenciais para eu pudesse desenvolver essa pesquisa.

À Universidade Federal do Maranhão e à CAPES pelo apoio estrutural, suporte e financeiro durante essa trajetória de doutorado para que essa pesquisa fosse desenvolvida.

Aos professores e profissionais do Instituto Federal do Maranhão, em especial ao Campus Imperatriz, por terem me dado todo apoio para desenvolver esse trabalho. Agradeço ainda a todas as pessoas com que convivi ao longo desses anos, sendo que a comunhão com os amigos e o aprendizado em grupo foram experiências enriquecedoras na minha formação.

A todos, os meus sinceros agradecimentos.

*"A vitalidade é demonstrada não apenas pela persistência,
mas pela capacidade de começar de novo."
(F. Scott Fitzgerald)*

RESUMO

Essa tese propõe uma metodologia evolutiva para previsão de séries temporais univariáveis ou multivariáveis baseada em uma estrutura de rede neuro-*fuzzy*, capaz de atualizar sua base de conhecimento para cada nova observação que chega. Essa proposta possui uma característica híbrida, pois utiliza uma rede neuro-*fuzzy* cujas regras *fuzzy* contêm em seus consequentes modelos de vetores autorregressivos (VAR). Esses modelos são constituídos pelas componentes não-observáveis, ou seja, são padrões ocultos extraídos a partir da série temporal. Para extrair essas componentes, uma versão recursiva e paralela do método SSA (*Singular Spectrum Analysis*) é proposta no trabalho. Denominou-se essa versão de PRSSA (*Parallel Recursive Singular Spectrum Analysis*), sendo que o nome paralelo está relacionado ao fato que, para cada série, há um procedimento de decomposição associado. Unificando esses métodos, a metodologia proposta tem o nome de PRSSA+ENFN (*Parallel Recursive Singular Spectrum Analysis and Evolving Neuro-Fuzzy Network*), destacando novamente seu perfil híbrido. Dessa maneira, o método PRSSA+ENFN proposto nessa pesquisa aplica a abordagem "dividir para conquistar", isto é, extrai as componentes não-observáveis e realiza suas previsões separadamente por meio da rede neuro-*fuzzy*, pois estas componentes apresentam comportamento menos complexo que a série temporal completa. Após a previsão das componentes, esses resultados são agrupados para prever a série temporal original. Ademais, a rede neuro-*fuzzy* tem característica evolutiva, ou seja, considera o comportamento dinâmico dessas componentes para evoluir sua estrutura a cada nova observação, cuja quantidade de regras *fuzzy* pode aumentar ou diminuir de acordo com essa dinâmica. Avaliou-se nos resultados experimentais a flexibilidade do método PRSSA+ENFN em trabalhar tanto com séries univariáveis quanto multivariáveis, realizando previsões do tipo recorrente ou direta. Além disso, a abordagem proposta foi comparada com outros trabalhos e métodos utilizados na literatura para previsão de séries temporais, apresentando resultados competitivos de previsão e que mostraram que a metodologia desenvolvida pode ser usada em casos envolvendo séries temporais complexas e não-estacionárias. Realizou-se também a aplicação do PRSSA+ENFN em um problema real relacionado à pandemia do Covid-19 no estado do Maranhão - Brasil, avaliando seu comportamento para previsão dos números diários de casos e mortes devido a essa doença. Dessa forma, esse método proposto se mostrou como uma potencial abordagem a fim de auxiliar os especialistas no processo de tomada de decisão e para continuar a ser explorada em trabalhos futuros.

Palavras-chaves: Redes Neuro-*Fuzzy*; Previsão de Séries Temporais; Modelos Evolutivos; Análise Espectral Singular; Componentes Não-Observáveis.

ABSTRACT

This thesis proposes an evolving methodology for univariate or multivariate time series forecasting based on neuro-fuzzy network structure, updating its knowledge base for each new observation. This proposal has a hybrid characteristic, considering a neuro-fuzzy network whose consequents of fuzzy rules contain vector autoregression models. These models are composed by unobservable components, i. e., hidden patterns extracted from the time series. To extract these components, a recursive and parallel version of the Singular Spectrum Analysis (SSA) method is proposed in the paper. This version is named Parallel Recursive Singular Spectrum Analysis (PRSSA), and the parallel name is used because, for each time series, there is an associated decomposition procedure. Unifying these methods, the proposed methodology is called PRSSA+ENFN (Parallel Recursive Singular Spectrum Analysis and Evolving Neuro-Fuzzy Network), highlighting its hybrid profile again. Hence, the PRSSA+ENFN method proposed applies the "divide to conquer" idea, i.e., it extracts the unobservable components and forecasts them separately using the neuro-fuzzy network, because these components has less complex behavior than the complete time series. After components forecasting, these results are grouped together to predict the original time series. Furthermore, the neuro-fuzzy network has an evolving characteristic, i.e., it considers the dynamic behavior of these components to evolve its structure for each new observation, where the number of fuzzy rules can increase or decrease according to this dynamic. The flexibility of the PRSSA+ENFN method with both univariate or multivariate time series was evaluated on the experimental results, considering the recurrent or direct type of forecasting. In addition, the proposed approach was compared with other studies and methods used in literature for time series forecasting, showing competitive results and the developed methodology can be used in cases involving complex and non-stationary time series. The PRSSA+ENFN was applied to a real problem related to the Covid-19 pandemic in Maranhão - Brazil, evaluating its behavior to forecast the daily numbers of cases and deaths caused by this disease. Then, this proposed method proved to be a potential approach to assist specialists in the decision-making process and to be explored in future works.

Keywords: Neuro-Fuzzy Networks; Time Series Forecasting; Evolving Models; Singular Spectrum Analysis; Unobservable Components.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de Série Temporal: Quantidade de Passageiros em Milhares em uma Companhia Aérea Internacional.	31
Figura 2 – Séries Temporais para Análise da Estacionariedade.	35
Figura 3 – Funções de Autocorrelação Amostral das Séries da Figura 2. <i>Lag</i> é o atraso h e ACF significa <i>Autocorrelation Function</i>	36
Figura 4 – Série Temporal Sazonal e sua Função de Autocorrelação Amostral.	37
Figura 5 – Exemplo de Previsão de uma Série Temporal Usando os Métodos Simples Apresentados ($H = 12$).	50
Figura 6 – Exemplo de Previsão de uma Série Temporal Usando os Métodos Holt-Winters e Holt ($H = 12$).	53
Figura 7 – Série Temporal Venda de Livros para Estudo da Análise Espectral Singular.	66
Figura 8 – Exemplo de Componentes Não-Observáveis e Efeito da Variação do Parâmetro L para a Série Temporal de Vendas de Livros	67
Figura 9 – Valores Singulares para as Decomposições da Figura 8.	68
Figura 10 – Série Temporal de Vendas de Livros em Comparação com a Soma das Componentes Não-Observáveis.	69
Figura 11 – Matriz W -Correlação para as Decomposições da Figura 8.	70
Figura 12 – Representação do Modelo <i>Fuzzy</i> Evolutivo Takagi-Sugeno por meio de uma Rede Neuro- <i>Fuzzy</i> de 6 Camadas.	82
Figura 13 – Processo de Aprendizagem dos Modelos <i>Fuzzy</i> Evolutivos.	83
Figura 14 – Exemplo da Atualização de <i>Clusters</i> Devido aos Novos Dados.	85
Figura 15 – Exemplo da Adição de <i>Clusters</i> Devido aos Novos Dados.	86
Figura 16 – Exemplo da Fusão de <i>Clusters</i> Devido aos Novos Dados.	87
Figura 17 – Relação entre as Etapas do PRSSA+ENFN: Etapas de Pré-processamento, Treinamento Incremental e Previsão.	97
Figura 18 – Extração das Componentes Não-Observáveis na Etapa de Pré-processamento. Cada Série Temporal l ($l = 1, \dots, m$) possui um Bloco SSA Exclusivo (Característica Paralela).	99
Figura 19 – Extração das Componentes Não-Observáveis na Etapa de Treinamento Incremental. Cada Série Temporal ($l = 1, \dots, m$) possui um Bloco Exclusivo do SSA Recursivo ou RSSA (Característica Paralela do PRSSA).	106
Figura 20 – Relação dos Procedimentos da Etapa de Pré-Processamento e de Treinamento Incremental do PRSSA para a Série Temporal l	107
Figura 21 – Estrutura Geral da ENFN na Etapa de Treinamento.	108
Figura 22 – Fluxograma para Aplicação das Condições 1 e 2.	112
Figura 23 – Previsão Recorrente do Método PRSSA+ENFN.	115

Figura 24 – Previsão Direta do Método PRSSA+ENFN.	116
Figura 25 – Série Temporal Mackey-Glass (Equação 6.1).	124
Figura 26 – Resultados de Previsão Direta e Recorrente da Série Mackey-Glass Relacionados às Tabelas 4 e 5.	127
Figura 27 – Histograma das Medidas RMSE dos Erros da Etapa de Previsão do PRSSA+ENFN da População Final do Algoritmo Genético (Tabela 6).	129
Figura 28 – Conjunto Inicial de Dados Disponíveis da Série Temporal ($W = 44$ primeiras observações).	131
Figura 29 – Determinação de s_j para Cada Componente Não-Observável Inicial.	132
Figura 30 – Resultados Gráficos de Treinamento e para as Previsões Recorrente e Direta com e sem Uso do Parâmetro s_j	136
Figura 31 – Resultados Gráficos de Treinamento e Previsão para a Série Temporal <i>Número de Passageiros de Companhias Aéreas</i> Considerando os Parâmetros da Primeira Linha da Tabela 13.	137
Figura 32 – Dinâmica da Quantidade de Regras <i>Fuzzy</i> ou <i>Data Clouds</i> devido a Variação do Parâmetro η_0 Conforme Tabela 17.	140
Figura 33 – Resultados Gráficos de Treinamento e Previsão para a Série Temporal <i>Mancha Solar</i> Considerando os Parâmetros da Linha 2 da Tabela 13.	141
Figura 34 – Resultados de Previsão para as Séries Temporais Univariáveis Relacionados às Tabelas 23 e 24 Aplicando o Caso 2 do Algoritmo Genético (Primeira Parte).	148
Figura 35 – Resultados de Previsão para as Séries Temporais Univariáveis Relacionados às Tabelas 23 e 24 Aplicando o Caso 2 do Algoritmo Genético (Segunda Parte).	149
Figura 36 – Componentes Não-Observáveis da Série Temporal N2090 nas Etapas de Pré-Processamento, Treinamento Incremental e Previsão Aplicando o Caso 2 do Algoritmo Genético.	150
Figura 37 – Análise dos Erros de Treinamento e Previsão para a Série N2090 Aplicando o Caso 2 do Algoritmo Genético.	151
Figura 38 – Informações do Melhor Cromossomo e da População Final do Algoritmo Genético no Caso 2 para a Série Temporal N2090.	153
Figura 39 – Intervalo de Previsão do PRSSA+ENFN para os Números Diários de Casos e Mortes causadas por Covid-19 no Estado do Maranhão, Brasil. Período 1: 25/02/2020 a 21/09/2020.	163
Figura 40 – Intervalo de Previsão do PRSSA+ENFN para os Números Diários de Casos e Mortes causadas por Covid-19 no Estado do Maranhão, Brasil. Período 2: 25/02/2020 a 31/10/2020.	164

Figura 41 – Intervalo de Previsão do PRSSA+ENFN para os Números Diários de Casos e Mortes causadas por Covid-19 no Estado do Maranhão, Brasil. Período 3: 25/02/2020 a 01/12/2020.	165
Figura 42 – Intervalo de Previsão do PRSSA+ENFN para os Números Diários de Casos e Mortes causadas por Covid-19 no Estado do Maranhão, Brasil. Período 4: 25/02/2020 a 21/04/2021.	166
Figura 43 – Histogramas que Representam as Realizações das Variáveis Aleatórias do Números de Casos e Mortes por Covid-19 no Maranhão Considerando o Primeiro Dia de Previsão ($t = T + 1$) para o Período 3.	167
Figura 44 – Decomposição das Quatro Séries Temporais usando o MRSSA para $n = 3$ e $W = 20$	192
Figura 45 – Comparação entre as Séries Temporais e Soma das Componentes Não-Observáveis da Figura 44.	193
Figura 46 – Resultados de Treinamento e Previsão Multivariável do PRSSA+ENFN com $H = 12$ Considerando o Melhor Cromossomo Selecionado pelo Algoritmo Genético. Parâmetros Críticos: $W = 38$; $n = [2\ 3\ 3\ 3]$, $\phi = 0.14$, $\eta_0 = 0.1$	195
Figura 47 – Resultados de Treinamento e Previsão Multivariável do PRSSA+ENFN com $H = 24$ Considerando o Melhor Cromossomo Selecionado pelo Algoritmo Genético. Parâmetros Críticos: $W = 27$; $n = [3\ 4\ 4\ 3]$, $\phi = 0.13$, $\eta_0 = 0.29$	196
Figura 48 – Resultados de Treinamento e Previsão Multivariável do PRSSA+ENFN com $H = 36$ Considerando o Melhor Cromossomo Selecionado pelo Algoritmo Genético. Parâmetros Críticos: $W = 27$, $n = [3\ 4\ 4\ 3]$, $\phi = 0.32$ e $\eta_0 = 0.24$	197
Figura 49 – Resultados de Treinamento e Previsão Multivariável do PRSSA+ENFN com $H = 12$ Considerando o Melhor Cromossomo Selecionado pelo Algoritmo Genético. Parâmetros Críticos: $W = 39$, $n = [2\ 3\ 3\ 3]$, $\phi = 0.5$ e $\eta_0 = 0.25$	198
Figura 50 – Resultados de Treinamento e Previsão Multivariável do PRSSA+ENFN com $H = 24$ Considerando o Melhor Cromossomo Selecionado pelo Algoritmo Genético. Parâmetros Críticos: $W = 32$, $n = [6\ 3\ 2\ 5]$, $\phi = 0.1$ e $\eta_0 = 0.04$	199
Figura 51 – Resultados de Treinamento e Previsão Multivariável do PRSSA+ENFN com $H = 36$ Considerando o Melhor Cromossomo Selecionado pelo Algoritmo Genético. Parâmetros Críticos: $W = 29$, $n = [5\ 2\ 2\ 2]$, $\phi = 0.43$ e $\eta_0 = 0.03$	200
Figura 52 – Componentes Não-Observáveis da Série Temporal TS1 nas Etapas de Pré-Processamento, Treinamento e Previsão Aplicando o Caso 2 do Algoritmo Genético (Parte 1).	202
Figura 53 – Componentes Não-Observáveis da Série Temporal TS1 nas Etapas de Pré-Processamento, Treinamento e Previsão Aplicando o Caso 2 do Algoritmo Genético (Parte 2).	203
Figura 54 – Análise dos Erros de Treinamento e Previsão para a Série TS1 Aplicando o Caso 2 do Algoritmo Genético.	204

Figura 55 – Informações do Melhor Cromossomo e da População Final do Algoritmo Genético no Caso 2 para a Série Temporal TS1.	205
Figura 56 – Componentes Não-Observáveis da Série Temporal TS2 nas Etapas de Pré-Processamento, Treinamento e Previsão Aplicando o Caso 2 do Algoritmo Genético (Parte 1).	206
Figura 57 – Componentes Não-Observáveis da Série Temporal TS2 nas Etapas de Pré-Processamento, Treinamento e Previsão Aplicando o Caso 2 do Algoritmo Genético (Parte 2).	207
Figura 58 – Análise dos Erros de Treinamento e Previsão para a Série TS2 Aplicando o Caso 2 do Algoritmo Genético.	208
Figura 59 – Informações do Melhor Cromossomo e da População Final do Algoritmo Genético no Caso 2 para a Série Temporal TS2.	209
Figura 60 – Componentes Não-Observáveis da Série Temporal TS2 nas Etapas de Pré-Processamento, Treinamento e Previsão Aplicando o Caso 2 do Algoritmo Genético.	209
Figura 61 – Análise dos Erros de Treinamento e Previsão para a Série TS3 Aplicando o Caso 2 do Algoritmo Genético.	210
Figura 62 – Informações do Melhor Cromossomo e da População Final do Algoritmo Genético no Caso 2 para a Série Temporal TS3.	210
Figura 63 – Componentes Não-Observáveis da Série Temporal TS4 nas Etapas de Pré-Processamento, Treinamento e Previsão Aplicando o Caso 2 do Algoritmo Genético. (Parte 1).	211
Figura 64 – Componentes Não-Observáveis da Série Temporal TS4 nas Etapas de Pré-Processamento, Treinamento e Previsão Aplicando o Caso 2 do Algoritmo Genético. (Parte 2).	212
Figura 65 – Análise dos Erros de Treinamento e Previsão para a Série TS4 Aplicando o Caso 2 do Algoritmo Genético.	213
Figura 66 – Informações do Melhor Cromossomo e da População Final do Algoritmo Genético no Caso 2 para a Série Temporal TS4.	213
Figura 67 – Componentes Não-Observáveis da Série Temporal N912 nas Etapas de Pré-Processamento, Treinamento e Previsão Aplicando o Caso 2 do Algoritmo Genético. (Parte 1).	214
Figura 68 – Análise dos Erros de Treinamento e Previsão para a Série N912 Aplicando o Caso 2 do Algoritmo Genético.	215
Figura 69 – Informações do Melhor Cromossomo e da População Final do Algoritmo Genético no Caso 2 para a Série Temporal N912.	215
Figura 70 – Componentes Não-Observáveis da Série Temporal N2076 nas Etapas de Pré-Processamento, Treinamento e Previsão Aplicando o Caso 2 do Algoritmo Genético.	216

Figura 71 – Análise dos Erros de Treinamento e Previsão para a Série N2076 Aplicando o Caso 2 do Algoritmo Genético.	217
Figura 72 – Informações do Melhor Cromossomo e da População Final do Algoritmo Genético no Caso 2 para a Série Temporal N2076.	217
Figura 73 – Componentes Não-Observáveis da Série Temporal N897 nas Etapas de Pré-Processamento, Treinamento e Previsão Aplicando o Caso 2 do Algoritmo Genético (Parte 1).	218
Figura 74 – Componentes Não-Observáveis da Série Temporal N897 nas Etapas de Pré-Processamento, Treinamento e Previsão Aplicando o Caso 2 do Algoritmo Genético (Parte 2).	219
Figura 75 – Análise dos Erros de Treinamento e Previsão para a Série N897 Aplicando o Caso 2 do Algoritmo Genético.	220
Figura 76 – Informações do Melhor Cromossomo e da População Final do Algoritmo Genético no Caso 2 para a Série Temporal N897.	221

LISTA DE TABELAS

Tabela 1	– Série Temporal de Vendas de um Produto por Mês.	42
Tabela 2	– Principais Casos Especiais do Modelo $ARIMA(p, d, q)$	55
Tabela 3	– Lista de Parâmetros usados pelo Método PRSSA+ENFN.	103
Tabela 4	– Resultados do Treinamento Incremental do PRSSA+ENFN com a Abordagem Recorrente e Direta para a Série Mackey-Glass usando o Critério RMSE.	125
Tabela 5	– Resultados de Previsão do PRSSA+ENFN com a Abordagem Recorrente e Direta para a Série Mackey-Glass usando o Critério RMSE.	126
Tabela 6	– Melhores Resultados de Previsão do PRSSA+ENFN com a Abordagem Recorrente e Direta para a Série Mackey-Glass usando o Critério RMSE e Aplicando o Algoritmo Genético para Seleção de Parâmetros Críticos.	128
Tabela 7	– Parâmetros Críticos Selecionados pelo Algoritmo Genético Relacionados aos Resultados de Previsão da Tabela 6.	128
Tabela 8	– Resultados de Previsão Recorrente e Direta de acordo com o Critério RMSE para Previsão Multivariável Considerando Diferentes Horizontes de Previsão.	130
Tabela 9	– Valores de s_j Devido à Variação de W	133
Tabela 10	– Valores de s_j Devido à Variação de n	134
Tabela 11	– Parâmetros Utilizados para cada Previsão na Análise da Contribuição de s_j	134
Tabela 12	– Critérios de Erros de Treinamento e para cada Resultado de Previsão na Análise da Contribuição de s_j	136
Tabela 13	– Parâmetros do PRSSA+ENFN Usados como Referência Considerando os Casos da Previsão Recorrente (Rec.) e Direta (Dir.).	138
Tabela 14	– Resultados do PRSSA+ENFN Devido ao Ajuste do Parâmetro W Considerando o Caso da Previsão Recorrente.	138
Tabela 15	– Resultados do PRSSA+ENFN Devido ao Ajuste do Parâmetro n Considerando o Caso da Previsão Recorrente.	139
Tabela 16	– Resultados do PRSSA+ENFN Devido ao Ajuste do Parâmetro ϕ Considerando o Caso da Previsão Recorrente.	139
Tabela 17	– Resultados do PRSSA+ENFN Devido ao Ajuste do Parâmetro η_0 Considerando o Caso da Previsão Recorrente.	140
Tabela 18	– Resultados do PRSSA+ENFN Devido ao Ajuste do Parâmetro W Considerando o Caso da Previsão Direta.	142
Tabela 19	– Resultados do PRSSA+ENFN Devido ao Ajuste do Parâmetro n Considerando o Caso da Previsão Direta.	142
Tabela 20	– Resultados do PRSSA+ENFN Devido ao Ajuste do Parâmetro ϕ Considerando o Caso da Previsão Direta.	142

Tabela 21 – Resultados do PRSSA+ENFN Devido ao Ajuste do Parâmetro η_0 Considerando o Caso da Previsão Direta.	143
Tabela 22 – Valores para os Parâmetros do Problema de Previsão para cada Série Temporal Univariável do Experimento 1.	145
Tabela 23 – Valores para os Parâmetros Críticos do PRSSA+ENFN para cada Série Temporal Univariável do Experimento e para cada Caso da Execução do Algoritmo Genético.	146
Tabela 24 – Resultados Comparativos entre o PRSSA+ENFN (Casos 1 e 2 do Algoritmo Genético), a ENFN (sem uso de componentes) e os Métodos Apresentados em Nguyen e Novák (2019) de acordo com o critério RMSE (resultados do PRSSA+ENFN que superaram as referências em Negrito).	147
Tabela 25 – Descrição das Séries Temporais para Estudo e os Parâmetros do Problema de Previsão.	155
Tabela 26 – Resultados de Previsão do Experimento 2 de acordo o Critério MSE Médio a partir de 30 Execuções de cada Método (menores erros em negrito) - Parte 1.	156
Tabela 27 – Resultados de Previsão do Experimento 2 de acordo o Critério MSE Médio a partir de 30 Execuções de cada Método (menores erros em negrito) - Parte 2.	157
Tabela 28 – Comparação do RMSE Médio e do MAE Médio do Passo $p = 1$ ao Passo $p = 6$ considerando o Método PRSSA+ENFN e Outras Abordagens em Du et al. (2020) (melhores resultados em negrito).	159
Tabela 29 – Comparação do PRSSA+ENFN com Outros Métodos para o Conjunto de Dados Pequim PM2.5 Considerando o Critério RMSE e Cada Passo $p = 1, \dots, 6$. (melhores resultados em negrito).	160
Tabela 30 – Comparação do PRSSA+ENFN com Outros Métodos para o Conjunto de Dados Consumo de Energia Considerando o Critério RMSE e Cada Passo p . (melhores resultados em negrito).	161
Tabela 31 – Detalhamento dos Períodos de Datas para o Experimento 4 e os Parâmetros T , N e H para cada caso.	163
Tabela 32 – Parâmetros Críticos do PRSSA+ENFN para o Conjunto de Dados Pequim PM2.5.	222
Tabela 33 – Parâmetros Críticos do PRSSA+ENFN para o Conjunto de Dados Consumo de Energia Elétrica.	222
Tabela 34 – Parâmetros Críticos do PRSSA+ENFN para o Conjunto de Dados Qualidade do Ar.	222

LISTA DE ABREVIATURAS E SIGLAS

ALMMo	<i>Autonomous Learning of Multi-Model Systems.</i>
ANFIS	<i>Adaptive Neuro-Fuzzy Inference System.</i>
AR	<i>Autoregressive Models.</i>
ARIMA	<i>Autoregressive Integrated Moving Average.</i>
ARMA	<i>Autoregressive Moving Average.</i>
CNN	<i>Convolutional Neural Network.</i>
DLSTM	<i>Deep Long-Short Term Memory.</i>
DWT	<i>Discrete Wavelet Transform.</i>
ELM	<i>Extreme Learning Machine.</i>
EMD	<i>Empirical Mode Decomposition.</i>
ENFN	<i>Evolving Neuro-Fuzzy Network.</i>
eTS+	<i>evolving Takagi-Sugeno Plus.</i>
EWRLS	<i>Extended Weighted Recursive Least Squares.</i>
FWRLS	<i>Fuzzy Weighted Recursive Least Squares.</i>
GA	<i>Genetic Algorithm.</i>
G-FWRLS	<i>Global Fuzzy Weighted Recursive Least Squares.</i>
GRU	<i>Gated Recurrent Unit.</i>
HW	<i>Holt-Winters.</i>
L-FWRLS	<i>Local Fuzzy Weighted Recursive Least Squares.</i>
LEOA	<i>Local Error Optimization Approach .</i>
LSTM	<i>Long-Short Term Memory.</i>
MA	<i>Moving Average.</i>
MAE	<i>Mean Absolute Error.</i>
MAPE	<i>Mean Absolute Percentage Error.</i>

MASE	<i>Mean Absolute Scaled Error.</i>
MIMO	<i>Multi-Input and Multi-Output.</i>
MISO	<i>Multi-Input and Single-Output.</i>
MLP	<i>Multi-Layer Perceptron.</i>
MRSSA	<i>Multivariate Recursive Singular Spectrum Analysis.</i>
MSE	<i>Mean Squared Error.</i>
MSSA	<i>Multivariate Singular Spectrum Analysis.</i>
PRSSA	<i>Parallel Recursive Singular Spectrum Analysis.</i>
PRSSA+ENFN	<i>Parallel Recursive Singular Spectrum Analysis and Evolving Neuro-Fuzzy Network.</i>
PSO	<i>Particle Swarm Optimization.</i>
RMSE	<i>Root Mean Squared Error.</i>
RNN	<i>Recurrent Neural Networks.</i>
RSSA	<i>Recursive Singular Spectrum Analysis.</i>
SARIMA	<i>Seasonal Autoregressive Integrated Moving Average.</i>
SEQ2SEQ	<i>Sequence-to-Sequence Deep Learning Model.</i>
SEQ2SEQ-BI	<i>Sequence-to-Sequence Deep Learning Model and Bi-LSTM.</i>
SSA	<i>Singular Spectrum Analysis.</i>
STL	<i>Seasonal Trend decomposition based on Loess.</i>
SVD	<i>Singular Value Decomposition.</i>
SVM	<i>Support Vector Machine.</i>
SVR	<i>Support Vector Regression.</i>
TS	<i>Takagi-Sugeno.</i>
VAR	<i>Vector Autoregressive Models.</i>
VARIMA	<i>Vector Autoregressive Integrated Moving Average.</i>

SUMÁRIO

I	CONTEXTUALIZAÇÃO INICIAL	23
1	INTRODUÇÃO	24
1.1	Motivação e Justificativa	25
1.2	Objetivos da Pesquisa	26
1.2.1	Objetivo Geral	26
1.2.2	Objetivos Específicos	26
1.3	Trabalhos Anteriores e Principais Contribuições da Tese	26
1.4	Artigos Publicados	28
1.4.1	Artigos em Congressos	28
1.4.2	Artigos em Periódicos	28
1.5	Organização da Tese	28
II	FUNDAMENTAÇÃO TEÓRICA	30
2	SÉRIES TEMPORAIS	31
2.1	Fundamentos sobre Séries Temporais	31
2.2	Classificação de Séries Temporais	33
2.2.1	Séries Contínuas ou Discretas	33
2.2.2	Séries Estacionárias ou Não-Estacionárias	33
2.2.3	Séries Temporais Sazonais	36
2.2.4	Séries Univariáveis e Multivariáveis	37
2.3	Aspectos Gerais Sobre a Previsão de Séries Temporais	38
2.3.1	Previsão Qualitativa e Quantitativa: Disponibilidade das Observações das Séries Temporais	38
2.3.1.1	<i>Previsão Qualitativa</i>	38
2.3.1.2	<i>Previsão Quantitativa</i>	39
2.3.2	Boas Práticas para a Previsão de Séries Temporais	41
2.3.3	Estratégias para Previsão de Múltiplos Passos a Frente	41
2.3.3.1	<i>Previsão Recorrente</i>	42
2.3.3.2	<i>Previsão Direta</i>	43
2.3.4	Avaliação dos Resultados de Previsão de Séries Temporais	44
2.3.4.1	<i>Erro Absoluto Médio</i>	45
2.3.4.2	<i>Erro Quadrático Médio</i>	45
2.3.4.3	<i>Raiz do Erro Quadrático Médio</i>	46
2.3.4.4	<i>Erro Absoluto Médio Percentual</i>	46

2.3.4.5	<i>Erro Absoluto Médio em Escala</i>	46
2.3.5	Intervalos de Previsão	47
2.4	Métodos para Previsão de Séries Temporais	48
2.4.1	Métodos Simples para Previsão de Séries Temporais	49
2.4.2	Suavização Exponencial	50
2.4.3	Modelos ARIMA	53
2.5	Estado da Arte na Previsão de Séries Temporais	56
2.6	Comentários Finais do Capítulo	59
3	ANÁLISE ESPECTRAL SINGULAR	60
3.1	Descrição Geral do Método de Análise Espectral Singular - SSA	60
3.2	Estágio de Decomposição	61
3.2.1	Etapa de Incorporação	61
3.2.2	Etapa de Decomposição em Valores Singulares - SVD	62
3.3	Estágio de Reconstrução	63
3.3.1	Agrupamento de Índices	63
3.3.2	Etapa da Média Diagonal	63
3.4	Métodos para Agrupamento de Índices das Matrizes Elementares	64
3.5	Separabilidade	65
3.6	Seleção do Parâmetro L	66
3.7	Estudo Gráfico da Análise Espectral Singular	66
3.8	Estado da Arte da Análise Espectral Singular	70
3.9	Comentários Finais do Capítulo	72
4	MODELAGEM EVOLUTIVA	73
4.1	Aspectos Gerais sobre Modelagem Matemática	73
4.2	Modelos Evolutivos	74
4.3	Estrutura Básica dos Modelos <i>Fuzzy</i> Evolutivos	77
4.3.1	Partição Evolutiva do Espaço de Dados	83
4.3.1.1	<i>Geração de Regras Fuzzy ou Clusters</i>	84
4.3.1.2	<i>Fusão de Regras Fuzzy ou Clusters</i>	86
4.3.1.3	<i>Remoção de Regras Fuzzy ou Clusters</i>	87
4.3.2	Padronização e Normalização dos Dados	88
4.4	Modelo <i>Autonomous Learning Multi-Model</i> (ALMMo)	89
4.5	Estado da Arte dos Modelos Evolutivos	91
4.6	Comentários Finais do Capítulo	93
III	CONTEXTUALIZAÇÃO METODOLÓGICA	94
5	METODOLOGIA PARA PREVISÃO DE SÉRIES TEMPORAIS	95

5.1	Descrição Geral da Metodologia Proposta: PRSSA+ENFN	95
5.2	Etapa de Pré-processamento	97
5.2.1	Determinação dos Parâmetros do Método PRSSA+ENFN	97
5.2.2	Extração das Componentes Não-Observáveis Iniciais	98
5.2.3	Inicialização da Rede Neuro-Fuzzy Evolutiva (ENFN)	102
5.3	Etapa de Treinamento Incremental	103
5.3.1	Extração Recursiva das Componentes Não-Observáveis usando o PRSSA . .	103
5.3.2	Mapeamento das Entradas e Saídas da ENFN	107
5.3.3	Aprendizagem Evolutiva da ENFN Baseada no Método ALMMo-1	108
5.4	Etapa de Previsão	114
5.4.1	Previsão Recorrente do PRSSA+ENFN	115
5.4.2	Previsão Direta pela PRSSA+ENFN	116
5.5	Estratégias para Seleção dos Parâmetros Críticos do PRSSA+ENFN . .	117
5.5.1	Aplicação de Algoritmo Genético	117
5.5.2	Seleção Aleatória dos Parâmetros Críticos da PRSSA+ENFN	118
5.6	Análise da Complexidade Computacional do PRSSA+ENFN	118
5.6.1	Complexidade Computacional do Pré-processamento	119
5.6.2	Complexidade Computacional do Treinamento Incremental	119
5.6.3	Complexidade Computacional da Previsão	120
5.6.4	Complexidade Computacional Geral do PRSSA+ENFN e da Seleção dos Parâmetros Críticos	120
5.7	Comentários Finais do Capítulo	121
6	RESULTADOS EXPERIMENTAIS	123
6.1	Descrição Geral dos Experimentos	123
6.2	Avaliação de Características do PRSSA+ENFN	123
6.2.1	Avaliação das Previsões do Tipo Recorrente e Direta do PRSSA+ENFN . .	123
6.2.2	Determinação do Parâmetro s_j para cada Componente Não-Observável . . .	131
6.2.3	Análise da Influência nos Resultados do PRSSA+ENFN Devido ao Ajuste dos Parâmetros Críticos	137
6.3	Experimento 1: Previsão Recorrente de Séries Temporais Univariáveis .	143
6.4	Experimento 2: Previsão Direta Univariável de Séries Temporais	153
6.5	Experimento 3: Previsão Direta de Séries Temporais Multivariáveis . . .	157
6.6	Experimento 4: Previsão Multivariável de Séries Temporais dos Núme- ros de Casos e Mortes Causadas pela Covid-19 no Estado do Maranhão	161
6.7	Comentários Finais do Capítulo	167

IV	CONTEXTUALIZAÇÃO FINAL	168
7	CONSIDERAÇÕES FINAIS	169
7.1	Conclusões	169
7.2	Trabalhos Futuros	170
	REFERÊNCIAS	171
	APÊNDICE A – ALGORITMOS UTILIZADOS NA TESE	183
A.1	Algoritmo da Análise Espectral Singular (SSA)	183
A.2	Algoritmo da Etapa de Pré-processamento do PRSSA+ENFN	184
A.3	Algoritmo da Etapa de Treinamento Incremental do PRSSA+ENFN	185
A.4	Algoritmo da Etapa de Previsão do PRSSA+ENFN	186
A.5	Algoritmo para Seleção de Parâmetros	187
	APÊNDICE B – MÉTODO MRSSA PARA EXTRAÇÃO DE COMPO- NENTES NÃO-OBSERVÁVEIS	188
B.1	Comentários Iniciais sobre o MRSSA	188
B.2	Condições Iniciais do MRSSA	188
B.3	Atualização Recursiva da Matriz Covariância pelo MRSSA	189
B.4	Atualização dos Autovalores e Matriz de Valores Singulares pelo MRSSA	190
B.5	Determinação das Componentes Não-Observáveis pelo MR-SSA	191
B.6	Resultados de Decomposição do Método MRSSA	191
	APÊNDICE C – RESULTADOS ADICIONAIS DOS EXPERIMENTOS	195
C.1	Resultados Gráficos da Previsão Multivariável Recorrente (Seção 6.2)	195
C.1.1	Resultados Gráficos para Previsão Recorrente Quando $H = 12$	195
C.1.2	Resultados Gráficos para Previsão Recorrente Quando $H = 24$	196
C.1.3	Resultados Gráficos para Previsão Recorrente Quando $H = 36$	197
C.1.4	Resultados Gráficos para Previsão Direta Quando $H = 12$	198
C.1.5	Resultados Gráficos para Previsão Direta Quando $H = 24$	199
C.1.6	Resultados Gráficos para Previsão Direta Quando $H = 36$	200
C.2	Resultados Gráficos da Previsão Univariável Recorrente do Experimento 1 (Seção 6.3)	202
C.2.1	Série Temporal TS1	202
C.2.2	Série Temporal TS2	206
C.2.3	Série Temporal TS3	209
C.2.4	Série Temporal TS4	211
C.2.5	Série Temporal N912	214
C.2.6	Série Temporal N2076	216

C.2.7	Série Temporal N897	218
C.3	Tabelas com os Valores para os Parâmetros Críticos do Experimento 3	
	(Seção 6.5)	222
C.3.1	Tabelas para o Conjunto de Dados Pequim PM2.5	222
C.3.2	Tabelas para o Conjunto de Dados Consumo de Energia Elétrica	222
C.3.3	Tabelas para o Conjunto de Dados Qualidade do Ar	222

Parte I

CONTEXTUALIZAÇÃO INICIAL

1 INTRODUÇÃO

A previsão de séries temporais é um tema importante em vários campos de estudo, como física, economia, meteorologia, medicina e engenharia. Os resultados dos métodos de previsão de séries temporais auxiliam no processo de tomada de decisão e no planejamento de ações preventivas em diversas atividades. De forma geral, uma série temporal pode ser definida como qualquer conjunto de dados que estejam ordenados no tempo (HYNDMAN; ATHANASOPOULOS, 2014), logo, os métodos de previsão buscam identificar padrões em dados históricos de séries temporais de tal forma que um modelo seja desenvolvido para construir seus padrões temporais futuros (BROCKWELL; DAVIS, 2002).

Além da previsão, os especialistas também podem estar interessados na caracterização de uma série temporal, ou seja, identificar suas componentes não-observáveis, que são padrões ocultos contidos nos dados. Usualmente, essas componentes são mais simples que a série original e ajudam a entender o comportamento geral da série (ABDOLLAHZADE et al., 2015). A quantidade e o tipo de componentes não-observáveis dependem do método usado para extraí-las dos dados de séries temporais.

Atualmente, sabe-se que há um grande volume de dados disponível em vários formatos, sendo que esses dados são gerados em alta velocidade (GE; ZENGE, 2020), e as séries temporais também estão inseridas nesse contexto. Assim, os especialistas precisam lidar com uma quantidade crescente de informações de maneira eficiente, extraindo conhecimento útil para modelar sistemas complexos. Além disso, dados reais são geralmente complexos, não-lineares e não-estacionários (GALICIA et al., 2019). Portanto, os pesquisadores são desafiados a desenvolver algoritmos que consideram grandes quantidades de dados, atendam requisitos de tempo real de acordo com cada aplicação, se adaptem às mudanças de dinâmica nos dados, sejam computacionalmente eficientes e, ao mesmo tempo, preservem a transparência do modelo (ANGELOV, 2013).

Tendo em vista esse cenário, o paradigma dos sistemas evolutivos se apresenta como uma alternativa para lidar com esses desafios (ANGELOV; ZHOU, 2006). Neste caso, os modelos desenvolvidos podem modificar sua estrutura, ou seja, são capazes de se adaptar de acordo com as mudanças no comportamento dos dados, alterando sua arquitetura interna e ajustando seus parâmetros (LUGHOFER, 2011). As técnicas relacionadas aos sistemas evolutivos são constituídas por algoritmos que trabalham de forma recursiva, sendo capazes de extrair conhecimento de amostras dos dados a cada instante. Geralmente, os modelos evolutivos são aplicados sobre uma estrutura *fuzzy* ou *neuro-fuzzy*, aproveitando suas características de aproximação universal, de trabalhar com incertezas e apresentar interpretabilidade e transparência por meio de suas regras *fuzzy* (WANG, 1997). Assim sendo, a metodologia desenvolvida nessa tese aplica os conceitos de sistemas evolutivos por meio de uma estrutura *neuro-fuzzy*.

Ademais, diversas metodologias de previsão de séries temporais baseadas em inteligência computacional e aprendizagem de máquina vêm sendo desenvolvidas pela comunidade científica (GAN et al., 2014). Entretanto, Makridakis, Spiliotis e Assimakopoulos (2018) afirmam que muitas dessas propostas são limitadas e que aplicar técnicas avançadas e inteligentes não necessariamente garante sua superioridade. Eles listam algumas dessas limitações, como por exemplo: avaliar a abordagem para poucas ou até mesmo uma única série temporal; realizar a previsão somente a curto prazo, sem considerar experimentos a médio e longo prazo; e não comparar com outras técnicas para previsão de séries temporais. Makridakis, Spiliotis e Assimakopoulos (2018) destacam ainda que muitas dessas propostas até apresentam menores erros em alguns casos, porém muitas a preço de um elevado custo computacional. Uma das sugestões desses autores é o desenvolvimento de métodos híbridos, que aproveitem a vantagem de abordagens tradicionais de previsão de séries temporais e de novas técnicas baseadas em inteligência computacional e aprendizagem de máquina. De acordo Hajirahimi e Khashei (2019), a aplicação de modelos híbridos para modelagem e previsão de séries temporais continua a crescer nos estudos recentes, com a produção e publicação de várias pesquisas.

1.1 Motivação e Justificativa

Muitas das séries temporais que estão presentes em problemas reais são bastante complexas, sendo possível apresentar alta dimensionalidade em seus dados e comportamentos como não-linearidade e não-estacionariedade. Dessa forma, distintas abordagens são propostas para lidar com a tarefa de previsão de séries temporais univariáveis ou multivariáveis. Outro desafio é lidar com a velocidade dos fluxos de dados, que chegam a todo momento, sendo interessante que os métodos sejam capazes de se atualizar e melhorar sua estrutura a cada instante. Uma das estratégias ao desenvolver técnicas para superar essas adversidades e buscando mais eficiência computacional é utilizando abordagens híbridas compostas por métodos tradicionais e de inteligência computacional, aproveitando as vantagens de ambas as classes.

Logo, a pesquisa desenvolvida nessa tese visa desenvolver uma metodologia híbrida e inteligente, unindo as teorias de previsão de séries, de componentes não-observáveis, modelos *fuzzy* (neuro-*fuzzy*) e modelos evolutivos, com a finalidade de apresentar resultados competitivos de previsão em relação a outras alternativas na literatura. A abordagem proposta nesse trabalho segue o princípio do "dividir para conquistar" (ANGELOV, 2013), dividindo um problema complexo em vários subproblemas mais simples. Com essa ideia, essa metodologia decompõe a série temporal complexa em várias componentes não-observáveis que são mais simples de lidar. Além disso, essa metodologia deve ser capaz de atuar em um cenário em que os dados cheguem a todo instante, atualizando a sua estrutura para se ajustar à dinâmica desses dados e não sendo necessário considerar todo o histórico de observações da série para realizar a aprendizagem. Outra importante motivação é a proposta de uma abordagem que trabalhe com dois objetivos, que são a previsão e a caracterização de séries temporais, disponibilizando um conjunto de resultados

para apoiar o processo de tomada de decisão pelos especialistas.

1.2 Objetivos da Pesquisa

1.2.1 Objetivo Geral

Desenvolver uma metodologia inteligente para previsão de séries temporais univariáveis ou multivariáveis baseada em suas componentes não-observáveis utilizando uma rede neuro-*fuzzy* evolutiva.

1.2.2 Objetivos Específicos

- Desenvolver uma metodologia evolutiva, que se adapte à dinâmica dos dados que chegam a todo instante;
- Definir os procedimentos de pré-processamento, treinamento incremental e previsão;
- Implementar uma versão recursiva e paralela do método SSA (*Singular Spectrum Analysis*), denominada de PRSSA (*Parallel Recursive Singular Spectrum Analysis*), tal que identifique as componentes não-observáveis;
- Propor estratégias para seleção dos parâmetros da metodologia desenvolvida;
- Analisar a complexidade computacional da metodologia proposta;
- Realizar experimentos para comparar a metodologia proposta com outras técnicas no mesmo contexto e com resultados de outros trabalhos científicos;
- Aplicar a metodologia proposta para previsão de dados relacionados à Covid-19 no Maranhão - Brasil.

1.3 Trabalhos Anteriores e Principais Contribuições da Tese

O trabalho apresentado nessa tese evoluiu a partir dos resultados da dissertação de mestrado dos mesmos autores, intitulada "Metodologia Evolutiva para Previsão Inteligente de Séries Temporais Sazonais Baseada em Espaço de Estados Não-Observáveis"(RODRIGUES JÚNIOR, 2017). O objetivo da dissertação foi desenvolver uma metodologia também evolutiva e baseada em uma estrutura neuro-*fuzzy Takagi Sugeno*, aplicada para previsão recorrente múltiplos passos a frente de séries temporais exclusivamente univariáveis e sazonais. Por outro lado, essa tese de doutorado aprimorou e expandiu a metodologia para fazer a previsão de séries temporais sem sazonalidades definidas, tendo a flexibilidade de trabalhar no contexto univariável ou multivariável. Além disso, a proposta dessa tese é capaz de não somente fazer previsões recorrentes, como também realizar a previsão do tipo direta, tópicos que serão descritos na fundamentação teórica.

Apesar de ambos trabalhos utilizarem redes neuro-*fuzzy*, elas são baseadas em algoritmos evolutivos distintos. Na dissertação de mestrado, a estrutura evolutiva da proposta baseou-se nos métodos *evolving Takagi-Sugeno Plus* (eTS+) e *eClustering+*. Por sua vez, a metodologia dessa tese tem como alicerce o método ALMMo (*Autonomous Learning of Multi-Models*), que será apresentado nesse texto. Com essa modificação, ampliou-se o nível de autonomia da abordagem neuro-*fuzzy*, isto é, são necessários menos parâmetros para inicialização desse novo método, sendo que esses parâmetros são distintos em relação à dissertação. Ademais, o ALMMo requer menos passos para realizar o procedimento evolutivo da proposta em comparação aos métodos anteriores.

As componentes não-observáveis na dissertação eram identificadas pelo método *Recursive SSA* (RSSA), uma versão alternativa proposta. Contudo, o RSSA descrito na dissertação tinha um custo computacional elevado a cada instante, pois necessitava executar o procedimento da Decomposição em Valores Singulares (SVD, *Singular Value Decomposition*) a fim de obter os valores singulares e, por conseguinte, os autovalores e autovetores. Nessa tese, essa característica foi removida, sendo que os autovalores e autovetores foram atualizados recursivamente por meio do conceito de matrizes de perturbação, reduzindo o custo computacional. Outra modificação ocorre quando se trabalha com séries temporais multivariáveis, em que esse algoritmo para extração de componentes é representado de forma paralela para cada série.

As proposições consequentes das regras *fuzzy* na dissertação são modelos lineares no espaço de estados, cujos estados são formados pelas próprias componentes extraídas. Porém, no caso de séries multivariáveis nessa tese, as representações nos consequentes das regras são modelos de vetores autorregressivos (VAR) que buscam considerar as relações entre as componentes das diferentes séries temporais, explorando as correlações entre elas para obter os resultados de previsão.

Outra característica exclusiva da tese é que são apresentados critérios para definição dos parâmetros da metodologia desenvolvida. Uma alternativa é a aplicação dos algoritmos genéticos (GA, *Genetic Algorithm*), obtendo uma previsão de acordo com os resultados do procedimento de otimização do GA. A outra possibilidade é fazer uma seleção aleatória de parâmetros, sendo que os resultados de previsão podem ser representados por um intervalo de valores, dando uma indicação das possibilidades do comportamento futuro da série temporal.

Em resumo, as principais contribuições desse trabalho de tese são listadas a seguir:

- Uma nova metodologia inteligente, híbrida e evolutiva para previsão de séries temporais múltiplos passos a frente;
- Proposta do método PRSSA (*Parallel Recursive Singular Spectrum Analysis*);
- Uso de algoritmos genéticos para determinação dos parâmetros críticos da metodologia proposta;

- Aplicação de intervalos de previsão para exibir as incertezas dos resultados futuros;
- Aplicação em problemas de previsão no contexto univariável e multivariável;
- Utilização da abordagem proposta em problemas de previsão recorrente e direta;
- Uma nova alternativa de previsão para lidar com os procedimentos da aprendizagem incremental a cada instante.

1.4 Artigos Publicados

As publicações produzidas durante a pesquisa de doutorado até o momento são apresentadas nessa seção em ordem cronológica e por tipo, conforme a seguir.

1.4.1 Artigos em Congressos

- RODRIGUES JÚNIOR, S. E.; SERRA, G. L. O. *Metodologia Evolutiva em Espaço de Componentes Não-Observáveis para Previsão de Séries Temporais Multivariáveis*. Congresso Brasileiro de Automática (CBA), João Pessoa - PB, 2018.
- RODRIGUES JÚNIOR, S. E.; SERRA, G. L. O. *Evolving Methodology in Unobservable Components Space for Multivariate Time Series Forecasting*. IEEE World Congress on Computational Intelligence (IEEE WCCI), Rio de Janeiro. 2018.
- RODRIGUES JÚNIOR, S. E.; SERRA, G. L. O. *Previsão Inteligente de Séries Temporais Baseada em Redes Neuro-Fuzzy Takagi-Sugeno Evolutivas Distribuídas*. 14^o Simpósio Brasileiro de Automação Inteligente (SBAI, 2019). Ouro Preto - MG. Anais do 14^o Simpósio Brasileiro de Automação Inteligente, 2019.

1.4.2 Artigos em Periódicos

- RODRIGUES JÚNIOR, S. E.; SERRA, G. L. O. *Intelligent Forecasting of Time Series Based on Evolving Distributed Neuro-Fuzzy Network*. Computational Intelligence Journal, 2020.
- RODRIGUES JÚNIOR, S. E.; SERRA, G. L. O. *An Approach for Evolving Neuro-Fuzzy Forecasting of Time Series Based on Parallel Recursive Singular Spectrum Analysis*. Fuzzy Sets and Systems, 2021.

1.5 Organização da Tese

A organização dessa tese será descrita logo a seguir. O Capítulo 1 descreveu o contexto da pesquisa, as motivações, justificativas, objetivos e principais contribuições, assim como alguns trabalhos publicados em congressos e periódicos.

O Capítulo 2 apresenta os conceitos fundamentais sobre séries temporais e suas principais classificações. Em seguida, esse capítulo comenta alguns aspectos gerais e importantes sobre a previsão de séries temporais. Uma subseção que descreve o estado da arte no campo de pesquisa sobre a previsão de séries temporais também é apresentada.

O Capítulo 3 descreve o método de Análise Espectral Singular (SSA, *Singular Spectrum Analysis*), utilizado como base nesse trabalho para extrair as componentes não-observáveis a partir da série temporal. Todas suas etapas fundamentais serão descritas matematicamente, assim como alguns aspectos importantes para a análise do método. Por fim, o estado da arte no uso dessa técnica é apresentado.

Por sua vez, o Capítulo 4 exhibe os principais fundamentos sobre os modelos evolutivos, tendo como foco principal as estruturas dos modelos *fuzzy* e *neuro-fuzzy*. Comenta também sobre os principais procedimentos das abordagens evolutivas, como a geração de regras, a fusão de regras e a remoção de regras. O método ALMMo (*Autonomous Learning Multi-Model*) é descrito de forma geral nesse capítulo, cujos procedimentos evolutivos e de aprendizagem incremental servirão como base para a definição da metodologia proposta. Além disso, o estado da arte sobre modelos evolutivos também é descrito.

No Capítulo 5, a metodologia evolutiva proposta para previsão de séries temporais é detalhada, apresentando suas três etapas: pré-processamento, treinamento incremental e previsão. Apresentam-se também seus fundamentos matemáticos, os processos para determinação dos parâmetros e uma análise assintótica da complexidade computacional do método proposto.

Os resultados experimentais analisados através de quatro experimentos são apresentados e discutidos no Capítulo 6. São feitas comparações com outras técnicas e trabalhos similares para problemas de previsão de séries. Avaliou-se também a aplicação do método proposto para previsão de dados relacionados à Covid-19 no contexto do estado do Maranhão, Brasil. Além desses experimentos, são estudadas algumas das principais características do método proposto.

Finalmente, o Capítulo 7 traz as conclusões e sugestões de trabalhos futuros para o desenvolvimento da tese. Tem-se ainda os Apêndices que exibirão os algoritmos utilizados na pesquisa, alguns resultados gráficos e uma versão multivariável e recursiva do método SSA proposto no trabalho e que será tratada em futuras pesquisas.

Parte II

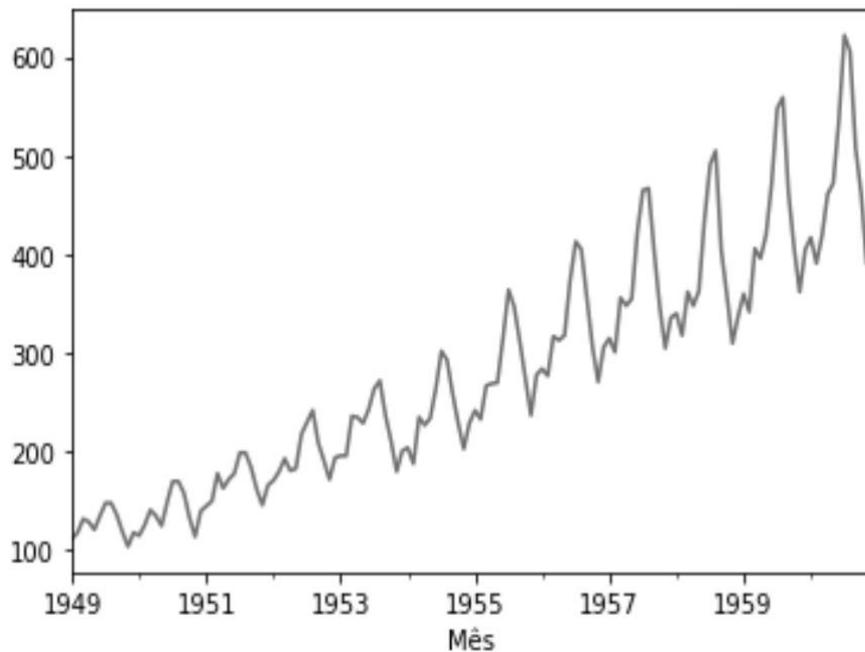
FUNDAMENTAÇÃO TEÓRICA

2 SÉRIES TEMPORAIS

2.1 Fundamentos sobre Séries Temporais

Devido aos avanços tecnológicos e científicos, a quantidade de dados disponíveis em diversas áreas do conhecimento vem aumentando exponencialmente. Além do elevado volume de dados brutos disponíveis, estes são produzidos a uma velocidade crescente, apresentando-se em uma variedade de formatos, por exemplo, dados numéricos, imagens, vídeos, textos, etc. (GALICIA et al., 2018). Um tipo específico de dados que está incluso nesse contexto são as séries temporais. De acordo com Montgomery, Jennings e Kulahci (2015), as séries temporais podem ser definidas como uma sequência cronológica de dados de uma variável de interesse, ou seja, os dados estão sequencialmente organizados de acordo com o tempo, embora esta ordem também pode estar relacionada a outros tipos de dimensões, tal como o espaço (WEI, 2006). Todavia, nesse trabalho, serão consideradas apenas séries temporais ordenadas no tempo. A ordem temporal dos dados é relevante e deve ser considerada em qualquer análise envolvendo séries temporais. Um exemplo de série temporal pode ser visualizado na Figura 1.

Figura 1 – Exemplo de Série Temporal: Quantidade de Passageiros em Milhares em uma Companhia Aérea Internacional.



Fonte: Adaptado de Brockwell e Davis (2002).

Geralmente, as séries temporais se apresentam em formato numérico, entretanto, há

outros formatos de dados que podem ser considerados nesse contexto. Por exemplo, as séries temporais podem ser compostas por um conjunto de imagens organizadas sequencialmente em intervalos regulares de tempo. Essa categoria de série vem sendo muito utilizada em aplicações com imagens de satélites (*Satellite Image Time Series*), como o acompanhamento de uma área de vegetação (GUYET; NICOLAS, 2015; KHALI et al., 2019). Outros exemplos são as variáveis linguísticas ordenadas no tempo que, em geral, aplica-se o conceito de Séries Temporais *Fuzzy* para analisá-las (CHENG; YANG, 2018).

A análise de séries temporais sem dúvidas possui vasta aplicação em muitos campos do conhecimento. Na economia e nos negócios, pode-se encontrá-las em vendas de produtos, no mercado de ações e taxas de câmbio (CAO; LI; LI, 2019), (ALHNAITY; ABBOD, 2020), (NIU et al., 2020). Séries temporais meteorológicas, tais como temperatura, velocidade do vento e precipitações, são muito importantes para realizar previsões climáticas (DAS; GHOSH, 2017), (ADEWOLE et al., 2020). Na medicina, estudos envolvendo séries temporais vêm sendo aplicados na prevenção e previsões da quantidade de atendimentos em hospitais (SHI; WORDEN; CROSS, 2019), em exames de eletrocardiograma e eletroencefalograma (WEI, 2006) (KANANI; PADOLE, 2020) e para previsão da evolução da epidemia relacionada à Covid-19 (PETROPOULOS; MAKRIDAKIS; STYLIANOU, 2020). Ademais, as séries são bastante aplicadas no monitoramento industrial, no controle e na análise de sinais diversos (LI et al., 2021). Podem ser exploradas ainda por técnicas de reconhecimento de padrões, de aprendizagem de máquinas e mineração de dados (*data mining*) (PALIT; POPOVIC, 2005; ABDOLLAHZADE et al., 2015).

É possível utilizar dois tipos de enfoques principais para estudar séries temporais, que são o domínio do tempo ou o domínio da frequência (MORETTIN; TOLOI, 2006). No domínio do tempo, exemplos de técnicas para o estudo de séries temporais são a regressão, métodos de decomposição, suavização exponencial, modelos Autorregressivos Integrativos de Média Móvel (ARIMA), etc. O principal objetivo dessa análise é obter os modelos que representam a relação entre as observações da série temporal, estudar as funções de autocorrelação, analisar tendências e as variações sazonais nos dados. Em alternativa, a análise no domínio da frequência é aplicada geralmente para séries que apresentam padrões periódicos e cíclicos destacados, contando com métodos baseados em transformadas de Fourier e na densidade espectral, o que permite determinar o número de componentes de frequência e detectar a frequência cíclica dominante (JOO; KIM, 2015). É importante frisar que ambos enfoques não são excludentes entre si, mas sim complementares na análise de séries temporais.

Quando pesquisadores exploram os variados métodos de análise de séries temporais, eles podem estar interessados em diferentes objetivos. Dentre os mais comuns destacam-se (BROCKWELL; DAVIS, 2002), (MORETTIN; TOLOI, 2006):

- Modelagem da série temporal, representando o mecanismo gerador da série;

- Análise por inspeção gráfica da série temporal;
- Previsão de observações para a série temporal, podendo ser a curto, a médio ou a longo prazo;
- Caracterização dos padrões da série, consistindo na extração de suas componentes não-observáveis.

Em especial, essa tese preocupa-se principalmente com os dois últimos objetivos acima descritos.

2.2 Classificação de Séries Temporais

As séries temporais podem ser classificadas em muitas categorias, sendo que uma série pode pertencer a uma ou mais delas simultaneamente, conforme apresentam Palit e Popovic (2005). No entanto, são destacadas as seguintes classificações de séries temporais nessa tese:

- Séries contínuas ou discretas;
- Estacionárias ou não-estacionárias;
- Sazonais ou não-sazonais;
- Univariáveis ou multivariáveis;
- Entre outras classificações (PALIT; POPOVIC, 2005).

2.2.1 Séries Contínuas ou Discretas

Em relação ao tempo, séries temporais podem ser classificadas em contínuas ou discretas (BROCKWELL; DAVIS, 2002). As séries contínuas contêm observações que são obtidas continuamente no intervalo de tempo $t \in [0, \infty)$. Por outro lado, séries temporais discretas são denotadas por $\{y_1, y_2, \dots, y_N\}$, isto é, y_k são amostras discretas da série, para $k = 1, \dots, N$ (N é a quantidade total de amostras). Todas as séries temporais abordadas nessa tese são discretas, aplicando a letra k para representação do tempo discreto.

2.2.2 Séries Estacionárias ou Não-Estacionárias

Por sua vez, a estacionariedade é uma das suposições mais consideradas no estudo de séries temporais, pois simplifica alguns procedimentos analíticos. Essa propriedade está relacionada ao valor médio e a variância dos dados, sendo que a série se desenvolve em torno de uma média constante e com covariância independente do tempo (PALIT; POPOVIC, 2005). Segundo Aguirre (2015), a estacionariedade indica que as leis de probabilidade não variam com o tempo, apresentando um equilíbrio estatístico. Sendo assim, há uma forte relação com o

conceito de invariância no tempo. Contudo, a maioria das séries temporais de problemas reais apresentam algum comportamento não-estacionário (MORETTIN; TOLOI, 2006).

A estacionariedade pode ser de dois tipos: estacionariedade no sentido forte ou estrito e a estacionariedade no sentido fraco ou amplo. No primeiro caso, todas as distribuições de probabilidade unidimensionais são invariantes sob a translação do tempo, logo, a média e a variância são constantes (MORETTIN; TOLOI, 2006). Contudo, na prática, nem sempre é fácil provar a estacionariedade no sentido forte, pois necessita-se ter em mãos o modelo de probabilidade completo. Em alternativa, pode-se avaliar a estacionariedade no sentido fraco, averiguando medidas de valores esperados, variâncias, correlações e covariâncias (YATES; GOODMAN, 2017). Segundo Yates e Goodman (2017), a estacionariedade no sentido fraco se o valor esperado for constante com o tempo e a autocorrelação depender apenas da diferença de tempo entre duas variáveis aleatórias.

De acordo com Palit e Popovic (2005), muitas vezes é difícil avaliar a estacionariedade de forma analítica seguindo os critérios de sua definição. Como alternativa, na prática, são comumente consideradas duas abordagens: a primeira usa as propriedades das funções de autocorrelação e de autocovariância da série; enquanto que a segunda baseia-se em uma inspeção visual dos dados.

Brockwell e Davis (2002) trazem a definição da função de autocovariância, que é considerada uma medida do grau de interdependência entre as observações. Logo, a função de autocovariância de y_k com atraso ou *lag* h é denotada por:

$$\gamma_y(h) = Cov(y_k, y_{k-h}). \quad (2.1)$$

Para $h = 0$, obtém-se a variância conforme a seguir:

$$Var[y_k] = \gamma_y(0) = Cov(y_k, y_k). \quad (2.2)$$

Quando a função de autocovariância $\gamma_y(h)$ for nula, y_{k-h} e y_k são totalmente independentes entre si. Por outro lado, em relação a função de autocorrelação, esta traz uma medida que indica o quanto o valor de uma observação é influenciada por outras, sendo que seu valor está normalizado entre -1 e 1 (BROCKWELL; DAVIS, 2002). O valor 1 indica que há uma correlação direta e perfeita entre as observações consideradas, enquanto o valor -1 expressa uma correlação invertida perfeita. Da mesma forma, o valor nulo denota que não existe correlação entre ambas observações da série temporal. A função de autocorrelação de y_k com *lag* h é calculada conforme a seguir (BROCKWELL; DAVIS, 2002):

$$\rho_y(h) = \frac{\gamma_y(h)}{\gamma_y(0)} = Corr(y_k, y_{k-h}). \quad (2.3)$$

Entretanto, Brockwell e Davis (2002) citam que essas funções necessitam de um modelo teórico da série temporal. Como alternativa, pode-se aplicar a função de autocovariância amostral

e a função de autocorrelação amostral, consideradas aproximações das respectivas funções anteriormente apresentadas (MONTGONOMERY; JENNINGS; KULAHCI, 2015).

Seja N o número total de observações da série, calcula-se a média amostral da série representada como segue (BROCKWELL; DAVIS, 2002):

$$\bar{y} = \frac{1}{N} \sum_{k=1}^N y_k, \quad (2.4)$$

logo, a função de autocovariância amostral é:

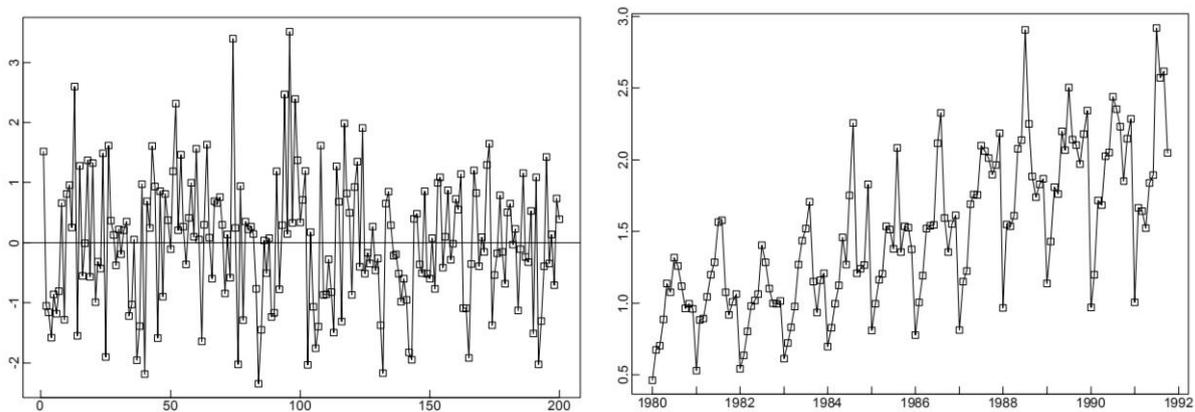
$$\hat{\gamma}_y(h) = \frac{1}{N} \sum_{k=1}^{N-|h|} (y_{k-|h|} - \bar{y})(y_k - \bar{y}), \quad -N < h < N, \quad (2.5)$$

e a função de autocorrelação amostral é definida por:

$$\hat{\rho}_y(h) = \frac{\hat{\gamma}_y(h)}{\hat{\gamma}_y(0)}, \quad -N < h < N. \quad (2.6)$$

Para indicar a estacionariedade, aproximadamente 95% dos valores de autocorrelação amostral devem estar situados entre os limites de $\pm 1.96/N$, para N sendo a quantidade de observações disponíveis da série temporal (BROCKWELL; DAVIS, 2002). Como exemplo, as Figuras 2.(a) e 2.(b) mostram respectivamente uma série temporal estacionária (ruído branco) e uma do tipo não-estacionária. Realizando primeiramente uma análise por inspeção visual, percebe-se que a primeira varia em torno de um valor médio, enquanto o segundo caso apresenta diferentes padrões nos dados, como uma tendência de crescimento e um comportamento sazonal. Para confirmar essa interpretação, aplicou-se a função de autocorrelação amostral para ambos os casos.

Figura 2 – Séries Temporais para Análise da Estacionariedade.

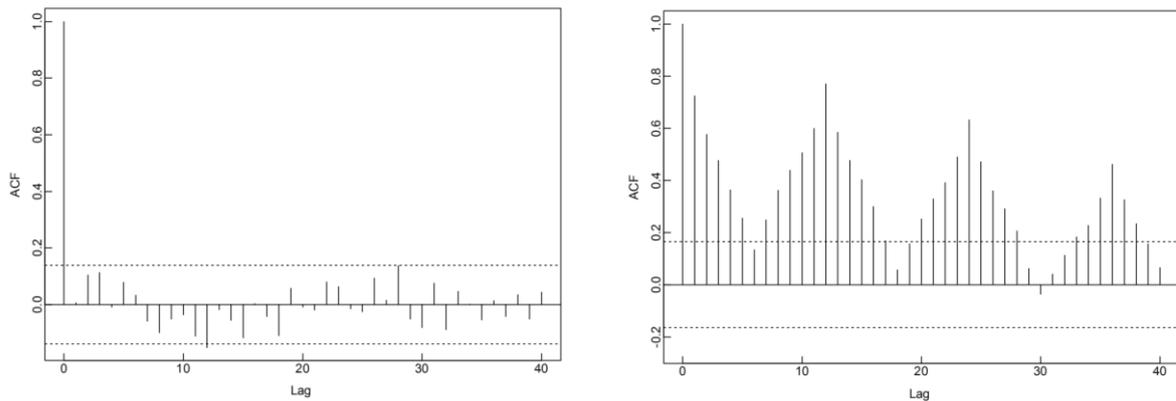


(a) Ruído Branco.

(b) Vendas em Milhares de Vinhos Australianos.

Fonte: (BROCKWELL; DAVIS, 2002).

Figura 3 – Funções de Autocorrelação Amostral das Séries da Figura 2. *Lag* é o atraso h e ACF significa *Autocorrelation Function*.



(a) Função de Autocorrelação Amostral da Figura 2.(a). (b) Função de Autocorrelação Amostral da Figura 2.(b).

Fonte: (BROCKWELL; DAVIS, 2002).

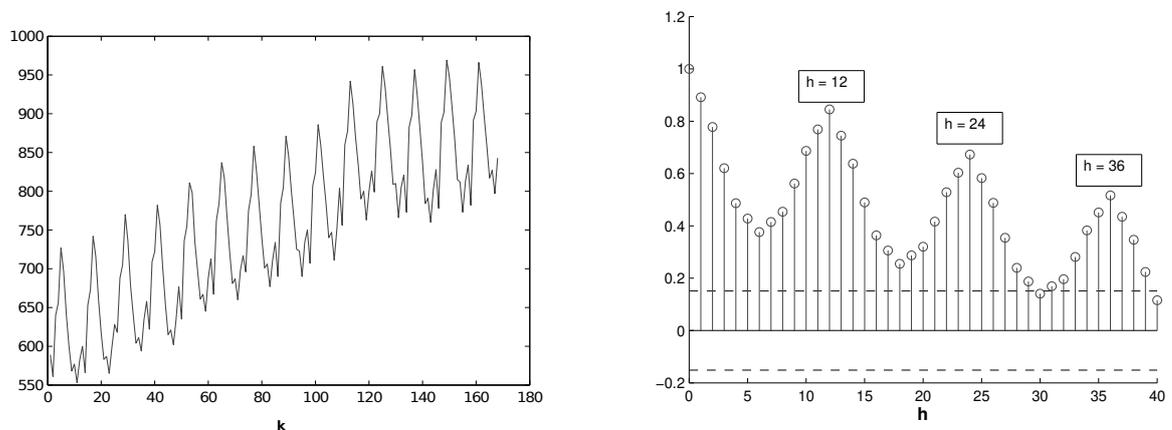
As Figuras 3.(a) e 3.(b) exibem as funções de autocorrelação amostral de ambas as séries acima. No primeiro caso, verifica-se que pelo menos 95% das autocorrelações amostrais está entre os limites especificados, indicando a estacionariedade dessa série. Em contrapartida, isso não ocorre para o segundo caso, devido a presença de muitos valores além do limite $\pm 1.96/N$, confirmando-se a característica não-estacionária dessa série.

2.2.3 Séries Temporais Sazonais

As séries temporais sazonais têm como principal propriedade exibir um comportamento periódico, sendo anual, mensal ou semanal, por exemplo. Uma série que apresenta um padrão sazonal, como consequência, também é classificada como não-estacionária (PALIT; POPOVIC, 2005). O período sazonal, denotado por s , pode ser indicado pelo conhecimento do especialista, determinado com a visualização dos dados, ou estimado segundo alguns critérios (HYNDMAN; ATHANASOPOULOS, 2014). Para comprovar a sazonalidade e verificar qual o período sazonal s , pode-se analisar o comportamento da função de autocorrelação amostral a partir dos dados. Por exemplo, seja a série temporal representada na Figura 4.(a) com dados mensais da taxa de produção de leite (libras por cada vaca) em determinada região durante o período de janeiro de 1962 a dezembro de 1975, ou seja, total de $N = 167$ meses, onde $k = 1, \dots, N$.

Nesse caso, a função de autocorrelação amostral é exibida na Figura 4.(b), na qual é possível observar o comportamento periódico para $h = 12$, $h = 24$ e $h = 36$. Devido a esse padrão, conclui-se que o período sazonal dessa série temporal é $s = 12$, que é o passo entre os valores de h indicados. Note ainda que esta série não é estacionária, já que ultrapassa os limites de $\pm 1.96/N$, para N sendo a quantidade de observações. Outro caso de sazonalidade para $s = 12$ pode ser observado nas Figuras 2.(b) e 3.(b) da seção anterior, sendo que o comportamento sazonal é comprovado pela função de autocorrelação amostral.

Figura 4 – Série Temporal Sazonal e sua Função de Autocorrelação Amostral.



(a) Série Temporal da Taxa de Produção de Leite Mensal (b) Função de Autocorrelação Amostral da Série de Taxa de Produção de Leite.

Fonte: Adaptado de (BROCKWELL; DAVIS, 2002).

2.2.4 Séries Univariáveis e Multivariáveis

Destaca-se ainda a classificação entre séries univariáveis e multivariáveis. As séries temporais univariáveis possuem dados relacionados a apenas uma variável dependente do tempo. Por outro lado, séries temporais multivariáveis são compostas por observações de duas ou mais séries temporais simultâneas, sendo possível representá-las por:

$$\mathbf{y}_k = [y_k^1 \ y_k^2 \ \cdots \ y_k^m], \quad (2.7)$$

onde \mathbf{y}_k é uma observação da série temporal multivariável no instante k , y_k^l ($l = 1, \dots, m$) indica a l -ésima série temporal e m é a quantidade de séries temporais que formam \mathbf{y}_k . Geralmente, os dados das séries temporais em problemas reais são multivariáveis, sendo que diversos processos apresentam dependências e correlações entre si. A complexidade em se trabalhar com séries temporais multivariáveis é principalmente devido a necessidade de modelar essas dependências (TSAY, 2014). Em princípio, as séries temporais univariáveis podem apresentar resultados suficientes ao reconstruir a dinâmica dos sistemas. No entanto, em alguns casos pode haver grandes ganhos no uso de todas as medições e variáveis disponíveis. Como apresenta o trabalho de Chayama e Hirata (2016), para estudos de curto prazo, a previsão de série temporal univariável se mostrou superior à multivariável de modo geral, enquanto para previsões de longo prazo ou na presença de grandes níveis de ruído, a previsão multivariável teve um desempenho melhor nas análises aplicadas. Entretanto, destaca-se nesse mesmo estudo que, de forma ampla, essas questões dependem de cada problema e estudo envolvendo séries específicas.

2.3 Aspectos Gerais Sobre a Previsão de Séries Temporais

A previsão de séries temporais é realizada a partir de métodos que visam a obtenção de dados futuros, baseando-se em suas observações passadas e/ou em outros dados relacionados a série a ser prevista (KOTU; DESHPANDE, 2019). A previsão é um dos principais objetivos ao analisar séries temporais, sendo aplicada em muitas situações como, por exemplo, para prever demandas futuras ou o comportamento de determinadas variáveis a um prazo de minutos, dias ou mesmo anos. Ademais, os resultados da previsão são relevantes para planejar ações e para o processo de tomada de decisão. Os métodos de previsão visam capturar os padrões e relacionamentos que existem nos dados e tentam reconstruir esses comportamentos para realizar a previsão (BROCKWELL; DAVIS, 2002). Contudo, alguns tipos de séries possuem pouca previsibilidade, ou seja, são difíceis de prever devido a sua complexidade. De acordo com Hyndman e Athanasopoulos (2014), uma boa previsibilidade depende de vários fatores, como a quantidade de dados disponíveis da série, do seu desenvolvimento dinâmico e de quão bem os analistas entendem os fatores que contribuem para a previsão.

Dado um conjunto de N observações ou amostras conhecidas de uma série temporal \mathbf{y}_k , para $k = 1, \dots, N$. A previsão visa obter os dados $\mathbf{y}_{N+1}, \mathbf{y}_{N+2}, \dots, \mathbf{y}_{N+H}$, para H o horizonte de previsão (HAMILTON, 1994). O horizonte de previsão corresponde a quantidade de observações que se deseja prever. Geralmente, o valor de H é definido de acordo com a natureza do problema e dos objetivos dos especialistas (MONTGONOMERY; JENNINGS; KULAHCI, 2015).

De acordo com a definição do horizonte H , as previsões podem ser classificadas quanto ao prazo: curto, médio ou longo prazo. Os erros de previsão a curto e médio prazo usualmente são menores que no longo prazo em uma previsão do tipo recorrente (Seção 2.3.3.1), sobretudo devido ao acúmulo desses erros que podem influenciar mais significativamente no longo prazo ou devido a mudança de comportamento da série no futuro mais longínquo (BROCKWELL; DAVIS, 2002). Todavia, a distinção de curto, médio e longo prazo é subjetiva e depende da definição do especialista e de cada caso específico analisado.

2.3.1 Previsão Qualitativa e Quantitativa: Disponibilidade das Observações das Séries Temporais

De acordo com Montgomery, Jennings e Kulahci (2015), há duas grandes classes de técnicas de previsão quando se considera a disponibilidade de observações das séries temporais: os métodos qualitativos e os métodos quantitativos. Seus conceitos serão apresentados nas Subseções 2.3.1.1 e 2.3.1.2 a seguir.

2.3.1.1 Previsão Qualitativa

As técnicas de previsão qualitativa são de natureza subjetiva, ou seja, consideram o julgamento por parte dos especialistas. Essas previsões são frequentemente aplicadas em situações em

que há pouca ou nenhuma observação do comportamento histórico disponível (CHAMBERS; MULLICK; SMITH, 1971). Os métodos para previsão qualitativa variam em relação ao custo e complexidade. Eles podem ser usados separadamente, porém, são geralmente integrados com outros métodos qualitativos ou com abordagens quantitativas (MAKRIDAKIS; WHEELWRIGHT; HYNDMAN, 1998).

A alternativa de previsão qualitativa mais amplamente utilizada é o Método Delphi (DALKEY, 1967). Este método considera um conjunto de especialistas que devem estar fisicamente separados para evitar que suas opiniões sejam afetadas por outros indivíduos. Cada um deve responder perguntas sobre o domínio do problema. Depois da aplicação desse primeiro questionário, as próximas perguntas são submetidas a cada especialista, assim como 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 converjam 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, pois a qualidade da previsão depende desses envolvidos (DIETZ, 1987).

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. 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. O erro das respostas ocorre 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 ou quando há pouca informação 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 foram 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).

2.3.1.2 *Previsão Quantitativa*

As técnicas de previsão quantitativa usam um modelo de previsão que precisa ser formulado com base em um grande conjunto de observações históricas da série temporal. Esse modelo relaciona os padrões nos dados e expressa uma relação entre os valores anteriores e os valores atuais da variável, sendo aplicado em seguida para projetar observações no futuro. Segundo

Goodwin (2002), os resultados dos métodos quantitativos podem ser refinados pelos qualitativos ao considerar o conhecimento subjetivo do especialista.

O primeiro passo na abordagem quantitativa é a definição do problema, que consiste na compreensão de como os resultados da previsão serão aplicados em determinado caso. Em seguida, define-se o horizonte de previsão H e o intervalo de previsão (MONTGONOMERY; JENNINGS; KULAHCI, 2015). Na coleta de dados da série temporal, busca-se obter informações históricas relevantes para realização da previsão, pois nem todos os dados são úteis e alguns podem prejudicar o desempenho do modelo de previsão. Além disso, muitas vezes é necessário lidar com valores ausentes da série, valores discrepantes (*outliers*) ou outros problemas relacionados aos dados coletados. Dessa maneira, a análise ou o pré-processamento dos dados é uma estratégia interessante de ser aplicada antes de selecionar modelos candidatos para previsão, visando resolver esses desafios supracitados.

Ressalta-se a importância de inspecionar visualmente os dados e verificar se há padrões como tendências, sazonalidades e outras componentes cíclicas (BROCKWELL; DAVIS, 2002). Essa visualização é relevante, pois proporciona uma intuição ao analista sobre quais modelos serão adequados de acordo com o comportamento da série temporal. Após o pré-processamento dos dados e a inspeção visual da série temporal, deve-se 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, podem ser considerados modelos regressivos, de suavização exponencial, ou representações no espaço de estados (BROCKWELL; DAVIS, 2002). Porém, para séries temporais complexas, pode ser necessário utilizar modelos não-lineares ou mesmo abordagens híbridas que unem métodos clássicos e técnicas de inteligência computacional, como redes neurais, sistemas *fuzzy*, *neuro-fuzzy*, etc (ABDOLLAHZADE et al., 2015).

De outro modo, a validação do modelo verifica se o modelo de previsão atende os requisitos de desempenho que satisfaça as exigências pré-definidas (MONTGONOMERY; JENNINGS; KULAHCI, 2015). 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 (AGUIRRE, 2015).

Finalmente, é importante garantir que o usuário do modelo de previsão desenvolvido entenda como aplicá-lo e como gerar previsões úteis. Deve-se realizar o constante monitoramento do desempenho do modelo de previsão. Um modelo que funcionou bem em uma situação pode deteriorar seu desempenho no futuro, pois a dinâmica dos dados e demais condições podem mudar 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 importante nos sistemas de previsão (BROCKWELL; DAVIS, 2002) (MONTGONOMERY;

JENNINGS; KULAHCI, 2015).

2.3.2 Boas Práticas para a Previsão de Séries Temporais

Ao selecionar um método de previsão, o analista da série temporal deve considerar 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 (HYNDMAN; ATHANASOPOULOS, 2014).

Segundo Hyndman e Athanasopoulos (2014), o procedimento de previsão geralmente envolve cinco etapas básicas:

1. **Definição do problema:** é uma etapa bem subjetiva, pois requer um entendimento de como as previsões serão aplicadas e quem irá utilizar os resultados das previsões. Portanto, deve-se realizar todo um levantamento de requisitos junto aos especialistas na coleta de dados e no uso das previsões;
2. **Levantamento de informações:** somente os dados brutos da série temporal podem ser insuficientes para realizar a previsão, sendo assim, métodos qualitativos de previsão podem ser considerados para que modelos de previsão melhores sejam desenvolvidos;
3. **Análise preliminar:** 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;
4. **Escolha e ajuste dos modelos:** nessa etapa, identifica-se o modelo apropriado para prever observações da série, avaliando e comparando modelos candidatos em potencial;
5. **Aplicação e avaliação do modelo de previsão:** Uma vez que um modelo tenha sido selecionado e seus parâmetros estimados, este é aplicado para fazer previsões. O desempenho do modelo pode ser avaliado e validado por medidas estatísticas.
6. **Revisão e atualização dos modelos:** revisar e ajustar os modelos periodicamente são relevantes para que continuem realizando previsões que atendam os requisitos preestabelecidos, pois, ao longo do tempo, o desempenho do modelo ajustado tende a deteriorar.

2.3.3 Estratégias para Previsão de Múltiplos Passos a Frente

Os métodos de previsão de séries temporais podem ser de um passo a frente ou de múltiplos passos a frente. Para a previsão de um passo a frente, tem-se a seguinte formulação:

$$\hat{y}_{k+1} = M(y_k, y_{k-1}, y_{k-2}, \dots, y_{k-r}) \quad (2.8)$$

sendo que \hat{y}_{k+1} é a previsão de uma série um passo a frente, e $M(\cdot)$ corresponde a um modelo que irá prever a série temporal com base em suas $r + 1$ observações passadas, do instante $k - r$

ao k . Todavia, somente a previsão um passo a frente não é um indicativo suficiente para avaliar a qualidade e desempenho de um modelo (BONTEMPI; TAIEB; BORGNE, 2013) (AGUIRRE, 2015).

Para que efetivamente se possa avaliar a qualidade de um modelo, é interessante também aplicar a abordagem de múltiplos passos a frente, verificando o desempenho dos métodos de previsão para além de um passo apenas (TAIEB; BONTEMPI, 2011). Como exemplo, seja a Tabela 1 a seguir que informa dados de uma série temporal hipotética da quantidade de vendas de um determinado produto:

Tabela 1 – Série Temporal de Vendas de um Produto por Mês.

Mês	Vendas
Janeiro	102
Fevereiro	105
Março	98
Abril	114
Mai	103
Junho	95
Julho	?
Agosto	?
Setembro	?

Fonte: Autor.

A abordagem de um passo a frente iria obter apenas a previsão no mês julho. De outra forma, para uma previsão de três passos a frente, os resultados serão os meses de julho, agosto e setembro. Entretanto, quanto mais adiante se projeta, mais incerteza está associada à previsão (TAIEB; BONTEMPI, 2011). A fim de realizar a previsão de múltiplos passos a frente, destacam-se 2 abordagens básicas na literatura, cujas descrições serão apresentadas nas subseções a seguir.

2.3.3.1 Previsão Recorrente

A estratégia recorrente envolve o uso de um modelo cuja previsão computada é considerada para obter uma nova previsão no passo seguinte (TAIEB; BONTEMPI, 2011). Portanto, tem-se para um horizonte de previsão H :

$$\begin{aligned}
 \hat{y}_{k+1} &= M(y_k, y_{k-1}, y_{k-2}, \dots, y_{k-r}) \\
 \hat{y}_{k+2} &= M(\hat{y}_{k+1}, y_k, y_{k-1}, \dots, y_{k-r+1}) \\
 \hat{y}_{k+3} &= M(\hat{y}_{k+2}, \hat{y}_{k+1}, y_k, \dots, y_{k-r+2}) \\
 &\vdots \\
 \hat{y}_{k+H} &= M(\hat{y}_{k+H-1}, \hat{y}_{k+H-2}, \hat{y}_{k+H-3}, \dots, \hat{y}_{k-r+H-1})
 \end{aligned} \tag{2.9}$$

sendo y_k a observação conhecida da série no instante k e $\hat{y}_{k+1}, \hat{y}_{k+2}, \hat{y}_{k+3}, \dots, \hat{y}_{k+H}$ são as previsões computadas pelo modelo $M(\cdot)$, sabendo que esse modelo considera as últimas $r + 1$

observações históricas da série como entradas para obter a previsão. Conforme se observa na equação acima na estratégia recorrente, o modelo usualmente realiza H previsões de um passo a frente até que o valor final no horizonte de previsão seja alcançado (TAIEB; BONTEMPI, 2011).

Como as previsões calculadas $\hat{y}_{k+1}, \hat{y}_{k+2}, \hat{y}_{k+3}, \dots$ são realimentadas como entradas do modelo, a estratégia recorrente permite que erros de previsão se acumulem, de modo que seja possível que o desempenho se degrade à medida que o horizonte de previsão aumenta. Apesar disso, essa estratégia examina a relação entre as previsões sucessivas e é muito utilizada por boa parte das abordagens de previsão de séries temporais, devido a sua simplicidade e baixo custo computacional (BONTEMPI; TAIEB; BORGNE, 2013).

2.3.3.2 Previsão Direta

A previsão direta necessita do desenvolvimento de um modelo separado para cada tipo de mapeamento realizado, ou seja, são calculadas as previsões seguindo esse mapeamento até que o horizonte H seja alcançado (TAIEB; BONTEMPI, 2011). Esse mapeamento deve considerar as observações históricas da série até o instante atual k , sendo que um modelo deve ser treinado para computar dados futuros até o instante $k + p$, sendo p o passo adiante que se deseja prever. De forma geral, sejam as $r + 1$ últimas observações da série temporal disponíveis como regressores e o passo $p = 1$. Considerando o horizonte de previsão H , tem-se o modelo $M_1(\cdot)$ com mapeamento $p = 1$:

$$\begin{aligned}\hat{y}_{k+1} &= M_1(y_k, y_{k-1}, y_{k-2}, \dots, y_{k-r}) \\ \hat{y}_{k+2} &= M_1(y_{k+1}, y_k, y_{k-1}, \dots, y_{k-r+1}) \\ \hat{y}_{k+3} &= M_1(y_{k+2}, y_{k+1}, y_k, \dots, y_{k-r+2}) \\ &\vdots \\ \hat{y}_{k+H} &= M_1(y_{k+H-1}, y_{k+H-2}, y_{k+H-3}, \dots, y_{k-r+H-1}),\end{aligned}\tag{2.10}$$

sendo y_k a observação conhecida da série no instante k e $\hat{y}_{k+1}, \hat{y}_{k+2}, \hat{y}_{k+3}, \dots, \hat{y}_{k+H}$ são as previsões computadas pelo modelo $M_1(\cdot)$ para $p = 1$. Logo, esse modelo vai realizando previsões de um passo a frente até alcançar o horizonte H . Diferente da previsão recorrente que realimenta suas próprias previsões como entradas no modelo, a abordagem direta usa dados disponíveis da própria série para computar as previsões.

Modificando-se o valor de p , são necessários novos modelos para efetuar o mapeamento do instante $k \rightarrow k + p$. Como exemplo, busca-se desenvolver três modelos para cada mapeamento ou passo $p = 1, 2, 3$ conforme a seguir:

$$\begin{aligned}\hat{y}_{k+1} &= M_1(y_k, y_{k-1}, y_{k-2}, \dots, y_{k-r}), \quad \text{para } p = 1, \\ \hat{y}_{k+2} &= M_2(y_k, y_{k-1}, y_{k-2}, \dots, y_{k-r}), \quad \text{para } p = 2, \\ \hat{y}_{k+3} &= M_3(y_k, y_{k-1}, y_{k-2}, \dots, y_{k-r}), \quad \text{para } p = 3,\end{aligned}\tag{2.11}$$

onde $r + 1$ é o número de observações passadas consideradas como entradas para realizar a previsão e M_1, M_2 e M_3 são distintos modelos para cada mapeamento definido. Destaca-se que,

nesse exemplo, o modelo M_1 pode ser aplicado apenas para obter a previsão um passo a frente até o horizonte H (conforme Eq. 2.10), enquanto o M_3 é apropriado para prever dados com 3 passos adiante.

Como essa estratégia não utiliza as próprias previsões para computar novos resultados, a propagação de erros ao longo do horizonte de previsão é reduzida. No entanto, pode apresentar uma carga computacional maior devido a quantidade de modelos diferentes para cada mapeamento. Além disso, como diferentes modelos desacoplados são usados, isso significa que estes não avaliam as dependências entre as previsões, que podem contribuir para melhores resultados (BONTEMPI; TAIEB; BORGNE, 2013).

2.3.4 Avaliação dos Resultados de Previsão de Séries Temporais

Para comparar o desempenho dos diferentes métodos de previsão de séries temporais, são necessárias métricas de erro. Dessa forma, os dados da série temporal são subdivididos em dados para treinamento e dados para teste, sendo que ambos devem apresentar observações distintas da série. Os dados de treinamento são usados para estimar quaisquer parâmetros e a estrutura do modelo, preparando-o para realizar a previsão. Enquanto os dados de teste são aplicados para avaliar sua performance durante a previsão (HYNDMAN; ATHANASOPOULOS, 2014).

Os resíduos em um modelo de série temporal correspondem aos erros na etapa de treinamento. Calcula-se esses resíduos por meio da diferença entre as observações reais da série e os valores ajustados correspondentes, isto é:

$$e_k = y_k - \hat{y}_k, \quad k = 1, \dots, N, \quad (2.12)$$

para N a quantidade de observações usadas no treinamento, y_k a observação real da série no instante k e \hat{y}_k a aproximação ajustada pelo modelo no treinamento. A avaliação do comportamento do resíduo é útil para verificar se os modelos desenvolvidos para previsão conseguiram capturar os padrões nos dados no treinamento. Logo, um modelo ideal apresentaria em seus resíduos apenas flutuações aleatórias que não podem ser modeladas (HYNDMAN; ATHANASOPOULOS, 2014).

Segundo Hyndman e Athanasopoulos (2014), os resíduos devem apresentar as seguintes características:

- As amostras dos resíduos devem ser não-correlacionadas, pois se houver correlação, então há ainda padrões que podem ser úteis para previsão da série temporal que o modelo não considerou. Para efetuar essa análise, a função de autocorrelação pode ser aplicada;
- Os resíduos devem ter média zero, caso contrário apresenta um viés. Para resolver esse problema, basta somar essa média diferente de zero ao resultado no treinamento do modelo, o que pode melhorar os resultados de previsão em alguns casos.

Sendo assim, qualquer método cujos resíduos não obedecem ambas premissas ainda pode ser melhorado (HYNDMAN; ATHANASOPOULOS, 2014). Além dessas propriedades, é útil mas não necessário que os resíduos tenham também as duas características a seguir:

- Os resíduos devem ter variância constante;
- Os resíduos devem ter distribuição de probabilidade normal.

No entanto, nem sempre essas duas condições acima vão impactar de forma significativa nos resultados de previsão (HYNDMAN; ATHANASOPOULOS, 2014). Makridakis, Spiliotis e Assimakopoulos (2018) destacam que um modelo que se ajuste bem aos dados de treinamento não garante necessariamente uma previsão de qualidade.

Tendo em mãos um modelo de previsão que apresente resíduos que atendam aos requisitos de forma aceitável, pode-se avaliar efetivamente o desempenho na previsão (KOTU; DESHPANDE, 2019). Seja o erro de previsão a diferença entre o valor real e o valor previsto pelo modelo, conforme a seguir:

$$e_{N+h} = y_{N+h} - \hat{y}_{N+h}, \quad h = 1, \dots, H, \quad (2.13)$$

para N a quantidade de observações usadas no treinamento, H o horizonte de previsão, e_{N+h} o erro de previsão, y_{N+h} a observação real da série temporal e \hat{y}_{N+h} sendo a previsão computada. A seguir, serão descritas algumas das principais métricas de desempenho da previsão de séries temporais.

2.3.4.1 Erro Absoluto Médio

O Erro Absoluto Médio (*Mean Absolute Error*, MAE) é calculado usando a seguinte formulação (KOTU; DESHPANDE, 2019):

$$\text{MAE} = \frac{\sum_{h=1}^H |y_{N+h} - \hat{y}_{N+h}|}{H} = \frac{\sum_{h=1}^H |e_{N+h}|}{H}. \quad (2.14)$$

O MAE mede a magnitude média dos erros em um conjunto de previsões, tal que todas as diferenças individuais tenham peso igual. Essa métrica é adequada para comparar métodos que usam a mesma série temporal ou quando diferentes séries apresentam observações na mesma escala ou unidade de dados (KOTU; DESHPANDE, 2019).

2.3.4.2 Erro Quadrático Médio

A métrica denominada Erro Quadrático Médio (*Mean Squared Error*, MSE) realiza a medida da média dos quadrados dos erros, ou seja, a diferença média quadrática entre o valor

previsto e o valor real da série, conforme a fórmula a seguir (SAMMUT; WEBB, 2010):

$$\text{MSE} = \frac{1}{H} \sum_{h=1}^H (e_{N+h})^2. \quad (2.15)$$

Logo, o MSE verifica quão próxima uma linha ajustada está dos pontos de dados reais. O MSE tem como característica colocar um peso maior nos erros mais discrepantes. Contudo, tem a desvantagem de representar as medidas em função dos quadrados das escalas da série temporal, sendo mais difícil de interpretar que o RMSE a seguir (SAMMUT; WEBB, 2010).

2.3.4.3 Raiz do Erro Quadrático Médio

Por sua vez, a Raiz do Erro Quadrático Médio (*Root Mean Squared Error*, RMSE) é a raiz quadrada da média das diferenças quadráticas entre a previsão e a observação real da série temporal, ou seja, a raiz quadrada do MSE (KOTU; DESHPANDE, 2019). Sua formulação matemática é:

$$\text{RMSE} = \sqrt{\frac{1}{H} \sum_{h=1}^H (e_{N+h})^2}. \quad (2.16)$$

Assim como o MAE e MSE, o RMSE também é dependente da escala dos dados, ou seja, não podem ser utilizadas para comparação envolvendo séries temporais com diferentes escalas de dados. Assim como no MSE, como os erros são elevados ao quadrado antes da média, o RMSE atribui um peso relativamente alto a erros maiores. Sendo assim, o RMSE é útil quando deseja-se penalizar erros grandes (HYNDMAN; ATHANASOPOULOS, 2014). O RMSE pode ser interpretado diretamente em termos de unidades da série, portanto, é uma medida de ajuste mais fácil de ser interpretada que o MSE.

2.3.4.4 Erro Absoluto Médio Percentual

Ao contrário das métricas anteriores, o Erro Absoluto Médio Percentual (*Mean Absolute Percentage Error*, MAPE) é muito útil para comparação entre vários modelos diferentes usando séries temporais em diferentes escalas. Isso porque sua representação é percentual. A seguir, a fórmula do MAPE é apresentada (HYNDMAN; ATHANASOPOULOS, 2014):

$$\text{MAPE} = \frac{100}{H} \sum_{h=1}^H \left| \frac{e_{N+h}}{y_{N+h}} \right|. \quad (2.17)$$

Embora o MAPE seja simples, tem uma limitação significativa quando se aplica a dados em que valores nulos ou próximos de zero são possíveis, pois pode tornar-se infinita ou indefinida.

2.3.4.5 Erro Absoluto Médio em Escala

Outra métrica importante é o Erro Absoluto Médio em Escala (*Mean Absolute Scaled Error*, MASE), que supera as limitações do MAPE, comparando os resultados do método

analisado com a previsão do Método Naïve, que será comentado na Seção 2.4.1. Sendo assim, a formulação matemática com MASE é (HYNDMAN; ATHANASOPOULOS, 2014):

$$\text{MASE} = \frac{\sum_{h=1}^H |e_{N+h}|}{\frac{H}{H-1} \sum_{h=2}^H |y_{N+h} - y_{N+h-1}|}. \quad (2.18)$$

Acentua-se que o MASE é independente da escala dos dados. Caso o resultado do MASE seja menor que 1, obteve-se uma previsão melhor que o método Naïve. Se o MASE for maior que 1, significa que o método Naïve teve um melhor desempenho que a outra abordagem aplicada para previsão. Para séries temporais sazonais, o MASE pode ser ajustado conforme a seguir (HYNDMAN; ATHANASOPOULOS, 2014):

$$\text{MASE} = \frac{\sum_{h=1}^H |e_{N+h}|}{\frac{H}{H-s} \sum_{h=s+1}^H |y_{N+h} - y_{N+h-s}|}, \text{ para } H > s, \quad (2.19)$$

sendo s o período sazonal.

2.3.5 Intervalos de Previsão

A previsão de séries temporais envolve incertezas, devido ao objetivo de se obter observações futuras desconhecidas. Para formalizar essas incertezas, os intervalos de confiança podem ser aplicados. Um intervalo de confiança pode ser definido como uma faixa calculada a partir dos dados de tal forma que haja uma probabilidade da observação estar incluída nessa faixa. Essa probabilidade é chamada de nível de confiança, que em muitos casos é definido tradicionalmente em 95%, embora os níveis 90%, 99% e 99,9% também sejam comumente usados. Entretanto o valor de 95% será aplicado nos estudos dessa tese. Quanto maior for o nível de confiança, maior será o intervalo de confiança (SIEGEL, 2012) (RAMACHANDRAN; TSOKOS, 2021).

Para aplicação dos intervalos de confiança, há duas premissas necessárias: disponibilidade de dados aleatórios da população avaliada; e os dados devem ser normalmente distribuídos (SIEGEL, 2012). De acordo com o teorema central do limite, a segunda suposição será satisfeita desde que a quantidade de observações disponíveis seja grande o suficiente, fazendo com que a distribuição amostral se aproxime cada vez mais de uma distribuição normal (HOFFMAN, 2019).

De forma específica, o intervalo de previsão corresponde a uma faixa de valores que indica onde se espera que uma nova observação futura estará, com base em um modelo existente. Assim como para o intervalo de confiança, a incerteza associada ao intervalo de previsão deve ser definida antecipadamente e faz parte do cálculo dessa faixa (SIEGEL, 2012).

Os intervalos de previsão são frequentemente confundidos com intervalos de confiança. Embora estejam relacionados, os dois processos têm cálculos e finalidades diferentes. A distinção básica entre os dois é que o intervalo de previsão indica em que faixa uma observação individual futura cairá, enquanto um intervalo de confiança mostra a faixa provável de valores associados a algum parâmetro estatístico dos dados, como a média da população. Esta é uma distinção importante, porque o intervalo de confiança dos valores médios para as populações amostradas será muito menor ou mais estreito do que o intervalo de predição para os mesmos dados (WILCOX, 2017).

Seja Y_{N+h} uma variável aleatória com Q realizações representadas por y_{N+h} , ou seja, um experimento com essa variável foi executado Q vezes. Essa variável aleatória representa o resultado de previsão de um método no instante $N+h$, para cada $h = 1, \dots, H$, lembrando que H é o horizonte de previsão e N é a quantidade de dados disponível da série temporal. Para obter o intervalo de previsão, o erro padrão de previsão, que mensura a variabilidade da distância entre a média da amostra e a nova observação, é calculado por meio da seguinte fórmula (SIEGEL, 2012):

$$\text{erro padrão de previsão} = S_d \sqrt{1 + \frac{1}{Q}}, \quad (2.20)$$

para S_d o desvio padrão da variável aleatória Y_{N+h} e Q a quantidade de realizações y_{N+h} dessa variável aleatória. Segundo Siegel (2012), a fórmula que identifica os limites do intervalo de previsão para uma nova observação é representada por:

$$\bar{y}_{N+h} \pm t_\alpha S_d \sqrt{1 + \frac{1}{Q}}, \quad (2.21)$$

onde \bar{y}_{N+h} é a média das Q realizações da variável aleatória e t_α está relacionado ao valor equivalente da distribuição de *t-student*, relacionado ao nível de confiança desejado, sendo que esse valor é tabelado (SIEGEL, 2012). É importante o valor de Q ser grande o suficiente a fim de que a distribuição de probabilidade da variável aleatória Y_{N+h} seja aproximadamente normal, de acordo com o teorema do limite central, para que se tenha um intervalo de previsão melhor definido. Sendo assim, para $Q \rightarrow \infty$, o valor de t_α tende para 1.96, pois esse valor é baseado no fato de que 95% da área de uma distribuição normal está dentro de 1.96 desvios padrão da média. Note que para cada observação prevista considerando $h = 1, \dots, H$, é realizado Q experimentos para calcular a faixa de valores da Equação 2.21.

2.4 Métodos para Previsão de Séries Temporais

Nesta seção, alguns métodos tradicionais para previsão de séries temporais serão apresentados e descritos de forma sucinta.

2.4.1 Métodos Simples para Previsão de Séries Temporais

A abordagem mais elementar para previsão de séries temporais é o Método Naïve ou Método Ingênuo, cujo resultado de previsão é igual a última observação da série temporal (KOTU; DESHPANDE, 2019). Sejam y_1, y_2, \dots, y_N observações de uma série. A previsão resultante desse método é:

$$\hat{y}_{N+h} = y_N, \text{ para } h = 1, \dots, H. \quad (2.22)$$

onde \hat{y}_{N+h} é a previsão ao longo do horizonte de previsão H .

Uma variação do Método Naïve é conhecido como Método Naïve Sazonal (KOTU; DESHPANDE, 2019). Neste caso, a previsão é igual a observação da série temporal em um período sazonal anterior. Sendo assim:

$$\hat{y}_{N+h} = y_{N+h-s}, \text{ para } h = 1, \dots, H, \quad (2.23)$$

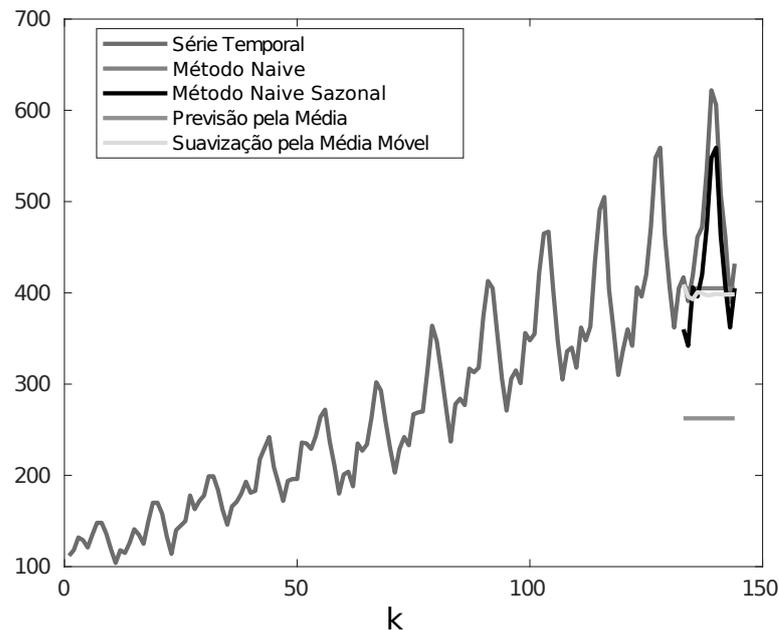
sendo que s é o período sazonal da série temporal.

Outro método simples para previsão de séries temporais é a Previsão pela Média, sendo que os valores futuros são simplesmente a média de todos os dados históricos da série temporal. Logo, o resultado da previsão pela média é denotado por:

$$\hat{y}_{N+h} = \bar{y} = \frac{1}{N} \sum_{k=1}^N y_k, \text{ para } h = 1, \dots, H. \quad (2.24)$$

Entretanto, uma das desvantagens do método da Previsão pela Média ocorre quando há uma grande quantidade de observações da série temporal, situação em que a média pode ter elevadas medidas de erro (HYNDMAN; ATHANASOPOULOS, 2014). Para atenuar esse problema, é possível utilizar uma técnica de janelamento, considerando apenas uma parte das observações para realizar a previsão. Logo, essa janela continua avançando a cada instante, calculando-se a média das observações contidas nela. Essa técnica é conhecida como Suavização pela Média Móvel. Todavia, essa abordagem ainda enfrenta problemas com dados sazonais, como afirmam Kotu e Deshpande (2019).

Na Figura 5, tem-se a previsão utilizando os quatro métodos supracitados em comparação com os dados reais da série temporal. Percebe-se que essas abordagens são elementares e nem todas as vezes irá fornecer bons resultados. Os experimentos apresentados nessa figura foram realizados considerando um horizonte de previsão $H = 12$, sendo o período sazonal $s = 12$ e o tamanho da janela do método de Suavização pela Média Móvel é de quatro observações.

Figura 5 – Exemplo de Previsão de uma Série Temporal Usando os Métodos Simples Apresentados ($H = 12$).

Fonte: Autores.

2.4.2 Suavização Exponencial

Os métodos de suavização exponencial formam uma grande classe de técnicas para a modelagem e previsão de séries temporais. As equações dessa classe de métodos são ponderadas exponencialmente por alguns parâmetros. Em outras palavras, as suas previsões são médias ponderadas de observações passadas, sendo que os pesos decaem exponencialmente à medida que as observações ficam mais antigas. Essa classe de métodos permite a decomposição de padrões como a tendência, o nível, a sazonalidade e um termo aleatório (BROCKWELL; DAVIS, 2002) (MONTGONOMERY; JENNINGS; KULAHCI, 2015). A tendência está relacionada com a diferença sequencial de dois níveis consecutivos, o nível é o valor médio da observação no período e a sazonalidade é um evento que se repete com uma periodicidade constante.

Uma das técnicas mais utilizadas dessa classe é o método de Holt-Winters (HW), que considera as três componentes supracitadas e o termo aleatório (SZMIT; SZMIT, 2012). Quando na inspeção visual da série temporal percebe-se que claramente há 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 (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 observação no instante k

de uma série temporal y_k com período sazonal s . Segundo Tratar e Strmcnik (2016) e Szmit e Szmit (2012), o modelo multiplicativo para obter um aproximação \hat{y}_k da série é expressado por:

$$\hat{y}_k = z_k s_k + q_k + w_k, \quad k = 1, \dots, N, \quad (2.25)$$

na qual z_k é o nível, s_k é a sazonalidade, q_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:

$$\begin{aligned} \text{(Nível)} \quad z_k &= A \left(\frac{y_k}{s_{k-s}} \right) + (1-A)(z_{k-1} + q_{k-1}), \quad 0 < A < 1, \quad k = s+1, \dots, N; \\ \text{(Tendência)} \quad q_k &= C(z_k - z_{k-1}) + (1-C)q_{k-1}, \quad 0 < C < 1, \quad k = s+1, \dots, N; \\ \text{(Sazonalidade)} \quad s_k &= D \left(\frac{y_k}{z_k} \right) + (1-D)s_{k-s}, \quad 0 < D < 1, \quad k = s+1, \dots, N, \end{aligned} \quad (2.26)$$

sendo que representam respectivamente o nível, a tendência e a sazonalidade, assim como suas constantes de suavização associadas A , C e D . A fórmula para previsão do método de HW multiplicativo para um horizonte de previsão H é:

$$\hat{y}_{k+h} = (z_k + hq_k)s_{k+h-s}, \quad h = 1, \dots, H. \quad (2.27)$$

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:

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

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

$$\begin{aligned} \text{(Nível)} \quad z_k &= A(y_k - s_{k-s}) + (1-A)(z_{k-1} + q_{k-1}), \quad 0 < A < 1, \quad k = s+1, \dots, N; \\ \text{(Tendência)} \quad q_k &= C(z_k - z_{k-1}) + (1-C)q_{k-1}, \quad 0 < C < 1, \quad k = s+1, \dots, N; \\ \text{(Sazonalidade)} \quad s_k &= D(y_k - z_k) + (1-D)s_{k-s}, \quad 0 < D < 1, \quad k = s+1, \dots, N. \end{aligned} \quad (2.29)$$

Por fim, a fórmula para previsão do método de HW aditivo para um horizonte de previsão H é (MORETTIN; TOLOI, 2006):

$$\hat{y}_{k+h} = z_k + hq_k + s_{k+h-s}, \quad h = 1, \dots, H. \quad (2.30)$$

Os valores iniciais para as equações das componentes de nível, tendência e sazonalidade em ambos os procedimentos apresentados podem ser calculados pelas seguintes fórmulas (MORETTIN; TOLOI, 2006):

$$\begin{aligned}
 s_j &= \frac{y_j}{\frac{1}{s} \sum_{i=1}^s y_i}, \quad j = 1, 2, \dots, s; \\
 z_s &= \frac{1}{s} \sum_{i=1}^s y_i; \\
 q_s &= 0.
 \end{aligned} \tag{2.31}$$

Há a necessidade de determinar valores apropriados para as constantes de suavização A , C e D (TRATAR; STRMCNIK, 2016) (SZMIT; SZMIT, 2012). Para isso, pode-se utilizar técnicas de otimização de maneira que possa tornar mínimo o erro quadrático médio dos erros de ajuste, ou seja:

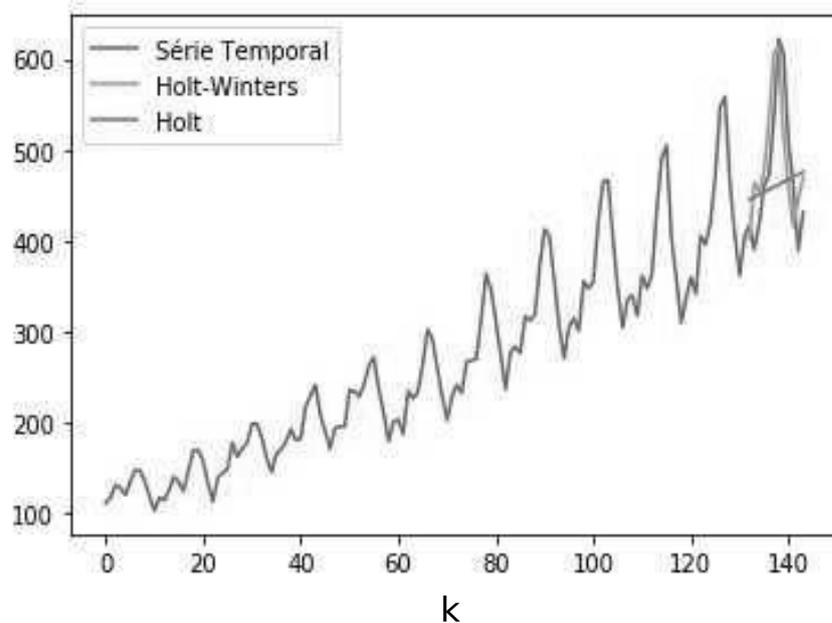
$$E = \frac{1}{N} \sum_{k=1}^N (y_k - \hat{y}_k)^2 \tag{2.32}$$

em que \hat{y}_k é a série temporal gerada pelo modelo multiplicativo ou aditivo.

O método de Holt é outra abordagem dessa classe de métodos e é muito semelhante ao método de HW, contudo, utiliza apenas as componentes de nível e tendência, sendo adequado para séries temporais que não apresentam comportamento sazonal destacado (KOTU; DESHPANDE, 2019). A seguir, as equações fundamentais do método de Holt:

$$\begin{aligned}
 (\text{Nível}) \quad z_k &= Ay_k + (1 - A)(z_{k-1} + q_{k-1}), \quad 0 < A < 1, \quad k = 2, \dots, N; \\
 (\text{Tendência}) \quad q_k &= C(z_k - z_{k-1}) + (1 - C)q_{k-1}, \quad 0 < C < 1, \quad k = 2, \dots, N; \\
 (\text{Previsão}) \quad \hat{y}_{k+h} &= z_k + hq_k.
 \end{aligned} \tag{2.33}$$

As condições iniciais desse caso são as mesmas apresentadas na Equação 2.31, ignorando a sazonalidade s_k . Os resultados para uma previsão aplicando os métodos de Holt-Winters e de Holt para $H = 12$ podem ser visualizados na Figura 6. Nesse caso, como a série temporal é sazonal, o método de Holt-Winters apresentou uma previsão melhor.

Figura 6 – Exemplo de Previsão de uma Série Temporal Usando os Métodos Holt-Winters e Holt ($H = 12$).

Fonte: Autores.

2.4.3 Modelos ARIMA

A classe de modelos Autorregressivos Integrados de Médias Móveis (*Autoregressive Integrated Moving Average* - ARIMA) é comumente utilizada em várias aplicações envolvendo séries temporais. Os modelos ARIMA são apropriados para lidarem com séries temporais não-estacionárias, aplicando um número finito de diferenças das suas observações sucessivas para eliminar a não-estacionariedade (MONTGONOMERY; JENNINGS; KULAHCI, 2015). A formulação matemática de um modelo ARIMA(p, d, q) é descrita como:

$$\left(1 - \sum_{i=1}^p \varphi_i B^i\right) (1 - B)^d y_k = \left(1 + \sum_{j=1}^q \theta_j B^j\right) w_k, \quad (2.34)$$

sendo que:

- p , d e q são inteiros maiores ou iguais a zero que representam a ordem dos termos autorregressivos, dos integrativos e dos termos de médias móveis, respectivamente.
- o parâmetro d indica a quantidade de diferenças das observações da série que serão realizadas. Normalmente, uma ou duas diferenças já remove a não-estacionariedade de uma série temporal.
- o operador B é um operador de atraso, onde $By_k = y_{k-1}$, $B^2 y_k = y_{k-2}$, ou de forma geral, $B^n y_k = y_{k-n}$;
- φ_i e θ_j são os parâmetros da parte autorregressiva e da parte da média móvel, respectivamente;

- w_k é um ruído gaussiano com média nula.

Como exemplo, um modelo ARIMA(1,1,1) pode ser formulado conforme a seguir:

$$(1 - \phi_1 B)(1 - B)y_k = (1 + \theta_1 B)w_k,$$

$$y_k - y_{k-1} = \phi_1(y_{k-1} - y_{k-2}) + w_k + \theta_1 w_{k-1}, \quad (2.35)$$

$$\Delta y_k = \phi_1 \Delta y_{k-1} + w_k + \theta_1 w_{k-1}, \text{ para } \Delta y_k = y_k - y_{k-1}.$$

Quando um modelo ARIMA($p, 0, 0$) é definido, ele passa a ser denominado por Modelo Autorregressivo de ordem p ou AR(p), mais apropriado para séries temporais estacionárias e com padrões mais simples (BROCKWELL; DAVIS, 2002). Um modelo AR(1) teria a seguinte formulação:

$$(1 - \phi_1 B)y_k = w_k, \quad (2.36)$$

$$y_k = \phi_1 y_{k-1} + w_k$$

Por sua vez, um modelo ARIMA(0,0, q) é denominado de Média Móvel de ordem q ou MA(q). Em vez de usar valores passados da própria série, esse modelo de média móvel usa erros de previsão passados em um modelo semelhante a regressão (BROCKWELL; DAVIS, 2002). Um modelo MA(1) é dado como exemplo a seguir:

$$y_k = (1 - \theta_1 B)w_k, \quad (2.37)$$

$$y_k = w_k - \theta_1 w_{k-1}.$$

Outro modelo importante dessa classe não considera a parte integrativa, sendo representado por ARIMA($p, 0, q$) ou ARMA(p, q) e denominado de Autorregressivo de Média Móvel. Essa abordagem é uma das mais utilizadas para trabalhar com séries temporais, considerando tanto observações passadas da série quanto termos do ruído. Entretanto, assim como AR(p) e MA(q), é mais apropriado para trabalhar com séries estacionárias (HYNDMAN; ATHANASOPOULOS, 2014). Um modelo ARMA(1,1) pode ser descrito como:

$$(1 - \phi_1 B)y_k = (1 + \theta_1 B)w_k, \quad (2.38)$$

$$y_k = \phi_1 y_{k-1} + w_k + \theta_1 w_{k-1}.$$

A Tabela 2 a seguir exhibe alguns dos principais casos especiais dos modelos ARIMA.

Tabela 2 – Principais Casos Especiais do Modelo ARIMA(p, d, q).

Modelo ARIMA(p, d, q)	Caso Especial
ARIMA(0,0,0)	Ruído Branco
ARIMA(0,1,0)	Passeio Aleatório (<i>Random Walk</i>)
ARIMA($p,0,0$)	Autorregressivo AR(p)
ARIMA(0,0, q)	Média Móvel MA(q)
ARIMA($p,0,q$)	Autorregressivo de Média Móvel ARMA(p, q)

Fonte: Adaptado de Hyndman e Athanasopoulos (2014).

Os modelos ARIMA consideram a autocorrelação entre os valores da série em instantes próximos e sucessivos. Contudo, existem situações que a série também pode apresentar autocorrelação com outros instantes, como é o caso das séries sazonais, por exemplo. Os modelos que contemplam as séries com autocorrelação sazonal são conhecidos como SARIMA (*Seasonal Autoregressive Integrated Moving Average*) (HYNDMAN; ATHANASOPOULOS, 2014). O modelo SARIMA contém uma parte não sazonal representada pelos mesmos parâmetros (p, d, q) do ARIMA e uma sazonal com novos parâmetros (P, D, Q)_s, sendo que:

- P : parâmetro para os termos autorregressivos sazonais;
- D : parâmetro para diferença sazonal;
- Q : parâmetro a média móvel sazonal;
- s : período sazonal da série temporal.

Dessa forma, o modelo geral SARIMA(p, d, q)(P, D, Q)_s é denotado por (HYNDMAN; ATHANASOPOULOS, 2014):

$$\left(1 - \sum_{i=1}^p \phi_i B^i\right) \left(1 - \sum_{i=1}^P \phi_i B^{s \cdot i}\right) (1 - B)^d (1 - B^s)^D y_k = \left(1 + \sum_{j=1}^q \theta_j B^j\right) \left(1 + \sum_{j=1}^Q \theta_j B^{s \cdot j}\right) w_k. \quad (2.39)$$

Um modelo SARIMA(1, 1, 1)(1, 1, 1)₄, por exemplo, pode ser escrito conforme a seguir:

$$(1 - \phi_1 B) (1 - \phi_1 B^4) (1 - B) (1 - B^4) y_k = (1 + \theta_1 B) (1 + \theta_1 B^4) w_k. \quad (2.40)$$

Para lidar com séries temporais multivariáveis, existe a classe de modelos VARIMA (*Vector Autoregressive Integrated Moving Average*) (MONTGONOMERY; JENNINGS; KULAHCI, 2015). Sua formulação matemática é:

$$\left(I - \sum_{i=1}^p \Phi_i B^i\right) \mathbf{D}(B) \mathbf{y}_k = \left(I + \sum_{j=1}^q \Theta_j B^j\right) \mathbf{w}_k, \quad (2.41)$$

sendo que I é a matriz identidade, \mathbf{y}_k representa uma série temporal multivariável, assim como \mathbf{w}_t é um ruído gaussiano multivariável e o termo $\mathbf{D}(B) = \text{diag} \{(1 - B)^{d_1}, (1 - B)^{d_2}, \dots, (1 - B)^{d_m}\}$,

para m a quantidade de séries. Note que para cada série, há um termo d que está relacionado a parte integrativa do modelo. Da mesma forma que o ARIMA univariável, há alguns casos especiais do modelo VARIMA, como exemplo, o VARMA e o VAR para séries multivariáveis estacionárias (MONTGONOMERY; JENNINGS; KULAHCI, 2015). Nessa tese, o modelo VAR é considerado nos consequentes das regras *fuzzy*.

2.5 Estado da Arte na Previsão de Séries Temporais

Diversas abordagens vêm sendo propostas para a previsão de séries temporais visando lidar com os desafios atuais, como o grande volume de dados e a velocidade com que estes são gerados. Sendo assim, existe a necessidade de trabalhar com essa quantidade crescente de dados disponíveis. Séries temporais reais geralmente são não-estacionárias, apresentam incertezas em seu comportamento e muitas possuem características cíclicas, 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*.

Para se ter um panorama dos avanços dessa área de pesquisa, em Gooijer e Hyndman (2006) é apresentado de forma geral um interessante histórico de trabalhos envolvendo a previsão de séries temporais no período 1982-2005. Este artigo analisa cerca de 40 trabalhos publicados altamente influentes sobre previsão de séries temporais, tratando de abordagens tradicionais como Suavização Exponencial, modelos ARIMA, modelos no Espaço de Estados, filtros de Kalman, modelos ARCH/GARCH, dentre outras alternativas que envolvem modelos não-lineares para previsão.

Os métodos tradicionais de suavização exponencial são ainda utilizados para a análise e previsão de séries temporais em muitas pesquisas. Singh et al. (2020) demonstram o desempenho de um modelo estatístico baseado no método clássico de Holt-Winters a fim de fornecer previsões da COVID-19 na Índia. Com base em séries temporais diárias de infecções acumuladas, infecções ativas e mortes, esses autores aplicaram seu modelo para prever essas variáveis da doença. Em Liu et al. (2020a), um modelo híbrido é proposto para previsões de curto prazo do consumo de eletricidade residencial com base no método Holt-Winters (HW) e na rede *Extreme Learning Machine* (ELM). Primeiramente, os dados originais são decompostos em uma componente linear estacionária e uma residual não-linear, usando um filtro de média móvel (MA). O método HW é responsável por estabelecer um modelo para a previsão da componente linear. Com os resultados dessa previsão linear, dados do resíduo não-linear e observações originais como entradas, o ELM pode construir um modelo de previsão não-linear para o consumo residencial de eletricidade. Outro trabalho que lida com aplicações na área de consumo de eletricidade está em Jiang et al. (2020), que propõem um método de previsão híbrido para prever essa série temporal. Os autores utilizam o método de HW otimizado pelo algoritmo *fruit fly*, cujo papel é selecionar os melhores parâmetros de suavização exponencial. Em Godinho, Maçaira e Souza (2019), quatro modelos

diferentes baseados em Holt-Winters e a abordagem SARIMA foram aplicados para prever a geração de energia eólica do parque eólico Rio do Fogo, localizado no nordeste brasileiro. Os resultados mostram que a combinação ponderada pelo modelo SARIMA(1,0,0)(1,1,1)₁₂ e um modelo Holt-Winters multiplicativo fornece o melhor ajuste para as séries estudadas nesse trabalho.

Ademais, os modelos baseados em ARIMA, SARIMA e VARIMA também foram considerados para previsão de séries temporais nos últimos anos. O trabalho de Júnior, de Oliveira e Neto (2019) propõe um sistema híbrido baseado no modelo ARIMA, na Rede Neural *Perceptron* Multicamadas (*Multi-Layer Perceptron*, MLP) e no método Regressão por Vetores de Suporte (*Support Vector Regression*, SVR), visando uma função adequada para combinar as previsões de modelos lineares e não-lineares. Outra abordagem híbrida envolvendo os modelos ARIMA é apresentada em Buyuksahin e Ertekin (2019), aplicando uma combinação entre os modelos ARIMA e uma Rede Neural Artificial para prever componentes da série temporal por meio da técnica de Decomposição de Modo Empírico (*Empirical Mode Decomposition*, EMD). Além disso, algumas alternativas híbridas, unindo ao SARIMA, são propostas em aplicações como na área da indústria da aviação, combinando os métodos SVR e SARIMA (XU; CHAN; ZHANG, 2019), assim como na área da saúde, usando os modelos SARIMA e de Suavização Exponencial para prever a incidência de conjuntivite hemorrágica aguda na China (LIU et al., 2020b). Em relação a estudos que envolvem séries temporais multivariáveis, usa-se o VARIMA em abordagens híbridas. Por exemplo, o trabalho de Lusia e Ambarwati (2018) foca em expor as vantagens de utilizar o método VARIMA em combinação com uma rede neural para previsão multivariável dos preços de ações, e a pesquisa de Yang et al. (2018), que desenvolvem uma nova estrutura de previsão de série temporal *online* para o modelo VARMA .

À vista disso, percebe-se que, ao citar as referências acima, muitos trabalhos vêm propondo abordagens híbridas. Esses modelos híbridos combinam diferentes técnicas e exploram suas vantagens, gerando resultados relevantes em comparação com modelos simples. Outras pesquisas com propostas híbridas podem ser citadas para previsão de séries temporais. Uma análise de variadas estruturas híbridas de mais de 150 artigos empregadas na modelagem e previsão de séries temporais é apresentada na pesquisa de Hajirahimi e Khashei (2019). Em Jiang e Heng (2019), um novo sistema híbrido de previsão consiste em três módulos (um módulo de pré-processamento de dados, módulo de otimização e módulo de previsão) desenvolvidos para a previsão de séries temporais. No módulo de pré-processamento de dados, remove-se componentes ruidosas da série temporal, enquanto que o módulo de previsão é otimizado por um algoritmo evolucionário multi-objetivo que busca equilibrar a precisão da previsão e sua estabilidade. Outrossim, Graf, Zhu e Sivakumar (2019) elaboraram um novo modelo híbrido para prever a temperatura da água em rios, acoplando a Transformada Discreta de *Wavelet* e uma Rede Neural Artificial.

Como já mencionado em alguns trabalhos nessa seção, métodos baseados em redes

neurais e *deep learning* apresentam destaque também para previsão de séries temporais. Por exemplo, em Sagheer e Kotb (2019) é proposta uma abordagem de *deep learning* capaz de superar as limitações das abordagens tradicionais de previsão, exibindo resultados melhores em comparação com outros trabalhos. A abordagem proposta de previsão é denominada de *Deep Long-Short Term Memory* (DLSTM), que considera estudos de caso presentes na indústria de petróleo. Um algoritmo genético é considerado na ordem para otimizar a arquitetura ideal do DLSTM. Em Yang e Chen (2019), desenvolveu-se um novo modelo híbrido entre conceitos de *deep learning* e decomposição em modo empírico para previsão de séries temporais. Este artigo destaca o potencial de unir o *deep learning* e métodos de decomposição para a abordagem de previsão de séries temporais. O trabalho de Du et al. (2020) desenvolvem uma estrutura de *deep learning* que se baseia na arquitetura de uma *Long short-term memory* (LSTM). Por fim, em Shen et al. (2020), um novo modelo de previsão de série temporal baseado também em LSTM é proposto, denominado de *SeriesNet*.

Mais uma técnica que é relevante nesse tema de pesquisa e possui muitas aplicações é a Máquina de Vetores de Suporte (SVM, do inglês *Support Vector Machine*). Em Liu et al. (2020), um modelo de SVM híbrido, denominado de *Jaya-SVM*, é proposto para a previsão da velocidade do vento a um curto prazo. Nesse caso, os hiperparâmetros do SVM são otimizados usando o algoritmo de otimização Jaya. No trabalho de Hitam, Ismail e Saeed (2019), uma SVM otimizada baseada em *Particle Swarm Optimization* (PSO) é aplicada para previsão do preço futuro de criptomoedas. De forma semelhante, o artigo de Chen, Wang e Zhang (2019) também considera a SVM otimizada pelo PSO para previsão de poluentes atmosféricos.

Uma tendência importante também no estudo de séries temporais é a combinação de redes neurais e lógica *fuzzy*. No trabalho de Samanta et al. (2019), os autores apresentam um novo sistema de inferência *neuro-fuzzy* com neurônios dinâmicos para a identificação de sistemas em tempo discreto e problemas de previsão de séries temporais. Por sua vez, o estudo realizado em Zeinalnezhad et al. (2020) tem como objetivo obter a previsão diária de poluentes por meio da análise de dados de séries temporais usando o modelo *neuro-fuzzy*. Outra importante aplicação é demonstrada em Ly (2020), que utilizam um Sistema de Inferência *Neuro-Fuzzy* Adaptativo (*Adaptive Neuro-Fuzzy Inference System*, ANFIS) para prever o número de casos da COVID-19 no Reino Unido.

Para lidar com a complexidade da previsão de séries temporais reais, muitos artigos têm usado a abordagem "dividir para conquistar", isto é, dividir uma tarefa complexa em diversos subproblemas que são mais fáceis de lidar. Em Cao, Li e Li (2019), uma série temporal financeira é decomposta em sub-séries sob diferentes frequências pelo método EMD. A previsão para cada sub-série é realizada usando LSTM. Finalmente, todas as previsões são reconstruídas para obter o resultado final de previsão. Em Nguyen e Novák (2019), a ideia principal é decompor a série temporal em componentes sazonais, de tendências, cíclicas e irregulares, identificando a repetição de certos padrões nessas componentes. A previsão da componente sazonal é baseada

em sua estimação e na aplicação da lógica *fuzzy* para componentes sazonais. Para a previsão da componente irregular, os autores usaram a metodologia Box Jenkins. Finalmente, a série temporal é prevista por uma associação aditiva de todas as previsões individuais das componentes. Na pesquisa de Yang e Chen (2019), foi desenvolvido um novo modelo híbrido de previsão de série temporal múltiplos passos a frente com base em *deep learning* e no método de decomposição EMD, sendo que a proposta dos autores foi batizada de EMD-SAE.

2.6 Comentários Finais do Capítulo

A metodologia apresentada nessa tese, que será descrita na Seção 5, tem como principais objetivos fazer a previsão quantitativa de séries temporais e a caracterização destas, identificando suas componentes não-observáveis. Essa abordagem proposta pode lidar tanto com séries univariáveis ou multivariáveis, estacionárias ou não-estacionárias, assim como séries sazonais ou sem sazonalidade definida. No entanto, todos os experimentos e estudos de casos envolveram séries discretas e não-estacionárias nessa pesquisa.

A previsão é definida para múltiplos passos a frente, podendo ser tanto do tipo recorrente quanto direta, de acordo com a escolha do usuário ou aplicação. Ademais, as medidas de erro apresentadas nessa seção são aplicadas para avaliar os resultados de previsão, bem como o intervalo de previsão pode ser apresentado como uma faixa de possibilidades de resultados do método. O método desenvolvido nessa tese tem característica híbrida e tem sua estrutura baseada em uma rede *neuro-fuzzy*, sendo que ambas propriedades estão em destaque no cenário de previsão de séries temporais nos últimos anos, conforme a Seção 2.5 sobre o estado da arte. Mais uma característica da proposta é a decomposição da série temporal em componentes, também se mostrando como outra alternativa em evidência em artigos atuais.

3 ANÁLISE ESPECTRAL SINGULAR

3.1 Descrição Geral do Método de Análise Espectral Singular - SSA

A Análise Espectral Singular, do inglês *Singular Spectrum Analysis* (SSA), é um método que é aplicado na análise e decomposição de séries temporais. Essa abordagem pode ser usada em séries complexas com variadas características e em diversas áreas do conhecimento, como Matemática, Física, Economia, Meteorologia e Ciências Sociais (ABDOLLAHZADE et al., 2015).

Este método se caracteriza por ser em batelada e por englobar conceitos clássicos envolvendo as séries temporais, estatística, sistemas dinâmicos e processamentos de sinais (GOLYANDINA; NEKRUTKIN; ZHIGLJAVSKY, 2001). O método SSA básico visa decompor uma série temporal univariável em uma soma de componentes não-observáveis independentes. Essas componentes são consideradas como padrões ocultos contidos nos dados das séries temporais, cujo comportamento não é possível visualizar claramente em alguns casos quando se observa a série temporal completa.

A decomposição em componentes não-observáveis clássica consiste na identificação dos padrões de tendência, de sazonalidade, cíclicos e aleatórios. Essas componentes são predefinidas e interpretáveis, e a equação tradicional do modelo de componentes não-observáveis consiste de um modelo aditivo como segue (PELAGATTI, 2015):

$$y_k = t_k + \phi_k + s_k + e_k, \quad (3.1)$$

onde y_k é uma observação da série temporal no instante k , t_k é a tendência, ϕ_k é o padrão cíclico, s_k a sazonalidade e e_k representa o ruído branco. No entanto, muitas séries são complexas de tal forma que somente essas componentes não são suficientes para representar todos os padrões.

Os resultados de decomposição do método SSA são oriundos a partir dos dados da série temporal. Logo, uma das principais vantagens do método SSA é a sua característica não paramétrica, ou seja, não é necessário conhecer o modelo paramétrico da série temporal nessa abordagem. Além disso, não são requisitadas quaisquer suposições probabilísticas ou estatísticas dos dados da série temporal tais como as funções densidade de probabilidade predefinidas, a estacionariedade da série ou normalidade dos resíduos (GOLYANDINA; NEKRUTKIN; ZHIGLJAVSKY, 2001). Uma outra vantagem é a flexibilidade na escolha da quantidade de componentes não-observáveis, ou seja, o especialista pode escolher quantas componentes deseja extrair.

Algumas das principais aplicações do SSA são citadas a seguir:

1. Extrair componentes dos dados da série temporal;

2. Suavizar séries temporais e remover componentes ruidosas;
3. Encontrar modelos matemáticos apropriados para séries temporais;
4. Fazer previsão de séries temporais.

O método tradicional SSA é organizado em dois estágios, cada um com duas etapas organizadas conforme a seguir:

1. Estágio de Decomposição:
 - a) Etapa de Incorporação;
 - b) Etapa da Decomposição em Valores Singulares, do inglês *Singular Value Decomposition* (SVD);
2. Estágio de Reconstrução:
 - a) Etapa de Agrupamento de Índices;
 - b) Etapa da Média Diagonal.

A seguir, o procedimento tradicional do método SSA é descrito conforme essa organização acima.

3.2 Estágio de Decomposição

3.2.1 Etapa de Incorporação

O objetivo dessa etapa é representar uma série temporal unidimensional de forma multidimensional ou matricial, cuja dimensão depende do parâmetro denominado tamanho de janela L e do parâmetro \mathcal{K} . Seja também uma série temporal univariável não-nula y_k com N observações ($k = 1, \dots, N$), para $N > 1$. O valor de L deve ser determinado pelo especialista, contudo, seu valor deve estar compreendido no intervalo $2 \leq L \leq N/2$. Sabendo que $\mathcal{K} = N - L + 1$, a representação multidimensional da série, denominada de Matriz Trajetória \mathbf{M} , é descrita conforme a seguir (GOLYANDINA; NEKRUTKIN; ZHIGLJAVSKY, 2001):

$$\mathbf{M} = \begin{bmatrix} y_1 & y_2 & \cdots & y_{\mathcal{K}} \\ y_2 & y_3 & \cdots & y_{\mathcal{K}+1} \\ y_3 & y_4 & \cdots & y_{\mathcal{K}+2} \\ \vdots & \vdots & \vdots & \vdots \\ y_L & y_{L+1} & \cdots & y_N \end{bmatrix}_{L \times \mathcal{K}} \quad (3.2)$$

tal que $\mathbf{M} \in \mathbb{R}^{L \times \mathcal{K}}$. A Matriz Trajetória pode ser caracterizada como uma matriz Hankel¹, já que os elementos ao longo das diagonais que ascendem da esquerda para a direita são iguais.

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

3.2.2 Etapa de Decomposição em Valores Singulares - SVD

O próximo passo é realizar a Decomposição em Valores Singulares (do inglês, *Singular Value Decomposition*, SVD) da Matriz Trajetória \mathbf{M} previamente definida, ou seja (GOLYANDINA; NEKRUTKIN; ZHIGLJAVSKY, 2001):

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \quad (3.3)$$

onde:

- \mathbf{U} é uma matriz $L \times L$ que contém os vetores singulares à esquerda de \mathbf{M} ;
- $\mathbf{\Sigma}$ é uma matriz diagonal $L \times \mathcal{K}$, cujos elementos diagonais são os valores singulares σ_i , $i = 1, \dots, L$.
- \mathbf{V} é uma matriz $\mathcal{K} \times \mathcal{K}$ que contém os vetores singulares à direita de \mathbf{M} .

Seja a Matriz Covariância $\mathbf{S} = \mathbf{M}\mathbf{M}^T$, para $\mathbf{S} \in \mathbb{R}^{L \times L}$. Define-se ainda $\rho_1, \rho_2, \dots, \rho_L$ como os autovalores de \mathbf{S} em ordem decrescente de magnitude ($\rho_1 > \rho_2 > \dots > \rho_L$). Como citam Golyandina, Nekrutkin e Zhigljavsky (2001), os valores singulares σ_i nesse caso possuem uma relação com os autovalores denotada por:

$$\sigma_i = \sqrt{\rho_i}, \quad i = 1, \dots, L. \quad (3.4)$$

Em outra perspectiva, denota-se que $\mathbf{u}_1, \dots, \mathbf{u}_L \in \mathbb{R}^{L \times 1}$ contidos em \mathbf{U} (colunas da matriz \mathbf{U}) correspondem a um sistema ortonormal de autovetores de \mathbf{S} . Apresenta-se também o parâmetro $d = \max \{i, \text{tal que } \rho_i > 0\}$, que indica a quantidade de autovalores não-nulos. Logo, é possível reescrever a Matriz Trajetória como uma soma de Matrizes Elementares conforme a seguir (GOLYANDINA; NEKRUTKIN; ZHIGLJAVSKY, 2001):

$$\mathbf{M} = \mathbf{M}_1 + \mathbf{M}_2 + \dots + \mathbf{M}_d, \quad (3.5)$$

sendo que, para $i = 1, \dots, d$, tem-se:

$$\mathbf{v}_i = \frac{\mathbf{M}^T \mathbf{u}_i}{\sqrt{\rho_i}} \quad (3.6)$$

$$\mathbf{M}_i = \sqrt{\rho_i} \mathbf{u}_i \mathbf{v}_i^T,$$

onde \mathbf{v}_i são os vetores singulares à direita contidos em \mathbf{V} (colunas da matriz \mathbf{V}). A Matriz Elementar i é descrita por \mathbf{M}_i , cujo posto² é igual a 1. Destaca-se ainda que é chamado de i -ésima autotripla o conjunto composto por $(\sqrt{\rho_i}, \mathbf{u}_i, \mathbf{v}_i)$.

² 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).

O comportamento dos valores singulares σ_i pode ser observado graficamente em forma decrescente de magnitude, exibindo o chamado espectro de \mathbf{S} . O primeiro valor singular, isto é, aquele com maior valor absoluto está associado a um padrão de tendência, portanto, com variação mais lenta e estável no decorrer do tempo em relação as outras componentes. Os outros geralmente estão associados a diferentes periodicidades da série temporal (GOLYANDINA; NEKRUTKIN; ZHIGLJAVSKY, 2001).

3.3 Estágio de Reconstrução

3.3.1 Agrupamento de Índices

Após a expansão realizada na Equação 3.5, a etapa de agrupamento é iniciada, cuja função é particionar o conjunto de índices $\{1, \dots, d\}$ 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).

Considera-se um subconjunto genérico I contendo uma certa quantidade de índices, isto é, $I = \{i_p, \dots, i_q\}$. Sendo assim, a Matriz Agrupada \mathbf{M}_I associada a este subconjunto genérico é uma soma de Matrizes Elementares \mathbf{M}_i ($i = 1, \dots, d$) definida por:

$$\mathbf{M}_I = \mathbf{M}_{i_p} + \dots + \mathbf{M}_{i_q}. \quad (3.7)$$

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

$$\mathbf{M} = \mathbf{M}_{I_1} + \dots + \mathbf{M}_{I_n}. \quad (3.8)$$

Para esclarecer o procedimento dessa etapa, considere $d = 4$ e $n = 2$. Dessa forma, a partir da Equação 3.5 tem-se:

$$\mathbf{M} = \mathbf{M}_1 + \mathbf{M}_2 + \mathbf{M}_3 + \mathbf{M}_4. \quad (3.9)$$

Organizando as Matrizes Elementares em $n = 2$ subconjuntos disjuntos (I_1 e I_2), pode-se representar o resultado da etapa de agrupamento conforme a seguir:

$$\begin{aligned} \mathbf{M}_{I_1} &= \mathbf{M}_1 + \mathbf{M}_2, \\ \mathbf{M}_{I_2} &= \mathbf{M}_3 + \mathbf{M}_4, \\ \mathbf{M} &= \mathbf{M}_{I_1} + \mathbf{M}_{I_2}. \end{aligned} \quad (3.10)$$

3.3.2 Etapa da Média Diagonal

Por fim, cada matriz resultante do agrupamento realizado na Equação 3.8 é transformada em uma componente não-observável c_k^j extraída da série temporal original, para $j = 1, \dots, n$. Considerando que $m_{l,q}^j$ é o elemento da linha l e coluna q da Matriz Agrupada \mathbf{M}_{I_j} , tem-se

a média diagonal de \mathbf{M}_I^j , realizada de acordo com as formulações abaixo (GOLYANDINA; NEKRUTKIN; ZHIGLJAVSKY, 2001):

$$c_k^j = \begin{cases} \frac{\sum_{l=1}^k m_{l,k-l+1}^j}{k}, & \text{se } 1 \leq k < L; \\ \frac{\sum_{l=1}^L m_{l,k-l+1}^j}{L}, & \text{se } L \leq k \leq \mathcal{K}; \\ \frac{\sum_{l=k-\mathcal{K}+1}^{N-\mathcal{K}+1} m_{l,k-l+1}^j}{N-\mathcal{K}+1}, & \text{se } \mathcal{K} < k \leq N. \end{cases} \quad (3.11)$$

O resultado dessa operação nos fornece n componentes não-observáveis, cuja soma destas a cada instante k é exatamente igual a própria série temporal, ou seja:

$$y_k = c_k^1 + c_k^2 + \dots + c_k^n. \quad (3.12)$$

O algoritmo que resume o procedimento do SSA básico apresentado até aqui está descrito no Apêndice A.1.

3.4 Métodos para Agrupamento de Índices das Matrizes Elementares

Um aspecto importante na etapa de agrupamento do SSA (Seção 3.3.1) é qual critério será utilizado para separar as Matrizes Elementares em n subconjuntos disjuntos (Equações 3.8 e 3.9), isso porque resultados distintos de decomposição podem ocorrer dependendo da abordagem utilizada. Para algumas soluções, é mais conveniente primeiro fazer a média diagonal das matrizes resultantes da etapa do SVD, obtendo as componentes da série temporal e, em seguida, realizar a etapa de agrupamento. Sendo assim, é possível inverter a ordem dos procedimentos no estágio de reconstrução do método SSA (CASSIANO, 2014).

Uma primeira abordagem pode ser a análise dos valores singulares, através do gráfico do espectro da Matriz Covariância \mathbf{S} . Sendo assim, pode-se considerar somente os valores singulares mais significativos ou de maior magnitude entre os L elementos obtidos, removendo aqueles com menor magnitude que geralmente estão associados a componentes ruidosas. Outra forma de realizar a etapa de agrupamento do SSA é por meio de técnicas de *clustering*, que visa organizar as componentes não-observáveis em grupos disjuntos de acordo com as similaridades e dissimilaridades em relação aos objetos pertencentes aos outros grupos. Logo, métodos de análise de *clusters* podem ser utilizados na classificação dos vetores singulares na SVD. A análise de periodograma também pode auxiliar na Etapa de Agrupamento. O periodograma do vetor singular de cada autotripla fornece informações sobre o comportamento periódico das

componentes e frequência (período) das oscilações. Portanto, o agrupamento adequado pode ser feito com a ajuda de análise do periodograma (CASSIANO, 2014). Para as componentes periódicas, o periodograma tem picos em torno da frequência da componente não-observáveis, portanto, a identificação visual é simples, facilitando a organização em Matrizes Agrupadas.

3.5 Separabilidade

As etapas de SVD e Agrupamento podem considerar a propriedade denominada separabilidade, que caracteriza o quão bem as diferentes componentes estão distintas uma das outras em termos de dependência linear (ELSNER; TSONIS, 1996).

Segundo Golyandina, Nekrutkin e Zhigljavsky (2001), na prática não ocorre separabilidade perfeita, somente sendo possível alcançar uma separabilidade aproximada. Para mensurar essa separabilidade aproximada pode-se usar uma medida de correlação ponderada a fim de quantificar a dependência linear entre duas componentes não-observáveis.

Sejam c_k^i e c_k^j duas componentes não-observáveis, a correlação ponderada representada por $\omega_{i,j}$ é calculada como segue:

$$\omega_{i,j} = \frac{(c_k^i, c_k^j)_w}{\|c_k^i\|_w \|c_k^j\|_w}, \quad (3.13)$$

onde tem-se:

$$\begin{aligned} \|c_k^i\|_w &= \sqrt{(c_k^i, c_k^i)_w} \rightarrow \text{norma euclidiana da } i\text{-ésima componente,} \\ (c_k^i, c_k^j)_w &= \sum_{k=1}^N w_k c_k^i c_k^j \rightarrow \text{produto interno entre o par de componentes,} \end{aligned} \quad (3.14)$$

$$w_k = \min(k, L, N - k) \rightarrow \text{ponderador.}$$

Caso o valor absoluto da correlação ponderada seja pequeno, então as componentes não-observáveis são classificadas como w-ortogonais e apresentam separabilidade aproximada. Caso contrário, são ditas mal separadas. Em outras palavras, as componentes altamente correlacionadas se tornam pertencentes a um mesmo grupo de autotriplas ou a mesma Matriz Agrupada (GOLYANDINA; NEKRUTKIN; ZHIGLJAVSKY, 2001).

Outra ferramenta que pode ser usada para estudar a separabilidade das componentes é matriz w-correlação. A matriz w-correlação pode indicar as correlações graficamente através de uma escala de tons de cinza, variando do branco ao preto, correspondendo aos valores absolutos de 0 a 1. A forma da matriz correlação fornece um indicativo de como realizar um agrupamento adequado (GOLYANDINA; NEKRUTKIN; ZHIGLJAVSKY, 2001). Um exemplo de análise usando a matriz w-correlação será apresentada nas próximas seções.

3.6 Seleção do Parâmetro L

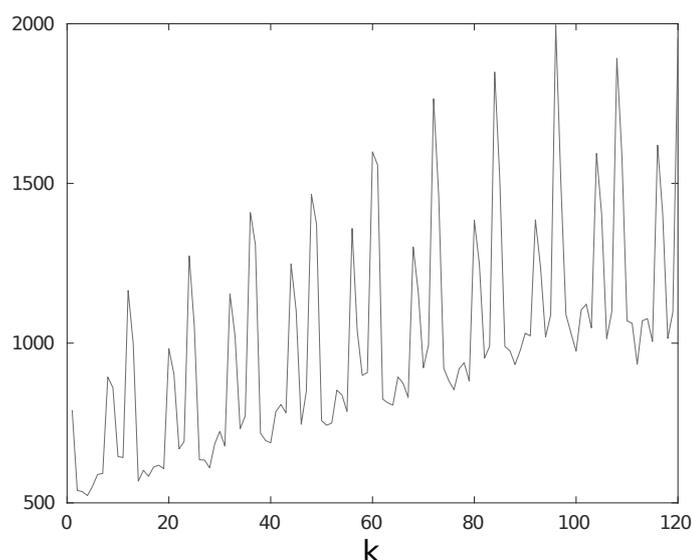
A determinação de um valor para o parâmetro L sempre foi uma das principais discussões em relação ao SSA, pois tem influência nos resultados de extração das componentes e no nível de separabilidade entre elas. Na maioria dos casos, valores entre 2 e $N/2$, para N a quantidade total de amostras, são suficientes para realização de uma decomposição da série temporal de forma apropriada. À medida que aumenta-se L , a decomposição da série temporal torna-se mais detalhada (GOLYANDINA; NEKRUTKIN; ZHIGLJAVSKY, 2001).

Em resumo, a maioria dos trabalhos fazem recomendações de intervalos de confiança para L , entretanto, a escolha deste parâmetro depende da natureza dos dados da série temporal. Uma forma de verificar um bom valor para L é implementar alguma abordagem de otimização tal que maximize a separabilidade entre as componentes não-observáveis. Outra alternativa, a condição $L_l = n$, para n a quantidade de componentes, pode ser aplicada, de acordo com Groth et al. (2017). Isso significa que cada valor singular está relacionado a uma componente não-observável específica, sem se preocupar com a separabilidade ideal.

3.7 Estudo Gráfico da Análise Espectral Singular

Essa seção apresenta algumas análises importantes em relação ao método SSA. Para esse estudo, é considerada uma série temporal sobre a Venda de **livros** abordada no trabalho de Gan et al. (2014), com $N = 120$ amostras no total e período sazonal definido por $s = 12$. Esta série pode ser visualizada na Figura 7.

Figura 7 – Série Temporal Venda de Livros para Estudo da Análise Espectral Singular.



Fonte: (GAN et al., 2014).

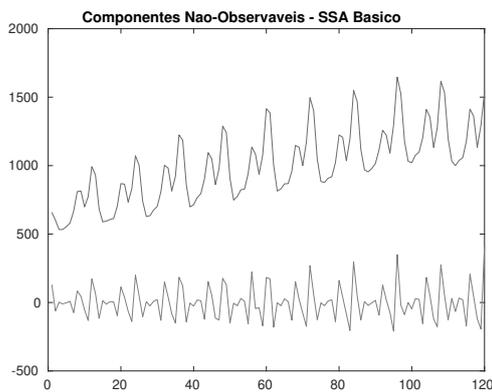
A partir dessa série temporal, realizou-se a extração de suas componentes não-observáveis,

verificando os efeitos da alteração do parâmetro L . O valor de n , ou seja, a quantidade de componentes não-observáveis, foi considerado igual a L , para simplificar a etapa de Agrupamento de Índices do SSA. A Figura 8 mostra o efeito das modificações de L em quatro situações:

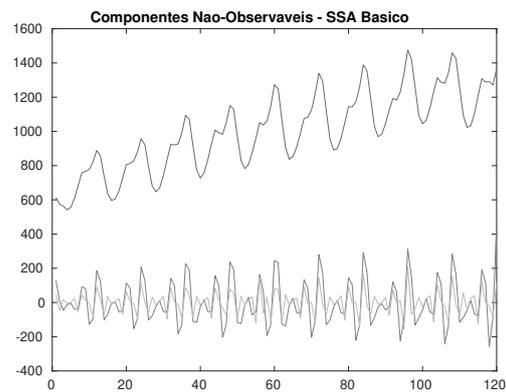
- $L = 2$ e $n = 2$;
- $L = 3$ e $n = 3$;
- $L = 6$ e $n = 6$;
- $L = 10$ e $n = 10$.

Note que conforme aumenta-se o parâmetro L , a decomposição da série temporal torna-se mais detalhada e, além disso, a componente não-observável mais destacada torna-se mais suavizada, pois parte de seu comportamento vai sendo representado por outras componentes não-observáveis.

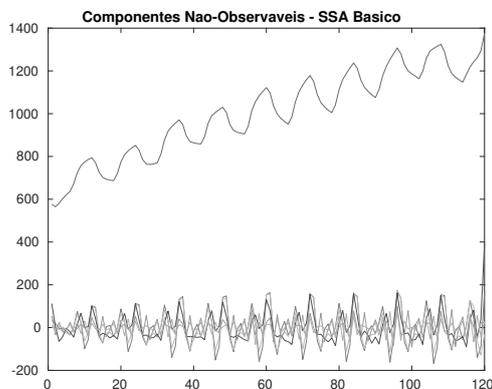
Figura 8 – Exemplo de Componentes Não-Observáveis e Efeito da Variação do Parâmetro L para a Série Temporal de Vendas de **Livros**.



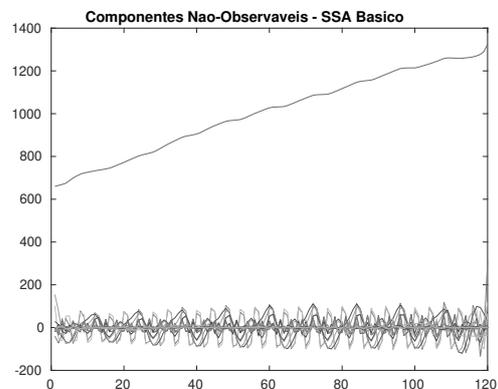
(a) Decomposição para $L = 2$ e $n = 2$.



(b) Decomposição para $L = 3$ e $n = 3$.



(c) Decomposição para $L = 6$ e $n = 6$.

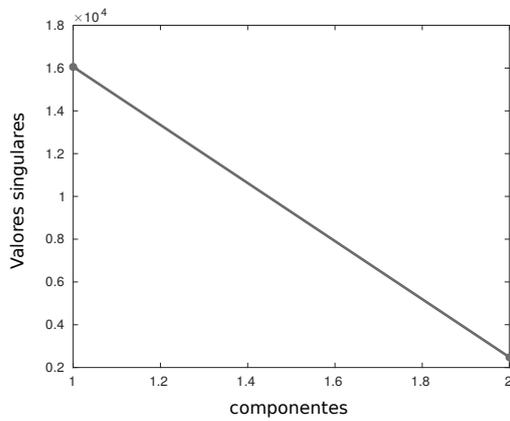


(d) Decomposição para $L = 20$ e $n = 20$.

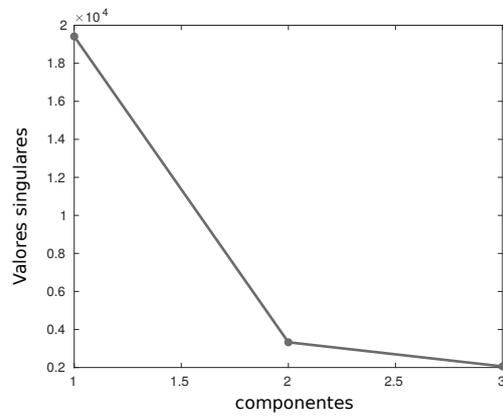
Fonte: Autor.

Os valores singulares para os quatro casos de decomposições realizadas na Figura 8 são apresentados na Figura 9. Esse gráfico pode ser denominado de espectro da Matriz Covariância, demonstrando o comportamento dos valores singulares em ordem decrescente de magnitude. O valor singular mais significativo está associado a componente de maior destaque e mais suavizada na Figura 8, enquanto os outros valores estão associados a outras componentes periódicas. Percebe-se que conforme aumenta-se o valor de L e n , maior a magnitude desse valor singular, portanto, mais suavizada fica a componente relacionada. Valores singulares com magnitude próximas possuem comportamentos semelhantes e, em uma análise mais profunda, podem ser associados a uma mesma componente não-observável, simplificando a decomposição da série temporal e reduzindo o número de componentes.

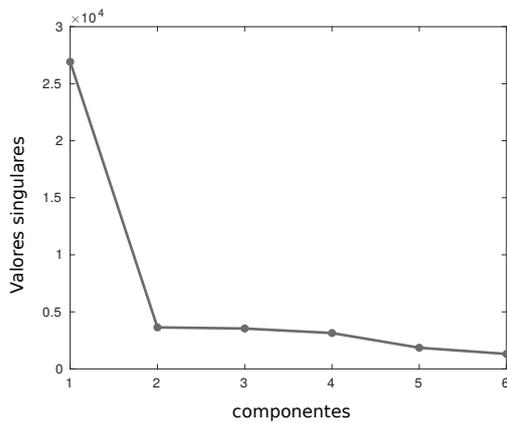
Figura 9 – Valores Singulares para as Decomposições da Figura 8.



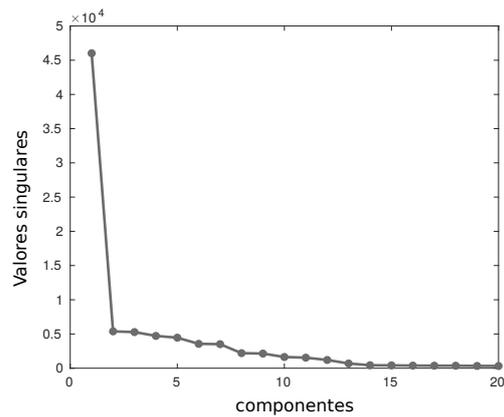
(a) Valores Singulares para $L = 2$ e $n = 2$.



(b) Decomposição para $L = 3$ e $n = 3$.



(c) Decomposição para $L = 6$ e $n = 6$.



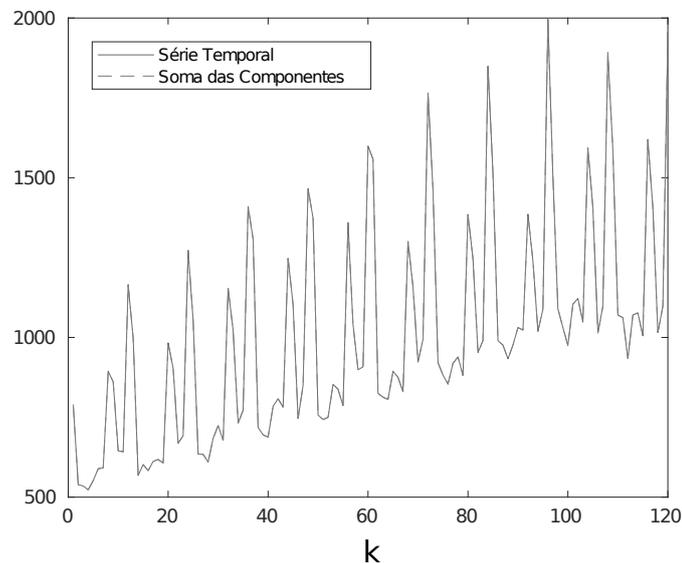
(d) Decomposição para $L = 20$ e $n = 20$.

Fonte: Autor.

Na Figura 10, pode-se observar a comparação da série original com a soma das componentes não-observáveis. A soma das componentes é exatamente igual a série temporal, conforme Equação 3.12, para quaisquer valores dos parâmetros L e n . Alguns métodos podem ignorar uma quantidade dessas componentes, associadas aos menores valores singulares, ao considerá-las como ruído (GOLYANDINA; NEKRUTKIN; ZHIGLJAVSKY, 2001). Entretanto, na metodo-

logia proposta, todas as componentes serão consideradas na estrutura, pois a remoção de um número de componentes ruidosas elevou os erros de previsão apresentados.

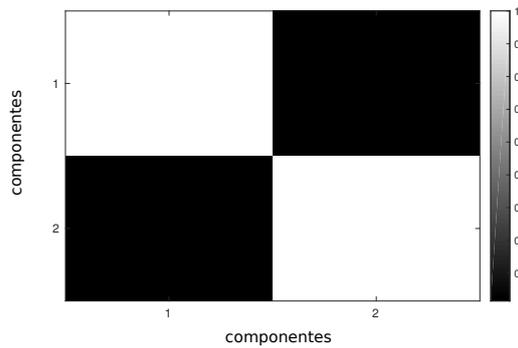
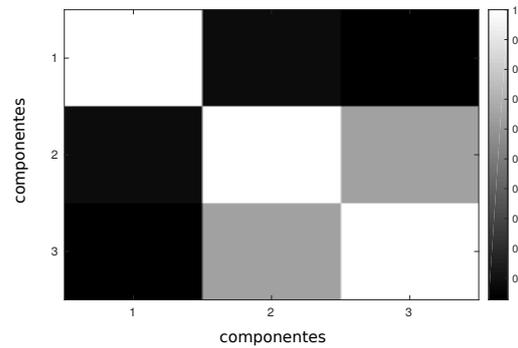
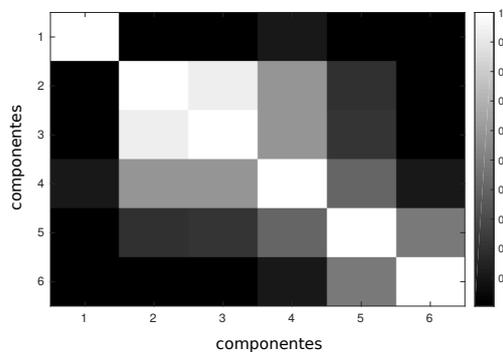
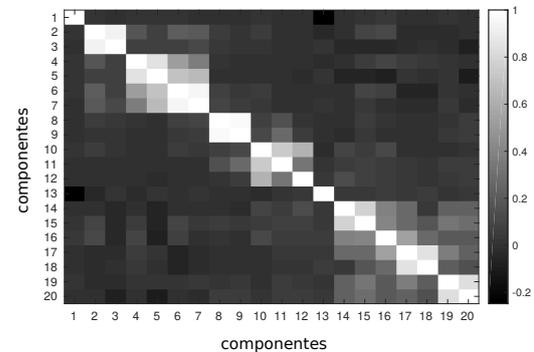
Figura 10 – Série Temporal de Vendas de **Livros** em Comparação com a Soma das Componentes Não-Observáveis.



Fonte: Autor.

Por fim, na Figura 11 tem-se a análise da separabilidade entre as componentes não-observáveis nos 4 casos de decomposição apresentados na Figura 8. A separabilidade pode ser verificada pela matriz de w -correlação em escalas de cinza, onde a cor branca indica a correlação máxima igual a 1 entre a mesma componente não-observável. Quanto mais escura a cor, menor a correlação, ou seja, mais separáveis são as componentes. O ideal é que a matriz seja branca na diagonal principal e totalmente preta nos outros quadros, porém, não existe separabilidade perfeita. Nos 4 gráficos percebe-se essa característica. A partir dessa matriz, é possível analisar se é possível associar algumas componentes com elevada correlação. Por exemplo, na Figura 11.(c), as componentes de número 2 e 3 apresentam elevadas correlações, portanto, elas podem ser agrupadas em apenas uma componente ou um tipo de padrão.

Figura 11 – Matriz W-Correlação para as Decomposições da Figura 8.

(a) Matriz W-Correlação para $L = 2$ e $n = 2$.(b) Matriz W-Correlação para $L = 3$ e $n = 3$.(c) Matriz W-Correlação para $L = 6$ e $n = 6$.(d) Matriz W-Correlação para $L = 20$ e $n = 20$.

Fonte: Autor.

3.8 Estado da Arte da Análise Espectral Singular

O método SSA começou a ser difundido a partir do trabalho de Elsner e Tsonis (1996). Desde então, começaram a surgir aplicações em diversas áreas como na Meteorologia (ALLEN; SMITH, 1996), finanças (HASSANI; THOMAKOS, 2010) e na área farmacêutica (BAPTISTA, 2008).

Em Lin et al. (2020), uma aplicação do SSA é utilizada na análise da doença de Parkinson, sendo que o papel desse método é extrair e remover componentes correspondentes aos movimentos voluntários e os ruídos de várias fontes. Outra aplicação na remoção de ruídos está em Lu et al. (2020), caso em que a função do SSA é remover o ruído durante o pré-processamento da frequência cardíaca fetal. Em Mukhopadhyay e Krishnan (2020), também tem-se uma aplicação em sinais de eletrocardiograma para remoção de ruídos.

Por sua vez, o objetivo do trabalho de Kalantari (2021) é examinar as vantagens potenciais do SSA para a previsão do número de casos confirmados diariamente, mortes e recuperações causadas por COVID-19. Nesse caso, um algoritmo é proposto para calcular os parâmetros ótimos do SSA, incluindo o parâmetro L e o número de componentes n , sendo que os resultados indicam que o SSA é uma opção viável para melhorar os resultados de previsão nessa aplicação. No estudo

de Krishnannair e Aldrich (2019), um novo método de monitoramento estatístico multivariável baseado no SSA é proposto para detecção de falhas em sistemas de processos químicos. O SSA é aplicado também em Sulandari et al. (2020), sendo que este estudo apresenta vários métodos para prever a demanda do uso de eletricidade na Indonésia e compara o desempenho dos métodos. O SSA é selecionado devido à sua capacidade de decompor a série em componentes separáveis, modelando essas componentes e ruídos por sistemas *fuzzy* e redes neurais.

Recentemente, vêm sendo desenvolvidos outros métodos com inspiração no SSA tradicional. Em Leles et al. (2018), a implementação de um novo algoritmo denominado *Overlap-SSA* é detalhada. Esse algoritmo pode ser usado em muitas áreas do conhecimento em que o SSA padrão já está sendo usado. Contribuições desse novo algoritmo são a melhoria na reconstrução de séries temporais longas; minimização dos efeitos de contorno associados à análise segmentada; e melhor caracterização tempo-frequência de sinais não-estacionários. Em Rodrigues, Lourenço e Mahmoudvand (2018), uma abordagem SSA robusta (*robSSA*), em que um algoritmo SVD robusto substitui o SVD padrão na segunda decomposição é proposto quando a série temporal é contaminada com observações externas. O SSA tradicional e o *robSSA* são comparados em termos da qualidade do ajuste do modelo através de simulações de Monte Carlo. Em Rodrigues, Tuy e Mahmoudvand (2018), um SSA alternativo adequado para séries temporais longas é proposto. A abordagem proposta é comparada com o SSA tradicional em termos de qualidade do ajuste, previsão do modelo e tempo computacional. Isso é feito usando simulações de Monte Carlo e dados reais sobre os preços diários de cinco das principais *commodities* mundiais. Os autores em Bógalo, Poncela e Senra (2021) propõem uma nova variante do SSA denominada de *Circulant SSA* (*CiSSA*), que permite extrair o sinal associado a qualquer frequência previamente especificada.

O SSA vem sendo aplicado juntamente com outras técnicas, fazendo parte de novas metodologias híbridas com distintas finalidades. O estudo realizado em Zubaidi et al. (2018) fornece uma nova abordagem para quantificar a demanda de água através da avaliação de fatores climáticos, usando uma combinação do método SSA, o algoritmo de otimização PSO e uma rede neural artificial. Nesse sentido, o método SSA foi adotado para decompor e reconstruir o consumo de água em relação a seis variáveis climáticas criando uma série temporal sazonal. Em Kolidakis et al. (2019), o artigo apresenta uma metodologia híbrida que combina o SSA com Redes Neurais Artificiais, aplicada na análise de séries temporais e previsão do volume de tráfego rodoviário. Em Liu et al. (2019), um novo modelo híbrido de previsão de velocidade do vento a curto prazo é proposto com base no método SSA, CNN (*Convolutional Neural Network*), método GRU (*Gated Recurrent Unit*) e método SVR (*Support Vector Regression*). Nesse caso, o SSA é usado para decompor a série temporal original de velocidade do vento em várias componentes. No trabalho de Mi e Zhao (2020), um modelo baseado em LSTM é proposto para capturar as características dinâmicas de dados relacionados à velocidade do vento. O método SSA trabalha de forma conjunta com a LSTM para extração de características dos dados.

O método SSA tradicional lida apenas com uma série temporal, contudo muitas aplicações práticas envolvem a análise de múltiplas séries temporais. No artigo de Mao e Shang (2019), esse método de Análise Espectral Singular Multivariável (MSSA) é utilizado para decompor séries temporais de tráfego em padrões complexos e tendência não-estacionária. Hassani et al. (2019) apresentam um método que usa uma combinação de séries mensais e trimestrais e aplicam a decomposição e reconstrução MSSA para obter estimativas e previsões mensais para as séries trimestrais. Por fim, o MSSA é aplicado em Yan et al. (2021) para obter as componentes e identificar padrões nos dados de vibração coletados do sistema de acionamento de turbina eólica.

3.9 Comentários Finais do Capítulo

O método SSA básico apresentado nesse capítulo tem sua execução em batelada e necessita de toda a informação histórica da série temporal disponível para realizar a extração das componentes. Contudo, a abordagem proposta dessa tese é aplicada em problemas em que os dados da série chegam a cada instante. Logo, a estrutura proposta, baseada em uma rede *neuro-fuzzy* evolutiva, aplica uma aprendizagem incremental a cada instante, tópicos que serão comentados no próximo capítulo.

Sendo assim, para se adaptar a forma desse treinamento incremental, essa tese propõe modificações no método SSA para executar de forma recursiva, auxiliando a rede *neuro-fuzzy* na aprendizagem incremental considerando uma nova observação da série a cada instante. Portanto, a proposta da tese aplica de forma híbrida o SSA e a rede *neuro-fuzzy* evolutiva.

Analisando o estado da arte, a proposta da tese considera essa característica híbrida, que muitos trabalhos atuais empregam. Além disso, como foi discutido na Seção 2.5, a decomposição pode melhorar os resultados em problemas de previsão de séries temporais, como se observa em algumas das aplicações apresentadas.

4 MODELAGEM EVOLUTIVA

4.1 Aspectos Gerais sobre Modelagem Matemática

De acordo com Aguirre (2015), a modelagem matemática é a área do conhecimento que estuda meios para desenvolver e implementar modelos matemáticos de sistemas reais. Logo, um modelo matemático representa algumas das características do sistema, mas não todas, pois pode-se elevar bastante a complexidade em seu desenvolvimento (AGUIRRE, 2015). É fato que, ao longo da história, a modelagem matemática foi de grande importância para muitas aplicações e problemas envolvendo as diversas áreas do conhecimento (SCHICHL, 2004).

Há várias classificações para os modelos matemáticos na literatura segundo diferentes critérios. Lughofer (2011) destaca quatro classes de modelos:

- Modelos conceituais;
- Modelos baseados em conhecimento;
- Modelos baseados em dados;
- Modelos híbridos.

Na modelagem conceitual, também chamada de modelagem caixa branca (AGUIRRE, 2015), são desenvolvidos modelos que são deduzidos analiticamente a partir de uma teoria estabelecida de leis físicas e químicas bem conhecidas ou comportamentos de processos biológicos, de acordo com a natureza do sistema a ser modelado. Quando os comportamentos e fenômenos do sistema em questão são totalmente compreendidos, esses modelos são representações bastante confiáveis e precisas. Porém, de acordo com Aguirre (2015), em algumas situações pode ser inviável essa classe de modelos, devido ao alto custo de tempo e computacional, ou mesmo a grande complexidade em desenvolvê-los. Nesses casos, os modelos baseados em conhecimento ou baseados em dados são alternativas.

Segundo Lughofer (2011), os modelos baseados em conhecimento são extraídos a partir das experiências de especialistas, operadores e usuários que trabalham no sistema real. Essas experiências podem ser recolhidas e formuladas principalmente na forma de regras linguísticas, podendo exibir dependências muito complexas que não seriam capturadas pelos modelos conceituais. O conjunto de regras corresponde ao núcleo dos chamados sistemas baseados em conhecimento ou sistemas especialistas (CASTILLO; ALVAREZ, 1991). Como exemplo, os sistemas *fuzzy* (WANG, 1997) são um caso específico de tais sistemas especialistas baseados em regras e têm a vantagem de lidar com as incertezas inerentes do problema de maneira aproximada, levando em conta as possibilidades que não foram explicitamente formuladas pelos especialistas ou usuários.

Todavia, os especialistas, operadores ou usuários conseguem expressar somente seu conhecimento consciente, enquanto o conhecimento subconsciente não é relatado facilmente por causa da sua subjetividade (WANG, 1997). À vista disso, os modelos baseados em dados são alternativas relevantes. Lughofer (2011) apresenta a definição de modelos baseados em dados como modelos matemáticos que são totalmente projetados, extraídos ou aprendidos a partir dos dados do sistema. Esses dados podem ser oriundos a partir de canais de medição dos sistemas, dados baseados em contexto (áudio, imagens, vídeos, etc.), dados estruturados ou mesmo *data streams*, que são fluxos de dados caracterizados por um comportamento dinâmico, geralmente em processamento *online*. Esses *data streams* podem se apresentar em diversos formatos, como dados de medição, dados baseados em contexto e dados estruturados, os tipos mencionados anteriormente. Segundo Skrjanc et al. (2019), os *data streams* são comumente encontrados no comércio e negócios, análise financeira, casas inteligentes, assistência médica, sistemas de transporte, cadeias de logística de suprimento, redes inteligentes, controle industrial, segurança cibernética e muitas outras áreas. A modelagem baseada em dados também é conhecida como modelagem ou identificação caixa preta (AGUIRRE, 2015).

O processo de aprendizagem dos modelos orientados a dados é efetuado a partir de um conjunto de dados de treinamento, portanto, a capacidade de generalização do modelo depende da seleção desses dados de treinamento. Outros aspectos significativos nesse tipo de modelagem são a escolha da representação matemática utilizada e a estimação de seus parâmetros, assim como o processo de validação dos modelos, que é imprescindível em qualquer procedimento de modelagem (AGUIRRE, 2015).

Por fim, há ainda a possibilidade de desenvolver modelos híbridos, que também são conhecidos como modelos caixa cinza. A modelagem caixa cinza visa combinar as vantagens dos procedimentos de identificação caixa preta e caixa branca. Nesse caso, tanto os dados do sistema quanto informações auxiliares como leis físicas ou conhecimento consciente dos especialistas são usados no processo de modelagem (CORRÊA; AGUIRRE, 2004). Um exemplo de modelagem híbrida ocorre quando projeta-se um modelo baseado em sistemas *fuzzy*, que possui a flexibilidade de ser elaborado a partir dos dados de entrada e saída do sistema real, sendo possível combinar a abordagem baseada em dados com a abordagem baseada em conhecimento (conhecimento consciente do especialista) (WANG, 1997).

4.2 Modelos Evolutivos

O paradigma evolutivo engloba modelos que possuem a capacidade de expandir ou simplificar sua estrutura para se adaptar às alterações do ambiente (ANGELOV, 2002), (KASABOV, 2002), (ANGELOV; KASABOV, 2005). Sendo assim, os modelos evolutivos não são apenas adaptativos, que mantêm sua representação matemática e cujos parâmetros vão se ajustando aos dados, como também evoluem suas estruturas conforme o comportamento destes (ANGELOV, 2013). Lughofer (2011) os define como modelos baseados em dados que se adaptam automatica-

mente, evoluindo e ajustando dinamicamente sua estrutura em operação baseando-se nos dados que chegam a todo instante.

Há casos em que somente a adaptação é suficiente para que o modelo incorpore novos dados, mantendo sua estrutura matemática fixa e ajustando apenas seus parâmetros. A teoria dos modelos adaptativos já é muito bem estabelecida e possui vastas aplicações. No entanto, existem outras situações que apenas essa característica adaptativa não é suficiente, exigindo-se que sua estrutura seja reajustada para representar bem os novos dados do ambiente (ANGELOV, 2002). Como alternativa, os modelos evolutivos são capazes de modificar ou evoluir sua estrutura além de seus parâmetros, ajustando-se a fim de representar bem o conjunto de dados de treinamento. De acordo com Angelov (2013), todo sistema evolutivo é adaptativo, porém, nem todo sistema adaptativo é evolutivo.

Uma das principais motivações para o estudo de modelos evolutivos é a necessidade de lidar com a grande quantidade de dados que são gerados diariamente em vários formatos, produzidos nas redes de computadores, por consumidores, por sensores industriais, por índice econômicos e por diversas áreas científicas. Quando novos dados são recebidos, é dispendioso fazer o treinamento novamente dos modelos matemáticos considerando todo o conjunto de dados nessas situações. Dessa maneira, busca-se extrair conhecimentos úteis em tempo-real a partir dos *data streams* a todo instante, buscando atender os rigorosos requisitos de tempo exigidos em alguns casos e não precisando fazer novamente o treinamento para todo o histórico de dados (ANGELOV, 2013). Angelov, Gu e Kangin (2017) citam outras motivações para os modelos evolutivos:

- ter um grau de autonomia elevado para evoluir sua estrutura com pouca contribuição humana;
- considerar o comportamento complexo dos dados presentes na maioria dos problemas reais;
- aprender de forma contínua conforme novas observações dos dados chegam;
- preservar a transparência e interpretabilidade do modelo;
- ser computacionalmente eficiente;
- ter desempenho superior ou comparável a outros métodos de classificação, previsão, etc.

Assim sendo, Lughofer (2011) afirma que as aplicações de modelos evolutivos têm forte relação com o conceito de aprendizagem incremental. De acordo com esse autor, a aprendizagem incremental é definida como o processo de treinamento considerando amostra por amostra ou por blocos de dados, sendo que o modelo a ser treinado pode ser iniciado do zero ou mesmo começando com um já existente. Dessa forma, o modelo é construído passo a passo com cada

parte dos dados, em vez utilizar todo o conjunto de observações completo e refazer o treinamento a cada nova informação, isto é, aplicar o procedimento em batelada. Em alguns casos, assim que novas amostras de entrada são enviadas para o algoritmo de aprendizagem incremental, esses dados podem ser imediatamente descartados posteriormente, apresentando a vantagem de usar pouca memória virtual e esforço computacional. No procedimento em batelada, quando são necessários reajustes no modelo devido a novos dados ou requisitos, o modelo 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 possui modelos que são projetados de forma batelada, utilizando um conjunto de dados históricos que não cobrem todas as possibilidades de operações e ocorrências (READ et al., 2012). Ao aplicar a aprendizagem incremental com os modelos evolutivos, busca-se desenvolver uma solução que seja o mais próximo quanto possível àquela alcançada por sua versão em batelada (LUGHOFER, 2011).

Do ponto de vista matemático, seja $\mathbf{y}_{N+1, \dots, N+p}$ as p novas observações das variáveis y^1, y^2, \dots, y^m . Uma atualização de modelo incremental I a partir do modelo anterior f_N , que foi estimado a partir das N amostras iniciais, é definido por:

$$f_{N+p} = I(f_N, \mathbf{y}_{N+1, \dots, N+p}) \quad (4.1)$$

sendo que a atualização incremental do modelo é feita apenas considerando as novas p amostras e o modelo antigo, não usando nenhum dado anterior. Caso $p = 1$, tem-se uma aprendizagem incremental amostra por amostra, caso contrário, tem-se uma aprendizagem incremental em bloco (LUGHOFER, 2011).

Outra classificação de aprendizagem que foi relacionada aos modelos evolutivos é debatida em Angelov (2013), que descreve o conceito de aprendizagem autônoma (*Autonomous Learning*). 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 modelos evolutivos e autônomos devem possuir a capacidade de aprender rapidamente e extrair conhecimento desses dados coletados, 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 evolutivo (ANGELOV, 2013; LUGHOFER, 2011; KASABOV, 2007). Portanto, esse tipo de abordagem deve ter característica de auto-monitoramento e auto-adaptação para alcançar essas funcionalidades, principalmente no contexto de *data streams*.

Na maioria das aplicações, os modelos evolutivos contam com uma estrutura baseada em modelos *fuzzy*, redes neurais ou redes *neuro-fuzzy* (SKRJANC et al., 2019). Em especial, os sistemas *fuzzy* e *neuro-fuzzy* representam o conhecimento humano no formato de regras SE-ENTÃO, sendo interpretáveis e apresentando transparência (WANG, 1997), que são metas buscadas ao desenvolver modelos evolutivos. Outro aspecto importante é que os modelos *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 (SKRJANC et al., 2018).

4.3 Estrutura Básica dos Modelos *Fuzzy* Evolutivos

Os modelos *fuzzy* se baseiam no conceito de lógica *fuzzy* (ZADEH, 1965). Ao contrário da lógica *booleana* tradicional, na qual os valores de pertinência são iguais a 1 ou a 0, na lógica *fuzzy* esses valores de pertinência podem variar no intervalo de [0,1]. Portanto, uma das principais características da lógica *fuzzy* é a capacidade de representar afirmações vagas, expressando incertezas e o conhecimento incompleto do ser humano (WANG, 1997). Os elementos funcionais básicos de um sistema *fuzzy* são os denominados conjuntos *fuzzy*, que particionam o espaço associado ao conjunto de dados de entradas e saída. Cada conjunto *fuzzy* é relacionado a uma função de pertinência que apresenta valores entre [0,1], podendo ser do tipo triangular, trapezoidal, gaussiana, entre outras formas (WANG, 1997). As funções de pertinência são combinadas por operadores de norma-t ou norma-s para formular as partes antecedentes das regras *fuzzy*. Por outro lado, é possível também que os consequentes sejam formulados a partir desses conjuntos *fuzzy* de saída, mas, em alternativa, podem ser valores *singleton* ou mesmo outros tipos de funções matemáticas. Os valores de pertinência dos antecedentes e consequentes podem ser aplicados para produzir o raciocínio aproximado *fuzzy*, através das diversas alternativas de máquinas de inferência *fuzzy*. O modelo *fuzzy* pode produzir um valor de saída a partir de um conjunto de entradas utilizando o resultado desse raciocínio aproximado aplicado ao conjunto de regras *fuzzy* que compõem esse sistema (BABUSKA, 1998).

De forma tradicional, a estrutura dos sistemas *fuzzy* é fixa e sua definição é baseada no conhecimento *a priori* a respeito do problema em questão por parte dos especialistas ou a partir dos dados disponíveis para treinamento (WANG, 1997). Todavia, na maioria desses casos, ignora-se a evolução dinâmica desses problemas, sendo que pode ser necessário atualizar o modelo vigente em alguns casos. À vista disso, os modelos *fuzzy* evolutivos ajudam a lidar com esse aspecto (ANGELOV, 2002). A combinação das características de aproximação universal dos sistemas *fuzzy* com a aprendizagem incremental e a modificação da estrutura de base de regras *fuzzy* tornou-se bem vantajosa para aplicações que lidam com fluxo de dados ou *data streams* que chegam a todo instante.

Lughofer (2011) lista algumas aplicações interessantes de modelos *fuzzy* evolutivos:

- Previsão, estimação e filtragem de dados;
- Aplicações de classificação;
- Aplicações de 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 aos 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.

A determinação da estrutura do modelo evolutivo depende da representação aplicada em sua modelagem. No caso de modelos *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 e os operadores de defuzzificação (centro de gravidade, vencedor leva tudo, etc.) (ANGELOV, 2013). Essa tese está focada nas estruturas de sistemas *fuzzy* e *neuro-fuzzy*, pois se considera vantajosa a capacidade de lidar com incertezas, de apresentar aproximação universal, de ter regras interpretáveis e transparentes e, por fim, de se adaptar bem às características da teoria de modelos evolutivos.

Sendo assim, duas principais arquiteturas dos sistemas *fuzzy* tradicionais podem ser implementadas no contexto evolutivo, que são as do tipo Mamdani ou Takagi-Sugeno (TS). A diferença fundamental entre ambas é que o Mamdani fornece uma saída representada por conjuntos *fuzzy*, enquanto o tipo TS fornece uma saída que é uma expressão matemática (WANG, 1997). Entretanto, essa tese foca no modelo TS, pois a metodologia de previsão de séries temporais desenvolvida aplica uma formulação matemática nos consequentes, o que será detalhado no Capítulo 5. Usualmente, as partes consequentes das regras do modelo *fuzzy* TS são representações matemáticas lineares e o mapeamento entre entradas e saídas é do tipo MISO (*multi-input-single-output*). Todavia, no ponto de vista evolutivo, esse mapeamento pode ser do tipo MIMO (*multi-input-multi-output*), cujas regras são representadas por (ANGELOV, 2010):

$$\begin{aligned} \text{Regra } i: & \text{ SE } (d_1 \sim f_1^i) \text{ E } \dots \text{ E } (d_n \sim f_n^i) \\ & \text{ ENTÃO } \left(y_1^i = a_{01}^i + \sum_{j=1}^n a_{j1}^i d_j \right) \text{ E } \dots \text{ E } \left(y_m^i = a_{0m}^i + \sum_{j=1}^n a_{jm}^i d_j \right) \end{aligned} \quad (4.2)$$

sendo que $i = 1, \dots, R$ (R é a quantidade de regras *fuzzy*); d_1, \dots, d_n correspondem as n entradas incluídas no vetor de entradas \mathbf{d} ; f_1^i, \dots, f_n^i são os pontos focais do j -ésimo conjunto *fuzzy* de entrada e para a regra i ; y_1^i, \dots, y_m^i são as m saídas locais da i -ésima regra e; $a_{0l}^i, \dots, a_{jm}^i$ ($j = 1, \dots, n$ e $l = 1, \dots, m$) são os parâmetros dos consequentes relacionados a cada regra *fuzzy* i e entrada j . O símbolo \sim pode ser interpretado como "está próximo", indicando a proximidade da entrada com o ponto focal do conjunto *fuzzy* correspondente.

Cada saída local associada a regra *fuzzy* tem uma contribuição na saída global do sistema. Comumente, para mensurar essa contribuição, o grau de ativação τ^i é definido por uma norma-

t (geralmente o produto) entre as funções de pertinência de cada regra, conforme a seguir (ANGELOV, 2010):

$$\tau^i = \prod_{j=1}^n \mu_j^i(\mathbf{d}) = \mu_1^i(\mathbf{d}) \times \cdots \times \mu_n^i(\mathbf{d}), \quad (4.3)$$

para $i = 1, \dots, R$ e $\mu_j^i(\mathbf{d})$ sendo uma função de pertinência que pode ser de vários tipos, mas usualmente é considerada gaussiana como segue:

$$\mu_j^i(\mathbf{d}) = e^{-\frac{(f_j^i - d_j)^2}{2(\sigma_j^i)^2}}, \quad (4.4)$$

para σ_j^i a zona de influência do *cluster*, que será comentado mais a frente. Entretanto, é necessário normalizar o grau de ativação τ^i para que a soma das contribuições na saída global de todas as regras seja igual a 1 ou 100%. Logo, o grau de ativação normalizado é calculado como (LUGHOFER, 2011):

$$\lambda^i = \frac{\tau^i}{\sum_{l=1}^R \tau^l}. \quad (4.5)$$

A saída global do modelo *fuzzy* é considerada como a soma ponderada das saídas locais produzidas por cada regra *fuzzy*:

$$\mathbf{y} = \sum_{i=1}^R \lambda^i \mathbf{y}^i, \quad (4.6)$$

sendo \mathbf{y} o vetor de m saídas globais e \mathbf{y}^i o vetor de m saídas locais de cada regra.

De acordo com Angelov (2009), o processo de aprendizagem do modelo TS evolutivo é computacionalmente eficiente porque é totalmente recursivo. A primeira parte da aprendizagem está relacionada ao antecedente das regras *fuzzy*, sendo que é aplicado algoritmos de agrupamento (*clustering*) para particionar o espaço de dados e atualizar a estrutura da base de regras do modelo. Nesse caso, cada *cluster* no espaço possui uma regra *fuzzy* associada. Em segundo lugar, ocorre a aprendizagem dos parâmetros dos consequentes das regras *fuzzy*, que estão relacionados aos submodelos TS de cada regra.

Portanto, a aprendizagem a partir dos dados para modelos *fuzzy* evolutivos consiste na identificação da estrutura, na aprendizagem não-supervisionada dos antecedentes e na aprendizagem supervisionada 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, depende do problema específico (ANGELOV, 2010; ANGELOV, 2013; LUGHOFER, 2011). Esses submodelos podem ter diversas formas de acordo com o problema, como funções polinomiais. Contudo, sem perda de generalidade, assume-se que os submodelos são representados pelas equações da saída local da Equação 4.2. Seja \mathbf{y} um vetor com m saídas de um sistema no instante k (suprimiu-se o k para simplificar a notação) dado pela seguinte expressão:

$$\mathbf{y} = \Psi^T \boldsymbol{\theta}, \quad (4.7)$$

onde $\mathbf{y} = [y_1 \cdots y_m]^T$, $\boldsymbol{\psi} = [\lambda^1 \mathbf{d}_e^T \cdots \lambda^R \mathbf{d}_e^T]^T$ representa um vetor ponderado por λ^i com base no vetor de entradas expandido $\mathbf{d}_e = [1 \ d_1 \cdots d_n]^T$, $\boldsymbol{\theta} = [(\gamma^1)^T \cdots (\gamma^R)^T]^T$ corresponde ao vetor de parâmetros da base de regras e $\boldsymbol{\gamma}^i$ denota a matriz que contém os parâmetros dos consequentes relacionados com os R submodelos lineares, ou seja:

$$\boldsymbol{\gamma}^i = \begin{bmatrix} a_{01}^i & \cdots & a_{0m}^i \\ \cdots & \cdots & \cdots \\ a_{n1}^i & \cdots & a_{nm}^i \end{bmatrix}^T, \text{ para } i = 1, 2, \dots, R. \quad (4.8)$$

Para modelos que são formulados em batelada, técnicas para estimar seus parâmetros são facilmente aplicáveis, como o método dos mínimos quadrados. Porém, modelos evolutivos não possuem estrutura fixa, sendo que seus submodelos podem ser removidos e novos submodelos podem ser gerados a cada instante. Logo, a solução comumente utilizada para sistemas *fuzzy* TS evolutivos é uma versão ponderada por λ^i do método dos mínimos quadrados recursivos, denominada de *Fuzzy Weighted Recursive Least Squares* (FWRLS) (ANGELOV, 2013), que apresenta uma versão global (G-FWRLS) ou local (L-FWRLS).

A estimação dos parâmetros dos submodelos pode ser realizada de forma global ou local e deve ser aplicada a cada instante k . Para determinados dados de entrada e saída, a função custo global que precisa ser minimizada, considerando que a saída real \mathbf{y} está disponível, é representada por (ANGELOV, 2010):

$$E_G = \sum_{q=1}^k (\mathbf{y}_q - \boldsymbol{\psi}_q^T \boldsymbol{\theta}_q)^2, \quad (4.9)$$

ou de forma matricial:

$$E_G = (\mathbf{Y} - \boldsymbol{\Psi}^T \boldsymbol{\Theta})^T (\mathbf{Y} - \boldsymbol{\Psi}^T \boldsymbol{\Theta}) \quad (4.10)$$

para \mathbf{Y} a matriz de saída ao longo do tempo de dimensão $m \times k$, $\boldsymbol{\Psi}$ é uma matriz $R \times (n+1)$ que contém as ponderações λ^i de todas as regras nas entradas, e $\boldsymbol{\Theta}$ é a matriz global de todos os parâmetros do consequente de cada regra ao longo do tempo.

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

$$\boldsymbol{\Theta} = (\boldsymbol{\Psi}^T \boldsymbol{\Psi})^{-1} \boldsymbol{\Psi}^T \mathbf{Y}. \quad (4.11)$$

A minimização das Equações 4.9 e 4.10 garantem o melhor desempenho global para o 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. Em outras palavras, o procedimento global visa obter a melhor aproximação global para os dados, enquanto o local visa alcançar a melhor aproximação dos dados referentes a cada submodelo (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:

$$E_L^i = \sum_{q=1}^k \lambda^i(\mathbf{d}_q) (\mathbf{y}_q - \mathbf{d}_e^T \boldsymbol{\gamma}_q^i)^2. \quad (4.12)$$

É importante destacar que o critério local é dado para cada uma das R regras. Para obter uma aproximação do critério global a partir da equação acima, faz-se:

$$E_L = \sum_{i=1}^R E_L^i. \quad (4.13)$$

Representando o critério local de forma matricial:

$$E_L^i = (\mathbf{Y} - \mathbf{D}^\top \mathbf{\Gamma}^i)^\top \mathbf{\Lambda}^i (\mathbf{Y} - \mathbf{D}^\top \mathbf{\Gamma}^i), \text{ para } i = 1, 2, \dots, R, \quad (4.14)$$

sendo que a matriz $\mathbf{\Lambda}^i$ é diagonal e conta com as ponderações λ^i dos submodelos ao longo do tempo, \mathbf{D} é a matriz com as entradas em todos os instantes e $\mathbf{\Gamma}^i$ é uma matriz composta por valores instantâneos de \mathbf{d} e γ^i (ANGELOV, 2010).

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 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 covariância. Por outro lado, ao adicionar ou remover submodelos, a matriz covariância global precisa ter suas dimensões modificadas, pois existe apenas uma matriz e esta precisa agregar ou retirar informações dos submodelos gerados ou removidos (ANGELOV, 2013). Sendo assim, o procedimento local é recomendável na maioria dos casos envolvendo sistemas evolutivos, pois é mais simples de realizar a modificação da estrutura, possuem matrizes covariância com dimensões menores, são mais estáveis e mais robustos (YEN; WANG; GILLESPIE, 1998).

O método mais comumente aplicado para estimar esses parâmetros dos submodelos no caso local é o L-FWRLS, como segue (ANGELOV, 2010):

$$\begin{aligned} \gamma_{k+1}^i &= \gamma_k^i + \mathbf{C}_k^i \mathbf{d}_e \lambda_k^i (\mathbf{y}_k - \mathbf{d}_e^\top \gamma_k^i), \quad \gamma_1^i = 0 \\ \mathbf{C}_{k+1}^i &= \mathbf{C}_k^i - \frac{\lambda_k^i \mathbf{C}_k^i \mathbf{d}_e \mathbf{d}_e^\top \mathbf{C}_k^i}{1 + \lambda_k^i \mathbf{d}_e^\top \mathbf{C}_k^i \mathbf{d}_e}, \quad \mathbf{C}_1^i = \beta \mathbf{I}, \end{aligned} \quad (4.15)$$

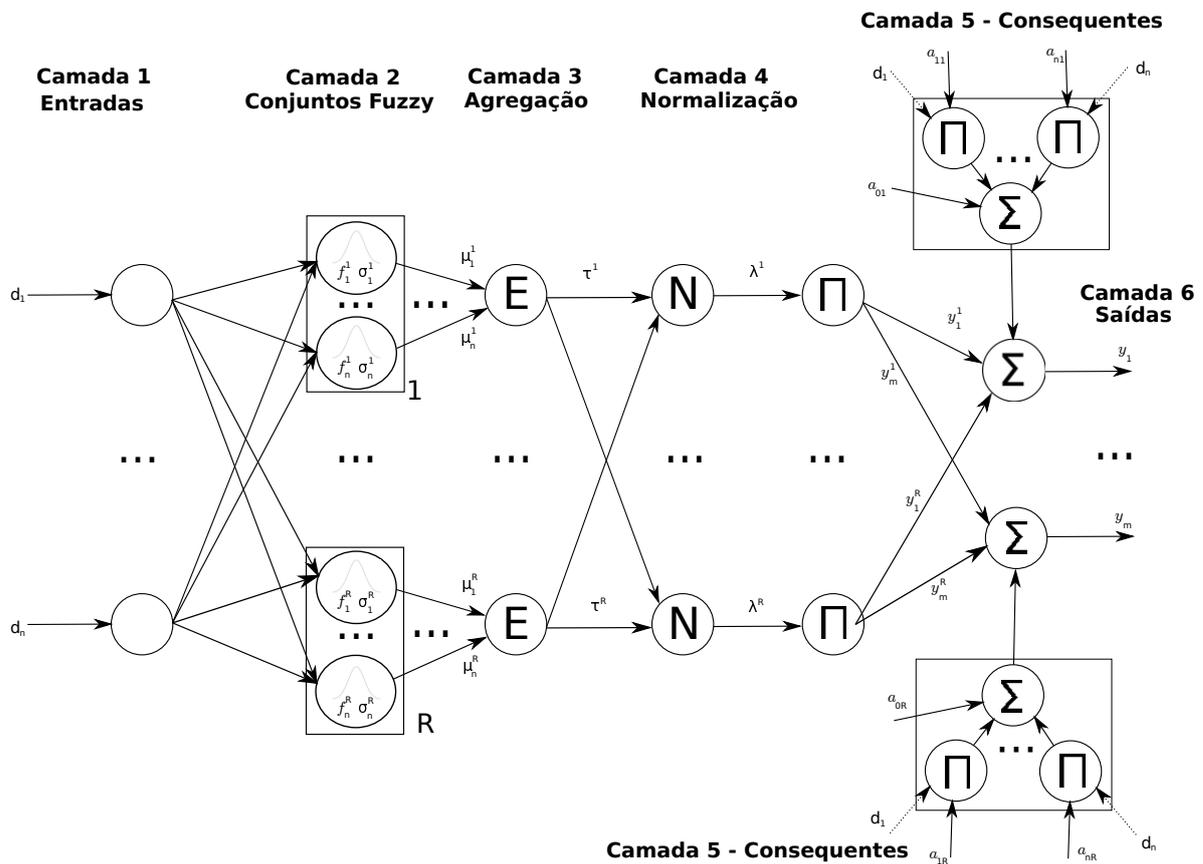
para $i = 1, \dots, R$; \mathbf{d}_e o vetor de entradas expandido $[1 \ d_1 \ \dots \ d_n]^\top$ no instante k ; \mathbf{C}_k^i a matriz covariância da regra i no instante k ; β um valor escalar elevado para inicialização da matriz covariância e; \mathbf{I} uma matriz identidade de ordem $(n+1) \times (n+1)$.

Conforme Jang (1993), o modelo *fuzzy* TS pode ser representado como uma estrutura de rede neural. Logo, a relação entre as redes neurais e modelos *fuzzy* TS leva às estruturas neuro-*fuzzy* (SERRA; BOTTURA, 2007). Os modelos *fuzzy* TS evolutivos são caracterizados, de forma alternativa, como uma rede neuro-*fuzzy* de 6 camadas, como mostra a Figura 12.

A primeira camada dessa rede neuro-*fuzzy* recebe as entradas do problema em questão, alimentando a camada 2 que possui neurônios com funções de ativação determinadas pelas

funções de pertinência no formato da Equação 4.4, isto é, as funções associadas aos conjuntos *fuzzy*. A terceira camada tem com objetivo computar o grau de ativação τ^i (Equação 4.3) de cada regra, geralmente usando a norma-t produto associado ao conectivo "E". A camada 4 fica com a responsabilidade de normalizar esses graus de ativação, de forma que as contribuições λ^i das regras para a saída global somam 1 (ou 100%), conforme Equação 4.5. Essas quatro primeiras camadas estão associadas aos antecedentes das regras *fuzzy*, enquanto as camadas 5 e 6 se preocupam com os seus consequentes. Na camada 5, os parâmetros da equação dos consequentes das regras são estimados, principalmente utilizando o método L-FWRLS. Além disso, as saídas locais y^i de cada regra são computadas nesse momento. Por fim, a camada 6 obtém a saída global do sistema a partir das saídas locais e dos graus de ativação normalizados λ^i , produzindo as m saídas, conforme Equação 4.6 (ANGELOV, 2010).

Figura 12 – Representação do Modelo *Fuzzy* Evolutivo Takagi-Sugeno por meio de uma Rede Neuro-*Fuzzy* de 6 Camadas.

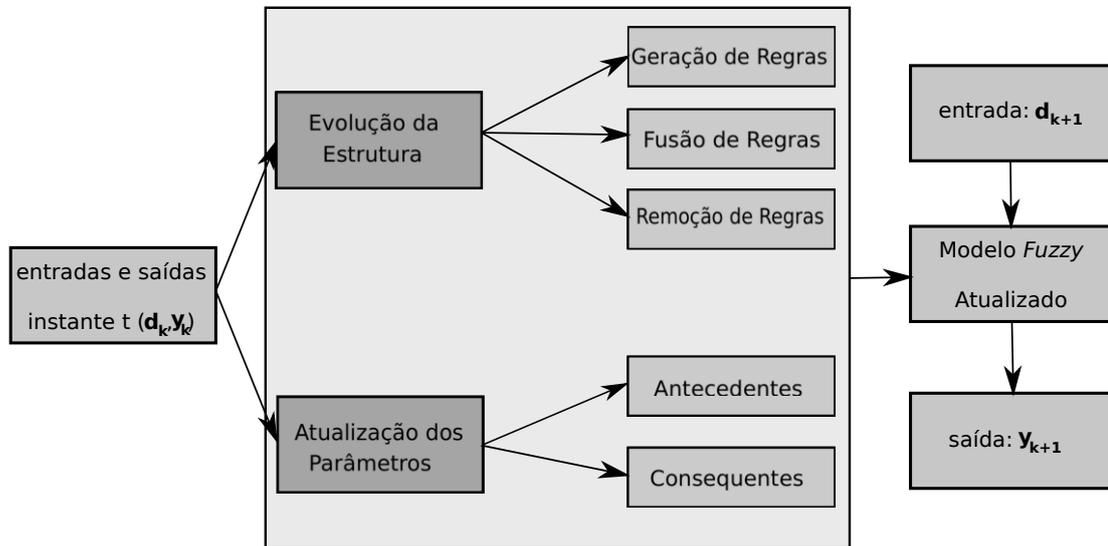


Fonte: Adaptado de Angelov (2010).

De forma geral, os modelos *fuzzy* (neuro-*fuzzy*) evolutivos apresentam a processo de aprendizagem de acordo com a Figura 13. Tem-se a evolução da sua estrutura, que está relacionada aos processos de geração, fusão e remoção de regras *fuzzy*, e a atualização dos parâmetros, associada ao cálculo de parâmetros dos antecedentes e consequentes ao longo do tempo no ponto de vista da aprendizagem incremental, como foi detalhado nessa seção (GE; ZENGE, 2020). Nas

próximas seções, serão debatidos os aspectos referentes a evolução da estrutura desses modelos evolutivos.

Figura 13 – Processo de Aprendizagem dos Modelos *Fuzzy* Evolutivos.



Fonte: Adaptado de Ge e Zenge (2020).

4.3.1 Partição Evolutiva do Espaço de Dados

Uma alternativa muito utilizada para identificar a estrutura dos sistemas é através do particionamento do espaço de dados em regiões locais (WANG, 1997). No contexto *fuzzy* ou *neuro-fuzzy*, o espaço de dados completo com dimensão n pode ser particionado considerando os valores linguísticos e as funções de pertinência. No entanto, essa forma de particionar o espaço é muito dispendiosa e pode ser intratável quando houver muitas entradas e valores linguísticos, tratando-se da chamada "maldição da dimensionalidade" (WANG, 1997; ANGELOV, 2013). A quantidade de regras nesse caso cresce exponencialmente e é calculada por:

$$R = p^n, \quad (4.16)$$

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 aprendizagem não-supervisionada a partir dos dados, ou seja, descobre 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 diferentes entre si. Cada grupo ou *cluster* pode estar associado a um submodelo que gera uma saída local que contribui para a resposta global do

sistema. Os submodelos são gerados para representar o comportamento dos dados pertencentes ao grupo que está relacionado (AGGARWAL; REDDY, 2014). Em se tratando do agrupamento evolutivo, a quantidade desses *clusters* pode variar a cada instante, sendo possível aumentar ou diminuir seu número conforme o comportamento dos dados que chegam.

No contexto dos modelos *fuzzy* 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, 2010; LUGHOFER, 2011). O 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 *cluster*.

Um das principais questões envolvendo o agrupamento é a determinação do número apropriado de *clusters* no espaço de dados. Algumas técnicas tradicionais consideram um número que é prefixado pelo especialista antes do início do procedimento de agrupamento (BABUSKA, 1998). Por sua vez, as técnicas de agrupamento para modelos 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).

Segundo Angelov (2013), o agrupamento para os modelos evolutivos deve considerar as seguintes condições 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;
 - Apresentar baixa qualidade.

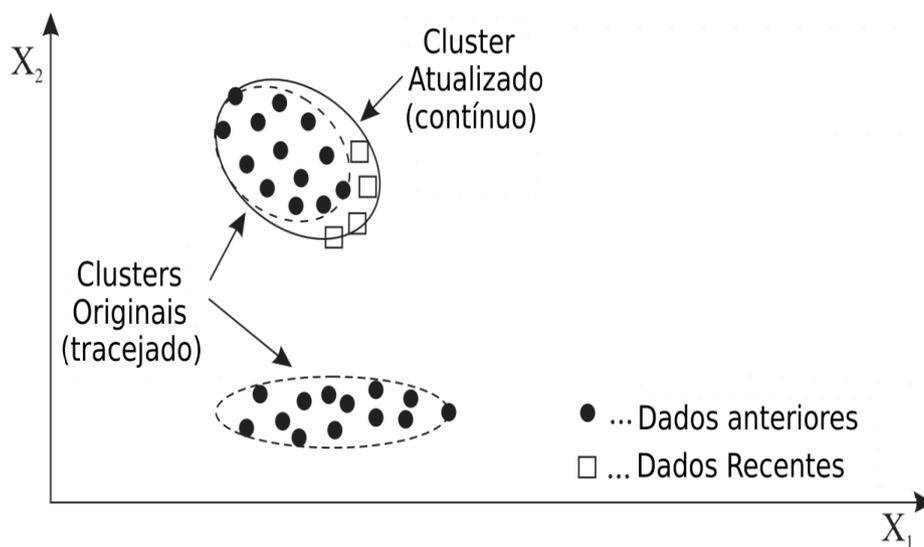
4.3.1.1 Geração de Regras Fuzzy ou Clusters

A geração de regras *fuzzy* ou de *clusters* está relacionada com a expansão do modelo evolutivo. Na literatura, vários critérios foram propostos para realizar essa tarefa de expansão.

Usualmente, é verificado se os novos dados que chegam no instante k estão situados na zona de abrangência *clusters* definidos no espaço (GE; ZENGE, 2020). Segundo Ge e Zenge (2020), de forma geral a geração de regras pode ser simplificada da seguinte forma: "Se uma determinada métrica for maior ou menor do que um limiar, uma nova regra deve ser gerada". Dentre os critérios comumente utilizados para modelos evolutivos, tem-se os baseados na distância entre os *clusters* (BARUAH; ANGELOV, 2014), nos graus de ativação (RUBIO, 2009), critérios baseados em potencial (ANGELOV; FILEV, 2004), na qualidade de dados (PRATAMA et al., 2015), etc.

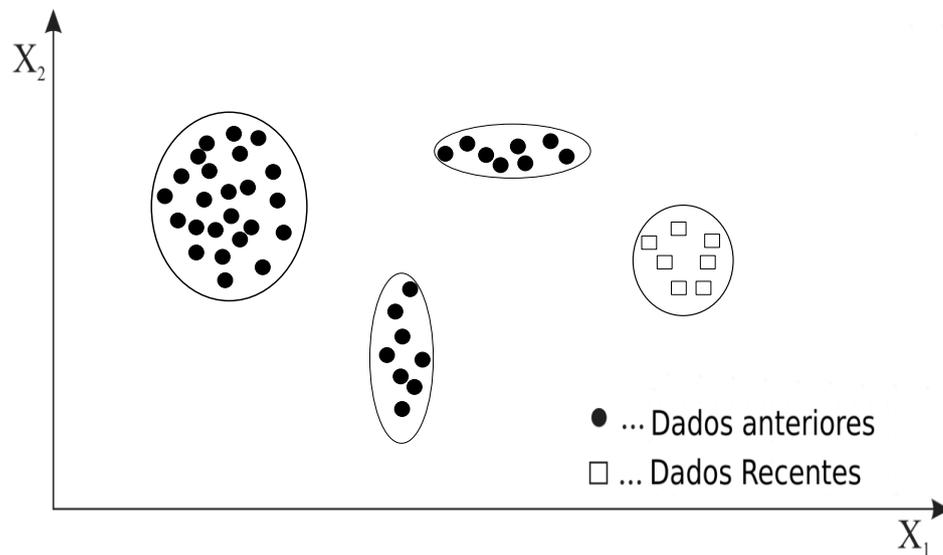
Conforme Skrjanc et al. (2019), na maioria das aplicações com sistemas evolutivos, o processo de aprendizagem começa sem modelos ou *clusters* locais. Estes são adicionados ao modelo global em tempo real para expandir seu conhecimento para novas regiões de interesse no espaço.

Figura 14 – Exemplo da Atualização de *Clusters* Devido aos Novos Dados.



Fonte: Adaptado de Skrjanc et al. (2019).

A Figura 14 mostra um exemplo de atualização de *clusters* em um espaço bidimensional quando novas amostras de dados estão chegando (marcadores retangulares). Quando as novas amostras de dados satisfazem o critério de criação de novos *clusters*, como por exemplo a distância, um novo *cluster* é adicionado para melhor representar as novas amostras, conforme a Figura 15. Por outro lado, se essas novas amostras estão próximas a um *cluster* existente, este *cluster* é atualizado para incorporar esses novos dados (Figura 14) (SKRJANC et al., 2019).

Figura 15 – Exemplo da Adição de *Clusters* Devido aos Novos Dados.

Fonte: Adaptado de Skrjanc et al. (2019).

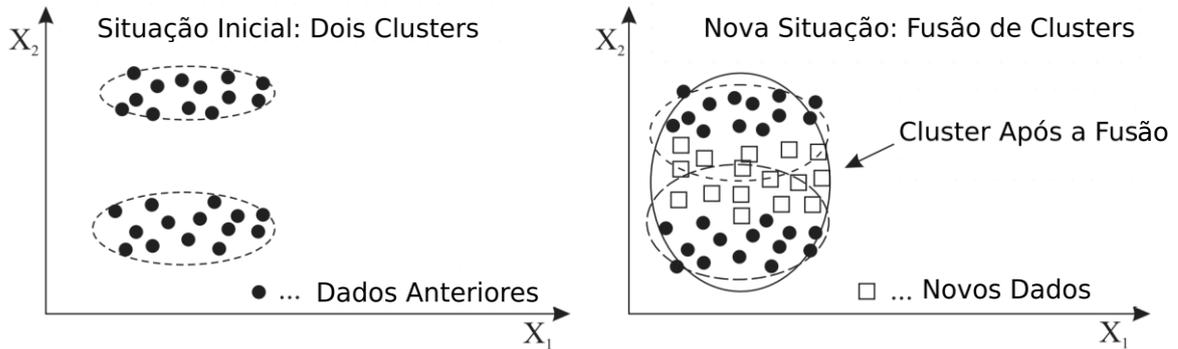
4.3.1.2 Fusão de Regras Fuzzy ou Clusters

O mecanismo para fusão de regras ou *clusters* tem como papel principal a simplificação da base de regras que compõem o modelo *fuzzy* evolutivo, avaliando-se alguns critérios. As abordagens para essa fusão evitam o conflito e redundância de regras, por meio da combinação de *clusters* similares. De acordo com Ge e Zenge (2020), de forma geral a fusão de regras pode ser resumida da seguinte forma: "Se a semelhança entre duas regras for maior do que um limiar, essas duas regras devem ser combinadas".

Portanto, a fusão é necessária se os *clusters* se sobrepõem significativamente com a evolução da estrutura (SKRJANC et al., 2019). Essas medidas de similaridade das regras *fuzzy* podem ser divididas em duas categorias: métodos baseados na teoria dos conjuntos e na teoria geométrica (SETNES et al., 1998). Medidas baseadas na teoria de conjuntos comparam a proporção da interseção de dois *clusters* com a união desses dois. As medidas geométricas são medidas baseadas em distâncias que visam comparar a similaridade de conjuntos *fuzzy* com as formas das funções de pertinência (GE; ZENGE, 2020).

Um exemplo considerando um espaço bidimensional do efeito da fusão entre *clusters* é mostrado na Figura 16 (SKRJANC et al., 2019).

Figura 16 – Exemplo da Fusão de *Clusters* Devido aos Novos Dados.



Fonte: Adaptado de Skrjanc et al. (2019).

4.3.1.3 Remoção de Regras Fuzzy ou Clusters

A remoção de regras *fuzzy* ou *clusters* também tem o objetivo de simplificar a estrutura do modelo evolutivo, removendo as regras desprezíveis. De acordo com Ge e Zenge (2020), a remoção de regras pode ser avaliada pela seguinte condição: "Se uma determinada métrica for menor/maior do que um limiar, então uma regra deve ser removida". Além disso, o algoritmo evolutivo deve apresentar uma estratégia para os chamados *outliers*, que são valores esporádicos e que podem fazer com que o sistema apresente erros. Se por ventura for criado um *cluster* para esses *outliers*, as medidas de qualidade devem remover esses *clusters* que não estão sendo utilizados, isto é, aqueles com baixa qualidade (ANGELOV, 2010).

Existem várias métricas para o monitoramento da qualidade das regras ou *clusters*. Um exemplo de parâmetro de qualidade é a idade de um *cluster* ou submodelo (ANGELOV, 2013). A idade está relacionada com o tempo de existência de um *cluster* no espaço de dados. Sua fórmula matemática é:

$$idade_k^i = k - \frac{\sum_{q=1}^{S_k^i} T_q}{S_k^i}, \text{ para } i = 1, 2, \dots, R, \quad (4.17)$$

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:

$$\begin{aligned} \text{SE } idade^i > \overline{idade^i} + \delta_{idade^i} \\ \text{ENTÃO } R = R - 1, \end{aligned} \quad (4.18)$$

para $\overline{idade^i}$ sendo a média da variável $idade^i$ considerando cada *cluster* e δ_{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). A utilidade Ω_k^i indica o quão vem sendo utilizado um submodelo associado ao *cluster* i . Sua equação é denotada por:

$$\Omega_k^i = \frac{\sum_{q=1}^k \lambda_q^i}{k - T^i} \quad (4.19)$$

em que T^i corresponde ao instante em o *cluster* i foi gerado e λ_q^i é o grau de ativação ou ponderação do submodelo i . 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 (ANGELOV, 2010):

$$\begin{aligned} & \text{SE } (\Omega_k^i < \epsilon) \\ & \text{ENTÃO } (R = R - 1), \end{aligned} \quad (4.20)$$

sendo que ϵ geralmente é definido entre 0.03 e 0.1. Essas condições são apenas alternativas de monitoramento, sendo que outras propostas e até mesmo novas formulações podem ser aplicadas para avaliar a qualidade dos *clusters* (GE; ZENGE, 2020).

4.3.2 Padronização e Normalização dos Dados

A padronização e a normalização dos dados são aplicadas a fim de que as escalas das 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 modelos evolutivos, o fluxo de dados chega a cada instante, sendo necessárias alternativas 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 $\bar{\mathbf{d}}_k$ é (ANGELOV, 2010):

$$\bar{\mathbf{d}}_k = \frac{k-1}{k} \bar{\mathbf{d}}_{k-1} + \frac{1}{k} \mathbf{d}_k, \quad \bar{\mathbf{d}}_1 = \mathbf{d}_1, \quad (4.21)$$

onde \mathbf{d}_k são os *data streams* que chegam no instante k . O desvio padrão σ_k pode ser atualizado de forma recursiva conforme a seguir:

$$\sigma_k = \sqrt{\frac{1}{k-1} \left((k-2)[\sigma_{k-1}]^2 + \frac{k}{k-1} (\bar{\mathbf{d}}_k - \mathbf{d}_k)^2 \right)} \quad (4.22)$$

A padronização dessa amostra é baseada na subtração da média e pelo desvio padrão calculados acima considerando cada coluna do vetor \mathbf{d}_k , conforme a seguir (DUDA; HART; STORK, 2000):

$$\hat{d}_k^j = \frac{d_k^j - \bar{d}_k^j}{\sigma_k^j}, \quad j = 1, \dots, n, \quad (4.23)$$

onde $\hat{\mathbf{d}}_k$ representa o vetor de dados padronizados no instante k e n , para $j = 1, \dots, n$, indica a dimensão desses dados a cada k . Para retornar a escala original, a seguinte transformação é necessária (ANGELOV, 2010):

$$d_k^j = \hat{d}_k^j \sigma_k^j + \bar{d}_k^j, \quad j = 1, \dots, n. \quad (4.24)$$

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}_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 *min* e *max*, respectivamente, ambos na mesma ordem de \mathbf{d}_k .

Para cada valor em \mathbf{d}_k ($d_k^1, d_k^2, \dots, d_k^n$) verifica-se caso a amostra ultrapassa os limites conforme a expressão a seguir (DUDA; HART; STORK, 2000):

$$\min^j \leq d_k^j \leq \max^j, \quad \text{para } j = 1, 2, \dots, n. \quad (4.25)$$

Se a Equação 4.25 for satisfeita, então a amostra será normalizada de acordo com a seguinte relação (DUDA; HART; STORK, 2000):

$$\hat{d}_k^j = \frac{d_k^j - \min^j}{\max^j - d_k^j}, \quad \text{para } j = 1, 2, \dots, n. \quad (4.26)$$

para \hat{d}_k^j sendo a normalização para o j -ésimo valor de $\hat{\mathbf{d}}_k$. Caso a Equação 4.25 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 conforme segue:

$$d_k^j = \frac{\hat{d}_k^j \max^j + \min^j}{\hat{d}_k^j + 1}, \quad \text{para } j = 1, 2, \dots, n. \quad (4.27)$$

A padronização sempre atualiza a média e a variância para toda amostra de dados que chega, enquanto a normalização apenas atualiza suas equações quando os novos dados ultrapassam um dos limites definidos. No entanto, Angelov (2010) recomenda o uso da padronização para modelos evolutivos, pois a média e variância acumulam recursivamente as informações de todo o conjunto de dados e seus valores não mudam abruptamente.

4.4 Modelo *Autonomous Learning Multi-Model* (ALMMo)

O trabalho de Skrjanc et al. (2019) cita que várias propostas usando a teoria de modelos evolutivos foram desenvolvidas nos últimos anos. Destacam-se alguns métodos baseados na lógica *fuzzy* e *neuro-fuzzy*, como o eTS, eTS+, FLEXFIS, GS-EFS, DENFIS, SAFIS, PANFIS, etc (SKRJANC et al., 2019).

De forma específica, o modelo ALMMo (*Autonomous Learning Multi-Model*) foi apresentado no trabalho de Angelov, Gu e Principe (2017), sendo eficaz para lidar com problemas complexos, utilizando a ideia de "dividir para conquistar", ou seja, decompondo um problema complexo em um conjunto de subproblemas mais simples e combinando-os depois. Além disso, o ALMMo é ideal para lidar com *data streams*, portanto, tem a possibilidade de uma ampla variedade de aplicações e extensões que podem ser desenvolvidas em áreas como análise de dados *online*, classificação, previsão, processamento de imagem, etc. (ANGELOV; GU; PRINCIPE, 2017).

Esse modelo se caracteriza por ser orientado a dados, com sua estrutura definida pelos *data clouds* identificados a partir das observações, sem fazer quaisquer suposições anteriores sobre a distribuição de dados e outras propriedades de dados. Os *data clouds* são similares aos *clusters* definidos por densidade dos dados, não possuindo formas bem definidas, e seus centros podem ser denominados de pontos focais. Segundo Angelov e Gu (2019), os *data clouds* consistem de amostras de dados afiliadas aos pontos focais mais próximos, semelhantes ao diagrama de Voronoi.

A abordagem ALMMo tem uma elevada característica autônoma, pois seus parâmetros podem ser obtidos diretamente dos dados e atualizados recursivamente, o que melhora a eficiência do algoritmo proposto. Esse modelo pode ser observado como um conjunto de submodelos baseados em regras *fuzzy*, mas também pode ser visto como um modelo neuro-*fuzzy*. O modelo ALMMo é capaz de se auto-organizar e evoluir sua arquitetura de vários modelos a partir dos dados que chegam a todo instante de maneira não iterativa, poupando recursos computacionais (ANGELOV; GU; PRINCIPE, 2017).

Os modelos principais do ALMMo podem ser de (ANGELOV; GU, 2019):

- ordem zero, ALMMo-0;
- primeira ordem, ALMMo-1.

A estrutura da parte do antecedente das regras em todos os três tipos acima listados é representada por *data clouds* ou nuvens de dados, definidas em torno de pontos no espaço que representam os padrões de dados locais em termos de densidade dos dados. Por outro lado, a diferença entre o ALMMo-0 e ALMMo-1 está na parte consequente, sendo que o primeiro apresenta como saída uma classe ou um valor *singleton*, e o segundo contém modelos lineares em seu consequente, onde é preciso identificar os valores dos parâmetros que minimizam o erro entre a previsão do modelo e o valor verdadeiro, tipicamente utilizando o algoritmo FWRLS para essa tarefa (ANGELOV; GU, 2019).

Entretanto, o foco desse trabalho será no ALMMo-1, pois a metodologia proposta se baseou na estrutura desse modelo, conforme será detalhado no Capítulo 5. Como descrito anteriormente, a definição dos antecedentes diz respeito à identificação dos pontos focais dos

data clouds. Enquanto a parte consequente das regras, busca-se determinar os valores ótimos dos parâmetros dos modelos matemáticos locais. Esse método possui ainda uma etapa de monitoramento *online* da qualidade dos *clouds*, usando o critério de utilidade, e uma etapa de seleção *online* de entradas, cujo papel é identificar quais entradas são mais relevantes para o processo de aprendizagem, desconsiderando aquelas que o procedimento não selecionou a fim de melhorar o desempenho do algoritmo e poupar recursos computacionais (ANGELOV; GU; PRINCIPE, 2017).

4.5 Estado da Arte dos Modelos Evolutivos

As primeiras ideias sobre modelos evolutivos podem ser encontradas no trabalho de Platt (1991), onde foram realizados estudos com uma rede neural. Contudo, esse campo de pesquisa teve um avanço significativo a partir dos anos 2000, com trabalhos de Angelov (2002), de Kasabov (2002) e de Angelov e Kasabov (2005), por exemplo. Frequentemente, os modelos evolutivos são representados aplicando lógica *fuzzy*, redes neurais ou redes *neuro-fuzzy* (SKRJANC et al., 2019). Esse campo de pesquisa vem tendo destaque em alguns trabalhos nos últimos anos. No trabalho de Skrjanc et al. (2019), um panorama geral da teoria de modelos evolutivos é apresentado.

Um dos principais objetivos nas pesquisas recentes é propor melhorias para o processo de aprendizagem dos modelos evolutivos em geral. No artigo de Lughofer (2021), são propostas e examinadas variações do método FWRLS para efetuar a atualização dos parâmetros dos consequentes das regras *fuzzy* de forma mais robusta. De acordo com Ge e Zenge (2020), uma das desvantagens de alguns tipos de modelos evolutivos é a definição de limiares para os processos de aprendizagem da estrutura e a determinação de seus parâmetros, em alguns casos, sendo estabelecidos por tentativa e erro. Esses autores propõem uma nova abordagem cujos limiares são dinâmicos e que podem variar ao longo do tempo e da evolução da estrutura, elevando o nível de autonomia dos algoritmos. O artigo de Ge e Zeng (2018) propõe uma abordagem de otimização de erro local (LEOA, *Local Error Optimization Approach*) para identificar sistemas *fuzzy* TS evolutivos. LEOA tem seu método de aprendizagem dos antecedentes derivado da minimização dos erros locais e garante a otimização dos parâmetros dos consequentes das regras por um novo método de mínimos quadrados recursivos ponderados estendidos (EWRLS, *Extended Weighted Recursive Least Squares*).

Atualmente, várias novas abordagens usando a teoria de sistemas evolutivos vem sendo desenvolvidas e apresentadas. No trabalho de Evangelista e Serra (2020), é proposto um método para a identificação *neuro-fuzzy* tipo-2 de sistemas multivariáveis dinâmicos, em um contexto de aprendizagem evolutiva e incremental. Em Severiano et al. (2021), os autores apresentam o método e-MVFTS (*evolving Multivariate Fuzzy Time Series*), que é um modelo evolutivo de previsão que usa séries temporais multivariáveis em um contexto de espaço e tempo. Seu desempenho é avaliado para previsão de energia solar e eólica. No artigo de Bezerra et al. (2020), propõe-se um algoritmo de agrupamento *online* de *data streams*, denominado de AutoCloud.

O AutoCloud é uma técnica evolutiva, *online* e recursiva que não requer treinamento ou conhecimento prévio sobre o conjunto de dados. Permite ainda a criação e fusão de *clusters* de forma autônoma, conforme novas observações de dados se tornam disponíveis. Em Ashrafi, Prasad e Quek (2020), uma versão evolutiva denominada de IT2-GSETSK (*Interval Type-2 Generic Self-Evolving Takagi-Sugeno-Kang*) é apresentada neste artigo. Seus conjuntos *fuzzy* podem lidar com incertezas em dados ruidosos. Uma abordagem evolutiva simples e interpretável aplicando a aprendizagem incremental é proposta em Sa'ad, Isa e Ahmed (2020). Essa proposta começa com um sistema *fuzzy* simples contendo somente uma regra *fuzzy* que não possui nenhum termo *fuzzy* para a parte antecedente. Após disso, ele continua evoluindo adicionando novos termos nebulosos ao atributo selecionado. Além disso, é desenvolvido um mecanismo de redução de regras capaz de localizar regras com baixo impacto no sistema. Em Bao et al. (2018), um sistema neuro-*fuzzy* evolutivo baseado em correntropia é proposto para aproximação de sistemas não-lineares. Diferente do critério de erro quadrático médio comumente usado, a correntropia tem uma forte capacidade de rejeição de *outliers* por meio da captura dos momentos mais altos da distribuição do erro. Considerando os méritos da correntropia, este artigo traz contribuições para construir um modelo neuro-*fuzzy* evolutivo para alcançar uma evolução mais estável da base de regras e atualização de seus parâmetros.

Destacam-se os pesquisadores P. Angelov, X. Gu e outros que trabalharam em conjunto, cujas contribuições recentes são relevantes para o campo de sistemas evolutivos. Em Gu, Angelov e Rong (2019), é proposto um método de otimização para o SONFIS, uma versão evolutiva do sistema de inferência neuro-*fuzzy*, que auxilia o sistema a apresentar uma otimização local em algumas iterações usando os dados históricos. No trabalho de Gu e Angelov (2019), uma análise matemática detalhada para otimização das partes consequentes do modelo ALMMo é conduzida. Logo, foi apresentado um novo algoritmo de auto-reforço para otimização da estrutura e parâmetros, denominado de SBALMMo (*self-boosting ALMMo*). Em Soares, Angelov e Gu (2020), apresenta-se uma aplicação da versão estendida do ALMMo-0 no contexto neuro-*fuzzy* para classificar diferentes cardiopatias por meio de sons. Em Gu, Shen e Angelov (2020), apresenta-se uma abordagem baseada no método PSO (*Particle Swarm Optimization*) para a otimização dos parâmetros antecedentes e consequentes do ALMMo e efetivamente melhorar o desempenho do sistema, procurando iterativamente por soluções ótimas.

Há também aplicações diversas usando a teoria dos sistemas evolutivos em áreas como identificação, previsão, controle e detecção de falhas. O desenvolvimento de um modelo de sistema de inferência neuro-*fuzzy* evolutivo, baseado na abordagem DENFIS, foi proposto em Kisi, Heddam e Yaseen (2019) para modelar a radiação solar com base em um esquema univariável de temperatura do ar, apresentando resultados de previsão superiores e robustos. Em Filho e Serra (2018), 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 experimental. O artigo de Souza, Ponce e

Lughofer (2020) apresenta um novo modelo inteligente evolutivo que aplica uma aprendizagem incremental a partir de *data streams*, sendo capaz de combinar as técnicas e conceitos de redes neurais artificiais, sistemas *fuzzy* e redes artificiais de hidrocarbonetos, em que esta última agrega conceitos de química orgânica para realizar o treinamento de modelos inteligentes. O modelo proposto possui três camadas onde as duas primeiras formam um sistema de inferência *fuzzy* e a terceira camada é responsável pelo processo de defuzzificação através de conceitos baseados na ligação entre carbonos e hidrogênios. No trabalho de Ferdaus et al. (2020b), um controlador neuro-*fuzzy* evolutivo, denominado Controlador Parsimonioso (PAC) é proposto para micro veículos aéreos. Esse controlador caracteriza-se por ser flexível e computacionalmente eficiente, e adota mecanismos de crescimento e redução de regras. Em Ferdaus et al. (2020a), um novo controlador evolutivo, denominado de controlador genérico (*G-Controller*), é proposto. Esse controlador funciona em um modo totalmente *online* com um pequeno conhecimento de domínio especializado. É desenvolvido incorporando a teoria de controle por modos deslizantes com aprendizagem incremental, ou seja, sistema de inferência neuro-*fuzzy* evolutivo. Por fim, o trabalho de Camargos et al. (2020) propõe um novo modelo *fuzzy* TS evolutivo baseado em erro (EBeTS, *Error Based Evolving Takagi–Sugeno Fuzzy Model*) para prognósticos de falha quando os fenômenos de degradação exibem dinâmica não-linear e variável no tempo.

4.6 Comentários Finais do Capítulo

A metodologia proposta nessa tese para previsão de séries temporais, que será detalhada no próximo capítulo, se caracteriza por ser uma abordagem neuro-*fuzzy* evolutiva, baseando-se no método ALMMo-1. O ALMMo-1 foi selecionado devido às suas principais vantagens, que são a eficiência computacional, elevado nível de autonomia em seus parâmetros e na definição de seus *data clouds*, propriedade de dividir um problema complexo em subproblemas mais simples, e possibilidade de aplicar a aprendizagem incremental a partir de *data streams*. Portanto, o modelo neuro-*fuzzy* apresentado na tese é baseado em dados e é preparado para lidar com *data streams*, dados da série temporal que chegam a todo instante k .

Sendo assim, o processo de aprendizagem incremental é interessante para lidar com esse fluxo de dados a cada momento, não sendo necessário fazer o treinamento aplicando todo o conjunto de dados histórico das séries temporais para atualizar a estrutura da neuro-*fuzzy*. Como a proposta é evolutiva, a atualização dessa estrutura consiste em modificar a quantidade de regras que pode aumentar ou diminuir, ajustando-se à dinâmica das amostras da série temporal.

Como se observou no estado da arte dos modelos evolutivos, essa tese visa contribuir com esse campo de pesquisa que vem apresentando destaque nos últimos anos, propondo uma abordagem híbrida que utiliza conceitos de séries temporais, Análise Espectral Singular, componentes não-observáveis e sistemas evolutivos com o objetivo de realizar previsões. Logo, a proposta está em um contexto que precisa lidar com dados que chegam a todo instante e necessita atualizar sua base de conhecimento *fuzzy* para se adaptar aos dados.

Parte III

CONTEXTUALIZAÇÃO METODOLÓGICA

5 METODOLOGIA PARA PREVISÃO DE SÉRIES TEMPORAIS

5.1 Descrição Geral da Metodologia Proposta: PRSSA+ENFN

A metodologia proposta nessa tese para previsão de séries temporais foi nomeada de PRSSA+ENFN, uma combinação dos métodos PRSSA (*Parallel Recursive Singular Spectrum Analysis* ou Análise Espectral Singular Recursiva e Paralela) e ENFN (*Evolving Neuro-Fuzzy Network* ou Rede Neuro-Fuzzy Evolutiva), formando uma abordagem híbrida. O objetivo do PRSSA é extrair as componentes não-observáveis de séries temporais recursivamente para cada instante k . Esta proposta é baseada no método SSA tradicional (Capítulo 3) e é uma nova versão do SSA recursivo proposto em Rodrigues Júnior e Serra (2017), onde os principais diferenciais são a extensão para o caso multivariável e a aplicação do conceito de Matrizes de Perturbação que será tratado nas próximas seções. No caso específico de séries multivariáveis, cada uma possuirá um procedimento computacional associado a fim de extrair suas componentes, sendo assim, a essa propriedade do PRSSA denominou-se como paralela.

Por outro lado, a ENFN tem a função de aprender a partir dos dados dessas componentes extraídas para realizar as previsões. Ela considera esses dados para adaptar e evoluir sua estrutura. Portanto, a ENFN se caracteriza por ser adaptativa e evolutiva. É adaptativa pois seus parâmetros são atualizados a cada instante k com a chegada de novos dados. A característica evolutiva advém da mudança da estrutura do modelo no formato de regras *fuzzy*. Dessa forma, a ENFN realiza uma aprendizagem incremental (LUGHOFER, 2011), sendo que para cada nova informação, esta rede *neuro-fuzzy* evolutiva pode adicionar em sua base de conhecimento novas regras, removê-las ou atualizá-las, dependendo da dinâmica das observações das componentes não-observáveis. Logo, esta abordagem não necessita considerar toda a informação histórica da série para atualizar sua estrutura a cada nova observação, uma característica vantajosa que poupa recursos computacionais. Esta aprendizagem incremental e evolutiva é aplicada no método ALMMo-1 (*Autonomous Learning Multi-Model Systems*), proposto em Angelov, Gu e Principe (2017) e apresentado de forma geral na Seção 4.4. O modelo ALMMo-1 serviu como base para o desenvolvimento da ENFN proposta na tese.

O método ALMMo-1 se caracteriza por ser evolutivo, recursivo e orientado a dados, ou seja, boa parte de seus parâmetros e sua estrutura podem ser definidos diretamente a partir dos dados. Apresenta também um elevado nível de característica autônoma, reduzindo a intervenção humana em seus procedimentos. A estrutura de aprendizagem é baseada em regiões no espaço denominadas *data clouds*, que assemelham-se aos *clusters*, principalmente com aqueles baseados em densidade e *fuzzy clustering* (LAOHAKIAT; SA-ING, 2021). No entanto, o termo *data cloud* foi apresentado pela primeira vez em Angelov e Yager (2011) para aplicação em algoritmos evolutivos e autônomos. Dessa forma, é possível associar a cada *cloud* uma regra *fuzzy*.

As proposições consequentes das regras *fuzzy* na ENFN proposta são constituídas por modelos de vetores autorregressivos, sendo que os regressores correspondem às próprias componentes não-observáveis extraídas. Outrossim, essa abordagem possibilita obter uma relação multivariável entre as várias componentes, conforme será analisado na Seção 5.3.3.

Tendo em vista os objetivos principais que existem ao analisar séries temporais (Seção 2.1), a metodologia proposta foca em dois destes: a previsão quantitativa e a decomposição em componentes não-observáveis. Além do mais, o método PRSSA+ENFN tem a flexibilidade de lidar com séries temporais univariáveis ou multivariáveis. Dependendo do problema de previsão em questão, a abordagem proposta pode trabalhar tanto com a previsão direta (Seção 2.3.3.2) quanto a recorrente (Seção 2.3.3.1).

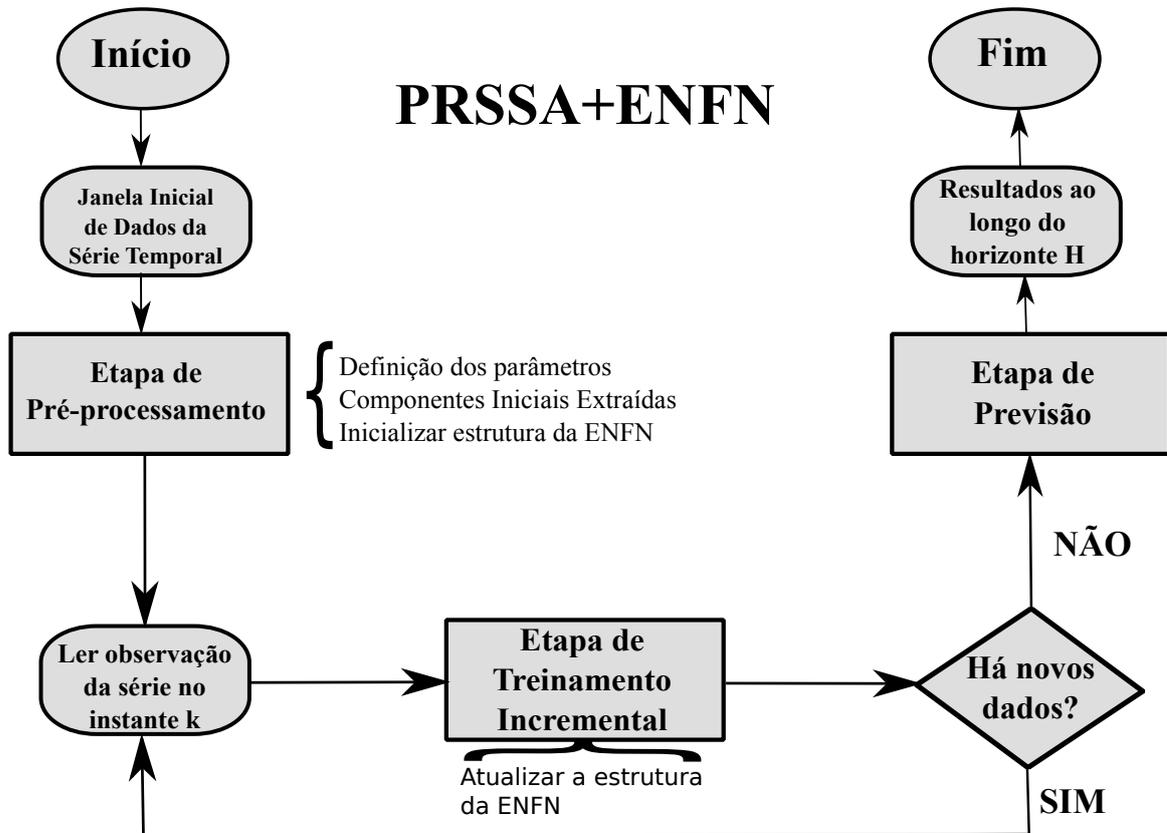
Outra característica importante do PRSSA+ENFN proposto é que este realiza primeiramente a previsão das componentes não-observáveis, para depois reconstruir as séries originais no futuro usando as observações das componentes previstas. Dessa forma, essa abordagem considera a estratégia de "dividir para conquistar" (ANGELOV, 2013), pois decompõe uma série temporal complexa em um conjunto de padrões mais fáceis de lidar e de prever.

Levando em consideração essas características apresentadas, o PRSSA+ENFN proposto é organizado em três etapas de operação para previsão de séries temporais:

- Etapa de Pré-processamento;
- Etapa de Treinamento Incremental;
- Etapa de Previsão.

Na etapa de pré-processamento, um dos objetivos é determinar alguns parâmetros e variáveis necessárias, considerando um conjunto inicial de observações das séries temporais. Além disso, esse mesmo conjunto de dados é aplicado para extrair as componentes não-observáveis iniciais e para inicializar a estrutura da ENFN. No treinamento incremental, novos dados da série temporal são recebidos a cada instante k , sendo assim, a ENFN realiza uma aprendizagem incremental para atualizar sua base de conhecimento no formato de regras *fuzzy*. Finalmente, as observações futuras das componentes não-observáveis são determinadas na etapa de previsão, que pode ser direta ou recorrente. Nessa etapa, a ENFN não evolui, mantendo sua estrutura fixa para computar os valores futuros das componentes. Em seguida, os resultados de previsão das componentes são agrupados para prever a série temporal original. O procedimento geral da abordagem PRSSA+ENFN é descrito na Figura 17. Detalhes sobre cada etapa serão apresentados nas seções a seguir.

Figura 17 – Relação entre as Etapas do PRSSA+ENFN: Etapas de Pré-processamento, Treinamento Incremental e Previsão.



Fonte: Autor.

5.2 Etapa de Pré-processamento

Antes do treinamento incremental do PRSSA+ENFN, a etapa de pré-processamento conta com um conjunto inicial de dados da série temporal para determinar alguns parâmetros e variáveis necessárias, para extrair as componentes não-observáveis iniciais e para inicializar a estrutura da rede neuro-*fuzzy* evolutiva.

Considere $\mathbf{y}_k \in \mathbb{R}^{1 \times m}$ uma observação de uma série temporal no instante k , sendo que $\mathbf{y}_k = [y_k^1 \cdots y_k^m]$, para m a quantidade de séries envolvidas. As subseções a seguir descrevem cada uma das fases do pré-processamento.

5.2.1 Determinação dos Parâmetros do Método PRSSA+ENFN

Alguns parâmetros dependem da análise do problema de previsão de séries temporais e devem ser informados diretamente pelo usuário do PRSSA+ENFN. São eles:

- H : horizonte de previsão, que corresponde a quantidade de observações da série temporal que se deseja prever;

- T : quantidade de observações disponíveis para serem usadas no pré-processamento e treinamento do PRSSA+ENFN;
- N : quantidade total de observações da série, sendo que $N = T + H$;
- m : quantidade de séries temporais em $\mathbf{y}_k = [y_k^1 \cdots y_k^m]$, para $m \geq 1$. Se $m = 1$ então é um problema de previsão univariável, caso seja $m > 1$, é multivariável;
- p : corresponde ao passo de previsão, onde a ENFN deve aprender o mapeamento das componentes do instante $k - p$ para o instante k na etapa de treinamento incremental. Portanto, as componentes no instante $k - p$ são consideradas as entradas da ENFN;
- n_{max} : quantidade máxima de componentes que podem ser extraídas para cada uma das m séries temporais. Geralmente, n_{max} não é superior a 10, a fim de evitar um custo computacional excessivo.

Os próximos parâmetros são mais críticos e estão relacionados especificamente ao PRSSA+ENFN, influenciando de forma importante nos resultados da previsão. Portanto, seus valores devem ser selecionados adequadamente. A descrição desses parâmetros está abaixo:

- W : quantidade de observações da série temporal \mathbf{y}_k utilizadas exclusivamente na etapa de pré-processamento, isto é, para $k = 1, \dots, W$, sendo que $W < T$;
- n_l : quantidade de componentes extraídas de cada série temporal l em $\mathbf{y}_k = [y_k^1 \cdots y_k^m]$, para $l = 1, \dots, m$. Considera-se o intervalo $2 \leq n_l \leq n_{max}$;
- ϕ : fator de sobreposição dos *data clouds*, sendo que $0.1 \leq \phi \leq 0.5$;
- η_0 : fator de esquecimento para simplificação da base de regras *fuzzy*, onde $0.01 \leq \eta_0 \leq 0.3$.

Conforme Angelov e Gu (2019), os intervalos dos fatores de sobreposição e de esquecimento dependem de cada aplicação específica e da experiência do usuário. Na Seção 5.5, duas estratégias para atribuição de valores a esses parâmetros críticos acima são apresentadas.

5.2.2 Extração das Componentes Não-Observáveis Iniciais

Cada série temporal l possui um bloco dedicado para extrair suas componentes não-observáveis no pré-processamento, conforme Figura 18. Seja y_k^l uma observação da série temporal l no instante k , lembrando que $\mathbf{y}_k = [y_k^1 \cdots y_k^m]$. A notação matemática das componentes é

denotada por:

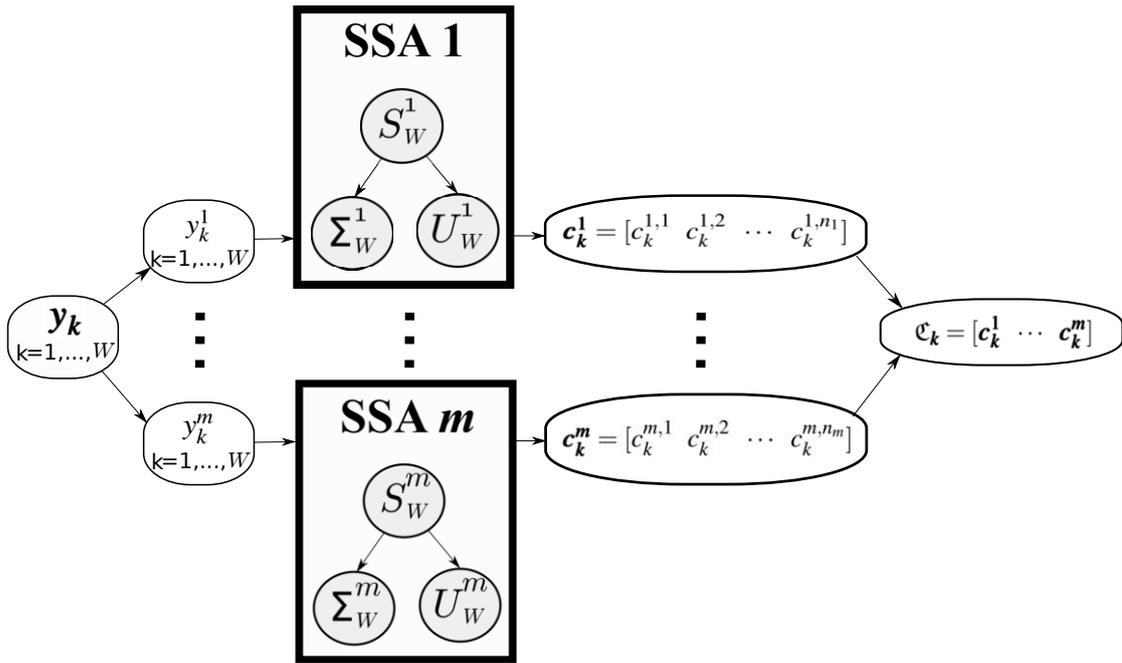
$$\left\{ \begin{array}{l} \text{Componentes de } y_k^1 : \mathbf{c}_k^1 = [c_k^{1,1} \ c_k^{1,2} \ \dots \ c_k^{1,n_1}], \ \mathbf{c}_k^1 \in \mathbb{R}^{1 \times n_1}; \\ \vdots \\ \text{Componentes de } y_k^m : \mathbf{c}_k^m = [c_k^{m,1} \ c_k^{m,2} \ \dots \ c_k^{m,n_m}], \ \mathbf{c}_k^m \in \mathbb{R}^{1 \times n_m}. \end{array} \right. \quad (5.1)$$

Notação Geral ($l = 1, \dots, m$):

$$\text{Componentes de } y_k^l : \mathbf{c}_k^l = [c_k^{l,1} \ c_k^{l,2} \ \dots \ c_k^{l,n_l}], \ \mathbf{c}_k^l \in \mathbb{R}^{1 \times n_l},$$

onde $c_k^{l,j}$, para $l = 1, \dots, m$ e $j = 1, \dots, n_l$, corresponde a j -ésima componente extraída de y_k^l no instante k , enquanto \mathbf{c}_k^l é um vetor que engloba esse conjunto de n_l componentes da série temporal l no instante k .

Figura 18 – Extração das Componentes Não-Observáveis na Etapa de Pré-processamento. Cada Série Temporal l ($l = 1, \dots, m$) possui um Bloco SSA Exclusivo (Característica Paralela).



Fonte: Autor.

Define-se n como sendo a quantidade total de componentes extraídas das m séries temporais em \mathbf{y}_k , ou seja:

$$n = \sum_{l=1}^m n_l. \quad (5.2)$$

Logo, um vetor n -dimensional que agrega todas essas componentes no instante k é descrito nesse trabalho por \mathbf{C}_k :

$$\mathbf{C}_k = [\mathbf{c}_k^1 \ \dots \ \mathbf{c}_k^m], \ \mathbf{C}_k \in \mathbb{R}^{1 \times n}. \quad (5.3)$$

Para extrair as componentes não-observáveis e formular o vetor \mathfrak{C}_k na Equação 5.3, emprega-se nesse trabalho uma abordagem baseada no método SSA. Na etapa de pré-processamento e para as primeiras W observações de \mathbf{y}_k , o método SSA tradicional (Capítulo 3) é aplicado para cada série l ($l = 1, \dots, m$), logo, uma característica paralela conforme Figura 18.

Uma janela inicial de dados para cada série temporal l , representada por \mathbf{Y}_W^l , é exigida nessa etapa, sendo:

$$\mathbf{Y}_W^l = [y_1^l \quad y_2^l \quad \dots \quad y_W^l]^T, \quad \mathbf{Y}_W^l \in \mathbb{R}^{W \times 1}, \quad (5.4)$$

ou seja, \mathbf{Y}_W^l contém as W primeiras observações da série temporal l , para $k = 1, \dots, W$. Como notação nesse trabalho, o índice l indica que o termo está relacionado a l -ésima série e o índice W indica que as primeiras W observações da série l são consideradas por esse termo. Então, os dados de cada \mathbf{Y}_W^l são organizados de forma alternativa denominada de Matriz Trajetória \mathbf{M}_W^l , semelhante ao SSA tradicional como segue:

$$\mathbf{M}_W^l = \begin{bmatrix} y_1^l & y_2^l & \dots & y_{\mathcal{K}_W^l}^l \\ y_2^l & y_3^l & \dots & y_{\mathcal{K}_W^l+1}^l \\ \vdots & \vdots & \vdots & \vdots \\ y_{L_l}^l & y_{L_l+1}^l & \dots & y_W^l \end{bmatrix}_{L_l \times \mathcal{K}_W^l}, \quad (5.5)$$

$$\text{para } \mathcal{K}_W^l = W - L_l + 1,$$

onde o termo \mathcal{K}_W^l é a quantidade de colunas na Matriz Trajetória \mathbf{M}_W^l após W observações e L_l é o tamanho da janela do operador SSA para série l , apresentado no Capítulo 3 (não está relacionado a definição da janela \mathbf{Y}_W^l). Destaca-se que a condição $L_l = n_l$ é praticada nesse trabalho, ou seja, cada autovetor ou autovalor está relacionado a uma componente não-observável específica. Essa condição é empregada pois facilita a formulação matemática recursiva na etapa de treinamento do PRSSA+ENFN, reduz a quantidade de parâmetros a serem determinados e, além disso, considerar $L_l \neq n_l$ não influenciou significativamente os resultados de previsão da metodologia proposta. Destaca-se que Groth et al. (2017) também aplicaram essa mesma condição em seu trabalho.

Por conseguinte, a Matriz Covariância Inicial relacionada com a série l é definida por:

$$\mathbf{S}_W^l = [\mathbf{M}_W^l][\mathbf{M}_W^l]^T, \quad \mathbf{S}_W^l \in \mathbb{R}^{L_l \times L_l}. \quad (5.6)$$

Em seguida, usando a Matriz de Covariância Inicial \mathbf{S}_W^l , as etapas do SSA denominadas de Decomposição de Valores Singulares ou SVD (Seção 3.2.2), Agrupamento de Índices (Seção 3.3.1) e Média Diagonal (Seção 3.3.2) são aplicadas da forma original. Na etapa do SVD, os autovetores e autovalores de \mathbf{S}_W^l são calculados. Logo, \mathbf{U}_W^l é uma matriz $L_l \times L_l$ composta por um conjunto de autovetores em cada uma de suas colunas e $\mathbf{\Sigma}_W^l$ contém os L_l valores singulares em sua diagonal.

O procedimento da Média Diagonal do método SSA tradicional é aplicado para obter os dados das componentes \mathbf{c}_k^l (Equação 5.1), considerando cada série temporal l . Essa extração inicial de componentes não-observáveis para as W primeiras amostras é importante para que a decomposição recursiva obtenha melhores resultados na etapa de treinamento incremental. Novamente, a Figura 18 representa o procedimento descrito de extração paralela de componentes não-observáveis na etapa de pré-processamento.

Após esse procedimento, é necessário padronizar o vetor de componentes \mathbf{C}_k (Equação 5.3), ou seja, todos os dados das componentes não-observáveis devem ser padronizados em escalas comparáveis para contribuir igualmente na aprendizagem da ENFN. Assim, a média inicial desse vetor $\bar{\mathbf{C}}_W$ e o desvio padrão $\boldsymbol{\sigma}_W$ para as W primeiras observações são calculados como segue:

$$\begin{aligned}\bar{\mathbf{C}}_W &= \text{mean}(\mathbf{C}_k), \quad \bar{\mathbf{C}}_W \in \mathbb{R}^{1 \times n}, \\ \boldsymbol{\sigma}_W &= \text{std}(\mathbf{C}_k), \quad \boldsymbol{\sigma}_W \in \mathbb{R}^{1 \times n}, \\ &\text{para } k = 1, \dots, W.\end{aligned}\tag{5.7}$$

Tendo calculado a média e desvio padrão acima, é realizada a padronização inicial dos dados dos componentes para cada elemento da dimensão n dos vetores:

$$\hat{\mathbf{C}}_k = \frac{\mathbf{C}_k - \bar{\mathbf{C}}_W}{\boldsymbol{\sigma}_W}, \quad k = 1, \dots, W,\tag{5.8}$$

para $\hat{\mathbf{C}}_k$ sendo os dados das componentes padronizadas no instante k . Na etapa de treinamento incremental, a média e o desvio padrão podem ser atualizados recursivamente para cada k , computando o novo vetor $\hat{\mathbf{C}}_k$. Destaca-se que o circunflexo " $\hat{}$ " indica nesse trabalho que os dados estão padronizados.

Depois de obter $\hat{\mathbf{C}}_k$ ($k = 1, \dots, W$), o parâmetro s_j é estimado usando a função de autocorrelação amostral (Seção 2.2.2) para cada componente j , sendo que $j = 1, \dots, n$. Dessa forma, s_j pode assumir um valor distinto para cada componente j . O parâmetro s_j indica o atraso no tempo que contém uma observação com significativa correlação com a observação a ser prevista. O gráfico da função de autocorrelação amostral pode identificar uma observação $k - s_j$ que tem uma boa correlação com a amostra atual no instante k . Na metodologia proposta, o valor de s_j é determinado automaticamente a partir dos dados de cada componente. Para séries sazonais, esse termo s_j representa naturalmente a sazonalidade da componente (Seção 2.2.3), enquanto para componentes que não possuem sazonalidade definida, pode representar uma observação com certa correlação com o instante atual e associadas a outros padrões, como tendência (BROCKWELL; DAVIS, 2002). Os dados das componentes no instante $k - s_j$ são importantes nos modelos de vetores autorregressivos nos consequentes das regras *fuzzy*, o que será apresentado na Equação 5.36. Ademais, exemplos de estimação de s_j serão descritos na Seção 6.2.2 de Resultados Experimentais. Para determinação de cada s_j , o processo de extração de componentes não-observáveis iniciais efetuado no pré-processamento para as W primeiras observações é essencial.

5.2.3 Inicialização da Rede Neuro-Fuzzy Evolutiva (ENFN)

A última fase da etapa de pré-processamento é a inicialização da ENFN, baseada no algoritmo ALMMo-1 e considerando as W primeiras observações de \mathbf{y}_k . Na etapa de treinamento, a ENFN aprende a partir do vetor $\hat{\mathbf{C}}_k$ para cada $k = W + 1, \dots, T$, organizando o espaço de dados em *data clouds* n -dimensionais, sendo que cada *cloud* está associado a uma regra *fuzzy*. O número de regras *fuzzy* ou *data clouds* é denotado por R e $\boldsymbol{\rho}_i \in \mathbb{R}^{1 \times n}$ representa o centro do *cloud* i no espaço, para $i = 1, \dots, R$. Por outro lado, na etapa de pré-processamento, a ENFN é inicializada com $R = 1$, e as condições iniciais de seus dois parâmetros globais são determinadas a seguir:

- $\boldsymbol{\mu}_k$: média global do vetor de componentes padronizado $\hat{\mathbf{C}}_k$, onde $\boldsymbol{\mu}_W = \hat{\mathbf{C}}_{W-p}$, para $k = W$ e p sendo o passo definido previamente na Seção 5.2.1;
- X_k : quadrado da norma Euclidiana de $\hat{\mathbf{C}}_k$, com $X_W = \|\hat{\mathbf{C}}_{W-p}\|^2$ para $k = W$ e p sendo o passo definido previamente na Seção 5.2.1.

Ambos parâmetros globais acima passam a ser atualizados recursivamente para cada instante $k = W + 1, \dots, T$ na etapa de treinamento posteriormente. Como já mencionado, o primeiro *cloud* ($R = 1$) é gerado na etapa de pré-processamento, e o centro deste é representado por $\boldsymbol{\rho}_1 = \hat{\mathbf{C}}_{W-p}$. Sempre que um novo *cloud* ou regra *fuzzy* é gerado, seus parâmetros locais e dos consequentes das regras podem ser inicializados conforme a lista abaixo, e o mesmo ocorre para o primeiro *cloud*. A seguir, os parâmetros locais de cada *cloud* são descritos:

- τ_i : instante em que o *data cloud* i é gerado;
- S_i : suporte ou o número de pontos associados ao *data cloud* i , onde $S_i = 1$ é a condição de inicialização no instante τ_i ;
- η_i : utilidade do *data cloud* i , sendo que $\eta_i = 1$ é a condição de inicialização no instante τ_i ;
- q_i : produto escalar local do *data cloud* i , onde $q_i = \|\hat{\mathbf{C}}_{\tau_i-p}\|^2$ é a condição de inicialização no instante τ_i ;
- $\boldsymbol{\rho}_i$: centro do *data cloud* i , sendo que $\boldsymbol{\rho}_i = \hat{\mathbf{C}}_{\tau_i-p}$ é a condição de inicialização no instante τ_i ;
- \mathbf{J}_k^i : matriz covariância para estimação das matrizes dos consequentes das regras *fuzzy*, onde $\mathbf{J}_{\tau_i}^i = \mathbf{I} \in \mathbb{R}^{2n \times 2n}$ é a condição de inicialização no instante τ_i ;
- $\boldsymbol{\Delta}_k^i$: parâmetros das matrizes dos consequentes, sendo que $\boldsymbol{\Delta}_{\tau_i}^i = \mathbf{0} \in \mathbb{R}^{2n \times n}$ é a condição de inicialização no instante τ_i .

Em resumo, a Tabela 3 exibe a lista dos parâmetros utilizados pelo PRSSA+ENFN, categorizando-os em 4 partes: a parte 1 mostra os parâmetros relacionados ao tipo de problema de previsão; a parte 2 apresenta os parâmetros críticos do PRSSA+ENFN que devem ser devidamente selecionados pelo usuário; a parte 3 mostra os parâmetros globais inicializados quando a ENFN é gerada após a extração inicial de componentes não observáveis; por fim, a parte 4 apresenta os parâmetros locais que são inicializados sempre que uma regra *fuzzy* ou *data cloud* é gerado.

Tabela 3 – Lista de Parâmetros usados pelo Método PRSSA+ENFN.

Parte 1 - Parâmetros do Problema de Previsão H, T, N, m, p, n_{max}
Parte 2 - Parâmetros Críticos do PRSSA+ENFN W, n_l, ϕ, η_0
Parte 3 - Parâmetros Globais da Inicialização da ENFN $\boldsymbol{\mu}_k, X_k$
Parte 4 - Parâmetros Locais do <i>Data Cloud</i> ou <i>Regra Fuzzy i</i> $\tau_i, S_i, \eta_i, q_i, \boldsymbol{\rho}_i, \mathbf{J}_k^i, \boldsymbol{\Delta}_k^i$

O Algoritmo A.2 descreve o procedimento executado pela etapa de pré-processamento.

5.3 Etapa de Treinamento Incremental

Na etapa de treinamento incremental, novas observações da série $\mathbf{y}_k = [y_k^1 \cdots y_k^m]$ chegam a cada instante k após as W primeiras observações consideradas no pré-processamento, ou seja, para $k = W + 1, W + 2, \dots, T$, sendo T (Parte 1 da Tabela 3) a quantidade de amostras para o treinamento incremental do método PRSSA+ENFN.

O primeiro passo é a extração recursiva das componentes não-observáveis, aplicando o PRSSA. Destaca-se que para cada série temporal, há um RSSA (*Recursive Singular Spectrum Analysis*) associado, portanto, o conjunto de blocos RSSA formam o PRSSA proposto. Em seguida, essas componentes são consideradas como entradas da ENFN, sendo aplicadas para atualizar, adicionar ou remover regras *fuzzy*. Essa etapa é finalizada com a determinação das componentes no instante k resultantes do método PRSSA+ENFN. As seções a seguir vão descrever detalhadamente esses passos.

5.3.1 Extração Recursiva das Componentes Não-Observáveis usando o PRSSA

Como novos dados da série temporal chegam a cada instante $k = W + 1, W + 2, \dots, T$, há um impacto na dimensão da Matriz Trajetória (Equação 5.5) de cada série temporal l , pois a quantidade de colunas dessa matriz, representada por \mathcal{X}_k^l (relacionado ao termo \mathcal{X}_W^l da Equação

5.5, para $k = W + 1, W + 2, \dots, T$), deve ser incrementada por um para incorporar as novas observações da série l no instante k , isto é, $\mathcal{X}_k^l = \mathcal{X}_{k-1}^l + 1$.

Devido a mudança do valor de \mathcal{X}_k^l a cada instante, a Matriz Trajetória de dimensão $L_l \times \mathcal{X}_k^l$, deveria acrescentar uma nova coluna para todo k em sua formulação para comportar os novos dados da série temporal l , o que não seria viável segundo uma perspectiva computacional. Portanto, o PRSSA não utiliza a Matriz Trajetória diretamente, pois considera que toda a informação histórica da série temporal l está na Matriz Covariância \mathbf{S}_k^l que apresenta dimensão fixa $L_l \times L_l$. Em outras palavras, para cada informação nova no instante k da série l , atualiza-se \mathbf{S}_k^l sem modificar sua dimensão matricial. Para atualizar cada \mathbf{S}_k^l recursivamente, o seguinte vetor é exigido:

$$\mathbf{D}_k^l = \begin{bmatrix} y_{\mathcal{X}_k^l}^l \\ y_{\mathcal{X}_k^l+1}^l \\ \vdots \\ y_k^l \end{bmatrix}_{L_l \times 1}, \quad (5.9)$$

sendo que \mathbf{D}_k^l é o Vetor de Atualização que contém observações da série temporal l em uma janela de tempo entre $k = \mathcal{X}_k^l$ e o instante atual k , em outras palavras, as L_l últimas observações dessa série. Logo, a atualização recursiva da Matriz Covariância é:

$$\mathbf{S}_k^l = \mathbf{S}_{k-1}^l + [\mathbf{D}_k^l][\mathbf{D}_k^l]^\top, \quad k = W + 1, W + 2, \dots, T, \quad (5.10)$$

$$\mathbf{D}_k^l \in \mathbb{R}^{L_l \times 1} \text{ e } \mathbf{S}_k^l \in \mathbb{R}^{L_l \times L_l},$$

lembrando que a Matriz Covariância Inicial \mathbf{S}_W^l é determinada na Equação 5.6. Para uma breve demonstração do uso da Equação 5.10, seja uma Matriz Trajetória no instante $k - 1$ definida como segue (ignora-se o índice l por simplicidade nessa demonstração):

$$\mathbf{M}_{k-1} = \begin{bmatrix} a & c \\ b & d \end{bmatrix}_{2 \times 2}, \quad (5.11)$$

logo, calculando-se \mathbf{S}_{k-1} conforme Equação 5.6, tem-se:

$$\mathbf{S}_{k-1} = \mathbf{M}_{k-1} \mathbf{M}_{k-1}^\top = \begin{bmatrix} a & c \\ b & d \end{bmatrix} \begin{bmatrix} a & c \\ b & d \end{bmatrix}^\top = \begin{bmatrix} a^2 + c^2 & ab + cd \\ ab + cd & b^2 + d^2 \end{bmatrix}_{2 \times 2}. \quad (5.12)$$

Suponha agora que, no instante k , um novo conjunto de dados, denotado por $\mathbf{D}_k = [e \ f]^\top$ (Equação 5.9), é recebido. Dessa maneira, precisa-se incorporá-los na Matriz Trajetória como uma nova coluna, modificando a dimensão dessa matriz. Realizando o mesmo procedimento acima para esse novo caso:

$$\mathbf{M}_k = \begin{bmatrix} a & c & e \\ b & d & f \end{bmatrix}_{2 \times 3}, \quad (5.13)$$

$$\mathbf{S}_k = \mathbf{M}_k \mathbf{M}_k^\top = \begin{bmatrix} a & c & e \\ b & d & f \end{bmatrix} \begin{bmatrix} a & c & e \\ b & d & f \end{bmatrix}^\top = \begin{bmatrix} a^2 + c^2 + e^2 & ab + cd + ef \\ ab + cd + ef & b^2 + d^2 + f^2 \end{bmatrix}_{2 \times 2}$$

sendo que a Matriz Covariância \mathbf{S}_k incorporou a nova informação sem modificar suas dimensões. Para observar a aplicação da Equação 5.10 proposta, usam-se as mesmas condições para \mathbf{S}_{k-1} e \mathbf{D}_k das demonstrações acima, sendo assim:

$$\mathbf{S}_k = \mathbf{S}_{k-1} + [\mathbf{D}_k][\mathbf{D}_k]^\top \quad (5.14)$$

$$\mathbf{S}_k = \begin{bmatrix} a^2 + c^2 & ab + cd \\ ab + cd & b^2 + d^2 \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \begin{bmatrix} e & f \end{bmatrix} = \begin{bmatrix} a^2 + c^2 + e^2 & ab + cd + ef \\ ab + cd + ef & b^2 + d^2 + f^2 \end{bmatrix}_{2 \times 2},$$

obtendo-se em ambos os casos matrizes equivalentes. Com base nessa demonstração, percebe-se que a Equação 5.10 pode ser aplicada recursivamente a cada instante para atualização da Matriz Covariância.

Para atualizar os autovetores \mathbf{U}_k^l e autovalores Σ_k^l para cada instante k e série l , utilizou-se a ideia de Matrizes de Perturbação, uma contribuição do trabalho de Peddaneni et al. (2004). Nesse caso, um vetor \mathbf{F}_k^l é exigido para efetuar essas atualizações recursivamente. Dessa maneira, \mathbf{F}_k^l é calculado como segue:

$$\mathbf{F}_k^l = [\mathbf{U}_{k-1}^l]^\top \mathbf{D}_k^l, \quad \mathbf{F}_k^l \in \mathbb{R}^{L_l \times 1}, \quad \mathbf{U}_{k-1}^l \in \mathbb{R}^{L_l \times L_l} \text{ e } k = W + 1, W + 2, \dots, T, \quad (5.15)$$

sendo que \mathbf{F}_k^l é um vetor L_l -dimensional, cujos elementos são representados por f_k^j ($j = 1, \dots, L_l$) e \mathbf{D}_k^l é calculado na Equação 5.9. Após o cálculo de \mathbf{F}_k^l , duas matrizes de perturbação representadas por \mathbf{P}_k^l e \mathbf{Q}_k^l são formuladas para cada série l (PEDDANENI et al., 2004):

- Matriz de Perturbação \mathbf{P}_k^l :

$$\mathbf{P}_k^l = \begin{bmatrix} p_k^1 & 0 & \dots & 0 \\ 0 & p_k^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & p_k^{L_l} \end{bmatrix}_{L_l \times L_l}; \quad (5.16)$$

- Matriz de Perturbação \mathbf{Q}_k^l :

$$\mathbf{Q}_k^l = \begin{bmatrix} 0 & q_k^{1,2} & \dots & q_k^{1,L_l} \\ -q_k^{1,2} & 0 & \dots & q_k^{2,L_l} \\ \vdots & \vdots & \vdots & \vdots \\ -q_k^{1,L_l} & -q_k^{2,L_l} & \dots & 0 \end{bmatrix}_{L_l \times L_l}, \quad (5.17)$$

onde:

$$p_k^j = \frac{[f_k^j]^2 - v_{k-1}^j}{k}, \quad j = 1, \dots, L_l;$$

$$\left\{ \begin{array}{ll} q_k^{j,r} = 0, & j, r = 1, \dots, L_l; \text{ quando } j = r; \\ q_k^{j,r} = \frac{f_k^j f_k^r}{k [p_k^r - p_k^j + v_{k-1}^r - v_{k-1}^j]}, & j, r = 1, \dots, L_l; \text{ quando } j \neq r. \end{array} \right. \quad (5.18)$$

sendo que p_k^j são os elementos da diagonal da matriz de perturbação \mathbf{P}_k^l , f_k^j são os elementos do vetor \mathbf{F}_k^l , $q_k^{j,r}$ são elementos de \mathbf{Q}_k^l , e v_{k-1}^j, v_{k-1}^r são os valores singulares contidos em Σ_{k-1}^l , considerando $j, r = 1, \dots, L_l$.

Com a definição de ambas matrizes de perturbação, é possível atualizar recursivamente \mathbf{U}_k^l e Σ_k^l conforme a seguir (PEDDANENI et al., 2004):

$$\begin{aligned} \mathbf{U}_k^l &= \mathbf{U}_{k-1}^l (\mathbf{I} + \mathbf{Q}_k^l) \\ \Sigma_k^l &= \Sigma_{k-1}^l + \mathbf{P}_k^l. \end{aligned} \quad (5.19)$$

onde \mathbf{I} é a matriz identidade.

Utilizando \mathbf{D}_k^l e cada autovetor contido em \mathbf{U}_k^l , as componentes não-observáveis podem ser determinadas. Ressalta-se que a condição $L_l = n_l$ foi aplicada, ou seja, cada autovalor e autovetor estão relacionados a uma componente específica da série temporal l . Logo, para obter as componentes a cada instante k a partir de y_k^l , são calculados n_l vetores \mathbf{g}_k^j ($j = 1, \dots, n_l$) como segue:

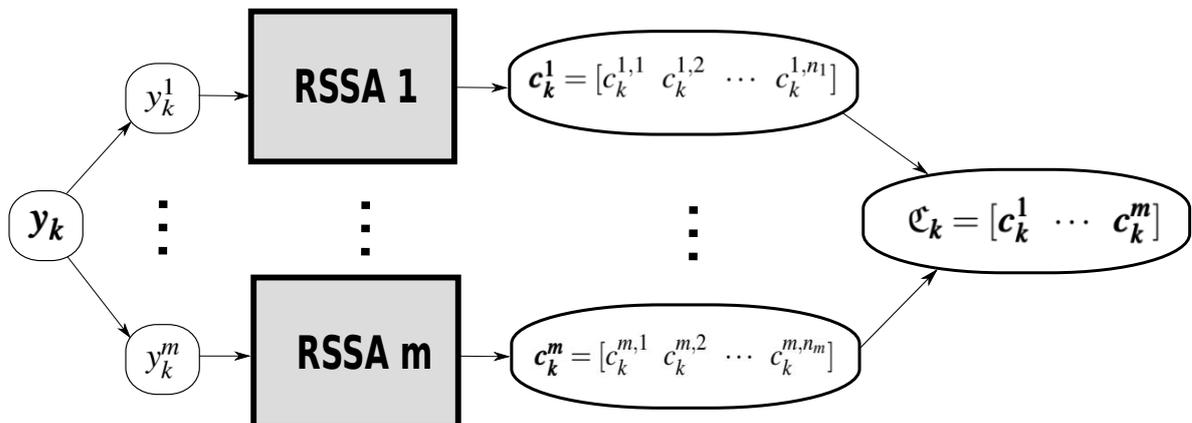
$$\begin{aligned} \mathbf{g}_k^j &= \mathbf{u}_k^j [\mathbf{D}_k^l]^\top \mathbf{u}_k^j, \quad j = 1, \dots, L_l = n_l, \\ \mathbf{g}_k^j &\in \mathbb{R}^{L_l \times 1}, \quad \mathbf{u}_k^j \in \mathbb{R}^{L_l \times 1}, \quad \mathbf{D}_k^l \in \mathbb{R}^{L_l \times 1}, \end{aligned} \quad (5.20)$$

sendo que \mathbf{u}_k^j é o autovetor j ou j -ésima coluna de \mathbf{U}_k^l . Dessa forma, as n_l componentes da l -ésima série temporal são:

$$\begin{aligned} \mathbf{c}_k^l &= [c_k^{l,1} \quad c_k^{l,2} \quad \dots \quad c_k^{l,n_l}] = [\tilde{g}^1 \quad \tilde{g}^2 \quad \dots \quad \tilde{g}^{n_l}], \\ &\text{para } j = 1, \dots, n_l, \text{ e } L_l = n_l, \end{aligned} \quad (5.21)$$

onde \tilde{g}^j indica o último elemento de cada vetor \mathbf{g}_k^j ($j = 1, \dots, n_l$), pois somente esse elemento está relacionado aos dados no instante k . Conforme apresentado, a extração de componentes é realizada para cada série temporal $l = 1, \dots, m$, formulando assim o vetor n -dimensional \mathbf{c}_k de componentes (Equação 5.3). A Figura 19 destaca a extração das componentes na etapa de treinamento incremental para $k = W + 1, W + 2, \dots, T$.

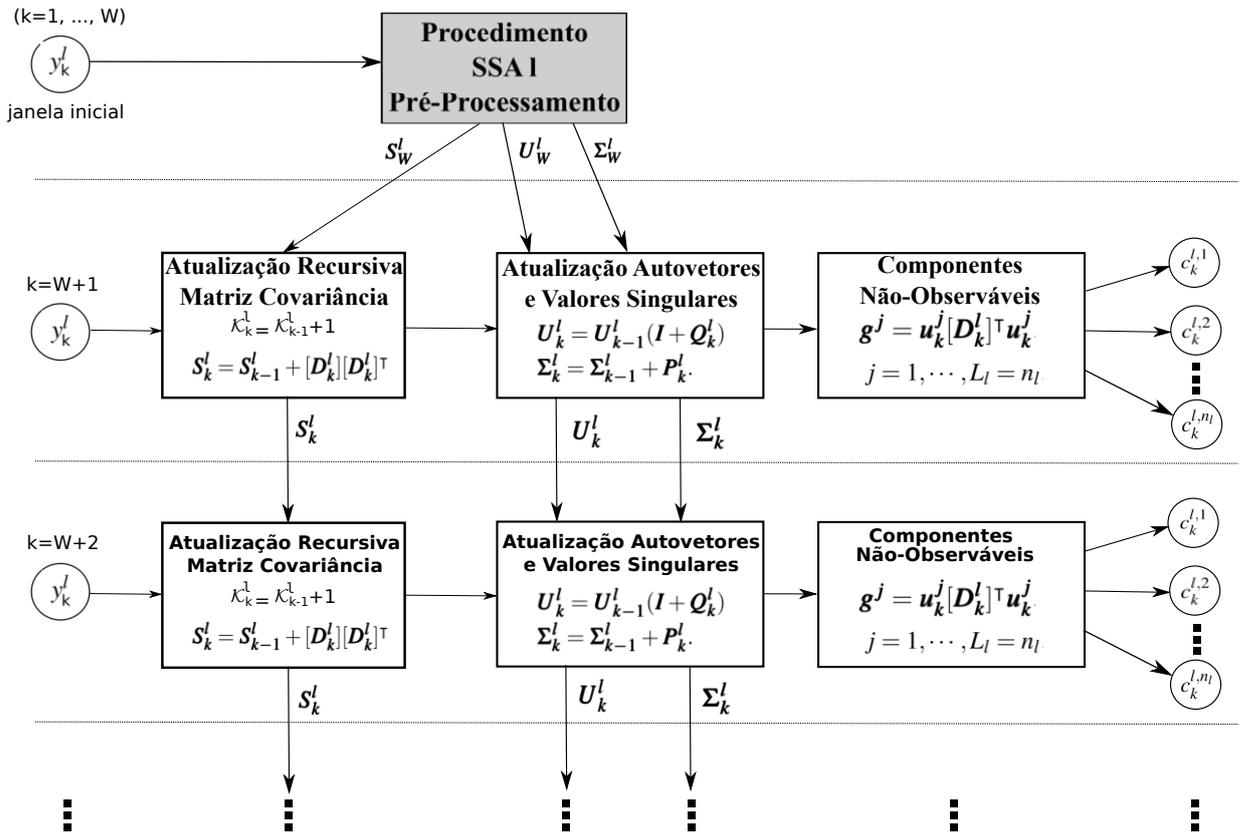
Figura 19 – Extração das Componentes Não-Observáveis na Etapa de Treinamento Incremental. Cada Série Temporal ($l = 1, \dots, m$) possui um Bloco Exclusivo do SSA Recursivo ou RSSA (Característica Paralela do PRSSA).



Fonte: Autor.

Por sua vez, a Figura 20 mostra o método PRSSA mais detalhadamente, exibindo a relação entre a extração das componentes não-observáveis iniciais efetuada na etapa de pré-processamento e o procedimento recursivo da etapa de treinamento incremental para cada nova observação da série temporal l .

Figura 20 – Relação dos Procedimentos da Etapa de Pré-Processamento e de Treinamento Incremental do PRSSA para a Série Temporal l .



Fonte: Autor.

5.3.2 Mapeamento das Entradas e Saídas da ENFN

Antes de seguir com o processo de aprendizagem da ENFN, é necessário definir o mapeamento entre as entradas e saídas. Nesse caso, o mapeamento é realizado entre o vetor de componentes no instante $k - p$ para o instante k , onde p é o passo definido anteriormente pelo usuário no pré-processamento. Dessa maneira, as entradas da ENFN são representadas pelo vetor de componentes padronizado $\hat{\mathbf{C}}_{k-p}$, sendo que a ENFN visa calcular um resultado que se aproxime de $\hat{\mathbf{C}}_k$. Destaca-se então que o mapeamento feito pela ENFN é MIMO, pois as entradas e saídas são as várias componentes nos instantes $k - p$ e k respectivamente, mesmo que a série temporal seja univariável. Devido a essa característica MIMO, a metodologia proposta PRSSA+ENFN apresenta flexibilidade para lidar com m séries temporais, para $m \geq 1$. A estrutura MIMO do mapeamento entre entrada e saída é definida por um modelo VAR de ordem n como

segue:

$$\hat{\mathcal{C}}_k^T = \mathbf{A}_k \hat{\mathcal{C}}_{k-p}^T + \mathbf{B}_k \hat{\mathcal{C}}_{k-s}^T \quad (5.22)$$

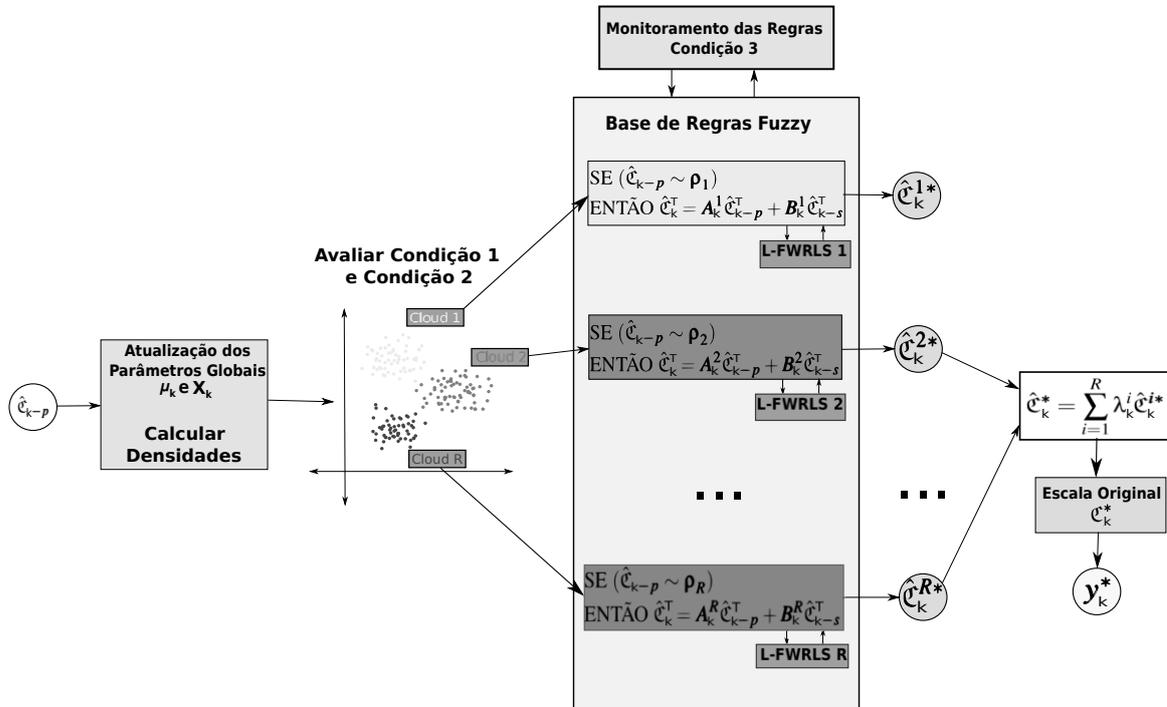
sendo $\mathbf{A}_k \in \mathbb{R}^{n \times n}$, $\mathbf{B}_k \in \mathbb{R}^{n \times n}$. Pode-se perceber que essa estrutura multivariável se adequa ao valor de n ao modificar as dimensões dessas matrizes, apresentando a flexibilidade mencionada acima.

Nota-se ainda que o vetor de componentes padronizado $\hat{\mathcal{C}}_{k-s}$ também é aplicado na estrutura da ENFN, trabalhando como uma contribuição relevante relacionada ao instante $k - s_j$ ($j = 1, \dots, n$) que tem comportamento similar ao instante k que se deseja prever, sendo que os valores de s são determinados com base na função de autocorrelação amostral.

5.3.3 Aprendizagem Evolutiva da ENFN Baseada no Método ALMMo-1

A metodologia de previsão proposta apresenta uma rede *neuro-fuzzy* ENFN que tem o papel de prever as componentes não-observáveis, com base no método ALMMo-1, proposto em Angelov, Gu e Principe (2017). A Figura 21 exibe a estrutura geral da ENFN na etapa de treinamento incremental. Ao longo do texto, serão explicados cada passo apresentado nessa figura.

Figura 21 – Estrutura Geral da ENFN na Etapa de Treinamento.



Fonte: Autor.

Como foi mencionado anteriormente, o vetor de componentes \mathcal{C}_k dever ser padronizado para proceder com o treinamento da ENFN. Sendo assim, atualiza-se recursivamente para cada k ($k = W + 1, W + 2, \dots, T$) a média e o desvio padrão a partir das suas condições iniciais na

Equação 5.7 no pré-processamento. Logo, a atualização recursiva para efetuar a padronização de \mathfrak{C}_k é definida por:

$$\begin{aligned}\bar{\mathfrak{C}}_k &= \frac{k-1}{k}\bar{\mathfrak{C}}_{k-1} + \frac{1}{k}\mathfrak{C}_k, \quad \bar{\mathfrak{C}}_k \in \mathbb{R}^{1 \times n}, \\ \boldsymbol{\sigma}_k &= \sqrt{\frac{1}{k-1} \left((k-2)[\boldsymbol{\sigma}_{k-1}]^2 + \frac{k}{k-1}(\bar{\mathfrak{C}}_k - \mathfrak{C}_k)^2 \right)}, \quad \boldsymbol{\sigma}_k \in \mathbb{R}^{1 \times n}, \\ \hat{\mathfrak{C}}_k &= \frac{\mathfrak{C}_k - \bar{\mathfrak{C}}_k}{\boldsymbol{\sigma}_k}, \quad \text{para } k = W+1, W+2, \dots, T.\end{aligned}\tag{5.23}$$

Tendo como base o método ALMMo-1 e considerando as entradas da ENFN $\hat{\mathfrak{C}}_{k-p}$, a média global $\boldsymbol{\mu}_k$ e o quadrado da norma Euclidiana X_k (Parte 3 da Tabela 3) podem ser atualizados usando as seguintes equações:

$$\begin{aligned}\boldsymbol{\mu}_k &= \frac{k-1}{k}\boldsymbol{\mu}_{k-1} + \frac{1}{k}\hat{\mathfrak{C}}_{k-p}, \quad \boldsymbol{\mu}_k \in \mathbb{R}^{1 \times n}, \\ X_k &= \frac{k-1}{k}X_{k-1} + \frac{1}{k}\|\hat{\mathfrak{C}}_{k-p}\|^2, \quad \text{para } k = W+1, W+2, \dots, T.\end{aligned}\tag{5.24}$$

Por conseguinte, três tipos de medidas de densidade de dados são examinados no procedimento evolutivo dos *data clouds*:

- Densidade Discreta Unimodal Global de $\hat{\mathfrak{C}}_{k-p}$:

$$D_{global} = \frac{1}{1 + \frac{\|\hat{\mathfrak{C}}_{k-p} - \boldsymbol{\mu}_k\|^2}{X_k - \|\boldsymbol{\mu}_k\|^2}};\tag{5.25}$$

- Densidade Discreta Unimodal do centro $\boldsymbol{\rho}_i$ do *data cloud* i ($i = 1, \dots, R$):

$$D_{centro}^i = \frac{1}{1 + \frac{\|\boldsymbol{\rho}_i - \boldsymbol{\mu}_k\|^2}{X_k - \|\boldsymbol{\mu}_k\|^2}};\tag{5.26}$$

- Densidade Discreta Unimodal Local calculada para cada *data cloud* i ($i = 1, \dots, R$):

$$D_{local}^i = \frac{1}{1 + \frac{[S_i]^2 \|\hat{\mathfrak{C}}_{k-p} - \boldsymbol{\mu}_k\|^2}{(S_i + 1)(S_i q_i + \|\hat{\mathfrak{C}}_{k-p}\|^2) - \|\hat{\mathfrak{C}}_{k-p} + S_i \boldsymbol{\rho}_i\|^2}},\tag{5.27}$$

onde S_i é o suporte do *data cloud* i e q_i é o produto escalar local do *cloud* i , ambos apresentados na Parte 4 da Tabela 3.

Essas medidas de densidade são inspiradas no processo de aprendizagem do ALMMo-1 e são aplicadas na avaliação das condições para atualização da estrutura da ENFN. Em primeiro lugar, a Condição 1 a seguir é analisada, verificando se o ponto $\hat{\mathbf{c}}_{k-p}$ no espaço n -dimensional tem o potencial para representar o centro de um novo *data cloud*:

$$\begin{aligned} \text{Condição 1: } & \text{SE} \left(D_{global} > \max_{i=1, \dots, R} [D_{centro}^i] \right) \\ & \text{OU} \left(D_{global} < \min_{i=1, \dots, R} [D_{centro}^i] \right) \\ & \text{ENTÃO } \hat{\mathbf{c}}_{k-p} \text{ é um candidato a ser o centro de um novo } data \text{ cloud} \end{aligned} \quad (5.28)$$

Essa condição aponta que se D_{global} é maior que o máximo D_{centro}^i dos R *data clouds*, significa que $\hat{\mathbf{c}}_{k-p}$ é representativo o suficiente para ser um novo centro. Em contrapartida, se D_{global} é menor que o mínimo D_{centro}^i , $\hat{\mathbf{c}}_{k-p}$ passa a indicar um novo padrão nos dados que é diferente de todos os outros que foram identificados anteriormente (ANGELOV; GU; PRINCIPE, 2017). Todavia, somente essa condição não é suficiente para efetivamente adicionar esse candidato a *cloud* à estrutura da ENFN.

Quando a Condição 1 é satisfeita, a Condição 2 a seguir é avaliada para limitar a sobreposição excessiva entre *data clouds*, cujas áreas de influência podem estar bem próximas ao ponto candidato identificado na Condição 1. Portanto, a Condição 2 tem o papel de atenuar as possíveis redundâncias que podem ocorrer na definição dos *data clouds* ou *regras fuzzy*, e é definida conforme segue:

$$\begin{aligned} \text{Condição 2: } & \text{SE} \left(D_{local}^i \geq \frac{1}{1 + \phi^2} \right) \\ & \text{ENTÃO } \hat{\mathbf{c}}_{k-p} \text{ está muito próximo de } \mathbf{p}_i \end{aligned} \quad (5.29)$$

onde ϕ é o fator de sobreposição definido na etapa de pré-processamento, um dos parâmetros críticos do PRSSA+ENFN (Parte 2 da Tabela 3). Logo, o parâmetro ϕ está relacionado ao nível de sobreposição entre *clouds* que é permitido. Se ambas as Condições 1 (Equação 5.28) e 2 (Equação 5.29) forem verdadeiras, infere-se que há uma sobreposição acima do permitido com a inserção desse candidato como *cloud* no espaço n -dimensional. Quando isso ocorre, o *data cloud* mais próximo ao candidato identificado na Condição 1 é indicado por:

$$i^* = \operatorname{argmin}(\|\hat{\mathbf{c}}_{k-p} - \mathbf{p}_i\|), \quad i = 1, \dots, R. \quad (5.30)$$

Nesse caso, não é necessário adicionar este novo *data cloud* ao espaço, apenas atualiza-se

os parâmetros do *cloud* i^* mais próximo como segue:

$$\begin{aligned}\mathbf{p}_{i^*} &= \frac{\mathbf{p}_{i^*} + \hat{\mathbf{c}}_{k-p}}{2}, \text{ (centro do } cloud \text{ mais próximo)} \\ q_{i^*} &= \frac{q_{i^*} + \|\hat{\mathbf{c}}_{k-p}\|^2}{2}, \text{ (produto escalar local do } cloud \text{ mais próximo)} \\ S_{i^*} &= \text{round}\left(\frac{S_{i^*} + 1}{2}\right), \text{ (suporte do } cloud \text{ mais próximo)}.\end{aligned}\tag{5.31}$$

Entretanto, se somente a Condição 1 for verdadeira, o ponto candidato é efetivamente adicionado como *data cloud* ao espaço n -dimensional e à estrutura da ENFN, como consequência, $R = R + 1$. Então, uma nova regra *fuzzy* associada a esse *cloud* é gerada e seus parâmetros locais (Parte 4 da Tabela 3) são inicializados da mesma forma que na etapa de pré-processamento.

Mas, se a Condição 1 não for verdadeira, $\hat{\mathbf{c}}_{k-p}$ será um novo ponto atribuído ao *data cloud* i^* mais próximo, cujos parâmetros desse *cloud* são atualizados como segue:

$$\begin{aligned}S_{i^*} &= S_{i^*} + 1, \\ \mathbf{p}_{i^*} &= \frac{S_{i^*} - 1}{S_{i^*}} \mathbf{p}_{i^*} + \frac{1}{S_{i^*}} \hat{\mathbf{c}}_{k-p}, \\ q_{i^*} &= \frac{S_{i^*} - 1}{S_{i^*}} q_{i^*} + \frac{1}{S_{i^*}} \|\hat{\mathbf{c}}_{k-p}\|^2.\end{aligned}\tag{5.32}$$

A Figura 22 exibe por meio de um fluxograma a aplicação conjunta das condições 1 e 2 para o processo de aprendizagem da ENFN.

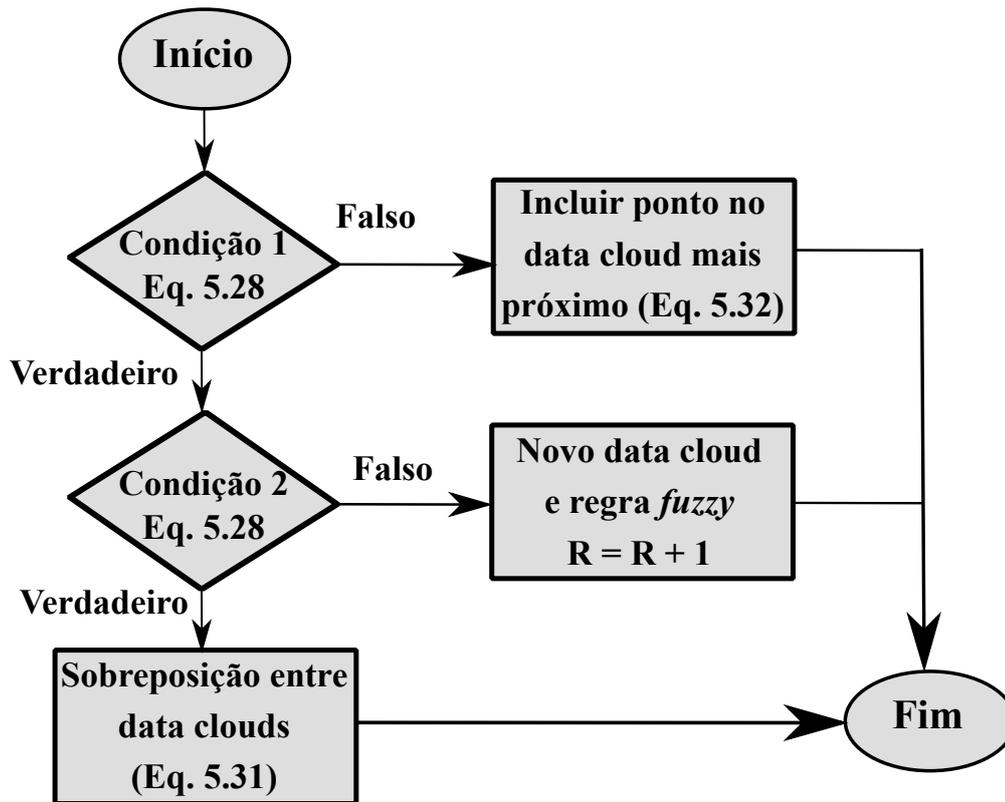
O próximo passo é calcular a contribuição λ_k^i das regra *fuzzy* i na saída global da ENFN a cada instante, também conhecida como grau de ativação normalizado, pois apresenta valor entre 0 e 1. Para $i = 1, \dots, R$, λ_k^i é definido como segue:

$$\lambda_k^i = \frac{D_{local}^i}{\sum_{i=1}^R D_{local}^i}\tag{5.33}$$

sendo que D_{local}^i foi apresentado na Equação 5.27.

Segundo Angelov, Gu e Principe (2017), algumas regras *fuzzy* ou *clouds* podem perder desempenho ou utilidade ao longo do tempo, contribuindo muito pouco para a saída ou mesmo influenciando negativamente na resposta, pois podem estar associados a *outliers* presentes nos dados. Dessa forma, para simplificar o modelo, é interessante remover esses *clouds*. Logo, a

Figura 22 – Fluxograma para Aplicação das Condições 1 e 2.



Fonte: Autor.

medida de utilidade η_i do *cloud* i pode ser calculada para cada instante k como segue:

$$\eta_i = \frac{1}{k - \tau_i} \Lambda_i, \quad i = 1, \dots, R. \quad (5.34)$$

$$\Lambda_i = \sum_{r=\tau_i}^k \lambda_r^i$$

onde τ_i é o instante k em que o *data cloud* ou regra *fuzzy* i foi gerado, e Λ_i corresponde ao grau de ativação acumulado da i -ésima regra *fuzzy* desde o momento em que foi criada. Aquelas regras *fuzzy* com baixa utilidade podem ser removidas para simplificar a estrutura da ENFN, seguindo a Condição 3:

$$\begin{aligned} \text{Condição 3: } & \text{SE } (\eta_i < \eta_0) \\ & \text{ENTÃO a regra } \textit{fuzzy} \textit{ } i \text{ deve ser removida } (R = R - 1) \end{aligned} \quad (5.35)$$

para η_0 sendo fator de esquecimento definido na etapa de pré-processamento, outro parâmetro crítico do método PRSSA+ENFN (Parte 2 da Tabela 3). Quanto maior for o parâmetro η_0 , mais chance das regras *fuzzy* com baixa utilidade serem removidas.

Nessa tese, o formato da regra *fuzzy* completa do PRSSA+ENFN pode ser visualizado na

Figura 21 e apresenta a seguinte composição:

$$\begin{aligned} \text{Regra } Fuzzy_i: \quad & \text{SE } (\hat{\mathbf{C}}_{k-p} \sim \boldsymbol{\rho}_i) \\ & \text{ENTÃO } \hat{\mathbf{C}}_k^\top = \mathbf{A}_k^i \hat{\mathbf{C}}_{k-p}^\top + \mathbf{B}_k^i \hat{\mathbf{C}}_{k-s}^\top, \end{aligned} \quad (5.36)$$

para $\mathbf{A}_k^i \in \mathbb{R}^{n \times n}$, $\mathbf{B}_k^i \in \mathbb{R}^{n \times n}$, sendo que a proposição antecedente é composta pelo vetor de entrada $\hat{\mathbf{C}}_{k-p}$ e o centro $\boldsymbol{\rho}_i$ do *cloud* i , e a proposição consequente são modelos de vetores autorregressivos (VAR) de ordem n . De acordo com a representação da regra *fuzzy* adotada, as componentes não-observáveis extraídas são tratadas como os regressores e esses modelos representam o comportamento dinâmico dessas componentes. Além disso, note que a base de regras *fuzzy* é preparada para mapear os dados das componentes do instante $k-p$ para k , considerando a contribuição do instante $k-s_j$, em que o parâmetro s_j foi estimado na etapa de pré-processamento para cada componente ($j = 1, \dots, n$). Essa forma de representação aplicada é apropriada para modelar de forma multivariável (MIMO) as relações existentes entre as componentes de todas as m séries temporais, através das matrizes \mathbf{A}_k^i e \mathbf{B}_k^i . Outro fator importante é a flexibilidade para representação por meio desses modelos devido a modificação da quantidade de componentes e séries temporais.

O método dos Mínimos Quadrados Recursivos Ponderados *Fuzzy* Local, do inglês *Local Fuzzy Weighted Recursive Least Squares* (L-FWRLS) (ANGELOV, 2010), é responsável nesta metodologia por estimar as matrizes \mathbf{A}_k^i e \mathbf{B}_k^i de cada regra i , considerando como fator de ponderação o grau de ativação normalizado λ_k^i (Equação 5.33). Dessa forma, aplica-se o procedimento matemático a seguir:

$$\begin{aligned} \hat{\mathbf{C}}_k^\top &= \mathbf{A}_k^i \hat{\mathbf{C}}_{k-p}^\top + \mathbf{B}_k^i \hat{\mathbf{C}}_{k-s}^\top, \\ \hat{\mathbf{C}}_k^\top &= \begin{bmatrix} \mathbf{A}_k^i & \mathbf{B}_k^i \end{bmatrix} \begin{bmatrix} \hat{\mathbf{C}}_{k-p}^\top \\ \hat{\mathbf{C}}_{k-s}^\top \end{bmatrix}, \end{aligned} \quad (5.37)$$

$$\hat{\mathbf{C}}_k^\top = (\boldsymbol{\Delta}_k^i)^\top \mathbf{v}^\top.$$

para $\boldsymbol{\Delta}_k^i \in \mathbb{R}^{2n \times n}$ sendo a matriz que contém os elementos de \mathbf{A}_k^i e \mathbf{B}_k^i a serem estimados e $\mathbf{v} \in \mathbb{R}^{1 \times 2n}$ o vetor que considera os dados das componentes padronizadas. Portanto, a estimação recursiva para cada regra *fuzzy* $i = 1, 2, \dots, R$ é dada pela expressão:

$$\begin{aligned} \mathbf{Z}_k^i &= \frac{\mathbf{J}_{k-1}^i \mathbf{v}^\top}{\mathbf{v} \lambda_k^i \mathbf{J}_{k-1}^i \mathbf{v}^\top + 1}, \quad \mathbf{Z}_k^i \in \mathbb{R}^{2n \times 1}, \\ (\boldsymbol{\Delta}_k^i)^\top &= (\boldsymbol{\Delta}_{k-1}^i)^\top + \lambda_k^i \mathbf{Z}_k^i (\hat{\mathbf{C}}_k - \mathbf{v} \boldsymbol{\Delta}_{k-1}^i), \quad \boldsymbol{\Delta}_k^i \in \mathbb{R}^{2n \times n}, \end{aligned} \quad (5.38)$$

$$\mathbf{J}_k^i = \mathbf{J}_{k-1}^i - \lambda_k^i \mathbf{Z}_k^i \mathbf{v} (\mathbf{J}_{k-1}^i)^\top, \quad \mathbf{J}_k^i \in \mathbb{R}^{2n \times 2n},$$

onde \mathbf{Z}_k^i é a matriz de ganho de cada regra e \mathbf{J}_k^i é a matriz covariância da i -ésima regra *fuzzy*. Note que o vetor de componentes extraídas pelo PRSSA no instante k , denotado por $\hat{\mathbf{C}}_k$, é aplicado para estimação das matrizes \mathbf{A}_k^i e \mathbf{B}_k^i no treinamento.

A saída ou resposta local de cada regra *fuzzy* i ($i = 1, \dots, R$) é dada por:

$$[\hat{\mathbf{c}}_k^{i*}]^T = \mathbf{A}_k^i \hat{\mathbf{c}}_{k-p}^T + \mathbf{B}_k^i \hat{\mathbf{c}}_{k-s}^T, \quad (5.39)$$

sendo que $\hat{\mathbf{c}}_k^{i*}$ representa a saída da regra *fuzzy* i no instante k na etapa de treinamento. Calculando todas as saídas de cada regra em R , a saída ou resultado global da PRSSA+ENFN é:

$$\hat{\mathbf{c}}_k^* = \sum_{i=1}^R \lambda_k^i \hat{\mathbf{c}}_k^{i*}, \quad (5.40)$$

para $\hat{\mathbf{c}}_k^*$ sendo o vetor de componentes calculado pelo método proposto. É importante enfatizar que os termos $\hat{\mathbf{c}}_k$ e $\hat{\mathbf{c}}_k^*$ são diferentes. Enquanto o primeiro vetor $\hat{\mathbf{c}}_k$ corresponde aos dados das componentes padronizadas extraídas pelo PRSSA, o vetor $\hat{\mathbf{c}}_k^*$ indica os resultados das componentes padronizadas calculadas pela ENFN.

Para retornar o vetor de componentes padronizadas $\hat{\mathbf{c}}_k^*$ para a escala original, a seguinte transformação é aplicada:

$$\mathbf{c}_k^* = \hat{\mathbf{c}}_k^* \boldsymbol{\sigma}_k + \bar{\mathbf{c}}_k \quad (5.41)$$

onde \mathbf{c}_k^* é o vetor de componentes calculado pelo PRSSA+ENFN. Por fim, cada série temporal l é reconstruída com base nesse vetor de componentes:

$$\begin{aligned} y_k^{l*} &= \sum_{j=1}^{n_l} c_k^{l,j*}, \\ \mathbf{y}_k^* &= [y_k^{1*} \cdots y_k^{m*}], \\ \text{tal que :} \\ \mathbf{c}_k^* &= [c_k^{1*} \cdots c_k^{m*}] \text{ e} \\ c_k^{l*} &= [c_k^{l,1*} \ c_k^{l,2*} \ \cdots \ c_k^{l,n_l*}], \ l = 1, \dots, m, \end{aligned} \quad (5.42)$$

sendo que y_k^{l*} é a l -ésima série temporal calculada e \mathbf{y}_k^* é o conjunto de resultados do método proposto para as m séries no instante k . Destaca-se novamente que o asterisco $*$ representa as observações no instante k calculadas pelo método PRSSA+ENFN. O Algoritmo A.3 resume o procedimento realizado na etapa de treinamento.

5.4 Etapa de Previsão

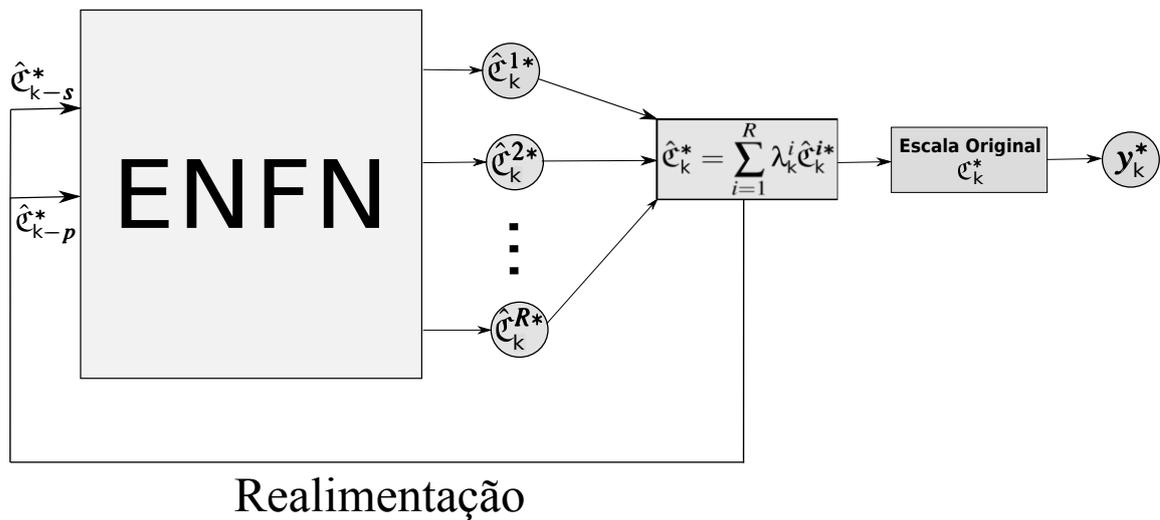
Na etapa de previsão, o método PRSSA+ENFN realiza a previsão da série temporal nos instantes $k = T + 1, \dots, T + H$, onde H é o horizonte de previsão definido no pré-processamento, T é o número de dados na etapa de treinamento incremental, e $N = T + H$, sendo que N é a quantidade total de dados. Nesse caso, o método proposto tenta obter dados em instantes futuros, sendo que a avaliação do desempenho do PRSSA+ENFN deve ser efetuada nessa etapa. O método proposto desenvolveu e evoluiu sua estrutura no treinamento incremental, mas nessa etapa de previsão, ela não mais evolui, ou seja, a quantidade de regras *fuzzy* não se modifica. Esse critério é utilizado para verificar se o treinamento evolutivo teve sucesso para aprender

sobre a dinâmica dos dados. Dependendo da disponibilidade das entradas do PRSSA+ENFN, esse método pode prever as séries temporais de duas formas, que serão descritas nas subseções a seguir.

5.4.1 Previsão Recorrente do PRSSA+ENFN

O PRSSA+ENFN realiza uma previsão recorrente quando não há dados disponíveis da série temporal para extrair as componentes não-observáveis ao longo do horizonte de previsão, ou seja, o vetor de componentes $\hat{\mathbf{c}}_{k-p}^*$, que são as entradas da ENFN, e o vetor $\hat{\mathbf{c}}_{k-s}^*$ não podem ser obtidos. Logo, a previsão recorrente é realizada nesse caso, sendo que as componentes previstas pelo PRSSA+ENFN são levadas em conta para prever os próximos valores. Em outras palavras, os próprios resultados de previsão das componentes são realimentados como entradas da ENFN, conforme Figura 23.

Figura 23 – Previsão Recorrente do Método PRSSA+ENFN.



Fonte: Autor.

Nessa situação, o primeiro passo é calcular o grau de ativação λ_k^i da i -ésima regra *fuzzy*, para $i = 1, \dots, R$. Então, as Equações 5.27 e 5.33 são computadas. Em seguida, a base de regras *fuzzy* é avaliada para o cálculo dos resultados de previsão das componentes, de acordo com a Equação 5.36. De forma similar às Equações 5.39 e 5.40, a previsão local e global das componentes são, respectivamente:

$$[\hat{\mathbf{c}}_k^{i*}]^T = \mathbf{A}_k^i [\hat{\mathbf{c}}_{k-p}^*]^T + \mathbf{B}_k^i [\hat{\mathbf{c}}_{k-s}^*]^T, \tag{5.43}$$

$$\hat{\mathbf{c}}_k^* = \sum_{i=1}^R \lambda_k^i \hat{\mathbf{c}}_k^{i*},$$

onde $\hat{\mathbf{c}}_{k-p}^*$ e $\hat{\mathbf{c}}_{k-s}^*$ são calculados pela PRSSA+ENFN e ambos são considerados ao longo do horizonte de previsão para calcular os valores de $\hat{\mathbf{c}}_k^*$ a cada instante k . O diferencial da expressão

acima para as Equações 5.39 e 5.40 é que os termos $\hat{\mathcal{C}}_{k-p}^*$ e $\hat{\mathcal{C}}_{k-s}^*$, resultantes do PRSSA+ENFN, são aplicados para calcular $\hat{\mathcal{C}}_k^*$, o que está relacionado a realimentação e recorrência da previsão.

Após isso, $\hat{\mathcal{C}}_k^*$ é colocado na escala original na forma \mathcal{C}_k^* (Equação 5.41) e os resultados de previsão das séries temporais são:

$$y_k^{l*} = \sum_{j=1}^{n_l} c_k^{l,j*}, \quad (5.44)$$

$$\mathbf{y}_k^* = [y_k^{1*} \cdots y_k^{m*}].$$

Na previsão recorrente, geralmente $p = 1$, ou seja, o método proposto realiza várias previsões de um passo a frente até alcançar o horizonte de previsão H .

5.4.2 Previsão Direta pela PRSSA+ENFN

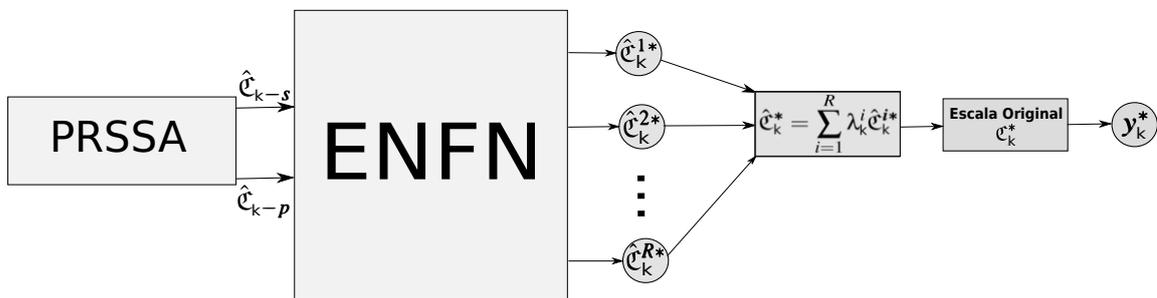
Em contrapartida, as observações da série temporal estão disponíveis nesse caso. Portanto, é possível extrair as componentes não-observáveis \mathcal{C}_{k-p} e \mathcal{C}_{k-s} por meio do PRSSA, como se observa na Figura 24. O procedimento de extração e padronização do vetor de componentes é realizado normalmente como no treinamento. O grau de ativação λ_k^i da i -ésima regra fuzzy é computado ($i = 1, \dots, R$). Dessa forma, a previsão das componentes é indicada por:

$$[\hat{\mathcal{C}}_k^{i*}]^T = \mathbf{A}_k^i \hat{\mathcal{C}}_{k-p}^T + \mathbf{B}_k^i \hat{\mathcal{C}}_{k-s}^T, \quad (5.45)$$

$$\hat{\mathcal{C}}_k^* = \sum_{i=1}^R \lambda_k^i \hat{\mathcal{C}}_k^{i*},$$

e a previsão da série temporal completa é efetuada pela Equação 5.44. Note que $\hat{\mathcal{C}}_{k-p}$ e $\hat{\mathcal{C}}_{k-s}$ são compostos por dados extraídos pelo método PRSSA, diferente da Equação 5.43 da previsão recorrente.

Figura 24 – Previsão Direta do Método PRSSA+ENFN.



Fonte: Autor.

Na previsão direta, os usuários definem como será o mapeamento desejado entre $k - p$ e k de acordo com o objetivo, definindo um valor para o passo p . Logo, o PRSSA+ENFN será preparado na etapa de treinamento para realizar esse mapeamento ao longo do horizonte de previsão H . O Algoritmo A.4 exhibe o procedimento realizado em ambos os casos na etapa de previsão.

5.5 Estratégias para Seleção dos Parâmetros Críticos do PRSSA+ENFN

Conforme mencionado anteriormente, a determinação dos parâmetros críticos (Parte 2 da Tabela 3) do PRSSA+ENFN é importante e influencia nos resultados de previsão. A seguir, duas estratégias são apresentadas nas próximas subseções e podem ser aplicadas pelo usuário do método PRSSA+ENFN a fim de selecionar esses parâmetros.

5.5.1 Aplicação de Algoritmo Genético

Com a finalidade de selecionar valores apropriados para os parâmetros críticos do método PRSSA+ENFN, os algoritmos genéticos (GOLDBERG, 1989) foram considerados como uma das alternativas nesse trabalho. O cromossomo ou solução nesse caso são compostos por esses parâmetros do PRSSA+ENFN (Parte 2 da Tabela 3), apresentando a estrutura como segue:

$$\text{cromossomo} = [W \ n_1 \ \cdots \ n_m \ \phi \ \eta_0] \in \mathbb{R}^{1 \times (3+m)}. \quad (5.46)$$

sendo que sua dimensão depende da quantidade de séries temporais m no problema.

A população é inicializada com um número definido de soluções aleatórias ou cromossomos conforme equação acima. A função custo ou *fitness* pode ser um dos critérios de erro de previsão apresentados na Seção 2.3.4, sendo que os cromossomos com menor valor considerando o critério de erro são melhores soluções, por exemplo, o RMSE, MAPE, MSE, MAE, etc.

A Seleção por Roleta Viciada é aplicada para escolha dos pais, isto é, a seleção de um par de cromossomos que combinados irão gerar os cromossomos filhos. Aqueles cromossomos com maior qualidade possuem mais chance de serem escolhidos como pais. Além disso, o operador de cruzamento ou *crossover* do tipo uniforme entre os cromossomos é considerado nesse método. A mutação também pode ser efetuada com uma certa probabilidade a cada nova geração, onde um cromossomo aleatoriamente determinado é inserido na população.

Finalmente, este algoritmo genético possui os seguintes critérios de parada:

- quando um número determinado de gerações foi criado, ou;
- quando um valor pré-definido foi alcançado pela função custo.

Essa forma para seleção dos parâmetros deve ser aplicada quando o usuário do método PRSSA+ENFN busca ter um resultado de previsão que tenha as menores métricas de erro o possível, dependendo da execução do algoritmo genético. Ressalta-se que diversas outras abordagens de busca podem ser aplicadas para selecionar soluções a esses parâmetros críticos, como por exemplo, o método *Particle Swarm Optimization* (PSO).

5.5.2 Seleção Aleatória dos Parâmetros Críticos da PRSSA+ENFN

Nessa alternativa, o PRSSA+ENFN é executado várias vezes, obtendo um conjunto de possíveis resultados de previsão. Os parâmetros críticos do PRSSA+ENFN (Parte 2 da Tabela 3) são selecionados aleatoriamente em um intervalo de valores, como segue:

- $W = W_{min}, \dots, W_{max}$;
- $n_l = 2, \dots, 10$, para $l = 1, \dots, m$;
- $0.1 \leq \phi \leq 0.5$;
- $0.01 \leq \eta_0 \leq 0.3$.

Como resultado de previsão, um intervalo de resultados possíveis para as séries temporais é obtido, conforme definido na Seção 2.3.5. Essa forma para seleção dos parâmetros deve ser aplicada quando, por meio do PRSSA+ENFN, busca-se fazer uma projeção de dados futuros sendo que não é possível verificar se a previsão se aproximou dos dados reais, pois eles não estão ainda disponíveis. Logo, apresenta-se um intervalo de previsão como possibilidade de resultados ao usuário.

O Algoritmo A.5 resume ambas alternativas para seleção de parâmetros do PRSSA+ENFN.

5.6 Análise da Complexidade Computacional do PRSSA+ENFN

Quando se tenta caracterizar a eficiência de um algoritmo em termos do tempo de execução independente de qualquer programa ou sistema computacional particular, é importante analisar suas operações. Uma das formas para realizar essa análise é a função $O(\cdot)$, conhecida como *Big O Notation*, que é uma notação assintótica para representar a complexidade computacional do algoritmo. Como essa notação é assintótica, geralmente descreve a execução do algoritmo no pior caso para o número de entradas ou valor dos parâmetros. Usando essa notação $O(\cdot)$, verifica-se como o tempo de execução de um algoritmo se eleva à medida que o valor dos parâmetros ou quantidade de entradas aumenta sem limites (CORMEN et al., 2009).

Pode-se seguir as etapas abaixo para obter a notação $O(\cdot)$ de um algoritmo (CORMEN et al., 2009):

1. Dividir o algoritmo em operações individuais;
2. Calcular a notação $O(\cdot)$ de cada operação individual;
3. Somar o $O(\cdot)$ de todas operações individuais;
4. Remover termos constantes;

5. Considerar apenas o termo de ordem mais alta, pois é o mais significativo em uma análise assintótica.

À vista disso, o PRSSA+ENFN será analisado por partes, verificando a complexidade das Etapas de Pré-processamento, Treinamento e Previsão e, em seguida, somando-se as notações $O(\cdot)$ de cada etapa. A análise geral para o caso do uso de algoritmos genéticos (Seção 5.5.1) ou para seleção aleatória dos parâmetros críticos do PRSSA ENFN (Seção 5.5.2) também será efetuada.

5.6.1 Complexidade Computacional do Pré-processamento

Para a etapa de pré-processamento apresentada na Seção 5.2 e pelo Algoritmo A.2, verifica-se que o custo computacional mais significativo e de maior ordem está na extração das componentes pelo SSA para cada série temporal l ($l = 1, \dots, m$). Logo, em uma análise assintótica, a notação $O(\cdot)$ desse procedimento prevalece.

De acordo com Korobeynikov (2010), a complexidade do método SSA básico é $O(N^3)$ no pior caso, sendo N a quantidade total de dados da série temporal. Porém, no pré-processamento do PRSSA+ENFN, considera-se somente as W primeiras observações da série temporal l , sendo assim, a notação para a metodologia proposta é $O(W^3)$. Como cada série $l = 1, \dots, m$ apresenta um procedimento de extração paralela e, portanto, um laço de repetição de m vezes, a complexidade computacional da extração de componentes iniciais e da etapa de pré-processamento resulta em $O(mW^3)$.

5.6.2 Complexidade Computacional do Treinamento Incremental

O treinamento do PRSSA+ENFN é descrito na Seção 5.3 e no Algoritmo A.3. Destaca-se que a extração recursiva das componentes pelo PRSSA para a série temporal l tem custo $O(n_l^2)$, já que depende da sua quantidade de componentes não-observáveis n_l , o que afeta a ordem das matrizes que estão presentes.

Contudo, conforme apresentado na Seção 5.2.1 sobre a determinação do parâmetro crítico n_l do PRSSA+ENFN, o maior valor para n_l é n_{max} , sendo que na metodologia proposta, recomendou-se um valor não superior a $n_{max} = 10$. Dessa maneira, no pior caso, $n_l = 10$. Conclui-se que o custo do PRSSA para a série temporal l é constante em uma análise assintótica, ou seja, $O(n_l^2) = O(10^2) = O(1)$. Como são m séries temporais, a notação que leva em conta todas as m séries fica $O(m)$.

Sendo assim, verifica-se que o maior custo computacional está associado ao método L-FWRLS presente na ENFN, que baseia-se no ALMMo. A complexidade do ALMMo tradicional foi analisada no trabalho de Gu, Shen e Angelov (2020), sendo que o custo computacional mais significativo está relacionado à atualização dos consequentes pelo método L-FWRLS. No entanto, para o PRSSA+ENFN proposto, há o diferencial de atualizar as matrizes que dependem

da quantidade total de componentes n . A notação para essa etapa, portanto, é quadrática $O(n^2)$. Pode-se representar essa notação em termos de m . Sabendo que $n_l = n_{max} = 10$ no pior caso, nota-se que $n = 10m$. De forma alternativa, $O(n^2) = O(100m^2) = O(m^2)$.

Ademais, a atualização das matrizes ocorre para cada regra *fuzzy* R , sendo assim, o custo computacional nesse caso e para um ciclo ou instante k da etapa de treinamento fica $O(Rm^2)$. Considerando agora todas as observações da série temporal utilizadas no treinamento, a notação para a etapa de treinamento completa é denotada por $O((T - W)Rm^2)$, onde $T - W$ é a quantidade de amostras aplicadas exclusivamente nesse procedimento.

5.6.3 Complexidade Computacional da Previsão

Na etapa de previsão, a análise do Algoritmo A.4 deve ser realizada em duas partes distintas: para o caso da previsão recorrente e para a previsão direta. Para a previsão recorrente, o maior custo está associado à computação dos resultados de cada regra *fuzzy* (\hat{C}_k^{i*}), pois há um laço de R repetições para a Equação 5.43. Como o custo computacional de \hat{C}_k^{i*} é $O(n^2) = O(m^2)$, para todas as regras, o custo de previsão da observação em um instante torna-se $O(Rm^2)$. Considerando todo o horizonte de previsão H , o resultado fica $O(HRm^2)$.

Por sua vez, na previsão direta ocorre uma extração de componentes recursiva pelo PRSSA, sendo que o custo é de $O(m)$, como calculado na seção de treinamento. A partir disso, o procedimento é o mesmo que ocorreu para previsão recorrente, ou seja, o custo é de $O(Rm^2)$. Somando-se a extração e a previsão de uma observação, tem-se $O(m + Rm^2)$. Considerando somente os termos de maior ordem, a notação da previsão direta de uma observação é $O(Rm^2)$. Finalmente, para todo o horizonte de previsão, o custo resultante do procedimento de previsão direta é $O(HRm^2)$.

Nota-se que ambos procedimentos de previsão tiveram mesmo custo, apesar da diferença entre eles. No entanto, como a análise usando a notação $O(\cdot)$ é assintótica e para o pior caso, ambas tendem para as notações similares.

5.6.4 Complexidade Computacional Geral do PRSSA+ENFN e da Seleção dos Parâmetros Críticos

A complexidade computacional geral do PRSSA+ENFN é obtida somando-se a notação $O(\cdot)$ de cada uma das etapas desse método, como segue:

$$\begin{aligned} &O[mW^3 + (T - W)Rm^2 + HRm^2] \\ &O[mW^3 + (T - W + H)Rm^2] \\ &O[\max(mW^3, (T - W + H)Rm^2)] \end{aligned} \tag{5.47}$$

sendo que o impacto do crescimento do valor de W na etapa de pré-processamento é de ordem cúbica, devido sobretudo à aplicação do SVD do método SSA para cada uma das séries temporais. Percebe-se também uma influencia importante da quantidade de séries m , assim como

a quantidade de regras R e dos parâmetros relacionados a quantidade de observações. Essa notação acima representa o efeito do crescimento desses no custo computacional do método PRSSA+ENFN.

Ademais, tem-se ainda as duas alternativas para seleção dos parâmetros críticos do PRSSA+ENFN, que acrescentam complexidade computacional em suas respectivas operações. Em relação ao algoritmo genético, o custo computacional usando a notação $O(\cdot)$ será mais significativo no laço de avaliação do desempenho do PRSSA+ENFN (Linhas 17-20 do Algoritmo A.5). Isso porque essa avaliação é executada para todas as gerações e toda a população de cromossomos ou soluções de parâmetros, considerando que são produzidas Q_g gerações de uma população com o total de T_p cromossomos.

Logo, a notação para a execução do algoritmo genético dependerá do custo do método PRSSA+ENFN:

$$O[Q_g T_p (mW^3 + (T - W + H)Rm^2)] \quad (5.48)$$

sendo que há um impacto dos fatores Q_g e T_p .

Quando ocorre a seleção aleatória de parâmetros, o custo fica dependente de T_p , que nesse caso corresponde a quantidade de vezes que o PRSSA+ENFN é executado. Dessa forma:

$$O[T_p (mW^3 + (T - W + H)Rm^2)] \quad (5.49)$$

Finalmente, destaca-se que, quando não se está lidando com uma série com uma quantidade grande de amostras, a execução do PRSSA+ENFN em qualquer configuração dura poucos segundos nos computadores habituais. A quantidade de regras é controlada, fazendo com que seu impacto também não seja percebido nesses casos. Contudo, quando aumenta-se consideravelmente W , cuja ordem é cúbica na notação, percebe-se que o PRSSA+ENFN apresenta maior complexidade computacional no pré-processamento e, portanto, demora mais para finalizar sua execução.

5.7 Comentários Finais do Capítulo

O método proposto nessa tese, denominado de PRSSA+ENFN, foi apresentado nesse Capítulo. Essa abordagem se caracteriza por ser híbrida, agregando as funcionalidades do PRSSA para extração recursiva e paralela de componentes não-observáveis com o mapeamento realizado pela ENFN para previsão das componentes. Destacou-se a característica evolutiva e adaptativa da ENFN, que modifica a quantidade de regras *fuzzy* e adapta seus parâmetros globais, locais e matrizes do consequente conforme novas observações da série temporal são recebidas.

O PRSSA+ENFN tem sua operação subdividida em três etapas que foram detalhadas: pré-processamento, treinamento incremental e previsão. Esse método proposto tem a flexibilidade para trabalhar com séries temporais univariáveis e multivariáveis, assim como para previsão recorrente ou direta.

Além disso, a determinação dos parâmetros críticos é importante e afeta os resultados de previsão da abordagem proposta. Para escolha desses parâmetros críticos do PRSSA+ENFN, recomendou-se nesse trabalho o uso dos algoritmos genéticos ou seleção aleatória dos parâmetros dentro de intervalos preestabelecidos. Verificou-se ainda a complexidade computacional através de uma análise assintótica no pior caso por meio da notação $O(\cdot)$ de cada uma das etapas e dos critérios de seleção dos parâmetros críticos do PRSSA+ENFN.

6 RESULTADOS EXPERIMENTAIS

6.1 Descrição Geral dos Experimentos

Esse capítulo apresenta resultados experimentais e uma análise do método PRSSA+ENFN desenvolvido nessa tese. Uma avaliação das principais características desse método proposto é realizada na Seção 6.2, comparando as previsões do tipo recorrente e direta e quais vantagens e desvantagens de cada uma das abordagens, além de verificar em que situações cada um é mais adequada. Estuda-se também nessa seção como determinar o parâmetro s_j de cada componente não-observável, por meio da função de autocorrelação amostral. Além disso, são examinados os efeitos nos resultados e no processo evolutivo do PRSSA+ENFN ao ajustar seus parâmetros críticos.

Por conseguinte, esse capítulo descreve 4 experimentos envolvendo o PRSSA+ENFN, comparando com outros trabalhos e técnicas para o problema de previsão de séries temporais e aplicando em um problema real. O Experimento 1 envolve a previsão de séries temporais univariáveis do tipo recorrente, fazendo um paralelo com os resultados apresentados no trabalho de Nguyen e Novák (2019). Por sua vez, o Experimento 2 realiza uma previsão direta considerando séries temporais univariáveis, visando comparar o PRSSA+ENFN com outros trabalhos que apresentaram técnicas híbridas na literatura, de acordo com Oliveira et al. (2020). O Experimento 3 compara as previsões do PRSSA+ENFN com várias abordagens trazidas no trabalho de Du et al. (2020), no contexto de previsão direta de séries multivariáveis. Finalmente, o Experimento 4 descreve o comportamento do PRSSA+ENFN diante de um problema complexo e desafiador, relativo à previsão do número diário de casos e mortes causados pela Covid-19 no estado do Maranhão, Brasil.

6.2 Avaliação de Características do PRSSA+ENFN

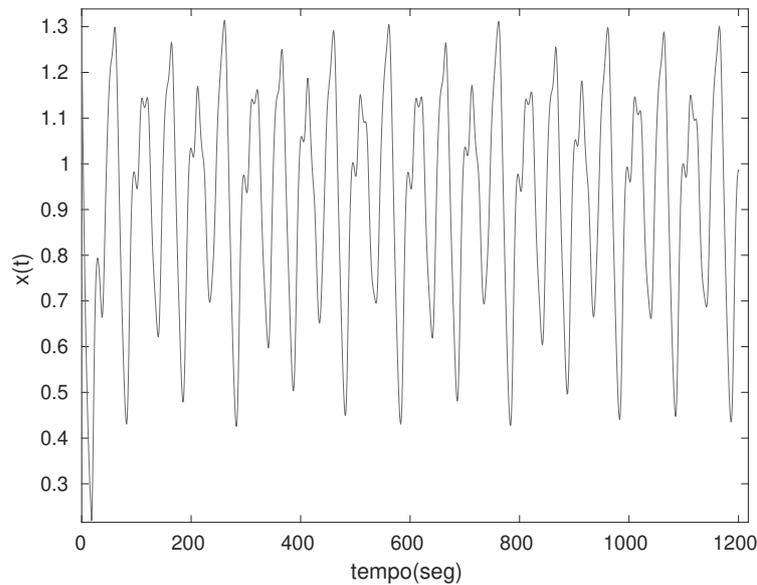
6.2.1 Avaliação das Previsões do Tipo Recorrente e Direta do PRSSA+ENFN

Nessa seção, avalia-se os resultados da previsão recorrente e da previsão direta do método PRSSA+ENFN, verificando em que situação é mais adequado utilizar uma ou outra abordagem. A série temporal considerada nesse estudo é a Mackey-Glass (MACKEY; GLASS, 1977), definida como uma série caótica representada pela seguinte equação diferencial:

$$\dot{x}(t) = \frac{0.2x(t - \tau)}{1 + x^{10}(t - \tau)} - 0.1x(t), \quad (6.1)$$

cuja condição inicial é $x(0) = 1.2$ e $\tau = 17$. Na Figura 25, exhibe-se graficamente essa série temporal.

Figura 25 – Série Temporal Mackey-Glass (Equação 6.1).



Fonte: Autor.

Nesse experimento, a série temporal para estudo tem $N = 1201$ observações no total. Porém, como simulação nesse estudo, supõe-se que há disponível somente $W = 361$ amostras inicialmente. Esses dados vão ser utilizados pelo PRSSA+ENFN na etapa de pré-processamento. Em seguida, novos dados vão chegando a cada instante após W , momento em que o PRSSA+ENFN está executando sua etapa de treinamento incremental para aprender a partir desses dados. Essa aprendizagem finaliza após $T = 901$ observações da Mackey-Glass. A partir desse momento, deseja-se fazer a previsão com horizonte H , isto é, busca-se obter os dados de $T + 1$ a $T + H$. Logo, após T observações, supõe-se que não se conhece o comportamento desses dados nos próximos instantes, pois estão no futuro. Essa previsão pelo PRSSA+ENFN pode ser realizada usando a abordagem recorrente ou direta.

Na previsão recorrente, as próprias componentes previstas pelo PRSSA+ENFN são realimentadas como suas entradas até que todos os dados ao longo do horizonte de previsão H sejam computados. Nesse caso, $p = 1$, ou seja, o PRSSA+ENFN é preparado no treinamento para mapear as componentes de $k - 1$ a k , a fim de obter as próximas componentes com base nas últimas computadas pelo PRSSA+ENFN.

O diferencial da previsão direta é que os resultados do PRSSA+ENFN não são realimentados, sendo que se utiliza as próprias componentes extraídas pelo PRSSA para obter as previsões ao longo do horizonte H . Na etapa de treinamento incremental, o PRSSA+ENFN evolui sua estrutura para mapear as componentes de $k - p$ a k . Em problemas práticos de previsão, só há dados disponíveis até T , que é o caso simulado nesse experimento. Sendo assim, no caso específico em que o PRSSA+ENFN deseja computar a previsão da última observação no instante

$T + H$, somente as componentes até o instante T conhecidas podem ser aplicadas. Logo, nesses casos práticos, pode-se supor que $p = H$ para que o PRSSA+ENFN aprenda a mapear de $k - H$ a k e utilizar as informações disponíveis das componentes para obter valores futuros para a série temporal.

Dessa forma, esse experimento consiste em fazer a previsão recorrente e direta da série temporal Mackey-Glass de acordo com essas condições supracitadas, analisando seus resultados de treinamento e de previsão para diferentes valores do horizonte H . Os parâmetros críticos do PRSSA+ENFN (Parte 2 da Tabela 3) foram fixados em um primeiro momento, com o objetivo de avaliar essas variações para o mesmo conjunto de valores. Esses valores são:

- $W = 361$;
- $n_1 = n = 4$;
- $\phi = 0.3$;
- $\eta_0 = 0.1$,

enquanto os parâmetros do problema de previsão são (Parte 1 da Tabela 3):

- $H =$ variados valores;
- $T = 901$;
- $m = 1$ (série univariável);
- $p = H$ (previsão direta) ou $p = 1$ (previsão recorrente);
- $n_{max} = 10$.

Tabela 4 – Resultados do Treinamento Incremental do PRSSA+ENFN com a Abordagem Recorrente e Direta para a Série Mackey-Glass usando o Critério RMSE.

Horizonte	Direta	Recorrente
$H = 5$	0.031319	0.002316
$H = 10$	0.085121	0.002316
$H = 15$	0.107617	0.002316
$H = 25$	0.118912	0.002316
$H = 150$	0.060696	0.002316
$H = 300$	0.053035	0.002316

Fonte: Autor.

A Tabela 4 exhibe os resultados de treinamento incremental do PRSSA+ENFN, segundo os parâmetros definidos acima, preparando-se para realizar a previsão direta ou recorrente, considerando os diferentes valores de H descritos. O critério de erro utilizado nesse estudo para

comparar os dados reais com os dados de treinamento do PRSSA+ENFN é o RMSE (Eq. 2.16), para $k = W + 1, \dots, T$. Como na previsão direta a condição $p = H$ é aplicada, nota-se que os erros de treinamento são variados devido aos diferentes mapeamentos do instante $k - p$ a k efetuados pelo PRSSA+ENFN. Esses distintos valores de $p = H$ afetam o processo de aprendizagem incremental e, portanto, distintas regras *fuzzy* são geradas. Por outro lado, os valores de RMSE são os mesmos para qualquer horizonte H na previsão recorrente, pois o mapeamento executado pelo PRSSA+ENFN é de $k - 1$ a k em todos os casos da tabela ($p = 1$). Em resumo, caso a ENFN utilize o mesmo mapeamento de $k - p$ a k , as medidas de erro durante as amostras de treinamento serão sempre as mesmas, independente do valor de H .

Em contrapartida, comparando-se os valores reais da série temporal com as amostras previstas pela PRSSA+ENFN na etapa de previsão segundo o critério RMSE, nota-se um comportamento diferente ao treinamento. Para cada horizonte H , o PRSSA+ENFN apresenta um resultado distinto na etapa de previsão no caso recorrente, conforme Tabela 5. Nesse caso, o erro vai crescendo de acordo com o aumento de H , pois o PRSSA+ENFN realimenta as previsões como entradas juntamente com os erros, que vão se propagando ao longo do horizonte de previsão. Por outro lado, o PRSSA+ENFN mostra resultados mais consistentes na previsão direta, pois não ocorre o problema de propagação de erros em suas entradas, sendo que os erros dependem sobretudo da qualidade do mapeamento promovido.

Tabela 5 – Resultados de Previsão do PRSSA+ENFN com a Abordagem Recorrente e Direta para a Série Mackey-Glass usando o Critério RMSE.

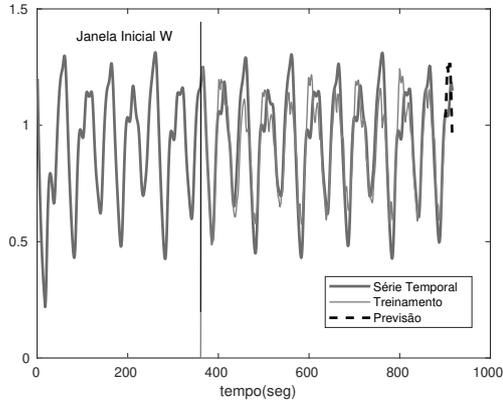
Horizonte	Direta	Recorrente
$H = 5$	0.053198	0.017427
$H = 10$	0.084494	0.038567
$H = 15$	0.149526	0.114799
$H = 25$	0.074024	0.151345
$H = 150$	0.065952	0.220718
$H = 300$	0.129888	0.233804

Fonte: Autor.

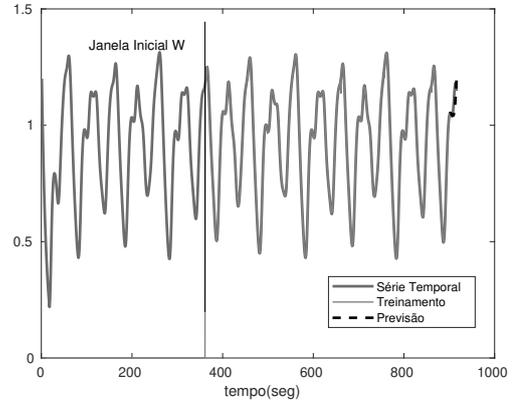
A Figura 26 exhibe os resultados de treinamento e previsão direta e recorrente da série Mackey-Glass relacionados às Tabelas 4 e 5, focando nos horizontes de previsão $H = 15$ e $H = 300$. Nesses gráficos, em azul são representados os dados reais da série, sendo que a janela inicial de dados para o pré-processamento ($W = 361$) está antes da linha preta vertical. Em vermelho, a aproximação feita pelo PRSSA+ENFN na etapa de treinamento incremental até $T = 901$ e, em preto, estão as previsões do PRSSA+ENFN até $T + H$ comparadas com os dados reais da série. A previsão recorrente teve resultado aproximado aos valores reais da série para $H = 15$ como mostra a Figura 26.(b), contudo, o PRSSA+ENFN não se saiu bem para $H = 300$ com a abordagem recorrente na Figura 26.(d). Por sua vez, a previsão direta mostrou sua consistência de acordo com as tabelas e figuras, mesmo considerando os diferentes valores

de H .

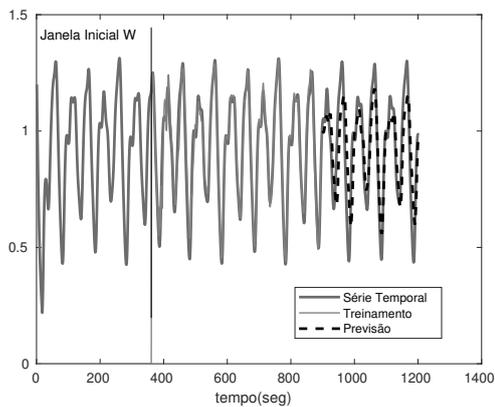
Figura 26 – Resultados de Previsão Direta e Recorrente da Série Mackey-Glass Relacionados às Tabelas 4 e 5.



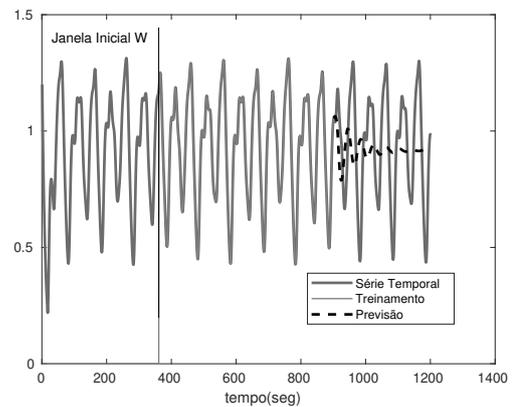
(a) Previsão Direta para $H = 15$.



(b) Previsão Recorrente para $H = 15$.



(c) Previsão Direta para $H = 300$.



(d) Previsão Recorrente para $H = 300$.

Fonte: Autor.

Com esses resultados, verifica-se que para horizontes de previsão mais longos, a previsão recorrente do PRSSA+ENFN terá mais dificuldades devido a propagação dos erros, como ocorre normalmente com qualquer método de previsão recorrente. Contudo, para previsões mais curtas, pode apresentar melhores resultados que a previsão direta. No estudo acima, os parâmetros críticos foram considerados fixos, no entanto, é interessante observar o comportamento para diferentes valores a esses parâmetros.

Aplicando outra perspectiva na análise, utilizou-se o algoritmo genético (Seção 5.5.1) para encontrar valores para os parâmetros críticos do PRSSA+ENFN em cada horizonte H apresentados nas tabelas anteriores, nos casos de previsão direta e recorrente. Similarmente à Tabela 5, a Tabela 6 a seguir exhibe os resultados de previsão do PRSSA+ENFN de acordo com o critério RMSE, relacionados aos parâmetros críticos encontrados pelo algoritmo genético, contando com uma população de 50 cromossomos em 1000 gerações. Por sua vez, os valores desses parâmetros críticos aplicados são mostrados na Tabela 7.

Tabela 6 – Melhores Resultados de Previsão do PRSSA+ENFN com a Abordagem Recorrente e Direta para a Série Mackey-Glass usando o Critério RMSE e Aplicando o Algoritmo Genético para Seleção de Parâmetros Críticos.

Horizonte	Direta	Recorrente
$H = 5$	0.0046946	0.0009331
$H = 10$	0.0292173	0.0132242
$H = 15$	0.0628741	0.0214182
$H = 25$	0.0298122	0.0270144
$H = 150$	0.0503961	0.1545328
$H = 300$	0.0964298	0.2059543

Fonte: Autor.

Tabela 7 – Parâmetros Críticos Selecionados pelo Algoritmo Genético Relacionados aos Resultados de Previsão da Tabela 6.

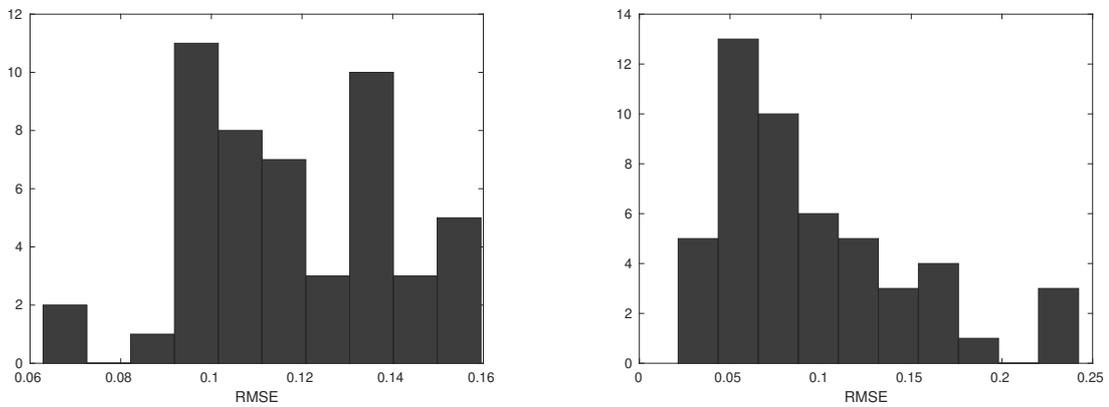
Horizonte	Previsão Direta				Previsão Recorrente			
	W	n	ϕ	η_0	W	n	ϕ	η_0
$H = 5$	354	2	0.49	0.16	328	7	0.11	0.14
$H = 10$	352	3	0.31	0.03	355	8	0.39	0.08
$H = 15$	317	8	0.2	0.18	327	2	0.32	0.23
$H = 25$	334	7	0.5	0.09	331	7	0.44	0.24
$H = 150$	317	5	0.25	0.12	304	6	0.31	0.02
$H = 300$	318	2	0.38	0.07	320	4	0.26	0.27

Fonte: Autor.

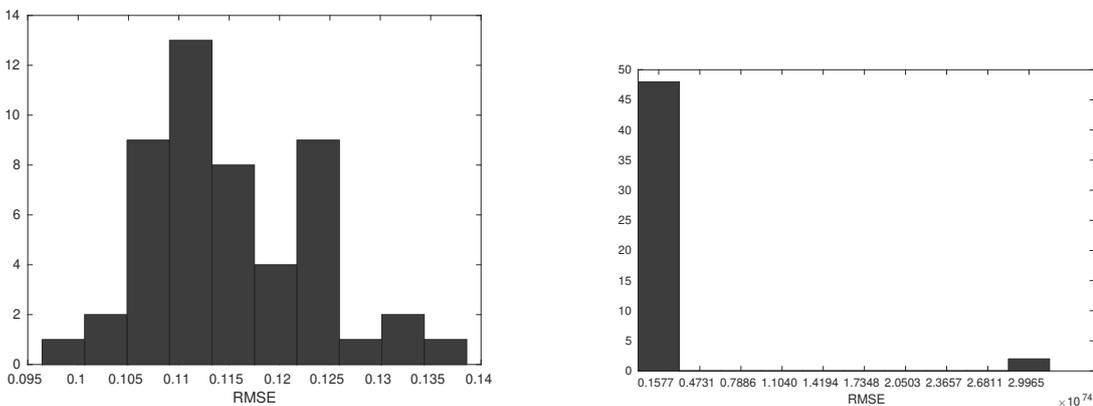
Confirma-se que o PRSSA+ENFN pode ter menores critérios de erro RMSE utilizando a previsão recorrente a curto prazo, com estreitos horizontes H . Contudo, verifica-se observando a Tabela 6 que os erros vão se elevando conforme aumenta-se o horizonte de previsão. Em contrapartida, a previsão direta efetuada pelo PRSSA+ENFN mantém consistente seus resultados de previsão, ou seja, valores próximos um dos outros.

Além disso, é importante analisar como ficaram os resultados de previsão do método PRSSA+ENFN relacionados à população final de soluções ou parâmetros críticos descobertos pelo algoritmo genético em cada caso. A Figura 27 exibe por meio de histogramas os valores de RMSE da população final considerando os horizontes de previsão específicos $H = 15$ e $H = 300$, para a previsão direta e recorrente. Destaca-se que o critério RMSE mantém uma dispersão menor mesmo para diferentes valores de H e para distintos parâmetros críticos do PRSSA+ENFN no caso da previsão direta. Entretanto, os resultados para a previsão recorrente podem ser mais dispersos, pois alguns valores de previsão podem divergir quando estuda-se longos horizontes de previsão, devido à propagação de erros característica nesse caso, o que pode-se perceber nesses histogramas abaixo.

Figura 27 – Histograma das Medidas RMSE dos Erros da Etapa de Previsão do PRSSA+ENFN da População Final do Algoritmo Genético (Tabela 6).



(a) Histograma no caso da Previsão Direta para $H = 15$. (b) Histograma no caso da Previsão Recorrente para $H = 15$.



(c) Histograma no caso da Previsão Direta para $H = 300$. (d) Histograma no caso da Previsão Recorrente para $H = 300$.

Fonte: Autor.

Logo, é aconselhável o usuário do PRSSA+ENFN aplicar tanto a previsão recorrente quanto direta para os problemas de previsão. Se os resultados de previsão no caso recorrente forem excessivamente dispersos, pode-se ter a condição de um horizonte de previsão muito longo para que o PRSSA+ENFN possa atuar para esse problema em estudo. Para horizontes de previsão mais curtos, ambas abordagens podem fornecer bons resultados. Esses pontos analisados nessa seção também ocorrem da mesma forma para séries temporais multivariáveis, conforme a seguir.

Nesse outro estudo, quatro séries temporais mensais, que podem ser encontradas no trabalho de Gan et al. (2014), apresentam $N = 120$ observações e são descritas a seguir:

1. *Book*: Vendas de **livros**;
2. *Clothing*: Vendas de **roupas**;
3. *Furniture*: Vendas de **móveis**;

4. Hardware: Vendas de *hardware*.

A previsão pelo PRSSA+ENFN nesse experimento deve ser realizada de forma multivariável, considerando $\mathbf{y}_k = [y_k^1 \ y_k^2 \ y_k^3 \ y_k^4]$, para $m = 4$ séries temporais e $p = 1$, no caso recorrente, e $p = H$, no caso da previsão direta. São analisados três horizontes de previsão para avaliar os resultados do PRSSA+ENFN proposto: $H = 12$, $H = 24$ e $H = 36$. Uma parte dos dados corresponde a janela inicial W para a etapa de pré-processamento ($k = 1, \dots, W$), outra parte deve ser utilizada para treinamento incremental ($k = W + 1, \dots, T$) e o último conjunto de observações para análise dos resultados de previsão ($k = T + 1, \dots, T + H$).

Os parâmetros PRSSA+ENFN foram obtidos a partir da execução do algoritmo genético, considerando uma população com 100 cromossomos após 1000 gerações. Os cromossomos para esse experimento possuem a seguinte estrutura:

$$\text{cromossomo} = [W \ n_1 \ n_2 \ n_3 \ n_4 \ \phi \ \eta_0], \quad (6.2)$$

sendo que:

$$\begin{aligned} W &= 20\%N, \dots, 30\%N; \\ n_l &= 2, 3, \dots, n_{max} = 10, \text{ para } l = 1, 2, 3, 4; \\ 0.1 &\leq \phi \leq 0.5; \\ 0.01 &\leq \eta_0 \leq 0.3. \end{aligned} \quad (6.3)$$

A Tabela 8 mostra os resultados comparativos entre a previsão recorrente e direta de acordo com o RMSE para os horizontes $H = 12$, $H = 24$ e $H = 36$, respectivamente. Percebe-se que para $H = 12$ e $H = 24$, a previsão recorrente apresentou medidas de erro menores que a previsão do tipo direta. Em contrapartida, a previsão direta se saiu melhor quando $H = 36$, sendo que apresentou menor valor de RMSE.

Tabela 8 – Resultados de Previsão Recorrente e Direta de acordo com o Critério RMSE para Previsão Multivariável Considerando Diferentes Horizontes de Previsão.

Horizonte de Previsão	<i>Book</i>	<i>Clothing</i>	<i>Furniture</i>	<i>Hardware</i>
$H = 12$ (Recorrente)	54.22	509.40	94.19	51.77
$H = 24$ (Recorrente)	91.05	800.67	150.64	82.07
$H = 36$ (Recorrente)	151.78	1359.37	249.02	148.16
$H = 12$ (Direta)	109.31	647.96	142.07	61.25
$H = 24$ (Direta)	115.91	873.42	201.35	85.15
$H = 36$ (Direta)	145.61	1020.36	241.60	100.92

Fonte: Autor.

Por fim, as Figuras no Apêndice C.1 mostram os resultados gráficos de treinamento incremental e previsão recorrente e direta, respectivamente, para a série temporal multivariável neste estudo. Na etapa de pré-processamento, o PRSSA+ENFN não computa valores para as séries. Durante o intervalo de tempo $k = W + 1, \dots, T$, a etapa de treinamento incremental é

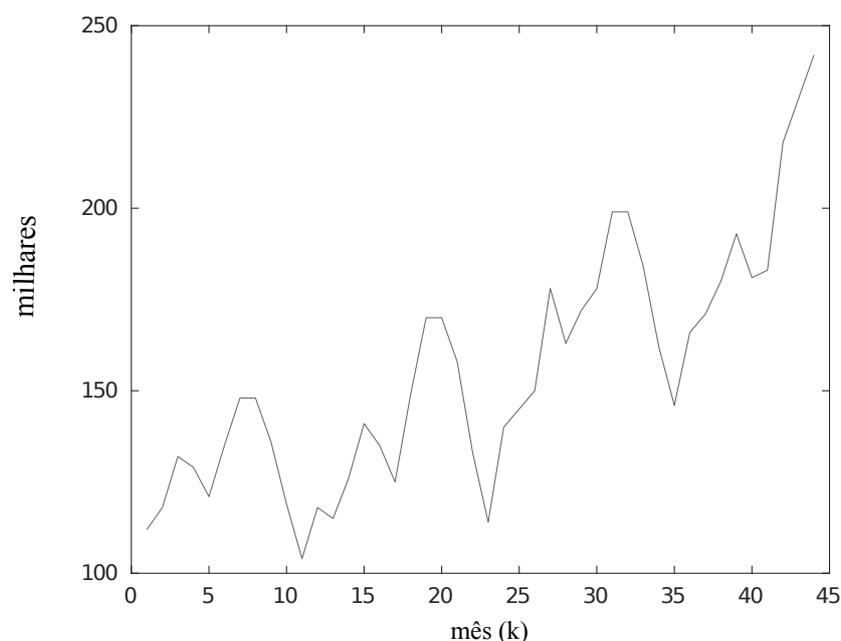
efetuada e, por fim, o PRSSA+ENFN interrompe sua evolução para computar os valores futuros das séries ao longo do horizonte de previsão. Além disso, mostra-se também o aspecto evolutivo com o comportamento da quantidade das regras *fuzzy* ou *data clouds*.

Outra característica da operação multivariável do PRSSA+ENFN é que as componentes não-observáveis das diferentes séries são postas na mesma estrutura do modelo de vetores autorregressivos (Equação 5.36), sendo que os diferentes submodelos locais consideram a relação entre as componentes para proceder com a previsão.

6.2.2 Determinação do Parâmetro s_j para cada Componente Não-Observável

Sabendo que n é a quantidade total de componentes não-observáveis, a determinação do parâmetro s_j , para $j = 1, \dots, n$, é necessária para operação do método PRSSA+ENFN, pois está presente na estrutura das regras *fuzzy*, conforme Equação 5.36. Sua definição está relacionada à análise da função de autocorrelação amostral de cada uma das n componentes. O parâmetro s_j deve ser determinado na etapa de pré-processamento, com base nos dados das componentes oriundas da extração inicial para um conjunto de dados com as W primeiras observações da série. Sejam as W primeiras observações de uma série temporal representada na Figura 28, para $W = 44$ e $n = 3$. Essa série temporal diz respeito ao *Número de Passageiros de Companhias Aéreas Internacionais em Milhares*, com dados mensais de Janeiro de 1949 a Dezembro de 1960 (BROCKWELL; DAVIS, 2002).

Figura 28 – Conjunto Inicial de Dados Disponíveis da Série Temporal ($W = 44$ primeiras observações).

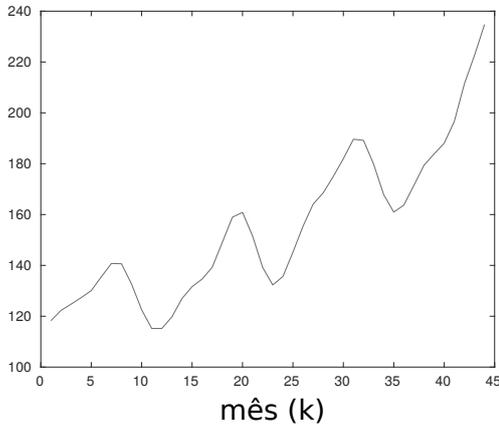


Fonte: Autor.

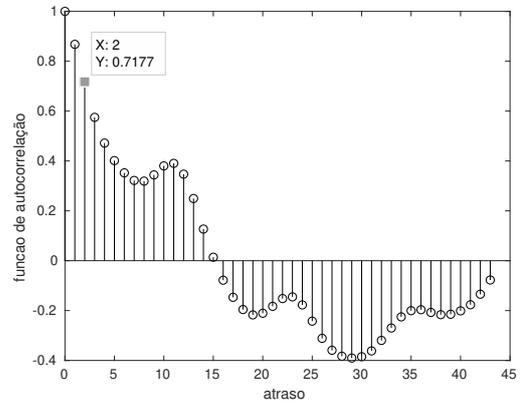
Dessa forma, a extração das $n = 3$ componentes iniciais é realizada no pré-processamento,

contando com essas $W = 44$ primeiras observações. As componentes estão representadas nas Figuras 29.(a), 29.(c) e 29.(e).

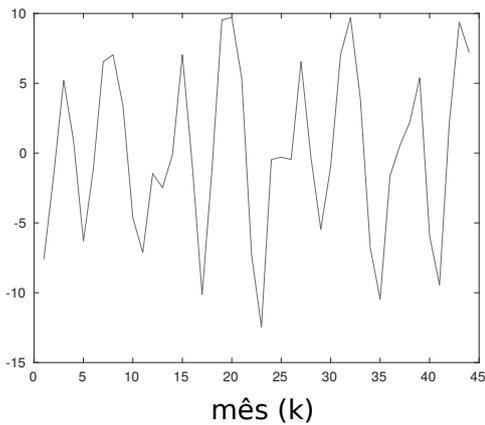
Figura 29 – Determinação de s_j para Cada Componente Não-Observável Inicial.



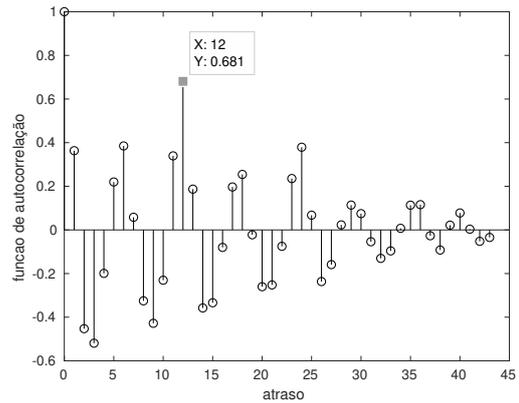
(a) Componente 1.



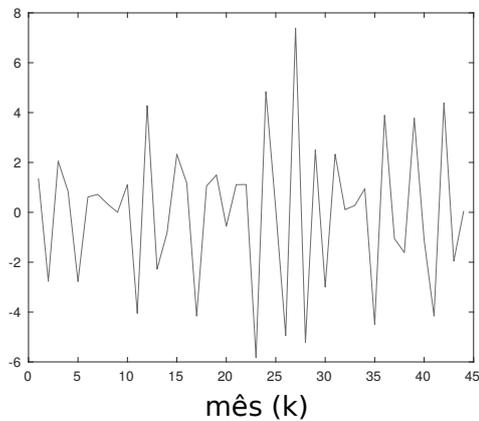
(b) Função de Autocorrelação Amostral da Componente 1.



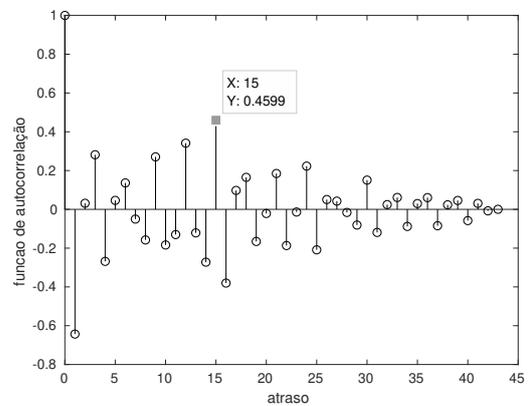
(c) Componente 2.



(d) Função de Autocorrelação Amostral da Componente 2.



(e) Componente 3.



(f) Função de Autocorrelação Amostral da Componente 3.

Fonte: Autor.

Por sua vez, as funções de autocorrelação amostral relacionadas a cada uma dessas componentes estão nas Figuras 29.(b), 29.(d) e 29.(f), respectivamente. Ao analisar as funções de autocorrelação amostral de cada componente, busca-se por observações que tenham uma autocorrelação de destaque com a amostra no instante k . Deve-se ignorar os atrasos em $k - 0$ e $k - p$, pois o primeiro indica a correlação com a própria observação no instante k , enquanto o segundo corresponde a observação no instante $k - p$ que já está sendo considerada na estrutura das regras *fuzzy* como entradas (Equação 5.36).

Observando a Figura 29.(b), nota-se que os primeiros atrasos são os que possuem um valor de autocorrelação mais significativo e, a partir dessa indicação decrescente, infere-se que há um comportamento de tendência. O valor do parâmetro fica $s_1 = 2$, pois é o ponto que se destaca nessa função. Em relação a Figura 29.(d), há um comportamento periódico de acordo com a função de autocorrelação amostral, definindo $s_2 = 12$. Por fim, a última componente tem $s_3 = 15$ como indicação mais significativa de autocorrelação, de acordo com a Figura 29.(f).

No caso de séries temporais multivariáveis, esse procedimento é o mesmo para todas as n componentes das m séries temporais. A determinação de cada s_j só é possível tendo em mãos uma quantidade de dados (W) das componentes não-observáveis. Devido principalmente a essa necessidade, a extração inicial de componentes no pré-processamento deve ser efetuada.

Entretanto, modificando os parâmetros W e n , variações irão ocorrer nos valores de s_j . Alterando-se W , há uma mudança na quantidade de observações da série temporal usadas no pré-processamento para extração das componentes, conseqüentemente, modifica-se a quantidade de dados para cada componente $j = 1, \dots, n$. Por outro lado, alterando-se n , padrões distintos dos dados podem ser encontrados, com diferentes valores para s_j . O impacto das variações de W e n depende da característica de cada série temporal.

Aumentando o valor de W a partir dos dados da Figura 28 e mantendo $n = 3$, percebe-se pela Tabela 9 que houve uma modificação em s_3 . Quanto maior W , mais próximo os s_j serão dos valores para o caso em que se considera a função de autocorrelação amostral de todo conjunto N de dados da série temporal.

Tabela 9 – Valores de s_j Devido à Variação de W .

W	s_1	s_2	s_3
44	2	12	15
58	2	12	12
72	2	12	12

Fonte: Autor.

Agora mantendo $W = 44$ fixo e alterando o valor de n , tem-se os resultados na Tabela 10. Atribuir distintos valores para n faz com que a extração inicial obtenha diferentes componentes, cada qual com um novo valor para s_j .

É necessário analisar também os resultados quando não se utiliza os parâmetros s_j no

Tabela 10 – Valores de s_j Devido à Variação de n .

n	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}
3	2	2	15	-	-	-	-	-	-	-
4	2	2	12	15	-	-	-	-	-	-
10	2	12	12	6	5	3	12	4	5	2

Fonte: Autor.

método PRSSA+ENFN. Nesse experimento, foram realizadas previsões do tipo recorrente e direta da série temporal univariável ($m = 1$) da Figura 28, tanto aplicando os parâmetros s_j para cada componente quanto não os utilizando, ou seja:

- Previsão recorrente usando os parâmetros s_j , para $j = 1, \dots, n$;
- Previsão recorrente sem utilizar os parâmetros s_j ;
- Previsão direta usando os parâmetros s_j , para $j = 1, \dots, n$;
- Previsão direta sem utilizar os parâmetros s_j .

Em todos os casos acima, os parâmetros do problema de previsão (Parte 1 da Tabela 3) e dos parâmetros críticos do PRSSA+ENFN (Parte 2 da Tabela 3) foram considerados os mesmos, conforme a Tabela 11 a seguir. Em geral, são os mesmos valores em todos os casos, exceto para o passo p , que tem valor $p = 1$ na previsão recorrente e $p = H = 15$ na previsão direta. Na recorrente, realizam-se várias previsões de um passo a frente até alcançar o horizonte de previsão H , enquanto a direta utiliza os dados das componentes disponíveis para obter a previsão ao longo de H com base em dados no instante $k - p$. Os valores de s_j ($j = 1, 2, 3$) são os mesmos apresentados na Tabela 9 para $W = 44$.

Tabela 11 – Parâmetros Utilizados para cada Previsão na Análise da Contribuição de s_j .

Parâmetros	Recorrente com s_j	Recorrente sem s_j	Direta com s_j	Direta sem s_j
H	15	15	15	15
T	129	129	129	129
N	144	144	144	144
p	1	1	15	15
W	44	44	44	44
$n_1 = n$	3	3	3	3
ϕ	0.3	0.3	0.3	0.3
η_0	0.15	0.15	0.15	0.15

Fonte: Autor.

Nos casos onde não se aplicam os parâmetros s_1, s_2 e s_3 , algumas adaptações são implementadas na estrutura do método PRSSA+ENFN, mais especificamente nas regras *fuzzy* e na

estimação das matrizes dos consequentes. A regra *fuzzy*, originalmente representada na Equação 5.36, nesses casos é modificada conforme a seguir:

$$\begin{aligned} \text{Regra } Fuzzy_i: & \text{ SE } (\hat{\mathbf{C}}_{k-p} \sim \mathbf{p}_i) \\ & \text{ ENTÃO } \hat{\mathbf{C}}_k^\top = \mathbf{A}_k^i \hat{\mathbf{C}}_{k-p}^\top. \end{aligned} \quad (6.4)$$

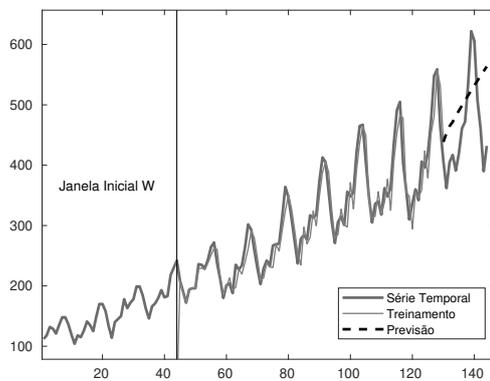
A estimação recursiva para cada regra *fuzzy* $i = 1, 2, \dots, R$ pelo método L-FWRLS também é afetada, pois as dimensões de algumas das matrizes são alteradas. Partindo no mesmo raciocínio da Equação 5.37, tem-se para o caso sem os parâmetros s_j :

$$\begin{aligned} \hat{\mathbf{C}}_k^\top &= \mathbf{A}_k^i \hat{\mathbf{C}}_{k-p}^\top, \\ \hat{\mathbf{C}}_k^\top &= \begin{bmatrix} \mathbf{A}_k^i \end{bmatrix} \begin{bmatrix} \hat{\mathbf{C}}_{k-p}^\top \end{bmatrix}, \\ \hat{\mathbf{C}}_k^\top &= (\Delta_k^i)^\top \mathbf{v}^\top. \end{aligned} \quad (6.5)$$

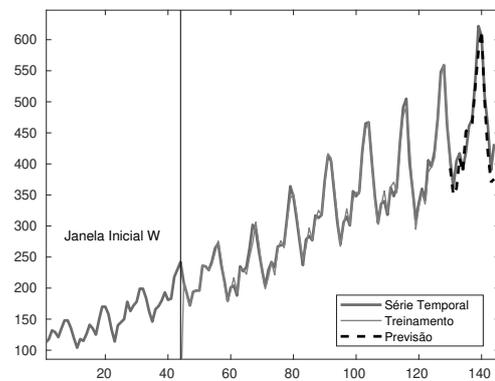
para $\Delta_k^i \in \mathbb{R}^{n \times n}$ e $\mathbf{v} \in \mathbb{R}^{1 \times n}$. Logo, a expressão apresentada na Equação 5.38 do método L-FWRLS permanece a mesma, exceto pela dimensão das matrizes $\mathbf{Z}_k^i \in \mathbb{R}^{n \times 1}$, $\Delta_k^i \in \mathbb{R}^{n \times n}$ e $\mathbf{J}_k^i \in \mathbb{R}^{n \times n}$.

A Figura 30 exibe os resultados gráficos de treinamento e de previsão direta e recorrente, tanto considerando os parâmetros s_j quanto não os aplicando no método PRSSA+ENFN. A janela inicial W para os 4 gráficos é definida antes da linha vertical, sendo os mesmos dados representados na Figura 28. Observando os gráficos, percebe-se que os resultados de previsão considerando as observações no instante $k - s_j$ para cada componente contribuem para um melhor resultado de previsão. Na Figura 30.(a), a previsão recorrente não conseguiu representar a sazonalidade dessa série, logo, somente os dados das componentes no instante $k - 1$ (Equação 6.4, para $p = 1$) não foram suficientes. Na previsão direta da Figura 28.(c), também houve problemas ao considerar somente os dados das componentes no instante $k - 15$ (Equação 6.4, para $p = 15$).

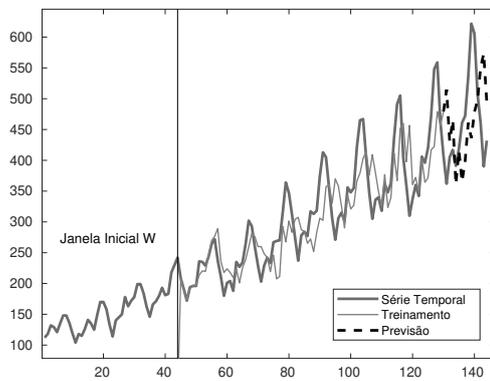
Figura 30 – Resultados Gráficos de Treinamento e para as Previsões Recorrente e Direta com e sem Uso do Parâmetro s_j .



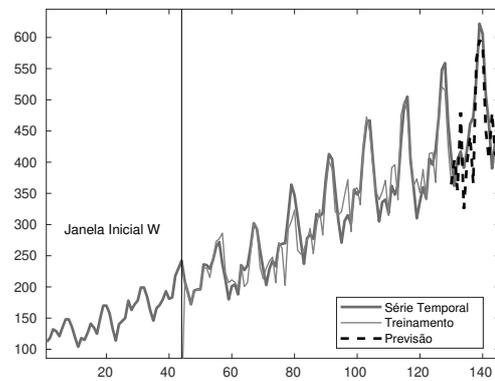
(a) Previsão Recorrente sem s_j .



(b) Previsão Recorrente com s_j .



(c) Previsão Direta sem s_j .



(d) Previsão Direta com s_j .

Fonte: Autor.

A Tabela 12 mostra os resultados de acordo com o RMSE, destacando que os casos em que foram aplicados os parâmetros s_j apresentaram menores valores para esse critério de erro. É importante ressaltar que o objetivo da análise dessa seção não é comparar os resultados entre previsão recorrente e direta, mas sim destacar que os parâmetros s_j contribuem de forma relevante para melhores resultados de previsão do PRSSA+ENFN.

Tabela 12 – Critérios de Erros de Treinamento e para cada Resultado de Previsão na Análise da Contribuição de s_j .

Critério de Erro	Recorrente com s_j	Recorrente sem s_j	Direta com s_j	Direta sem s_j
RMSE Treinamento	9.33	25.65	25.42	46.89
RMSE Previsão	26.89	81.86	53.61	99.16

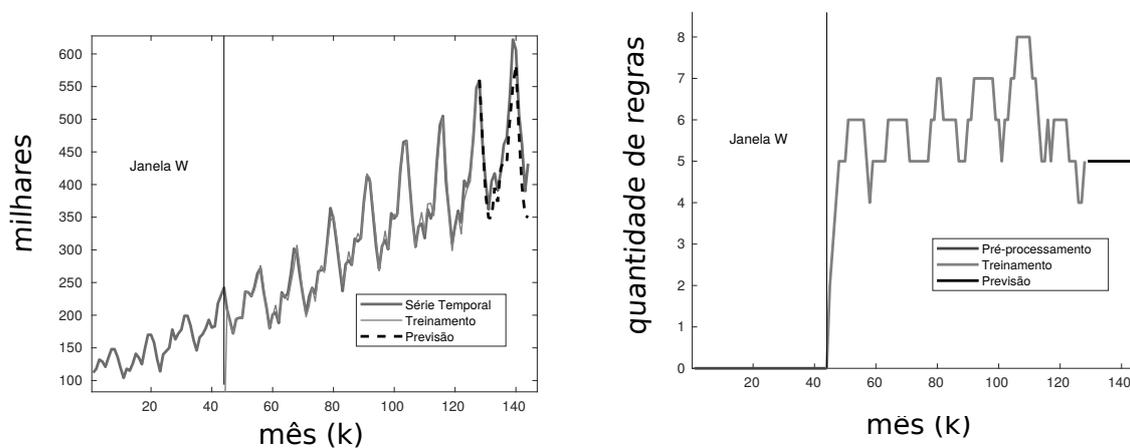
Fonte: Autor.

6.2.3 Análise da Influência nos Resultados do PRSSA+ENFN Devido ao Ajuste dos Parâmetros Críticos

Nessa seção, o impacto ao ajustar os parâmetros críticos do PRSSA+ENFN (Parte 2 da Tabela 3) sobre seus resultados de treinamento e previsão, bem como na evolução da quantidade de regras *fuzzy* será analisado. Nesse estudo, são abordados dois casos com séries temporais univariáveis, um lidando com a previsão do tipo recorrente, enquanto o outro, com a previsão direta. Os efeitos das variações desses parâmetros são os mesmos para séries multivariáveis. Os parâmetros críticos - W , n , ϕ , η_0 - são submetidos a variações um de cada vez nessa análise, averiguando o efeito nos resultados do PRSSA+ENFN para cada um deles.

Primeiramente, a série temporal *Número de Passageiros de Companhias Aéreas Internacionais em Milhares*, a mesma do estudo na seção anterior, é aplicada na perspectiva da previsão recorrente. Os resultados de treinamento e previsão, segundo o critério de erro RMSE, assim como a evolução de regras, são exibidos na Figura 31 e estão relacionados aos valores na primeira linha da Tabela 13.

Figura 31 – Resultados Gráficos de Treinamento e Previsão para a Série Temporal *Número de Passageiros de Companhias Aéreas* Considerando os Parâmetros da Primeira Linha da Tabela 13.



(a) Série Temporal *Número de Passageiros de Companhias Aéreas*.

(b) Evolução da Quantidade de Regras *Fuzzy* ou *Data Clouds*.

Fonte: Autor.

Essa tabela mostra ainda os parâmetros do problema de previsão que são determinados pelo usuário (H, T, N, p) e o vetor $\mathbf{s} = [s_1 \ \cdots \ s_n]$ que informa o atraso com correlação significativa para cada componente j ($j = 1, \dots, n$). Além disso, $m = 1$ pois lida-se com uma série univariável e n_{max} foi fixado em 10 ($n_{max} = 10$). Logo, tendo como ponto de partida esses valores da primeira linha da Tabela 13, os parâmetros críticos são alterados um por vez para verificar o efeito nos novos resultados do método PRSSA+ENFN.

Tabela 13 – Parâmetros do PRSSA+ENFN Usados como Referência Considerando os Casos da Previsão Recorrente (Rec.) e Direta (Dir.).

Tipo	Parâm. do Problema				Parâm. Críticos				Trei.	Prev.	R	\mathbf{s}
	H	T	N	p	W	n	ϕ	η_0				
Rec.	16	128	144	1	44	3	0.30	0.10	10.10	40.94	5	[2; 12; 15]
Dir.	200	2620	2820	15	846	4	0.30	0.10	30.37	38.31	12	[16;94;198;168]

Fonte: Autor.

Inicia-se a análise com a variação somente do parâmetro W , que é a quantidade de dados usados na etapa de pré-processamento, mantendo-se os outros fixos conforme Tabela 13. A Tabela 14 exhibe os resultados de treinamento e previsão de acordo com o critério RMSE, bem como a quantidade de regras e os valores de s_j , para $j = 1, \dots, n$. Nota-se que a variação de W influencia tanto nos resultados de treinamento quanto na previsão, pois modificando-se seu valor, altera-se a quantidade de dados aplicados no pré-processamento e, portanto, afeta a decomposição das componentes não-observáveis iniciais. Além disso, o W impacta de forma importante na definição de s_j ($j = 1, \dots, n$) a partir dos dados de cada componente, como foi debatido na Seção 6.2.2. Quanto maior o valor de W , mais próxima será a definição do valor de s_j ao considerar todo o conjunto total de dados da componente. Nesse exemplo, os valores de s_j para todo conjunto de dados seria $\mathbf{s} = [2; 12; 12]$, e percebe-se que esse valor foi alcançado somente para $W = 58$. Entretanto, aumentando-se o valor de W , haverá menos dados para o treinamento, podendo comprometer a previsão. Logo, o ajuste depende de cada conjunto de dados, pois essas características estão relacionadas às especificidades de cada série temporal e suas componentes não-observáveis. Em relação a quantidade de regras, também houve modificações com o ajuste de W , devido aos diferentes conjuntos de dados das componentes aplicados no treinamento.

Tabela 14 – Resultados do PRSSA+ENFN Devido ao Ajuste do Parâmetro W Considerando o Caso da Previsão Recorrente.

W	RMSE Trei.	RMSE Prev.	R	\mathbf{s}
15	9.99	101.89	6	[2; 12; 9]
29	9.54	58.09	6	[2; 12; 15]
44	10.10	40.94	5	[2; 12; 15]
58	6.29	86.06	7	[2; 12; 12]

Fonte: Autor.

Modificando-se agora a quantidade de componentes não-observáveis n e considerando os outros parâmetros com valores fixados de acordo com a Tabela 13, tem-se um efeito nos resultados de treinamento e previsão, bem como na forma com que as regras *fuzzy* evoluem. O parâmetro \mathbf{s} tem uma alteração natural em sua dimensão devido a quantidade de componentes, pois $j = 1, \dots, n$. Como a quantidade de componentes está se modificando, o PRSSA+ENFN realiza um processo evolutivo distinto em cada caso, pois são diferentes entradas relacionadas

às componentes. Sendo assim, a variação de componentes tem uma influência importante na dinâmica de geração e remoção de regras *fuzzy* ou *clouds* devido também a alteração da dimensão do espaço. A Tabela 15 exibe alguns desses resultados para $n = 2, 3, 7, 10$.

Tabela 15 – Resultados do PRSSA+ENFN Devido ao Ajuste do Parâmetro n Considerando o Caso da Previsão Recorrente.

n	RMSE Trei.	RMSE Prev.	R	s
2	8.07	58.16	5	[2; 12]
3	10.10	40.94	5	[2; 12; 15]
7	7.39	57.59	4	[2; 12; 12; 12; 3; 3; 2]
10	5.95	101.06	2	[2; 12; 12; 6; 5; 3; 3; 4; 5; 2]

Fonte: Autor.

Por sua vez, o parâmetro ϕ relacionado ao fator de sobreposição interfere levemente nos resultados de forma geral. Porém, a utilização desse parâmetro se mostrou suficientemente importante para tentar melhorar os resultados em vários experimentos que serão apresentados em outras seções, mesmo que de forma moderada. Como percebe-se na Tabela 16, os resultados com a variação do fator de sobreposição ϕ foram bem similares, sendo que a quantidade de regras ou *clouds* foi praticamente a mesma, pois esse parâmetro é ajustado para evitar redundância entre esses *data clouds*.

Tabela 16 – Resultados do PRSSA+ENFN Devido ao Ajuste do Parâmetro ϕ Considerando o Caso da Previsão Recorrente.

ϕ	RMSE Trei.	RMSE Prev.	R	s
0.01	10.33	41.42	5	[2; 12; 15]
0.1	10.10	40.94	5	[2; 12; 15]
0.3	10.06	40.96	5	[2; 12; 15]
0.5	9.89	40.23	5	[2; 12; 15]

Fonte: Autor.

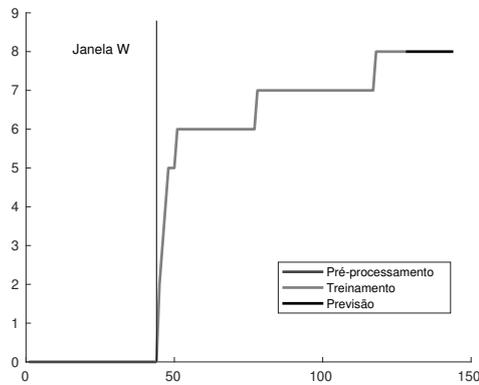
Ao contrário de ϕ , o fator de esquecimento η_0 tem influência relevante nos resultados do PRSSA+ENFN, principalmente no processo evolutivo de regras *fuzzy* ou *data clouds*, como mostra a Tabela 17. Isso porque esse parâmetro crítico está envolvido com o processo de remoção de regras, conforme a Condição 3 na Equação 5.35. Quanto maior seu valor, mais fácil as regras ou *clouds* podem ser removidos da estrutura devido a sua baixa utilidade. Nota-se na Tabela 17 que a quantidade de regras quando encerra o treinamento é menor de acordo com o aumento de η_0 . Modificando-se o processo de evolução das regras *fuzzy*, tem-se com consequência resultados de treinamento e previsão distintos. Tanto o parâmetro ϕ quanto η_0 não alteram a definição de s_j , para $j = 1, \dots, n$, já que não tem relação com a etapa de pré-processamento.

Tabela 17 – Resultados do PRSSA+ENFN Devido ao Ajuste do Parâmetro η_0 Considerando o Caso da Previsão Recorrente.

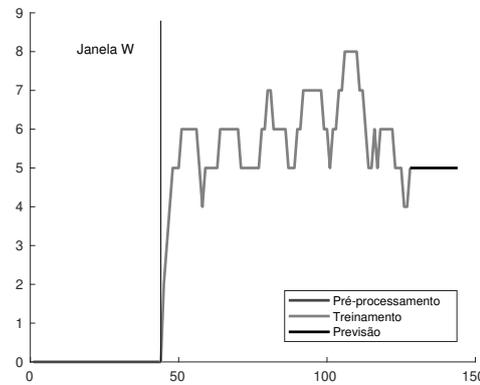
η_0	RMSE Trei.	RMSE Prev.	R	s
0.01	10.54	52.48	8	[2; 12; 15]
0.1	10.10	40.94	5	[2; 12; 15]
0.2	9.57	50.36	3	[2; 12; 15]
0.3	9.40	81.23	3	[2; 12; 15]

Fonte: Autor.

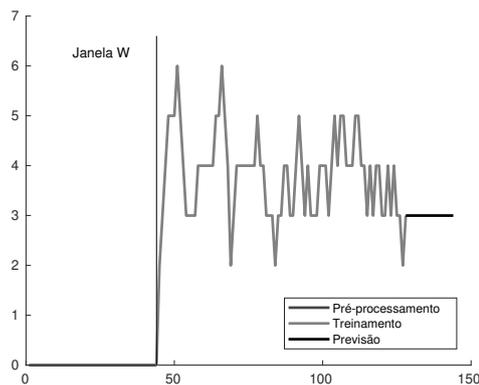
Figura 32 – Dinâmica da Quantidade de Regras *Fuzzy* ou *Data Clouds* devido a Variação do Parâmetro η_0 Conforme Tabela 17.



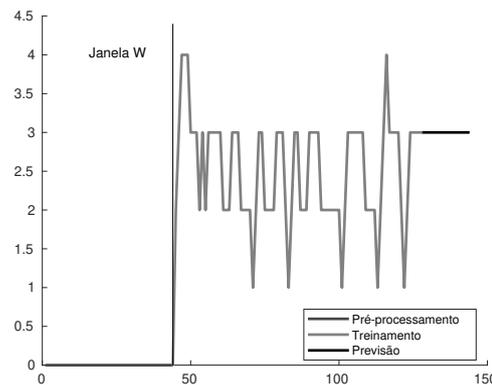
(a) Quantidade de Regras *Fuzzy* ou *Data Clouds* para $\eta_0 = 0.01$.



(b) Quantidade de Regras *Fuzzy* ou *Data Clouds* para $\eta_0 = 0.1$.



(c) Quantidade de Regras *Fuzzy* ou *Data Clouds* para $\eta_0 = 0.2$.



(d) Quantidade de Regras *Fuzzy* ou *Data Clouds* para $\eta_0 = 0.3$.

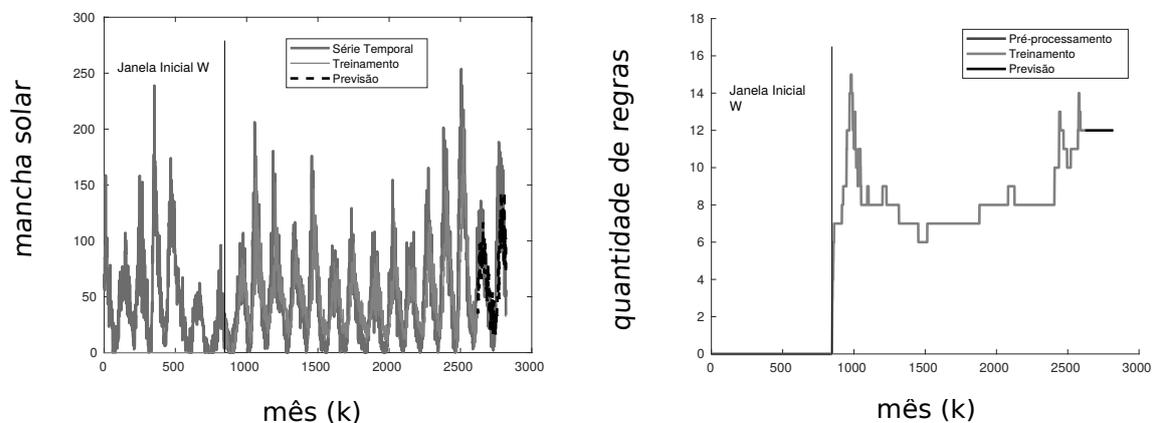
Fonte: Autor.

A Figura 32.(a) exibe a evolução das regras *fuzzy* ou *data clouds* quando $\eta_0 = 0.01$. Percebe-se que não houve a remoção de *clouds* nesse caso, pois a Condição 3 não foi satisfeita em nenhum instante (Equação 5.35). Conforme o valor de η_0 vai aumentando de acordo com os demais gráficos na Figura 32, a remoção de regras ou *clouds* por meio da Condição 3 é

cada vez mais efetuada, fazendo com que a sua quantidade resultante no treinamento seja menor. Em muitos casos, para $\eta_0 > 0.3$, a quantidade de regras fica igual a 1, adicionando e removendo-as a todo instante sem sair desse valor, o que não é interessante para a etapa de treinamento. Em contrapartida, quando $\eta_0 < 0.01$, fica muito difícil ocorrer alguma remoção de *clouds* no procedimento evolutivo, mantendo-se regras *fuzzy* com baixa utilidade na estrutura do PRSSA+ENFN. Por esses motivos, determinou-se um intervalo apropriado descrito por $0.01 \leq \eta_0 \leq 0.3$ para operação da metodologia proposta. Destaca-se ainda que esse comportamento está associado ao processo evolutivo de aprendizagem, ocorrendo para qualquer conjunto de dados.

É essencial fazer uma análise similar para a previsão do tipo direta e também avaliar os resultados para uma série temporal distinta, como a exibida na Figura 33, juntamente com a evolução da quantidade de regras *fuzzy*. Os parâmetros que serviram de referência estão na Linha 2 da Tabela 13, logo, a partir destes, seus valores são ajustados para verificar a influência nos resultados do PRSSA+ENFN. A série temporal nesse estudo descreve a contagem mensal do número de manchas solares observadas por pouco mais de 230 anos (1749-1983), sendo $N = 2820$ observações no total (ANDREWS; HERZBERG, 1985).

Figura 33 – Resultados Gráficos de Treinamento e Previsão para a Série Temporal *Mancha Solar* Considerando os Parâmetros da Linha 2 da Tabela 13.



(a) Série Temporal *Mancha Solar* .

(b) Evolução da Quantidade de Regras *Fuzzy*.

Fonte: Autor.

A variação de W influenciou nesse caso da mesma forma que o estudo de previsão recorrente anterior, modificando os valores dos parâmetros s_j , $j = 1, \dots, n$, assim como na definição das componentes não-observáveis. Portanto, a quantidade de regras e os resultados de treinamento e mapeamento são afetados. A Tabela 18 a seguir exibe essas modificações.

Tabela 18 – Resultados do PRSSA+ENFN Devido ao Ajuste do Parâmetro W Considerando o Caso da Previsão Direta.

W	RMSE Trei.	RMSE Prev.	R	s
282	28.39	41.47	12	[16; 258; 252; 168]
564	20.50	30.02	9	[112; 94; 198; 168]
846	30.37	38.31	12	[16; 94; 198; 168]
1128	27.59	40.31	12	[16; 95; 90; 17]

Fonte: Autor.

Na Tabela 19, pode-se visualizar os resultados para diferentes valores de n na previsão direta. Modificando-se n , distintas entradas são consideradas para a rede neuro-*fuzzy*, logo, os resultados de treinamento, previsão e quantidade de regras são afetados. O parâmetro s tem sua dimensão ajustada, pois depende de n .

Tabela 19 – Resultados do PRSSA+ENFN Devido ao Ajuste do Parâmetro n Considerando o Caso da Previsão Direta.

n	RMSE Trei.	RMSE Prev.	R	s
2	28.48	39.72	11	[16; 291]
4	30.37	38.31	12	[16; 94; 198; 168]
7	29.21	37.23	9	[16; 53; 84; 342; 198; 171; 218]
10	25.87	32.75	10	[16; 138; 94; 84; 89; 112; 198; 17; 168; 356]

Fonte: Autor.

Da mesma forma que para o estudo de previsão recorrente, o parâmetro ϕ impacta levemente os resultados de treinamento e previsão (Tabela 20), controlando o processo de sobreposição de *data clouds*.

Tabela 20 – Resultados do PRSSA+ENFN Devido ao Ajuste do Parâmetro ϕ Considerando o Caso da Previsão Direta.

ϕ	MSE Trei.	RMSE Prev.	R	s
0.01	29.90	38.31	11	[16; 94; 198; 168]
0.1	30.49	38.26	11	[16; 94; 198; 168]
0.3	30.37	38.31	12	[16; 94; 198; 168]
0.5	30.84	38.33	10	[16; 94; 198; 168]

Fonte: Autor.

Finalmente, a Tabela 21 destaca a significativa influência que o parâmetro η_0 tem no processo de evolução da quantidade de regras *fuzzy*, pois está relacionado ao processo de remoção e monitoramento de qualidade dos *clouds*. Quanto maior esse valor, mais regras serão removidas da estrutura do PRSSA+ENFN.

Tabela 21 – Resultados do PRSSA+ENFN Devido ao Ajuste do Parâmetro η_0 Considerando o Caso da Previsão Direta.

η_0	RMSE Trei.	RMSE Prev.	R	s
0.01	30.38	38.46	45	[16; 94; 198; 168]
0.1	30.37	38.31	12	[16; 94; 198; 168]
0.2	22.70	55.31	5	[16; 94; 198; 168]
0.3	27.64	38.39	2	[16; 94; 198; 168]

Fonte: Autor.

Observando esses resultados, o efeito do ajuste desses 4 parâmetros críticos afetam os mesmas particularidades do PRSSA+ENFN tanto para o caso da previsão recorrente quanto para previsão direta. Porém, não há um indicativo geral do melhor conjunto de valores para esses parâmetros, pois isso é dependente da característica de cada série temporal. Apesar disso, definiu-se nessa tese uma faixa de valores de segurança para esses parâmetros críticos, como segue:

$$\begin{aligned}
 W &= 20\%N, \dots, 30\%N; \\
 n &= 2, 3, \dots, n_{max} = 10; \\
 0.1 &\leq \phi \leq 0.5; \\
 0.01 &\leq \eta_0 \leq 0.3,
 \end{aligned} \tag{6.6}$$

6.3 Experimento 1: Previsão Recorrente de Séries Temporais Univariáveis

Nesta seção, a metodologia proposta é avaliada usando a previsão recorrente de séries temporais univariáveis ($m = 1$), comparando seus resultados com a pesquisa de Nguyen e Novák (2019). Como descrito na Seção 5.4.1 sobre a previsão recorrente, as componentes calculadas pelo método PRSSA+ENFN são realimentadas como entradas. Nesse caso, o passo é definido como $p = 1$, isto é, várias previsões de um passo a frente ao longo do horizonte de previsão H .

Em Nguyen e Novák (2019), os autores usam um procedimento semelhante à proposta da tese ao realizar a previsão de componentes separadamente *a priori* e, em seguida, agrupando os resultados individuais para prever de forma recorrente a série temporal original. O resultado é uma combinação da previsão individual do ciclo de tendência, uma componente sazonal e uma flutuação irregular, onde o ciclo de tendência e a componente sazonal são previstas baseando-se em lógica *fuzzy* e em técnicas de reconhecimento de padrões. Por outro lado, os autores aplicam a abordagem de Box-Jenkins para modelar a componente irregular.

Além do método proposto por Nguyen e Novák (2019), compara-se também os resultados do PRSSA+ENFN com os métodos SARIMA e STL (*Seasonal Trend decomposition based on Loess*). O primeiro é uma abordagem clássica que utiliza um modelo composto de termos autorregressivos, de médias móveis e operadores de diferença, assim como também considera termos autorregressivos, de médias móveis e operadores de diferença sazonais. Por sua vez, o STL é um método de previsão que é baseado na decomposição da série temporal, considerando

que esta série apresenta componentes sazonais e de tendência e realizando a previsão de ambas componentes (HYNDMAN; ATHANASOPOULOS, 2014).

Por fim, compara-se o PRSSA+ENFN com uma versão sem extração de componentes dessa mesma ENFN, com o objetivo de verificar se as componentes estão contribuindo com a melhoria da previsão das séries temporais. A configuração da ENFN é bastante similar ao PRSSA+ENFN, diferenciando no caso das entradas e na representação dos consequentes que, em vez de ser os dados das componentes, são as observações das próprias séries temporais.

À vista disso, as séries temporais univariáveis consideradas nesse experimento, a quantidade total de N observações e os horizontes de previsão H relacionados a cada uma são listados a seguir (NGUYEN; NOVÁK, 2019):

1. TS1: Quantidade mensal de pessoas desempregadas na Austrália (Set. 1978 – Ago. 1995). $N = 211$ observações. Horizonte de Previsão $H = 20$;
2. TS2: Despesa trimestral bruta de capital fixo - construção não-residencial privada, Austrália: em milhões de dólares. 3º Trimestre de 1960 a 2º Trimestre de 1991. $N = 127$ observações. Horizonte de Previsão $H = 12$;
3. TS3: Vazão média mensal em um rio, Jan. 1953 – Dez. 1981. $N = 348$ observações. Horizonte de Previsão $H = 18$;
4. TS4: Produção trimestral de tijolos de barro: milhões de unidades. (Q4 1956 - Q3 1994). $N = 155$ observações. Horizonte de Previsão $H = 16$;
5. N2090: Série temporal do desafio "M3 Competition"(MAKRIDAKIS; SPILIOTIS; ASSI-MAKOPOULOS, 2018). $N = 144$ observações. Horizonte de Previsão $H = 18$;
6. N912: Série temporal do desafio "M3 Competition". $N = 72$ observações. Horizonte de Previsão $H = 8$;
7. N2076: Série temporal do desafio "M3 Competition". $N = 144$ observações. Horizonte de Previsão $H = 18$;
8. N897: Série temporal do desafio "M3 Competition". $N = 72$ observações. Horizonte de Previsão $H = 8$.

Iniciando o pré-processamento, o primeiro passo deve ser a determinação dos parâmetros do problema de previsão para cada série temporal univariável, como apresentado na Parte 1 da Tabela 3. A Tabela 22 a seguir exhibe esses valores para o experimento dessa seção.

Tabela 22 – Valores para os Parâmetros do Problema de Previsão para cada Série Temporal Univariável do Experimento 1.

Série Temporal	H	T	N	m	p	n_{max}
TS1	20	191	211	1	1	10
TS2	12	115	127	1	1	10
TS3	18	330	348	1	1	10
TS4	16	139	155	1	1	10
N2090	18	126	144	1	1	10
N912	8	64	72	1	1	10
N2076	18	126	144	1	1	10
N897	8	64	72	1	1	10

Fonte: Autor.

Em seguida, para selecionar valores apropriados aos parâmetros críticos do método PRSSA+ENFN (Parte 2 da Tabela 3), o algoritmo genético, como apresentado na Seção 5.5.1, é inicializado para cada série temporal. Os cromossomos para este experimento têm a seguinte estrutura, de acordo com a Equação 5.46:

$$\text{cromossomo} = [W \ n \ \phi \ \eta_0], \quad (6.7)$$

onde $n = n_1$, pois é um experimento com séries univariáveis, e os intervalos de valores para os parâmetros são definidos como segue:

$$\begin{aligned} W &= 20\%N, \dots, 30\%N; \\ n &= 2, 3, \dots, n_{max} = 10; \\ 0.1 &\leq \phi \leq 0.5; \\ 0.01 &\leq \eta_0 \leq 0.3, \end{aligned} \quad (6.8)$$

sendo que W varia entre 20% e 30% do total de observações N , que é um intervalo que possibilita obter valores apropriados aos parâmetros s_j de cada componente. Os outros intervalos para os demais parâmetros críticos foram apresentados na etapa de pré-processamento do PRSSA+ENFN. Outras características do algoritmo genético em cada caso são: uma população de 100 cromossomos e uma taxa de mutação de 20%. O algoritmo genético foi aplicado nesse experimento para analisar se há um conjunto de parâmetros críticos que faça com que o PRSSA+ENFN consiga superar os resultados de Nguyen e Novák (2019).

Os resultados dos experimentos para cada série são avaliados de acordo com o critério RMSE a seguir:

$$RMSE = \sqrt{\frac{1}{H} \sum_{h=1}^H (y_{T+h} - y_{T+h}^*)^2}, \quad (6.9)$$

lembrando que H é o horizonte de previsão, T representa a quantidade de observações usadas no pré-processamento e treinamento, y_{T+h} corresponde ao valor real da série temporal durante o

horizonte de previsão, e y_{T+h}^* representa a previsão calculada pelo PRSSA+ENFN, sabendo que $h = 1, \dots, H$.

São analisadas duas perspectivas na aplicação do algoritmo genético nessa seção. Primeiramente, o Caso 1 do algoritmo genético indica a primeira geração cuja medida de erro RMSE do seu melhor cromossomo seja inferior ao apresentado no trabalho de Nguyen e Novák (2019). O intuito é verificar em qual geração os resultados do PRSSA+ENFN superam as outras abordagens. Por outro lado, o Caso 2 do algoritmo genético tem como critério de parada a formação de 500 gerações, visando obter qual melhor cromossomo do algoritmo genético após a avaliação de todas essas gerações.

Os parâmetros críticos do PRSSA+ENFN selecionados em ambos os casos do algoritmo genético e para cada série univariável são apresentados na Tabela 23. A coluna "Geração" indica a geração em que o algoritmo genético encontrou os parâmetros críticos do Caso 1, situação em que o PRSSA+ENFN supera os resultados de Nguyen e Novák (2019) pela primeira vez. Para as séries TS1, TS2, TS4, N912 e N897, logo na primeira geração se alcançou uma medida de erro menor que o alvo, por sua vez, para a série N2090 isso ocorreu na segunda geração. Em contrapartida, as séries TS3 e N2076 executaram todas as 500 gerações indicadas e, conforme Tabela 24, não apresentou o melhor resultado. Observando novamente a Tabela 23, os termos "X" informam que em nenhum momento um cromossomo conseguiu produzir uma medida RMSE menor que o alvo. Os parâmetros do Caso 2 correspondem as melhores soluções encontradas na última geração de cromossomos.

Tabela 23 – Valores para os Parâmetros Críticos do PRSSA+ENFN para cada Série Temporal Univariável do Experimento e para cada Caso da Execução do Algoritmo Genético.

Série Temporal	Caso 1					Caso 2			
	W	n	ϕ	η_0	Geração	W	n	ϕ	η_0
TS1	44	5	0.15	0.28	1	46	9	0.18	0.14
TS2	33	2	0.19	0.17	1	33	9	0.37	0.06
TS3	X	X	X	X	X	77	2	0.1	0.21
TS4	44	8	0.5	0.11	1	44	8	0.26	0.02
N2090	43	3	0.13	0.12	2	40	3	0.47	0.3
N912	21	5	0.23	0.23	1	20	6	0.48	0.28
N2076	X	X	X	X	X	43	6	0.42	0.3
N897	17	8	0.39	0.13	1	15	7	0.14	0.16

Fonte: Autor.

A Tabela 24 exibe os resultados da metodologia proposta, usando os parâmetros de ambos os casos da Tabela 23 e comparando com as três abordagens apresentadas no trabalho de Nguyen e Novák (2019). Em negrito, estão resultados de previsão do PRSSA+ENFN que superaram as outras abordagens. Percebe-se que o PRSSA+ENFN obteve os melhores resultados aplicando o critério RMSE durante o horizonte de previsão H em cada caso, exceto para as séries TS3 e N2076, onde o PRSSA+ENFN ficou em segundo lugar. Apesar de não ter sido

o melhor nesses casos, o PRSSA+ENFN apresentou resultados bem competitivos. Para estas séries, o Caso 1 não foi definido porque não houve cromossomos ou soluções que superassem os resultados de Nguyen e Novák (2019). Note que na maioria dos casos, logo nas primeiras gerações foram encontradas soluções adequadas de acordo com os critérios desse experimento para os parâmetros críticos do PRSSA+ENFN.

Tabela 24 – Resultados Comparativos entre o PRSSA+ENFN (Casos 1 e 2 do Algoritmo Genético), a ENFN (sem uso de componentes) e os Métodos Apresentados em Nguyen e Novák (2019) de acordo com o critério RMSE (resultados do PRSSA+ENFN que superaram as referências em **Negrito**).

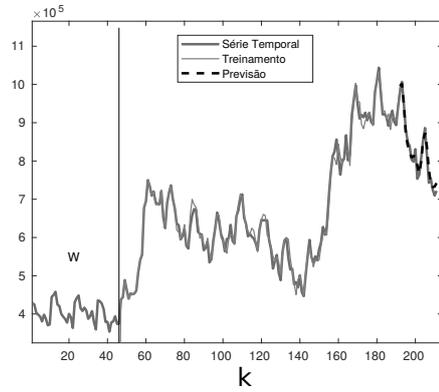
Série	Caso 1	Caso 2	ENFN (S. C.)	Nguyen e Novák (2019)	STL	SARIMA
TS1	22.363,85	13.849,31	53.218,68	47.807,00	58.662,00	89.893,00
TS2	177,47	131,33	335,94	274,01	274,39	337,08
TS3	X	13,81	14,19	10,29	16,42	15,82
TS4	20,02	15,64	47,45	23,25	29,46	81,85
N2090	3.979,05	3.293,81	3692,41	4.309,00	6.544,00	4.431,50
N912	82,93	44,36	156,63	218,59	158,29	205,81
N2076	X	151,97	184,91	65,48	206,18	171,35
N897	185,85	170,86	301,99	199,18	248,58	305,69

Fonte: Autor.

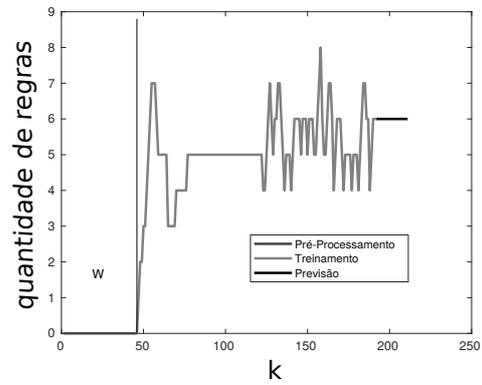
Em relação a ENFN sem componentes, esta apresentou resultados de previsão inferiores aos PRSSA+ENFN para todos esses casos apresentados, o que reforça a vantagem do “dividir para conquistar”, ou seja, dividir um conjunto de dados da série temporal mais complexa em componentes mais simples. Contudo, foi melhor que os outros métodos de referência para os casos N2090 e N912.

Pode-se visualizar a partir das Figuras 34 e 35 os resultados de previsão recorrente univariável para as séries temporais dessa seção, com base nas Tabelas 23 e 24 e no Caso 2. Em azul, tem-se a série temporal original, a cor vermelha corresponde ao treinamento e em preto estão os resultados de previsão. A janela inicial W também está representada nos gráficos antes da linha vertical. É interessante visualizar também a evolução da quantidade de regras *fuzzy* ou *clouds* (R) em cada caso, observando o número de regras ao longo do treinamento além da sua quantidade na etapa de de previsão, destacando que a estrutura da rede neuro-*fuzzy* se mantém fixa nessa etapa. Note que essa quantidade pode aumentar ou diminuir na etapa de treinamento, conforme discutido sobre as características dos sistemas evolutivos que modificam sua estrutura para se adaptarem a dinâmica dos dados. A Figura 34.(e) exhibe os resultados da série TS3, que possui comportamento bastante aleatório, fazendo com que o PRSSA+ENFN tivesse certa dificuldade em modelar seus padrões. Assim como para a série N2076 da Figura 35.(e), sendo que o PRSSA+ENFN representou a dinâmica de oscilação no horizonte de previsão, porém a amplitude dos dados maior do que esperado pelo PRSSA+ENFN.

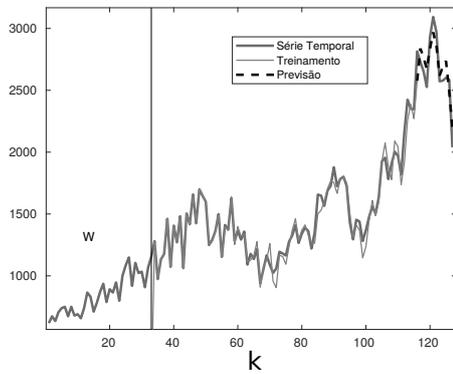
Figura 34 – Resultados de Previsão para as Séries Temporais Univariáveis Relacionados às Tabelas 23 e 24 Aplicando o Caso 2 do Algoritmo Genético (**Primeira Parte**).



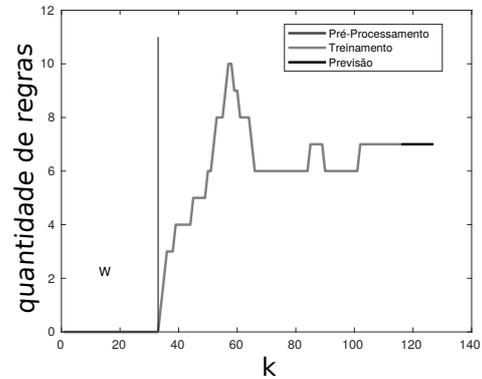
(a) Série TS1 ($h = 20$).



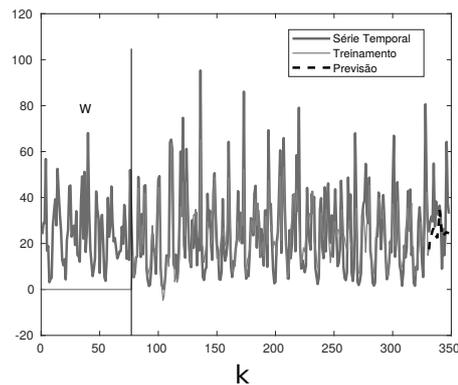
(b) Quantidade de Regras *Fuzzy* para TS1.



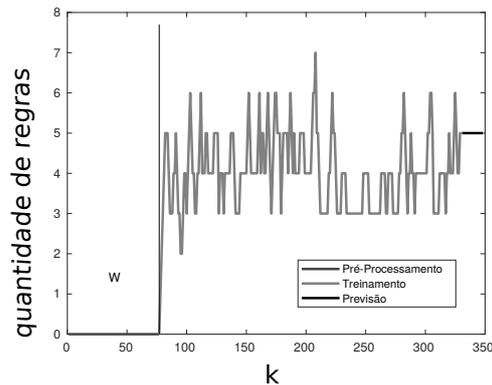
(c) Série TS2 ($h = 12$).



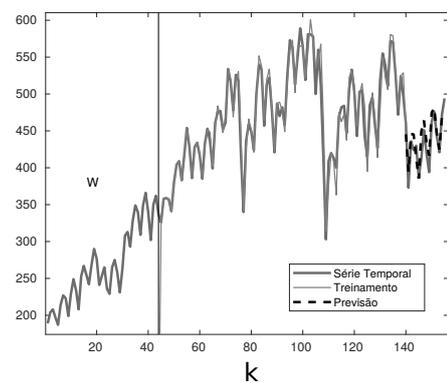
(d) Quantidade de Regras *Fuzzy* para TS2.



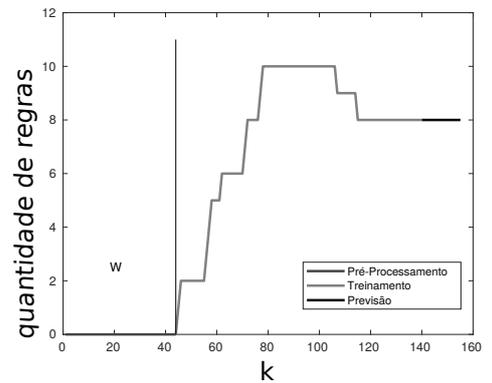
(e) Série TS3 ($h = 18$).



(f) Quantidade de Regras *Fuzzy* para TS3.

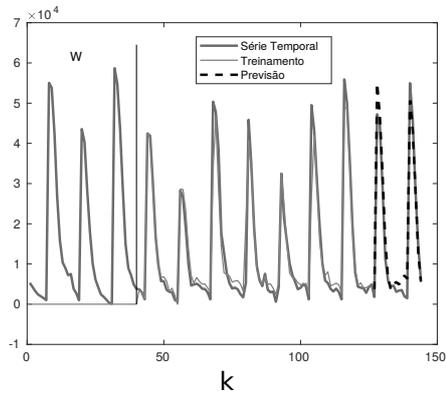


(g) Série TS4 ($h = 16$).

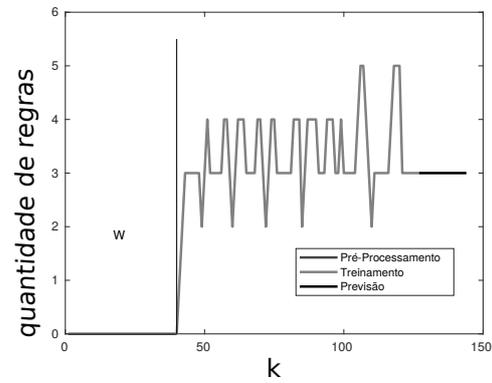


(h) Quantidade de Regras *Fuzzy* para TS4.

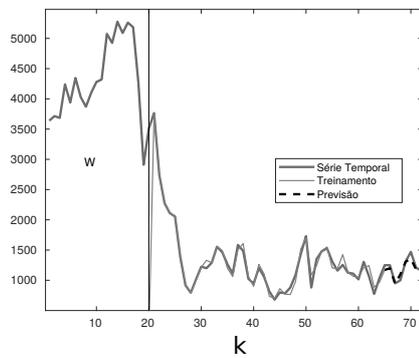
Figura 35 – Resultados de Previsão para as Séries Temporais Univariáveis Relacionados às Tabelas 23 e 24 Aplicando o Caso 2 do Algoritmo Genético (**Segunda Parte**).



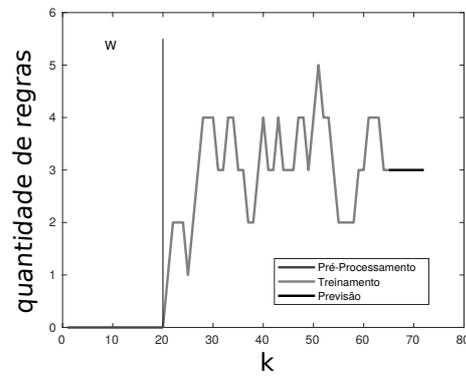
(a) Série N2090 ($h = 18$).



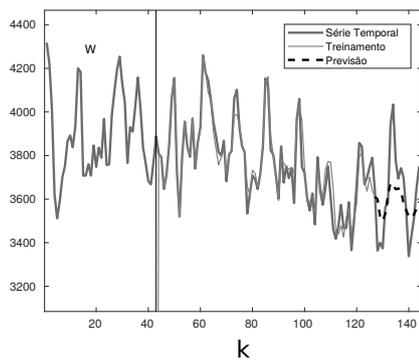
(b) Quantidade de Regras *Fuzzy* para N2090.



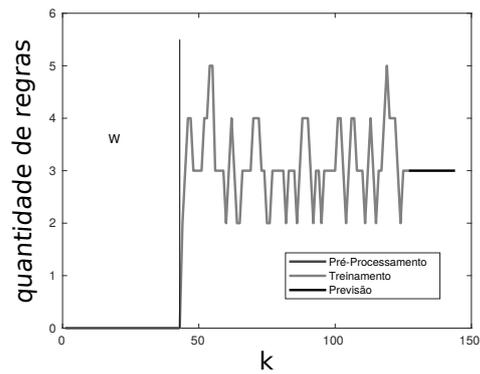
(c) Série N912 ($h = 8$).



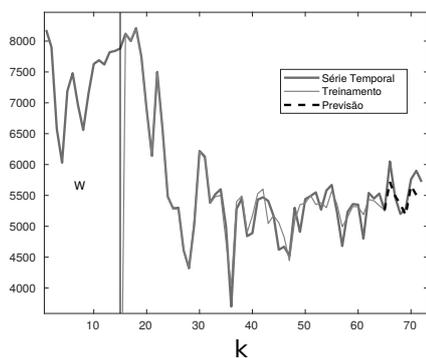
(d) Quantidade de Regras *Fuzzy* para N912.



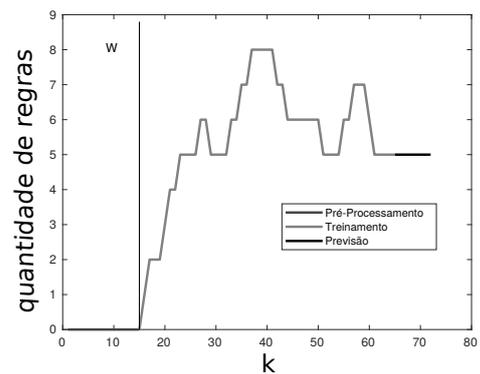
(e) Série N2076 ($h = 18$).



(f) Quantidade de Regras *Fuzzy* para N2076.

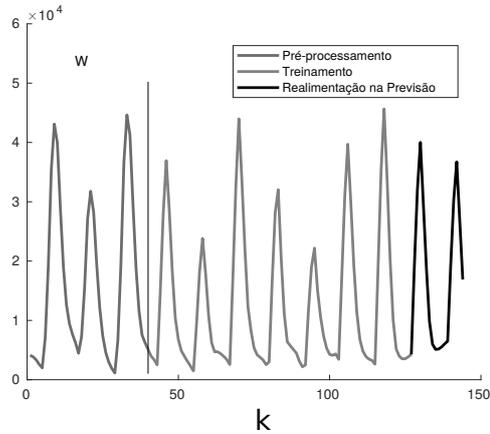


(g) Série N897 ($h = 8$).

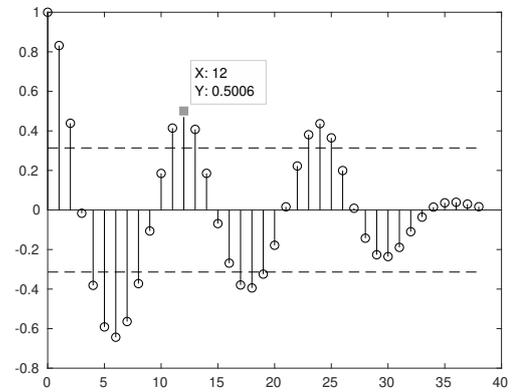


(h) Quantidade de Regras *Fuzzy* para N897.

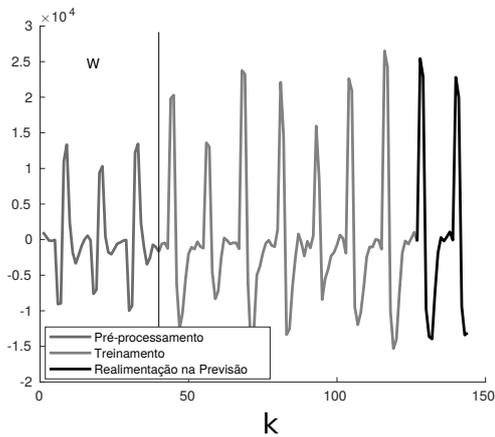
Figura 36 – Componentes Não-Observáveis da Série Temporal N2090 nas Etapas de Pré-Processamento, Treinamento Incremental e Previsão Aplicando o Caso 2 do Algoritmo Genético.



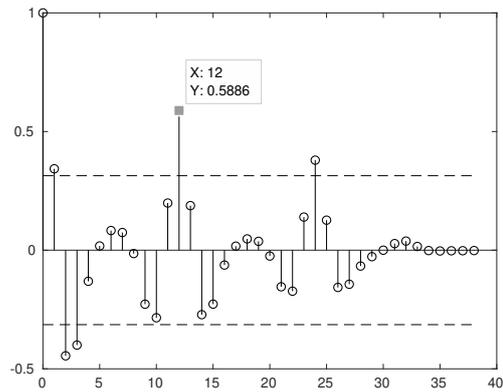
(a) Componente 1 da série N2090.



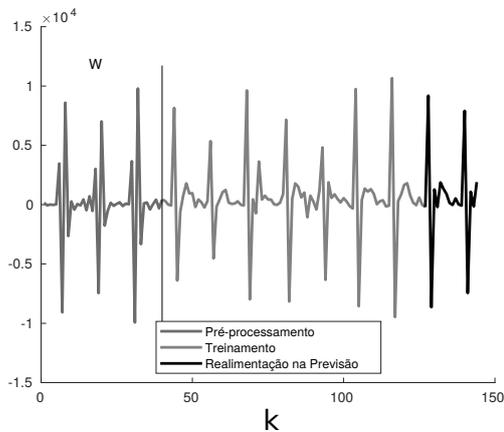
(b) Função de Autocorrelação da Componente 1 da N2090.



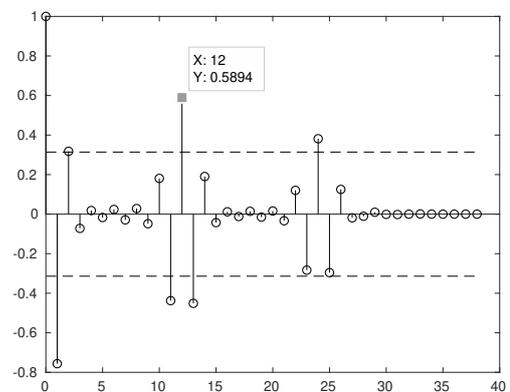
(c) Componente 2 da série N2090.



(d) Função de Autocorrelação da Componente 2 da N2090.



(e) Componente 3 da série N2090.

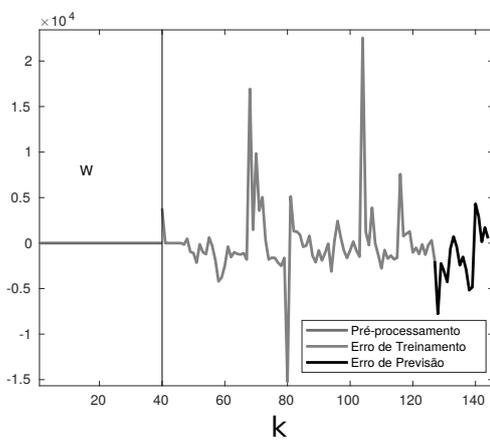


(f) Função de Autocorrelação da Componente 3 da N2090.

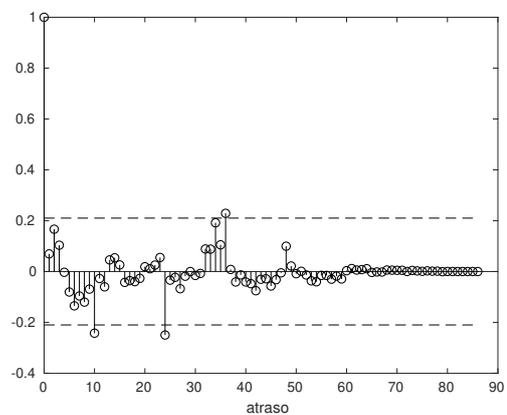
Fonte: Autor.

Por sua vez, a Figura 36 traz com mais detalhes os resultados envolvendo a série temporal N2090, exibindo as previsões das três componentes não-observáveis e suas funções de autocorrelação relacionadas. Conforme descrito no Capítulo 5, o PRSSA+ENFN realiza a previsão separadamente de cada componente e em seguida agrupa esses resultados para obter a previsão da série temporal original. Em azul, estão os dados das componentes no pré-processamento para as W primeiras observações. Em vermelho, o resultado da extração das componentes de forma recursiva é representado e, em preto, estão as componentes computadas pelo PRSSA+ENFN e que são realimentadas durante a etapa de previsão. Pelas funções de autocorrelação, percebe-se que os valores de $s_1 = s_2 = s_3 = 12$.

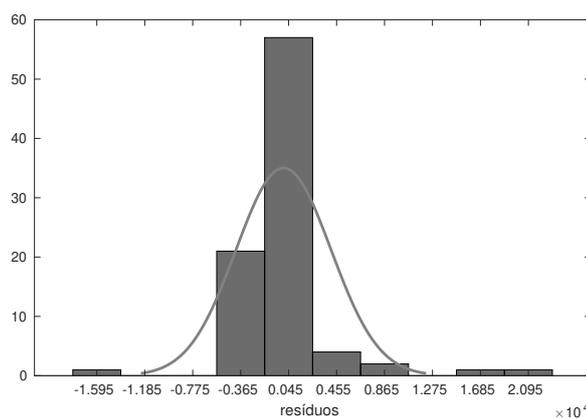
Figura 37 – Análise dos Erros de Treinamento e Previsão para a Série N2090 Aplicando o Caso 2 do Algoritmo Genético.



(a) Erros em Cada Etapa do PRSSA+ENFN.



(b) Função de Autocorelação Amostral dos Resíduos de Treinamento.



(c) Histograma dos Resíduos de Treinamento.

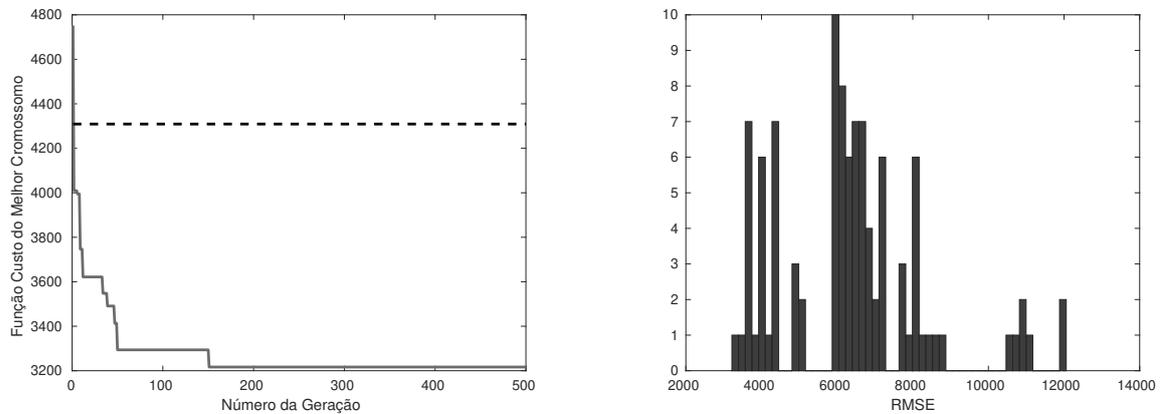
Fonte: Autor.

A Figura 37.(a) apresenta os erros de treinamento ou resíduos (em vermelho) e os de previsão (em preto). Em azul, não há a indicação do erro, pois no pré-processamento o PRSSA+ENFN não computa valores para a série temporal. A função de autocorrelação amostral

dos resíduos está representada na Figura 37.(b), sendo que há alguns pontos que passam um pouco dos limites das linhas tracejadas, indicando que os resíduos apresentam certa correlação. A média desses resíduos é diferente de zero, logo, existe um viés de valor 13.05. Essas características indicam que ainda é possível gerar um modelo que se ajusta melhor aos dados de treinamento, apesar disso, o desempenho do treinamento é considerado aceitável, fazendo com que o PRSSA+ENFN apresentasse erros inferiores às outras abordagens de previsão em Nguyen e Novák (2019). Além disso, adicionou-se o viés obtido no treinamento e na previsão para esse caso da série N2090 e para todas as outras séries temporais desse experimento, porém, não houve uma influência significativa nos resultados de previsão apresentados na Tabela 24. O histograma dos resíduos é apresentado na Figura 37.(c), onde fez-se uma tentativa de aproximar uma distribuição normal, sendo que é possível notar que não há uma variância constante e também não chega a ser perfeitamente uma distribuição com comportamento normal. No entanto, como comentado na Seção 2.3.4, um método que possua variância constante e resíduos com normalmente distribuídos nem sempre pode ser melhorado, mas é uma propriedade que pode ser útil. Por meio da análise dessas figuras, conclui-se que pode-se produzir um modelo que pode ser melhor ajustado aos dados de treinamento, mas ressalta-se que nem sempre o modelo com melhores comportamentos dos resíduos produzem os melhores resultados de previsão.

Finalmente, a Figura 38.(a) demonstra a evolução da função custo, no caso o RMSE, do melhor cromossomo ao longo das gerações para a série temporal N2090. Note que na geração 2 o cromossomo passa a indicar um conjunto de parâmetros críticos do PRSSA+ENFN que supera os resultados dos outros métodos para comparação, representado pela linha tracejada. A Figura 38.(b) exibe por meio de um histograma o valor de RMSE da população final após as 500 gerações, mostrando que outros cromossomos também apresentaram valores de RMSE inferiores a 4309, conforme Tabela 24.

Figura 38 – Informações do Melhor Cromossomo e da População Final do Algoritmo Genético no Caso 2 para a Série Temporal N2090.



(a) Evolução do RMSE do Melhor Cromossomo pelas Gerações. (b) Histograma das Medidas RMSE dos Cromossomos da Geração 500.

Fonte: Autor.

Essas mesmas ilustrações detalhadas acima para as demais séries temporais desse experimento são mostradas no Apêndice C.2.

6.4 Experimento 2: Previsão Direta Univariável de Séries Temporais

Esse estudo utilizou o total de 11 séries temporais univariáveis, encontradas em Hyndman (2010). O método PRSSA+ENFN foi executado nesse caso para efetuar a previsão do tipo direta dessas séries, comparando com outras 15 técnicas híbridas na literatura, cujos resultados de previsão foram apresentados no trabalho de Oliveira et al. (2020). A seguir, uma breve descrição de cada uma dessas abordagens que são consideradas para comparação com o PRSSA+ENFN:

1. *Hybrid semi-linear system (Hyselis)* (OLIVEIRA et al., 2017): Uso de uma rede neural artificial semi-linear evolucionária para previsão de séries temporais. Aplicação do algoritmo PSO para encontrar a arquitetura e pesos adequados para essa rede neural;
2. ARIMA (HYNDMAN; KHANDAKAR, 2008): Implementação do modelo ARIMA otimizado usando o pacote *Forecast Package* para linguagem R;
3. Babu (BABU; REDDY, 2014): Técnica híbrida combinando o modelo ARIMA e uma Rede Neural Artificial para previsão de séries temporais;
4. PSO-ARIMA-SVR (OLIVEIRA; LUDERMIR, 2014): Sistema híbrido composto por modelos ARIMA e SVR otimizados pelo algoritmo PSO para realizar as previsões de séries temporais;

5. Yolcu (YOLCU; EGRIOGLU; ALADAG, 2013): Rede Neural otimizada pelo algoritmo PSO para previsão de séries temporais baseadas em suas componentes lineares e não-lineares;
6. Zhang (ZHANG, 2003): Metodologia híbrida que combina os modelos ARIMA e Redes Neurais Artificiais para previsão de séries lineares e não-lineares;
7. Sum (OLIVEIRA et al., 2020): Sistema híbrido proposto que realiza a previsão e combinação dos componentes lineares e não-lineares empregando a otimização pelo PSO. Uma rede neural linear (*Linear*) e outra não-linear (*NonLinear*) são usadas para prever a série temporal e os erros, respectivamente. Por fim, a combinação de ambos os resultados das redes neurais é efetuada por meio da soma de suas previsões (*Sum*);
8. Panigrahi (PANIGRAHI; BEHERA, 2017): Metodologia híbrida que combina modelos de suavização exponencial com Redes Neurais Artificiais;
9. PSO-ARIMA-AR-SVR (OLIVEIRA; LUDERMIR, 2016): Sistema evolucionário híbrido composto por um filtro de suavização exponencial simples, modelos ARIMA e AR e um modelo SVR. O PSO é empregado para otimizar a ordem do modelo AR, os parâmetros do SVR e o número de *lags* ou atrasos das séries temporais;
10. Wang (WANG et al., 2013): Proposta de um modelo híbrido, que se diferencia por integrar as vantagens do ARIMA e das Redes Neurais Artificiais na modelagem dos comportamentos lineares e não-lineares no conjunto de dados;
11. Khandelwal (KHANDELWAL; ADHIKARI; VERMA, 2015): Este trabalho sugere uma nova técnica de previsão pela segregação de um conjunto de dados de série temporal em componentes lineares e não-lineares por meio do método DWT (*Discrete Wavelet Transform*). Então, os modelos ARIMA e de Redes Neurais Artificiais são aplicados para reconhecer e prever separadamente as componentes, respectivamente;
12. Merh (MERH; SAXENA; PARDASANI, 2010): Modelo híbrido composto por uma Rede Neural Artificial de três camadas e abordagem ARIMA aplicado nesse estudo para previsão da bolsa indiana;
13. Ozozen (OZOZEN et al., 2016): Este trabalho propõe uma metodologia híbrida que utiliza os modelos SARIMA e Rede Neural Artificial, sendo que a previsão é realizada a partir de uma série temporal do mercado turco de energia;
14. Aslanargun (ASLANARGUN et al., 2007): apresentação de modelo híbrido usando modelos ARIMA juntamente com uma MLP (*Multilayer Perceptron*) prever chegadas de turistas à Turquia.
15. MLP (OLIVEIRA et al., 2020): uso da MLP no trabalho de Oliveira et al. (2020) para combinação das previsões das componentes lineares e não-lineares.

Esse experimento é realizado nas mesmas condições apresentadas em Oliveira et al. (2020) para todos os métodos, onde 80% dos dados são aplicados na fase de treinamento incremental do PRSSA+ENFN e os 20% restantes são usados para verificar o desempenho da previsão. Outra característica importante dos dados é que estes estão todos normalizados no intervalo $[0,1]$.

Assim como no Experimento 1 (Seção 6.3), o PRSSA+ENFN foi executado tanto em sua versão original com a decomposição das componentes não-observáveis, quanto na versão sem componentes, usando os próprios dados originais da série temporal. Ambas abordagens foram otimizadas segundo o algoritmo genético conforme definido na metodologia, com a função custo sendo o critério MSE (Equação 2.15). O algoritmo genético foi configurado com uma população de 30 cromossomos que evoluíram através de 100 gerações, com a taxa de mutação em 20%. Esses parâmetros do GA foram assim definidos para se assemelhar às configurações da otimização dos métodos a serem comparados, conforme em Oliveira et al. (2020).

Tabela 25 – Descrição das Séries Temporais para Estudo e os Parâmetros do Problema de Previsão.

Série	Descrição	N	T	H
<i>Accidental</i>	Mortes acidentais mensais nos EUA	72	63	9
<i>Airlines</i>	Passageiros de companhias aéreas internacionais	144	121	23
<i>Red Wine</i>	Vendas mensais de vinho australiano	187	155	32
<i>Car Sales</i>	Vendas mensais de carros em Quebec	108	86	22
<i>Paper</i>	Vendas da indústria de papel para impressão e escrita	120	96	24
<i>Pigs</i>	Número total mensal de porcos abatidos em Victoria	188	150	38
<i>Pollution</i>	Remessa mensal de equipamentos de poluição	130	110	20
<i>Gas</i>	Demanda mensal de gasolina em Ontario	192	153	39
<i>Sunspot</i>	Número de Wolf das manchas solares	289	237	52
<i>Lake</i>	Níveis mensais do Lago Erie	600	486	114
<i>Lynx</i>	Número anual de linces capturados	105	88	17

Fonte: Adaptado de Oliveira et al. (2020).

A Tabela 25 exibe informações sobre as séries temporais que serão estudadas, assim como os parâmetros do problema de previsão N , T e H (Parte 1 da Tabela 3). O parâmetro $m = 1$ pois é um experimento com séries univariáveis, $p = 1$ para que o PRSSA+ENFN atue da mesma forma que os métodos para comparação e $n_{max} = 10$.

Por sua vez, as Tabelas 26 e 27 exibem os critérios MSE médio para 30 execuções com os melhores resultados de cada método. Essas tabelas foram adaptadas do trabalho de Oliveira et al. (2020), inserindo os resultados do PRSSA+ENFN com componentes e da ENFN sem extração de componentes. Da mesma forma que os outros métodos, os 30 melhores resultados de ambos os métodos foram coletados e, em seguida, computou-se a média do critério MSE para apresentação nas Tabelas 26 e 27.

A partir das Tabelas 26 e 27, nota-se que o PRSSA+ENFN obteve o menor valor de

MSE médio para quase todas as séries temporais do estudo, exceto para as séries *Car Sales* e *Pigs*. No caso da série *Pigs*, o PRSSA+ENFN apresentou o terceiro menor valor de MSE médio, entretanto, para a série *Car Sales*, o PRSSA+ENFN não esteve entre as melhores abordagens listadas, provando mais uma vez não há um único melhor método para todas as séries temporais, conforme também pondera Oliveira et al. (2020).

Também efetuou-se a execução da ENFN sem a extração de componentes, e verificou-se que em nenhum dos casos chegou a superar o método PRSSA+ENFN. Como há poucas amostras em todas as séries temporais, o custo computacional a mais da extração de componentes não tem impacto importante na execução.

Sendo assim, a partir desse experimento verifica-se que o PRSSA+ENFN apresentou resultados competitivos em relação a outros vários métodos híbridos na literatura. Contudo, ressalta-se novamente que o ideal em um estudo de previsão de séries temporais é estudar vários métodos em conjunto, pois podem haver casos que uma das abordagens consideradas não apresentem os melhores resultados de previsão.

Tabela 26 – Resultados de Previsão do Experimento 2 de acordo o Critério MSE Médio a partir de 30 Execuções de cada Método (menores erros em **negrito**) - Parte 1.

Método (Num.)	<i>Accidental</i>	<i>Airlines</i>	<i>Red Wine</i>	<i>Car Sales</i>	<i>Paper</i>	<i>Pigs</i>
Hyelis (1)	0.0051	0.0014	0.0105	0.0191	0.0132	0.0122
ARIMA (2)	0.0273	0.0088	0.0299	0.0279	0.0493	0.0129
Babu (3)	0.0275	0.0031	0.0161	0.0289	0.0122	0.0208
PSO-ARIMA-SVR (4)	0.0062	0.0017	0.0154	0.0434	0.0174	0.0132
Yolcu (5)	0.0146	0.0051	0.0147	0.0212	0.0194	0.0135
Zhang (6)	0.0121	0.0052	0.0161	0.0427	0.0101	0.0162
Sum (7)	0.0135	0.0011	0.0101	0.0074	0.0141	0.0114
Panigrahi (8)	0.0079	0.0012	0.0116	0.0530	0.0080	0.0089
PSO-ARIMA-AR-SVR (9)	0.0034	0.0016	0.0201	0.0182	0.0177	0.0125
Wang (10)	0.0121	0.0020	0.0136	0.0293	0.0122	0.0133
Khandelwal (11)	0.0119	0.0138	0.0195	0.0095	0.0066	0.0071
Merh (12)	0.0138	0.0091	0.0314	0.0173	0.0148	0.0135
Ozozen (13)	0.0146	0.0028	0.0141	0.0304	0.0099	0.0124
Aslanargun (14)	0.0168	0.0043	0.0200	0.0112	0.0124	0.0129
MLP (15)	0.0161	0.0202	0.0253	0.0176	0.0151	0.0141
PRSSA+ENFN	0.0030	0.0011	0.0078	0.0481	0.0058	0.0097
ENFN (S. Comp.)	0.0124	0.0044	0.0193	0.0624	0.0158	0.0128

Fonte: Adaptado de Oliveira et al. (2020).

Tabela 27 – Resultados de Previsão do Experimento 2 de acordo o Critério MSE Médio a partir de 30 Execuções de cada Método (menores erros em **negrito**) - Parte 2.

Método (Num.)	<i>Pollution</i>	<i>Gas</i>	<i>Sunspot</i>	<i>Lake</i>	<i>Lynx</i>
Hyelis (1)	0.0206	0.0038	0.0207	0.0015	0.0182
ARIMA (2)	0.0186	0.0102	0.0114	0.0013	0.0150
Babu (3)	0.0622	0.0043	0.0171	0.0023	0.0228
PSO-ARIMA-SVR (4)	0.0218	0.0045	0.0144	0.0013	0.0267
Yolcu (5)	0.0234	0.0090	0.0163	0.0033	0.0124
Zhang (6)	0.0589	0.0086	0.0137	0.0013	0.0213
Sum (7)	0.0166	0.0040	0.0128	0.0015	0.0138
Panigrahi (8)	0.0180	0.0038	0.0108	0.0019	0.0205
PSO-ARIMA-AR-SVR (9)	0.0168	0.0092	0.0148	0.0012	0.0099
Wang (10)	0.0211	0.0051	0.0354	0.0014	0.0462
Khandelwal (11)	0.0921	0.0165	0.0284	0.0011	0.0132
Merh (12)	0.0592	0.0129	0.0219	0.0015	0.0331
Ozozen (13)	0.0230	0.0055	0.0159	0.0014	0.0265
Aslanargun (14)	0.0268	0.0070	0.0181	0.0014	0.0534
MLP (15)	0.1422	0.0253	0.0206	0.0015	0.0259
PRSSA+ENFN	0.0104	0.0030	0.0099	0.0011	0.0068
ENFN (S. Comp.)	0.0182	0.0087	0.0178	0.0017	0.0136

Fonte: Adaptado de Oliveira et al. (2020).

6.5 Experimento 3: Previsão Direta de Séries Temporais Multivariáveis

O experimento apresentado nessa seção compara o desempenho do método PRSSA+ENFN com os resultados do trabalho de Du et al. (2020), que aplica uma previsão do tipo direta (Seção 5.4.2). Três séries temporais multivariáveis são consideradas neste estudo. O primeiro conjunto de dados está relacionado ao parâmetro PM_{2.5} (partículas inaláveis com de diâmetro inferior a 2.5 micrômetros) da cidade de Pequim, que consiste em diversas variáveis, por exemplo, temperatura, pressão, direção do vento, velocidade do vento, condições de neve ou chuva e o valor de PM_{2.5}. Este conjunto de dados multivariável contém $N = 43.824$ observações e $m = 9$ variáveis ou séries temporais (LIANG et al., 2015). Em segundo lugar, é analisado o conjunto de dados sobre o Consumo de Energia Elétrica Domiciliar Individual. Este conjunto de dados contém $N = 2.075.259$ amostras e $m = 7$ séries temporais coletadas em uma casa localizada em Sceaux, França, entre dezembro de 2006 e novembro de 2010 (47 meses) (DU et al., 2020). Por último, o conjunto de dados da Qualidade do Ar apresenta $N = 9.358$ observações e $m = 15$ séries temporais medidas a partir de 5 sensores químicos de óxido de metal localizados em uma área poluída de uma cidade italiana não especificada (VITO et al., 2008).

Em Du et al. (2020), diversos métodos foram implementados pelos autores para avaliar o desempenho da previsão em várias etapas. Os métodos considerados no trabalho desses autores foram: os modelos estatísticos clássicos ARIMA (*Autoregressive Integrated Moving Average*), VARMA (*Vector Autoregressive Moving Average*) e SVR (*Support Vector Regression*); os mode-

los de redes neurais RNN (*Recurrent Neural Network*), CNN (*Convolutional Neural Network*), LSTM (*Long Short-Term Memory*), GRU (*Gated Recurrent Unit*); os métodos SEQ2SEQ (*Sequence-to-Sequence Deep Learning Model*) e SEQ2SEQ-BI (*Sequence-to-Sequence Deep Learning Model and Bi-LSTM*), que são abordagens que usam *deep learning*; e SEQ2SEQ-ATT que é uma variante mais simples do método MTSMFF, proposto por Du et al. (2020) e também abordado na comparação. Portanto, o método PRSSA+ENFN proposto foi comparado com essas abordagens mencionadas acima e, assim como os experimentos anteriores, foi verificado os resultados da ENFN sem a as componentes não-observáveis.

Para todos os métodos mencionados em Du et al. (2020), o conjunto de dados foi dividido em 80% para treinamento e 20% para teste. Porém, para a abordagem PRSSA+ENFN, esses 80% foi distribuído com 30% para a etapa de pré-processamento e 50% para treinamento, mas os mesmos 20% de dados foram aplicados para teste ou etapa de previsão, comparando o desempenho dos métodos usando RMSE e MAE como métricas de avaliação dos erros de previsão.

Nesse estudo, o PRSSA+ENFN trabalha com previsões do tipo direta. Esta condição é aplicada para comparar o método proposto com os resultados do artigo de Du et al. (2020), que também considera este tipo de previsão. A previsão recorrente não é apropriada nesse experimento, pois o horizonte de previsão é bastante longo, sendo que a propagação dos erros comprometeria os resultados. Seguindo as mesmas condições dessa referência, a previsão neste experimento pode ser de 1 a 6 passos à frente. Portanto, o passo p do método PRSSA+ENFN pode assumir os valores de 1 a 6 para definição do mapeamento, similarmente aos métodos apresentados em Du et al. (2020). Após a etapa de pré-processamento com 30% dos dados iniciais, o PRSSA+ENFN é treinado para mapear $k - p \rightarrow k$ utilizando a parte dos dados de treinamento, e por fim realiza a previsão direta considerando os dados de teste.

Um diferencial importante entre o PRSSA+ENFN e os demais trazidos por Du et al. (2020) é que este último considera a previsão de uma variável alvo, isto é, as abordagens levam em conta os dados das m séries disponíveis para prever uma específica nesse conjunto. Por sua vez, o PRSSA+ENFN realiza a previsão de todas as m séries temporais. Todavia, para comparação entre os resultados das pesquisas, somente a variável alvo será considerada em cada um dos 3 casos, que são, respectivamente, o *Valor de PM2.5*, o *Consumo de Energia* e o *Índice de Qualidade do Ar* em uma cidade italiana.

Definido o problema de previsão, o algoritmo genético, usando as mesmas configurações do Experimento 1 e Experimento 2 é aplicado como função de aptidão o critério RMSE, executando para encontrar os parâmetros apropriados para o PRSSA+ENFN para cada um dos 3 casos especificados nessa seção. Os cromossomos apresentam a seguinte estrutura:

$$\text{cromossomo} = [W \ n_1 \ n_2 \ \cdots \ n_m \ \phi \ \eta_0], \quad (6.10)$$

sendo que:

$$\begin{aligned}
 W &= 20\%N, \dots, 30\%N; \\
 n_l &= 2, 3, \dots, n_{max} = 10, \text{ para } l = 1, 2, \dots, m; \\
 0.1 &\leq \phi \leq 0.5; \\
 0.01 &\leq \eta_0 \leq 0.3.
 \end{aligned}
 \tag{6.11}$$

As Tabelas 28, 29, e 30 são apresentadas em Du et al. (2020), contudo, os resultados do método PRSSA+ENFN foram incluídos nessa tese. A Tabela 28 mostra os resultados comparativos das médias dos critérios RMSE e MAE dos modelos descritos. Essas medidas correspondem a média dos erros de previsão do passo $p = 1$ até o passo $p = 6$, ou seja:

$$\begin{aligned}
 \overline{RMSE} &= \frac{1}{6} \sum_{r=1}^6 RMSE_r \\
 \overline{MAE} &= \frac{1}{6} \sum_{r=1}^6 MAE_r,
 \end{aligned}
 \tag{6.12}$$

onde \overline{RMSE} é a média e o $RMSE_r$ é o critério de erro para o passo $p = r$ (procedimento semelhante para o \overline{MAE} e MAE_r). De acordo com a Tabela 28, o método PRSSA+ENFN proposto possui os menores valores de erro médio em relação aos outros apresentados em Du et al. (2020).

Tabela 28 – Comparação do RMSE Médio e do MAE Médio do Passo $p = 1$ ao Passo $p = 6$ considerando o Método PRSSA+ENFN e Outras Abordagens em Du et al. (2020) (melhores resultados em **negrito**).

Métodos	Pequim PM2.5		Consumo de Energia		Qualidade do Ar	
	<i>RMSE</i>	<i>MAE</i>	<i>RMSE</i>	<i>MAE</i>	<i>RMSE</i>	<i>MAE</i>
ARIMA	53.41	41.06	0.87	0.69	59.07	50.66
VARMA	51.78	38.95	0.79	0.66	58.44	47.32
SVR-LINEAR	88.35	71.48	1.45	1.23	72.39	65.48
SVR-RBF	57.64	43.47	0.76	0.71	59.02	48.49
RNN	44.42	27.99	0.46	0.23	53.35	40.16
CNN	42.98	26.89	0.46	0.24	54.13	41.07
LSTM	43.12	27.04	0.44	0.23	55.19	41.60
GRU	43.04	26.95	0.47	0.25	55.02	41.36
SEQ2SEQ	42.53	26.42	0.46	0.24	54.89	41.25
SEQ2SEQ-BI	42.29	26.31	0.44	0.23	54.45	41.21
SEQ2SEQ -ATT	42.15	26.29	0.44	0.23	54.22	41.18
MTSMFF	39.83	25.45	0.42	0.21	53.28	40.25
ENFN (S. C.)	44.99	28.17	0.58	0.43	65.52	56.19
PRSSA+ENFN	32.61	21.93	0.39	0.19	44.79	33.69

Fonte: Adaptado de Du et al. (2020).

Tabela 29 – Comparação do PRSSA+ENFN com Outros Métodos para o Conjunto de Dados Pequim PM2.5 Considerando o Critério RMSE e Cada Passo $p = 1, \dots, 6$. (melhores resultados em **negrito**).

Métodos	Pequim PM2.5					
	$k + 1$	$k + 2$	$k + 3$	$k + 4$	$k + 5$	$k + 6$
ARIMA	24.21	41.17	56.43	62.96	65.28	70.38
VARMA	24.10	40.05	55.27	60.36	62.46	68.47
SVR-LINEAR	38.73	47.39	66.3	163.5	120.34	93.89
SVR-RBF	45.02	49.97	55.29	61.66	64.52	69.41
RNN	24.93	35.23	43.27	50.11	54.49	58.48
CNN	24.57	34.26	41.98	48.02	52.68	56.37
LSTM	24.16	34.35	42.06	48.14	53.04	57.01
GRU	23.99	34.34	41.95	48.08	52.94	56.97
SEQ2SEQ	24.42	33.36	41.83	48.07	52.17	55.34
SEQ2SEQ-BI	23.86	33.42	42.05	48.13	52.11	54.20
SEQ2SEQ -ATT	23.94	33.87	42.33	48.09	51.25	53.41
MTSMFF	23.52	33.06	41.09	47.12	48.05	46.15
ENFN (S. C.)	27.87	37.59	47.08	55.78	52.18	50.47
PRSSA+ENFN	23.68	33.28	34.07	34.60	34.95	35.10

Fonte: Adaptado de Du et al. (2020).

Além disso, as Tabelas 29 e 30 mostram resultados detalhados de cada passo $p = 1, \dots, 6$ definido para os conjuntos de dados Pequim PM2.5 e Consumo de Energia, respectivamente. Considerando o conjunto de dados Pequim PM2.5, o método MTSMFF apresentou a menor medida RMSE para $p = 1$ e para $p = 2$. Porém, para os demais casos, o PRSSA+ENFN obteve medidas menores para o critério RMSE. Observe que os resultados do método PRSSA+ENFN foram consistentes para este conjunto de dados, apesar do aumento do valor de p .

Tabela 30 – Comparação do PRSSA+ENFN com Outros Métodos para o Conjunto de Dados Consumo de Energia Considerando o Critério RMSE e Cada Passo p . (melhores resultados em **negrito**).

Métodos	Consumo de Energia					
	$k + 1$	$k + 2$	$k + 3$	$k + 4$	$k + 5$	$k + 6$
ARIMA	0.75	0.82	0.89	0.90	0.94	0.96
VARMA	0.71	0.73	0.79	0.82	0.84	0.85
SVR-LINEAR	0.97	1.44	2.52	0.88	1.49	1.40
SVR-RBF	0.67	0.74	0.78	0.77	0.78	0.83
RNN	0.29	0.41	0.48	0.50	0.54	0.57
CNN	0.28	0.40	0.46	0.51	0.53	0.56
LSTM	0.27	0.38	0.44	0.48	0.52	0.54
GRU	0.29	0.40	0.48	0.52	0.54	0.57
SEQ2SEQ	0.29	0.40	0.46	0.50	0.53	0.56
SEQ2SEQ-BI	0.28	0.39	0.46	0.48	0.51	0.55
SEQ2SEQ -ATT	0.28	0.40	0.45	0.47	0.50	0.54
MTSMFF	0.26	0.37	0.42	0.45	0.50	0.52
ENFN (S. C.)	0.43	0.46	0.57	0.55	0.67	0.79
PRSSA+ENFN	0.27	0.38	0.38	0.48	0.44	0.40

Fonte: Adaptado de Du et al. (2020).

Por outro lado, de acordo com a Tabela 30, o PRSSA+ENFN apresenta os menores erros nos passos $p = 3, 5$ e 6 , enquanto o método MTSMFF apresentou melhor desempenho nos demais valores de p . Analisando essas tabelas, os métodos PRSSA+ENFN e MTSMFF apresentam resultados semelhantes e desempenhos equilibrados, ambos sendo alternativas de previsão com resultados promissores.

Nesse experimento, o tempo de execução foi mais longo, principalmente devido ao pré-processamento, pois o valor de W foi elevado para todos os conjuntos de dados analisados, parâmetro que apresenta ordem cúbica da notação de complexidade computacional.

Por fim, o Apêndice C.3 exibe os valores dos parâmetros críticos encontrados pelo algoritmo genético associados a esses resultados apresentados nessa seção.

6.6 Experimento 4: Previsão Multivariável de Séries Temporais dos Números de Casos e Mortes Causadas pela Covid-19 no Estado do Maranhão

O objetivo nesse experimento é analisar o desempenho do PRSSA+ENFN em um caso prático e desafiador, envolvendo dados da Covid-19 no Estado do Maranhão, Brasil. Por meio do PRSSA+ENFN, busca-se realizar a previsão de 30 ($H = 30$) ou 60 ($H = 60$) dias a frente para diferentes períodos de duas séries temporais relacionadas ao número diário de casos e mortes devido a essa doença (COVID-19. . . , 2021). Nesse caso, o PRSSA+ENFN realiza uma previsão multivariável ($m = 2$) de forma direta, sendo $p = 30$ quando $H = 30$, e $p = 60$ quando $H = 60$. Logo, o método é preparado para mapear os dados do instante $k - p$ ao instante k .

A previsão direta foi escolhida nesse caso porque a do tipo recorrente apresentou resultados mais divergentes devido à propagação de erros quando considerou esses horizontes de previsão. A operação do PRSSA+ENFN para avaliação dos resultados é realizada em uma sequência de etapas. Em primeiro lugar, existe a disponibilidade de dados até uma determinada data. Em seguida, aplica-se o método PRSSA+ENFN para obter um intervalo de previsão com nível de confiança em 95%, conforme Equação 2.21. Por fim, os dados reais referentes a esses próximos 30 ou 60 dias são coletados para comparação.

Esse experimento é aplicado considerando 4 períodos com diferentes intervalos de datas, e para cada um, o ciclo completo do PRSSA+ENFN (pré-processamento, treinamento incremental e previsão) é executado 5000 vezes utilizando o critério de seleção aleatória de parâmetros (Seção 5.5.2), levando em conta os intervalos dos parâmetros críticos como segue:

- $W = 20\%N, \dots, 30\%N$;
- $n_l = 2, \dots, 10$, para $l = 1, 2$;
- $0.1 \leq \phi \leq 0.5$;
- $0.01 \leq \eta_0 \leq 0.3$.

Com essa seleção aleatória, obtém-se um intervalo de previsão contendo resultados possíveis que o PRSSA+ENFN calcula como comportamento futuro da dinâmica da doença. Do ponto de vista prático, esse intervalo de previsão é útil para que se possa entender a evolução da doença. A observação prevista em cada dia, isto é, para $T + 1, T + 2, \dots, T + H$, corresponde a uma variável aleatória distinta. A execução de 5000 vezes está relacionada a quantidade de realizações de cada variável aleatória, buscando satisfazer o teorema do limite central e atender aos critérios para definição apropriada dos intervalos de previsão. Sob essas condições, as H variáveis aleatórias tendem a apresentar distribuições de probabilidade aproximadamente normais.

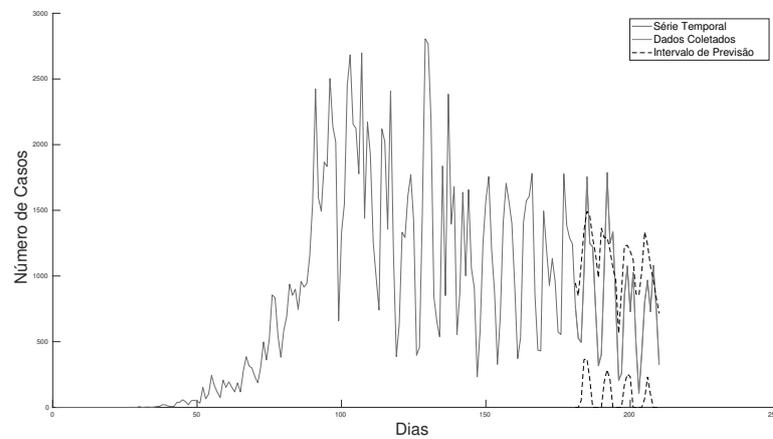
Conforme citado acima, quatro períodos de datas distintos foram estudados para que se verifique o desempenho do PRSSA+ENFN para diferentes dinâmicas da pandemia. Esses períodos são detalhados na Tabela 31 a seguir, que informa as datas de início e fim para todas as etapas do PRSSA+ENFN, assim como a quantidade de dados T (pré-processamento e treinamento), N (quantidade total de dias ou dados) e o horizonte de previsão H considerado.

Tabela 31 – Detalhamento dos Períodos de Datas para o Experimento 4 e os Parâmetros T , N e H para cada caso.

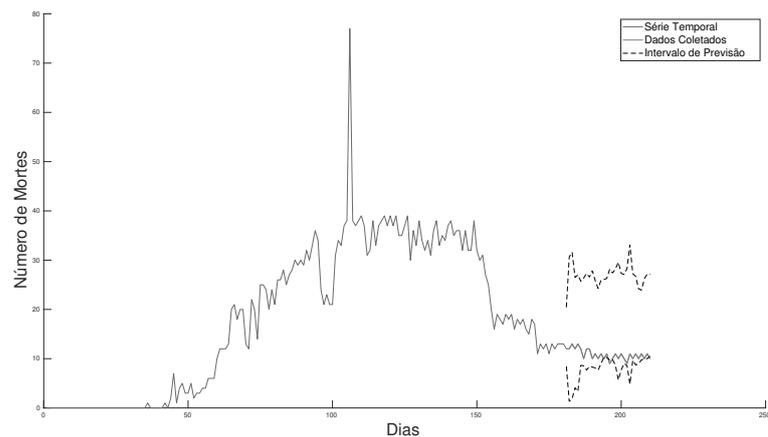
Período	Pré-Processam. e Treinamento		Previsão		T	N	H
	Data Início	Data Fim	Data Início	Data Fim			
Período 1	25/02/2020	22/08/2020	23/08/2020	21/09/2020	180	210	30
Período 2	25/02/2020	01/10/2020	02/10/2020	31/10/2020	220	250	30
Período 3	25/02/2020	02/10/2020	03/10/2020	01/12/2020	221	281	60
Período 4	25/02/2020	20/02/2021	21/02/2021	21/04/2021	362	422	60

Fonte: Autor.

Figura 39 – Intervalo de Previsão do PRSSA+ENFN para os Números Diários de Casos e Mortes causadas por Covid-19 no Estado do Maranhão, Brasil. Período 1: 25/02/2020 a 21/09/2020.



(a) Número de Casos por Dia.



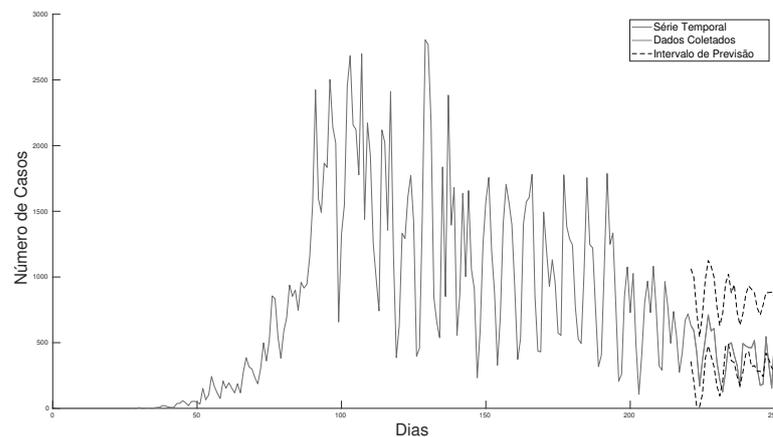
(b) Número de Mortes por Dia.

Fonte: Autor.

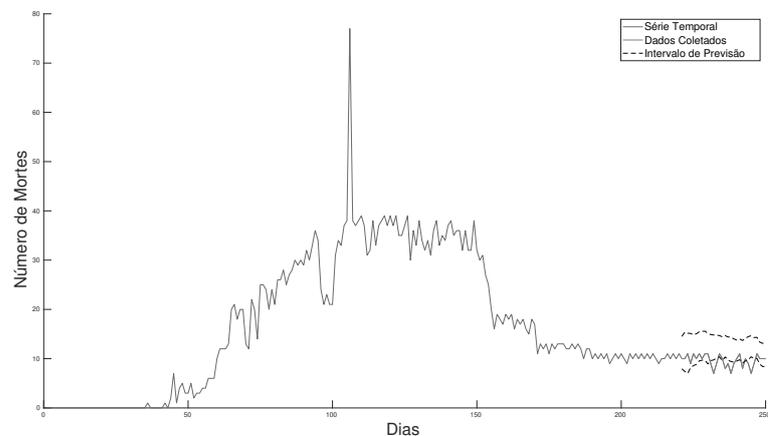
As Figuras 39, 40, 41 e 42 exibem os intervalos de previsão gerados pelo PRSSA+ENFN para as séries de números diários de casos e mortes por Covid-19 no Maranhão. Os dados em azul formam as T observações que foram aplicadas para as etapas de pré-processamento e

treinamento do PRSSA+ENFN. Após isso, calculou-se o intervalo de previsão definido pelas linhas pretas tracejadas, utilizando a seleção aleatória de parâmetros críticos para indicar a região onde as previsões computadas estão em 95% dos resultados para 5000 realizações.

Figura 40 – Intervalo de Previsão do PRSSA+ENFN para os Números Diários de Casos e Mortes causadas por Covid-19 no Estado do Maranhão, Brasil. Período 2: 25/02/2020 a 31/10/2020.



(a) Número de Casos por Dia.



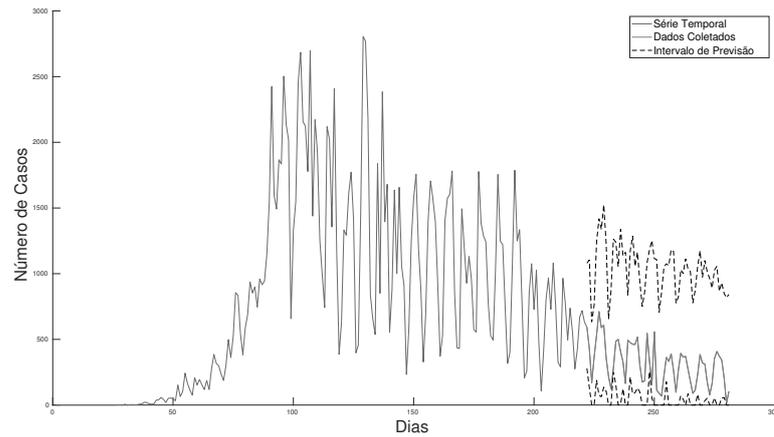
(b) Número de Mortes por Dia.

Fonte: Autor.

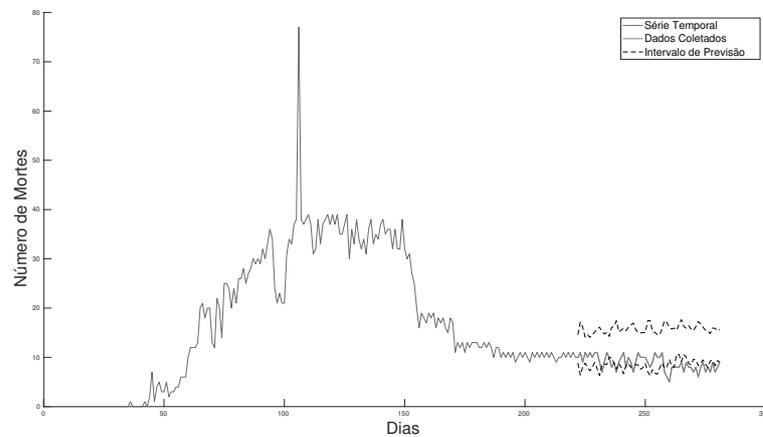
Em relação ao Período 1 na Figura 39, foi obtido um intervalo de previsão para $H = 30$. Nesse momento, o número de casos por dia tinha se tornado estável mas com uma leve queda, por outro lado, o número de mortes diárias teve uma queda mais brusca poucos dias antes, o que ocasionou uma faixa de incerteza maior para essa série temporal. Para o Período 2, ainda está ocorrendo um cenário de leve queda no número de casos e uma estabilidade no número de mortos por Covid-19. O PRSSA+ENFN computou uma faixa adequada que representa esse comportamento da pandemia nesse período. Modificou-se o horizonte de previsão para $H = 60$ a partir do Período 3, que continua de forma semelhante ao momento anterior, mas que mostra que

o PRSSA+ENFN pode fornecer resultados úteis para diferentes valores de H . Por fim, o Período 4 também considera $H = 60$, todavia, em um novo momento em que o número de casos e mortes começou a subir bastante. Os intervalos de previsão computados exibem essa provável evolução da pandemia, que se comprova quando os dados foram coletados (em vermelho).

Figura 41 – Intervalo de Previsão do PRSSA+ENFN para os Números Diários de Casos e Mortes causadas por Covid-19 no Estado do Maranhão, Brasil. Período 3: 25/02/2020 a 01/12/2020.



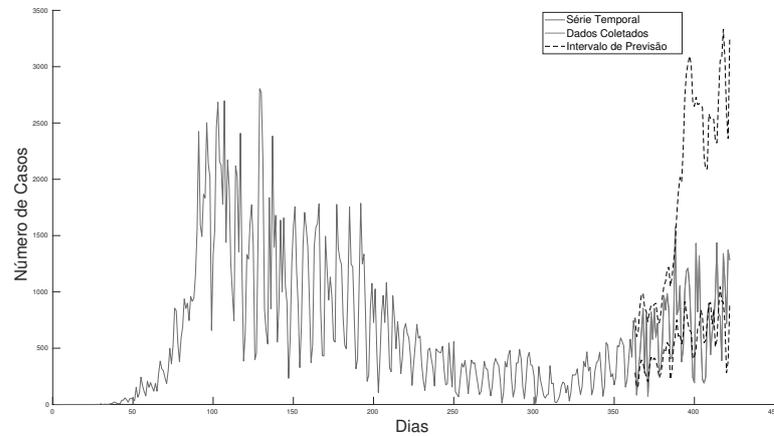
(a) Número de Casos por Dia.



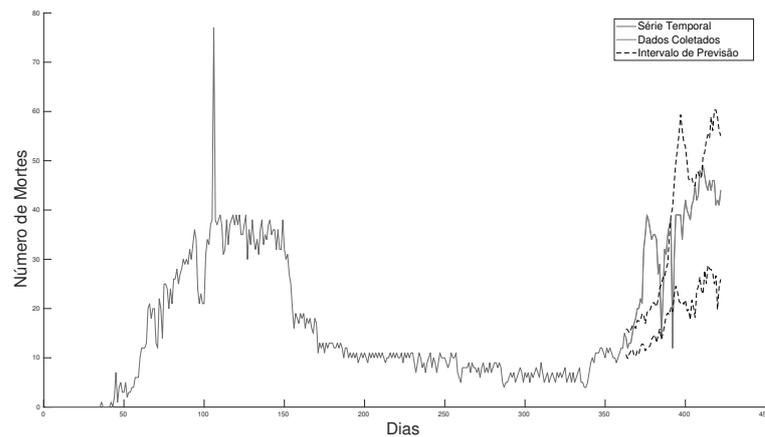
(b) Número de Mortes por Dia.

Fonte: Autor.

Figura 42 – Intervalo de Previsão do PRSSA+ENFN para os Números Diários de Casos e Mortes causadas por Covid-19 no Estado do Maranhão, Brasil. Período 4: 25/02/2020 a 21/04/2021.



(a) Número de Casos por Dia.

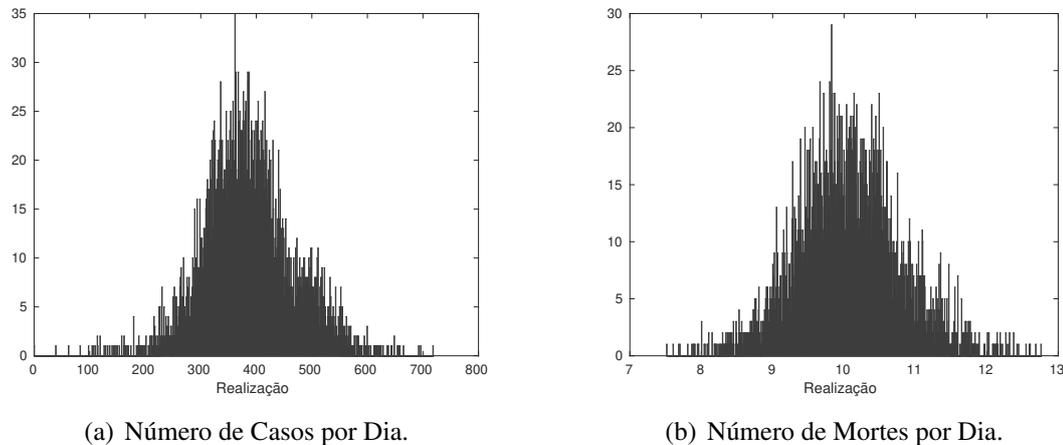


(b) Número de Mortes por Dia.

Fonte: Autor.

Cada dia previsto é uma variável aleatória no instante $T + h$, para $h = 1, \dots, H$. Devido a grande quantidade de realizações, as distribuições de probabilidade de cada uma das variáveis aleatórias para as duas séries em todos os períodos estudados se aproximaram do comportamento normal, conforme o teorema do limite central. Essa característica indica que há uma confiabilidade nesse intervalo de previsão estabelecido pelo PRSSA+ENFN. Como exemplo, a Figura 43 exibe dois histogramas específicos para o instante $k = T + 1$, o primeiro dia previsto para o Período 3 desse experimento, onde percebe-se essa distribuição normal aproximada. Portanto, esses gráficos mostram que o PRSSA+ENFN é uma ferramenta útil que fornece resultados que auxiliam os especialistas no processo de tomada de decisões e planejamento para o caso do estado do Maranhão.

Figura 43 – Histogramas que Representam as Realizações das Variáveis Aleatórias do Números de Casos e Mortes por Covid-19 no Maranhão Considerando o Primeiro Dia de Previsão ($t = T + 1$) para o Período 3.



Fonte: Autor.

6.7 Comentários Finais do Capítulo

Os resultados experimentais apresentados constataram que a abordagem proposta apresenta flexibilidade tanto de trabalhar de forma univariável quanto multivariável, dependendo do problema de previsão em questão. Observou-se algumas das principais características do PRSSA+ENFN, como por exemplo, seus resultados com a previsão direta ou recorrente. Ademais, percebeu-se ainda que a previsão usando componentes pode fornecer resultados apropriados para problemas reais e também os efeitos que os ajustes dos parâmetros críticos ocasionam nas previsões computadas pelo PRSSA+ENFN e em seu processo evolutivo.

Por meio das comparações com outros trabalhos e diversos métodos, avaliou-se que o PRSSA+ENFN apresentou resultados competitivos, sendo uma alternativa de previsão de séries temporais válida para ser aplicada em casos práticos. O Experimento 4 (Seção 6.6) envolvendo a Covid-19 no estado do Maranhão mostrou uma das utilidades desse método, fornecendo uma faixa de valores onde as observações podem se comportar no futuro. Essa abordagem proposta é apropriada para situações em que há *data streams*, ou seja, dados que chegam a todo instante, necessitando atualizar a base de conhecimento da sua estrutura *neuro-fuzzy* para cada observação nova que chega. Entretanto, ressalta-se que em problemas de previsões reais é recomendável aplicar vários métodos de previsão, fazendo com que o especialista tenha um leque de resultados para tomar as melhores decisões.

Parte IV

CONTEXTUALIZAÇÃO FINAL

7 CONSIDERAÇÕES FINAIS

7.1 Conclusões

Nessa tese, uma metodologia evolutiva denominada de PRSSA+ENFN é proposta para previsão de séries temporais univariáveis ou multivariáveis. Esta técnica extrai as componentes não-observáveis das séries temporais recursivamente por meio do método PRSSA desenvolvido nessa pesquisa, aprendendo na etapa de treinamento incremental a partir da dinâmica dos dados das componentes, preparando-se para realizar a previsão. Por sua vez, a ENFN é capaz de evoluir sua estrutura, aprendendo e modificando-se a partir do comportamento dos dados dessas componentes, sendo que a quantidade de regras *fuzzy* pode aumentar ou diminuir. Utilizando as regras *fuzzy*, o método proposto apresenta transparência em sua estrutura.

O PRSSA+ENFN considerou alguns aspectos importantes do estado da arte na área de previsão, como sua característica híbrida, aproveitando as vantagens de cada método em seus procedimentos; e o uso de componentes não-observáveis na previsão, decompondo uma série temporal em padrões mais simples. Verificou-se através dos experimentos que os resultados de previsão da rede neuro-*fuzzy* sem efetuar a extração das componentes foram inferiores para as séries temporais não-estacionárias consideradas, o que ressalta a vantagem do “dividir para conquistar” nesses casos. Além disso, a aprendizagem incremental a partir dos dados das componentes que chegam a cada instante é um atributo importante dessa abordagem.

Com essas peculiaridades, os resultados experimentais e as comparações com outras técnicas e trabalhos mostram que PRSSA+ENFN é uma alternativa competitiva para realizar a previsão de séries temporais univariáveis ou multivariáveis, utilizando diferentes horizontes de previsão, podendo ser uma previsão do tipo recorrente ou direta de acordo com o problema. Ademais, aplicou-se o PRSSA+ENFN em um problema real relacionado à pandemia do Covid-19, realizando a previsão em momentos distintos dos números de casos e das mortes causadas por esse vírus no estado do Maranhão, Brasil. Percebe-se que a proposta mostrou um indicativo por meio do intervalo de previsão sobre o comportamento futuro da pandemia com sucesso, mesmo quando a dinâmica da pandemia foi modificada.

Além disso, o PRSSA+ENFN apresenta um nível significativo de autonomia na sua execução, contendo apenas quatro parâmetros críticos que precisam ser determinados pelos usuários. O algoritmo genético é uma alternativa para determinação de valores para esses parâmetros críticos com reduzida influência do usuário, porém, o custo computacional aumenta ao longo das gerações. Outra forma de obter resultados do PRSSA+ENFN é selecionar aleatoriamente esses parâmetros críticos, tendo como resultado um intervalo que indica as possibilidades de previsão que o método proposto pode fornecer.

Conclui-se que a proposta dessa tese é ideal em um cenário onde os dados chegam a

cada instante, pois o PRSSA+ENFN é capaz de realizar uma aprendizagem incremental sem precisar recorrer aos dados históricos completos da série temporal. Portanto, é uma alternativa para ser utilizada em conjunto com outros métodos de previsão para problemas envolvendo séries temporais univariáveis e multivariáveis.

7.2 Trabalhos Futuros

Sugestões de trabalhos futuros são listadas a seguir:

- Aplicar mais estudos de caso em problemas reais na medicina, indústria, identificação de sistemas, entre outras áreas em que a previsão é importante;
- Inserção do conhecimento do especialista aplicado na previsão do tipo qualitativa no formato de regras *fuzzy* para refinar os resultados da alternativa proposta;
- Avaliação do método MRSSA, apresentado no Apêndice B, em combinação com a ENFN. O objetivo seria verificar se os resultados de previsão aplicando a decomposição de acordo com a ideia desenvolvida nesse apêndice seriam vantajosos ou não;
- Utilizar outras abordagens para decomposição, como o método *Recursive Principal Components Analysis* (RPCA) para extração dos padrões dos dados e aplicar de forma híbrida com a ENFN;
- Explorar a possibilidade da aprendizagem incremental ser por bloco de dados, reduzindo a quantidade de operações necessárias para o processo evolutivo;
- Pesquisar sobre a viabilidade dessa abordagem proposta para previsão de séries temporais no contexto de *big data*.
- Atualização do parâmetro s_j ao longo do treinamento incremental e conforme novos dados tornam-se disponíveis.

REFERÊNCIAS

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.

ADEWOLE, A. et al. Chaotic time series analysis of meteorological parameters in some selected stations in nigeria. *Scientific African*, v. 10, 2020.

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.

ALHNAITY, B.; ABBOD, M. A new hybrid financial time series prediction model. *Engineering Applications of Artificial Intelligence*, v. 95, 2020.

ALLEN, M.; SMITH, L. Monte Carlo SSA: Detecting Irregular Oscillations in the presence of coloured noise. *Journal of Climate*, v. 9, p. 3373–3404, 1996.

ANDREWS, D.; HERZBERG, A. Monthly mean sunspot numbers. In: _____. *Data: A Collection of Problems from Many Fields for the Student and Research Worker*. New York, NY: Springer New York, 1985. p. 67–74. ISBN 978-1-4612-5098-2.

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

ANGELOV, P. Evolving fuzzy systems. In: _____. *Encyclopedia of Complexity and Systems Science*. New York, NY: Springer New York, 2009. p. 3242–3255. ISBN 978-0-387-30440-3.

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. 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.; GU, X. *Empirical Approach to Machine Learning*. Switzerland: Studies in Computational Intelligence. Springer, 2019.

ANGELOV, P.; GU, X.; KANGIN, D. Empirical data analytics. *Int. J. Intell. Syst.*, p. 1261–1284, 2017.

ANGELOV, P.; GU, X.; PRINCIPE, J. Autonomous Learning Multi-Model Systems from Data Streams. *IEEE Transactions on Fuzzy Systems*, p. 1–12, 2017.

- ANGELOV, P.; KASABOV, N. Evolving Computational Intelligence Systems. International Workshop on Genetic Fuzzy Systems, Granada, Spain, p. 76–82, 2005.
- ANGELOV, P.; YAGER, R. A new type of simplified fuzzy rule-based system. *International Journal of General Systems*, v. 41, p. 163–185, 2011.
- 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.
- ARMSTRONG, J. *Principles of Forecasting: a Handbook for Researchs and Practitioners*. Boston: Kluwer Academic Publishers, 2001.
- ASHRAFI, M.; PRASAD, D.; QUEK, C. It2-gsetsk: An evolving interval type-ii tsf fuzzy neural system for online modeling of noisy data. *Neurocomputing*, v. 407, p. 1–11, 2020.
- ASLANARGUN, A. et al. Comparison of arima, neural networks and hybrid models in time series: tourist arrival forecasting. *Journal of Statistical Computation and Simulation*, Taylor & Francis, v. 77, n. 1, p. 29–53, 2007.
- BABU, C.; REDDY, B. A moving-average filter based hybrid arima–ann model for forecasting time series data. *Applied Soft Computing*, v. 23, p. 27–38, 2014. ISSN 1568-4946.
- BABUSKA, R. *Fuzzy Modeling for Control*. New York: Springer Science Business Media, 1998.
- BAO, R. et al. Correntropy-based evolving fuzzy neural system. *IEEE Transactions on Fuzzy Systems*, v. 26, n. 3, p. 1324–1338, 2018.
- BAPTISTA, M. The Portuguese pharmaceutical market in the near future – a time series exploration approach. *Dissertacao de Mestrado*, Universidade Nova de Lisboa, Portugal, 2008.
- BARUAH, R.; ANGELOV, P. Dec: Dynamically evolving clustering and its application to structure identification of evolving fuzzy models. *IEEE Transactions on Cybernetics*, v. 44, n. 9, p. 1619–1631, 2014.
- BEZERRA, C. et al. An evolving approach to data streams clustering based on typicality and eccentricity data analytics. *Information Sciences*, v. 518, p. 13–28, 2020.
- BÓGALO, J.; PONCELA, P.; SENRA, E. Circulant singular spectrum analysis: A new automated procedure for signal extraction. *Signal Processing*, v. 179, p. 107824, 2021.
- BONTEMPI, G.; TAIEB, S.; BORGNE, Y. L. Machine learning strategies for time series forecasting. In: _____. [S.l.: s.n.], 2013. v. 138.
- BROCKWELL, P. J.; DAVIS, R. A. *Introduction to Time Series and Forecasting*. [S.l.]: Springer, 2002. ISBN 9780387953519.
- BUYUKSAHIN, U.; ERTEKIN, S. Improving forecasting accuracy of time series data using a new arima-ann hybrid method and empirical mode decomposition. *Neurocomputing*, v. 361, p. 151–163, 2019.
- CAMARGOS, M. et al. Data-driven prognostics of rolling element bearings using a novel error based evolving takagi–sugeno fuzzy model. *Applied Soft Computing*, v. 96, p. 106628, 2020.
- CAO, J.; LI, Z.; LI, J. Financial time series forecasting model based on ceemdan and lstm. *Physica A: Statistical Mechanics and its Applications*, v. 519, p. 127–139, 2019.

- CASSIANO, K. *Análise de Séries Temporais Usando Análise Espectral Singular (SSA) e Clusterização de Suas Componentes Baseada em Densidade*. Rio de Janeiro, Brasil: Tese de Doutorado. Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, 2014.
- CASTILLO, E.; ALVAREZ, E. Expert systems: Uncertainty and learning. *Southampton Boston*, 1991.
- CHAMBERS, J.; MULLICK, S.; SMITH, D. How to Choose the Right Forecasting Technique. *Harvard Business Review*, v. 49, p. 45–57, 1971.
- CHAYAMA, M.; HIRATA, Y. When univariate model-free time series prediction is better than multivariate. *Physics Letters A*, 2016.
- CHEN, C.-T. *Linear System Theory and Design*. 3. ed. New York: Oxford University Press, 1999.
- CHEN, S.; WANG, J.; ZHANG, H. A hybrid pso-svm model based on clustering algorithm for short-term atmospheric pollutant concentration forecasting. *Technological Forecasting and Social Change*, v. 146, p. 41–54, 2019.
- CHENG, C.; YANG, J. Fuzzy time-series model based on rough set rule induction for forecasting stock price. *Neurocomputing*, 2018.
- CORMEN, T. et al. *Introduction to Algorithms. Third Edition*. Cambridge, Massachusetts: MIT Press, 2009.
- CORRÊA, M.; AGUIRRE, L. Identificação não-linear caixa-cinza: uma revisão e novos resultados. *Sba: Controle & Automação Sociedade Brasileira de Automatica*, scielo, v. 15, p. 109–126, 06 2004. ISSN 0103-1759.
- COVID-19 - Brasil I.O. 2021. <<https://brasil.io/dataset/covid19/>>. Acessado em: 21/04/2021.
- DALKEY, C. *Delphi Method*. [S.l.]: RAND Corporation, Santa Monica, CA, 1967.
- DAS, M.; GHOSH, S. Data-driven approaches for meteorological time series prediction: A comparative study of the state-of-the-art computational intelligence techniques. *Pattern Recognition Letters*, v. 180, p. 1–10, 2017.
- 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.
- DU, S. et al. Multivariate time series forecasting via attention-based encoder-decoder framework. *Neurocomputing*, v. 388, p. 269–279, 2020.
- DUDA, R.; HART, P.; STORK, D. *Pattern Classification*. 2. ed. Chichester, West Sussex, England: Wiley-Interscience, 2000.
- ELSNER, J.; TSONIS, A. *Singular Spectrum Analysis: A New Tool in Time Series Analysis*. [S.l.]: Springer US, 1996. ISBN 9781441932662.
- EVANGELISTA, A.; SERRA, G. State space black-box modelling via markov parameters based on evolving type-2 neural-fuzzy inference system for nonlinear multivariable dynamic systems. *Fuzzy Sets and Systems*, v. 394, p. 1–39, 2020. Neurofuzzy Systems and Learning.

- FERDAUS, M. et al. Generic evolving self-organizing neuro-fuzzy control of bio-inspired unmanned aerial vehicles. *IEEE Transactions on Fuzzy Systems*, v. 28, n. 8, p. 1542–1556, 2020.
- FERDAUS, M. et al. Pac: A novel self-adaptive neuro-fuzzy controller for micro aerial vehicles. *Information Sciences*, v. 512, p. 481–505, 2020.
- FILHO, O.; SERRA, G. Recursive fuzzy instrumental variable based evolving neuro-fuzzy identification for non-stationary dynamic system in a noisy environment. *Fuzzy Sets and Systems*, v. 338, p. 50–89, 2018.
- GALICIA, A. et al. Multi-step forecasting for big data time series based on ensemble learning. *Knowledge-Based Systems*, v. 163, p. 830–841, 2019.
- GALICIA, A. et al. A novel spark-based multi-step forecasting algorithm for big data time series. *Information Sciences*, v. 467, p. 800–818, 2018.
- 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.
- GE, D.; ZENG, X. Learning evolving ts fuzzy systems with both local and global accuracy - a local online optimization approach. *Applied Soft Computing*, v. 68, p. 795–810, 2018. ISSN 1568-4946.
- GE, D.; ZENGE, X. Learning data streams online - an evolving fuzzy system approach with self-learning/adaptative thresholds. *Information Sciences*, Elsevier, v. 507, p. 172–184, 2020.
- GODINHO, G.; Maçaira, P.; SOUZA, R. Combined forecast model for wind generation in brazilian monthly dispatch scheduling. *IEEE Latin America Transactions*, v. 17, n. 09, p. 1432–1438, 2019.
- 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.
- GOODWIN, P. Integrating management judgment and statistical methods to improve short-term forecasts. *Omega*, v. 30, p. 127–135, 2002.
- GOOIJER, J.; HYNDMAN, R. 25 years of time series forecasting. *International Journal of Forecasting*, v. 22, p. 443–473, 2006.
- GRAF, R.; ZHU, S.; SIVAKUMAR, B. Forecasting river water temperature time series using a wavelet-neural network hybrid modelling approach. *Journal of Hydrology*, v. 578, p. 124115, 2019.
- GROTH, A. et al. Interannual variability in the north atlantic ocean's temperature field and its association with the wind stress forcing. *Journal of Climate*, v. 30, p. 2655–2678, 2017.
- GU, X.; ANGELOV, P. Self-boosting first-order autonomous learning neuro-fuzzy systems. *Applied Soft Computing*, v. 77, p. 118–134, 2019.
- GU, X.; ANGELOV, P.; RONG, H. Local optimality of self-organising neuro-fuzzy inference systems. *Information Sciences*, v. 503, p. 351–380, 2019.

- GU, X.; SHEN, Q.; ANGELOV, P. Particle swarm optimized autonomous learning fuzzy system. *IEEE Transactions on Cybernetic*, p. 1–12, 2020.
- GUYET, T.; NICOLAS, H. Long term analysis of time series of satellite images. *Pattern Recognition Letters*, 2015.
- HAJIRAHIMI, Z.; KHASHEI, M. Hybrid structures in time series modeling and forecasting: A review. *Engineering Applications of Artificial Intelligence*, v. 86, p. 83–106, 2019.
- HAMILTON, J. D. *Time Series Analysis*. New Jersey: Pinceton University Press, 1994.
- HASSANI, H. et al. Monthly forecasting of gdp with mixed-frequency multivariate singular spectrum analysis. *International Journal of Forecasting*, v. 35, n. 4, p. 1263–1272, 2019.
- HASSANI, H.; THOMAKOS, D. A Review on Singular Spectrum Analysis for Economic and Financial Time Series. *Statistics and Its Interface*, v. 3, p. 377–397, 2010.
- HITAM, N.; ISMAIL, A.; SAEED, F. An optimized support vector machine (svm) based on particle swarm optimization (psa) for cryptocurrency forecasting. *Procedia Computer Science*, v. 163, p. 427–433, 2019.
- HOFFMAN, J. Chapter 7 - statistical inference: Confidence limits and the central limit theorem. In: HOFFMAN, J. I. (Ed.). *Basic Biostatistics for Medical and Biomedical Practitioners (Second Edition)*. Second edition. [S.l.]: Academic Press, 2019. p. 127–141. ISBN 978-0-12-817084-7.
- HYNDMAN, R. *Time Series Data Library*. 2010. <<http://data.is/TSDLdemo>>. Acessado em: 19/05/2021.
- HYNDMAN, R.; KHANDAKAR, Y. Automatic time series forecasting: The forecast package for r. *Journal of Statistical Software, Articles*, v. 27, n. 3, p. 1–22, 2008. ISSN 1548-7660.
- HYNDMAN, R. J.; ATHANASOPOULOS, G. *Forecasting: principles and practice*. [S.l.]: OTexts, 2014. ISBN 9780987507105.
- JANG, J. Anfis: Adaptive network-based fuzzy inference systems. *IEEE Trans. on SMC*, v. 23, p. 665–685, 1993.
- JIANG, H.; HENG, J. A hybrid forecasting system based on fuzzy time series and multi-objective optimization for wind speed forecasting. *Applied Energy*, v. 235, p. 786–801, 2019.
- JIANG, W. et al. Holt–winters smoothing enhanced by fruit fly optimization algorithm to forecast monthly electricity consumption. *Energy*, v. 193, 2020.
- JOO, T.; KIM, S. Time series forecasting based on wavelet filtering. *Expert Systems With Applications*, v. 42, p. 3868–3874, 2015.
- JÚNIOR, D.; DE OLIVEIRA, J.; NETO, P. An intelligent hybridization of arima with machine learning models for time series forecasting. *Knowledge-Based Systems*, v. 175, p. 72–86, 2019.
- KALANTARI, M. Forecasting covid-19 pandemic using optimal singular spectrum analysis. *Chaos, Solitons & Fractals*, v. 142, p. 110547, 2021. ISSN 0960-0779.
- KANANI, P.; PADOLE, M. Ecg heartbeat arrhythmia classification using time-series augmented signals and deep learning approach. *Procedia Computer Science*, v. 171, p. 524–531, 2020.

- 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.
- 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.
- KHANDELWAL, I.; ADHIKARI, R.; VERMA, G. Time series forecasting using hybrid arima and ann models based on dwt decomposition. *Procedia Computer Science. International Conference on Computer, Communication and Convergence*, v. 48, p. 173–179, 2015.
- KHIALI, L. et al. Detection of spatio-temporal evolutions on multi-annual satellite image time series: A clustering based approach. *Int J Appl Earth Obs Geoinformation*, p. 103–109, 2019.
- KISI, O.; HEDDAM, S.; YASEEN, Z. The implementation of univariable scheme-based air temperature for solar radiation prediction: New development of dynamic evolving neural-fuzzy inference system model. *Applied Energy*, v. 241, p. 184–195, 2019.
- KOLIDAKIS, S. et al. Road traffic forecasting — a hybrid approach combining artificial neural network with singular spectrum analysis. *Economic Analysis and Policy*, v. 64, p. 159–171, 2019.
- KOROBAYNIKOV, A. Computation- and space-efficient implementation of ssa. *Statistics and Its Interface*, v. 3, p. 357–368, 2010.
- KOTU, V.; DESHPANDE, B. Time Series Forecasting. In: KOTU, V.; DESHPANDE, B. (Ed.). *Data Science: Concepts and Practice*. [S.l.]: Elsevier. ISBN: 978-0-12-814761-0, 2019. cap. 12, p. 395–445.
- KRISHNANNAIR, S.; ALDRICH, C. Process monitoring and fault detection using empirical mode decomposition and singular spectrum analysis. *IFAC-PapersOnLine*, v. 52, n. 14, p. 219–224, 2019. ISSN 2405-8963. 18th IFAC Symposium on Control, Optimization and Automation in Mining, Mineral and Metal Processing, MMM 2019.
- LAOHAKIAT, S.; SA-ING, V. An incremental density-based clustering framework using fuzzy local clustering. *Information Sciences*, v. 547, p. 404–426, 2021.
- LELES, M. et al. Improving reconstruction of time-series based in singular spectrum analysis: A segmentation approach. *Digital Signal Processing*, v. 77, p. 63–76, 2018.
- LI, J. et al. Dtdr–alstm: Extracting dynamic time-delays to reconstruct multivariate data for improving attention-based lstm industrial time series prediction models. *Knowledge-Based Systems*, v. 211, 2021.
- LIANG, X. et al. Assessing beijing’s pm2.5 pollution: severity, weather impact, apec and winter heating. *Proceedings of the Royal Society A*, v. 471, p. 1364–5021, 2015.

- LIN, Y. et al. Effectiveness analysis of bio-electronic stimulation therapy to parkinson's diseases via joint singular spectrum analysis and discrete fourier transform approach. *Biomedical Signal Processing and Control*, v. 62, p. 102131, 2020. ISSN 1746-8094.
- LIU, C. et al. A hybrid prediction model for residential electricity consumption using holt-winters and extreme learning machine. *Applied Energy*, v. 275, 2020.
- LIU, H. et al. Forecast of the trend in incidence of acute hemorrhagic conjunctivitis in china from 2011–2019 using the seasonal autoregressive integrated moving average (sarima) and exponential smoothing (ets) models. *Journal of Infection and Public Health*, v. 13, n. 2, p. 287–294, 2020.
- LIU, H. et al. Smart wind speed deep learning based multi-step forecasting model using singular spectrum analysis, convolutional gated recurrent unit network and support vector regression. *Renewable Energy*, v. 143, p. 842–854, 2019.
- LIU, M. et al. Short-term wind speed forecasting based on the jaya-svm model. *International Journal of Electrical Power Energy Systems*, v. 121, p. 106056, 2020.
- LU, Y. et al. Estimation of the foetal heart rate baseline based on singular spectrum analysis and empirical mode decomposition. *Future Generation Computer Systems*, v. 112, p. 126–135, 2020.
- LUGHOFER, E. *Evolving Fuzzy Systems: Methodologies, Advanced Concepts and Applications*. [S.l.]: Springer, 2011.
- LUGHOFER, E. Improving the robustness of recursive consequent parameters learning in evolving neuro-fuzzy systems. *Information Sciences*, v. 545, p. 555–574, 2021.
- LUSIA, D.; AMBARWATI, A. Multivariate Forecasting Using Hybrid VARIMA Neural Network in JCI Case. In: . Yogyakarta, Indonesia: 2018 International Symposium on Advanced Intelligent Informatics (SAIN), 2018.
- LY, K. T. A covid-19 forecasting system using adaptive neuro-fuzzy inference. *Finance Research Letters*, p. 101844, 2020.
- MACKEY, M.; GLASS, L. Oscillation and chaos in physiological control systems. *Science*, v. 197, p. 287–289, 1977.
- MAKRIDAKIS, S.; SPILLOTIS, E.; ASSIMAKOPOULOS, V. Statistical and machine learning forecasting methods: Concerns and ways forward. *PLoS ONE*, p. 1–26, 2018.
- MAKRIDAKIS, S.; WHEELWRIGHT, S.; HYNDMAN, R. *Forecasting: Methods and Applications*. 3. ed. New York: John Wiley & Sons, 1998.
- MAO, X.; SHANG, P. Multivariate singular spectrum analysis for traffic time series. *Physica A: Statistical Mechanics and its Applications*, v. 526, p. 121063, 2019.
- MERH, N.; SAXENA, V.; PARDASANI, K. A comparison between hybrid approaches of ann and arima for indian stock trend forecasting. *Business Intelligence Journal*, v. 3, p. 23, 2010.
- MI, X.; ZHAO, S. Wind speed prediction based on singular spectrum analysis and neural network structural learning. *Energy Conversion and Management*, v. 216, p. 112956, 2020. ISSN 0196-8904.

- MONTGONOMERY, D. C.; JENNINGS, C. L.; KULAHCI, M. *Introduction to Time Series Analysis and Forecasting*. 2. ed. [S.l.]: Wiley, 2015.
- MORETTIN, P.; TOLOI, C. *Análise de Séries Temporais*. São Paulo: Editora Blucher, 2006.
- MUKHOPADHYAY, S.; KRISHNAN, S. A singular spectrum analysis-based model-free electrocardiogram denoising technique. *Computer Methods and Programs in Biomedicine*, v. 188, p. 105304, 2020. ISSN 0169-2607.
- NGUYEN, L.; NOVÁK, V. Forecasting seasonal time series based on fuzzy techniques. *Fuzzy Sets and Systems*, v. 361, p. 114–129, 2019.
- NIU, T. et al. Developing a deep learning framework with two-stage feature selection for multivariate financial time series forecasting. *Expert Systems with Applications*, v. 148, 2020.
- OLIVEIRA, J. et al. A hybrid semi-linear system for time series forecasting. In: *2017 Brazilian Conference on Intelligent Systems (BRACIS)*. [S.l.: s.n.], 2017. p. 79–84.
- OLIVEIRA, J.; LUDERMIR, T. A distributed pso-arima-svr hybrid system for time series forecasting. *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, p. 3867–3872, 2014.
- OLIVEIRA, J.; LUDERMIR, T. A hybrid evolutionary decomposition system for time series forecasting. *Neurocomputing*, v. 180, p. 27–34, 2016.
- OLIVEIRA, J. et al. A hybrid optimized error correction system for time series forecasting. *Applied Soft Computing*, v. 87, p. 105970, 2020. ISSN 1568-4946.
- OZOZEN, A. et al. A combined seasonal arima and ann model for improved results in electricity spot price forecasting: Case study in turkey. In: *2016 Portland International Conference on Management of Engineering and Technology (PICMET)*. [S.l.: s.n.], 2016. p. 2681–2690.
- 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.
- PANIGRAHI, S.; BEHERA, H. A hybrid ets–ann model for time series forecasting. *Engineering Applications of Artificial Intelligence*, v. 66, p. 49–59, 2017. ISSN 0952-1976.
- PEDDANENI, H. et al. Recursive principal components analysis using eigenvector matrix perturbation. *IEEE Workshop on Machine Learning for Signal Processing*, p. 83–92, 2004.
- PELAGATTI, M. *Time Series Modelling with Unobserved Components*. [S.l.: s.n.], 2015. ISBN 9781482225006.
- PETROPOULOS, F.; MAKRIDAKIS, S.; STYLIANOU, N. Covid-19: Forecasting confirmed cases and deaths with a simple time series model. *International Journal of Forecasting*, 2020.
- PLATT, J. A Resource-Allocating Network for Function Interpolation. *Neural Computation*, v. 3, n. 2, p. 213–225, 06 1991.
- PRATAMA, M. et al. pclass: An effective classifier for streaming examples. *IEEE Transactions on Fuzzy Systems*, v. 23, n. 2, p. 369–386, 2015.

- RAMACHANDRAN, K.; TSOKOS, C. Chapter 14 - some issues in statistical applications: an overview. In: RAMACHANDRAN, K. M.; TSOKOS, C. P. (Ed.). *Mathematical Statistics with Applications in R (Third Edition)*. Third edition. [S.l.]: Academic Press, 2021. p. 569–613. ISBN 978-0-12-817815-7.
- READ, J. et al. Batch-Incremental versus Instance-Incremental Learning in Dynamic and Evolving Data. In: HOLLMÉN, J.; KLAWONN, F.; TUCKER, A. (Ed.). *Advances in Intelligent Data Analysis XI. IDA 2012. Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer, 2012. v. 7619.
- RODRIGUES JÚNIOR, S. *Metodologia Evolutiva para Previsão Inteligente de Séries Temporais Sazonais Baseada em Espaço de Estados Não-Observáveis*. Orientador: Ginalber Luiz de Oliveira Serra. 134 p. Dissertação (Mestrado) — Programa de Pós-Graduação em Engenharia de Eletricidade, Universidade Federal do Maranhão, São Luís, 2017.
- RODRIGUES JÚNIOR, S. R.; SERRA, G. A novel intelligent approach for state space evolving forecasting of seasonal time series. *Engineering Applications of Artificial Intelligence*, v. 64, 2017.
- RODRIGUES, P.; LOURENÇO, V.; MAHMOUDVAND, R. A robust approach to singular spectrum analysis. *Qual Reliab Engng Int*, v. 0, p. 1–11, 2018.
- RODRIGUES, P.; TUY, P.; MAHMOUDVAND, R. Randomized singular spectrum analysis for long time series. *Journal of Statistical Computaion and Simulation*, 2018.
- RUBIO, J. Sofmls: Online self-organizing fuzzy modified least-squares network. *IEEE Transactions on Fuzzy Systems*, v. 17, n. 6, p. 1296–1309, 2009.
- SAGHEER, A.; KOTB, M. Time series forecasting of petroleum production using deep LSTM recurrent networks. *Neurocomputing*, v. 323, p. 203–213, 2019.
- SAMANTA, S. et al. A new neuro-fuzzy inference system with dynamic neurons (nfis-dn) for system identification and time series forecasting. *Applied Soft Computing*, v. 82, p. 105567, 2019.
- Mean squared error. In: SAMMUT, C.; WEBB, G. (Ed.). *Encyclopedia of Machine Learning*. Boston, MA: Springer US, 2010. p. 653–653. ISBN 978-0-387-30164-8.
- SA'AD, H.; ISA, N.; AHMED, M. A structural evolving approach for fuzzy systems. *IEEE Transactions on Fuzzy Systems*, v. 28, n. 2, p. 273–287, 2020.
- SCHICHL, H. *Models and the History of Modeling*. [S.l.: s.n.], 2004. v. 88. 25-36 p. ISBN 9781461302155.
- SERRA, G.; BOTTURA, C. An iv-qr algorithm for neuro-fuzzy multivariable online identification. *IEEE Transactions on Fuzzy Systems*, v. 15, n. 2, p. 200–210, 2007.
- SETNES, M. et al. Similarity measures in fuzzy rule base simplification. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 28, n. 3, p. 376–386, 1998.
- SEVERIANO, C. et al. Evolving fuzzy time series for spatio-temporal forecasting in renewable energy systems. *Renewable Energy*, v. 171, p. 764–783, 2021.

- SHEN, Z. et al. A novel time series forecasting model with deep learning. *Neurocomputing*, v. 396, p. 302–313, 2020.
- SHI, H.; WORDEN, K.; CROSS, E. A cointegration approach for heteroscedastic data based on a time series decomposition: An application to structural health monitoring. *Mechanical Systems and Signal Processing*, v. 120, p. 16–31, 2019.
- SIEGEL, A. Chapter 9 - confidence intervals: Admitting that estimates are not exact. In: SIEGEL, A. F. (Ed.). *Practical Business Statistics*. Boston: Academic Press, 2012. p. 219–247. ISBN 978-0-12-385208-3.
- SINGH, R. et al. Short-term statistical forecasts of covid-19 infections in india. *IEEE Access*, v. 8, p. 186932–186938, 2020.
- SKRJANC, I. et al. Evolving cloud-based system for the recognition of drivers' actions. *Expert Systems with Applications*, v. 99, p. 231–238, 2018.
- SKRJANC, I. et al. Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A Survey. *Information Sciences*, v. 490, p. 344–368, 2019.
- SOARES, E.; ANGELOV, P.; GU, X. Autonomous learning multiple-model zero-order classifier for heart sound classification. *Applied Soft Computing*, v. 94, p. 106449, 2020.
- SOUZA, P.; PONCE, H.; LUGHOFER, E. Evolving fuzzy neural hydrocarbon networks: A model based on organic compounds. *Knowledge-Based Systems*, v. 203, p. 106099, 2020.
- SULANDARI, W. et al. Indonesian electricity load forecasting using singular spectrum analysis, fuzzy systems and neural networks. *Energy*, v. 190, p. 116408, 2020.
- 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.
- TAIEB, S.; BONTEMPI, G. Recursive multi-step time series forecasting by perturbing data. In: . [S.l.: s.n.], 2011. p. 695–704.
- 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.
- TSAY, R. *Multivariate time series analysis: with R and financial applications*. New Jersey: John Wiley & Sons, 2014.
- VITO, S. et al. On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. *Sensors and Actuators B: Chemical*, v. 129, p. 750–757, 2008.
- WANG, L. *A Course in Fuzzy Systems and Control*. [S.l.]: Prentice-Hall, 1997.
- WANG, L. et al. An arima-ann hybrid model for time series forecasting. *Systems Research and Behavioral Science*, v. 30, n. 3, p. 244–259, 2013.
- WEI, W. *Time Series Analysis: Univariate and Multivariate Methods*. 2. ed. [S.l.]: Pearson Education, 2006.

- WILCOX, R. Chapter 4 - confidence intervals in the one-sample case. In: WILCOX, R. (Ed.). *Introduction to Robust Estimation and Hypothesis Testing (Fourth Edition)*. Fourth edition. [S.l.]: Academic Press, 2017, (Statistical Modeling and Decision Science). p. 107–144. ISBN 978-0-12-804733-0.
- XU, S.; CHAN, H.; ZHANG, T. Forecasting the demand of the aviation industry using hybrid time series sarima-svr approach. *Transportation Research Part E: Logistics and Transportation Review*, v. 122, p. 169–180, 2019.
- YAN, X. et al. Multichannel fault diagnosis of wind turbine driving system using multivariate singular spectrum decomposition and improved kolmogorov complexity. *Renewable Energy*, v. 170, p. 724–748, 2021.
- YANG, H.; CHEN, Y. Hybrid deep learning and empirical mode decomposition model for time series applications. *Expert Systems with Applications*, v. 120, p. 128–138, 2019.
- YANG, H. et al. Online learning for vector autoregressive moving-average time series prediction. *Neurocomputing*, v. 315, p. 19–17, 2018.
- YATES, R.; GOODMAN, D. *Probabilidade e Processos Estocásticos: Uma Introdução Amigável para Engenheiros Eletricistas e da Computação*. 3 ed. Rio de Janeiro: LTC, 2017.
- 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.
- YOLCU, U.; EGRIOGLU, E.; ALADAG, C. A new linear & nonlinear artificial neural network model for time series forecasting. *Decision Support Systems*, v. 54, n. 3, p. 1340–1347, 2013. ISSN 0167-9236.
- ZADEH, L. Fuzzy Sets. *Information and Control*, v. 8, p. 338–353, 1965.
- ZEINALNEZHAD, M. et al. Air pollution prediction using semi-experimental regression model and adaptive neuro-fuzzy inference system. *Journal of Cleaner Production*, v. 261, p. 121218, 2020.
- ZHANG, G. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, v. 50, p. 159–175, 2003.
- ZUBAIDI, S. et al. A novel approach for predicting monthly water demand by combining singular spectrum analysis with neural networks. *Journal of Hydrology*, v. 561, p. 136–145, 2018.

Apêndices

APÊNDICE A – ALGORITMOS UTILIZADOS NA TESE

A.1 Algoritmo da Análise Espectral Singular (SSA)

O algoritmo a seguir mostra o procedimento do método SSA tradicional para extração das componentes não-observáveis de uma série temporal.

Algoritmo 1 Método SSA Básico

- 1: Importar os dados da série temporal $y_k, k = 1, \dots, N$;
 - 2: Definir os parâmetros L e n ;
 - 3: Calcular o parâmetro $\mathcal{K} = N - L + 1$;
 - Incorporação:**
 - 4: Obter a Matriz Trajetória \mathbf{M} (Eq. 3.2);
 - 5: Matriz Covariância $\mathbf{S} = \mathbf{M}\mathbf{M}^T$;
 - Decomposição em Valores Singulares:**
 - 6: Calcular os autovalores e autovetores de \mathbf{S} ;
 - 7: Computar as d Matrizes Elementares (Eq. 3.6);
 - Agrupamento:**
 - 8: Definição das n Matrizes Agrupadas (Eq. 3.8);
 - Média Diagonal:**
 - 9: Obtenção das n componentes não-observáveis c_k^j (Eq. 3.11).
-

A.2 Algoritmo da Etapa de Pré-processamento do PRSSA+ENFN

O algoritmo a seguir descreve os passos da Etapa de Pré-processamento do método PRSSA+ENFN proposto nessa tese.

Algoritmo 2 Etapa de Pré-processamento do Método PRSSA+ENFN.

- 1: Determinação dos parâmetros relacionados ao problema de previsão: H, T, N, m, p, n_{max} (Parte 1 da Tabela 3);
 - 2: Determinação dos parâmetros críticos do PRSSA+ENFN: W, n_l, ϕ, η_0 (Parte 2 da Tabela 3);
 - 3: Calcule $n = \sum_{l=1}^m n_l$;
 - 4: **para** $l \leftarrow 1$ **até** m **faça**
 - 5: Formular \mathbf{Y}_W^l para $y_k^l (k = 1, \dots, W)$ (Equação 5.4);
 - 6: Calcular Matriz Trajetória \mathbf{M}_W^l (Equação 5.5);
 - 7: Computar Matriz Covariância Inicial \mathbf{S}_W^l (Equação 5.6);
 - 8: Obter \mathbf{U}_W^l e Σ_W^l ;
 - 9: Etapa de Agrupamento (Seção 3.3.1) e Média Diagonal (Seção 3.3.2) do SSA;
 - 10: Formular as componentes \mathbf{c}_k^l para $k = 1, \dots, W$ (Equação 5.1);
 - 11: **fim para**
 - 12: Obter \mathfrak{C}_k , para $k = 1, \dots, W$ (Equação 5.3);
 - 13: Vetor de componentes padronizados $\hat{\mathfrak{C}}_k$ para $k = 1, \dots, W$ (Equações 5.7 e 5.8);
 - 14: **para** $j \leftarrow 1$ **até** n **faça**
 - 15: Estimar s_j para cada componente $\hat{\mathfrak{C}}_k$ usando as funções de autocorrelação amostral;
 - 16: **fim para**
 - 17: Inicialização da ENFN (Parte 3 da Tabela 3);
 - 18: Gerar o primeiro *data cloud* ou regra *fuzzy* e ajustar seus parâmetros (Parte 4 da Tabela 3).
-

A.3 Algoritmo da Etapa de Treinamento Incremental do PRSSA+ENFN

O Algoritmo a seguir exibe em linhas gerais os procedimentos da etapa de treinamento da PRSSA+ENFN.

Algoritmo 3 Etapa de Treinamento Incremental do PRSSA+ENFN

```

1: para  $k \leftarrow W + 1$  até  $T$  faça
2:   para  $l \leftarrow 1$  até  $m$  faça
3:     Formular vetor de atualização  $D_k^l$  (Equação 5.9);
4:     Atualizar  $D_k^l$  e a Matriz Covariância  $S_k^l$  (Equação 5.10);
5:     Calcular  $F_k^l$  (Equação 5.15);
6:     Computar as Matrizes de Perturbação  $P_k^l$  (Equação 5.16) e  $Q_k^l$  (Equação 5.17);
7:     Atualizar  $U_k^l$  e  $\Sigma_k^l$  (Equação 5.19);
8:     Determinar as componentes (Eq. 5.20 e 5.21).
9:   fim para
10:  Padronização do vetor de componentes  $\hat{c}_k$  (Eq. 5.23);
11:  Atualizar  $\mu_k$  e  $X_k$  (Eq. 5.24);
12:  Calcular as medidas de densidade  $D_{global}$ ,  $D_{center}^i$ ,  $D_{local}^i$  (Eq. 5.25, 5.26, 5.27);
13:  se Condição 1 é verdadeira (Eq. 5.28) então
14:    se Condição 2 é verdadeira (Eq. 5.29) então
15:      Sobreposição excessiva de Data Clouds (Eq. 5.31);
16:    senão
17:      Novo Data Cloud ( $R = R + 1$ );
18:      Inicializar parâmetros do novo Data Cloud (Parte 4 da Tabela 3);
19:    fim se
20:  senão
21:    Associar o ponto ao data cloud  $i^*$  mais próximo (Eq. 5.32);
22:  fim se
23:  Calcular o grau de ativação  $\lambda_k^i$  das  $R$  regras fuzzy (Eq. 5.33);
24:  Computar a utilidade  $\eta_i$  das  $R$  regras fuzzy (Eq. 5.34);
25:  se Condição 3 é verdadeira (Eq. 5.35) então
26:    Remover regra fuzzy;
27:  fim se
28:  Estimação Recursiva das Matrizes dos Consequentes (Eq. 5.38);
29:  Obter as saídas do PRSSA+ENFN (Eq. 5.39 e 5.40);
30:  Vetor de componentes na escala original (Eq. 5.41);
31:  Resultados de Treinamento (Eq. 5.42).
32: fim para

```

A.4 Algoritmo da Etapa de Previsão do PRSSA+ENFN

O algoritmo a seguir exibe em linhas gerais os procedimentos da etapa de previsão da metodologia proposta.

Algoritmo 4 Etapa de Previsão do PRSSA+ENFN.

```

1: se Previsão Recorrente então
2:   para  $k \leftarrow T + 1$  até  $T + H$  faça
3:     Obter componentes realimentadas da ENFN  $\hat{\mathbf{C}}_{k-p}^*$  e  $\hat{\mathbf{C}}_{k-s}^*$ ;
4:     Computar  $D_{local}^i$  (Equação 5.27) e  $\lambda_k^i$  (Equação 5.33);
5:     Calcular a previsão local de cada regra fuzzy  $\hat{\mathbf{C}}_k^{i*}$  (Equação 5.43);
6:     Calcular a previsão global  $\hat{\mathbf{C}}_k^*$  (Equação 5.43);
7:     Componentes na escala original  $\mathbf{C}_k^*$  (Equação 5.41);
8:     Computar a previsão da série temporal  $\mathbf{y}_k^*$  (Equação 5.44).
9:   fim para
10: fim se
11: se Previsão Direta então
12:   para  $k \leftarrow T + 1$  até  $T + H$  faça
13:     para  $l \leftarrow 1$  até  $m$  faça
14:       Formular vetor de atualização  $\mathbf{D}_k^l$  (Equação 5.9);
15:       Atualizar  $\mathbf{D}_k^l$  e a Matriz Covariância  $\mathbf{S}_k^l$  (Equação 5.10);
16:       Calcular  $\mathbf{F}_k^l$  (Equação 5.15);
17:       Computar as Matrizes de Perturbação  $\mathbf{P}_k^l$  (Equação 5.16) e  $\mathbf{Q}_k^l$  (Equação 5.17);
18:       Atualizar  $\mathbf{U}_k^l$  e  $\mathbf{\Sigma}_k^l$  (Equação 5.19);
19:       Determinar as componentes (Eq. 5.20 e 5.21).
20:     fim para
21:     Padronização das componentes  $\hat{\mathbf{C}}_k$  (Equação 5.23);
22:     Computar  $D_{local}^i$  (Equação 5.27) e  $\lambda_k^i$  (Equação 5.33);
23:     Calcular a previsão local de cada regra fuzzy  $\hat{\mathbf{C}}_k^{i*}$  (Equação 5.45);
24:     Calcular a previsão global  $\hat{\mathbf{C}}_k^*$  (Equação 5.43);
25:     Componentes na escala original  $\mathbf{C}_k^*$  (Equação 5.41);
26:     Computar a previsão da série temporal  $\mathbf{y}_k^*$  (Equação 5.44).
27:   fim para
28: fim se

```

A.5 Algoritmo para Seleção de Parâmetros

O algoritmo a seguir exibe em linhas gerais os procedimentos da etapa de previsão da metodologia proposta.

Algoritmo 5 Seleção de Parâmetros do PRSSA+ENFN.

```

1: se Algoritmo Genético então
2:   Definir quantidade de gerações ( $Q_g$ ), taxa de mutação ( $T_m$ ) e tamanho da população ( $T_p$ );
3:   Gerar população inicial de cromossomos como na Equação 5.46;
4:   para  $r \leftarrow 1$  até  $T_p$  faça
5:     Executar o PRSSA+ENFN aplicando os parâmetros do cromossomo  $r$ ;
6:     Avaliar desempenho do PRSSA+ENFN para o cromossomo  $r$ .
7:   fim para
8:   enquanto número da geração  $< Q_g$  faça
9:     Identificar o cromossomo de elite e mantê-lo na próxima geração;
10:    enquanto quantidade de cromossomos  $< T_p$  faça
11:      Seleção de pais por Roleta Viciada;
12:      Crossover Uniforme;
13:      se mutação então
14:        Inserir cromossomo mutante;
15:      fim se
16:    fim enquanto
17:    para  $r \leftarrow 1$  até  $T_p$  faça
18:      Executar o PRSSA+ENFN aplicando os parâmetros do cromossomo  $r$ ;
19:      Avaliar desempenho do PRSSA+ENFN para o cromossomo  $r$ .
20:    fim para
21:    se função fitness for atendida então
22:      Interromper algoritmo.
23:    fim se
24:  fim enquanto
25: fim se
26: se Seleção Aleatória de Parâmetros então
27:   Definir  $T_p$ , que é a quantidade de operações do PRSSA+ENFN que serão avaliadas;
28:   enquanto quantidade de operações  $< T_p$  faça
29:     Determinação dos parâmetros específicos do PRSSA+ENFN:  $W, n_l, \phi, \eta_0$  (Parte 2 da Tabela 3);
30:     Aplicar parâmetros selecionados no PRSSA+ENFN.
31:   fim enquanto
32: fim se

```

APÊNDICE B – MÉTODO MRSSA PARA EXTRAÇÃO DE COMPONENTES NÃO-OBSERVÁVEIS

B.1 Comentários Iniciais sobre o MRSSA

Essa seção apresenta o método MRSSA (*Multivariate Recursive Singular Spectrum Analysis*), uma hipótese para extração de componentes não-observáveis desenvolvida durante essa pesquisa. O método PRSSA apresentado na metodologia dessa tese realiza uma previsão paralela para cada série temporal, sendo que, para cada uma, tem-se um procedimento computacional exclusivo, inclusive com quantidade de componentes não-observáveis distintas. A alternativa MRSSA busca fazer uma extração única com todas as m séries temporais de maneira conjunta, contendo apenas um valor para n a todas as séries.

Optou-se por investigar as consequências do uso do MRSSA nos resultados do método proposto em trabalhos futuros, pois existe a necessidade de uma extensa análise da sua aplicação, vantagens e desvantagens. Para dar um direcionamento futuro ao leitor, a descrição geral dessa ideia desenvolvida durante a tese será apresentada nas próximas seções.

B.2 Condições Iniciais do MRSSA

O método MRSSA tem o papel de extrair diretamente as componentes não-observáveis de uma série temporal multivariável. O MRSSA é organizado de forma similar ao PRSSA descrito na metodologia dessa tese. Logo, os parâmetros iniciais são os mesmos, conforme a seguir:

- W : quantidade de observações da série temporal para o pré-processamento, sendo que $W < T$;
- n : quantidade de componentes não-observáveis extraídas da série temporal multivariável;
- m : quantidade de séries temporais no contexto multivariável ($m \geq 1$).

Da mesma forma que o PRSSA, considera-se $L = n$. Um dos diferenciais do MRSSA é que L e n são os mesmos para todas as séries temporais, enquanto o PRSSA contém um conjunto de parâmetros para cada série univariável. Seja \mathbf{y}_k uma observação de uma série temporal multivariável no instante k , onde $\mathbf{y}_k = [y_k^1 \ y_k^2 \ \cdots \ y_k^m]^\top$. Uma janela inicial de observações da

série temporal multivariável denotada por:

$$\mathbf{Y}_W = \begin{bmatrix} y_1^1 & y_1^2 & \cdots & y_1^m \\ y_2^1 & y_2^2 & \cdots & y_2^m \\ \vdots & \vdots & \vdots & \vdots \\ y_W^1 & y_W^2 & \cdots & y_W^m \end{bmatrix}_{W \times m} \quad (\text{B.1})$$

ou seja, cada coluna de \mathbf{Y}_W corresponde as W primeiras observações de cada uma das m séries temporais que compõem a multivariável. A Matriz Trajetória nesse caso é representada por:

$$\mathbf{M}_W = \begin{bmatrix} y_1^1 & y_2^1 & \cdots & y_{\mathcal{K}_W}^1 & \cdots & y_1^m & y_2^m & \cdots & y_{\mathcal{K}_W}^m \\ y_2^1 & y_3^1 & \cdots & y_{\mathcal{K}_W+1}^1 & \cdots & y_2^m & y_3^m & \cdots & y_{\mathcal{K}_W+1}^m \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ y_L^1 & y_{L+1}^1 & \cdots & y_W^1 & \cdots & y_L^m & y_{L+1}^m & \cdots & y_W^m \end{bmatrix}_{L \times (\mathcal{K}_W \cdot m)}, \quad (\text{B.2})$$

onde $\mathcal{K}_W = W - L + 1$. Por conseguinte, calcula-se a Matriz Covariância inicial:

$$\mathbf{S}_W = [\mathbf{M}_W][\mathbf{M}_W]^\top, \quad \mathbf{S}_W \in \mathbb{R}^{L \times L}. \quad (\text{B.3})$$

Nota-se que a Matriz Trajetória \mathbf{M}_W comporta todas as W primeiras amostras da série multivariável nesse caso, diferente do processo do PRSSA, em que a Matriz Trajetória \mathbf{M}_W^l era obtida para cada série l (Equação 5.5). Para terminar as condições iniciais, determina-se os autovetores \mathbf{U}_W e autovalores $\mathbf{\Sigma}_W$ de \mathbf{S}_W , similarmente ao processo feito para o PRSSA.

B.3 Atualização Recursiva da Matriz Covariância pelo MRSSA

Essa etapa é muito semelhante a do PRSSA, sendo que o diferencial está apenas na Matriz de Atualização a seguir:

$$\mathbf{D}_t = \begin{bmatrix} y_{\mathcal{K}_k}^1 & y_{\mathcal{K}_k}^2 & \cdots & y_{\mathcal{K}_k}^m \\ y_{\mathcal{K}_k+1}^1 & y_{\mathcal{K}_k+1}^2 & \cdots & y_{\mathcal{K}_k+1}^m \\ \vdots & \vdots & \cdots & \vdots \\ y_k^1 & y_k^2 & \cdots & y_k^m \end{bmatrix}_{L \times m}. \quad (\text{B.4})$$

sendo que \mathbf{D}_k é a Matriz de Atualização que contém observações da série multivariável em uma janela de tempo entre \mathcal{K}_k e o instante atual k , em outras palavras, as L últimas observações da série temporal multivariável. Logo, a atualização recursiva da Matriz Covariância torna-se:

$$\mathbf{S}_k = \mathbf{S}_{k-1} + [\mathbf{D}_k][\mathbf{D}_k]^\top, \quad \text{onde } k = W + 1, W + 2, \dots, T, \quad (\text{B.5})$$

$$\mathbf{S}_k \in \mathbb{R}^{L \times L}, \quad \mathbf{D}_k \in \mathbb{R}^{L \times m}.$$

B.4 Atualização dos Autovalores e Matriz de Valores Singulares pelo MRSSA

A definição de algumas matrizes nessa etapa apresenta um diferencial em relação ao método PRSSA. A primeira diferença está na matriz \mathbf{F}_k , pois a dimensão de \mathbf{D}_k foi modificada no MRSSA, logo:

$$\mathbf{F}_k = [\mathbf{U}_{k-1}]^\top \mathbf{D}_k, \quad \mathbf{F}_k \in \mathbb{R}^{L \times m} \quad \text{e} \quad k = W + 1, W + 2, \dots, T, \quad (\text{B.6})$$

onde \mathbf{U}_W é definida na etapa de pré-processamento e \mathbf{D}_k é calculado na Equação B.4. As definições das Matrizes de Perturbação \mathbf{P}_k e \mathbf{Q}_k ficaram da seguinte forma (PEDDANENI et al., 2004):

- Matriz de Perturbação \mathbf{P}_k :

$$\mathbf{P}_k = \begin{bmatrix} p_k^1 & 0 & \cdots & 0 \\ 0 & p_k^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & p_k^L \end{bmatrix}_{L \times L}; \quad (\text{B.7})$$

- Matriz de Perturbação \mathbf{Q}_k :

$$\mathbf{Q}_k = \begin{bmatrix} 0 & q_k^{1,2} & \cdots & q_k^{1,L} \\ -q_k^{1,2} & 0 & \cdots & q_k^{2,L} \\ \vdots & \vdots & \ddots & \vdots \\ -q_k^{1,L} & -q_k^{2,L} & \cdots & 0 \end{bmatrix}_{L \times L}, \quad (\text{B.8})$$

onde:

$$p_k^j = \frac{[f_k^j]^2 - v_{k-1}^j}{k}, \quad j = 1, \dots, L;$$

$$\begin{cases} q_k^{j,r} = 0, & j, r = 1, \dots, L; \text{ quando } j = r; \\ q_k^{j,r} = \frac{f_k^j f_k^r}{k [p_k^r - p_k^j + v_{k-1}^r - v_{k-1}^j]}, & j, r = 1, \dots, L; \text{ quando } j \neq r. \end{cases} \quad (\text{B.9})$$

sendo que p_k^j são os elementos da diagonal da matriz de perturbação \mathbf{P}_k , f_k^j são os elementos do vetor \mathbf{F}_k , $q_k^{j,r}$ são elementos de \mathbf{Q}_k , e v_{k-1}^j, v_{k-1}^r são os valores singulares contidos em $\mathbf{\Sigma}_{k-1}$, considerando $j, r = 1, \dots, L$.

Com a definição de ambas matrizes de perturbação, é possível atualizar recursivamente \mathbf{U}_k e $\mathbf{\Sigma}_k$ conforme a seguir (PEDDANENI et al., 2004):

$$\begin{aligned} \mathbf{U}_k &= \mathbf{U}_{k-1} (\mathbf{I} + \mathbf{Q}_k) \\ \mathbf{\Sigma}_k &= \mathbf{\Sigma}_{k-1} + \mathbf{P}_k. \end{aligned} \quad (\text{B.10})$$

onde \mathbf{I} é a matriz identidade.

B.5 Determinação das Componentes Não-Observáveis pelo MR-SSA

Para encerrar, a determinação das componentes não-observáveis deve se adaptar a mudança de dimensão da Matriz Atualização \mathbf{D}_k (Equação B.4). Para obter as componentes a cada instante k da série multivariável, são calculados n vetores \mathbf{g}_k^j ($j = 1, \dots, n$). Seja $\mathbf{d}_l \in \mathbb{R}^{L \times 1}$ ($l = 1, \dots, m$) a l -ésima coluna de \mathbf{D}_k . Para cada \mathbf{d}_l , realiza-se o seguinte procedimento

$$\mathbf{g}^j = \mathbf{u}_k^j [\mathbf{d}_l]^\top \mathbf{u}_k^j, \quad j = 1, \dots, L = n, \quad (\text{B.11})$$

$$\mathbf{g}^j \in \mathbb{R}^{L \times 1}, \quad \mathbf{u}_k^j \in \mathbb{R}^{L \times 1},$$

sendo que \mathbf{u}_k^j é o autovetor j ou j -ésima coluna de \mathbf{U}_k . Dessa forma, as n componentes de cada série temporal são:

$$\mathbf{y}_k = \mathbf{c}_k^1 + \mathbf{c}_k^2 + \dots + \mathbf{c}_k^n. \quad (\text{B.12})$$

B.6 Resultados de Decomposição do Método MRSSA

Nessa seção, alguns resultados sobre a decomposição do método MRSSA descrito nesse apêndice são apresentados. Em primeiro lugar, a Figura 44 exibe os resultados de decomposição multivariável para quatro séries temporais usando o MRSSA para $n = 3$. Destaca-se que, para esse procedimento, $W = 20$.

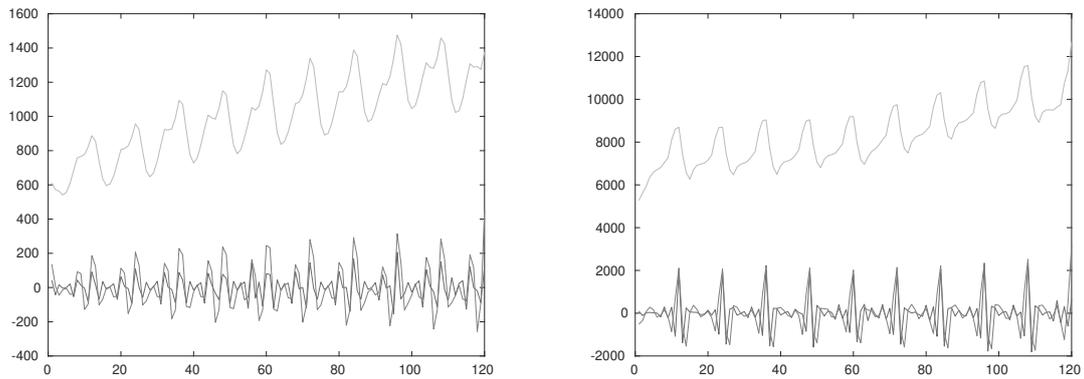
É interessante verificar também se o MRSSA segue a propriedade básica de que a soma das componentes é exatamente igual às séries temporais correspondentes. Para comprovar essa característica, a Figura 45 exibe a soma das componentes não-observáveis apresentadas na Figura 44.

A seguir, compara-se algumas matrizes fundamentais do método MSSA tradicional e MRSSA, verificando se seus valores se aproximam para as decomposições apresentadas. A matriz de autovetores de ambos os métodos ficaram bem próximas em relação aos valores. As diferenças ocorrem porque as Matrizes de Perturbação \mathbf{P}_k e \mathbf{Q}_k são aproximações.

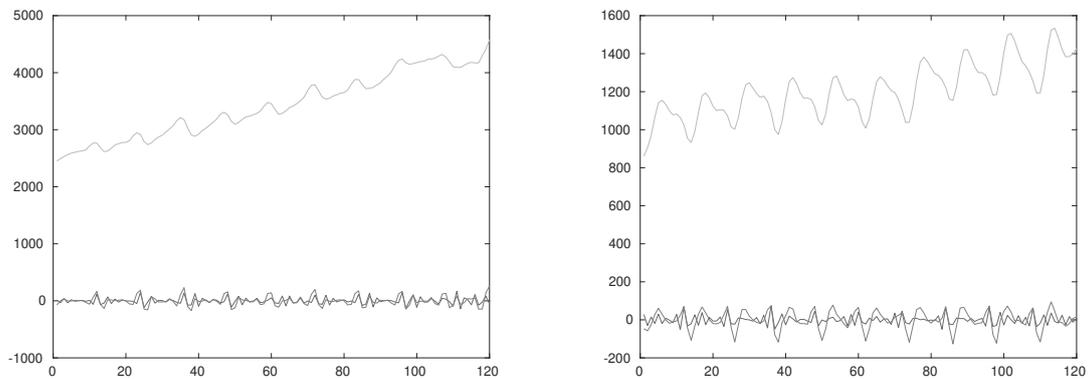
$$\mathbf{U} = \begin{matrix} \text{(MSSA)} \\ \begin{bmatrix} 0.44 & 0.69 & 0.57 \\ -0.82 & 0.04 & 0.58 \\ 0.37 & -0.72 & 0.58 \end{bmatrix} \end{matrix} \quad \mathbf{U}_k = \begin{matrix} \text{(MRSSA)} \\ \begin{bmatrix} 0.41 & 0.71 & 0.57 \\ -0.82 & 0.01 & 0.58 \\ 0.40 & -0.70 & 0.58 \end{bmatrix} \end{matrix} \quad (\text{B.13})$$

Compara-se também a matriz de valores singulares. Percebe-se novamente que são

Figura 44 – Decomposição das Quatro Séries Temporais usando o MRSSA para $n = 3$ e $W = 20$.



(a) Componentes Não-Observáveis para a Série de Ven- (b) Componentes Não-Observáveis para a Série de Ven-
das de *Book*. das de *Clothing*.



(c) Componentes Não-Observáveis para a Série de Ven- (d) Componentes Não-Observáveis para a Série de Ven-
das de *Furniture*. das de *Hardware*.

Fonte: Autor.

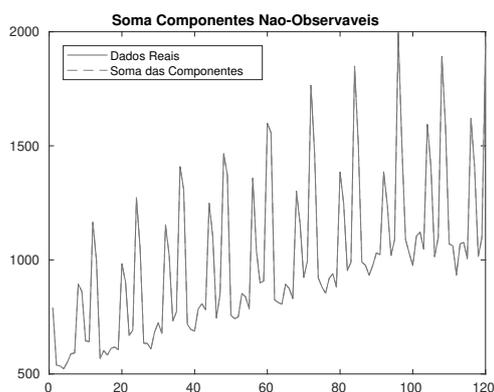
valores próximos e as diferenças ocorrem devido às aproximações das Matrizes de Perturbação.

$$\begin{aligned}
 & \text{(MSSA)} \\
 \Sigma = & \begin{bmatrix} 380337056.63 & 0 & 0 \\ 0 & 552340857.64 & 0 \\ 0 & 0 & 30270105893.73 \end{bmatrix}
 \end{aligned} \tag{B.14}$$

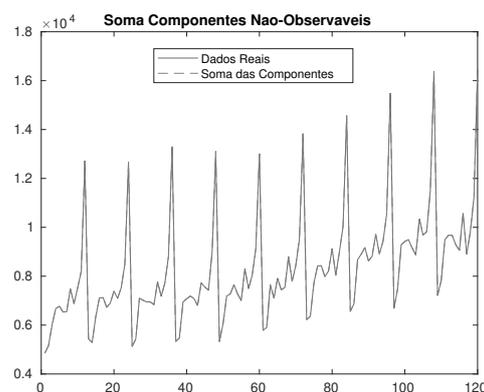
$$\begin{aligned}
 & \text{(MRSSA)} \\
 \Sigma_k = & \begin{bmatrix} 380666878.09 & 0 & 0 \\ 0 & 552432287.98 & 0 \\ 0 & 0 & 30269684641.93 \end{bmatrix}
 \end{aligned}$$

Por fim, ambas Matrizes Covariância são idênticas, o que demonstra o bom desempenho

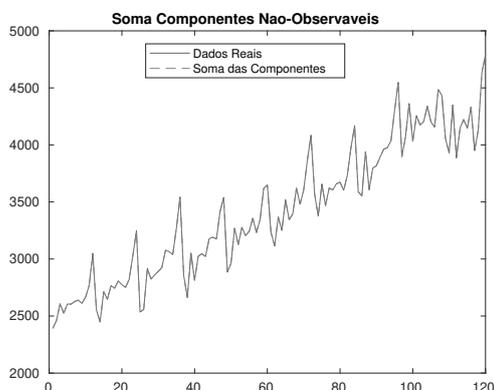
Figura 45 – Comparação entre as Séries Temporais e Soma das Componentes Não-Observáveis da Figura 44.



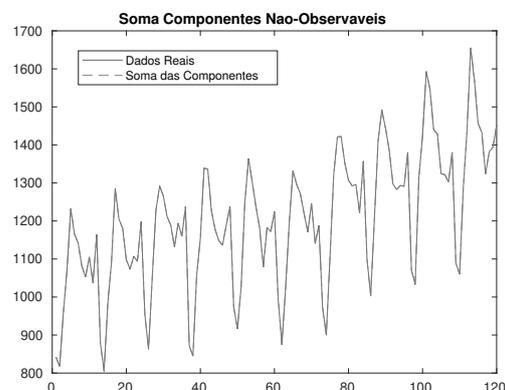
(a) Série de Vendas de *Book*.



(b) Série de Vendas de *Clothing*.



(c) Série de Vendas de *Furniture*.



(d) Série de Vendas de *Hardware*.

Fonte: Autor.

da atualização recursiva do MRSSA.

$$\begin{aligned}
 & \text{(MSSA)} \\
 \mathbf{S} = & \begin{bmatrix} 10232112964.00 & 9875086752.00 & 9871733995.00 \\ 9875086752.00 & 10351838805.00 & 10054248118.00 \\ 9871733995.00 & 10054248118.00 & 10618832039.00 \end{bmatrix} \\
 & \text{(B.15)}
 \end{aligned}$$

$$\begin{aligned}
 & \text{(MRSSA)} \\
 \mathbf{S}_t = & \begin{bmatrix} 10232112964.00 & 9875086752.00 & 9871733995.00 \\ 9875086752.00 & 10351838805.00 & 10054248118.00 \\ 9871733995.00 & 10054248118.00 & 10618832039.00 \end{bmatrix}
 \end{aligned}$$

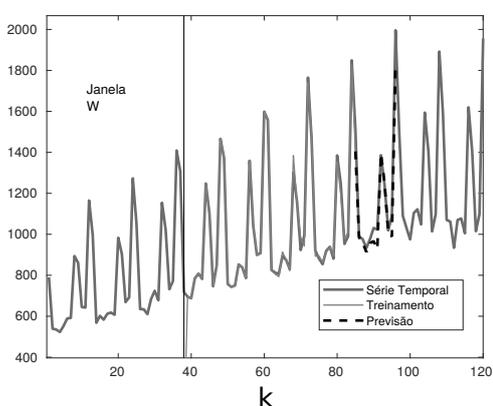
Esse método será aplicado em trabalhos futuros em substituição do PRSSA na metodologia proposta, verificando as vantagens e desvantagens, assim como o impacto no custo computacional segundo uma análise assintótica.

APÊNDICE C – RESULTADOS ADICIONAIS DOS EXPERIMENTOS

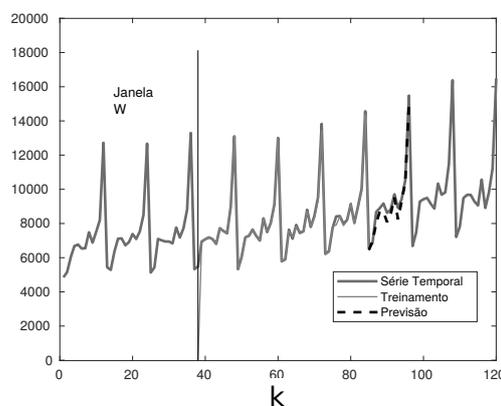
C.1 Resultados Gráficos da Previsão Multivariável Recorrente (Seção 6.2)

C.1.1 Resultados Gráficos para Previsão Recorrente Quando $H = 12$

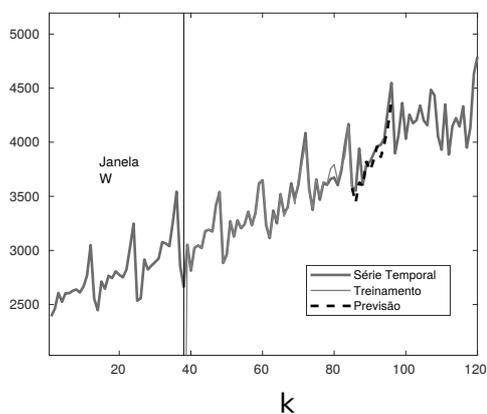
Figura 46 – Resultados de Treinamento e Previsão Multivariável do PRSSA+ENFN com $H = 12$ Considerando o Melhor Cromossomo Selecionado pelo Algoritmo Genético. Parâmetros Críticos: $W = 38$; $n = [2\ 3\ 3\ 3]$, $\phi = 0.14$, $\eta_0 = 0.1$.



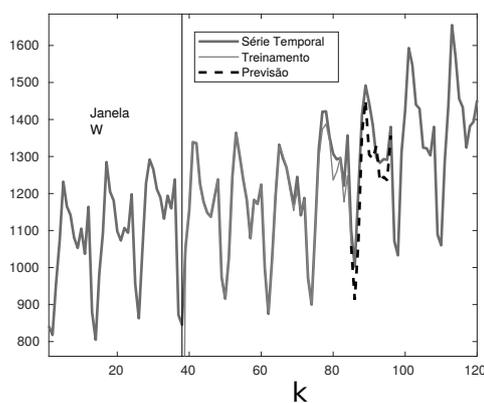
(a) Série Temporal *Book*.



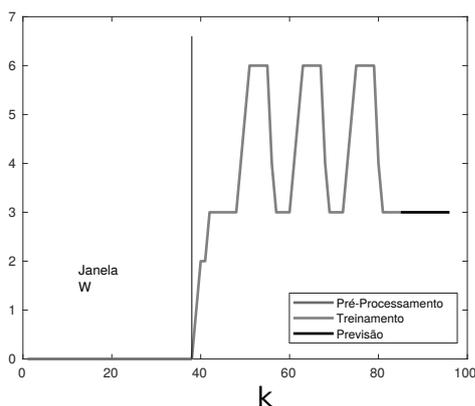
(b) Série Temporal *Clothing*.



(c) Série Temporal *Furniture*.



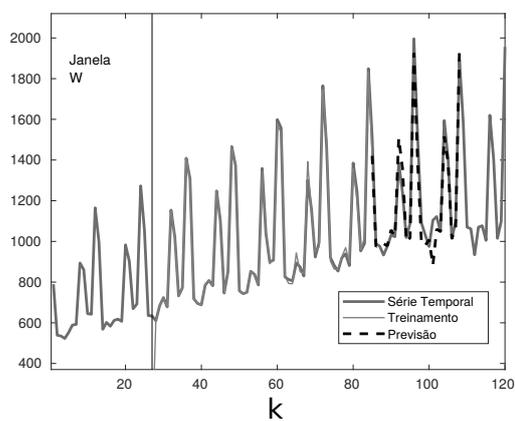
(d) Série Temporal *Hardware*.



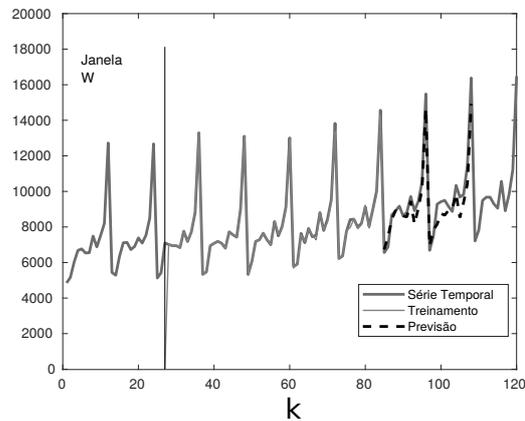
(e) Quantidade de *Clouds* ou Regras *Fuzzy*.

C.1.2 Resultados Gráficos para Previsão Recorrente Quando $H = 24$

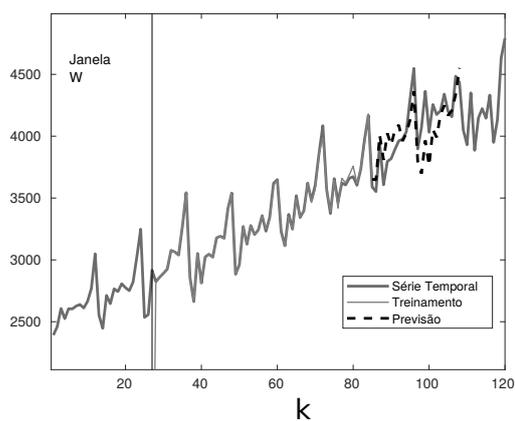
Figura 47 – Resultados de Treinamento e Previsão Multivariável do PRSSA+ENFN com $H = 24$ Considerando o Melhor Cromossomo Selecionado pelo Algoritmo Genético. Parâmetros Críticos: $W = 27$; $n = [3\ 4\ 4\ 3]$, $\phi = 0.13$, $\eta_0 = 0.29$.



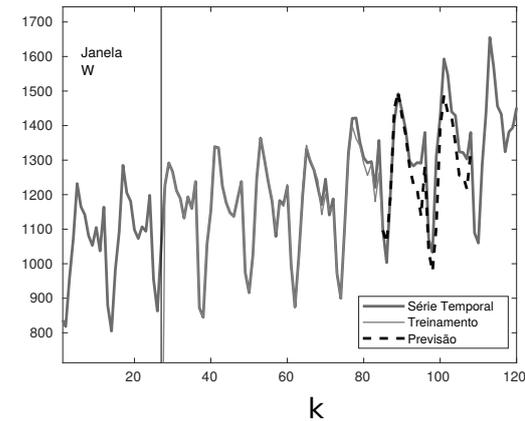
(a) Série Temporal *Book*.



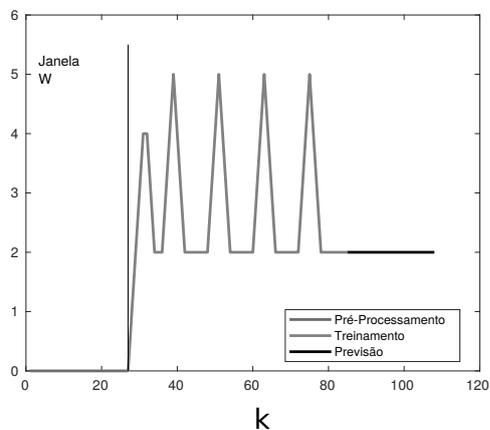
(b) Série Temporal *Clothing*.



(c) Série Temporal *Furniture*.



(d) Série Temporal *Hardware*.

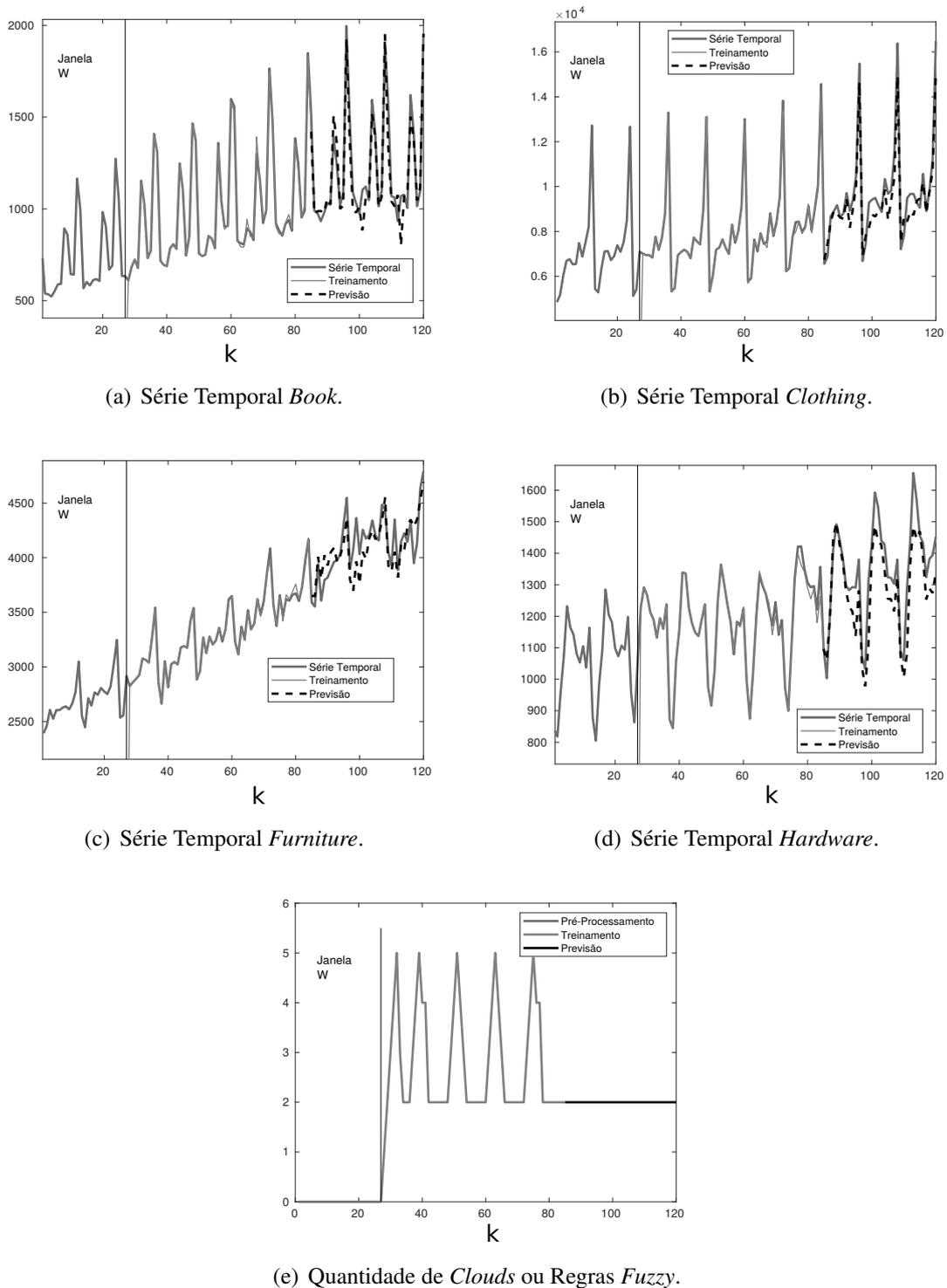


(e) Quantidade de *Clouds* ou Regras *Fuzzy*.

Fonte: Autor.

C.1.3 Resultados Gráficos para Previsão Recorrente Quando $H = 36$

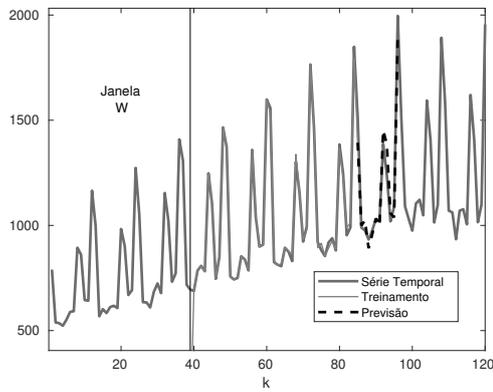
Figura 48 – Resultados de Treinamento e Previsão Multivariável do PRSSA+ENFN com $H = 36$ Considerando o Melhor Cromossomo Selecionado pelo Algoritmo Genético. Parâmetros Críticos: $W = 27$, $n = [3\ 4\ 4\ 3]$, $\phi = 0.32$ e $\eta_0 = 0.24$.



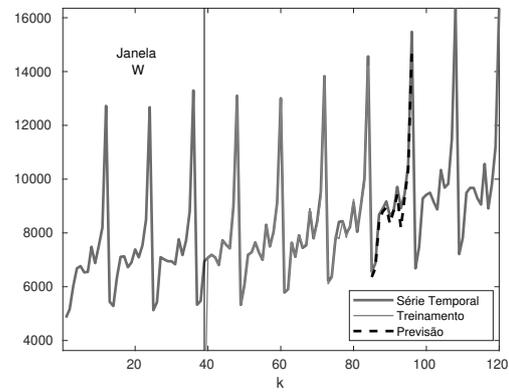
Fonte: Autor.

C.1.4 Resultados Gráficos para Previsão Direta Quando $H = 12$

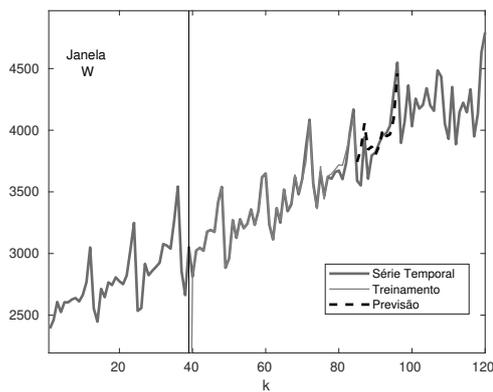
Figura 49 – Resultados de Treinamento e Previsão Multivariável do PRSSA+ENFN com $H = 12$ Considerando o Melhor Cromossomo Selecionado pelo Algoritmo Genético. Parâmetros Críticos: $W = 39$, $n = [2\ 3\ 3\ 3]$, $\phi = 0.5$ e $\eta_0 = 0.25$.



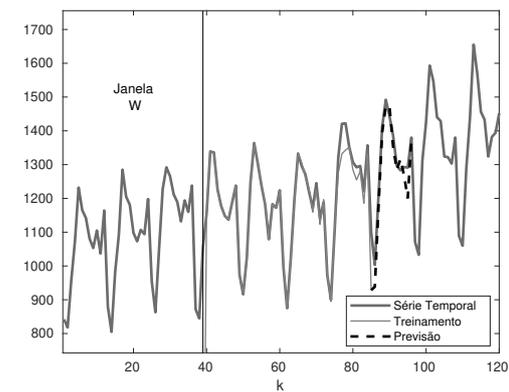
(a) Série Temporal *Book*.



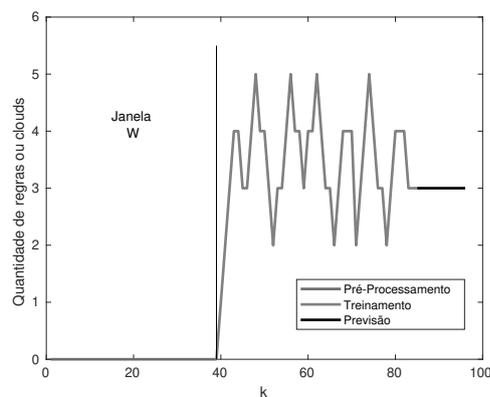
(b) Série Temporal *Clothing*.



(c) Série Temporal *Furniture*.



(d) Série Temporal *Hardware*.

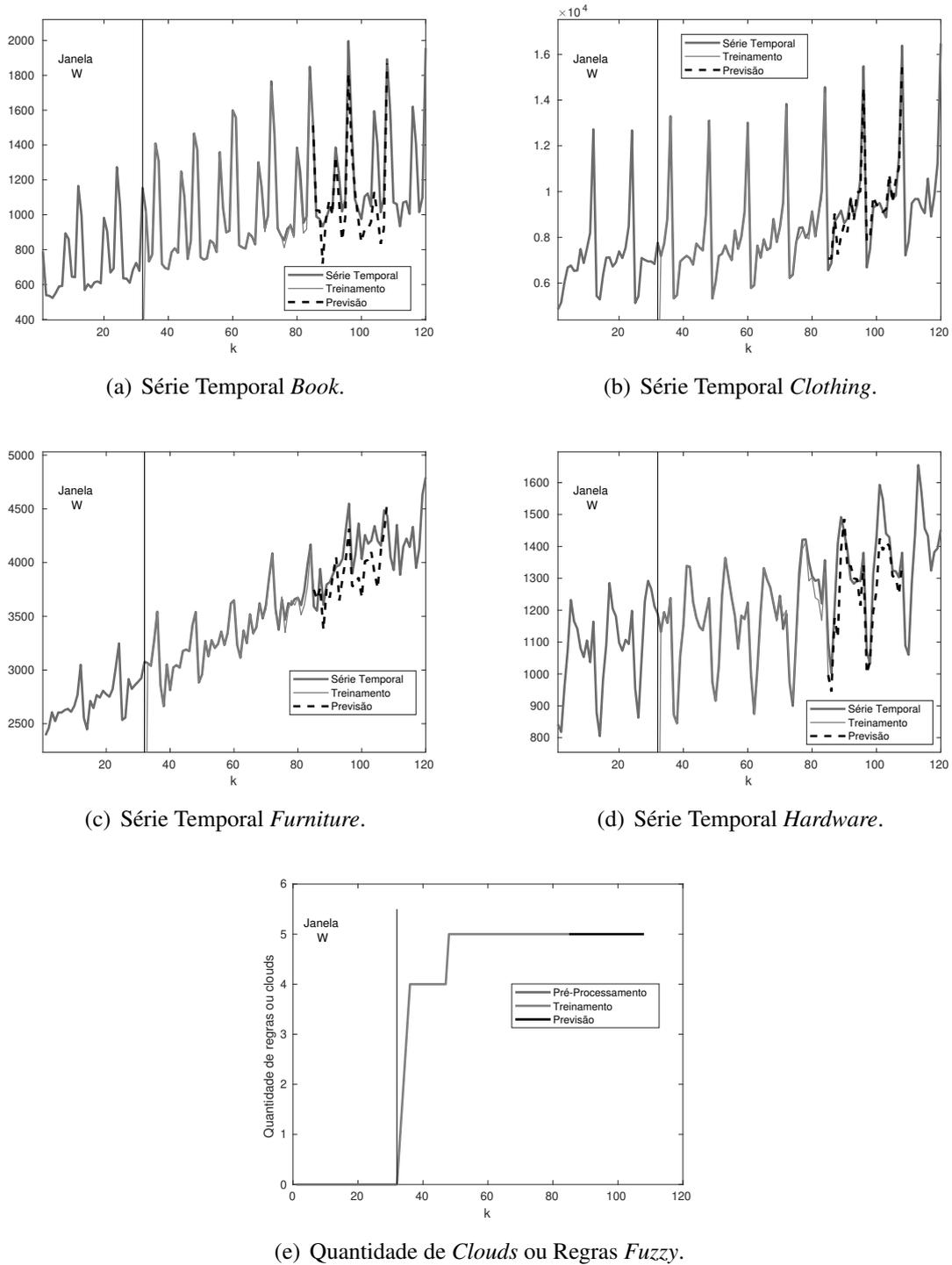


(e) Quantidade de *Clouds* ou Regras *Fuzzy*.

Fonte: Autor.

C.1.5 Resultados Gráficos para Previsão Direta Quando $H = 24$

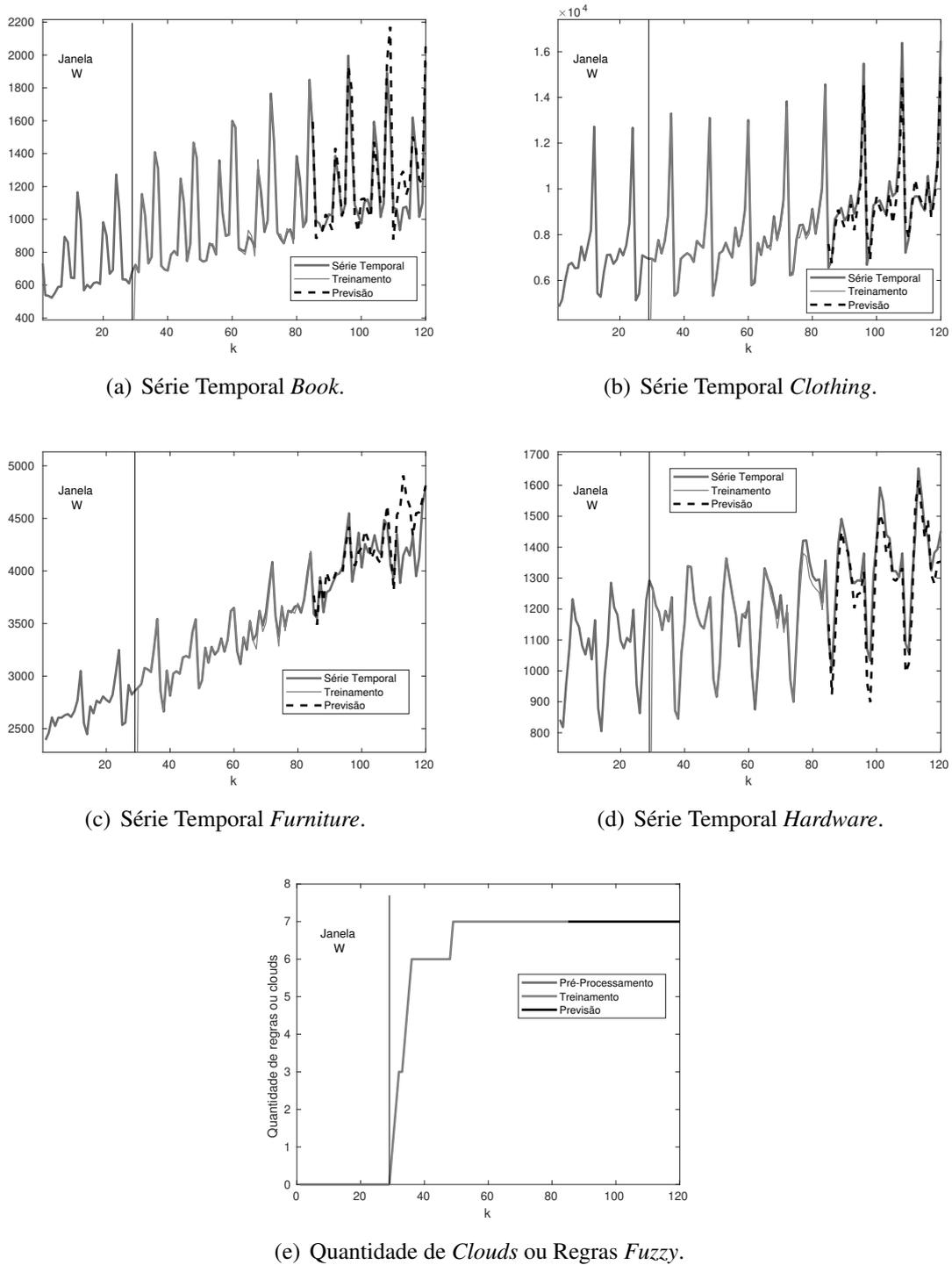
Figura 50 – Resultados de Treinamento e Previsão Multivariável do PRSSA+ENFN com $H = 24$ Considerando o Melhor Cromossomo Selecionado pelo Algoritmo Genético. Parâmetros Críticos: $W = 32$, $n = [6\ 3\ 2\ 5]$, $\phi = 0.1$ e $\eta_0 = 0.04$.



Fonte: Autor.

C.1.6 Resultados Gráficos para Previsão Direta Quando $H = 36$

Figura 51 – Resultados de Treinamento e Previsão Multivariável do PRSSA+ENFN com $H = 36$ Considerando o Melhor Cromossomo Selecionado pelo Algoritmo Genético. Parâmetros Críticos: $W = 29$, $n = [5 \ 2 \ 2 \ 2]$, $\phi = 0.43$ e $\eta_0 = 0.03$.

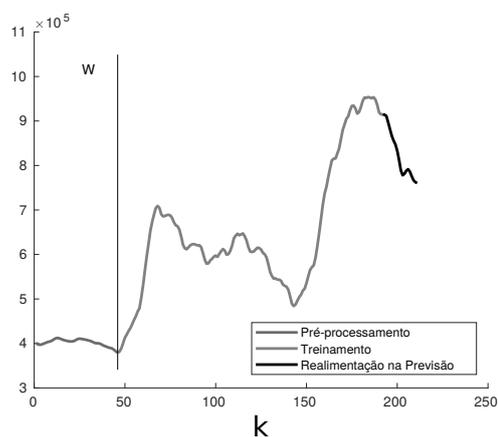


Fonte: Autor.

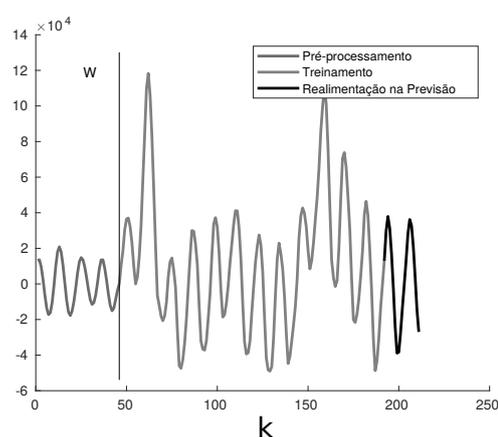
C.2 Resultados Gráficos da Previsão Univariável Recorrente do Experimento 1 (Seção 6.3)

C.2.1 Série Temporal TS1

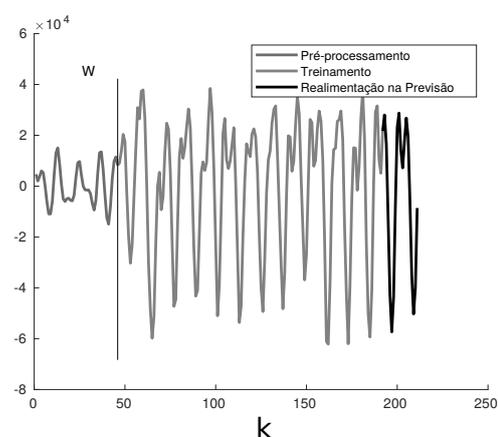
Figura 52 – Componentes Não-Observáveis da Série Temporal TS1 nas Etapas de Pré-Processamento, Treinamento e Previsão Aplicando o Caso 2 do Algoritmo Genético (**Parte 1**).



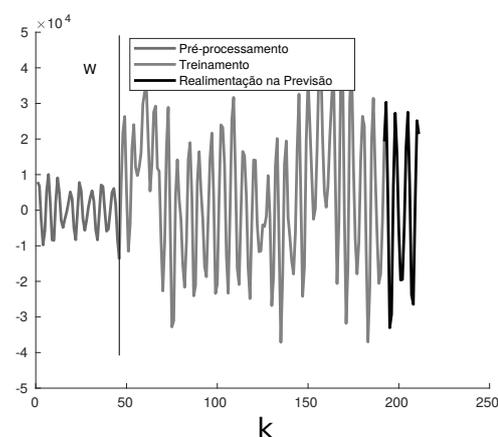
(a) Componente 1.



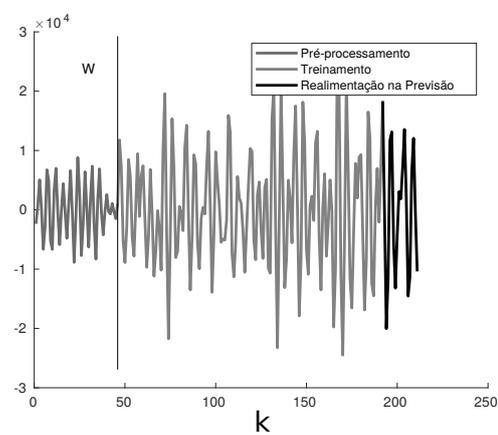
(b) Componente 2.



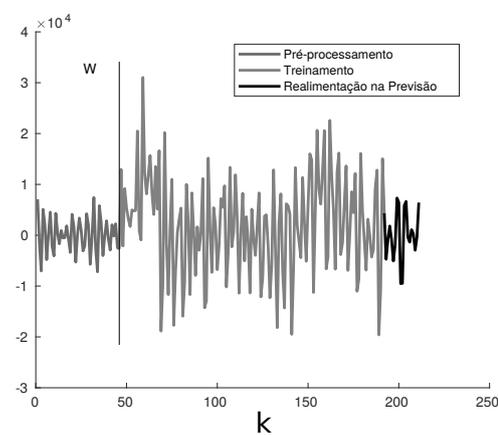
(c) Componente 3.



(d) Componente 4.



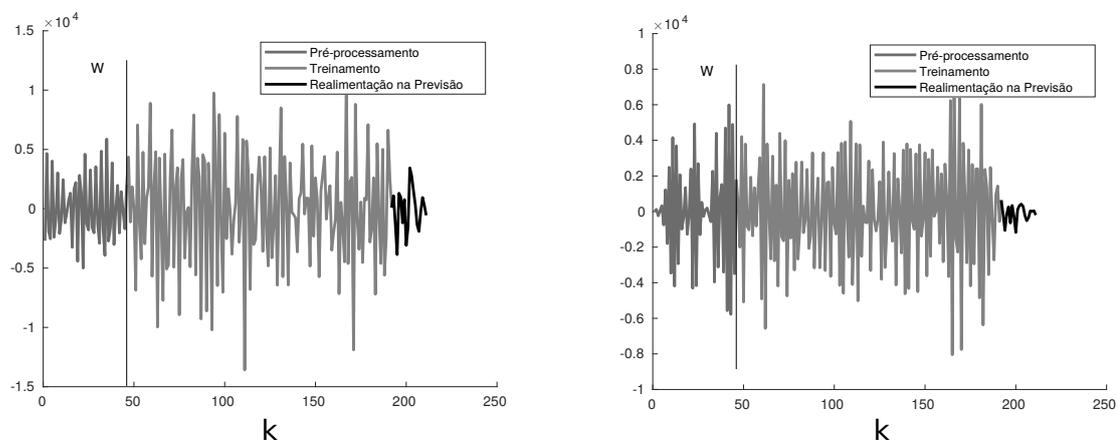
(e) Componente 5.



(f) Componente 6.

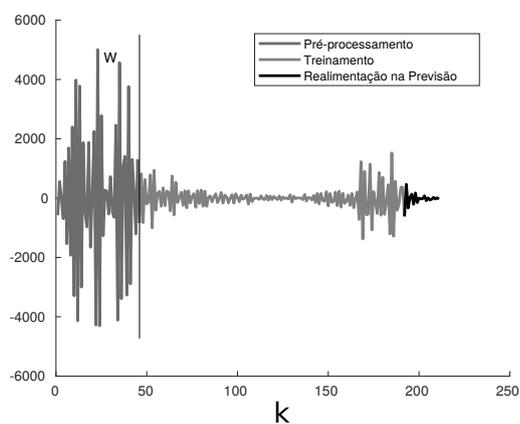
Fonte: Autor.

Figura 53 – Componentes Não-Observáveis da Série Temporal TS1 nas Etapas de Pré-Processamento, Treinamento e Previsão Aplicando o Caso 2 do Algoritmo Genético (**Parte 2**).



(a) Componente 7.

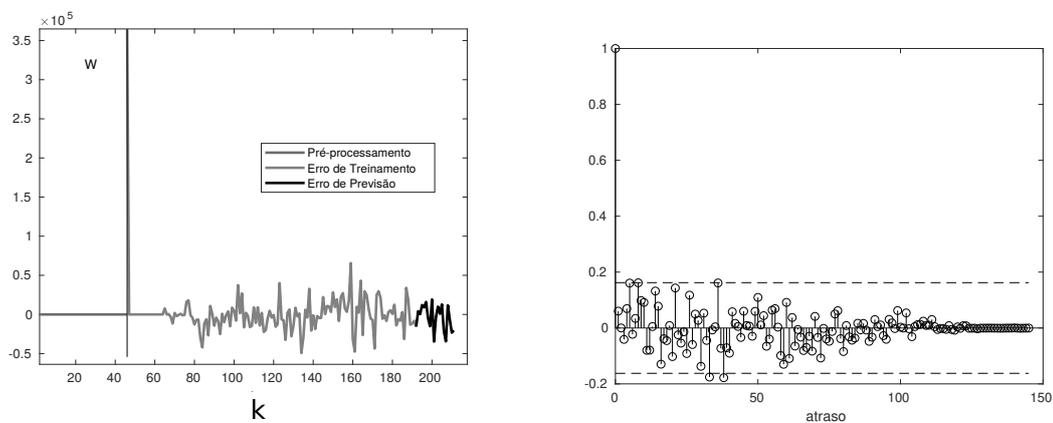
(b) Componente 8.



(c) Componente 9.

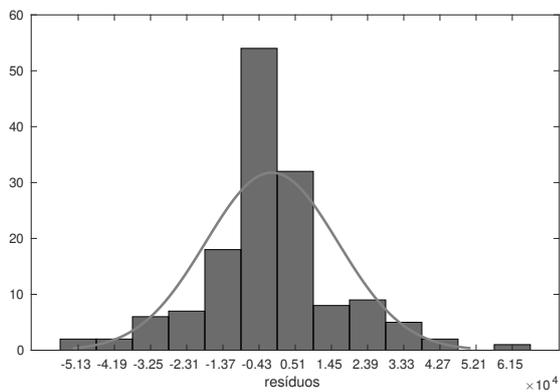
Fonte: Autor.

Figura 54 – Análise dos Erros de Treinamento e Previsão para a Série TS1 Aplicando o Caso 2 do Algoritmo Genético.



(a) Erros em Cada Etapa do PRSSA+ENFN.

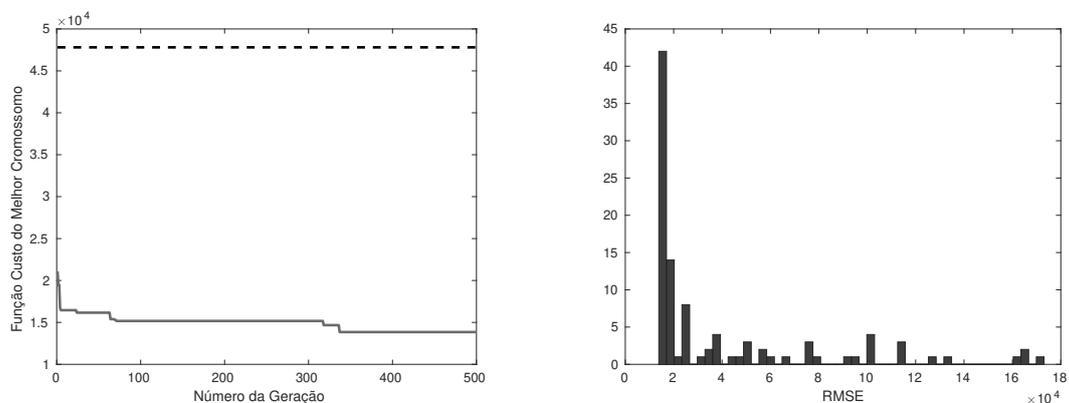
(b) Função de Autocorelação Amostral dos Resíduos de Treinamento.



(c) Histograma dos Resíduos de Treinamento.

Fonte: Autor.

Figura 55 – Informações do Melhor Cromossomo e da População Final do Algoritmo Genético no Caso 2 para a Série Temporal TS1.

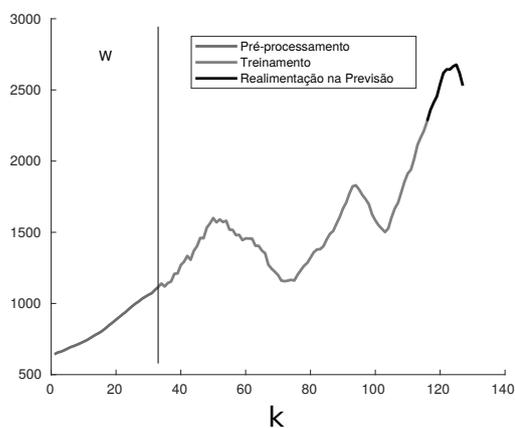


(a) Evolução do RMSE do Melhor Cromossomo Durante as Gerações. (b) Histograma das Medidas RMSE dos Cromossomos da Geração 500.

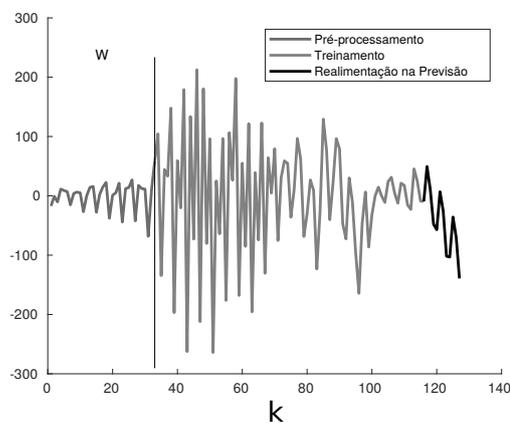
Fonte: Autor.

C.2.2 Série Temporal TS2

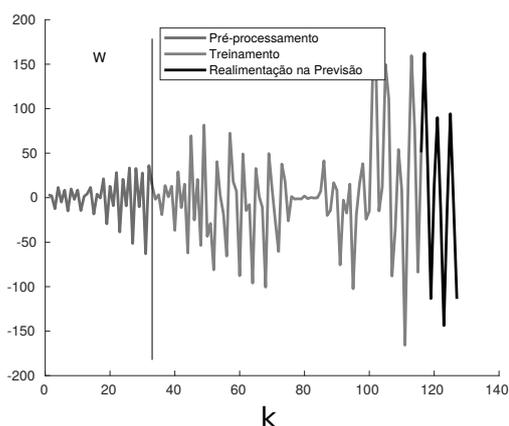
Figura 56 – Componentes Não-Observáveis da Série Temporal TS2 nas Etapas de Pré-Processamento, Treinamento e Previsão Aplicando o Caso 2 do Algoritmo Genético (**Parte 1**).



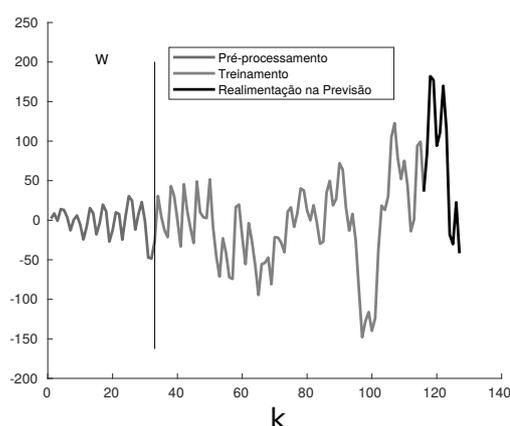
(a) Componente 1.



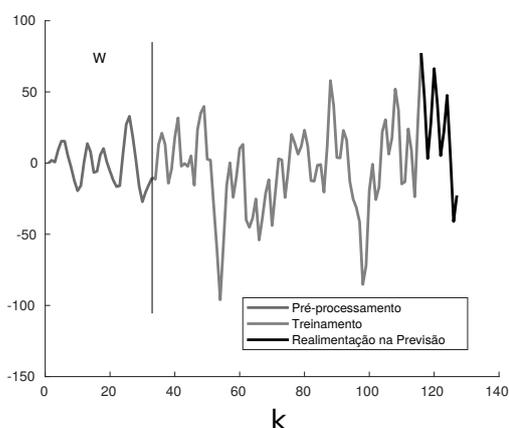
(b) Componente 2.



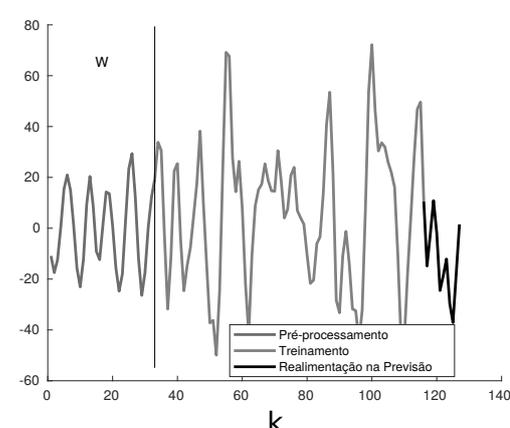
(c) Componente 3.



(d) Componente 4.



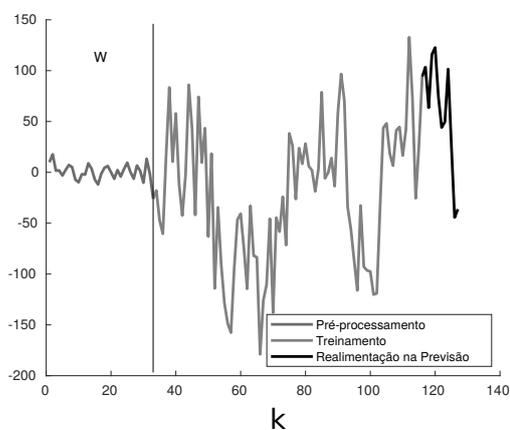
(e) Componente 5.



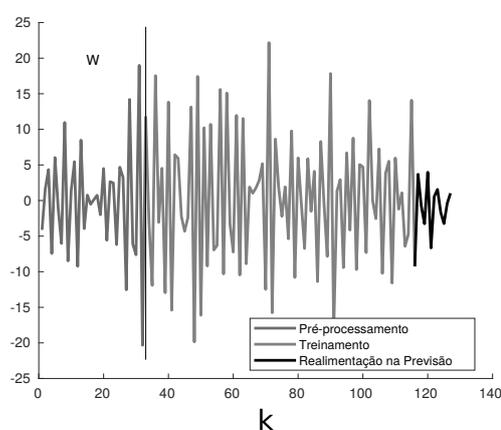
(f) Componente 6.

Fonte: Autor.

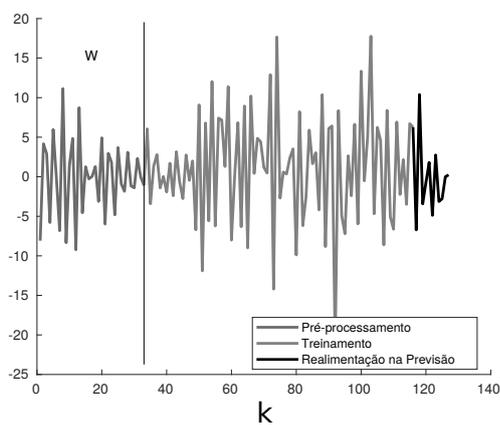
Figura 57 – Componentes Não-Observáveis da Série Temporal TS2 nas Etapas de Pré-Processamento, Treinamento e Previsão Aplicando o Caso 2 do Algoritmo Genético (**Parte 2**).



(a) Componente 7.



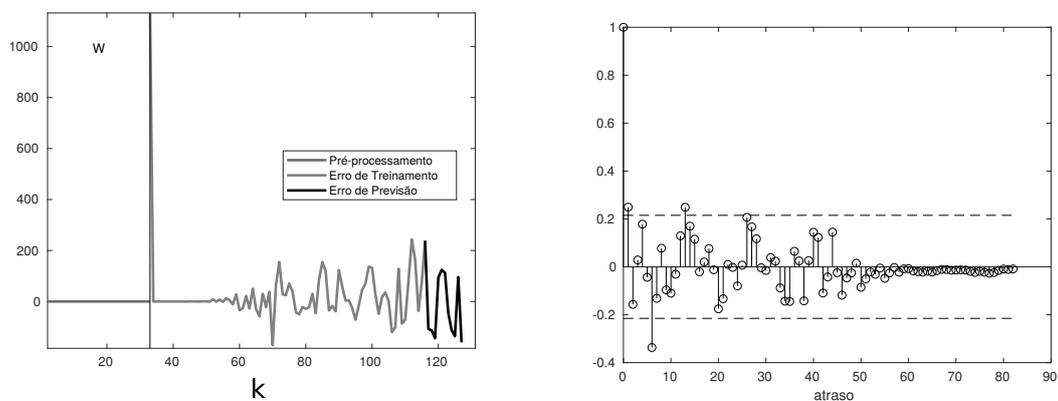
(b) Componente 8.



(c) Componente 9.

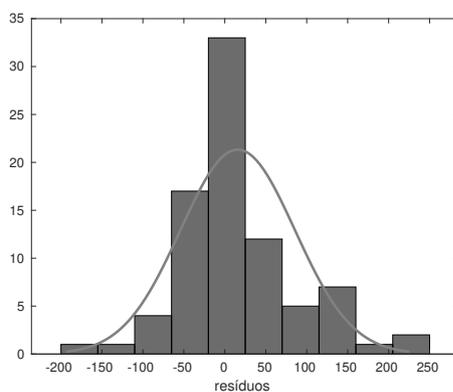
Fonte: Autor.

Figura 58 – Análise dos Erros de Treinamento e Previsão para a Série TS2 Aplicando o Caso 2 do Algoritmo Genético.



(a) Erros em Cada Etapa do PRSSA+ENFN.

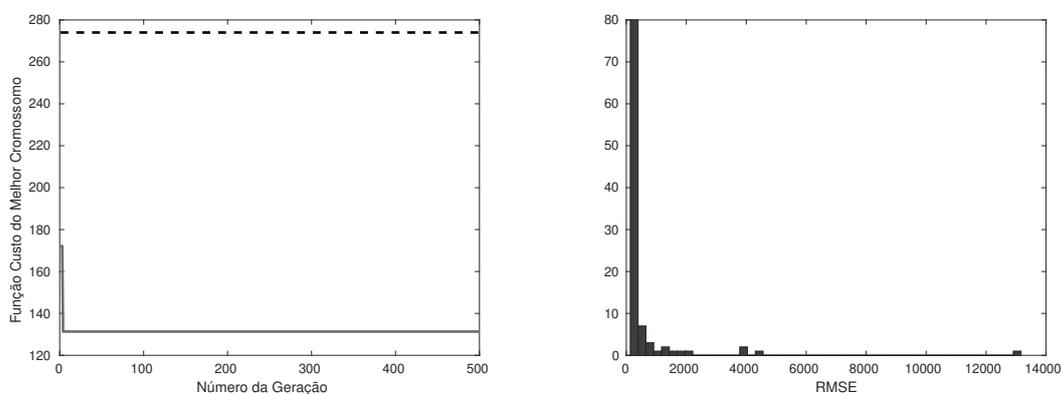
(b) Função de Autocorelação Amostral dos Resíduos de Treinamento.



(c) Histograma dos Resíduos de Treinamento.

Fonte: Autor.

Figura 59 – Informações do Melhor Cromossomo e da População Final do Algoritmo Genético no Caso 2 para a Série Temporal TS2.

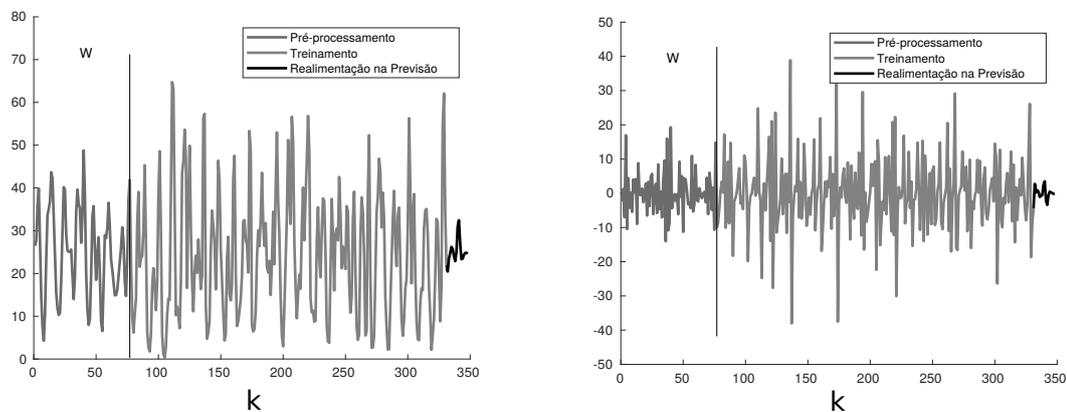


(a) Evolução do RMSE do Melhor Cromossomo Durante as Gerações. (b) Histograma das Medidas RMSE dos Cromossomos da Geração 500.

Fonte: Autor.

C.2.3 Série Temporal TS3

Figura 60 – Componentes Não-Observáveis da Série Temporal TS2 nas Etapas de Pré-Processamento, Treinamento e Previsão Aplicando o Caso 2 do Algoritmo Genético.

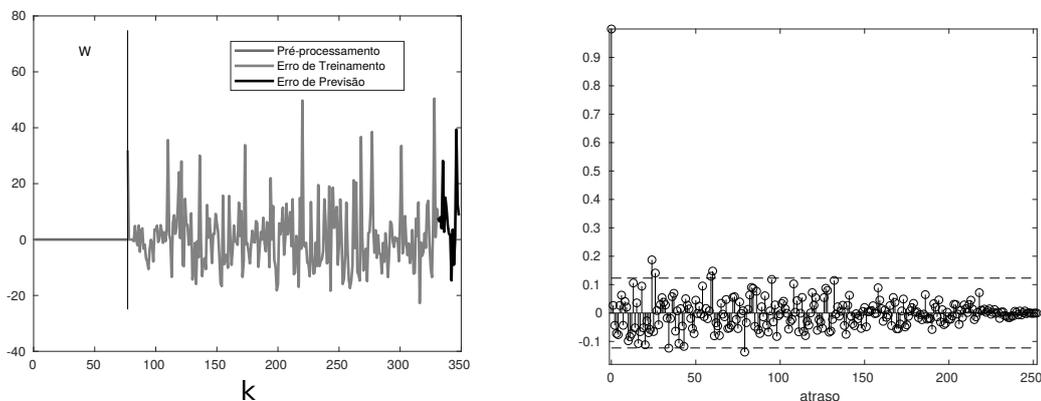


(a) Componente 1.

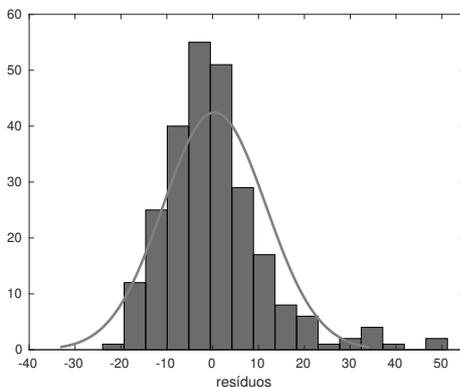
(b) Componente 2.

Fonte: Autor.

Figura 61 – Análise dos Erros de Treinamento e Previsão para a Série TS3 Aplicando o Caso 2 do Algoritmo Genético.



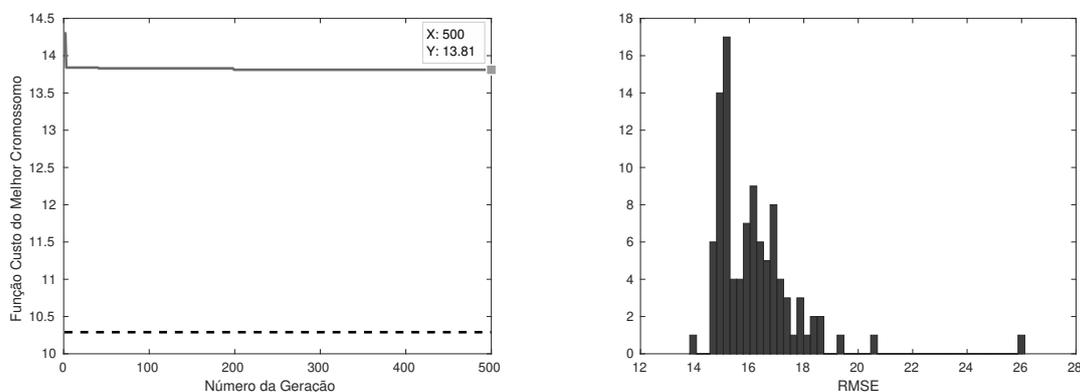
(a) Erros em Cada Etapa do PRSSA+ENFN. (b) Função de Autocorrelação Amostral dos Resíduos de Treinamento.



(c) Histograma dos Resíduos de Treinamento.

Fonte: Autor.

Figura 62 – Informações do Melhor Cromossomo e da População Final do Algoritmo Genético no Caso 2 para a Série Temporal TS3.

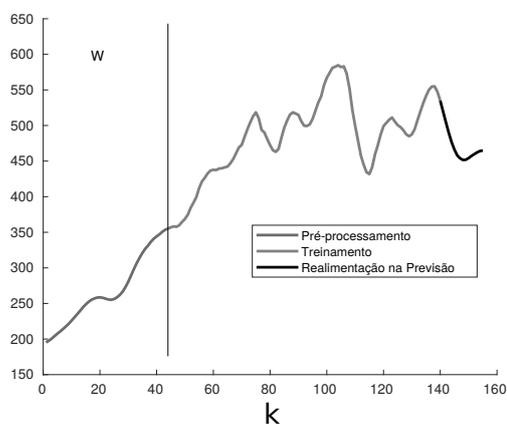


(a) Evolução do RMSE do Melhor Cromossomo Durante as Gerações. (b) Histograma das Medidas RMSE dos Cromossomos da Geração 500.

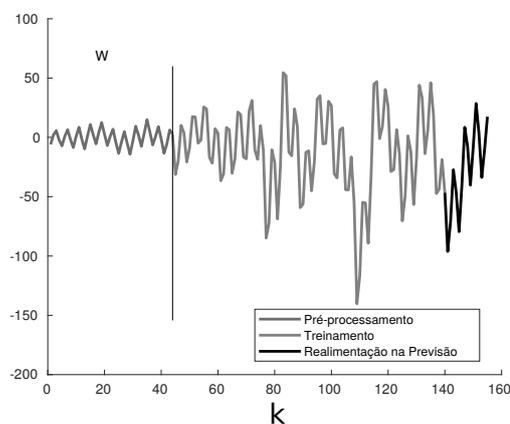
Fonte: Autor.

C.2.4 Série Temporal TS4

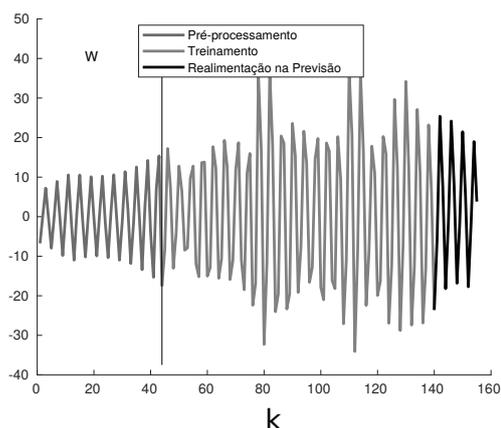
Figura 63 – Componentes Não-Observáveis da Série Temporal TS4 nas Etapas de Pré-Processamento, Treinamento e Previsão Aplicando o Caso 2 do Algoritmo Genético. **(Parte 1)**.



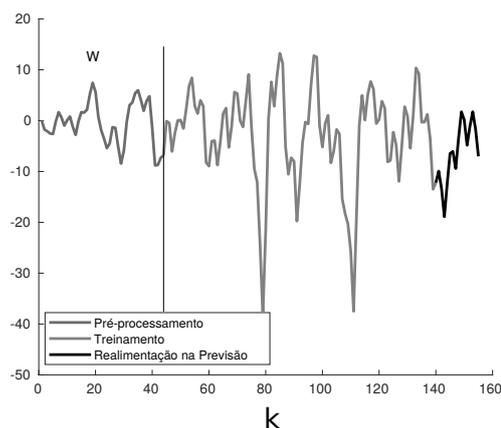
(a) Componente 1.



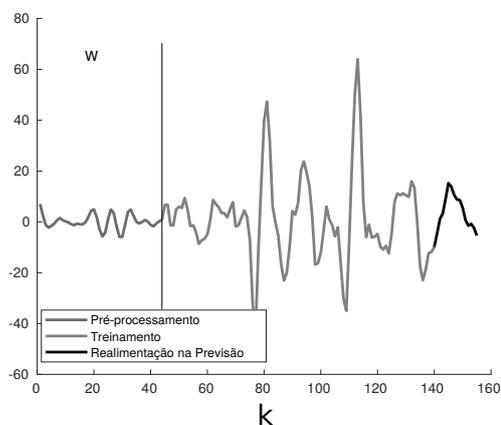
(b) Componente 2.



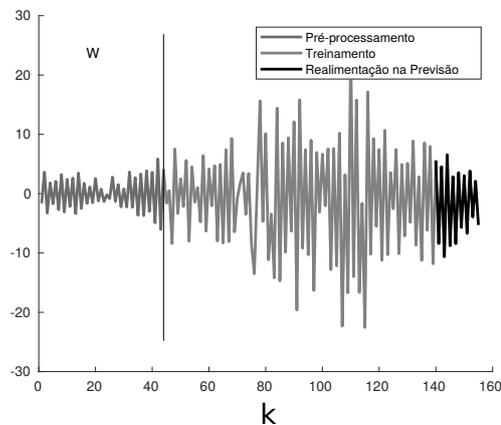
(c) Componente 3.



(d) Componente 4.



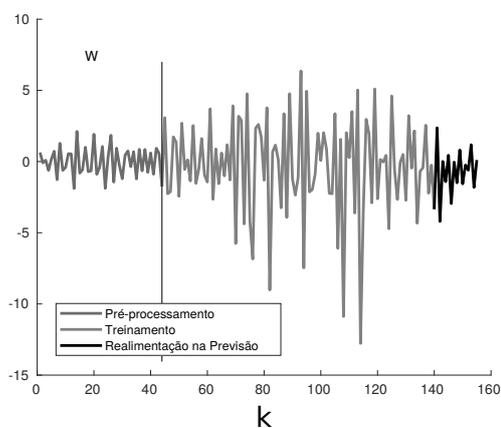
(e) Componente 5.



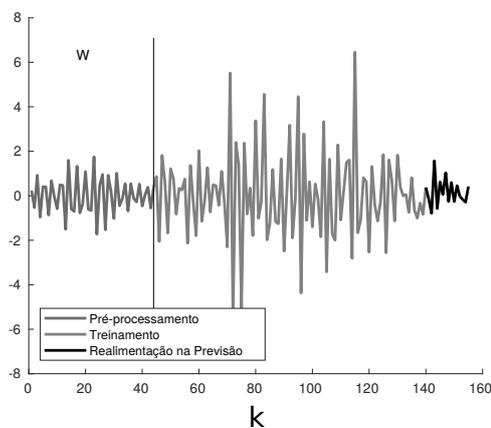
(f) Componente 6.

Fonte: Autor.

Figura 64 – Componentes Não-Observáveis da Série Temporal TS4 nas Etapas de Pré-Processamento, Treinamento e Previsão Aplicando o Caso 2 do Algoritmo Genético. **(Parte 2)**.



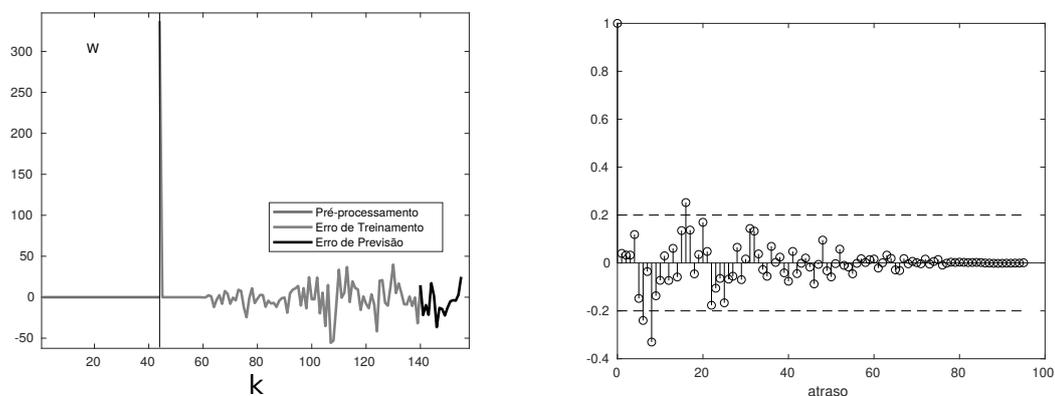
(a) Componente 7.



(b) Componente 8.

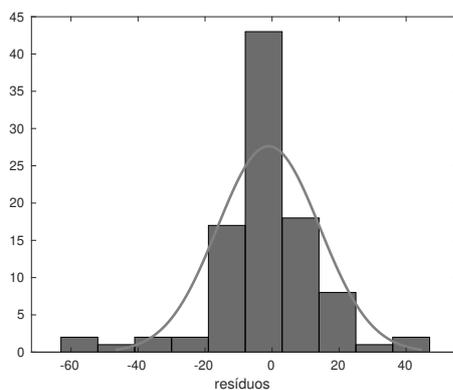
Fonte: Autor.

Figura 65 – Análise dos Erros de Treinamento e Previsão para a Série TS4 Aplicando o Caso 2 do Algoritmo Genético.



(a) Erros em Cada Etapa do PRSSA+ENFN.

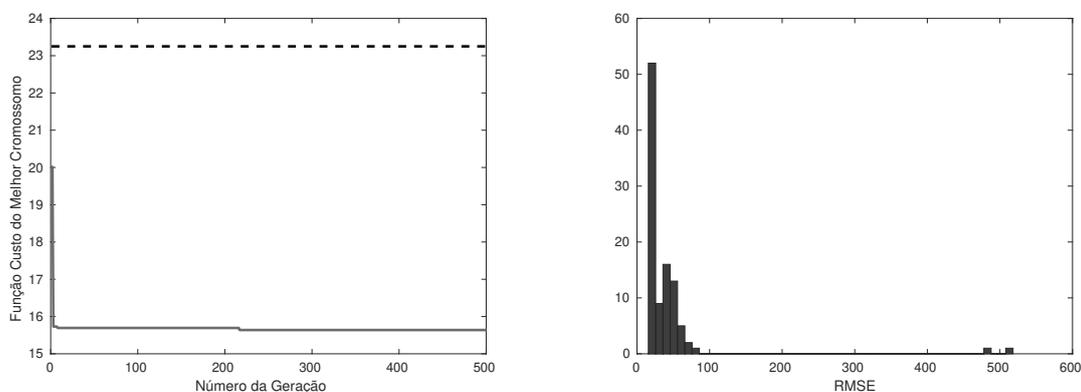
(b) Função de Autocorrelação Amostral dos Resíduos de Treinamento.



(c) Histograma dos Resíduos de Treinamento.

Fonte: Autor.

Figura 66 – Informações do Melhor Cromossomo e da População Final do Algoritmo Genético no Caso 2 para a Série Temporal TS4.



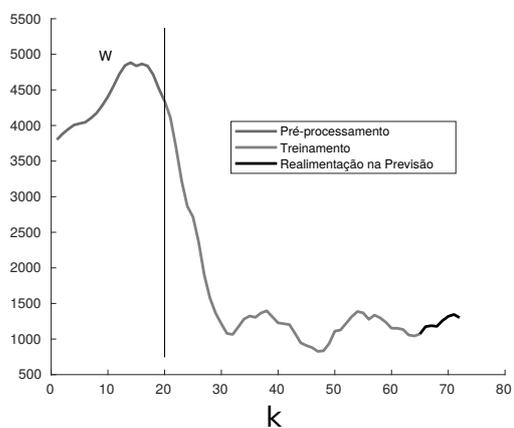
(a) Evolução do RMSE do Melhor Cromossomo Durante as Gerações.

(b) Histograma das Medidas RMSE dos Cromossomos da Geração 500.

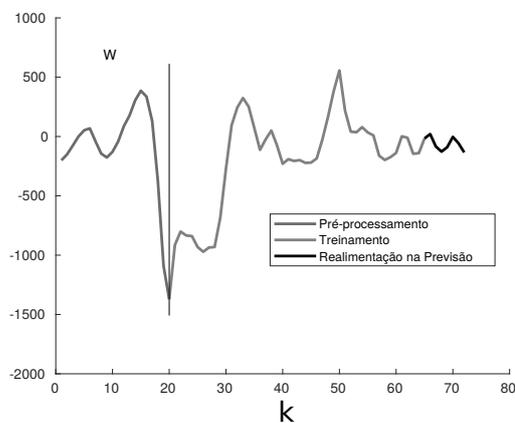
Fonte: Autor.

C.2.5 Série Temporal N912

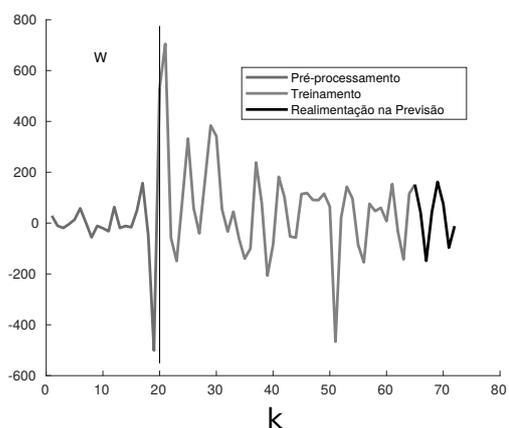
Figura 67 – Componentes Não-Observáveis da Série Temporal N912 nas Etapas de Pré-Processamento, Treinamento e Previsão Aplicando o Caso 2 do Algoritmo Genético. **(Parte 1)**.



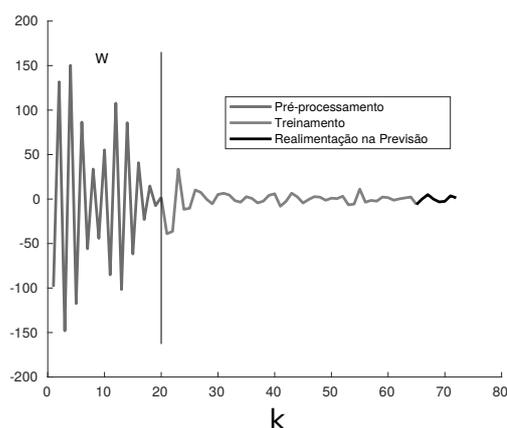
(a) Componente 1.



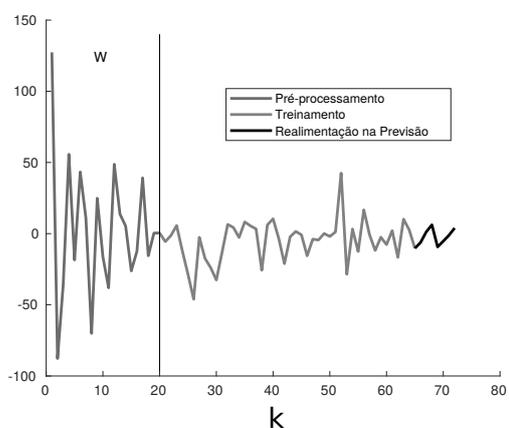
(b) Componente 2.



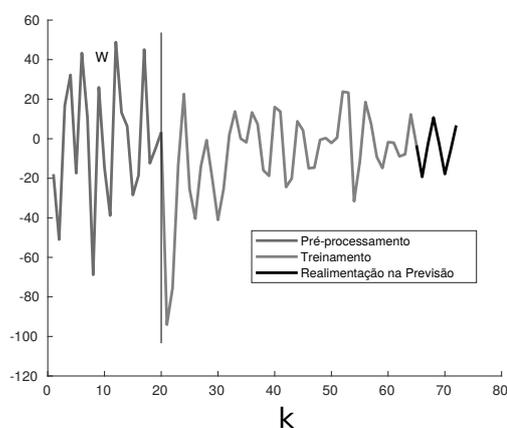
(c) Componente 3.



(d) Componente 4.



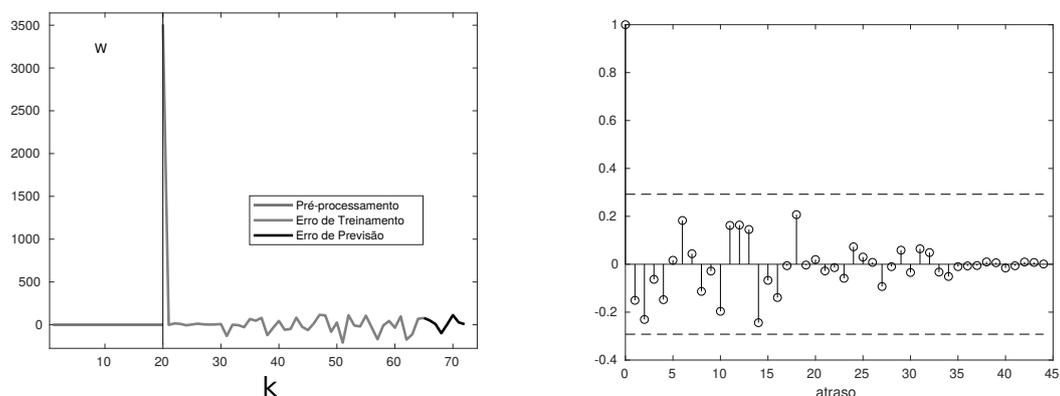
(e) Componente 5.



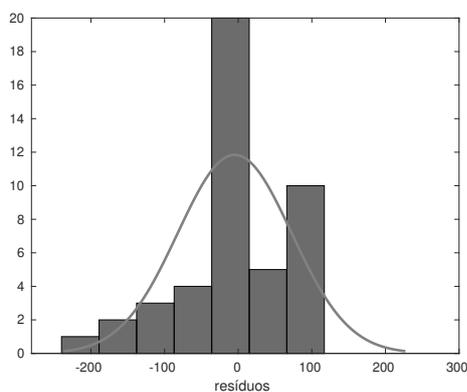
(f) Componente 6.

Fonte: Autor.

Figura 68 – Análise dos Erros de Treinamento e Previsão para a Série N912 Aplicando o Caso 2 do Algoritmo Genético.



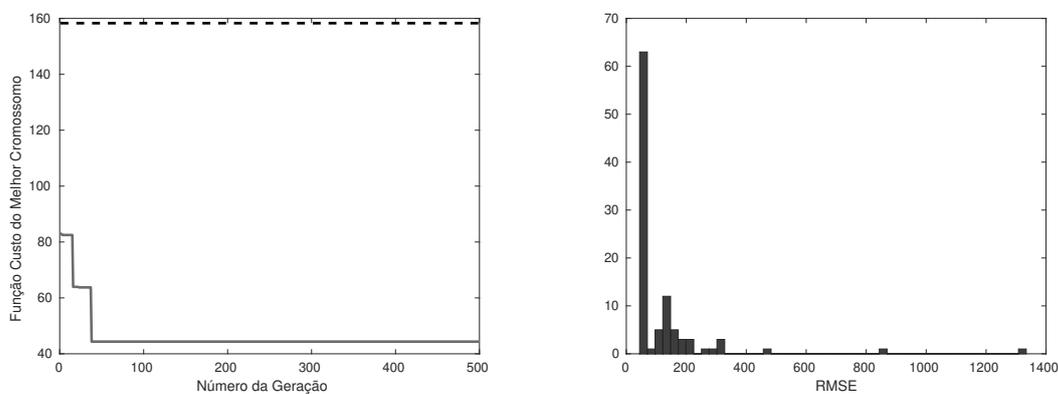
(a) Erros em Cada Etapa do PRSSA+ENFN. (b) Função de Autocorelação Amostral dos Resíduos de Treinamento.



(c) Histograma dos Resíduos de Treinamento.

Fonte: Autor.

Figura 69 – Informações do Melhor Cromossomo e da População Final do Algoritmo Genético no Caso 2 para a Série Temporal N912.

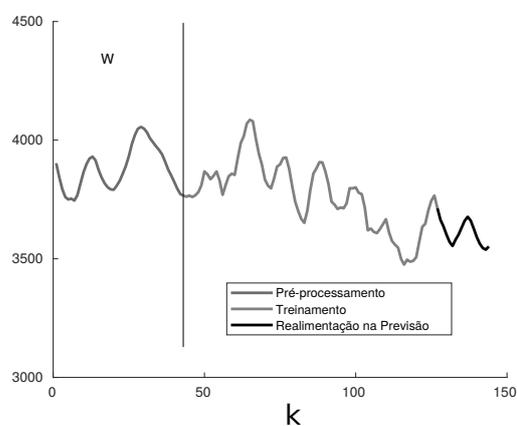


(a) Evolução do RMSE do Melhor Cromossomo Durante as Gerações. (b) Histograma das Medidas RMSE dos Cromossomos da Geração 500.

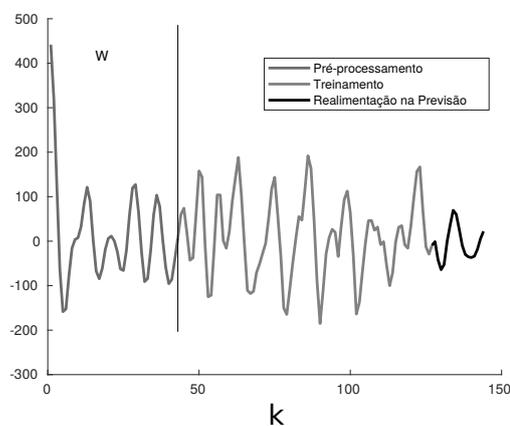
Fonte: Autor.

C.2.6 Série Temporal N2076

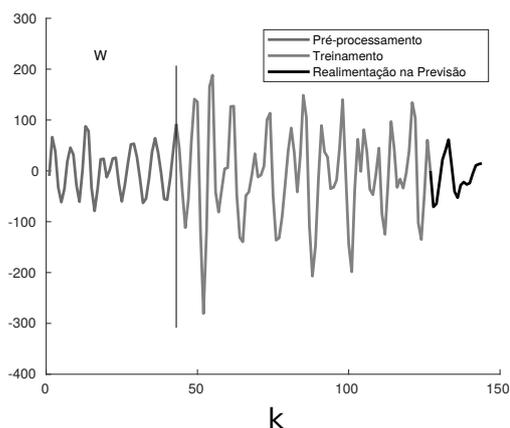
Figura 70 – Componentes Não-Observáveis da Série Temporal N2076 nas Etapas de Pré-Processamento, Treinamento e Previsão Aplicando o Caso 2 do Algoritmo Genético.



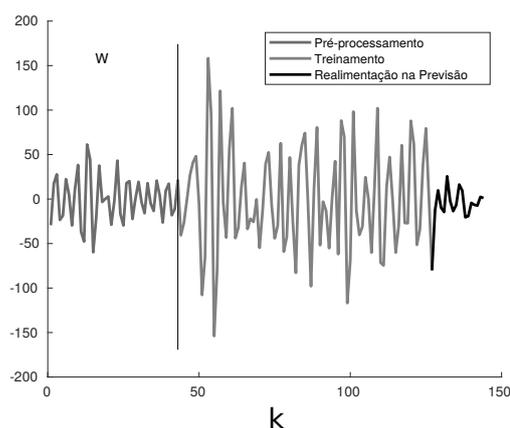
(a) Componente 1.



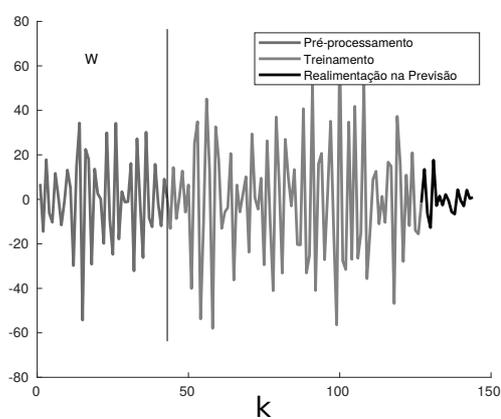
(b) Componente 2.



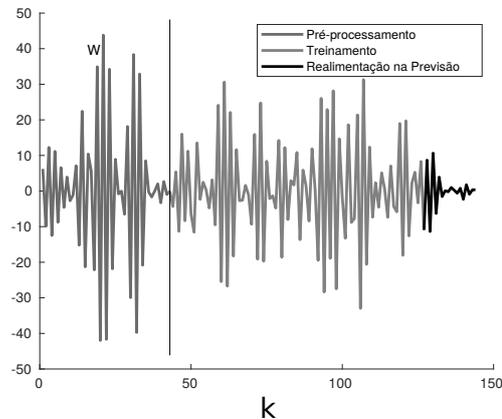
(c) Componente 3.



(d) Componente 4.



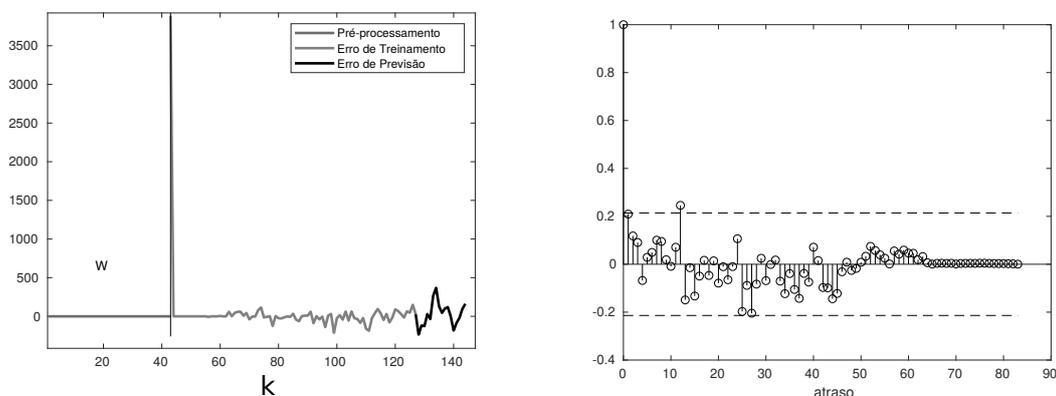
(e) Componente 5.



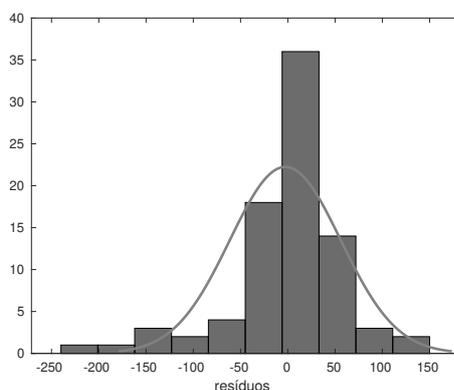
(f) Componente 6.

Fonte: Autor.

Figura 71 – Análise dos Erros de Treinamento e Previsão para a Série N2076 Aplicando o Caso 2 do Algoritmo Genético.



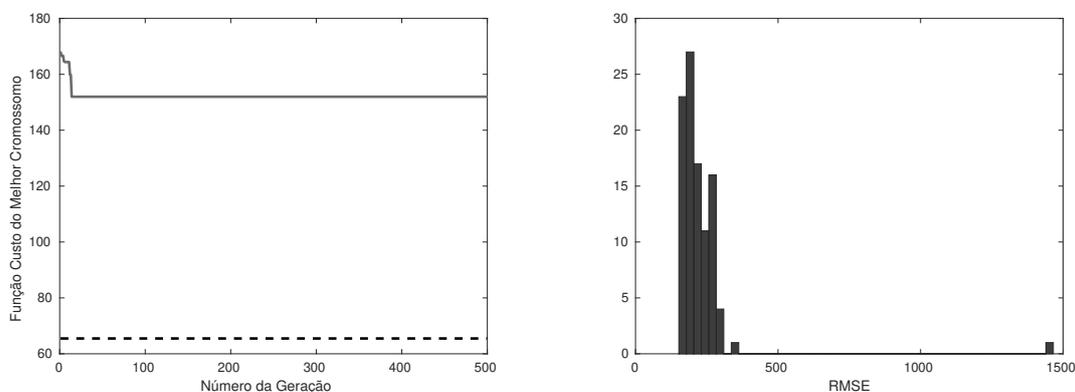
(a) Erros em Cada Etapa do PRSSA+ENFN. (b) Função de Autocorelação Amostral dos Resíduos de Treinamento.



(c) Histograma dos Resíduos de Treinamento.

Fonte: Autor.

Figura 72 – Informações do Melhor Cromossomo e da População Final do Algoritmo Genético no Caso 2 para a Série Temporal N2076.

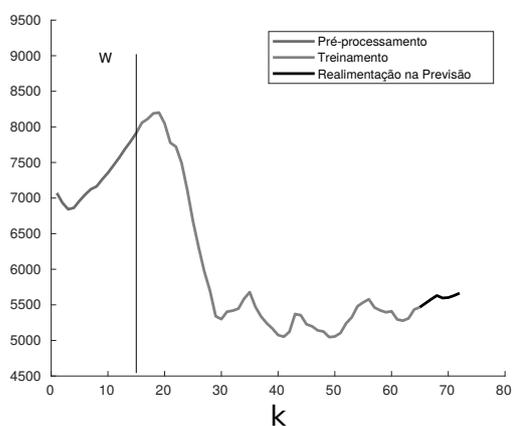


(a) Evolução do RMSE do Melhor Cromossomo Durante as Gerações. (b) Histograma das Medidas RMSE dos Cromossomos da Geração 500.

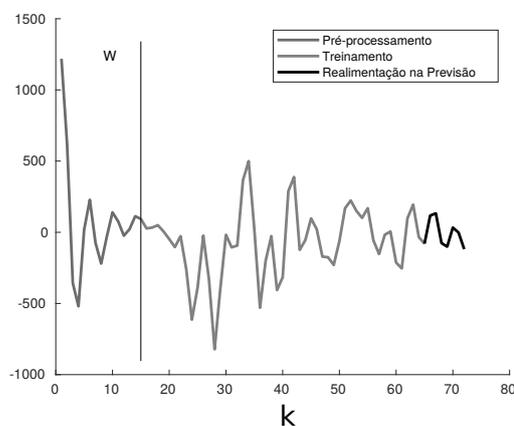
Fonte: Autor.

C.2.7 Série Temporal N897

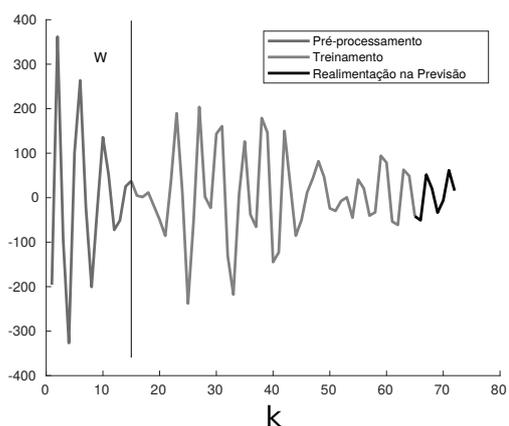
Figura 73 – Componentes Não-Observáveis da Série Temporal N897 nas Etapas de Pré-Processamento, Treinamento e Previsão Aplicando o Caso 2 do Algoritmo Genético (**Parte 1**).



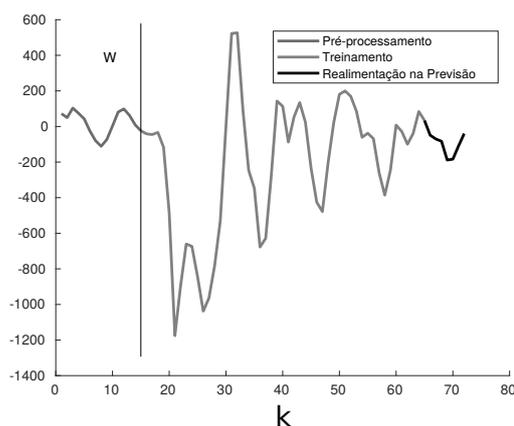
(a) Componente 1.



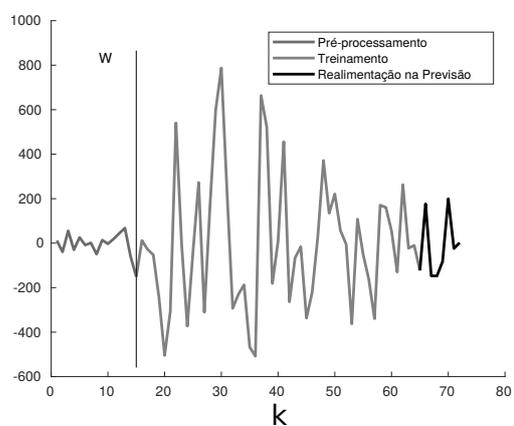
(b) Componente 2.



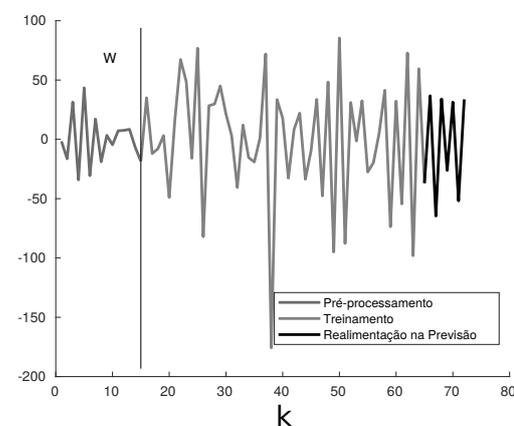
(c) Componente 3.



(d) Componente 4.



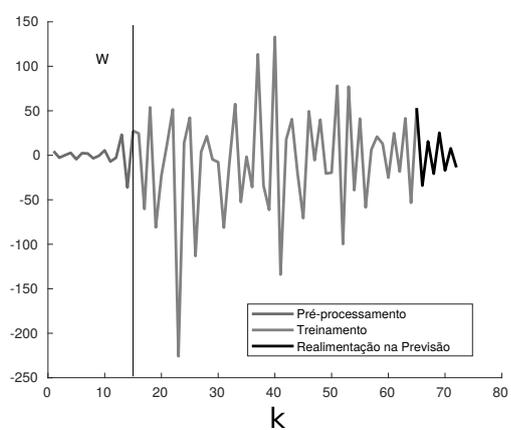
(e) Componente 5.



(f) Componente 6.

Fonte: Autor.

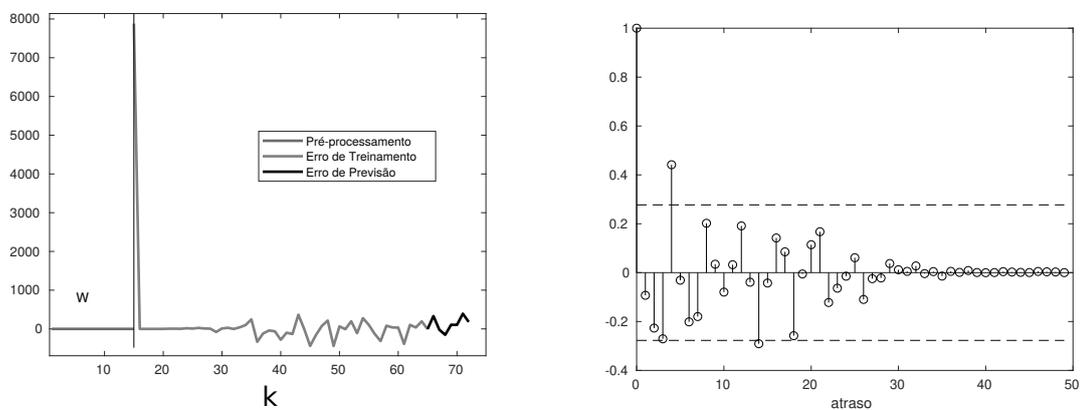
Figura 74 – Componentes Não-Observáveis da Série Temporal N897 nas Etapas de Pré-Processamento, Treinamento e Previsão Aplicando o Caso 2 do Algoritmo Genético (**Parte 2**).



(a) Componente 7.

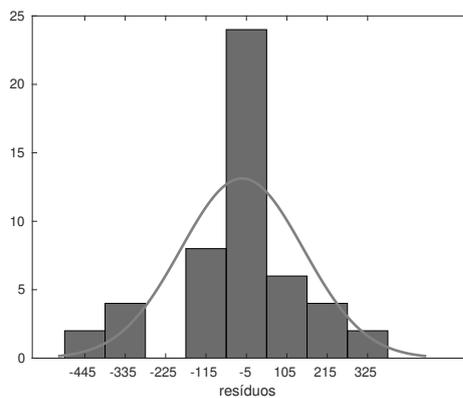
Fonte: Autor.

Figura 75 – Análise dos Erros de Treinamento e Previsão para a Série N897 Aplicando o Caso 2 do Algoritmo Genético.



(a) Erros em Cada Etapa do PRSSA+ENFN.

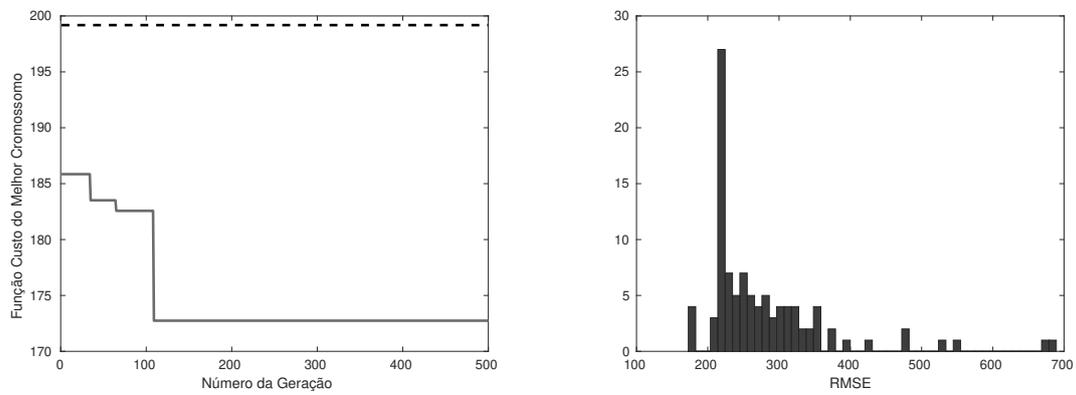
(b) Função de Autocorelação Amostral dos Resíduos de Treinamento.



(c) Histograma dos Resíduos de Treinamento.

Fonte: Autor.

Figura 76 – Informações do Melhor Cromossomo e da População Final do Algoritmo Genético no Caso 2 para a Série Temporal N897.



(a) Evolução do RMSE do Melhor Cromossomo Durante as Gerações. (b) Histograma das Medidas RMSE dos Cromossomos da Geração 500.

Fonte: Autor.

C.3 Tabelas com os Valores para os Parâmetros Críticos do Experimento 3 (Seção 6.5)

C.3.1 Tabelas para o Conjunto de Dados Pequim PM2.5

Tabela 32 – Parâmetros Críticos do PRSSA+ENFN para o Conjunto de Dados Pequim PM2.5.

Mapeamento	W	n_l	ϕ	η_0	RMSE
$k+1$	9476	[10; 6; 7; 10; 9; 3; 5; 8; 9]	0.38	0.08	23.68
$k+2$	9782	[9; 6; 8; 10; 7; 6; 4; 7; 5]	0.28	0.17	33.28
$k+3$	12036	[10; 7; 3; 8; 4; 2; 9; 10; 5]	0.14	0.16	34.07
$k+4$	10764	[10; 5; 3; 9; 6; 3; 10; 2; 6]	0.4	0.08	34.60
$k+5$	10959	[7; 9; 10; 9; 7; 2; 10; 7; 5]	0.4	0.11	34.95
$k+6$	12835	[6; 5; 10; 2; 4; 6; 2; 7; 4]	0.31	0.24	35.10

Fonte: Autor

C.3.2 Tabelas para o Conjunto de Dados Consumo de Energia Elétrica

Tabela 33 – Parâmetros Críticos do PRSSA+ENFN para o Conjunto de Dados Consumo de Energia Elétrica.

Mapeamento	W	n_l	ϕ	η_0	RMSE
$k+1$	463675	[2; 4; 3; 2; 6; 5; 6]	0.18	0.27	0.27
$k+2$	558892	[3; 5; 2; 8; 4; 3; 2]	0.11	0.14	0.38
$k+3$	534109	[2; 2; 6; 4; 4; 5; 4]	0.34	0.10	0.38
$k+4$	565372	[3; 4; 5; 5; 6; 4; 3]	0.22	0.26	0.48
$k+5$	619945	[2; 6; 4; 3; 5; 4; 2]	0.23	0.13	0.44
$k+6$	465552	[2; 2; 3; 2; 5; 3; 2]	0.26	0.10	0.40

Fonte: Autor

C.3.3 Tabelas para o Conjunto de Dados Qualidade do Ar

Tabela 34 – Parâmetros Críticos do PRSSA+ENFN para o Conjunto de Dados Qualidade do Ar.

Mapeamento	W	n_l	ϕ	η_0	RMSE
$k+1$	1949	[2; 4; 4; 9; 10; 2; 6; 5; 2; 9; 8; 3; 3; 9; 4]	0.46	0.13	45.26
$k+2$	2679	[3; 7; 4; 8; 4; 8; 5; 5; 2; 4; 3; 10; 4; 7; 2]	0.44	0.18	45.01
$k+3$	2170	[2; 7; 5; 8; 3; 5; 7; 3; 2; 9; 9; 10; 2; 3; 10]	0.32	0.20	44.64
$k+4$	2350	[2; 8; 7; 8; 2; 8; 2; 5; 8; 6; 8; 8; 3; 9; 4]	0.47	0.13	44.61
$k+5$	1954	[7; 2; 6; 7; 4; 8; 10; 4; 3; 2; 9; 8; 8; 6; 3]	0.36	0.12	44.61
$k+6$	1964	[7; 8; 8; 3; 5; 3; 9; 5; 2; 9; 5; 4; 4; 6; 4]	0.46	0.16	44.62

Fonte: Autor