

Universidade Federal de Juiz de Fora
Programa de Pós-Graduação em Modelagem Computacional

Samuel Belini Defilippo

**Previsão da demanda de energia elétrica por combinações de modelos
lineares e de inteligência computacional.**

Juiz de Fora

2017

Ficha catalográfica elaborada através do Modelo Latex do CDC da
UFJF com os dados fornecidos pelo(a) autor(a)

Defilippo, Samuel.

Previsão da demanda de energia elétrica por combinações de modelos lineares e de inteligência computacional. / Samuel Belini Defilippo. – 2017.
107 f. : il.

Orientador: D.Sc. Henrique Steinherz Hippert

Tese (Doutorado) – Universidade Federal de Juiz de Fora, . Programa de Pós-Graduação em Modelagem Computacional, 2017.

1. Previsão de carga elétrica. 2. Combinação de previsões. 3. Rede neural artificial. 4. Amortecimento exponencial. I. Hippert, Henrique Steinherz, orient. II. Título.

Samuel Belini Defilippo

Previsão da demanda de energia elétrica por combinações de modelos lineares e de inteligência computacional.

Tese apresentada ao Programa de Pós-Graduação em Modelagem Computacional da Universidade Federal de Juiz de Fora, na área de concentração em Modelagem Computacional, como requisito parcial para obtenção do título de Doutor em Modelagem Computacional.

Orientador: D.Sc. Henrique Steinherz Hippert

Juiz de Fora

2017

Samuel Belini Defilippo

Previsão da demanda de energia elétrica por combinações de modelos lineares e de inteligência computacional.

Tese apresentada ao Programa de Pós-Graduação em Modelagem Computacional da Universidade Federal de Juiz de Fora, na área de concentração em Modelagem Computacional, como requisito parcial para obtenção do título de Doutor em Modelagem Computacional.

Aprovada em: Juiz de Fora, 20 de setembro de 2017

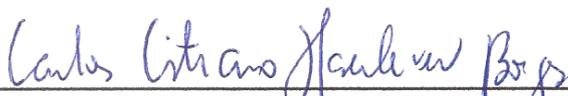
BANCA EXAMINADORA



Prof. Ph.D. Carlos Eduardo Pedreira
Universidade Federal do Rio de Janeiro



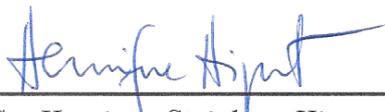
Prof. D.Sc. Francisco Augusto Lima Manfrini
Instituto Federal do Sudeste de Minas Gerais



Prof. D.Sc. Carlos Cristiano Hasenclever Borges
Universidade Federal de Juiz de Fora



Prof. Ph.D. Ronaldo Rocha Bastos
Universidade Federal de Juiz de Fora



Prof. D.Sc. Henrique Steinherz Hippert - Orientador
Universidade Federal de Juiz de Fora

AGRADECIMENTOS

Agradeço a todos que direta e indiretamente me ajudaram a concluir este trabalho. Especialmente ao meu Orientador que foi muito importante ao me ajudar e influenciar para o estudo aqui demonstrado.

RESUMO

Todo a produção, transmissão e distribuição de energia elétrica ocorre concomitantemente com o consumo da energia. Isso é necessário porque ainda não existe hoje uma maneira viável de se estocar energia em grandes quantidades. Dessa forma, a energia gerada precisa ser consumida quase que instantaneamente. Isso faz com que as previsões de demanda sejam fundamentais para uma boa gestão dos sistemas de energia.

Esse trabalho focaliza métodos de previsão de demanda a curto prazo, até um dia à frente. Nos métodos mais simples, as previsões são feitas por modelos lineares que utilizam dados históricos da demanda de energia. Contudo, modelos baseados em inteligência computacional têm sido estudados para este fim, por explorarem a relação não-linear entre a demanda de energia e as variáveis climáticas. Em geral, estes modelos conseguem melhores previsões do que os métodos lineares. Seus resultados, porém, são instáveis e sensíveis a erros de medição, gerando erros de previsão discrepantes, que podem ter graves consequências para o processo de produção.

Neste estudo, empregamos redes neurais artificiais e algoritmos genéticos para modelar dados históricos de carga e de clima, e combinamos estes modelos com métodos lineares tradicionais. O objetivo é conseguir previsões que não apenas sejam mais acuradas em termos médios, mas que também menos sensíveis aos erros de medição.

Palavras-chave: 1. Previsão de carga elétrica. 2. Combinação de previsões. 3. Rede neural artificial. 4. Amortecimento exponencial.

ABSTRACT

The production, transmission and distribution of electric energy occurs concomitantly with its consumption. This is necessary because there is yet no feasible way to store energy in large quantities. Therefore, the energy generated must be consumed almost instantaneously. This makes forecasting essential for the proper management of energy systems. This thesis focuses on short-term demand forecasting methods up to one day ahead.

In simpler methods, the forecasts are made by linear models, which use of historical data on energy demand. However, computer intelligence-based models have been studied for this end, exploring the nonlinear relationship between energy demand and climatic variables. In general, these models achieve better forecasts than linear methods. Their results, however, are unstable and sensitive to measurement errors, leading to outliers in forecasting errors, which can have serious consequences for the production process.

In this thesis, we use artificial neural networks and genetic algorithms for modelling historical load and climate data, and combined these models with traditional linear methods. The aim is to achieve forecasts that are not only more accurate in mean terms, but also less sensitive to measurement errors.

Key-words: 1. Short-term load forecasting. 2. Combination of forecasts. 3. Artificial neural network. 4. Exponential smoothing.

LISTA DE ILUSTRAÇÕES

Figura 1 – Fluxo de geração, transmissão e distribuição	13
Figura 2 – Exemplo da relação Carga x Temperatura, banco de dados <i>Washington</i>	14
Figura 3 – Cargas de uma quinzena de verão e outra de inverno – Rio	25
Figura 4 – Cargas médias diárias dos anos de 1996-97 – Rio	26
Figura 5 – Cargas de uma quinzena de verão e outra de inverno - <i>Washington</i>	26
Figura 6 – Cargas médias diárias dos anos de 1985-1986 – <i>Washington</i>	27
Figura 7 – Cargas em uma quinzena de verão e outra de inverno - Eslováquia	28
Figura 8 – Cargas médias diárias do ano de 1997-1998 – Eslováquia	28
Figura 9 – Gráfico de dispersão - Carga x Temperatura – série <i>Washington</i>	29
Figura 10 – <i>Modelo1</i> : RNA simples com 24 saídas	40
Figura 11 – Fluxograma - <i>Modelo1</i>	41
Figura 12 – <i>Modelo2</i> : 24 RNAs em paralelo	42
Figura 13 – Fluxograma - <i>Modelo2</i>	43
Figura 14 – Fluxograma - <i>Mod_Comb</i>	45
Figura 15 – Comparação dos MAPEs de todos os resultados - Rio I	48
Figura 16 – Comparação dos MAPEs de todos os resultados - Rio II	48
Figura 17 – Comparação dos MAPEs de todos os resultados - <i>Washington</i> I	49
Figura 18 – Comparação dos MAPEs de todos os resultados - <i>Washington</i> II	50
Figura 19 – Comparação dos MAPEs de todos os resultados - <i>Washington</i> III	50
Figura 20 – Comparação dos MAPEs de todos os resultados - Eslováquia I	51
Figura 21 – Comparação dos MAPEs de todos os resultados - Eslováquia II	51
Figura 22 – Comparação das combinações naïve e Byrd - Rio I	53
Figura 23 – Comparação das combinações naïve e Byrd - Eslováquia I	53
Figura 24 – Comparação das combinações naïve e Byrd - Eslováquia II	53
Figura 25 – Comparação das combinações naïve e Byrd - Rio II	54
Figura 26 – Comparação das combinações naïve e Byrd - <i>Washington</i> I	54
Figura 27 – Comparação das combinações naïve e Byrd - <i>Washington</i> II	55
Figura 28 – Comparação das combinações naïve e Byrd - <i>Washington</i> III	55
Figura 29 – Análise de sensibilidade - <i>Modelo2-3</i> e <i>Mod_Comb-3</i> - Rio I	57
Figura 30 – Análise de sensibilidade - <i>Modelo2-4</i> e <i>Mod_Comb-4</i> - Rio II	57
Figura 31 – Análise de sensibilidade - <i>Modelo2-4</i> e <i>Mod_Comb-4</i> - <i>Washington</i> I	58
Figura 32 – Análise de sensibilidade - <i>Modelo2-1</i> e <i>Mod_Comb-1</i> - <i>Washington</i> II	58
Figura 33 – Análise de sensibilidade - <i>Modelo2-2</i> e <i>Mod_Comb-2</i> - <i>Washington</i> III	59
Figura 34 – Análise de sensibilidade - <i>Modelo2-3</i> e <i>Mod_Comb-3</i> - Eslováquia I	61
Figura 35 – Análise de sensibilidade - <i>Modelo2-5</i> e <i>Mod_Comb-5</i> - Eslováquia II	61
Figura 36 – Sistemas adaptativos	78
Figura 37 – Modelo Adaline	80
Figura 38 – Topologia do Perceptron MLP	80

Figura 39 – MLP com base de funções	84
Figura 40 – Aproximação de funções com bases logísticas	84
Figura 41 – Fluxo de funcionamento de um AG simples	88
Figura 42 – Exemplo de codificação	89
Figura 43 – <i>Crossover</i> de um ponto	91
Figura 44 – <i>Crossover</i> de dois pontos	91
Figura 45 – <i>Crossover</i> uniforme	92
Figura 46 – Operador de mutação	92

LISTA DE TABELAS

Tabela 1 – Subséries – Rio	26
Tabela 2 – Subséries – Washington	27
Tabela 3 – Subséries – Eslováquia	29
Tabela 4 – Atributos de carga selecionados pelo AG - Rio	64
Tabela 5 – Atributos de carga selecionados pelo AG - Washington	65
Tabela 6 – Atributos de carga selecionados pelo AG - Eslováquia	66
Tabela 7 – Atributos de temperatura selecionados pelo AG - Rio	68
Tabela 8 – Atributos de temperatura selecionados pelo AG - Washington	69
Tabela 9 – Atributos de temperatura selecionados pelo AG - Eslováquia	70
Tabela 10 – Atributos para os dias de semana selecionados pelo AG - Rio	71
Tabela 11 – Atributos para os dias de semana selecionados pelo AG - Washington	71
Tabela 12 – Atributos para os dias de semana selecionados pelo AG - Eslováquia	71
Tabela 13 – Configurações da RNA selecionados pelo AG - Rio	73
Tabela 14 – Configurações da RNA selecionados pelo AG - Washington	74
Tabela 15 – Configurações da RNA selecionados pelo AG - Eslováquia	75

LISTA DE ABREVIATURAS E SIGLAS

UFJF	Universidade Federal de Juiz de Fora
RNA	Rede Neural Artificial
AG	Algoritmo Genético
ARIMA	Modelo auto-regressivo integrado de médias móveis (<i>autoregressive integrated moving average</i>)
HW	<i>Holt-Winters</i>
HWT	<i>Holt-Winters-Taylor</i>
MSE	Erro quadrático médio (<i>mean squared error</i>)
MAPE	Porcentagem do erro médio absoluto (<i>mean absolute percentage error</i>)
MLP	Perceptron multicamadas (<i>multilayer perceptron</i>)
RBF	Função de base radial (<i>radial basis function</i>)

SUMÁRIO

1	INTRODUÇÃO	12
1.1	CARACTERÍSTICAS DAS SÉRIES DE CARGA ELÉTRICA	12
1.2	REVISÃO BIBLIOGRÁFICA	14
1.3	REDES NEURAIS NA PREVISÃO DE CARGAS – UMA AVALIAÇÃO CRÍTICA	16
1.4	PROPOSTA DESTA TESE	19
2	MATERIAIS E MÉTODOS	21
2.1	PROCEDIMENTOS PARA VALIDAÇÃO DOS MODELOS	21
2.2	DADOS UTILIZADOS	22
2.2.1	<i>Partição das séries</i>	22
2.2.2	<i>Dados de temperatura</i>	24
2.2.3	<i>Dados de carga do Rio de Janeiro</i>	25
2.2.4	<i>Dados de carga de Washington (EUA)</i>	26
2.2.5	<i>Dados de carga da Eslováquia</i>	27
2.3	MODELOS LINEARES	29
2.3.1	<i>Previsor naïve</i>	30
2.3.2	<i>Método de amortecimento exponencial de Holt-Winters-Taylor (HWT)</i> .	31
2.4	REDE NEURAL ARTIFICIAL (RNA)	32
2.5	ALGORITMO GENÉTICO (AG)	34
2.5.1	<i>Parâmetros do AG</i>	36
2.5.2	<i>Função de aptidão</i>	37
2.6	COMBINAÇÃO DE MODELOS	38
2.7	FUNÇÃO DE ERRO	39
2.8	MODELOS DESENVOLVIDOS	39
2.8.1	<i>Modelo0</i>	39
2.8.2	<i>Modelo1</i>	40
2.8.3	<i>Modelo2</i>	42
2.8.4	<i>Mod_Comb</i>	44
2.9	ANÁLISE DE SENSIBILIDADE DO MODELO À TEMPERATURA	46
3	RESULTADOS	47
3.1	COMPARAÇÃO DOS MAPES	47
3.2	ANÁLISE DA COMBINAÇÃO	52
3.3	ANÁLISE DE SENSIBILIDADE À TEMPERATURA	56
3.4	ANÁLISE DA SELEÇÃO FEITA PELO AG	62
3.4.1	<i>Seleção de variáveis de entrada relacionadas às cargas passadas</i>	62

3.4.2	<i>Seleção de variáveis de entrada relacionadas às temperaturas</i>	67
3.4.3	<i>Seleção de variáveis indicadoras do dia da semana</i>	70
3.4.4	<i>Seleção de número de neurônios e de épocas de treinamento</i>	72
4	CONCLUSÃO	76
5	ANEXOS	78
5.1	REDES NEURAIS ARTIFICIAIS (RNA)	78
5.1.1	<i>ADALINE</i>	79
5.1.2	<i>Perceptron de múltiplas camadas (MLP)</i>	80
5.1.3	<i>Aproximação de funções</i>	81
5.1.4	<i>MLP e ajuste de função</i>	83
5.1.5	<i>Backpropagation</i>	85
5.1.6	<i>Critério de parada</i>	86
5.2	ALGORITMOS GENÉTICOS (AG)	87
5.2.1	<i>Processo de evolução</i>	88
5.2.2	<i>Função objetivo</i>	90
5.2.3	<i>Processo de seleção</i>	90
5.2.4	<i>Operadores genéticos</i>	90
5.2.5	<i>Operador de crossover</i>	91
5.2.6	<i>Operador de mutação</i>	92
5.3	AMORTECIMENTO EXPONENCIAL	92
5.3.1	<i>Método de Holt</i>	93
5.3.2	<i>Método de Holt-Winters (HW)</i>	94
5.4	MÉTODO DE OTIMIZAÇÃO DE BYRD	96
	REFERÊNCIAS	98

1 INTRODUÇÃO

Não é difícil perceber a importância da energia elétrica nos dias de hoje. Com ela o homem produz calor, frio, som, luz, força, e transmite informação, em várias formas. O desenvolvimento tecnológico contínuo tem gerado uma expansão constante no seu consumo; novas infra-estruturas têm sido criadas para levar a energia elétrica para novos consumidores, e investimentos tem sido feitos, em todo o mundo, na geração, transmissão e distribuição. No Brasil, durante o ano de 2012, próximo de 1 milhão e 800 mil consumidores tiveram suas residências adicionadas às redes de distribuição de energia elétrica [1]. A tendência é que este consumo aumente ainda mais; alguns estudos indicam que teremos um crescimento médio do consumo de 4,5% ao ano, de 2011 até 2021 [2]. Para dar conta dessa demanda, muito planejamento é necessário, em todas as etapas dos processos associados a energia elétrica.

Existem várias formas de gerar energia, seja hídrica, térmica, nuclear, entre outras. Todo o processo de produção, transmissão e distribuição ocorre concomitantemente com o seu consumo porque não existe, hoje, uma maneira viável de se estocar energia em grandes quantidades. Uma ilustração desse processo pode ser vista na Figura 1.

Devido a essa característica, a previsão do consumo é importante para todas as empresas que atuam no mercado de energia. Esta previsão permite otimizar o sistema nas duas pontas. Gerando apenas o necessário para o consumo no momento, reduz-se o desperdício da energia gerada. Por outro lado, se a geração não fica abaixo da demanda, e consegue acompanhar o consumo em tempo real, deixam de ocorrer prejuízos causados por apagões, quedas de tensões e perda da qualidade da energia (variações de tensão, etc.).

Este trabalho é focado na previsão de perfis de carga; isto é, na previsão das cargas observadas nas 24 horas de um dia, feita com base nas cargas observadas nos dias anteriores, ou em variáveis meteorológicas. Essa forma de previsão é conhecida na Engenharia como “previsão de carga a curto prazo” (*short-term load forecasting*) [3], e é fundamental para o planejamento das tarefas necessárias para a produção de energia, como a programação dos horários de ativação e desligamento das unidades geradoras, das vazões de água das hidrelétricas e da coordenação entre as várias usinas interconectadas. Além disso, estas previsões se tornaram a informação fundamental para a definição do preço da energia no mercado *spot*, depois que a privatização das concessionárias e a desregulamentação dos sistemas de energia criaram um mercado de energia competitivo.

1.1 CARACTERÍSTICAS DAS SÉRIES DE CARGA ELÉTRICA

A previsão do consumo de energia elétrica a curto prazo é um problema difícil, devido, em primeiro lugar, à complexidade das séries temporais envolvidas; em segundo

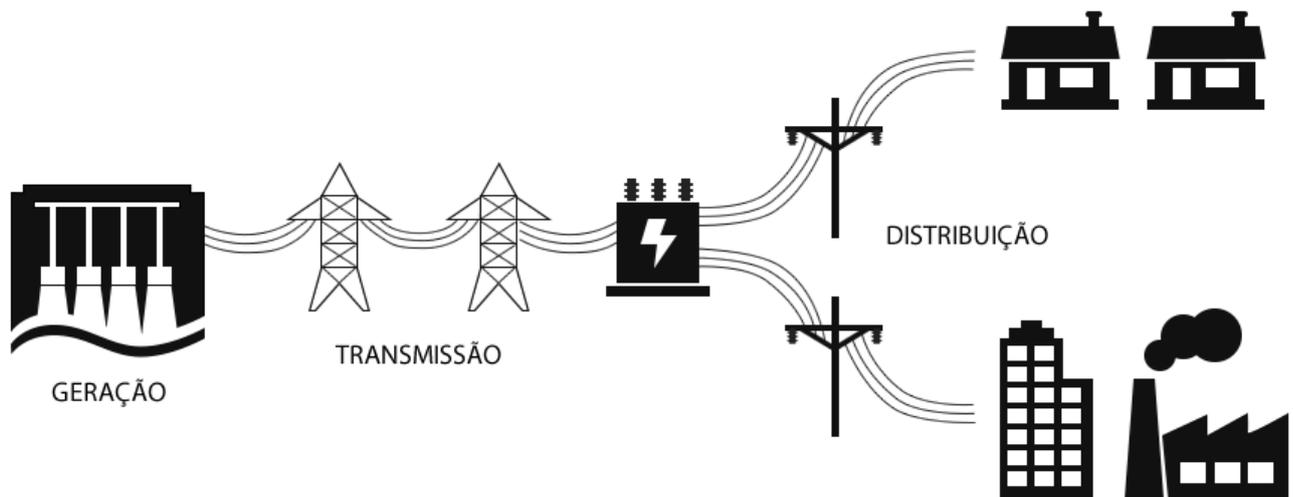


Figura 1 – Fluxo de geração, transmissão e distribuição

lugar, à não-linearidade das relações entre estas séries.

As séries de consumo podem ter três padrões sazonais sobrepostos. O “perfil” diário de consumo (isto é, a curva que mostra a variação do consumo a cada hora de um dia) é bastante similar nos dias da semana, entre segunda-feira e sexta-feira; há, portanto, uma sazonalidade diária evidente. O perfil dos sábados e domingos, porém, têm pouca semelhança com os dos dias da semana, e se aproximam mais dos perfis dos sábados e domingos das semanas anteriores; há aí, portanto uma sazonalidade semanal. Além disso, não apenas o nível da série, mas também os perfis diários mudam ao longo das estações (ver Capítulo 2); no Brasil, o consumo é maior nos meses de verão (devido ao uso de aparelhos de ar-condicionado) do que no resto do ano, o que caracteriza uma sazonalidade anual. Por sua vez, as séries temporais da temperatura horária (que geralmente é a variável explicativa mais importante para o consumo) têm também duas sazonalidades sobrepostas (diária e anual).

O que torna o problema ainda mais complexo, além destas sazonalidades, é o fato de a relação entre o consumo e a temperatura variar de uma localidade para outra, mas sendo em geral claramente não-linear. O diagrama de dispersão na Figura 2 exemplifica esta relação, com base nos dados de *Washington*, um dos bancos de dados usados nesta tese (ver Seção 2.2.4).

A complexidade das séries de cargas, portanto, torna sua previsão um problema bastante difícil, e sugere que métodos de inteligência computacional altamente não-lineares, como as RNAs, talvez sejam ferramentas adequadas para resolvê-lo. Faremos a seguir uma breve revisão da literatura existente sobre a previsão de cargas a curto prazo, e dos modelos já propostos para esta tarefa.

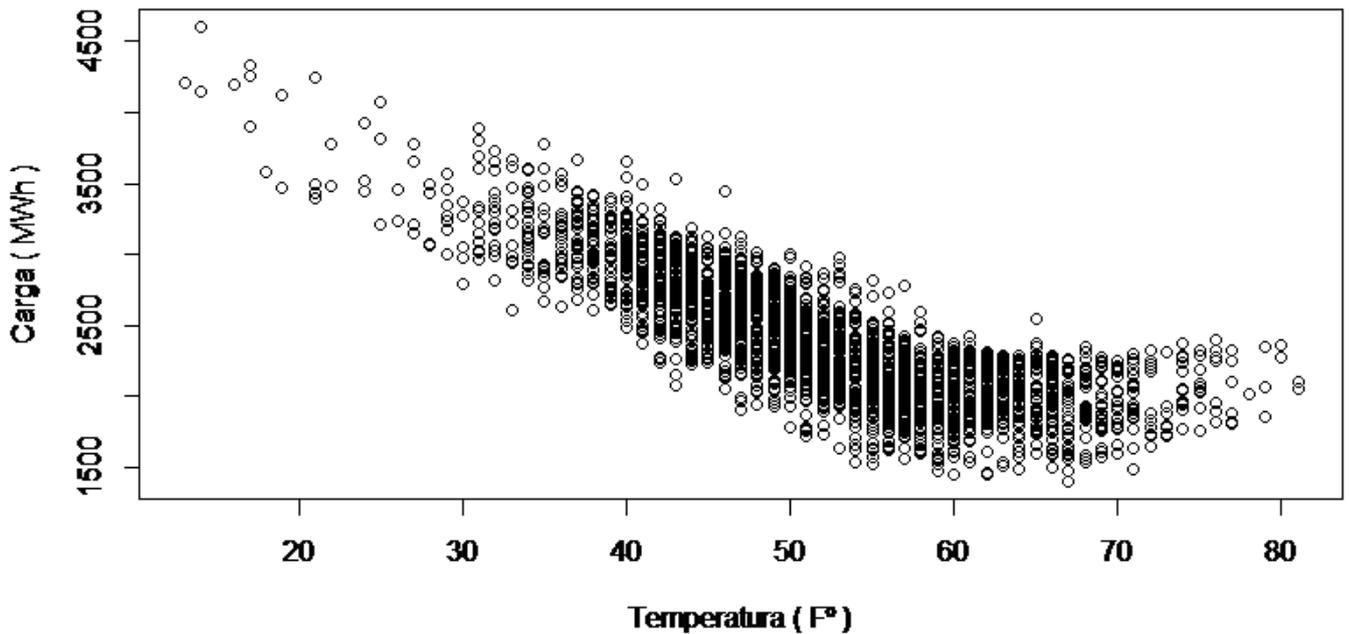


Figura 2 – Exemplo da relação Carga x Temperatura, banco de dados *Washington*

1.2 REVISÃO BIBLIOGRÁFICA

Na década de 1980, as privatizações no mercado de energia ocorridas em vários países do mundo deram à previsão de consumo uma grande importância econômica. A partir daí, praticamente todos os métodos então existentes foram experimentados neste problema de previsão; por exemplo, modelos ARIMA [4, 5, 6]; modelos de função de transferência de Box e Jenkins [4]; Modelos em espaço de estados usando o filtro de Kalman [7, 8, 9, 10, 11]; sistemas especialistas [12, 13]; decomposição em valores singulares [14, 15]. Os modelos vistos com maior frequência, porém são os de regressão linear [16, 17, 18, 19]. Estes modelos ainda são muito usados, como veremos mais abaixo.

A partir dos anos 1990, a maior parte o esforço de pesquisa foi focalizado no desenvolvimento de métodos baseados nas técnicas de inteligência computacional que começavam então a ser desenvolvidas. Uma ampla gama de métodos já foi testada, como a lógica *fuzzy* [20, 21, 22], máquinas de vetor suporte [23, 24, 25, 26], *wavelets* [27, 28]. Os modelos mais comumente utilizados, contudo, provavelmente são as redes neurais artificiais (RNAs). Estas redes, na sua arquitetura mais usual (percéptrons de múltiplas camadas; ver Anexo 5.1), funcionam como aproximadoras universais de qualquer função contínua; isto é, podem construir modelos capazes de se aproximarem, dentro de uma margem de erro especificada, qualquer função dada. Por isso, e por sua capacidade de lidar com dados multivariados (necessários para a previsão de perfis de carga), modelando as relações não-lineares entre entradas e saídas, RNAs parecem ser a ferramenta ideal para a previsão de carga a curto prazo. (O uso de RNAs na previsão de cargas será discutido

com mais detalhes na Seção 2.4).

Vários artigos já foram publicados revisando aplicações de técnicas de inteligência computacional à previsão de cargas. Por exemplo, a revisão em [29] analisa diversos artigos propondo estas técnicas, com ênfase nos sistemas especialistas, algoritmos genéticos e redes neurais. Estes mesmos métodos, e também métodos baseados em modelos lineares (regressão, amortecimento exponencial, ARMAX), são analisados em [30]. A revisão em [31] analisa sistemas de previsão de carga baseados exclusivamente em redes neurais; a revisão em [32], os baseados em redes neurais, máquinas de vetor suporte e métodos híbridos. A revisão em [33] discute sistemas aplicados no caso particular de previsões em *smart grids* e em edificações. A revisão em [34] analisa sistemas desenvolvidos para a previsão probabilística de carga (isto é, a previsão feita por meio de intervalos de confiança).

Na última década, a ênfase tem sido colocada no desenvolvimento de métodos híbridos que combinam várias técnicas de inteligência computacional. As RNAs, por exemplo, já foram combinadas com *wavelets* [35, 36, 37, 38, 39], com lógica *fuzzy* [40, 41, 42, 43], e com teoria do caos [44, 45, 46]. RNAs de diversos tipos (MLPs, RBF, redes recorrentes) e máquinas de vetor suporte também têm sido combinadas com diversos algoritmos de inspiração biológica, geralmente com o objetivo de otimizar as matrizes de pesos. Exemplos destes algoritmos são os de colônias de formigas [25], de colônia de abelhas [47], de morcegos [48], e os algoritmos *bacterial foraging* [49], *firefly* [50], *cuckoo search* [51], e *particle swarm optimization* - PSO [52, 53, 54].

Desta classe de algoritmos, os mais comumente empregados têm sido os *algoritmos genéticos* (AGs) ou de *computação evolucionista*, usados como ferramentas para treinamento de RNAs, particularmente quando estas RNAs são de tipos pouco usuais, diferentes dos percéptrons convencionais, como em [55, 56, 57, 58]. Estes algoritmos também podem ser usados para determinar a arquitetura da RNA (por exemplo, o número de neurônios ocultos), ou as variáveis de entrada, mas isto é pouco comum. AG foi usado em [59] e [55] para otimizar estruturas de uma rede *neuro-fuzzy*; em [60] para otimizar uma rede de Elman. Uma combinação de AG com algoritmo de colônia de formigas foi usada para selecionar variáveis de entrada, e PSO para escolher o número de neurônios em [61]. Nesta tese, usaremos AGs para apoiar a especificação do número de neurônios na camada oculta, do número de épocas de treinamento, e a seleção das variáveis de entrada (ver Seção 2.5)

Esta ênfase nos métodos de inteligência computacional não quer dizer que os modelos lineares ou estatísticos tenham se tornado obsoletos. Pelo contrário, numa competição de previsão organizada recentemente, a *Global Energy Forecasting Competition (GEF-Com2012)*, seis das dez equipes mais bem classificadas (inclusive a equipe vencedora [62]) usaram métodos baseados em alguma forma de regressão linear, isoladamente ou em combinação com outras técnicas de inteligência computacional [63]. Modelos lineares têm

como vantagens a simplicidade de uso, a disponibilidade de um arsenal de ferramentas estatísticas que ajudam na modelagem (testes para a seleção das variáveis de entrada no modelo, para a verificação da significância dos parâmetros, etc.), e uma relativa robustez em relação a valores discrepantes. Por outro lado, oferecem poucos recursos para a modelagem de relações não-lineares. Por isso, muitos pesquisadores têm estudado maneiras de combinar as vantagens dos dois tipos de métodos – os estatísticos lineares e os de inteligência computacional. Por exemplo, já foram feitas combinações de modelos ARMAX com lógica *fuzzy* [64], ARMAX com algoritmos genéticos e PSO [65]. Modelos de regressão linear com lógica *fuzzy* [66, 67, 68], modelos em espaço de estados com redes neurais [69], modelos ARIMA com redes neurais [70, 71, 72, 73, 74, 75], modelos SARIMA com regressão de vetor suporte [76].

O método de amortecimento exponencial de Holt-Winters, tradicionalmente usado para previsão de séries sazonais, foi revisto por Taylor [77], que o adaptou para uso em séries com vários padrões sazonais superpostos; variantes deste método Holt-Winters-Taylor (HWT) são mostradas em Taylor [78, 79, 80, 81, 82, 83, 84]. Modelos baseados neste método, contudo, têm sido pouco explorados por outros autores. *Wavelets* foram usados em [85] para decompor a série de cargas em duas subséries; a seguir, as previsões destas subséries foram feitas por meio de HWT e da técnica estatística de *weighted-nearest-neighbours*. A dissertação [86] fez experimentos com várias combinações lineares entre o HWT e outros métodos univariados (SARIMA, médias móveis, amortecimento exponencial), mas concluiu que nenhuma destas combinações levava a melhores resultados do que os obtidos pelo HWT usado isoladamente. Na literatura internacional, o único artigo encontrado que propõe um sistema de previsão combinando o método de HWT com técnicas de inteligência computacional é [87]. Nele, as previsões são feitas inicialmente pela RNA, e estas previsões são em seguida amortecidas por meio do HWT. A dissertação [88] adotou o caminho inverso: fez a previsão inicial por HWT, e em seguida modelou os erros de previsão obtidos como função da temperatura do ar, por meio de RNAs, obtendo resultados bastante promissores. Neste trabalho, combinamos linearmente as previsões feitas por RNAs (como função da temperatura) com as previsões univariadas feitas por HWT. O resultado disto foi um sistema híbrido de previsão que, embora nem sempre seja mais acurado do que os sistemas baseadas somente em RNAs, demonstrou ser muito mais estável, e menos sujeito aos grandes erros de previsão de carga que ocorrem quando as séries de medições de temperatura usadas na entrada das RNAs estão sujeitas a erros (ver Seção 2.8.4).

1.3 REDES NEURAS NA PREVISÃO DE CARGAS – UMA AVALIAÇÃO CRÍTICA

O número de artigos propondo métodos baseados em RNAs - que tem crescido de ano para ano desde a década de 1990 – parece que sugerir que as RNAs são os modelos

preferenciais para este tipo de previsão. Por exemplo, numa busca feita no portal de Periódicos da CAPES ([89], acesso dia 01/dez/2017) combinando a palavra-chave “*short-term load forecasting*” com “*neural networks*”, encontramos exatamente 100 referências, no período 1996-2015; em comparação, combinando com “*time series models*” encontramos 16 referências, e com “*linear regression*”, apenas nove.

No entanto, existem indícios que ainda levam vários autores ao ceticismo quanto às supostas vantagens do uso de RNAs. Um deles é a falta de uma validação adequada dos modelos propostos. A maioria destes artigos ignora as práticas de validação já bastante conhecidas em outras áreas de aplicação de modelos estatísticos ou de inteligência computacional; por exemplo, quase nunca discutem formas de re-amostragem, ou de análise de resíduos. Uma bem conhecida revisão da literatura [90] argumenta que a quase totalidade dos modelos propostos não são validados de forma convincente; uma revisão mas recente [34] reafirma este problema, tanto em RNAs como em outras áreas de inteligência artificial, e conclui que “*even now, it would be an exaggeration to consider artificial intelligence to be mature enough to offer real practical benefits for short-term load forecasting*”.

A validação feita pelos autores de um artigo, contudo, é apenas uma tentativa, realizada sob as condições artificiais de um teste de laboratório, de prever como um modelo irá desempenhar na prática. A validação definitiva de um modelo de previsão é aquela dada por seu uso bem sucedido em aplicações no mundo real. No entanto, existe um fosso separando o mundo acadêmico do mundo das aplicação prática. As empresas envolvidas no mercado de energia não divulgam os métodos que utilizam para a previsão, e os pesquisadores acadêmicos que trabalham na área não sabem o que tem funcionado na prática. Não há muitas evidências, por exemplo, de que as RNAs realmente estejam sendo usadas nas empresas. O único artigo encontrado na literatura relatando o uso no mundo real de um método baseado em RNAs por concessionárias de energia é o de [91], publicado a quase 20 anos; esta parece ser a única aplicação bem sucedida do uso de RNAs na previsão de cargas reportada em artigos acadêmicos [34].

Na falta de uma validação empírica, obtida em aplicações no mundo real, uma estimativa da validade de um modelo pode ser obtida através da análise de seu comportamento em competições, nas quais o desempenho preditivo é avaliado de acordo com critérios rigorosos e pré-determinados. Uma destas competições, realizada recentemente, foi a competição internacional de previsão de cargas (a GEFComp), cujos resultados foram publicados em 2014 na mais importante revista sobre métodos de previsão, o *International Journal of Forecasting* [63]. Estes resultados dão o que pensar; as RNAs foram usadas por várias equipes competidoras, mas seus resultados não foram tão destacados quando se poderia esperar. Três das cinco equipes melhor colocadas usaram regressão linear múltipla e outras técnicas estatísticas em seus sistemas de previsão; nenhuma delas usou RNAs ou outras técnicas de inteligência computacional. Equipes que usaram métodos baseados

em RNAs ficaram nos 8º, 9º e 10º lugares. Estes resultados não parecem congruentes com o que seria de se esperar, dada a literatura existentes; a julgar pelas conclusões da maioria dos artigos que tratam da previsão de cargas com RNAs, seria esperado que o desempenho das RNAs ultrapassasse claramente o dos modelos estatísticos tradicionais.

Devido a estas considerações, a real vantagem da utilização de métodos de inteligência computacional (particularmente RNAs) na previsão de cargas ainda não foi provada de forma conclusiva, apesar de toda a publicidade que tem sido dada a estes métodos na literatura.

Por que existe esta disparidade entre o desempenho das RNAs prometido nos artigos, e o desempenho observado na competição GEFComp? Uma resposta simples a esta pergunta poderia ser: a maioria destes artigos superestimou o desempenho dos modelos que propuseram, já que não validaram os modelos de forma correta; na verdade, estes modelos estavam longe de serem tão acurados como sugerido. Esta resposta, contudo, não cobre todos os casos; há com certeza uma minoria de artigos em que a validação foi feita criteriosamente.

Uma resposta mais adequada talvez possa ser baseada nas características das próprias RNAs, e da forma como são treinadas e testadas. RNAs são modelos não-lineares complicados, altamente parametrizados, capazes de modelar acuradamente relações não-lineares multivariadas entre entradas e saídas; para modelar séries de carga, que representam “processos complexos mas sistemáticos, com relativamente poucas mudanças estruturais [92]”, as RNAs podem parecer ideais. No entanto, existe uma condição: esta modelagem de processos complexos só pode ser feita com sucesso se considerarmos que os dados disponíveis são medidos sem erro. Se existe incerteza nestes dados, a acurácia da modelagem decresce, à medida que crescem os erros de medição.

Nos modelos propostos na literatura, quase sempre as cargas são previstas em função de cargas anteriores, de algumas variáveis *dummy*, e de algumas variáveis meteorológicas. Para prever a carga de um dia, as entradas são geralmente as cargas dos dias anteriores, e as temperaturas e umidades previstas para este dia. Na prática, estas previsões de temperatura e umidade geralmente não são disponíveis, e os pesquisadores usam em seu lugar os valores observados daquele dia. Este é o procedimento usual nos artigos, uma forma de contornar a falta das séries de previsões meteorológicas passadas; no entanto, ele tem a desvantagem de mascarar a complexidade do problema. É bem evidente que, numa aplicação prática, na entrada da RNA terão que ser usadas verdadeiras previsões meteorológicas, e estas com certeza serão afetadas por erros.

A dúvida é: até que ponto esta substituição de temperaturas previstas por temperaturas observadas pode levar os pesquisadores a superestimar a acurácia de seus modelos? Esta avaliação da sensibilidade das RNAs à incerteza das entradas, embora seja bem conhecida na literatura em outras áreas, não é feita na previsão de cargas. Conhecemos

apenas um artigo recente [76] que faz esta análise, usando séries de previsões de temperatura simuladas; contudo, a nosso ver, de forma inadequada, uma vez que o nível máximo de erro aleatório usado (1%) foi muito inferior ao que se poderia esperar em uma aplicação prática. Nesta tese, fizemos esta análise, simulando diferentes níveis de erro nas séries de temperatura na entrada (Seção 2.9). A conclusão geral foi a de que as RNAs são muito sensíveis ao erro; mesmo para níveis de erro relativamente baixos, bem menores do que os que são observados nas previsões meteorológicas divulgadas, o desempenho das RNAs é seriamente afetado. Isto pode explicar o desempenho relativamente insatisfatório das RNAs na GEFComp, onde os pesquisadores não tinham acesso às temperaturas observadas nas amostras de teste, e tinham que criar eles mesmos previsões para elas.

Em resumo, o excelente desempenho dos modelos baseados em RNAs na previsão de cargas a curto prazo, alegado pelas dezenas de artigos publicados a cada ano sobre o tema, não parece convincente. Em primeiro lugar, porque que a maioria destes modelos não parecem ter sido validados de forma adequada, se considerarmos os critérios recomendados em [90] para a validação de modelos de RNAs (estes critérios serão discutidos com mais detalhes na Seção 2.1). Em segundo lugar, porque estes modelos usam para treinamento e teste dados de entrada que não têm erro (valores observados, em vez de previsões), o que faz com que estes modelos pareçam ser mais acurados do que seriam numa aplicação no mundo real.

Para avaliar o efeito do erro de previsão de temperatura no desempenho das RNAs, fazemos neste trabalho uma análise da sensibilidade dos modelos, usando para isto séries de temperaturas com erros aleatórios adicionados por simulação (Seção 2.9). Esta análise mostrou que as RNAs são extremamente sensíveis, e perdem grande parte de sua acurácia quando a entrada é constituída por previsões e está, portanto, sujeita a erros consideráveis.

1.4 PROPOSTA DESTA TESE

Para evitar a perda de acurácia causada pela sensibilidade das RNAs à incerteza nas entradas, propomos nesta tese um modelo formado pela combinação de um método linear univariado (insensível à temperatura) com conjuntos de RNAs ; desta forma, são combinadas a capacidade das RNAs em fazer a modelagem acurada de relações não-lineares, com a insensibilidade de um modelo linear aos erros existentes nas previsões meteorológicas.

Na implementação feita nesta tese, usamos um conjunto de 24 RNAs em paralelo (cada uma responsável por prever a carga em uma hora determinada do dia); estas redes têm sua arquitetura e suas variáveis de entrada definidas por meio de rodadas de algoritmos genéticos. As previsões destas RNAs são combinadas com as de um método de amortecimento exponencial sazonal univariado (uma extensão do método de Holt-Winters). Os pesos que definem esta combinação são determinados durante o treinamento das redes;

se houver pouca incerteza nas previsões de temperatura, maior peso será dado às RNAs; se houver grande incerteza, maior peso será dado ao método linear, que não depende da temperatura.

Este modelo combinado de previsão teve seu desempenho comparado aos de vários *benchmarks*, quando testado em amostras de teste extraídas de três séries reais de cargas e de temperaturas, e mostrou boa acurácia, numa ampla gama de situações

O texto desta tese está estruturado da seguinte forma: no *Capítulo 2*, apresentamos os métodos e modelos a serem utilizados, os procedimentos adotados para sua validação, e as séries de cargas e temperaturas usadas nas simulações. No *Capítulo 3*, apresentamos os resultados obtidos pelos modelos propostos, analisamos a sensibilidade destes modelos ao erro nas temperaturas, e discutimos as contribuições trazidas pelos algoritmos genéticos à implementação das RNAs. Por fim, no *Capítulo 4*, fazemos um breve resumo das conclusões obtidas no trabalho, e sugerimos possíveis futuros desenvolvimentos nesta linha de pesquisa.

2 MATERIAIS E MÉTODOS

2.1 PROCEDIMENTOS PARA VALIDAÇÃO DOS MODELOS

Uma conhecida revisão da literatura [90] sugere que a validação de um modelo é adequada quando atende aos seguintes quesitos:

1. O desempenho do modelo proposto é comparado ao desempenho de padrões de referência adequados (*benchmarks*);
2. A avaliação destes desempenhos é feita em amostras de teste, separadas das amostras usadas para treinamento das RNAs;
3. Os tamanhos das amostras de treino e de teste são adequados.

O item (1) pode ser interpretado de duas maneiras. Em primeiro lugar, o método proposto pode ser comparado com um método naïve que forneça um ponto de referência, embora este seja um ponto de referência reconhecidamente baixo. (Um método de previsão *naïve* é aquele em que a previsão da variável um passo à frente é feita simplesmente repetindo-se alguma observação da série histórica daquela variável; ver Seção 2.3.1).

O método proposto deve ser claramente melhor que o método naïve, caso contrário não seria interessante sua utilização. O método naïve, além disso, tem outra importância: seu resultado pode sugerir quão complexo é um determinada problema de previsão. Se todos os artigos publicassem os resultados naïve obtidos nas séries que usaram, estes resultados poderiam servir para a comparação da dificuldade em se fazer a previsão de cada série (já que algumas séries são intrinsecamente mais difíceis de prever do que outras). Neste trabalho, usamos dois métodos naïves (descritos na Seção 2.3.1).

Em segundo lugar, o método proposto pode ser comparado a um método de referência bem aceito, que reconhecidamente consiga previsões razoáveis no problema; a comparação deve mostrar que o método proposto tem alguma vantagem sobre este método de referência. Este método deve ser de preferência um método estatístico, de implementação razoavelmente simples; comparações feitas com outros métodos de IA geralmente não são convincentes, porque estes métodos não são ainda considerados *standards*, e porque sua implementação em geral exige bastante experiência do usuário (por isso, este tipo de comparação foi criticado em [34]). Neste trabalho, utilizamos como referência a adaptação do método de amortecimento exponencial de *Holt-Winters*, feita por Taylor, que denominaremos *Holt-Winters-Taylor* [77]; este é um método linear de implementação relativamente fácil, mas que tem obtido resultados excelentes em vários testes empíricos. Este método é detalhado na Seção 2.3.2.

Para satisfazer aos quesitos (2) e (3), usamos amostras de teste claramente distintas das amostras de treino, para cada série, e todas de tamanho razoável; as amostras de treino cobriam em geral intervalos maiores que um ano, e as de teste foram sempre maiores que 10 semanas. Usamos para a partição das amostras a técnica de *rolling forecasts* [93]; detalhada na Seção 2.2.1.

2.2 DADOS UTILIZADOS

Este capítulo descreve as bases de dados utilizadas no trabalho. Na maioria dos estudos publicados sobre previsão de carga a curto prazo, os autores usam apenas uma base de dados (cargas e temperaturas) para ajuste e teste dos modelos que propõem. Contudo, decidimos usar três bases, bastante diferentes entre si, uma vez que nosso objetivo não era o de resolver um problema específico de previsão para um determinado local, e sim o de verificar se o método proposto - uma combinação de RNAs e métodos lineares, se mostraria robusto, e capaz de obter bons resultados quando aplicado sobre bases de dados com características dissimilares.

Foram utilizadas bases originárias de três localidades diferentes: Rio de Janeiro (Brasil), Washington (EUA) e Eslováquia (leste europeu). As três tem cargas mais ou menos de mesma ordem de grandeza (entre 0,5 e 3,5 GWh), e um comportamento sazonal anual similar, com cargas mais elevadas nos meses de final de um ano e início do outro. Este comportamento, porém, é explicado por razões opostas: nas séries do hemisfério norte (Washington e Eslováquia), estes meses correspondem ao inverno, e a carga é mais alta devido ao uso de aparelhos de aquecimento de ambientes; na série do Rio de Janeiro, estes meses correspondem ao verão, e a carga é mais alta devido ao uso de aparelhos de ar condicionado. Estas séries provêm de localidades com culturas e economias muito distintas, e isto é refletido nos padrões sazonais diários e semanais, que são claramente diferentes de uma série para outra, como será visto a seguir.

2.2.1 Partição das séries

As RNAs são modelos complexos e não-lineares, e a modelagem com elas apresenta algumas dificuldades de ordem prática. Em primeiro lugar, as RNAs têm em geral grande número de parâmetros que precisam ser estimados e, por isso, exigem grande quantidade de dados para seu treinamento e teste. Segundo, a estimação é feita por processos iterativos que partem de valores iniciais aleatórios e devem fazer a busca dos valores ótimos em superfícies bastante complexas, que exibem acentuada curvatura e dificuldades como mínimos locais e regiões planas; há sempre o risco de que, em uma rodada de treinamento, a busca termine em um mínimo local, ou uma região plana, levando a RNA a ter um desempenho sub-ótimo. Terceiro, as RNAs são modelos fortemente não-lineares, e correm o risco de terminarem super-ajustadas (*overfitted*) aos dados, o que tem que

ser evitado por meio de técnicas de parada precoce (*early stopping*) ou de regularização. Por fim, as RNAs são reconhecidamente algoritmos de aprendizado *instáveis*, no sentido de que pequenas mudanças na amostra de treino pode resultar em grandes diferenças no modelo final [94]. Por estas razões, o ideal é que RNAs sejam ajustadas sobre amostras grandes, e em seguida testadas em várias amostras de testes independentes. Se os dados são escassos, uma saída é usar técnicas de re-amostragem

Por estas razões, o ideal é que RNAs sejam ajustadas sobre amostras grandes, e em seguida testadas em várias amostras de testes independentes. Se os dados são escassos, uma saída é usar técnicas de re-amostragem.

Existem por exemplo as técnicas chamadas de *bootstrap*, que consistem em gerar novas amostras, simuladas, a partir de uma amostra original [95]. Estas novas amostras são obtidas por meio de sorteio aleatório de elementos da amostra original, com reposição, com base nas probabilidades observadas na distribuição empírica. Estes métodos, porém, não podem ser diretamente aplicados a séries temporais, uma vez que o sorteio aleatório destruiria a dependência serial entre os valores da série.

Outro grupo de técnicas bastante usadas são as de *k-fold validation* [96], que consistem em dividir os dados de treinamento em k subconjuntos disjuntos, de mesmo tamanho, e em seguida treinar o modelo k vezes, utilizando em cada rodada um dos subconjuntos para validação e os restantes dos dados para treinamento. Embora a dependência serial seja preservada (dentro de cada subconjunto), estes métodos também não podem ser aplicados diretamente em séries temporais, o que pode ser verificado por meio de um exemplo simples. Suponhamos que a série dada seja dividida em três subconjuntos sucessivos, A, B e C. Na primeira rodada, a rede é treinada em A e B, e testada em C, o que faz sentido. Na segunda rodada porém, a rede seria treinada em A e C, e testada em B; ou então, treinada em B e C e testada em A. Estaríamos, nestes casos, ajustando a rede em dados futuros, e usando-a para “prever” o passado. Isto poderia fazer sentido se a série fosse rigorosamente estacionária, porque então qualquer trecho da série teria o mesmo comportamento; as redes de carga porém são claramente não estacionárias, exibem tendência e sazonalidades múltiplas, e o comportamento da série varia de trecho para trecho.

A técnica mais adequada para aplicação em problemas de séries temporais é a que usa *rolling forecasts* [93]. A série é dividida em uma subsérie maior, para treino, contendo por exemplo 70% dos dados (subsérie A), e em várias subséries menores para teste (por exemplo, B, C e D, cada uma com 10% dos dados). Na primeira rodada, a rede é treinada em A e testada em B. Na segunda rodada, treinada na subsérie formada pela concatenação de A e B, e testada em C; por fim, treinada na subsérie formada pela concatenação de A, B e C, e testada em D. Nestes casos, a rede é testada três vezes, em amostras diferentes, e os dados de teste estão sempre no futuro, em relação ao dados de treino.

Neste trabalho, aplicamos esta técnica sobre as três bases de dados analisadas. Cada uma delas foi particionada em uma amostra de treino, e em duas ou três amostras de teste; da amostra de treino, por sua vez, foi retirada uma sub-amostra de validação cruzada, para controlar o interrupção do treinamento (exceto nos modelos que usavam regularização bayesiana, como discutido na Seção 2.4). Detalhes sobre estas partições são mostrados nas seções abaixo, que apresentam cada uma das bases de dados.

Em todas as séries de cargas elétricas, existem sempre alguns dias “especiais”, que apresentam perfis de cargas anormais. Geralmente, essas anomalias são causadas por feriados (quando o consumo é menor do que o usual) ou por eventos de grande impacto social, como comemorações esportivas, greves, etc. Na maioria dos artigos sobre previsão de cargas, a ênfase é na previsão de cargas nos dias normais, e os dias especiais são eliminados. As previsões para estes dias são difíceis, porque não existem dados disponíveis sobre estes dias, em quantidade suficiente para permitir a modelagem; há na literatura poucos artigos dedicados especialmente a este tipo de previsão [97]. Neste trabalho, iremos também eliminar os dias especiais, substituindo as cargas observadas nestes dias por valores imputados, calculados pela média dos valores observados no mesmo horário, nos dias de semana anteriores.

2.2.2 *Dados de temperatura*

As três bases de dados utilizadas incluem também dados de temperaturas observada nas regiões onde foram medidas as cargas elétricas. Nos bancos de dados do Rio de Janeiro e no de Washington, as séries de temperaturas são constituídas de observações horárias; na série da Eslováquia, apenas das médias diárias.

Alguns autores argumentam que não é necessário incluir a temperatura como uma das covariáveis de um modelo de previsão, uma vez que seu efeito sobre a carga já está embutido no histórico das cargas passadas [79, 98]. Isto faz sentido quando a série de temperaturas se comporta da forma usual, previsível; num dia de verão no Rio de Janeiro, por exemplo, a temperatura normalmente é alta, e a carga em consequência também provavelmente estará acima da média. Contudo, se num dia de verão a temperatura está muito abaixo de esperado, devido à ocorrência de uma frente fria, a carga será em consequência menor do que a usual. Mais importante do que os valores de temperatura observada, portanto, são os valores das diferenças da temperatura entre um dia e outro, pois são estes que fazem com que a carga seja aumentada ou diminuída, de um dia para o outro, e saia de seu padrão usual. Neste trabalho, usaremos como variáveis de entrada nos modelos as *diferenças* de temperatura entre um dia e outro, dadas por:

$$d_{d,h} = t_{d,h} - t_{d-1,h} \quad (2.1)$$

onde:

$d_{d,h}$ diferença de temperatura, no dia d , hora h

$t_{d,h}$ temperatura observada, no dia d , hora h

2.2.3 Dados de carga do Rio de Janeiro

Um dos materiais utilizados é a série temporal de cargas elétricas (em MWh) da cidade do Rio de Janeiro, fornecida por uma concessionária local. Esta série contém cargas horárias entre os dias 01 de janeiro de 1996 até 30 de dezembro de 1997, inclusive, num total de 17472 observações.

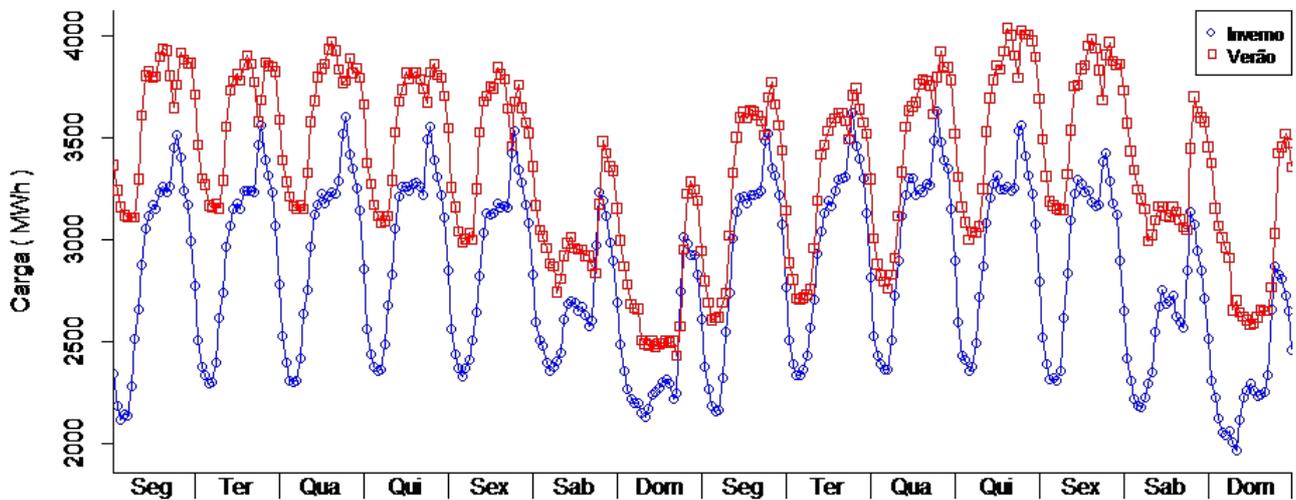


Figura 3 – Cargas de uma quinzena de verão e outra de inverno – Rio

As séries de carga elétrica na maioria das vezes apresentam padrões complexos, com flutuações cíclicas (sazonalidades) sobrepostas. O gráfico de linha na Figura 3 mostra a carga horária de quatro semanas, duas de inverno e duas de verão. A diferença que pode ser observada entre os perfis de carga dos dias da semana e os do fim de semana confirma a existência de padrões sazonais semanais. Os picos de carga em horários distintos nos dias de inverno e nos de verão, por outro lado, confirmam a existência de padrões sazonais diários. No inverno, existe um pico próximo das 19 h; no verão, um às 15h e outro às 23h. Estes padrões diários, aliás, são modificados nos fins de semana, o que torna o problema mais complexo. A diferença entre cargas no inverno e no verão é melhor destacada na Figura 4, que deixa claro que existe também um padrão sazonal anual. No verão a carga é maior, e esta diferença é normalmente atribuída ao uso intenso de aparelhos de ar condicionado, devido às altas temperaturas observadas.

Esta série original foi particionada em duas subséries, conforme explicitado na Tabela 1.

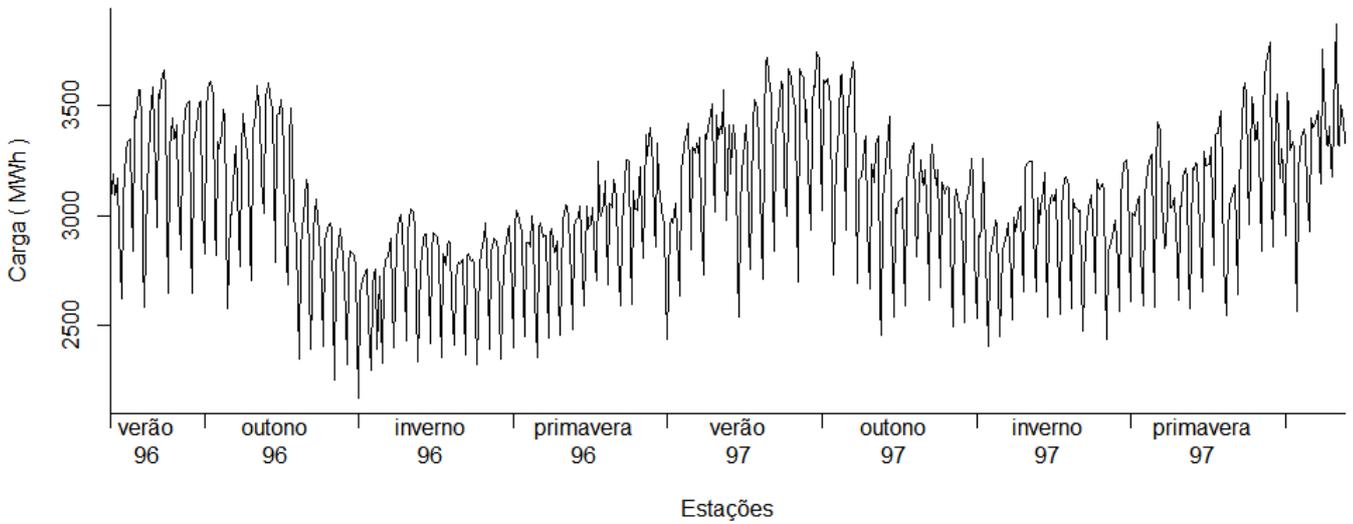


Figura 4 – Cargas médias diárias dos anos de 1996-97 – Rio

Tabela 1 – Subséries – Rio

Subsérie	Semanas		
	Treinamento	Validação	Teste
Rio I	14 a 53	54 a 63	64 a 83
Rio II	14 a 73	74 a 83	84 a 103

2.2.4 Dados de carga de Washington (EUA)

Com uma quantidade maior de dados, esta base de dados de carga e de temperatura, vindas de uma concessionária do estado de Washington (EUA), contém medições horárias feitas entre os dias 01 de janeiro de 1985 e 31 de março de 1991, num total de 68208 observações.

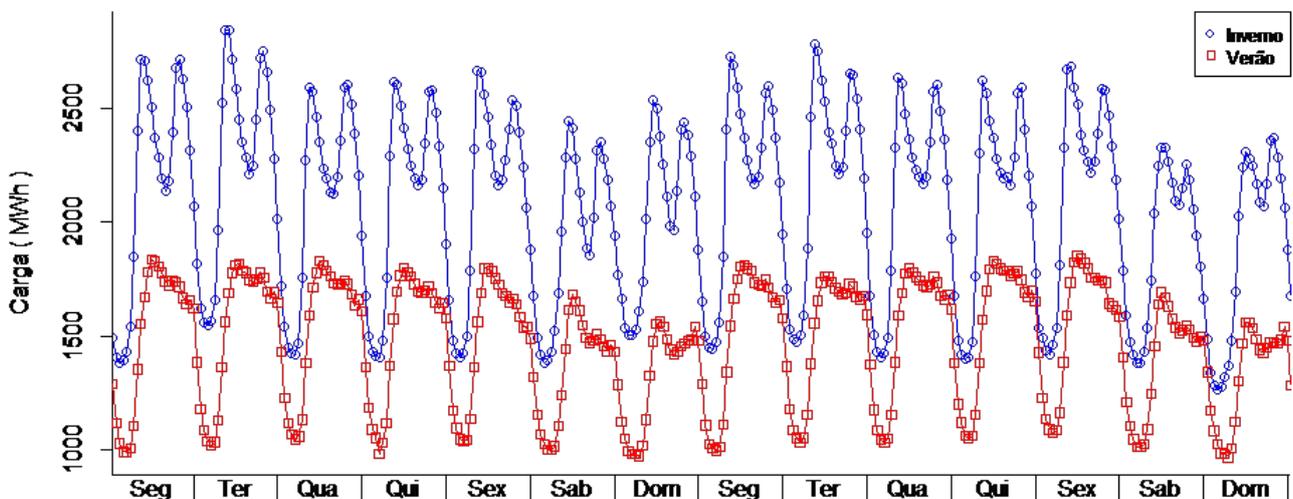


Figura 5 – Cargas de uma quinzena de verão e outra de inverno - Washington

Na Figura 5, é possível ver novamente os diferentes padrões sazonais que existem na série de carga. Existe uma diferença entre os perfis dos dias de semana e os dos fins de semana (embora menos evidente do que a encontrada nos dados do Rio de Janeiro), confirmando a existência de ciclos semanais. A localização similar dos picos de carga no decorrer dos dias de semana, por outro lado, confirma a existência do ciclo diário. A série apresenta também um padrão sazonal anual, como já era esperado: cargas menores nos meses situados no meio do ano (correspondentes ao verão, nos EUA), e maiores nos meses de fim de um ano, ou início do outro (inverno). Ele padrão pode ser claramente visto na Figura 6, que mostra as cargas de dois anos.

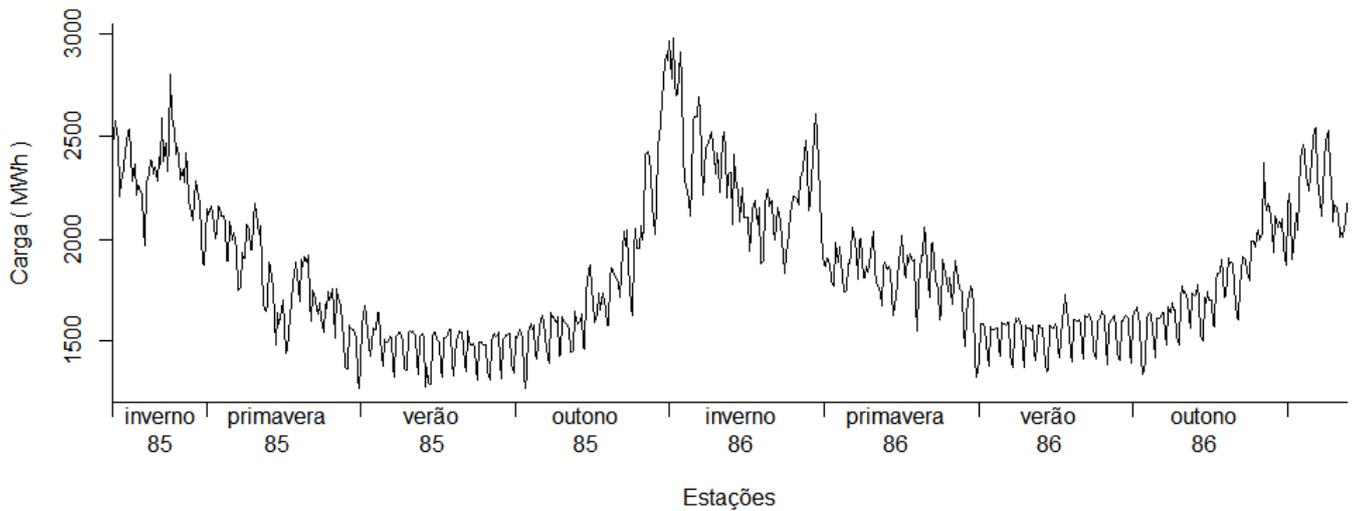


Figura 6 – Cargas médias diárias dos anos de 1985-1986 – Washington

Criamos três subséries a partir desta base de dados, conforme explicitado na Tabela 2.

Tabela 2 – Subséries – Washington

Subsérie	Semanas		
	Treinamento	Validação	Teste
Washington I	15 a 118	119 a 170	171 a 274
Washington II	15 a 170	171 a 222	223 a 326
Washington III	15 a 222	223 a 274	275 a 378

2.2.5 Dados de carga da Eslováquia

Esta base de dados contém medidas da carga elétrica a cada meia-hora, e de temperaturas médias diárias, feitas por um período de dois anos, pela *Eastern Slovakian Electricity Corporation*, uma concessionária na Eslováquia. Estes dados, obtidos online [99], foram usados para uma competição internacional de previsão, em 2001 [100].

A base original contém dados obtidos entre 01 de janeiro de 1997 a 31 de janeiro de 1999, totalizando 18264 horas. Neste trabalho, utilizamos apenas as cargas medidas nas

horas representadas por números inteiros (01:00, 02:00, ..., 24:00), e descartamos as cargas medidas nas meias-horas intermediárias (00:30, 01:30, ..., 23:30), a fim de simplificar o problema, e tornar seus resultados mais comparáveis com os obtidos nas séries horárias mencionadas nas seções anteriores, do Rio e de Washington.

No gráfico da Figura 7, vemos novamente que existe uma diferença entre o consumo nos dias de semana e o nos fins de semana, confirmando a existência de um padrão sazonal semanal. Vemos também a localização regular dos picos diários de carga, diferentes no verão e no inverno, confirmando a existência de um padrão sazonal diário. A série de cargas tem ainda um padrão sazonal anual evidente, como pode ser visto na Figura 8, que mostra as cargas de dois anos.

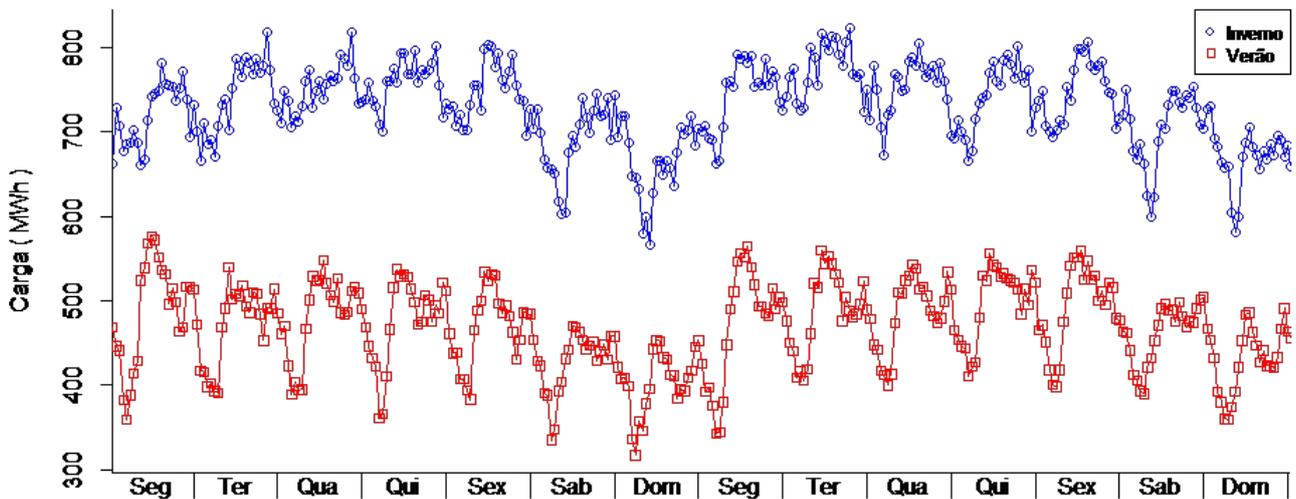


Figura 7 – Cargas em uma quinzena de verão e outra de inverno - Eslováquia

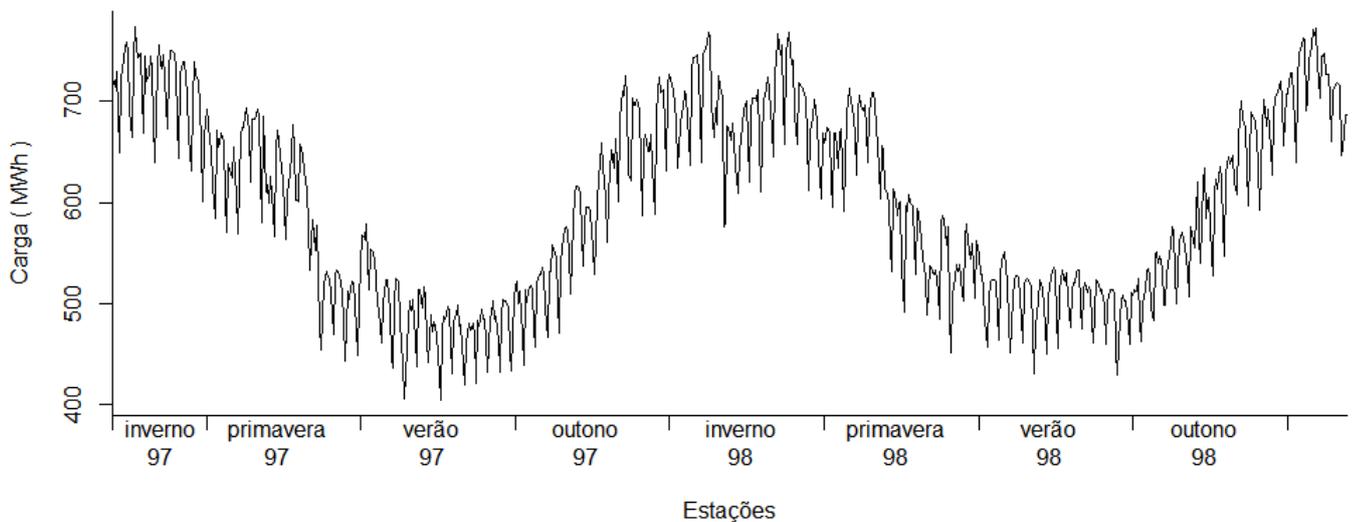


Figura 8 – Cargas médias diárias do ano de 1997-1998 – Eslováquia

A partir dos dados originais, criamos duas subséries, cada uma contendo 10 semanas para o intervalo de teste, como explicitado na Tabela 3.

Tabela 3 – Subséries – Eslováquia

Subsérie	Semanas		
	Treinamento	Validação	Teste
Eslováquia I	7 a 49	50 a 67	68 a 77
Eslováquia II	7 a 69	70 a 87	88 a 97

2.3 MODELOS LINEARES

Modelos não-lineares parecem ser as ferramentas naturais para problemas de previsão de cargas, dado que a relação entre a carga e as variáveis exógenas, especialmente a temperatura, tende a ser claramente não-linear, como mostra o exemplo da Figura 9.

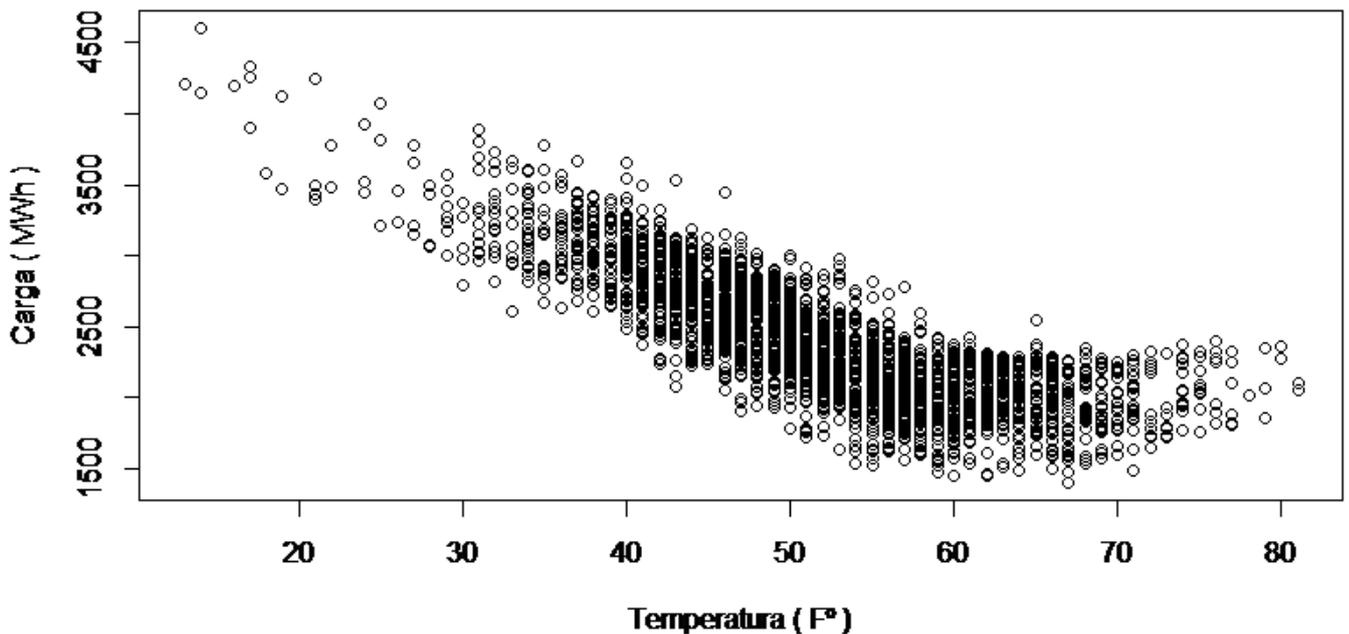


Figura 9 – Gráfico de dispersão - Carga x Temperatura – série Washington

As RNAs, portanto, parecem ser as candidatas ideais para este tipo de tarefa de previsão. Contudo, como veremos abaixo (Seção 2.4), os modelos baseados em RNAs tendem a ser altamente sensíveis aos erros nas variáveis de entrada. Para a previsão da carga um dia à frente, estes modelos geralmente usam as previsões das temperaturas para esse mesmo dia; como estas previsões são obviamente incertas, o desempenho destes modelos, na prática, fica muito aquém do que seria esperado em condições ideais.

Modelos lineares, em geral, são mais robustos do que os não-lineares, e tendem a ser menos sensíveis à incerteza nas suas entradas. Em contrapartida, talvez obtenham resultados menos acurados quando existem relações não-lineares entre as variáveis. Na

prática, nos problemas de previsão de carga a curto prazo, os modelos lineares (mais especificamente, os de regressão linear) têm sido usados com grande frequência em publicações recentes. Por exemplo, na competição internacional de previsão de carga organizada pelo *International Institute of Forecasters*, cujos resultados foram publicados em [63], cinco dos dez primeiros colocados usaram métodos baseados em regressão linear, inclusive os que obtiveram o primeiro lugar [101] e o segundo lugar [102].

Uma maneira de extrair o melhor dos dois tipos de modelos pode ser a de combinar, de alguma forma os seus resultados. Neste trabalho, iremos propor um método baseado numa combinação linear simples de dois modelos, um constituído de um conjunto de RNAs em paralelo, o outro de um método de amortecimento exponencial, o método de Holt-Winters-Taylor (descrito na Seção 2.3.2).

2.3.1 *Previsor naïve*

Para avaliar a complexidade de um problema de previsão, e para gerar um padrão básico (*benchmark*), em relação ao qual os demais modelos possam ser comparados, usaremos aqui dois métodos naïves (ingênuos). Um método de previsão *naïve* é aquele em que a previsão da carga L um passo à frente é feita simplesmente repetindo-se a observação mais recente da série histórica, isto é:

$$\hat{L}_t = L_{t-1} \quad (2.2)$$

onde \hat{L}_t é a previsão no tempo t e L_t o valor observado no tempo t .

Na previsão de perfis de carga, porém, este *benchmark* não pode ser usado desta forma, porque as previsões são feitas com horizontes variando de uma a 24 horas à frente; para fazer a previsão naïve da carga na hora $t = 2$ do dia seguinte, por exemplo, precisaríamos da carga na hora $t = 1$ desse dia, que é evidentemente desconhecida. Neste trabalho, utilizamos duas variações dos métodos naïve. A primeira (*Naïve24*) faz a previsão para uma hora t repetindo o valor observado no mesmo instante no dia anterior, $t - 24$, como na Equação (2.3).

$$\hat{L}_t = L_{t-24} \quad (2.3)$$

Este método consegue resultados que servem de padrão de comparação quando as previsões são feitas de um dia de semana para outro; por exemplo, quando usamos as observações de uma terça-feira para prever as cargas de uma quarta-feira. Ele dá resultados de baixa acurácia, contudo, quando dias de semana estão adjacentes a dias de fim de semana, que têm padrões muito diferentes; por exemplo, quanto tentamos prever as cargas de sábado a partir das de sexta-feira, ou as de segunda-feira a partir das de domingo.

O segundo método (*Naïve168*) faz a previsão repetindo o valor observado na mesma hora no dia de mesma denominação da semana anterior, $t - 168$, como na Equação (2.4).

$$\hat{L}_t = L_{t-168} \quad (2.4)$$

Neste caso, os padrões sazonais são respeitados – as previsões de um sábado são feitas a partir das observações do sábado da semana anterior; as de uma segunda-feira, a partir das da segunda-feira anterior. Estas previsões contudo são em geral piores do que as do método *Naïve24*, porque se baseiam em informação muito defasadas, de uma semana atrás.

2.3.2 Método de amortecimento exponencial de Holt-Winters-Taylor (HWT)

O método HWT se baseia no método de amortecimento exponencial de Holt-Winters, modificado de forma a tornar possível o trabalho com séries que apresentam vários padrões sazonais superpostos. (Mais informações sobre os métodos de amortecimento exponencial podem ser encontradas no Anexo 5.3).

O método HWT foi apresentado no artigo [77] e mais recentemente em outro artigo [80], onde o autor, compara com outros métodos capazes de lidar com séries contendo várias padrões sazonais simultâneos. Ele tem mostrado bons resultados em problemas de previsão de carga a curto prazo [103], pode ser facilmente utilizado na combinação com outros métodos, e também ser usado como um *benchmark* linear.

Neste trabalho utilizamos sua formulação com ciclos aditivos, como dada pelas Equações (2.5) até (2.8).

$$\hat{L}_{(t+k)} = l_t + d_{(t-s_1+k)} + w_{(t-s_2+k)} + \phi^k(L_t - (l_{(t-1)} + d_{(t-s_1)} + w_{(t-s_2)})) \quad (2.5)$$

$$l_t = \lambda(L_t - d_{(t-s_1)} - w_{(t-s_2)}) + (1 - \lambda)(l_{(t-1)}) \quad (2.6)$$

$$d_t = \delta(L_t - l_{(t-1)} - w_{(t-s_2)}) + (1 - \delta)d_{(t-s_1)} \quad (2.7)$$

$$w_t = \omega(L_t - l_{(t-1)} - d_{(t-s_1)}) + (1 - \omega)w_{(t-s_2)} \quad (2.8)$$

onde:

L : carga elétrica

l : nível

d : fatores sazonais diários

w : fatores sazonais semanais

λ : coeficiente do modelo auto-regressivo para o erro

δ, ω, ϕ : constantes que definem a ponderação nas equações de atualização do nível, da sazonalidade diária e da sazonalidade semanal, respectivamente

k : horizonte de previsão, que varia de 01 até 24

A Equação (2.5) calcula a previsão da carga L_t , dada pela soma do nível l_t com os dois fatores sazonais d_t e w_t , e com um termo de correção do erro de previsão; este termo é calculado por um modelo auto-regressivo aplicado sobre o último erro observado. A Equação (2.6) faz a atualização recursiva do nível l_t ; a Equação (2.7) faz a atualização do fator sazonal diário d_t ; a Equação (2.8) faz a atualização do fator sazonal semanal w_t . Note-se que esta formulação não inclui uma equação para cálculo da tendência da série, usual no método de Holt-Winters; esta equação foi descartada por considerarmos que o efeito do crescimento do nível da série, embora seja observável a longo prazo, tem pouco efeito sobre a previsão a curto prazo.

As constantes de amortecimento $\lambda, \delta, \omega, \phi$ foram otimizadas pelo método de Byrd no intervalo de treinamento de cada subsérie. As restrições de máximo e mínimo para os parâmetros foram de 1 e 0 respectivamente. (Mais informações sobre o método de Byrd podem ser encontradas no Anexo 5.4).

2.4 REDE NEURAL ARTIFICIAL (RNA)

A implementação de uma RNA requer várias decisões, a começar pela arquitetura da rede. Neste trabalho, usamos RNAs do tipo *perceptron*, com uma camada oculta e função de ativação sigmóide. Esse tipo de RNA é frequentemente utilizado em estudos de previsão de carga [90], devido à sua capacidade como aproximador universal de qualquer função derivável. (Informações mais detalhadas sobre o funcionamento de uma RNA estão no Anexo 5.1). O número de neurônios na camada oculta, e o conjunto de variáveis de entrada, foram definidos por meio de rodadas de um algoritmo genético (ver Seção 2.5).

Em seguida, é preciso decidir detalhes sobre o processo de estimação dos parâmetros da rede (isto é, do “treinamento” da RNA) – especialmente, que algoritmo usar, e quando parar este treinamento. Neste trabalho, fizemos testes preliminares utilizando dois algoritmos distintos, o de Levenberg-Marquardt e um de regularização bayesiana.

O algoritmo de Levenberg-Marquardt (LM) [104], que é uma adaptação do método de otimização de Newton, especificamente criada para a minimização de uma função de custo quadrática (como são as usadas pelas RNAs), fazendo uma aproximação da matriz

Hessiana. ([96] sect. 4.3.4; [104] p.290). Avaliações empíricas mostram que o LM converge muito mais rápido do que os métodos baseados em descida de gradiente; ele é, por isso, o método mais frequentemente usado em aplicações práticas, o que é corroborado pelo fato de ter sido escolhido como algoritmo de treinamento *default* no *toolbox* de redes neurais do Matlab, e em pacotes em outras linguagens.

Escolhido o algoritmo de treinamento, é preciso escolher o método que controlam a parada do treinamento. Se uma RNA de arquitetura suficientemente complexa for treinada indefinidamente, pode ocorrer que o erro de previsão seja diminuído até próximo de zero, dentro da amostra de treinamento. Isto não necessariamente significa, porém, que a previsão feita por esta RNA será mais acurada, quando testada numa amostra de teste; pode acontecer que a rede acabe super-ajustada (*overfitted*) à amostra de treino, e se revele com pouca capacidade de generalização em outras amostras. Uma das maneiras de evitar isto é reduzindo a complexidade da RNA (o que pode ser feito por meio de *regularização*, discutida mais abaixo); outra maneira é interrompendo o treinamento no ponto adequado, aquele onde RNA mostre melhor generalização. Uma técnica heurística para descobrir este ponto é a chamada *validação cruzada* [105]: ao fim de cada época de treinamento, o desempenho da RNA é testado tanto na amostra de treino quanto em uma amostra independente, a *amostra de validação*. Enquanto o desempenho da RNA melhorar nas duas amostras, o treinamento continua; quando o erro continua a diminuir na amostra de treino, mas passa a crescer na amostra de validação, o treinamento é interrompido, pois há sinais de que o modelo esteja se tornando super-ajustado. Usamos validação cruzada para interromper o treinamento, nos modelos nos quais o treinamento era feito pelo algoritmo LM.

Uma outra opção para evitar o super-ajuste de uma RNA é usar algoritmos de treinamento baseados em *regularização*. Métodos de regularização são amplamente usados em aprendizado de máquina, e também no ajuste de modelos estatísticos de previsão, como os ARIMA. A idéia é simples: para evitar que um modelo se torne demasiado complexo, modificamos a função de custo, acrescentando ao erro quadrático usual E um termo de penalização Ω (multiplicado por uma constante v) como na equação:

$$E^* = E + v\Omega \quad (2.9)$$

Este termo de penalização Ω pode ser definido de várias formas, mas deve refletir a complexidade do modelo (geralmente indicada por seu número de parâmetros). Ao minimizarmos a função de custo resultante E^* , estaremos minimizando o erro quadrático, mas também reduzindo simultaneamente a complexidade. Este método pode ser implementado em RNAs de várias maneiras; usamos neste trabalho o método de regularização bayesiana de McKay [106], que faz uma adaptação do algoritmo de LM para minimizar a

combinação linear:

$$J = \alpha \sum e^2 + \beta \sum w^2 \quad (2.10)$$

onde:

$\sum e^2$: somatório dos quadrados dos erros

$\sum w^2$: somatório dos quadrados dos pesos e bias da RNA

α e β : constantes de ponderação, modificadas ao longo do treinamento

Este método está implementado em Matlab, na função *trainbr*.

Quando um método de regularização é empregado, não é mais necessário usar validação cruzada para interromper o treinamento; nos modelos que testamos, o número de épocas de treino nas RNAs que usaram regularização foi determinado por rodadas de AG.

Fizemos vários experimentos preliminares com os dois métodos, o de LM (com parada por validação cruzada) e o de regularização bayesiana. O método de LM conseguiu geralmente erros médios nas amostras de teste equivalentes aos da regularização, embora convergindo mais rapidamente. Contudo, seus erros tiveram quase sempre uma dispersão muito maior que os da regularização. Essa variabilidade no desempenho não é interessante, pois dificulta o trabalho de seleção das variáveis de entrada por AG (discutido na próxima seção); decidimos, por fim, utilizar somente o método de regularização bayesiana para o treinamento dos modelos cujos resultados serão analisados no Capítulo 3.

2.5 ALGORITMO GENÉTICO (AG)

Uma etapa importante num problema de modelagem por RNAs é o da seleção das variáveis de entrada. Dado um conjunto de variáveis candidatas, há vários métodos que podem ser utilizados para selecionar dentre elas as que devem ser incluídas no modelo. Uma busca exaustiva entre todas os possíveis subconjuntos formados com estas variáveis seria inviável (já que, se existem n variáveis possíveis, haverá 2^n subconjuntos que podem ser formado entre elas). Métodos mais realísticos são os baseados em uma busca seqüencial, em que uma variável é adicionada ou removida do subconjunto de entradas a cada iteração, e o erro de previsão resultante é avaliado [107]; estes métodos são conhecidos como *forward selection* e *backward elimination*. Outra alternativa seria a de utilizar métodos de redução de dimensionalidade, como por exemplo o método estatístico de Análise de Componentes Principais (*Principal Component Analysis* - PCA), que faz uso da matriz de correlação linear entre as variáveis de entrada. Contudo, como a relação entre as variáveis de temperatura e carga presentes nos dados de entrada é claramente não-linear, este método não parece adequado para este problema.

Existem outros métodos e heurísticas já estudados na literatura. Um destes métodos, por exemplo, é o que faz a “Determinação Automática de Relevância” (*Automatic Relevance Determination*) de cada variável, por meio de uma análise baseada na Estatística bayesiana [108]. Estes métodos, em geral, são de aplicação trabalhosa, e nem sempre garantem bons resultados; nenhum deles, por isso, encontrou até agora aceitação geral (ver, por exemplo, [104]). Os artigos publicados sobre previsão de carga a curto prazo dão pouca importância a este tópico.

Como temos que selecionar entre um total de 103 variáveis de entrada (Ver Seção 2.8.2); escolhemos então utilizar o AG como um método para fazer essa busca [109]. O AG (ou outras técnicas de computação evolucionista) tem sido frequentemente usado em associação com as RNAs em problemas de previsão de carga; contudo, ele é utilizado quase sempre para a otimização dos pesos (por exemplo, [55, 44, 110]); há poucos trabalhos publicados descrevendo o uso de AG para auxiliar na definição da estrutura ou dos hiperparâmetros de RNAs.

Numa busca na literatura sobre previsão de cargas a curto prazo, encontramos apenas dois exemplos do uso de AGs para definir a estrutura de uma RNA (uma rede *neuro-fuzzy* em [59], e uma rede de Elmann em [60]), e dois exemplos do uso de AG, em combinação com algoritmos de colônia de formigas, para selecionar variáveis de entrada ([61, 111]).

Além da dificuldade em escolher as variáveis de entrada, outro problema a que surge quando se usam RNAs é a falta de critérios simples para definir os parâmetros do modelo ou de seu treinamento. Sendo assim, acrescentamos ao processo de seleção de variáveis feita pelo AG a busca também pelo número de neurônios da camada interna e o número de épocas no treinamento da RNA.

Como visto na seção anterior, escolhemos utilizar o método de treinamento da RNA com regularização bayesiana. Como esta regula a complexidade da rede, poderia parecer desnecessário usarmos o AG para buscar o número de neurônios na camada interna, ou o número de épocas de treinamento; contudo, testes preliminares mostraram que a escolha desses números influencia no resultado final, e decidimos portanto mantê-los na busca do AG.

Para avaliar a utilidade do AG, tanto na seleção das variáveis de entrada, quanto na definição dos parâmetros da RNA, iremos comparar o desempenho das redes assim especificadas com o de um modelo de RNA especificado heurísticamente [31], que será usado como *benchmark*. Esse modelo, que chamamos de *Modelo0*, será discutido na Seção 2.8.1.

2.5.1 Parâmetros do AG

Como em todo método de otimização, a utilização do AG requer que algumas escolhas sejam feitas previamente. A principal delas é a da função de aptidão (discutida na Seção 2.5.2); outra, a do formato do indivíduos que formam a população. Como a seleção de variáveis é parte do problema a ser otimizado pelo AG, o formato mais simples é o que utiliza uma estrutura binária, onde cada possível variável de entrada é representada por um *dummy*, que recebe valores 0 ou 1, mostrando se aquela variável foi ou não incluída nas entradas da RNA.

Outros variáveis ou características que têm que ser escolhidas, antes que o AG seja usado, são:

- O número de indivíduos da população
- o uso (ou não) de elitismo entre as gerações
- As probabilidades associadas aos operadores de combinação e de mutação
- O método de seleção
- O critério de parada

Essas características foram definidas de forma empírica, a partir da análise do comportamento da evolução dos indivíduos do AG em testes preliminares. Chegamos, depois desses testes, aos seguintes valores:

- População com 50 indivíduos
- Alto índice de elitismo; quase toda a população passava de uma geração para outra.
- O operador de combinação não foi usado; probabilidade de mutação igual a 0.01
- Seleção por torneio
- Parada depois de 200 gerações sem melhorias no valor da aptidão da população

Para avaliar o desempenho do AG definido pelas características acima experimentamos também, para fins de comparação, um AG com elitismo de apenas um indivíduo, operadores de mutação e combinação com probabilidades 0.01 e 0.8 respectivamente, e 100 indivíduos na população. Nas duas subséries do Rio, os dois métodos de AG obtiveram resultados similares; o segundo método não conseguiu resultados melhores que o primeiro, mesmo tendo um custo computacional aproximadamente cinco vezes superior. Esse resultado nos levou a focar o estudo apenas no primeiro método.

Nos dois tipos de AG utilizados, houve uma variabilidade muito grande no resultado das seleções, o que levou a resultados muito diferentes, tanto na composição dos genes, quanto na acurácia das previsões feitas pelas RNAs especificadas de acordo com estes genes. Por isso, rodamos o AG cinco vezes em cada subsérie; no Capítulo 3, iremos comparar os resultados das RNAs obtidas nestas cinco rodadas.

Cada indivíduo foi representado por 113 *bits*. Em resumo, esta foi sua composição:

- os primeiros 103 bits do gene representam as variáveis de entrada da rede neural; cada bit indica se uma variável participa ou não da rede neural.
- os cinco *bits* seguintes representam o número de neurônios ocultos (1 a 32); os cinco *bits* finais o número de épocas para treinamento da RNA (1 a 32).

A definição dos limites máximos atribuídos ao número de neurônios e de épocas de treinamento foi feita na etapa de testes preliminares; nenhum dos resultados obtidos nesta etapa ultrapassou estes limites definidos.

2.5.2 Função de aptidão

Para função de aptidão, utilizamos o conhecido erro médio quadrático (MSE), definido na Equação (2.11).

$$MSE = \frac{1}{N} \sum_{t=1}^N (\hat{L}_t - L_t)^2 \quad (2.11)$$

onde \hat{L}_t é a previsão no tempo t , L_t o valor observado no tempo t , e N o total de valores.

Em um estudo preliminar, testamos também na subsérie Rio II uma segunda função de aptidão F , criada com o objetivo de tentar minimizar a autocorrelação que tipicamente é encontrada na série de erros, nos problemas de previsão de perfis de carga [86]. Esta função F é dada pela Equação (2.13).

$$e_t = L_t - \hat{L}_t \quad (2.12)$$

$$F = \sum_{j=0}^{24} (e_t * e_{t-j})^2 \quad \forall t \quad (2.13)$$

Esta função é derivada da estatística de Ljung-Box [112], conhecida em séries temporais. Nela, tentamos tanto minimizar a variância do erro de previsão, quanto reduzir as autocorrelações da série de erros.

No entanto, o objetivo desta função F de aptidão não foi alcançado, e a mesma não trouxe melhores resultados que a função MSE . Ela foi portanto descartada, e todas as RNAs cujos resultados são mostrados a seguir foram encontradas pela minimização apenas função de aptidão MSE .

Durante o processo de otimização do AG, cada RNA (representada por um indivíduo) é treinada na amostra de treinamento da subsérie, e tem o valor de sua função de aptidão calculado na amostra de validação. Dessa forma, a amostra de teste continuou sendo mantida à parte, e foi usada exclusivamente para a comparação final entre todos os modelos.

2.6 COMBINAÇÃO DE MODELOS

A combinação de modelos é uma técnica muito utilizada nos mais diversos problemas de previsão. Os primeiros estudos foram feitos na década de 1960, e o artigo fundamental sobre o tema foi publicado em 1969 por Bates e Granger [113]. Como argumentado por estes autores, a combinação é útil porque, se há duas (ou mais) previsões para uma mesma variável, estas previsões podem ter sido baseadas em informações distintas, de duas maneiras:

1. Cada previsão pode ter sido feita em função de variáveis não consideradas nas outras previsões.
2. Cada previsão pode ter sido feita a partir de suposições diferentes sobre as relações entre as variáveis.

Estudos empíricos mostram que a combinação entre duas previsões diferentes reduz, em geral, a variância total do erro de previsão [113]. Uma forma usual de se fazer combinarmos de RNAs é por meio de “comitês”: RNAs de diferentes estruturas são treinadas para resolver um mesmo problema, e a previsão final é dada pela média das previsões individuais de cada rede, ponderadas ou não (ver [96], Seção 4.7.3).

No problema de previsão de carga elétrica a curto prazo, alguns estudos recentes [86, 88] têm indicado que combinação de previsões podem levar a previsões bem melhores (tanto em termos de acurácia quanto de precisão) do que as previsões feitas pelos modelos individuais. No modelo proposto nesta tese, iremos combinar as previsões feitas por métodos baseados em RNAs com as previsões feitas pelo método de HWT, por meio de uma média ponderada cujos pesos são estimados iterativamente (mais detalhes serão vistos a seguir, na Seção 2.8.4).

2.7 FUNÇÃO DE ERRO

Para compararmos os resultado do modelo proposto com os dos outros modelos, e com os de trabalhos já publicados, utilizamos como métrica o erro médio absoluto percentual (*Mean Absolute Percentage Error* - MAPE), dado pela Equação (2.14):

$$MAPE = 100 * \frac{1}{N} \sum_{(t=1)}^N \left| \frac{(\hat{L}_t - L_t)}{L_t} \right| \quad (2.14)$$

onde:

\hat{L}_t : previsão da carga no instante t

L_t : valor observado da carga no instante t

N : tamanho da amostra (número de valores observados)

2.8 MODELOS DESENVOLVIDOS

Para avaliar o desempenho do modelo proposto nesta tese (descrito na Seção 2.8.4, abaixo), iremos comparar suas previsões com as de três outros modelos baseados em RNAs, dois deles especificados com auxílio de AG, e um retirado da literatura. Estes modelos foram:

2.8.1 *Modelo0*

A fim de avaliarmos a eficácia do AG, tanto na seleção de variáveis quanto na definição da arquitetura das RNAs, comparamos o desempenho dos modelos que utilizam AG com o de um modelo que foi definido por métodos heurísticos em [31], com as seguintes características:

- 15 neurônios na camada interna
- 24 saídas (1 para cada hora do dia seguinte)
- 50 entradas
 - 24 diferenças de temperatura horárias de hoje para amanhã
 - 24 cargas horárias de hoje
 - 2 variáveis *dummies* representando os dias de sábado e domingo

Em [31], este modelo foi ajustado sobre uma das bases de dados que utilizamos (Rio de Janeiro), e testado na subsérie Rio I.

2.8.2 Modelo1

O primeiro modelo criado neste trabalho, com características especificadas por meio de AG, é uma RNA com 103 possíveis variáveis de entradas e 24 saídas (previsões das 24 cargas horárias do dia seguinte), como na Figura 10. Este modelo será denominado *Modelo1*. As variáveis de entrada são:

- 24 diferenças entre as temperaturas horárias de hoje e as de ontem.
- 24 diferenças entre a temperatura de amanhã e as de hoje.
- 48 cargas horárias, de hoje e de ontem.
- 7 *bits* para representar os dias da semana.

O AG irá selecionar as variáveis de entrada a serem incluídas no modelo, entre estas 103 candidatas; além disso, irá selecionar o número de neurônios na camada oculta, e o número de épocas de treinamento. Como foram feitas cinco rodadas do AG, isto significa que houve cinco versões destes modelos, cada uma com variáveis de entrada e características diferentes.

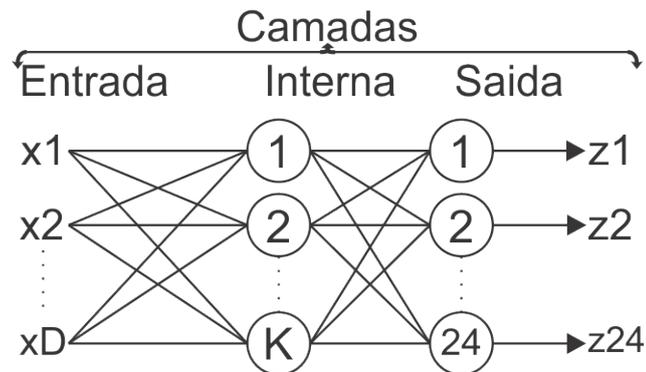
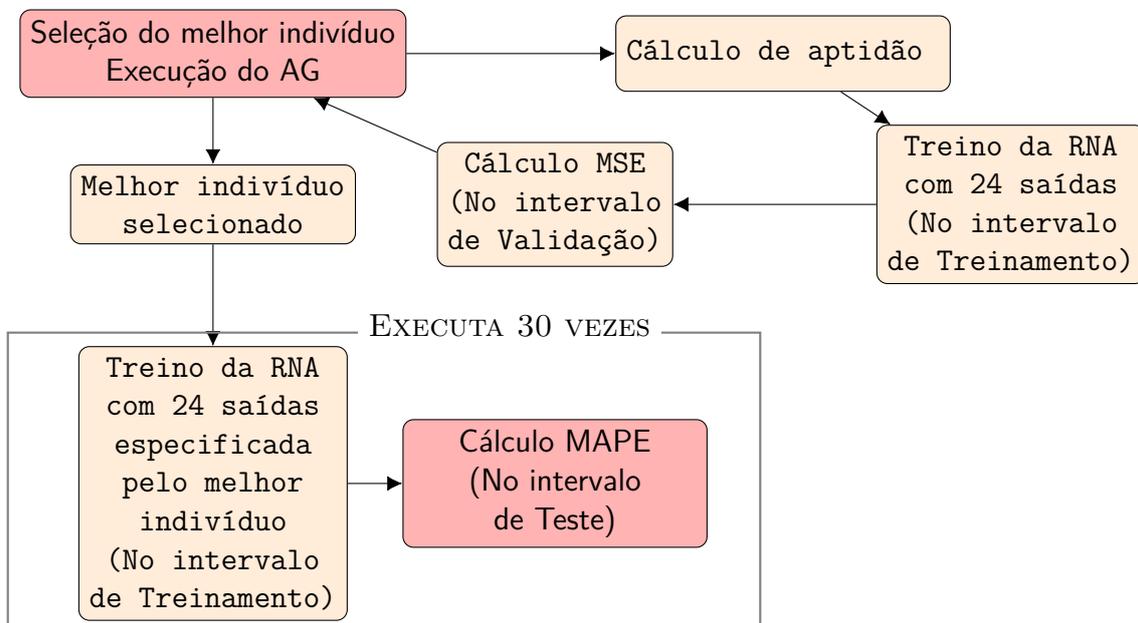


Figura 10 – *Modelo1*: RNA simples com 24 saídas

A Figura 11 mostra o fluxograma que descreve as etapas necessárias para a especificação dos modelos de RNA, seu treinamento e teste.

Figura 11 – Fluxograma - *Modelo1*

2.8.3 Modelo2

O segundo modelo criado neste trabalho é uma adaptação do *Modelo1*. São utilizadas redes com as mesmas variáveis candidatas de entrada, porém há apenas uma saída em cada rede (previsão da carga em uma hora específica para do dia seguinte). O modelo é composto por 24 redes separadas, uma para cada hora, como na Figura 12. Este modelo será denominado *Modelo2*.

Novamente, o AG irá selecionar as variáveis de entrada a serem incluídas em cada uma das 24 RNAs, entre as variáveis candidatas, além de selecionar o número de neurônios nas camadas ocultas, e o número de épocas de treinamento. Foram feitas cinco rodadas do AG, o que significa que houve cinco versões destes modelos, cada uma com variáveis de entrada e características diferentes.

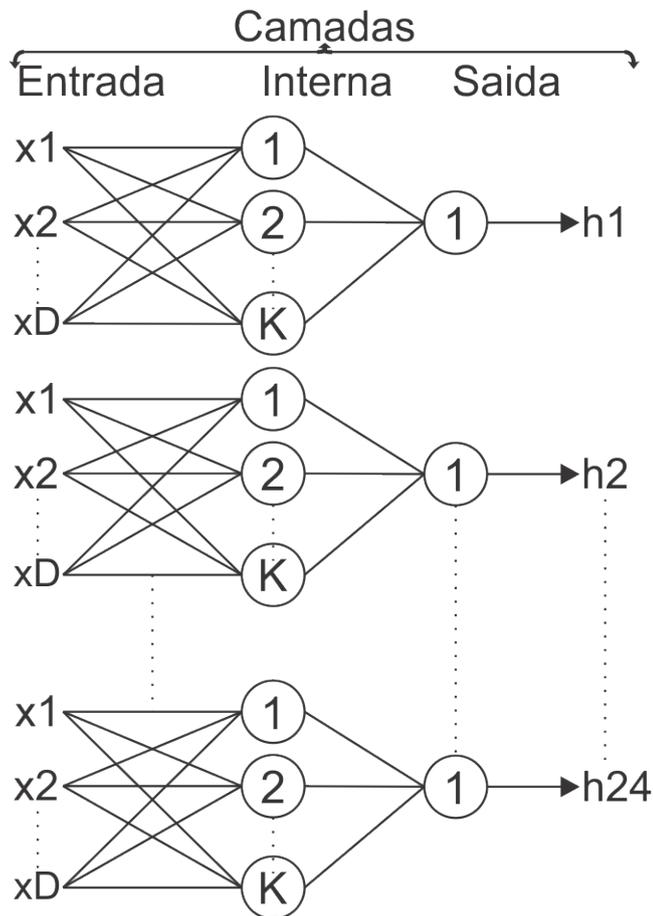
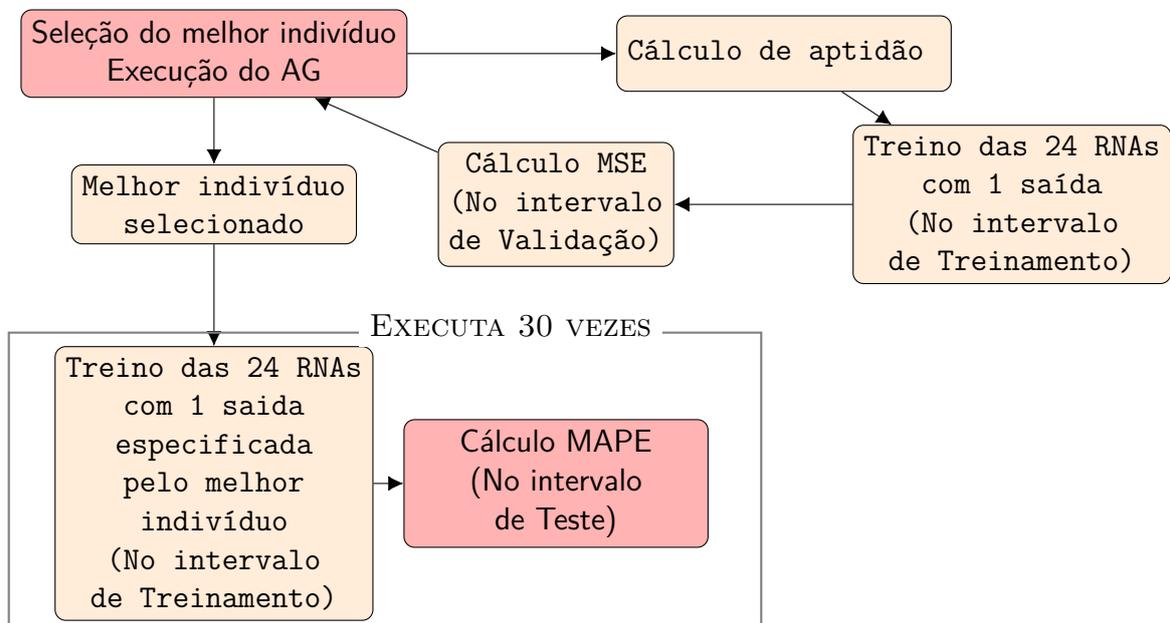


Figura 12 – *Modelo2*: 24 RNAs em paralelo

A Figura 13 mostra o fluxograma que descreve as etapas necessárias para a especificação dos modelos de RNA, seu treinamento e teste.

Figura 13 – Fluxograma - *Modelo2*

2.8.4 *Mod_Comb*

Este é o modelo que esta tese propõe. Nele, as previsões são feitas a partir da combinação das previsões do *Modelo2* com as do método linear HWT. Esta combinação é feita de forma linear, com apenas um parâmetro que precisa ser ajustado, como visto na Equação (2.15). Este modelo será denominado Modelo Combinado, e será representado por *Mod_Comb*.

$$\widehat{L}_i^{FINAL} = \alpha \widehat{L}_i^{Modelo2} + (1 - \alpha) \widehat{L}_i^{HWT} \quad (2.15)$$

onde:

$\widehat{L}_i^{Modelo2}$: previsão da carga no instante t feita pelo *Modelo2*

\widehat{L}_i^{HWT} : previsão da carga no instante t feita pelo HWT

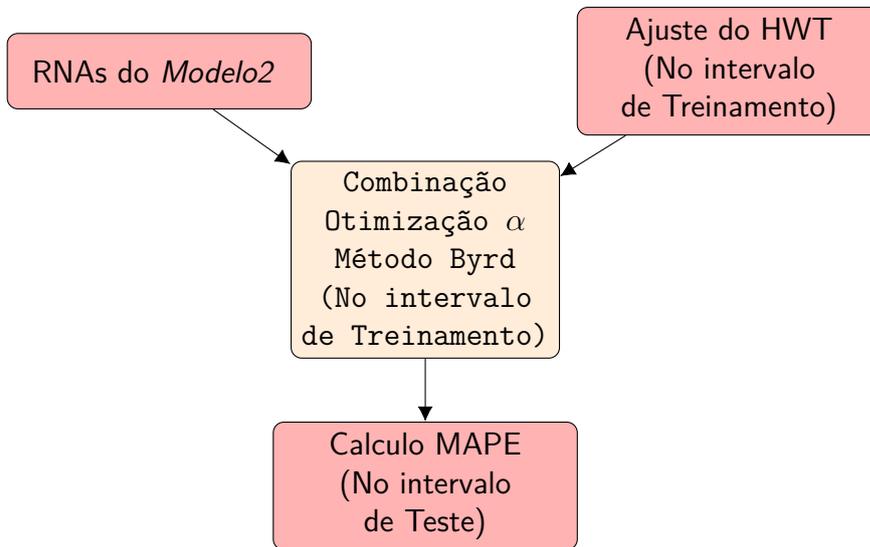
\widehat{L}_i^{FINAL} : previsão final combinada.

α : constante que pondera a combinação

O ajuste do parâmetro α foi obtido pelo método de Byrd [114], que inclui restrições de máximo e mínimo para os parâmetros (no caso, $\alpha \in [1, 0]$). O intervalo utilizado no processo de ajuste foi o mesmo intervalo de treinamento das subséries. (Mais informações sobre o método de Byrd podem ser encontradas no Anexo 5.4).

Como existem cinco versões diferentes do *Modelo2* (selecionadas em cinco rodadas diferentes do AG), e apenas uma do HWT, as combinações resultaram em cinco versões diferentes do *Mod_Comb*.

A Figura 14 mostra o fluxograma que descreve as etapas necessárias para a especificação e treinamento dos modelos de RNA, o ajuste do HWT, e a combinação destes dois métodos.

Figura 14 – Fluxograma - *Mod_Comb*

2.9 ANÁLISE DE SENSIBILIDADE DO MODELO À TEMPERATURA

Uma preocupação em relação à utilização dos dados de temperatura neste trabalho é o fato de termos utilizado nas entradas dos modelos os valores observados, e não os valores previstos para as temperaturas do dia seguinte. Isso é comumente feito na maior parte dos artigos publicados sobre o tema de previsão de carga a curto prazo; isto acontece, em parte, porque nesses artigos o foco está na previsão de carga, e não na previsão da temperatura; em parte, porque as séries de temperatura previstas em geral são difíceis de se obter.

A maioria dos artigos, porém, não discute as implicações desta substituições de valores previstos por valores observados na entrada dos modelos. Quando são usadas verdadeiras previsões, é lógico esperar que o desempenho dos modelos se deteriore, uma vez que previsões sempre são incertas, e têm erros embutidos em seus valores. Em consequência disto, modelos complexos, que mostram excelentes resultados nos experimentos relatados nos artigos, talvez tenham pouca acurácia quando usados em aplicações com dados reais. Há muito poucos artigos dedicados a estudar este problema. Um artigo que estuda o efeito da incerteza na previsão de temperatura sobre a previsão de cargas (embora utilizando modelos bayesianos, não RNAs), observa que os erros da maior parte dos artigos publicados são subestimados, pelo fato de os testes serem feitos sobre as temperaturas reais, não sobre as previsões [115]. Outro estudo do efeito da incerteza na previsão da temperatura (mas, desta vez, na previsão do custo operacional), está em [116].

Para avaliarmos a sensibilidade dos modelos ao erro nas previsões da temperatura, criamos cinco séries de “previsões” simuladas de temperatura, adicionando diferentes níveis de erros aleatórios às séries de valores observados. Estes erros tiveram suas variâncias especificadas de forma que as previsões simuladas tivessem MAPEs de 1%, 2%, 3%, 4% e 5%. Para cada nível de MAPE, geramos dez séries diferentes; obtivemos então 50 séries de previsões simuladas.

Analisaremos na Seção 3.3 o desempenho dos diferentes modelos de previsão quando as séries de previsões simuladas de temperatura foram usadas nas entradas das redes, e compararemos os resultados com os obtidos quando as temperaturas reais foram usadas. Para essa etapa do trabalho, usamos apenas uma das versões de cada modelo, em cada subsérie; para selecionar esta versão, utilizamos uma heurística simples: usamos a versão que obteve o melhor MAPE médio. No caso de empate (sensibilidade de duas casas decimais), usamos a versão que teve o menor MAPE mínimo e, em seguida, a que teve o menor intervalo interquartil nos MAPEs.

3 RESULTADOS

3.1 COMPARAÇÃO DOS MAPES

Para validar o modelo proposto, compararemos seu desempenho com o de dois *benchmarks* ingênuos (*Naïve24* e *Naïve168*), com o do método de HWT usado individualmente, e com três modelos baseados puramente em RNAs (para detalhes sobre a arquitetura destas redes, ver Seção 2.8):

- *Modelo0*: RNA com a arquitetura testada em [31] sobre a mesma subsérie Rio I usada neste trabalho;
- *Modelo1*: RNA de 24 nós de saídas, cada um responsável por prever a carga a cada hora do dia seguinte; seleção de variáveis de entradas e arquitetura feita por AG.
- *Modelo2*: 24 RNAs de um nó de saída único, em paralelo, cada RNA responsável por prever a carga em uma hora determinada do dia seguinte; seleção de variáveis de entradas e arquitetura feita por AG.

O modelo proposto nesta tese faz uma combinação linear das previsões do *Modelo2* com os do método *HWT*, e será denotado a seguir por "*Mod_Comb*".

Uma vez que treinamento das RNAs parte de pesos iniciais aleatórios, fizemos 30 rodadas de treinamos de cada uma das arquiteturas escolhidas, mostramos os resultados obtidos (os erros MAPES de cada rodada) por meio de *boxplots*, facilitando assim a comparação. Fizemos alguns testes com diferentes números de rodadas de treinamento, e verificamos que um número de trinta rodadas era suficiente para nos permitir comparar o comportamento final da RNAs.

A Figura 15 apresenta os resultados obtidos na subsérie Rio I. O *Modelo0* tinha sido originalmente estudado e ajustado sobre essa mesma subsérie, e obteve resultados melhores do que os do *Modelo1* (nas demais subséries, contudo, o desempenho do *Modelo0* piora significativamente). Observa-se que as redes do *Modelo2* tiveram em geral melhor desempenho do que as do *Modelo1*, particularmente no que se refere a dispersão dos erros. Os resultados das redes do *Mod_Comb*, contudo, são praticamente idênticos aos do *Modelo2*. Isso será discutido na próxima Seção (3.2), onde analisaremos o método de combinação.

Uma observação importante a ser feita é a de que existe, neste caso, uma grande variabilidade entre os resultados das RNAs selecionadas pelo AG, o que pode ser constatado comparando as cinco versões diferentes feitas para cada modelo; o problema da definição da arquitetura da redes pelo AG será discutido mais adiante, na Seção 3.4.

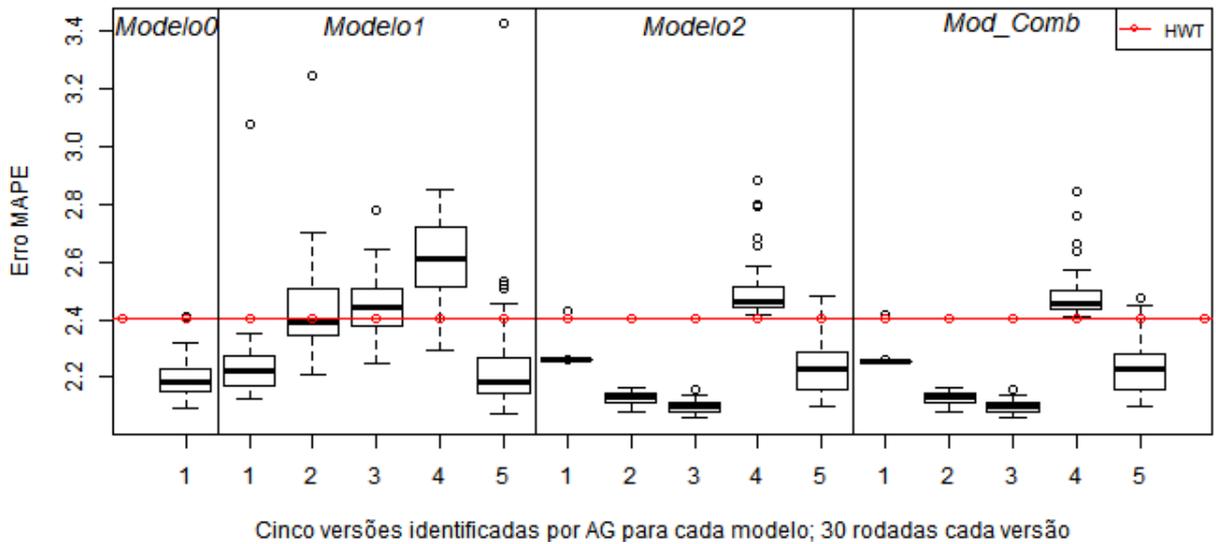


Figura 15 – Comparação dos MAPEs de todos os resultados - Rio I

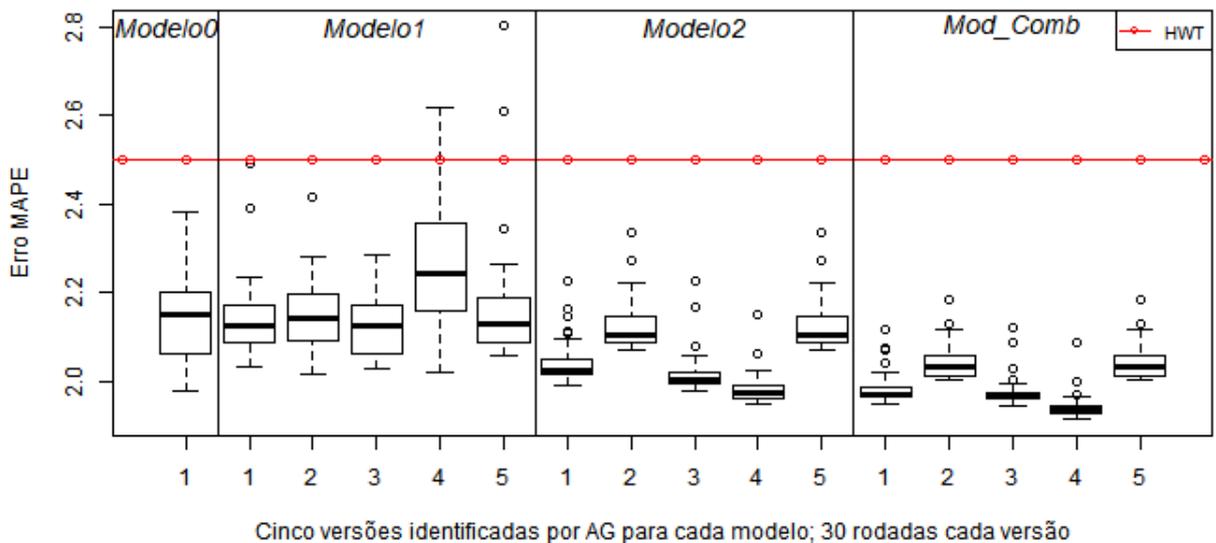


Figura 16 – Comparação dos MAPEs de todos os resultados - Rio II

Ainda na base de dados do Rio, a Figura 16 mostra os resultados obtidos na subsérie Rio II. Aqui, as redes do *Modelo1* tiveram em geral desempenho próximas da rede do *Modelo0*. As redes do *Modelo2* mostraram um resultado bem melhor, tanto em termos da mediana quanto da dispersão dos erros. Em relação a elas, as redes do *Mod_Comb* conseguiram apenas uma pequena melhora.

As Figuras 17, 18, 19 mostram os resultados obtidos nas subséries Washington I, II e III respectivamente. Nota-se que o desempenho das redes dos *Modelo1* e *Modelo2* foram melhores em mediana que a rede do *Modelo0*, em todas as três subséries; isto era de se esperar, uma vez que a RNA do *Modelo0* tinha sido especificada originalmente para os dados de outra base de dados (a subsérie Rio II). Os resultados das redes do *Modelo2*

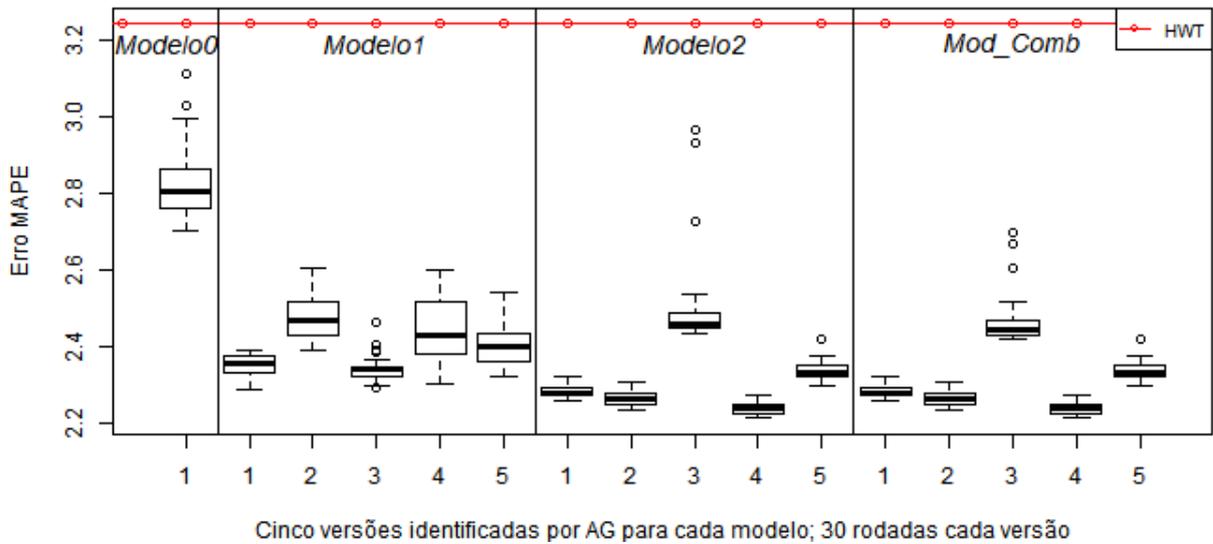


Figura 17 – Comparação dos MAPEs de todos os resultados - Washington I

foram melhores, em termos de dispersão, que os dos *Modelo1* e *Modelo0*. Novamente, verificamos também que o desempenho das redes do *Mod_Comb* foi pouco diferente das do *Modelo2*.

Há vários fatores que podem justificar a diferença de desempenho dos modelos nas diferentes bases de dados. Um deles é a possível diferença da composição das cargas elétricas (normalmente classificadas como residencial, comercial, industrial e pública). Em geral, as cargas industriais são mais fáceis de prever do que as outras, devido à regularidade nas rotinas de funcionamento das indústrias. Outro fator que pode tornar uma série mais fácil ou mais difícil de se prever é a estação do ano em que se encontra a subsérie testada; em algumas estações, as flutuações de temperatura e outras variáveis meteorológicas são mais severas, o que pode causar alteração no consumo de energia. Por fim, existem também os fatores culturais. As séries de carga, como quaisquer séries de consumo estudadas na Economia, indicam o comportamento de uma população, e são influenciadas pelas características sociais e econômicas desta população. Não conhecemos, porém, praticamente nenhum detalhe sobre a composição destas séries (que foram obtidas online) ou sobre o comportamento dos consumidores. Esta falta de conhecimento sobre a realidade em que o problema de previsão está inserido é uma justificativa para o emprego de técnicas de inteligência computacional, como o AG, que busquem extrair informação dos dados disponíveis, de forma automática, e usar esta informação para definir os detalhes das RNAs.

As Figuras 20 e 21 mostram os resultados obtidos nas duas subséries da base de dados da Eslováquia. Não existem, para estas séries, observações horárias da temperatura, mas apenas as médias diárias destas temperaturas. Estas médias, aparentemente, trazem pouca informação para os modelos, e nota-se que a RNA usada com *benchmark*

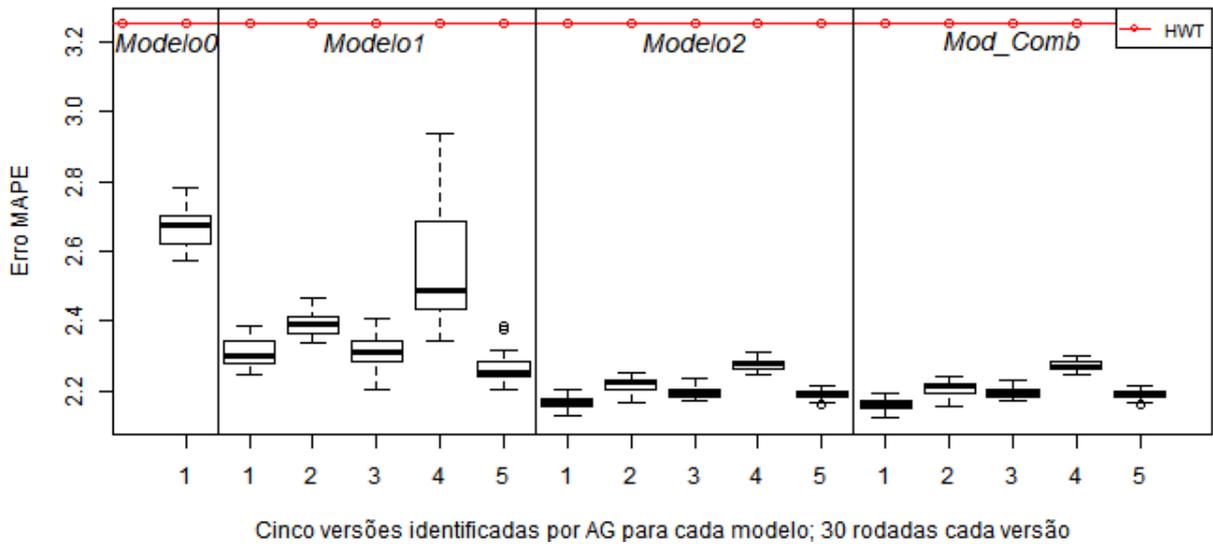


Figura 18 – Comparação dos MAPEs de todos os resultados - Washington II

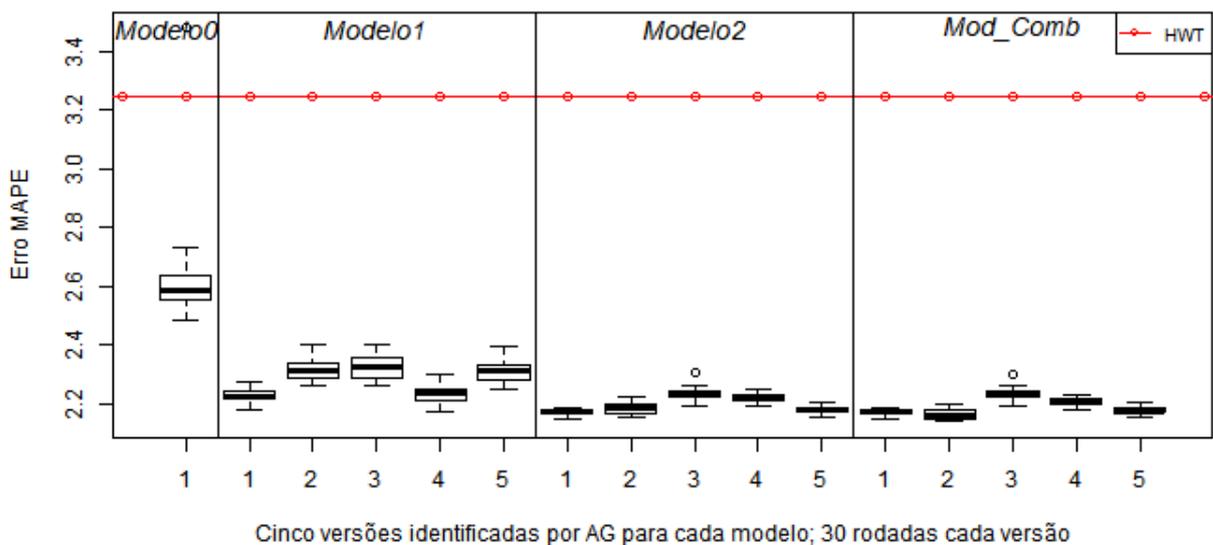


Figura 19 – Comparação dos MAPEs de todos os resultados - Washington III

(*Modelo0*) teve um desempenho muito pior do que o HWT (que é um método univariado, e não utiliza dados de temperatura). Nestas subséries, as versões do *Mod_Comb* tiveram um desempenho um pouco melhor do que as do *Modelo2*, especialmente em termos das dispersões dos MAPEs; ou seja, a combinação de RNAs com o método HWT conseguiu um resultado melhor do que os modelos usando simplesmente RNAs. Estes resultados serão melhor analisados na Seção 3.2.

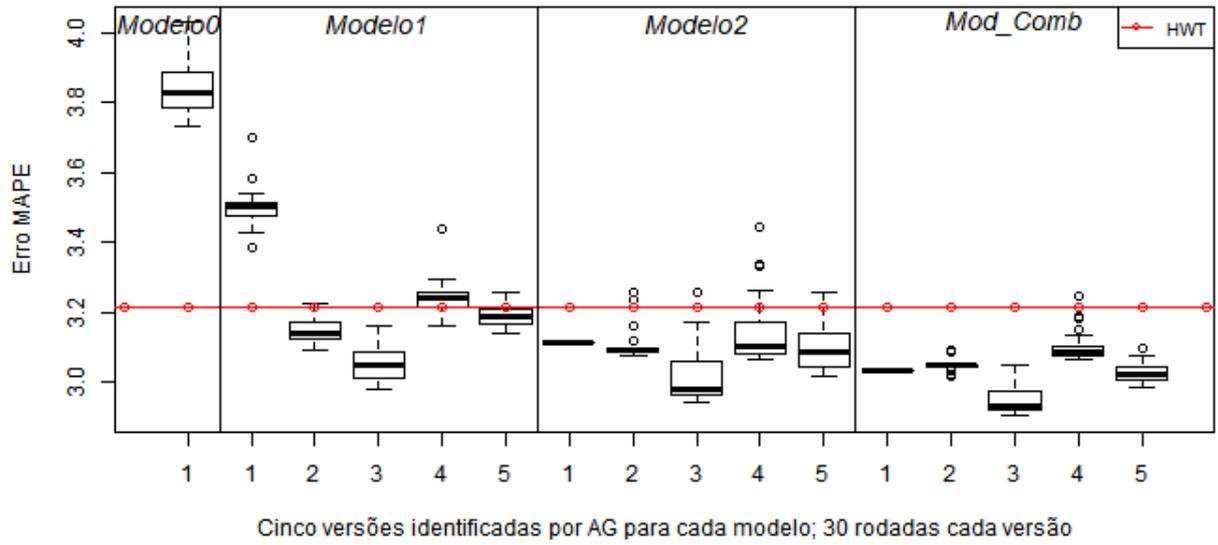


Figura 20 – Comparação dos MAPEs de todos os resultados - Eslováquia I

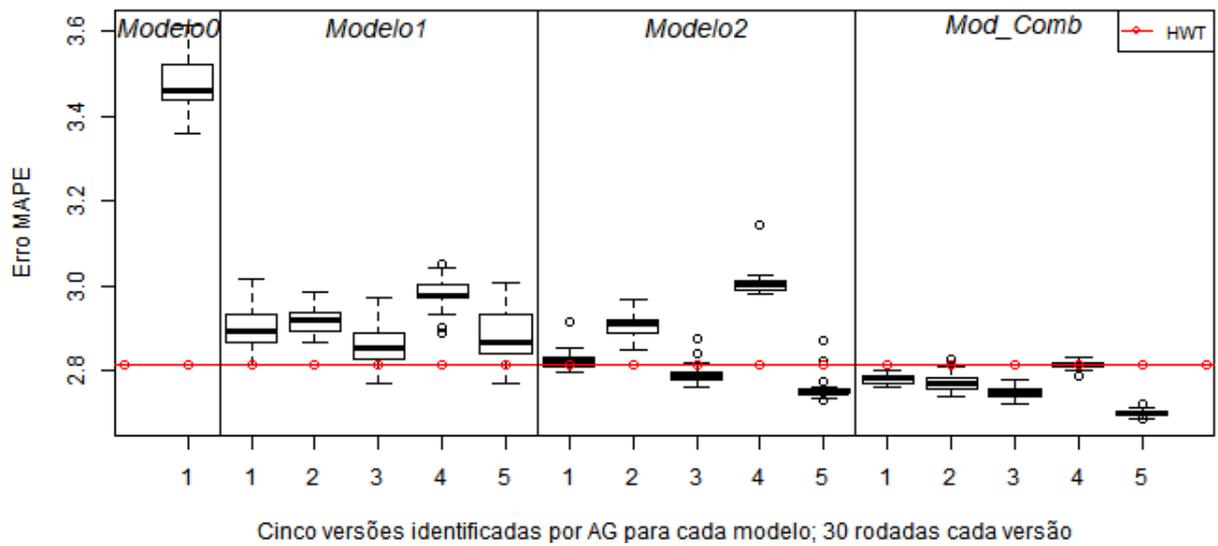


Figura 21 – Comparação dos MAPEs de todos os resultados - Eslováquia II

3.2 ANÁLISE DA COMBINAÇÃO

Faremos nesta seção uma análise das combinações feitas no *Mod_Comb*, reunindo as previsões das RNAs do *Modelo2* e as previsões do método linear de HWT.

Utilizamos o método de Byrd, explicado na Seção 5.4 para definir o valor do parâmetro α , que pondera a combinação, na forma linear dada por:

$$\widehat{L}_i^{FINAL} = \alpha \widehat{L}_i^{Modelo2} + (1 - \alpha) \widehat{L}_i^{HWT} \quad (3.1)$$

onde:

$\widehat{L}_i^{Modelo2}$ é a previsão no tempo t fornecida pelo *Modelo2*

\widehat{L}_i^{HWT} é a previsão no tempo t fornecida pelo HWT

α constante que pondera a combinação

\widehat{L}_i^{FINAL} a previsão final combinada.

O valor de α foi estimado utilizando-se os dados das mesmas subséries usadas para treinamento das RNAs do *Modelo2*; isto é, o ajuste foi feito em um intervalo diferente do intervalo de teste.

Para avaliarmos o comportamento da combinação utilizada no *Mod_Comb*, usamos como *benchmark* uma combinação *naïve* onde o α não foi definido por otimização, e sim definido previamente com o valor de 0.5, dando pesos iguais às previsões do *Modelo2* e às do HWT.

A combinação otimizada pelo método de Byrd obteve resultados melhores do que a *naïve* em quatro das sete subséries, em termos medianos, como pode ser visto nas Figuras 25, 26, 27 e 28 (subsérie Rio II, e todas as três subséries de Washington). Contudo, as Figuras 22, 23 e 24 mostram que em três das sete subséries de teste (Rio I, Eslováquia I e Eslováquia II, respectivamente), a combinação de Byrd obteve resultados piores que a *naïve*. Isto sugere que ainda podem ser feitas melhorias no método de otimização utilizado.

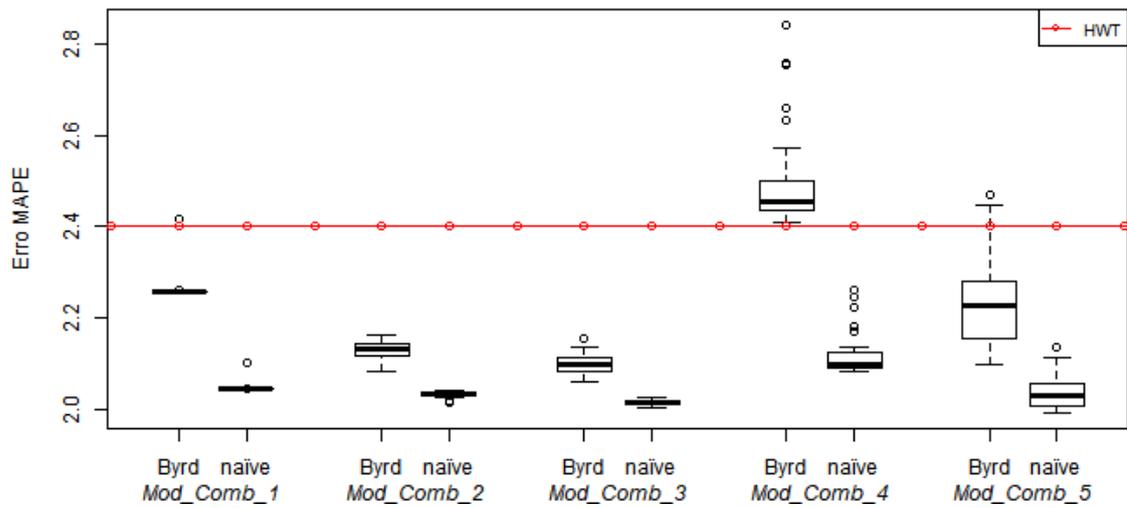


Figura 22 – Comparação das combinações naïve e Byrd - Rio I

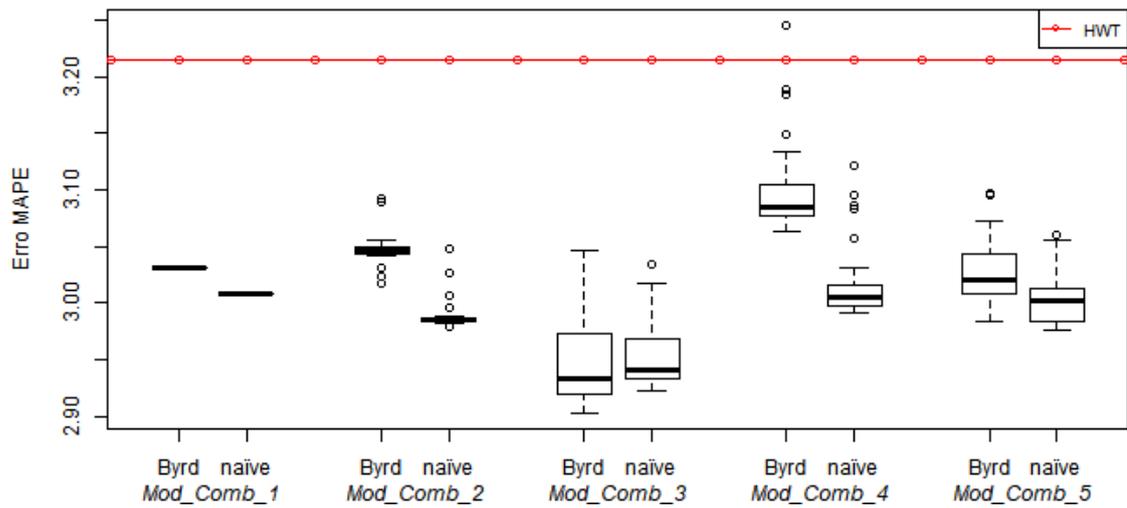


Figura 23 – Comparação das combinações naïve e Byrd - Eslováquia I

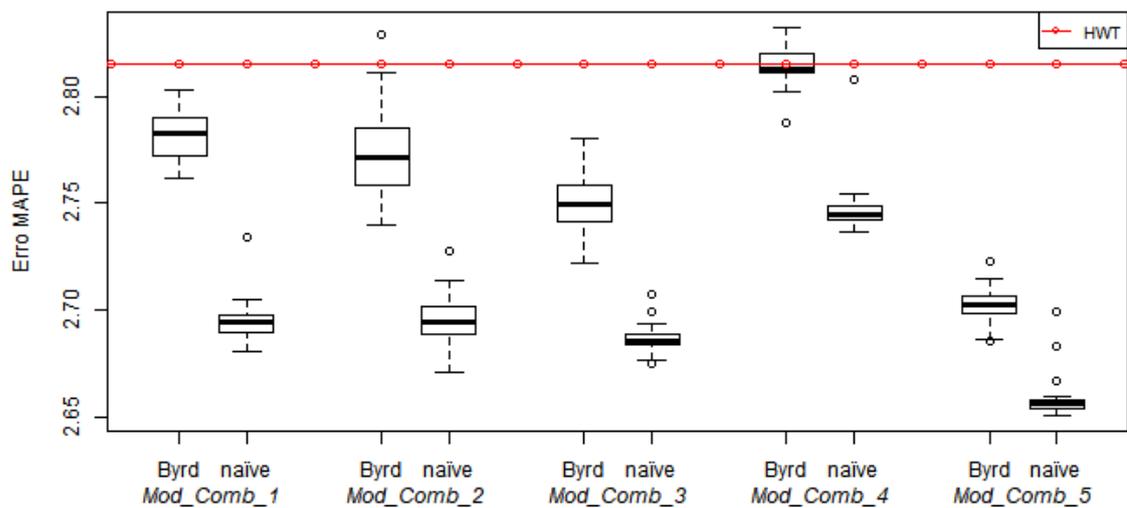


Figura 24 – Comparação das combinações naïve e Byrd - Eslováquia II

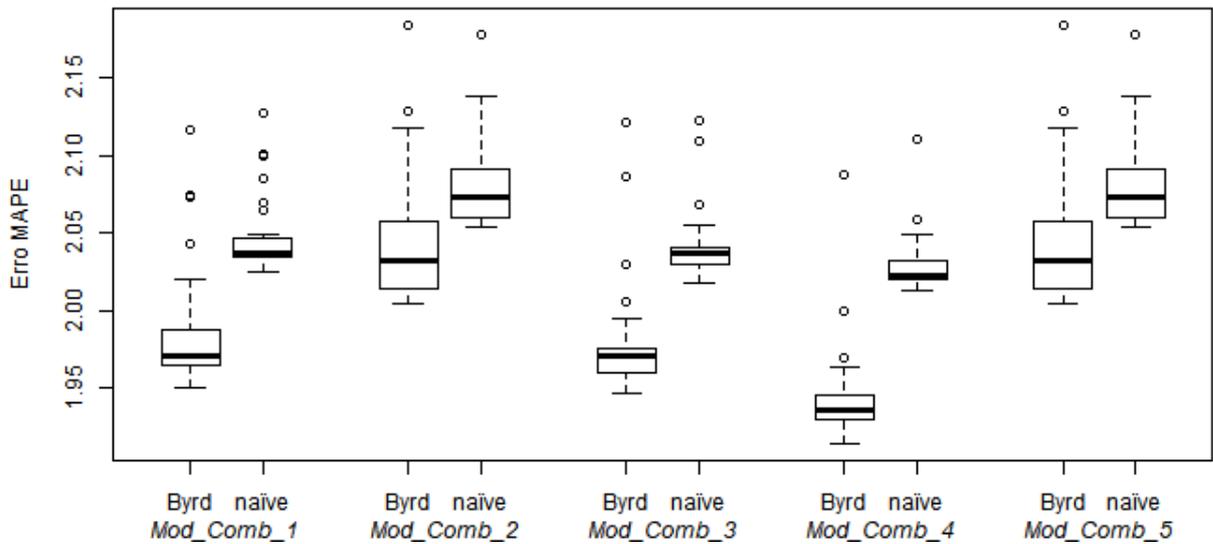


Figura 25 – Comparação das combinações *naïve* e Byrd - Rio II

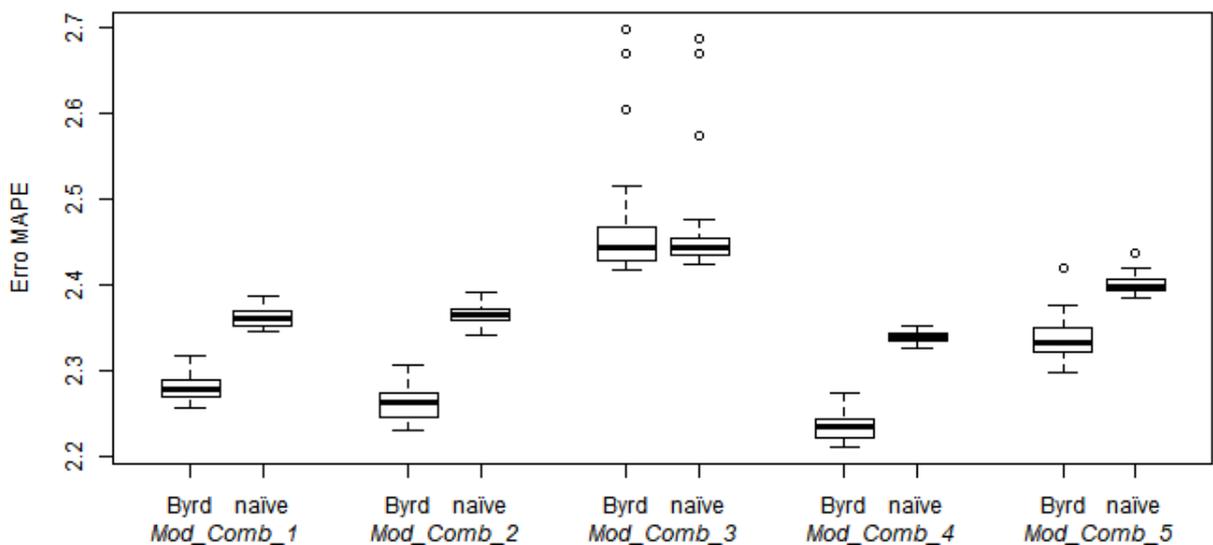


Figura 26 – Comparação das combinações *naïve* e Byrd - Washington I

Em geral, o comportamento foi bem parecido entre as várias combinações feitas sobre uma mesma subsérie. Quando uma destas combinações obtinha melhores resultados com a otimização de Byrd, as demais, na maioria das vezes, também conseguiam resultados melhores utilizando esta otimização; da mesma forma, quando uma das combinações *naïve* obtinha melhores resultados, as demais também ficavam melhores com a combinação *naïve*.

A grande contribuição da combinação de modelos será vista na Seção 3.3, que analisa a sensibilidade do *Mod_Comb* a possíveis erros na previsão da temperatura futura (até o momento, as temperaturas utilizadas na entrada das RNAs foram temperaturas observadas, e não previsões das mesmas).

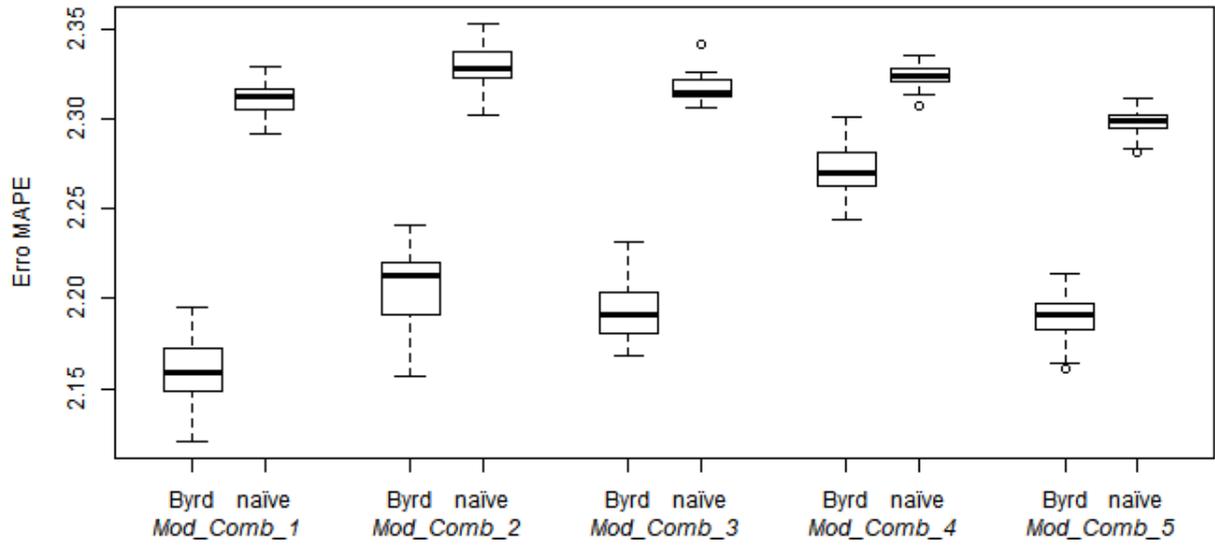


Figura 27 – Comparação das combinações naïve e Byrd - Washington II

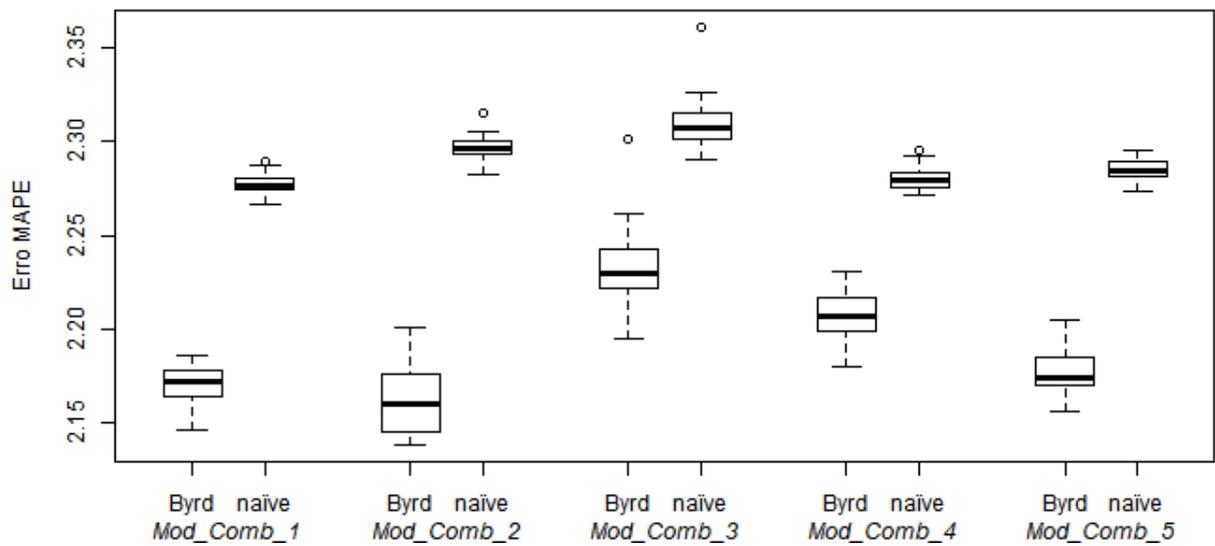


Figura 28 – Comparação das combinações naïve e Byrd - Washington III

3.3 ANÁLISE DE SENSIBILIDADE À TEMPERATURA

Para as análises feitas nesta seção, selecionamos apenas a melhor versão do *Mod_Comb* para subsérie, dentre as cinco disponíveis, e avaliamos seu desempenho quando são utilizadas na entrada das RNAs séries de temperatura com diferentes níveis de ruído adicionado. As versões do *Mod_Comb* selecionados para cada subsérie foram:

subsérie	versão do <i>Mod_Comb</i>
Rio I	3
Rio II	4
Washington I	4
Washington II	1
Washington III	2
Eslováquia I	3
Eslováquia II	5

Para cada subsérie, foram geradas séries simuladas, pela adição de ruídos gaussianos à série de temperaturas observadas. Empregamos cinco níveis crescentes de ruído, de forma que as séries resultantes tivessem erros MAPE, em relação às temperaturas observadas, de 1%, 2%, 3%, 4% e 5%. Para cada um destes níveis de ruído, geramos 10 séries simuladas distintas. Considerando que cada versão do *Mod_Comb* foi treinada 30 vezes em cada subsérie, obtivemos então 300 resultados para cada nível de ruído considerado.

Faremos a seguir uma análise da evolução do erro MAPE do modelo e do valor da constante α que pondera a combinação entre as previsão do *Modelo2* e as do HWT, à medida que aumenta o erro na temperatura. Em cada gráfico nas figuras a seguir, a janela superior mostra os MAPEs dos modelos; a janela inferior mostra a evolução do valor da constante α . Ambas as janelas estão divididas em três partes:

1. A primeira mostra os resultados das versões *Modelo2*, que não tem combinação com o HWT (o que é equivalente a uma combinação onde $\alpha = 1.$);
2. A segunda mostra os resultados do *Mod_Comb* (combinações entre o *Modelo2* e HWT), e os valores da constante α dessas combinações, conforme otimizado pelo método de Byrd (ver Seções 3.2 e 2.8.4), para os diferentes níveis de ruído na temperatura;
3. A terceira mostra os resultados do *Mod_Comb*, usando agora uma combinação *naïve* ($\alpha = 0.5$).

A primeira conclusão que pode ser obtida, a partir da análise destes gráficos, é a de que a combinação do *Modelo2* com o HWT, seja ela de Byrd ou *naïve*, conseguiu reduzir visivelmente a sensibilidade do *Mod_Comb* em relação aos erros de previsão da temperatura.

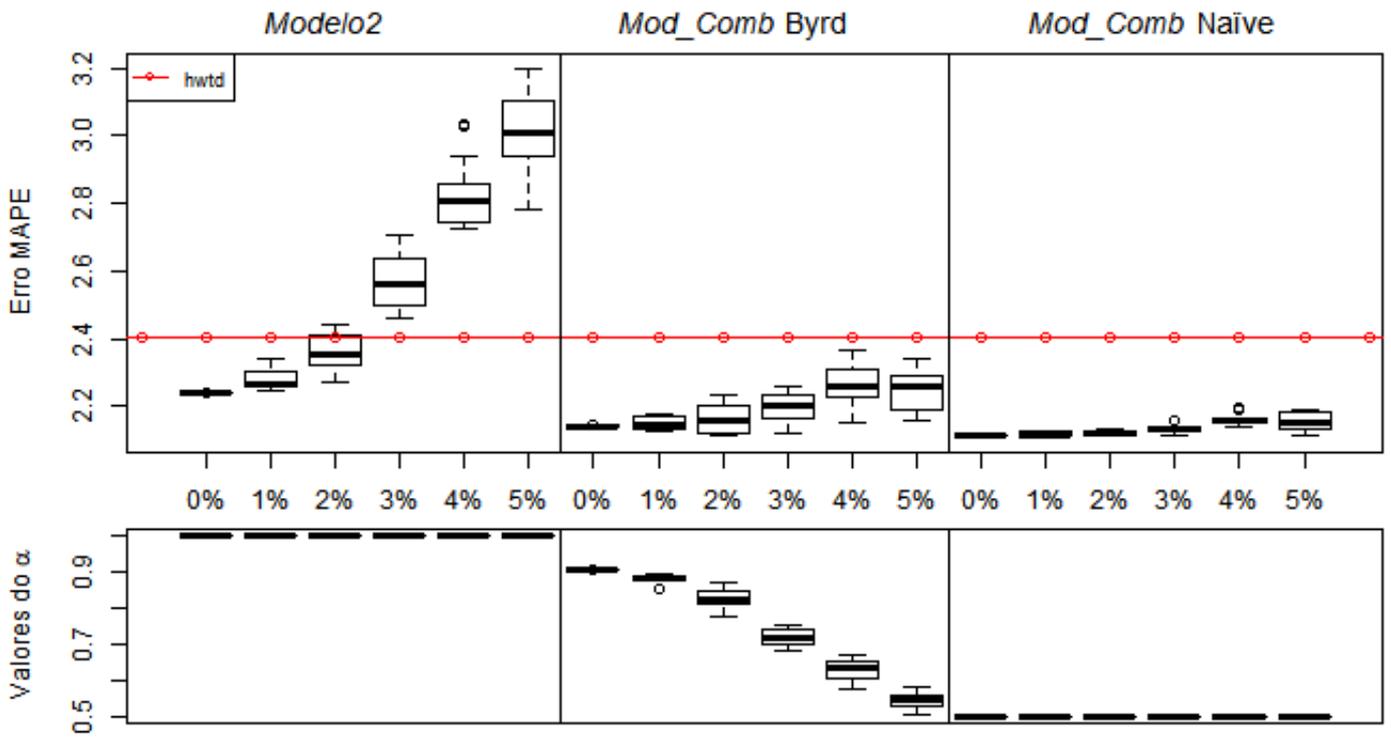


Figura 29 – Análise de sensibilidade - *Modelo2-3* e *Mod_Comb-3* - Rio I

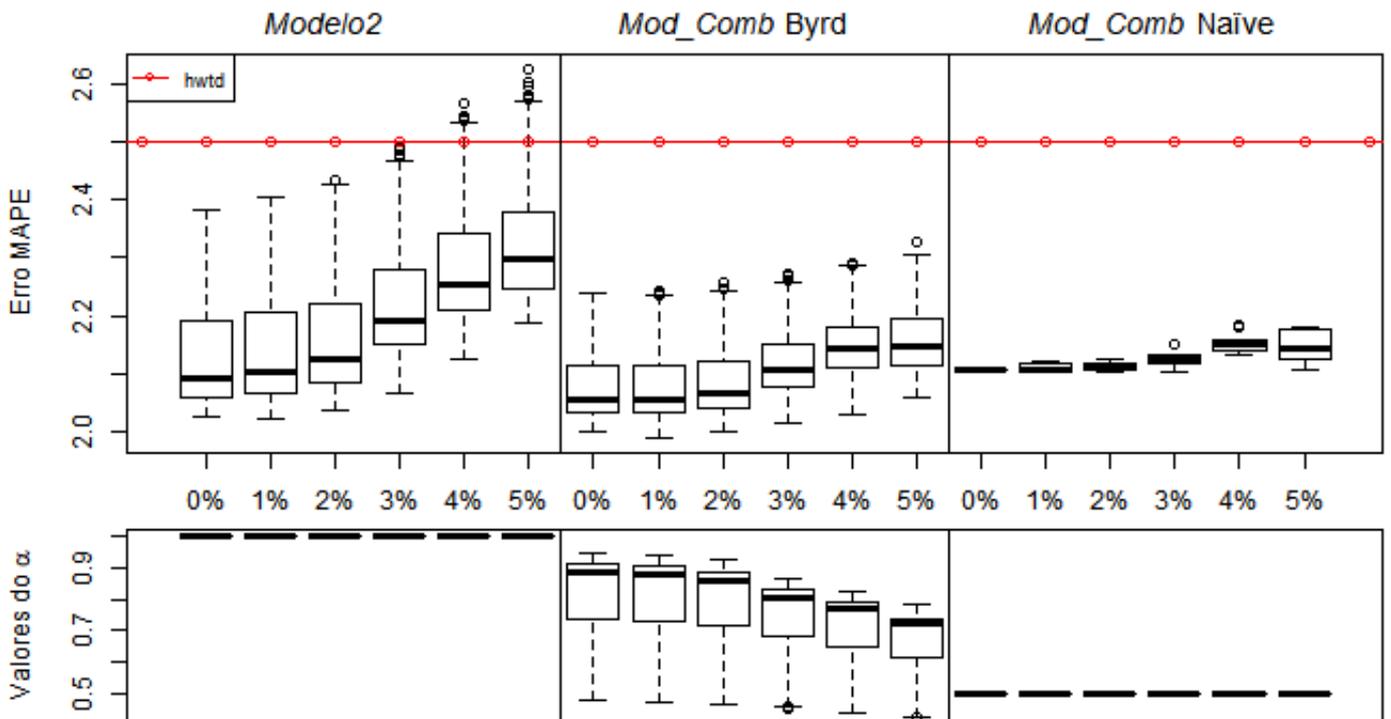


Figura 30 – Análise de sensibilidade - *Modelo2-4* e *Mod_Comb-4* - Rio II

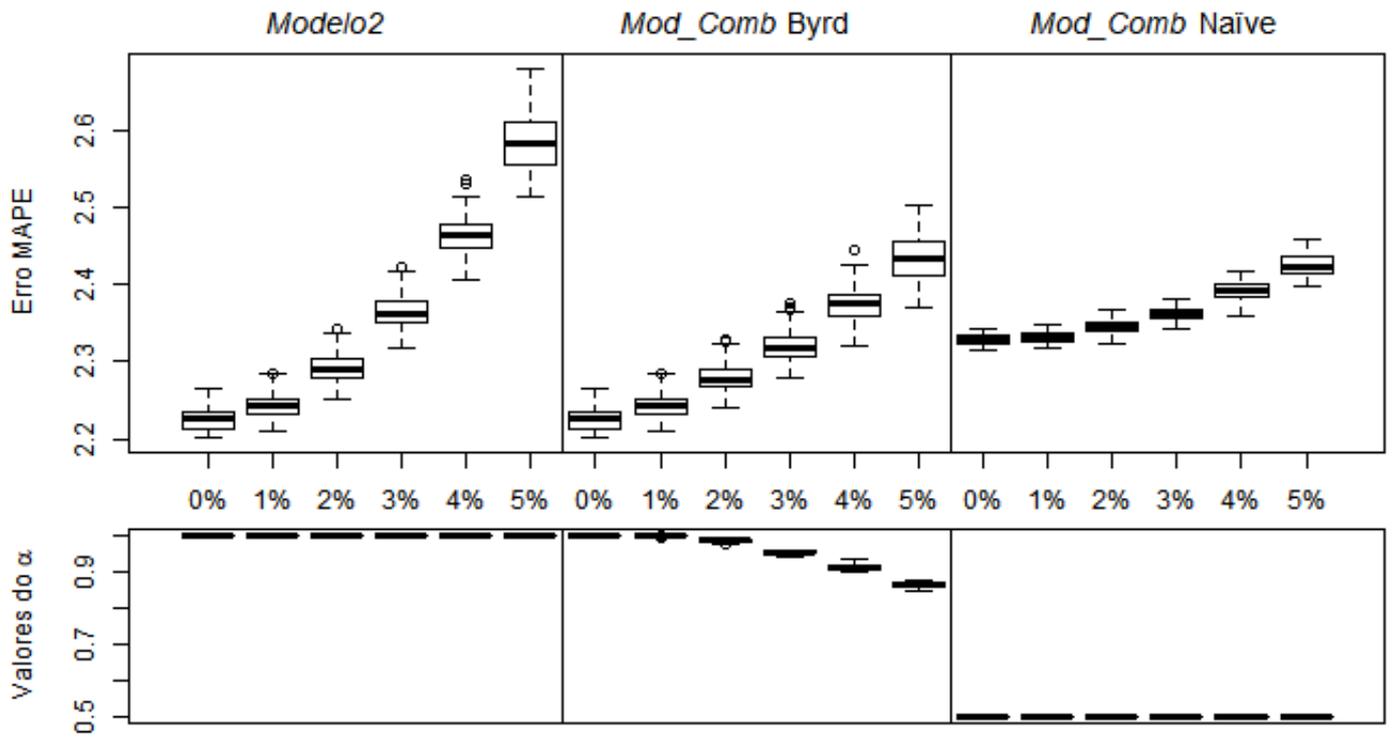


Figura 31 – Análise de sensibilidade - *Modelo2-4* e *Mod_Comb-4* - Washington I

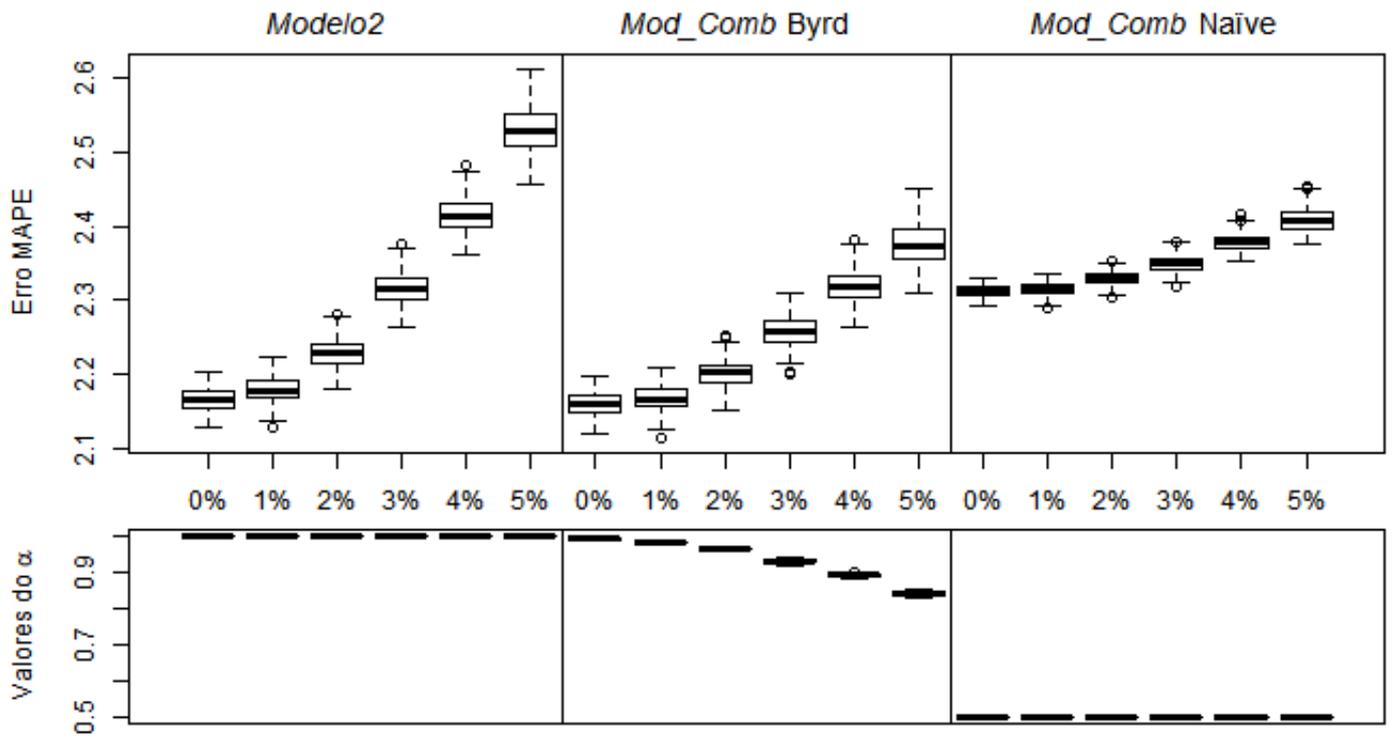


Figura 32 – Análise de sensibilidade - *Modelo2-1* e *Mod_Comb-1* - Washington II

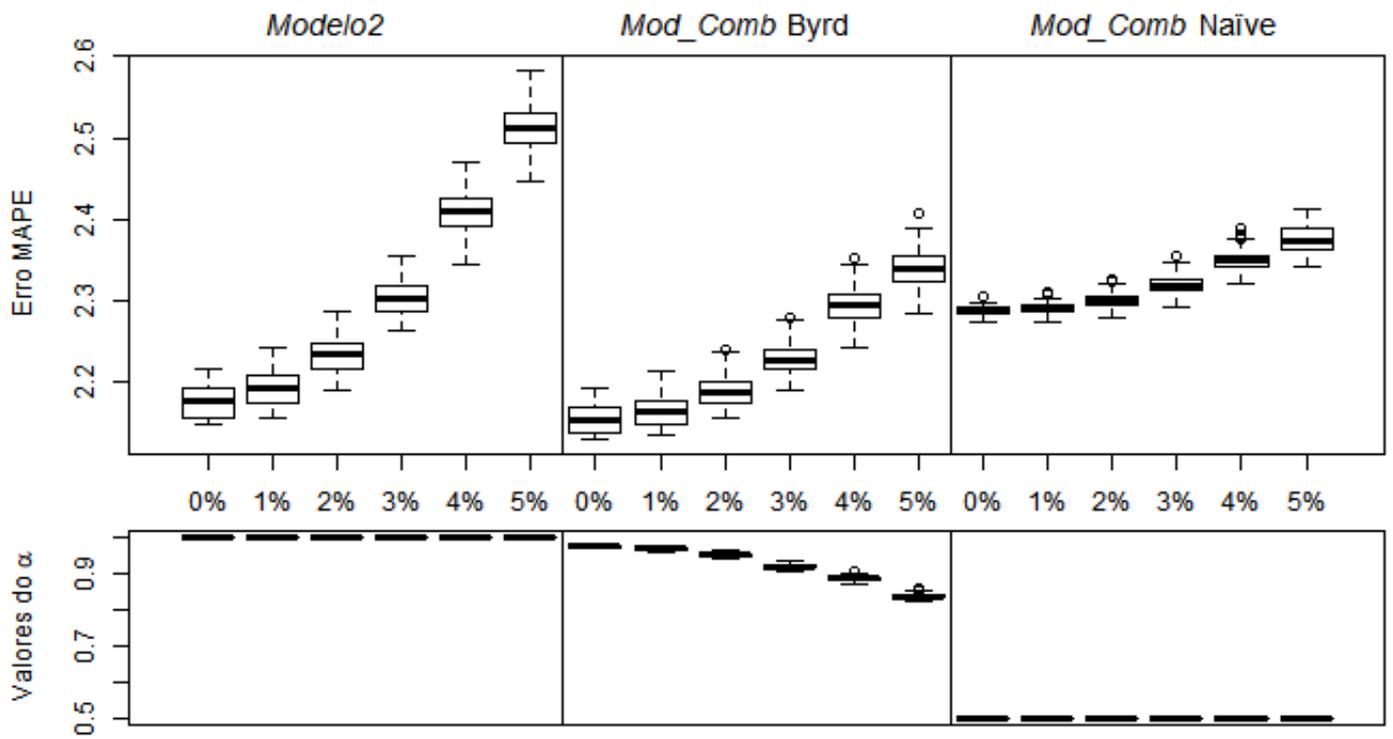


Figura 33 – Análise de sensibilidade - *Modelo2-2* e *Mod_Comb-2* - Washington III

Na subsérie Rio I (Figura 29), nota-se que o desempenho do *Modelo2* se deteriora sensivelmente à medida que aumenta o erro na série de entrada, e se torna nitidamente pior do que o do método univariado HWT quando o erro na previsão de temperatura ultrapassa 3%. Na subsérie Rio II (Figura 30), esta sensibilidade também é visível, embora não seja tão marcante. O gráfico das constantes α da combinação mostra um comportamento que faz sentido lógico: à medida que aumenta a incerteza na previsão da temperatura usada na entrada das redes, a combinação passa a dar mais peso à previsão do HWT, que não depende da temperatura. Nestas duas séries, a combinação de Byrd e a *naïve* resultaram em erros aproximadamente iguais, em termos medianos; a combinação *naïve*, porém, resultou em menor dispersão dos erros.

Nas três subséries da base de dados de Washington (Figuras 31 a 33), comportamentos similares também puderam ser observados. O *Modelo2* se mostrou bastante sensível, com seu MAPE aumentando à medida que cresce a incerteza na temperatura. A combinação de RNAs com um método linear tornou o *Mod_Comb* menos sensível, com menor variação na mediana e na dispersão dos MAPEs. A combinação de Byrd conseguiu melhores resultados em termos medianos do que a *naïve*, nas três subséries; a combinação *naïve*, contudo, conseguiu previsões com dispersão claramente menor.

As Figuras 34 e 35 mostram os resultados obtidos com as duas subséries do banco de dados da Eslováquia. Nestas subséries, a sensibilidade dos modelos em relação à temperatura se manifesta de forma muito diferente da que foi vista anteriormente nas séries do Rio e de Washington. O principal fator a causar esta mudança de comportamento é o fato da base da Eslováquia conter apenas a média da temperatura observada no dia e não todas as 24 temperaturas observadas a cada hora dos dias; há portanto muito menos informação sobre temperatura a ser utilizada pelos modelos.

A Figura 34 mostra que o *Modelo2* tem muita pouca sensibilidade, mantendo o MAPE mediano praticamente constante, para todos os níveis de ruído nas temperaturas na entrada. As duas formas de combinação, tanto a de Byrd quando a *naïve*, conseguem resultados praticamente equivalente, em termos da mediana dos MAPEs e de sua dispersão.

Na Figura 35 (subsérie Eslováquia 2), o *Modelo2* não mostra nenhuma sensibilidade ao erro na temperatura. Isto pode ser explicado, se analisarmos a seleção de variáveis feita pelo AG. Há apenas duas variáveis de temperatura possíveis, entre as candidatas para entrada nas redes, uma indicando a diferença da média da temperatura de hoje para amanhã e a outra para a diferença da média da temperatura de ontem para hoje. O AG, contudo, não incluiu nenhuma das duas entre as entradas usadas no *Modelo2*. Mais detalhes sobre a seleção de variáveis do método AG será vistos na Seção 3.4.

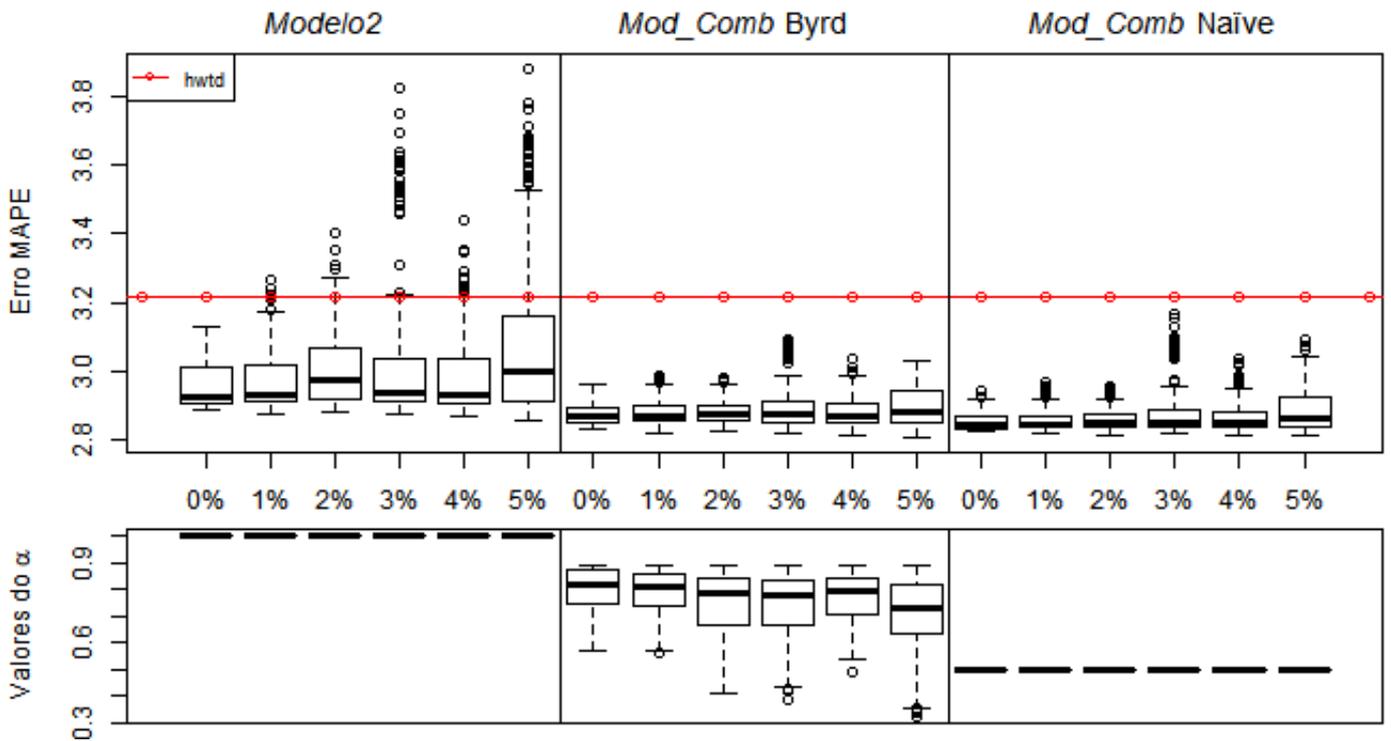


Figura 34 – Análise de sensibilidade - *Modelo2-3* e *Mod_Comb-3* - Eslováquia I

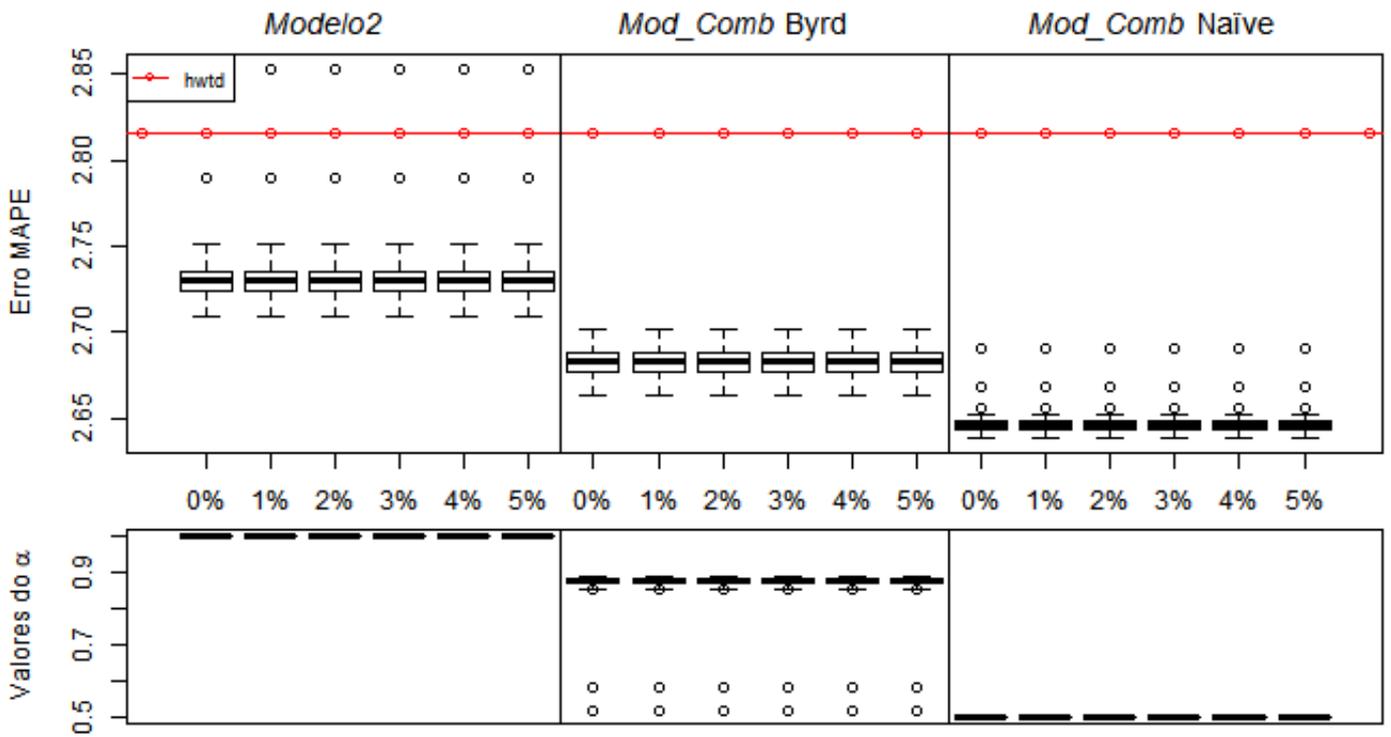


Figura 35 – Análise de sensibilidade - *Modelo2-5* e *Mod_Comb-5* - Eslováquia II

3.4 ANÁLISE DA SELEÇÃO FEITA PELO AG

As análises feitas na Seção 3.3 levantaram algumas questões sobre o comportamento dos modelos obtidos pelo AG neste trabalho. Houve uma variação considerável nos resultados obtidos dentro de uma mesma subsérie, e em alguns casos os resultados foram piores que os do *Modelo0*, utilizado como *benchmark*. Nesta Seção, analisaremos os resultados obtidos pelo AG para cada modelo e cada subsérie.

3.4.1 Seleção de variáveis de entrada relacionadas às cargas passadas

Analisaremos primeiramente a seleção das variáveis de entrada relacionadas aos valores históricos observados da carga elétrica. Nos modelos que empregamos, as variáveis de carga candidatas para inclusão nos modelos foram as cargas das 24 horas do dia em que é feita a previsão, e as cargas das 24 horas do dia anterior a este (para simplificar, consideraremos que todas as previsões são feitas no dia de hoje, e chamaremos estas cargas de cargas de *hoje* e de *ontem*).

O resultado da seleção feita pelo AG entre estas 48 candidatas pode ser visto nas Tabelas 4, 5 e 6. Estas tabelas indicam, em cada coluna, quais variáveis de carga foram selecionadas pelo AG para cada versão dos modelos testados. As colunas são agrupadas de acordo com o tipo de modelo (*Modelo1* e *Modelo2*) e com as subséries de cada base de dados usadas para teste. As linhas das tabelas contém as 24 observações das cargas de hoje e as 24 observações das cargas de ontem, e cada célula mostra se uma variável específica foi ou não selecionada pelo AG para participar do treinamento da RNA, por meio de uma variável dummy. Para facilitar a análise, adicionamos linhas intermediárias com o total de variáveis de entrada que cada RNA utilizou em seu treinamento.

A primeira observação que pode ser feita é a de que nenhum dos modelos utilizou todas as variáveis de carga disponíveis. Na subsérie Rio I, os modelos usaram de 7 a 16 das variáveis que representam as cargas de hoje (mediana = 9,5), e de 10 a 16 das cargas de ontem (mediana = 12). Na subsérie Rio II, os modelos usaram de 10 a 14 variáveis das cargas de hoje (mediana = 12), e de 10 a 16 das cargas de ontem (mediana = 12,5).

O AG faz, portanto, uma redução na dimensionalidade do espaço de entrada, reduzindo aproximadamente pela metade o número de variáveis usadas. Isto era de se esperar, uma vez que há uma forte autocorrelação na séries de carga; a informação existente na carga, por exemplo, da hora $h=13$ de um dia é bem similar à na da carga da hora $h=14$ do mesmo dia. Portanto, é possível descartar algumas destas variáveis e construir um modelo de arquitetura mais simples, que ainda assim consiga previsões com boa acurácia.

Outra observação é a de que nenhuma das variáveis foi selecionada pelo AG para inclusão em todos os modelos; todas as variáveis participaram de algum dos modelos, mas nenhuma delas foi consistentemente incluída ou excluída de todos os modelos. Al-

guns pesquisadores (por exemplo [117]) propõem modelos que selecionam alguns pontos cardeais na série de cargas, aqueles que definem de forma simplificada a forma do perfil de carga, fazem previsões para estes instantes, e depois prevêem o resto do perfil por interpolação. O resultado das rodadas do AG contudo mostra que, pelo menos nas subséries testadas, não há instantes de tempo que pareçam conter informação mais relevantes do que os outros; nem mesmo há uma tendência de inclusão das observações mais recentes do histórico de carga, em detrimento das mais antigas. Isto é um efeito lógico da forte autocorrelação existente entre as observações, que faz com que todas contenham mais ou menos as mesmas informações; como consequência, o trabalho de seleção de variáveis feito pelo AG se torna muito difícil, pois há um grande número de soluções possíveis (vetores de entrada) que levam a modelos cujas previsões têm aproximadamente a mesma acurácia.

Tabela 4 – Atributos de carga selecionados pelo AG - Rio

Hora	Rio I										Rio II									
	<i>Modelo1</i>					<i>Modelo2</i>					<i>Modelo1</i>					<i>Modelo2</i>				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
2	0	0	1	0	0	0	0	1	0	1	1	1	0	0	1	0	0	1	0	1
3	1	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	1	0	1
4	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	0
5	1	1	1	0	1	0	0	1	1	1	1	1	0	0	0	1	1	0	1	0
6	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0
7	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	1	0	1
8	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
9	0	1	1	1	1	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0
10	1	0	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1
11	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0
12	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1
13	0	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	0	1	0	0
14	0	0	0	1	0	1	1	1	1	1	0	1	0	0	1	1	1	0	1	1
15	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1
16	1	1	1	0	0	0	0	0	0	0	1	0	1	0	0	1	1	1	1	1
17	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
18	0	1	1	1	0	1	1	1	1	0	1	1	1	1	1	0	0	1	1	1
19	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	0	1	0
21	0	1	0	0	0	1	1	0	0	1	0	0	1	1	1	0	1	1	1	1
22	0	0	1	1	1	1	1	0	0	0	1	1	1	0	0	1	1	1	1	0
23	1	0	1	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1	0	0
24	0	1	1	0	1	0	0	1	0	1	1	0	0	1	1	0	1	1	1	1
Σ	10	13	16	10	10	9	9	8	7	7	12	10	12	11	13	10	14	13	12	14
1	1	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	1	1	1	0	0	0	0	0	0	1	0	1	0
3	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	0	1	0	0	0
4	1	0	1	1	0	1	1	0	0	0	0	1	1	1	1	0	0	0	0	0
5	0	1	0	0	1	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
6	0	1	0	0	0	0	0	0	0	0	1	1	0	1	0	0	1	0	1	0
7	0	0	1	0	1	1	1	1	1	1	0	1	0	1	0	0	0	0	1	0
8	1	1	0	0	1	0	0	1	1	1	0	1	1	1	0	1	0	0	0	0
9	0	0	0	0	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1
10	1	1	0	1	0	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1
11	1	0	0	1	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0
12	0	1	1	0	0	1	1	1	0	1	0	0	0	1	0	0	0	0	0	0
13	1	1	1	1	1	0	0	0	0	0	0	1	0	1	0	1	1	1	1	0
14	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	1	1	1	0	1
15	1	1	1	0	1	1	1	0	0	0	0	0	1	0	0	0	0	1	1	0
16	0	0	0	1	1	0	0	0	0	1	0	1	1	1	1	1	1	1	1	0
17	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
18	0	1	0	1	0	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1
19	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0
20	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	1	0	1
21	0	0	1	1	0	1	1	1	1	0	0	1	0	1	0	1	1	0	1	1
22	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	1	1
23	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1
24	0	0	1	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
Σ	11	12	11	10	12	12	12	16	14	16	10	16	14	16	10	12	13	12	13	10

Tabela 6 – Atributos de carga selecionados pelo AG - Eslováquia

Hora	Eslováquia I					Eslováquia II																
	<i>Modelo1</i>		<i>Modelo2</i>			<i>Modelo1</i>		<i>Modelo2</i>														
	1	2	3	4	5	1	2	3	4	5												
Cargas de Hoje	1	0	0	1	0	0	1	1	0	0	0	1	0	1	0	0	0	1	1	0		
	2	0	1	0	0	0	0	1	0	1	1	0	0	1	0	1	1	0	0	0		
	3	1	0	1	0	0	1	0	1	1	1	1	1	0	0	0	0	0	0	1	0	
	4	0	0	0	1	0	0	1	0	1	0	0	1	0	1	1	0	1	1	1	1	
	5	1	1	0	1	0	1	0	1	1	1	1	1	0	0	1	0	1	1	0	1	
	6	1	0	0	1	1	1	0	0	0	0	0	0	0	1	0	1	0	1	0	1	
	7	0	1	0	0	1	0	1	0	0	0	1	0	1	1	1	0	1	1	0	1	0
	8	1	1	1	0	0	0	1	1	1	1	1	0	1	0	1	0	1	0	1	0	
	9	0	0	0	0	0	0	0	1	1	0	1	0	1	1	0	0	0	0	0	1	0
	10	1	1	1	0	0	1	1	0	1	0	1	0	0	1	1	0	1	0	1	0	
	11	1	1	0	0	1	0	1	0	1	0	0	1	1	1	0	1	0	1	1	1	
	12	0	0	0	1	1	0	0	1	0	1	0	0	0	1	0	0	1	1	0	0	
	13	1	0	0	1	1	0	1	1	0	0	1	0	1	1	1	1	1	1	1	1	
	14	0	0	1	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	1	1	
	15	0	0	0	0	1	1	0	1	0	0	1	0	1	1	1	1	1	0	1	0	
	16	1	1	1	0	1	0	1	1	0	1	1	0	0	1	1	1	1	0	1	1	
	17	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1	1	1	
	18	0	0	0	1	1	0	0	1	0	1	1	1	1	1	0	1	1	0	1	0	
	19	1	0	0	0	0	0	1	0	1	1	0	1	1	0	1	0	0	0	0	1	
	20	0	1	1	0	0	1	0	1	1	1	1	1	0	1	1	0	1	1	1	1	
	21	0	0	0	1	0	1	1	1	1	1	1	0	0	1	1	1	1	0	1	1	
	22	1	1	0	1	1	0	0	1	0	1	0	1	0	0	1	1	1	0	1	0	
	23	1	1	0	0	1	0	1	1	1	0	1	0	1	1	0	0	0	0	1	0	
	24	1	1	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	1	1	1	
	Σ	13	12	8	9	11	10	15	14	14	12	13	11	12	17	14	11	15	9	19	12	
Cargas de Ontem	1	1	1	1	0	0	0	1	1	1	1	0	0	1	0	0	0	1	0	0	1	
	2	0	1	0	1	0	1	0	0	0	1	1	1	0	1	0	1	1	1	1	0	
	3	0	0	1	0	1	0	1	0	0	0	0	0	0	0	1	1	0	1	1		
	4	1	0	0	0	0	0	0	1	1	0	0	0	0	1	1	1	0	0	1		
	5	1	1	1	0	1	0	0	0	0	1	0	1	1	1	1	1	0	0	1		
	6	1	1	1	0	1	0	1	1	1	0	0	1	1	0	0	1	0	1	0	0	
	7	0	0	0	0	0	0	1	0	1	0	1	0	0	1	0	1	0	0	0		
	8	1	1	0	0	0	1	1	0	0	0	0	1	1	1	0	1	0	0	0	1	
	9	1	0	0	0	1	0	1	1	1	1	1	0	0	0	1	1	1	1	0	1	
	10	1	1	0	0	0	1	0	0	0	1	1	0	1	0	0	0	1	1	0	0	
	11	1	0	1	0	0	0	0	1	1	0	0	1	0	0	0	1	0	1	0	0	
	12	0	0	0	1	1	0	1	1	1	0	1	0	0	0	1	0	0	1	1	1	
	13	0	1	1	1	1	1	1	1	0	0	1	1	0	1	0	1	0	0	1	0	
	14	0	1	1	0	0	0	1	0	0	1	0	1	0	1	0	1	0	1	1	1	
	15	0	0	1	1	1	1	0	1	1	1	0	0	1	0	1	1	0	1	1	0	
	16	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	1	1	0	
	17	1	0	0	1	0	0	1	0	1	1	1	0	0	1	0	0	1	1	0	1	
	18	1	1	0	1	1	0	1	0	1	1	1	0	1	0	0	0	0	0	1	0	
	19	0	1	1	1	1	1	1	0	1	1	1	1	0	0	1	1	0	1	1	1	
	20	1	0	1	0	0	0	0	0	1	0	1	0	0	1	1	0	1	1	0	0	
	21	0	0	0	0	1	1	0	1	1	1	0	1	1	0	0	1	1	1	1	1	
	22	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	1	1	1	1	0	
	23	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	
	24	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
	Σ	12	11	13	8	12	9	14	11	16	13	13	13	10	9	11	17	15	16	13	13	

3.4.2 *Seleção de variáveis de entrada relacionadas às temperaturas*

Uma análise semelhante à da Seção anterior pode ser feita nos resultados da seleção das variáveis relacionadas à temperatura, mostrado nas Tabelas 7, 8 e 9. De forma similar ao que foi feito nas tabelas anteriores, estas tabelas indicam, em cada coluna, quais variáveis de diferença de temperatura foram selecionadas pelo AG para cada versão dos modelos testados. As colunas são agrupadas da mesma forma utilizada nas tabelas vistas anteriormente.

Novamente, verifica-se que nenhum dos modelos utilizou todas as variáveis disponíveis. Nos dados do Rio (Tabela 7), observamos que os modelos na subsérie Rio I usaram entre 8 a 14 das variáveis que representam as diferenças de temperatura de ontem para hoje (mediana = 10,5), e entre 9 a 15 das diferenças de temperatura de hoje para amanhã (mediana = 12,5). Na subsérie Rio II, por outro lado, os modelos usaram menos informação relacionada à temperatura; os vetores de entrada continham entre 2 a 9 (mediana = 5) variáveis representando as diferenças de temperatura de ontem para hoje, e entre 7 a 12 (mediana = 11) variáveis representando as diferenças de temperatura de hoje para amanhã.

Nos dados de Washington (Tabela 8), o comportamento foi similar. Nenhuma variável foi incluída em todas os modelos, e total de variáveis selecionadas não mostrou nenhum padrão identificável. O número de variáveis de diferença de temperatura nas entradas, entre os seis modelos considerados, variou entre 5 e 16 (com medianas entre 10 e 12,5).

Nos dados da Eslováquia (Tabela 9), há apenas temperaturas médias diárias, em vez de perfis horários, e portanto apenas duas variáveis de diferenças de temperatura. A variável que indica a diferença de temperatura média de hoje para amanhã foi selecionada em todos os modelos na subsérie Eslováquia I; na subsérie Eslováquia II, contudo, em apenas 3 dos 10 modelos. A variável que indica a diferença de temperatura média de ontem para hoje foi incluída em 9 dos 20 modelos usados nas duas subséries. Note-se que em quatro dos modelos na subsérie Eslováquia II nenhuma variável relacionada à temperatura foi selecionada; estes modelos são, em essência, univariados, uma vez que as rodadas de AG indicaram que a informação contida na temperatura não justificava a sua inclusão nos modelos (daí os resultados mostrados na Figura 35, que mostram que os modelos não têm nenhuma sensibilidade à temperatura).

Tabela 8 – Atributos de temperatura selecionados pelo AG - Washington

Hora	Washington I					Washington II					Washington III																					
	Modelo1		Modelo2			Modelo1		Modelo2			Modelo1		Modelo2																			
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5												
Diferença de temperatura de ontem para hoje	1	0	0	1	1	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	0	1	0	0	0	0	0			
	2	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	0	1	1	0	1	0	0	1	0	0	0		
	3	1	1	1	0	1	1	0	0	0	0	1	0	0	1	0	0	0	1	0	0	1	0	1	0	1	0	1	1	1		
	4	0	0	0	0	1	0	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0		
	5	1	1	1	0	0	0	1	1	0	1	0	1	1	1	1	0	1	0	1	0	1	0	1	1	0	1	0	0	0		
	6	1	1	1	1	1	1	1	1	0	0	1	0	0	1	1	0	1	0	0	1	0	0	1	0	0	1	1	0	0		
	7	0	1	1	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	1	0	1	0	1	0	1	0	1	0	0		
	8	1	0	1	1	0	1	1	1	0	1	1	0	1	1	0	0	1	1	0	0	1	0	1	0	1	0	1	0	1		
	9	1	0	0	1	0	1	0	1	1	1	1	0	0	0	1	0	0	1	0	0	1	0	1	0	0	0	1	1	0		
	10	1	1	1	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0	1	0	0	0	0	0	0	1	1	0	
	11	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	
	12	0	0	1	0	1	0	1	1	0	0	0	0	1	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	
	13	0	0	0	0	1	1	0	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	1	1	1	1	1	0	0	
	14	1	1	0	0	0	1	0	0	0	1	1	1	0	1	1	0	1	1	0	0	0	0	1	1	0	0	0	1	1	1	
	15	0	0	1	1	1	1	1	0	0	0	0	1	0	1	0	0	0	1	1	0	0	1	1	0	1	1	1	1	1	1	
	16	0	1	1	1	0	0	0	0	1	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	1	1	0	0	1	1	
	17	1	1	0	1	0	0	1	1	0	1	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	1	0	1	1	0	
	18	1	0	0	0	1	0	0	0	0	0	1	1	1	0	0	0	0	1	0	1	1	1	1	1	1	0	1	0	0	1	0
	19	1	1	0	1	0	1	0	0	0	0	1	1	1	0	0	0	0	0	1	0	1	0	1	0	1	0	0	0	0	1	
	20	1	0	1	0	0	0	1	0	0	1	1	1	1	0	0	0	0	0	1	0	0	0	0	1	1	1	0	1	0	0	
	21	0	1	1	1	1	1	0	0	0	1	0	1	0	1	1	1	0	1	0	0	0	0	1	0	1	0	1	1	1	1	
	22	0	0	0	0	1	0	0	0	1	0	0	1	1	1	0	0	0	0	0	1	0	0	0	1	0	0	1	1	1	1	
	23	0	0	1	0	1	0	0	1	0	0	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	0	
	24	0	0	1	1	1	1	1	0	0	1	1	0	1	1	1	1	0	0	1	0	0	0	1	0	0	0	0	1	0	0	
	Σ	12	11	14	10	12	11	9	10	5	12	13	12	13	16	7	6	9	10	10	12	9	10	15	13	9	10	9	13	15	9	
Diferença de temperatura de hoje para amanhã	1	0	1	0	0	0	1	0	0	0	0	1	1	1	1	1	0	0	0	0	1	1	0	1	1	1	0	0	0	1		
	2	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	1	1	1	1		
	3	0	0	1	1	1	0	0	1	0	1	0	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0	1		
	4	0	1	1	1	0	0	1	0	1	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	
	5	1	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	1	1	0	1	1	0	1	0		
	6	0	1	0	0	1	0	1	1	1	0	0	1	0	0	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1		
	7	0	1	0	0	0	1	1	0	1	1	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0	1	1	1	1		
	8	1	0	1	1	1	0	1	0	0	0	0	0	0	1	1	0	1	0	1	0	1	0	1	0	1	1	0	1	1		
	9	0	1	0	0	0	0	1	1	1	0	0	1	0	0	0	1	0	1	1	1	1	0	0	1	1	0	0	1	0		
	10	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	1	0	0	1	0	0	0	0	0	0	1		
	11	1	0	1	0	0	0	1	1	0	1	0	0	1	0	0	1	1	1	1	0	0	1	0	0	1	0	1	0	0		
	12	0	0	0	1	1	1	0	1	1	1	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	1	1	1	1	
	13	1	1	1	0	1	1	1	0	1	0	0	1	0	1	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0	
	14	1	0	1	1	0	0	0	1	0	1	0	1	0	1	1	0	1	0	0	1	1	1	1	1	0	1	0	1	0	0	
	15	0	1	0	1	1	1	1	1	1	0	1	0	0	0	1	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	
	16	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
	17	0	1	1	0	1	1	0	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	
	18	0	1	1	0	1	0	1	0	0	0	1	0	0	1	0	0	0	1	1	0	0	0	1	1	0	1	1	1	0	1	
	19	0	1	0	1	0	1	1	0	0	0	0	1	0	1	0	1	1	1	0	0	0	0	1	0	1	0	1	1	0	0	
	20	1	1	1	1	1	0	0	0	1	1	0	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1	1	
	21	1	0	1	1	0	1	1	0	0	1	1	0	1	1	0	0	0	1	1	0	0	1	1	0	0	1	0	0	1	0	
	22	1	1	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0	1	0	1	1	1	0	0	1	0	1	0	0	
	23	0	0	1	1	0	1	1	1	1	1	1	0	1	1	0	0	1	0	1	0	1	1	1	0	0	1	1	1	1	0	
	24	1	0	0	0	1	1	1	0	0	1	0	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	
	Σ	10	12	14	10	10	10	14	10	11	12	11	13	11	11	15	15	12	12	14	15	11	11	15	11	11	14	14	11	15	14	

Tabela 9 – Atributos de temperatura selecionados pelo AG - Eslováquia

Hora	Eslováquia I					Eslováquia II														
	<i>Modelo1</i>					<i>Modelo2</i>														
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5					
Diferença da temperatura média de ontem para hoje	1	0	0	0	1	0	1	1	0	0	0	0	1	0	1	1	1	1	0	0
Diferença da temperatura média de hoje para amanhã	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	0	0	0	0

3.4.3 Seleção de variáveis indicadoras do dia da semana

Na seleção das variáveis de entrada nas RNAs, o AG teve também que escolher entre os dummies usados para informar às RNAs o dia da semana a partir do qual seria feita a previsão. A Tabela 10 mostra o resultado dessa seleção, nas base de dados do Rio. Usamos uma codificação 1-of-(c-1) [104]; ou seja, os sete dias da semana foram representados por meio de seis variáveis dummy; o domingo é indicado quando as todas os dummies tiverem valores nulos. Uma vez que os dias da semana apresentam todos um perfil de cargas bastante semelhante (mas claramente distinto dos perfis observados nos sábados e nos domingos), poderíamos esperar que poucos dummies fossem necessários na entrada das RNAs; no caso extremos, poderia ser usado apenas um dummy que indicasse se o dia a ser previsto é um dia de semana ou um de fim-de-semana. No entanto, isto não aconteceu, e a maioria dos modelos usou 4 ou mais dummies, e 11 dos 20 modelos usaram todos os dummies disponíveis (a única exceção ocorreu no *Modelo1* da subsérie Rio I, que usou apenas dois dummies). O dummy menos usado foi o que representa os sábados (incluído em 16 dos 20 modelos); os mais usados foram os que representam as segundas-feiras e as quartas-feiras (em 19 de 20 modelos).

A Tabela 11 mostra o resultado da seleção, nas base de dados de Washington. Foi usada desta vez uma codificação em dummies 1-of-c [104], em que cada um dia da semana foi representado por um dummy. Novamente, todos os modelos usaram um grande número de dummies, entre 4 e 7; a exceção foi o *Modelo2* na subsérie Washington 1, que usou apenas três. Apenas dois dos modelos usaram todos os sete dummies disponíveis. O dummy mais frequentemente empregado foi o que representa as sextas-feiras (em 29 dos 30 modelos); o menos empregado foi o que representa as terças-feiras (em 11 dos 30 modelos).

Por fim, a Tabela 12 mostra os resultados na base de dados da Eslováquia. Em nenhum dos modelos o AG selecionou todos os dummies, o que corrobora a idéia de uma codificação 1-of-(c-1), com 6 dummies, poderia ter sido usada sem perda de acurácia. Em 11 dos 20 modelos, o AG selecionou seis dos dummies, o que é suficiente para indicar o dia exato da semana; nenhum dos modelos usou menos de 4 dummies.

Tabela 10 – Atributos para os dias de semana selecionados pelo AG - Rio

Dia	Rio I										Rio II									
	Modelo1					Modelo2					Modelo1					Modelo2				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Seg.	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Ter.	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
Qua.	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Qui.	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Sex.	1	1	0	1	0	1	1	1	1	1	1	0	1	1	0	1	1	1	1	1
Sab.	1	1	1	1	1	1	1	0	0	1	1	1	1	0	1	1	1	1	1	1
Σ	2	6	4	6	4	6	6	5	5	6	5	5	6	5	5	6	6	6	6	6

Tabela 11 – Atributos para os dias de semana selecionados pelo AG - Washington

Dia	Washington I										Washington II										Washington III									
	Modelo1					Modelo2					Modelo1					Modelo2					Modelo1					Modelo2				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Seg.	0	1	1	1	0	1	0	1	1	1	0	1	0	1	1	1	1	1	0	0	1	1	0	0	1	1	1	1	0	1
Ter.	0	0	0	0	1	0	0	0	0	1	1	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	1	0	1
Qua.	0	0	0	0	1	0	0	0	1	0	1	0	0	1	1	1	0	1	1	1	1	0	1	0	0	0	1	1	1	1
Qui.	1	0	1	0	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Sex.	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Sab.	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
Dom.	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	0	1	1	1	0	1	0	1
Σ	4	3	5	4	6	5	4	4	5	7	5	4	4	5	6	6	6	6	5	6	6	4	5	4	5	6	5	6	4	7

Tabela 12 – Atributos para os dias de semana selecionados pelo AG - Eslováquia

Dia	Eslováquia I										Eslováquia II									
	Modelo1					Modelo2					Modelo1					Modelo2				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Seg.	1	1	0	0	1	1	0	1	1	1	1	1	1	0	1	1	1	0	0	1
Ter.	1	1	1	1	1	1	0	1	1	0	0	0	1	1	0	1	0	1	1	0
Qua.	0	1	0	0	0	0	1	0	1	1	1	0	1	1	0	1	1	1	1	1
Qui.	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Sex.	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
Sab.	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1
Dom.	0	0	0	0	1	1	0	1	1	0	1	1	1	1	0	0	1	1	1	1
Σ	4	6	4	4	6	6	4	6	6	5	6	5	6	6	4	6	4	6	5	6

3.4.4 Seleção de número de neurônios e de épocas de treinamento

Além de fazer uma seleção binária, definindo se uma possível variável de entrada participa ou não do treinamento das RNAs, o AG também foi usado para definir o número de neurônios internos na camada oculta da cada RNA, e o número de épocas máximo para o treinamento.

As Tabelas 13, 14 e 15 mostram os resultados obtidos para as três bases de dados. Em cada linha, estão indicados o número de neurônios ocultos, e o número de épocas de treinamento, para cada um das cinco versões de cada modelo criadas para cada subsérie. Como o *Modelo1* tem 24 nós de saída (para prever o perfil de carga horária), e *Modelo2* tem apenas um (para prever a carga numa hora específica do dia), estes dois modelos têm número de neurônios ocultos bastante diferentes; é de se esperar que as RNAs do *Modelo1* sejam mais complexas, com maior número de neurônios. Arquiteturas com menos de 10 neurônios, de fato, foram vistas geralmente apenas no *Modelo2*, e só em duas versões do *Modelo1*.

A Tabela 13 mostra os resultados para os dados do Rio. Nota-se que as redes do *Modelo1* tiveram em geral entre 20 a 30 neurônios, e foram treinadas com um número de épocas variando de 10 a 14. As redes do *Modelo2*, contudo, tiveram um número muito variado de neurônios. Na subsérie Rio I, algumas tinha apenas um neurônio oculto, enquanto outras chegavam a ter 31. A terceira coluna da tabela mostra os MAPEs médios obtidos por cada modelo.

A Tabela 14 mostra os resultados para os dados de Washington. As redes do *Modelo1* tiveram em geral entre 10 a 32 neurônios (mediana = 20), e foram treinadas com um número de épocas variando de 11 a 31 (mediana = 21). As redes do *Modelo2*, novamente, tiveram em geral arquitetura mais simples. A maioria delas tiveram 11 ou menos neurônios; apenas uma teve 27. O número de épocas de treinamento variou entre 19 e 30. A terceira coluna da tabela mostra os MAPEs médios obtidos por cada modelo.

Por fim, a Tabela 15 mostra os resultados para os dados da Eslováquia. Esperaríamos que estas redes tivessem arquitetura um pouco mais simples, já que usam menos variáveis de entrada (já que há apenas duas variáveis relacionadas à temperatura); isto, contudo, não ocorreu. As redes do *Modelo1* tiveram, com apenas uma exceção, entre 21 a 31 neurônios (mediana = 27), e foram treinadas com um número de épocas variando de 10 a 16 (mediana = 13). As redes do *Modelo2* se revelaram mais complexas do que as usadas nas outras bases de dados, e tiveram em geral entre 9 e 32 neurônios (mediana = 16); apenas uma teve um único neurônio. O número de épocas de treinamento variou entre 12 e 27 (mediana = 20). A terceira coluna da tabela mostra os MAPEs médios obtidos por cada modelo.

O número de neurônios varia claramente dentro de um mesmo modelo e subsérie,

Tabela 13 – Configurações da RNA selecionados pelo AG - Rio

		Nº de Neurônios	Nº de Épocas	MAPE médio	
Rio I	<i>Modelo1</i>	1	30	14	2.25
		2	8	11	2.43
		3	30	11	2.46
		4	26	12	2.60
		5	25	11	2.39
	<i>Modelo2</i>	1	31	11	2.26
		2	31	11	2.13
		3	1	27	2.10
		4	1	29	2.51
		5	1	16	2.24
Rio II	<i>Modelo1</i>	1	27	12	2.15
		2	29	12	2.15
		3	21	11	2.13
		4	29	10	2.26
		5	26	13	2.18
	<i>Modelo2</i>	1	5	20	2.05
		2	17	12	2.13
		3	24	12	2.02
		4	25	10	1.98
		5	22	13	2.13

indicando uma dificuldade maior para o AG. O mesmo acontece com o número de épocas que mesmo sempre acima de 10 ainda varia muito.

A seleção feita pelo AG fez grande diferença nos resultados de previsão das bases de dados de Washington e da Eslováquia, comparados aos *Modelo0*. Logo é importante a seleção das variáveis de entrada e a definição do número de neurônios e número de épocas de treinamento da RNA.

Tudo leva para um maior estudo para essa etapa do trabalho. Um processo que leve a boas previsões e ao mesmo tempo próximas seria uma grande melhoria para os resultados.

Tabela 14 – Configurações da RNA selecionados pelo AG - Washington

		Nº de Neurônios	Nº de Épocas	MAPE médio	
Washington I	<i>Modelo1</i>	1	31	21	2.35
		2	10	26	2.47
		3	32	21	2.34
		4	27	26	2.44
		5	21	21	2.40
	<i>Modelo2</i>	1	7	29	2.28
		2	9	23	2.26
		3	27	23	2.50
		4	8	29	2.23
		5	11	19	2.34
Washington II	<i>Modelo1</i>	1	11	19	2.31
		2	10	31	2.39
		3	16	24	2.31
		4	29	11	2.54
		5	14	22	2.26
	<i>Modelo2</i>	1	3	27	2.16
		2	3	23	2.21
		3	3	30	2.19
		4	11	24	2.28
		5	7	23	2.19
Washington III	<i>Modelo1</i>	1	28	18	2.23
		2	20	16	2.32
		3	20	19	2.32
		4	24	18	2.23
		5	17	24	2.31
	<i>Modelo2</i>	1	4	29	2.17
		2	3	29	2.18
		3	8	27	2.23
		4	9	24	2.22
		5	4	27	2.18

Tabela 15 – Configurações da RNA selecionados pelo AG - Eslováquia

		Nº de Neurônios	Nº de Épocas	MAPE médio	
Eslováquia I	<i>Modelo1</i>	1	3	16	3.50
		2	30	13	3.15
		3	28	14	3.05
		4	31	10	3.24
		5	21	13	3.19
	<i>Modelo2</i>	1	1	24	3.11
		2	9	18	3.10
		3	15	12	3.01
		4	18	20	3.15
		5	32	27	3.10
Eslováquia II	<i>Modelo1</i>	1	26	16	2.90
		2	24	12	2.92
		3	30	10	2.86
		4	31	13	2.98
		5	23	10	2.88
	<i>Modelo2</i>	1	11	20	2.82
		2	19	24	2.91
		3	21	24	2.79
		4	10	17	3.00
		5	17	13	2.76

4 CONCLUSÃO

Estudamos neste trabalho modelos de previsão de cargas elétricas a curto prazo baseados na combinação entre RNAs e métodos lineares.

Em princípio, RNAs pareceriam ser as ferramentas ideais para este tipo de previsão, uma vez que são modelos extremamente flexíveis, capazes de modelar com precisão a relação linear entre a carga e a temperatura. Esta mesma flexibilidade, porém, faz com que as RNAs sejam sensíveis ao ruído presente nos dados de entrada, o que pode inviabilizar seu uso, se este ruído for demasiado.

Para avaliar esta sensibilidade, no caso do problema de previsão de cargas, analisamos o desempenho de um modelo constituído por um conjunto de RNAs em paralelo, quando diferentes níveis de ruído simulado foram adicionados às séries de temperatura (a variável exógena considerada no modelo). Verificamos que as RNAs são demasiadamente sensíveis a este erro, o que faz com que modelos de previsão baseados exclusivamente nelas tenham pouco aplicação prática.

Para lidar com esta sensibilidade, propusemos um modelo baseado na combinação linear entre um conjunto de 24 RNAs em paralelo e o método HWT de amortecimento exponencial univariado. Este modelo foi testado em sete amostras de teste, extraídas de três séries de carga reais, e mostrou excelente desempenho preditivo, quando comparado a diversos *benchmarks*, numa ampla gama de situações. O resultado mais importante, contudo, foi obtido a partir da análise da sensibilidade deste modelo. Observamos que o peso da combinação entre RNAs e HWT reage ao nível de erro presente nas séries de temperaturas. Quando estas séries tem pouco erro, as RNAs recebem maior peso, e são em consequência capazes de modelar com precisão a relação não-linear entre a temperatura e a carga; quando as séries têm alto nível de erro, porém, a combinação dá maior peso ao método HWT, que independe da temperatura. A combinação, em consequência, descarta os dados de temperatura, quando estes são ruidosos, e faz a previsão com base apenas nos dados passados da carga.

Este modelo combinado, portanto, parece mais adequado do que os modelos baseados unicamente em RNAs para previsões em aplicações reais, uma vez que estas sempre são feitas a partir de séries de temperaturas previstas, que trazem embutidas em si um alto nível de erro. Como extensão a este trabalho, acreditamos que seria interessante uma análise mais detalhada dos erros de previsão encontrados pelo sistema combinado; em particular, a identificação dos erros discrepantes, procurando identificar os fatores que levam à ocorrência de tais erros, e as medidas que podem vir a ser tomadas para evitá-los.

Para trabalhos futuros, podemos sugerir:

1. Avaliar o resultado do uso do AG na busca por modelos que utilizem 24 RNAs de

arquiteturas distintas, uma para cada hora do dia, cada uma tendo suas variáveis de entrada, número de neurônios e de épocas de treinamento;

2. Avaliar outras formas de ajuste do parâmetro alfa da combinação; por exemplo, avaliar um método dinâmico que utilize os erros mais recentes para o ajuste do parâmetro, e melhore assim a acurácia das previsões.
3. Melhorar a utilização do AG, separando a seleção das variáveis de entrada da definição da arquitetura (número de neurônios ocultos) e da treinamento (número de épocas), a fim de tornar os resultados das buscas mais estáveis, com menos variação entre as diferentes rodadas.

5 ANEXOS

5.1 REDES NEURAIAS ARTIFICIAIS (RNA)

A RNA é um produto do paradigma do aprendizado supervisionado. Apesar de existirem métodos no estudo de RNA que utilizam de aprendizado não supervisionado este não será utilizado em nosso trabalho.

O aprendizado supervisionado consiste de um processo de indução que procura inferir hipóteses a partir de um conjunto de dados. Torna-se importante o estudo de teorias de generalização no sentido de estabelecer a hipótese ou modelo que seja mais adequado ao problema. Primeiramente, vamos abordar de forma geral o problema de aprendizado supervisionado, em seguida introduziremos a principal teoria de generalização que utiliza o conceito de aprendizagem estatística de Vapnik e Chervonenkis [118].

Procuramos o desenvolvimento de técnicas e algoritmos no sentido de solucionar problemas de identificação, classificação, predição, estimativa e controle de sistemas adaptativos e paramétricos através de um processo contínuo de treinamento, considerando a existência de um conjunto de dados ou informações do ambiente. Este processo cíclico de retroalimentação possibilita o ajuste progressivo dos parâmetros até que um resultado ótimo seja alcançado.

É importante destacar que o desenvolvimento de modelos adaptativos, ver Figura 36, segundo Príncipe, Euliano e Lefebvre [96], é um processo construtivo, a exemplo dos projetos de engenharia, possuindo um conjunto de princípios biológicos, físicos e matemáticos que justificam o seu funcionamento.

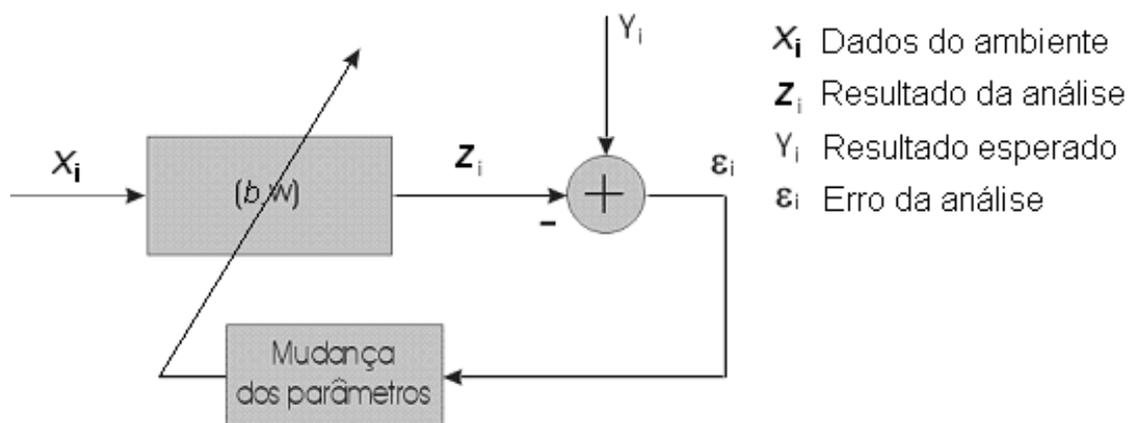


Figura 36 – Sistemas adaptativos

Embora muitos problemas desta natureza possuam uma modelagem analítica através de expressões algébricas, equações diferenciais ou sistemas discretos, procuramos uma alternativa de solução mais parecida com a forma humana de atuar. Certamente, a in-

teligência humana não é capaz de realizar cálculos matemáticos complexos. Entretanto, a sua habilidade em interagir com sistemas adaptativos de alta complexidade é notória e fruto, talvez, de milhões de anos de evolução em contato com a natureza.

Neste sentido, temos um problema fundamental, ou seja, temos que encontrar a forma mais eficiente de se utilizar o conhecimento ou a habilidade humana no desenvolvimento de modelos adaptativos.

A princípio, destacam-se duas categorias principais de sistemas. Na primeira abordagem estão os sistemas de arquitetura “cima-baixo” (*top-down*), onde o conhecimento humano é inserido na forma de heurísticas ou regras de julgamento em modelos simbólicos e declarativos, característicos das técnicas cognitivas de Inteligência Artificial. Apesar deste tipo de arquitetura ter sucesso na solução de inúmeras tarefas de altíssima complexidade, como jogos, planejamento, otimização combinatória e processamento da informação, a mesma é deficiente na incorporação de capacidades mentais como o aprendizado, a discriminação, a categorização, a generalização e a memorização, ingredientes fundamentais em qualquer sistema inteligente.

Na segunda abordagem, destacam-se os sistemas de arquitetura “baixo-cima” (*bottom-up*), provenientes dos modelos conexionistas de Inteligência Artificial, nos quais o conhecimento e a habilidade humana são conquistados ou aprendidos a partir da utilização de um conjunto de dados experimentais, a partir da interação com o ambiente e, também, a partir de um conjunto de paradigmas de aprendizado que vão direcionar a forma como os dados do modelo serão ajustados.

É nesse conjunto de sistemas que iremos destacar alguns de grande importância. Nosso problema resulta em um ajuste de função, portanto iremos estudar o modelo de RNA conhecido como Adaline, onde temos apenas um neurônio linear, e o modelo utilizado no Perceptron de Multi-Camadas, ou MLP, que adiciona ao modelo Adaline uma camada interna com neurônios não-lineares.

5.1.1 ADALINE

O modelo de rede neural artificial conhecido como Adaline [119], veio do trabalho no algoritmo *Least Mean Square* (LMS) [119]. Seu nome ADALINE é devido ao acrônimo – *ADaptive LINear Element* ou (*ADaptive Linear NEuron*).

Nele temos apenas um neurônio e este é linear. Ou seja, ele executa um ajuste de função totalmente linear. A Figura 37 mostra uma representação desse modelo. Nela temos o vetor X que representa o dado de entrada, o vetor W que representa os pesos do modelo, o scalar B que representa o bias e o scalar Z que representa o resultado obtido pelo modelo.

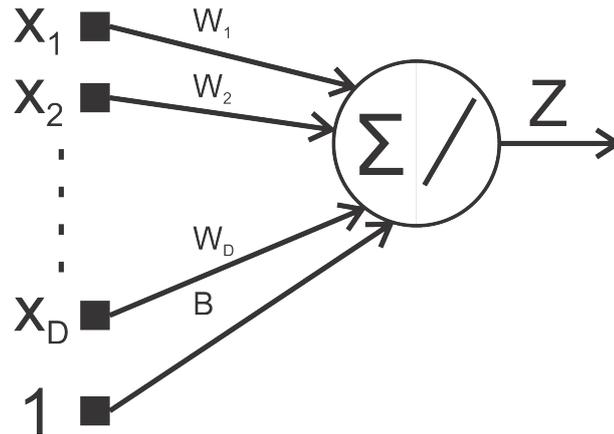


Figura 37 – Modelo Adaline

5.1.2 Perceptron de múltiplas camadas (MLP)

O Perceptron é um algoritmo proposto por Rosenblatt [120] que tem como objetivo o reconhecimento de padrões utilizado na classificação dos dados. Sua topologia é muito parecida com o Adaline, porém seu neurônio utiliza uma função não-linear. Foi considerado o primeiro algoritmo de aprendizado relacionado a modelos não-lineares.

Minsky e Papert [121], afirmaram que o poder de aproximação do Perceptron somente seria aumentado se fossem incluídas novas camadas de processamento.

Um Perceptron de multicamadas, ou simplesmente MLP, com uma camada interna, tem sua topologia definida pela Figura 38:

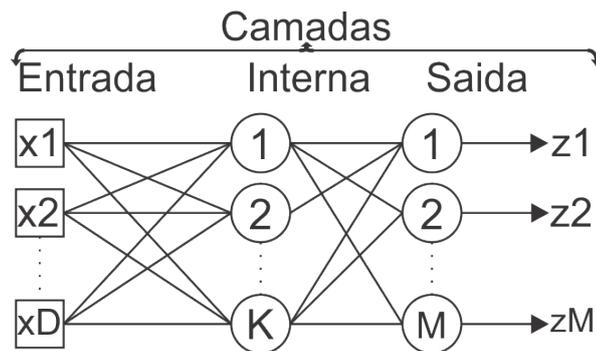


Figura 38 – Topologia do Perceptron MLP

Existem alguns resultados interessantes relacionados ao poder de computação das redes MLP, principalmente quando relacionadas ao problema de aproximação de funções.

O primeiro se refere ao fato de que o poder computacional de um Perceptron de múltiplas camadas somente é aumentado se as funções de ativação dos elementos processadores forem não-lineares. De fato, se considerarmos um Perceptron de uma camada

interna, ou de duas camadas, como uma composição de funções, ou seja:

$$f(\sum f(\sum(\cdot))) = f(\sum f(\sum_j(w_{ij}x_j + b_i))) = f(\sum f(net_i)) = f(\sum_j w_{ij}net_j + b_i) \quad (5.1)$$

podemos reduzi-lo a um Perceptron de uma única camada de saída se a função de ativação f for uma função linear.

O segundo se refere ao fato de que um Perceptron de uma camada interna produz uma região de classificação equivalente à existência de várias regiões convexas abertas ou fechadas, sendo cada uma formada pela interseção de semiespaços gerados pelos vários hiperplanos definidos pela camada anterior. Neste sentido, o mapeamento de funções na sua forma mais geral, definidas por regiões côncavas, somente é possível em Perceptrons de duas camadas, nos quais a camada de saída será responsável pela combinação de regiões convexas, gerando regiões de classificação côncavas ou não necessariamente convexas.

O terceiro se refere a um teorema derivado do trabalho de Kolmogorov [122]. Este teorema assegura que qualquer função contínua de várias variáveis pode ser representada ou aproximada por um pequeno número de funções de uma variável. Em uma rede MLP isto significa que uma função contínua de dimensão d , que mapeia valores reais pode ser computada por uma rede de três camadas com d unidades na camada de entrada, $d.(2d + 1)$ unidades na primeira camada interna, $(2d + 1)$ unidades na segunda camada interna e uma unidade na camada de saída.

O quarto resultado se refere ao poder de classificação das redes de saída sigmóide. Considerando a capacidade de mapeamento ou aproximação universal de uma rede MLP, e a utilização de uma função do tipo logística como função de ativação, teremos um classificador Bayesiano.

5.1.3 Aproximação de funções

A aproximação de funções tem como base o teorema da projeção linear, o qual garante que uma função pode ser aproximada por uma combinação linear de uma base de funções elementares ψ_i , associadas a um conjunto de parâmetros, ou seja:

$$\hat{f}(x, a) = \sum a_i \psi_i(x), \text{ para um erro } |f(x) - \hat{f}(x, a)| < \xi \quad (5.2)$$

Karl Weierstrass (1815-1897) provou que polinômios de alguma ordem podem aproximar funções contínuas em determinado intervalo. Formalmente, podemos citar o teorema de Weierstrass como:

“Seja $S[a, b]$ o espaço de funções reais e contínuas definidas no intervalo $[a, b]$. Se $f \in S[a, b]$ então existe uma aproximação polinomial para $f(x), x \in [a, b]$ na forma:

$$P(x) = \sum_i \alpha_i x_i^i, \text{ para } i = 0, \dots, n \quad (5.3)$$

para um conjunto de coeficientes α_i reais tal que $|f(x) - P(x)| < \epsilon$, para $\epsilon > 0$.”

Este teorema foi estendido para a classe de funções sigmóides por Funahashi [123], Cybenko [124] e, posteriormente, para as funções gaussianas, garantindo a ambos os modelos a capacidade de aproximadores universais.

A escolha das funções elementares, considerando a sua forma e quantidade, no sentido de formar uma base de aproximação, deve atender, a princípio, a condição de que as mesmas sejam linearmente independentes, ou seja:

$$\alpha_1 \psi_1 + \alpha_2 \psi_2 + \dots + \alpha_n \psi_n \text{ se e somente se } \alpha_1 = \alpha_2 = \dots = \alpha_n = 0 \quad (5.4)$$

Outra condição imposta ao conjunto de funções elementares, é que a base seja ortonormal, ou seja:

$$\int_a^b \psi_i(x) \psi_j(x) dx = \delta_{i,j} \text{ onde } \delta \text{ representa a função Kronecker} \quad (5.5)$$

Isto significa que a projeção ortogonal de um elemento da base em outro elemento da base fornecerá sempre o valor zero, fornecendo um único conjunto de multiplicadores α_i , como solução da aproximação ou projeção da função $f(x)$.

Neste sentido, devemos avaliar o conjunto de multiplicadores através da solução do sistema representado pela Equação (5.6):

$$\alpha_i = \langle f(x), \psi_i(y) \rangle \quad (5.6)$$

Pelo fato da base ser ortonormal, temos:

$$f(x) = \sum_i \alpha_i \psi_i(x) \quad (5.7)$$

Tomando o produto interno entre $f(x)$ e $\psi_1(x)$, computamos:

$$\langle f(x), \psi_1 \rangle = \sum_i \alpha_i \langle \psi_i(x), \psi_1(x) \rangle = \alpha_1 \langle \psi_1, \psi_1 \rangle \quad (5.8)$$

fornecendo:

$$\alpha_1 = \langle f(x), \psi_1(x) \rangle \quad (5.9)$$

De forma geral, para a avaliação de todos multiplicadores, temos:

$$\alpha_i = \langle f(x), \psi_i(x) \rangle, \text{ para a aproximação discreta e}$$

$$\alpha_i = \int f(x)\psi_i(x)dx, \text{ para a aproximação contínua.}$$

5.1.4 MLP e ajuste de função

Podemos considerar o modelo Perceptron, com uma camada interna com funções de ativação sigmóides e um combinador linear na camada de saída, como uma base de funções elementares, conforme a Figura 39. Na Figura 39 temos o vetor X que representa os dados de entrada, o vetor W que representa os pesos do modelo, o vetor B que representa os valores dos bias do modelo e o scalar Z que representa o valor de saída do modelo. Neste caso, cada função elementar tem a forma de uma função logística, cujo parâmetro representa uma combinação linear do vetor x de entrada acrescido de um bias/viés, ou seja, definimos:

$$\psi_i(x) = \frac{1}{(1 + \exp(-net_i))} \quad (5.10)$$

Onde:

$$net_i = \sum_j w_{i,j}x_j + b_i \quad (5.11)$$

Determinando, como saída do modelo, o valor correspondente ao valor da função:

$$z(x) = \sum_i \alpha_i \psi_i(x) \quad (5.12)$$

Neste exemplo, relacionado a um Perceptron MLP, consideramos as funções elementares como aproximadores globais, no sentido de que suas entradas respondem ao espaço global do problema. Também, não são estáticas, mas sim adaptativas, dependendo dos parâmetros associados ao processamento da camada interna. Neste sentido, podemos dizer que um Perceptron de uma camada interna, ou uma MLP, aproxima uma função arbitrária, real e contínua, decidindo a orientação, localização e amplitude de um conjunto de funções de ativação sigmóides multidimensionais, conforme a Figura 40. Temos então que a soma das logísticas (linhas mais finas) resultam na função aproximada (linha mais escura).

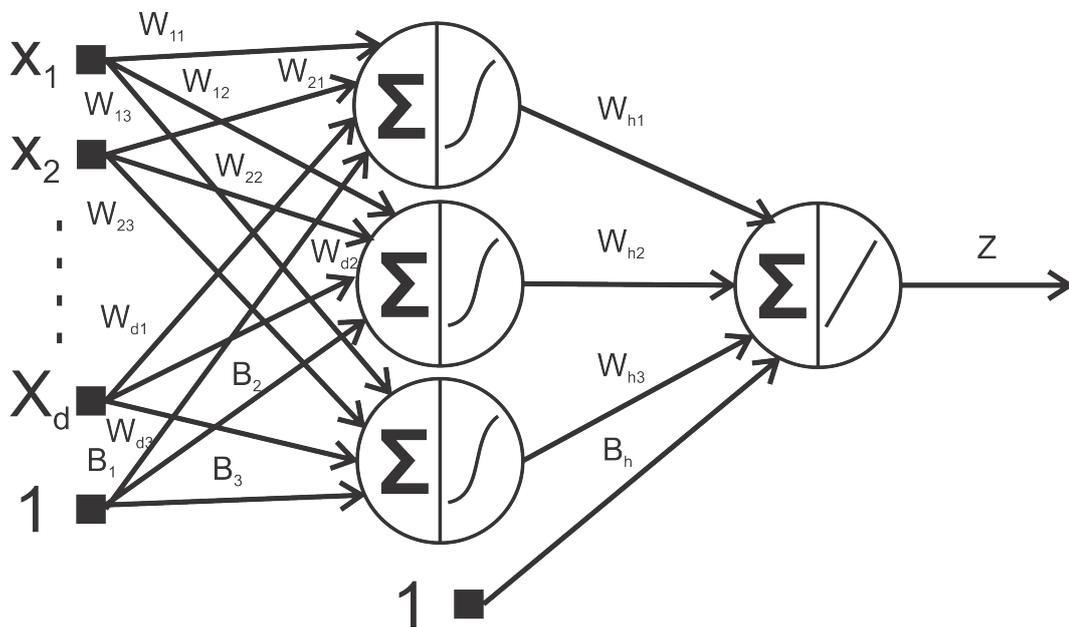


Figura 39 – MLP com base de funções

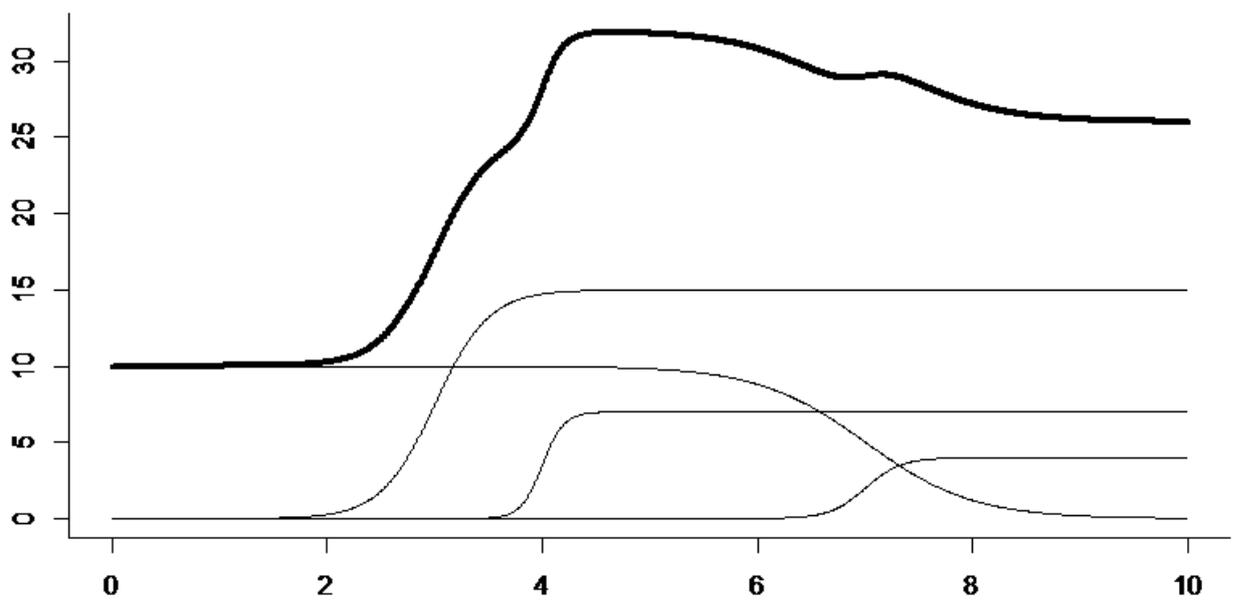


Figura 40 – Aproximação de funções com bases logísticas

5.1.5 *Backpropagation*

O treinamento da rede neural artificial é um procedimento que tem como objetivo ajustar ou adaptar o valor de seus pesos, convergindo assim, o resultado final da rede para o valor desejado.

O algoritmo *backpropagation*, é o procedimento mais conhecido e utilizado. Foi apresentado pela primeira vez por Paul Werbos em 1974 [125], porém somente depois do trabalho de David E. Rumelhart, Geoffrey E. Hinton e Ronald J. Williams em 1986 [126] ganhou fama e renovou o campo de pesquisa de redes neurais artificiais.

Ele é um algoritmo supervisionado, que utiliza o resultado esperado para adaptar seu modelo. Ele requer que a função de ativação dos neurônios da rede seja diferenciável, por isso muitas vezes utiliza-se a função logística ou tangente.

O treinamento pode ser em linha ou em lote. O treinamento em linha atualiza os pesos cada vez que um dado é inserido no modelo. O treinamento em lote atualiza os pesos em épocas. Consideramos uma época quando todos os dados forem inseridos no modelo.

Seu fluxo na prática é bem simples e pode ser demonstrado no resumo a seguir:

1. **Inicialização:** Inicialize os pesos e os bias aleatoriamente, com valores no intervalo $[-1; 1]$;

Definimos o grupo de treinamento: $T = \{x(n), d(n)\}$ onde $x(n)$ é a entrada e $d(n)$ a saída desejada.

2. **Apresentação dos dados de treinamento:** em linha, executar o passo 3 e 4 para cada indivíduo. Em lote executar o passo 3 e 4 para cada época.
3. **Propagação:** Calcular o valor de ativação pela Equação (5.13) e o cálculo da saída pela Equação (5.14) utilizando a função logística.

$$v_j = \sum_{i=1}^m w_{j,i} x_i + b \quad (5.13)$$

$$f(v) = \frac{1}{1 + e^{-av}} \quad (5.14)$$

A saída da última camada será a da rede.

4. **Sinal de Erro:** Calcular o erro.

$$e_j(n) = d_j(n) - O_j(n) \quad (5.15)$$

Onde $O_j(n)$ é a resposta da rede.

5. **Retropropagação:** Calcular os erros locais desde a saída até a entrada. O gradiente local é definido pela Equação (5.16) para a camada de saída ou pela Equação (5.17) para as camadas internas.

$$\delta_j(n) = e_j(n)O_j(n)(1 - O_j(n)) \quad (5.16)$$

$$\delta_j(n) = e_j(n)(1 - O_j(n)) \sum \delta_k w_{j,k} \quad (5.17)$$

Onde: $O_j(n)(1 - O_j(n))$ é a função de logística diferenciada, δ_k é o erro das camadas anteriores ligadas à camada j e $w_{j,k}$ os pesos das conexões da camada anterior.

Agora para ajustar os pesos temos:

$$\Delta w_{k,j}(n+1) = \alpha w_{k,j}(n) + \eta \delta_j \gamma_j \quad (5.18)$$

$$w(n+1) = w(n) + \Delta w_{k,j}(n) \quad (5.19)$$

onde α é a constante de *momentum*, η é a taxa de aprendizagem, δ_j o erro da unidade e γ_j a saída produzida pela unidade.

6. **Iteração:** O algoritmo repete os passos 3, 4, 5 até que o critério de parada seja satisfeito, como quando o erro está abaixo do definido ou quando o número de iterações chegou ao fim.

5.1.6 Critério de parada

O processo de treinamento de uma rede neural artificial requer um critério para ser interrompido ou finalizado.

Esse critério influencia no resultado final, pois não se deve treinar demais. Uma rede com *overfitting* representa bem os dados do grupo de treinamento, mas não representa bem os dados do grupo de teste.

Isso ocorre porque quanto maior o tempo de treinamento, maior é o ajuste do modelo ao conjunto de indivíduos do grupo de treinamento, tornando assim o modelo pouco geral. E ao utilizar o mesmo modelo num outro conjunto de indivíduos, como no conjunto de teste, o erro acaba sendo bem maior que o esperado.

Para tentar amenizar esse problema, um conjunto de dados, diferente dos dados de treinamento, é testado a cada época, e o valor do erro neste conjunto de dados é utilizado para definir o critério de parada. Esse conjunto de dados é denominado conjunto de validação cruzada.

Normalmente, após um número de épocas em que o erro do grupo de validação piora, o treinamento da rede é interrompido para tentar manter a generalização do modelo.

Outros critérios podem ser utilizados junto com a validação cruzada. Critérios mais simples como número de épocas ou tempo de treinamento são comuns.

5.2 ALGORITMOS GENÉTICOS (AG)

O AG é uma meta-heurística muito empregada em otimização. Diferente da programação matemática, praticamente todo tipo de problema pode ser acomodado ao AG, ou seja, não existem muitas restrições ao seu uso.

As técnicas empregadas no AG são inspiradas nos estudos de Charles Darwin [127], e seu objetivo é imitar o processo de evolução e tentar assim evoluir a melhor solução para um dado problema.

Nos anos 1950 e 1960, tivemos vários estudos com base nos resultados de Charles Darwin. Muitos problemas na área da ciência da computação [128] e na engenharia [129] foram solucionados com ferramentas utilizando o mesmo princípio.

Em 1973 John Holland publicou o primeiro artigo sobre o assunto [130] e em 1975 publicou o livro [131] que teve sua segunda edição publicada no início da década de 90 [109]. Diferente dos demais pesquisadores, Holland queria trabalhar a técnica da evolução de uma maneira mais geral e não focada em um problema específico. E foi dessas pesquisas que surgiu o conceito de Algoritmo Genético.

A ideia é codificar a solução do problema em indivíduos, uma estrutura de dados similar à de um cromossomo, e a partir dos melhores indivíduos gerarmos novas soluções [132]. Isso representa diretamente a teoria da evolução onde os indivíduos mais aptos tendem a se reproduzir mais e com isso passam seu material genético para seus descendentes.

Abaixo estão apresentadas algumas definições importantes para um melhor entendimento do AG [132]:

- Cromossomo: cadeia de caracteres representando informações relativas às variáveis do problema.
- Indivíduo: solução representada por um cromossomo
- Gene: uma unidade básica de um cromossomo
- População: o conjunto de indivíduos ou soluções candidatas
- Geração: Etapa do processo evolutivo da população
- Operações Genéticas: operações realizadas nos cromossomos

- Aptidão: valor numérico do desempenho da solução, utilizado para a seleção de indivíduos para reprodução.

O processo de evolução computacional do AG segue um esquema simples, como demonstrado na Figura 41. Existe um grande conjunto de opções que podem ser utilizadas para melhor processo computacional do método.

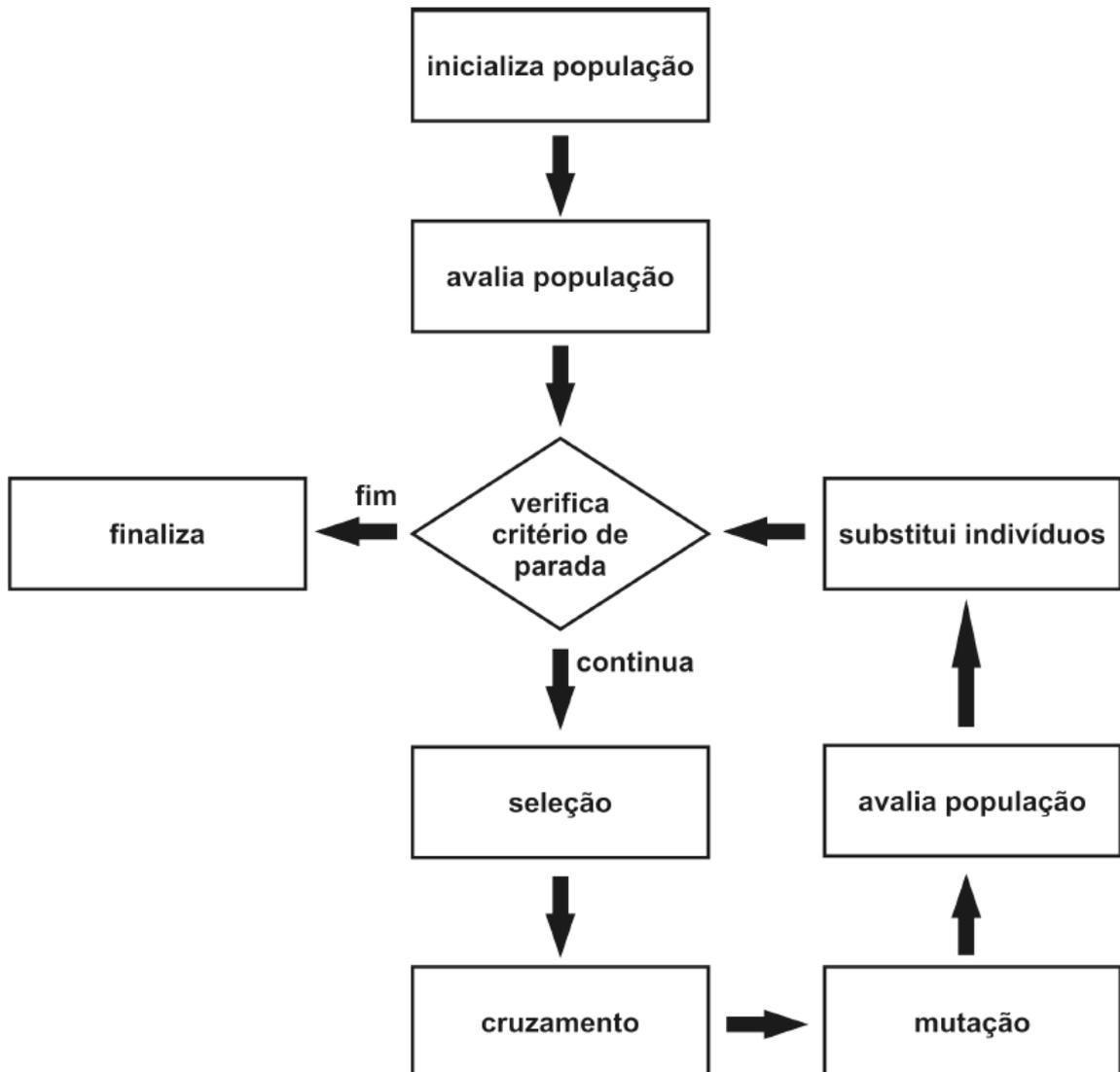


Figura 41 – Fluxo de funcionamento de um AG simples

5.2.1 Processo de evolução

Basicamente apenas dois passos da codificação de um AG são dependentes do problema: a codificação dos dados e a função objetivo.

A codificação é a metodologia a ser adotada na representação de uma solução candidata do problema em um cromossomo e a função objetivo é a função que irá informar a aptidão do indivíduo, ou seja, ela irá informar quão boa é uma dada solução.

A codificação é uma das mais importantes decisões do processo, pois é a partir dela que iremos representar computacionalmente uma solução candidata do problema [133]. Nele iremos levar as variáveis do problema para um cromossomo artificial. Continuando o paralelo, o gene do cromossomo artificial é a representação individual de cada uma das variáveis do problema.

Essa transformação entre solução/cromossomo é o que diferencia o genótipo do fenótipo. O genótipo é a representação de uma solução em um cromossomo, e o fenótipo representa a informação no contexto do problema. Existem diversas formas de representação do cromossomo. As mais importantes são:

- Binária: utilizada em diversos tipos de problemas, muito utilizado na representação de valores inteiros e contínuos.
- Real: mais utilizada em problemas de variáveis contínuas e muitas vezes também a mais adequada.
- Permutação de símbolos: importante para problemas onde a ordem da informação é importante.
- Árvore: utilizada principalmente em problemas de regressão simbólica. Ex. ajuste de uma expressão algébrica.

A decodificação do cromossomo binário pode ser facilmente exemplificada pela tabela da Figura 42:

Indivíduo	Cromossomo (Binário)	Fenótipo (Inteiro)
1	100011	35
2	001000	8
3	101011	43
4	111000	56

Figura 42 – Exemplo de codificação

5.2.2 *Função objetivo*

A função objetivo, como já foi dito, leva uma solução ou cromossomo a um valor que será utilizado para avaliar a aptidão do mesmo. Nesta etapa, após a decodificação do cromossomo, as variáveis do problema são repassadas para a função objetivo e a partir do seu resultado podemos inferir sobre a aptidão do indivíduo.

Num exemplo simples podemos utilizar a função $f(x) = x^2$. Utilizando os indivíduos da tabela da Figura 42 temos respectivamente os valores de aptidão: 1225, 64, 1849, 3136. Considerando uma otimização para encontrar o ponto de máximo, temos como melhor o indivíduo 4. No caso de encontrar o ponto de mínimo, temos como melhor o indivíduo 2.

Como a função objetivo varia para cada problema, ela pode ser tanto uma simples função matemática, como requerer uma completa simulação computacional. Devido a essa variação no tipo de função objetivo, existem algumas variações no AG para funções mais simples ou mais complexas.

5.2.3 *Processo de seleção*

O processo de seleção é o momento de escolhermos, dentre a população atual, os indivíduos que irão propagar seu material genético para a próxima geração.

Existem diversas formas de seleção e cada uma leva em consideração de forma diferente a aptidão do indivíduo. O trabalho de Blickle e Thiele [134] faz um estudo comparativo entre as diversas formas de seleção utilizadas, em especial o seguinte conjunto de processos seletivos: torneio, truncamento, ranking e roleta.

- **Roleta:** conhecida como seleção proporcional. A probabilidade de um indivíduo ser escolhido é proporcional a sua aptidão.
- **Torneio:** dois ou mais indivíduos são escolhidos aleatoriamente da população e comparados. O melhor indivíduo é selecionado.
- **Truncada:** somente uma fração mais apta da população participa da seleção, e neste conjunto todos tem a mesma probabilidade de serem escolhidos.
- **Ranking:** os indivíduos são ordenados de acordo com sua aptidão e a probabilidade de serem escolhidos é proporcional a esta posição. Essa proporção pode ser linear ou exponencial.

5.2.4 *Operadores genéticos*

Em AG temos um conjunto de operadores. São os responsáveis pela manipulação genética dos indivíduos e na geração de novos. É a partir desse processo que os indivíduos

passam adiante seu material genético.

Eles podem trabalhar com um ou mais cromossomos. Os operadores de cruzamento ou *crossover* trabalham com mais de um cromossomo, normalmente dois deles. Os operadores de mutação trabalham em apenas um cromossomo por vez.

5.2.5 Operador de *crossover*

A operação de *crossover*, ou recombinação, cria um ou mais filhos a partir de seus pais. Normalmente são utilizados dois indivíduos e criados dois novos. Em uma codificação binária costumam ser utilizados *crossover* de um ou dois pontos ou então o *crossover* uniforme.

O *crossover* de um ou dois pontos define posições aleatórias entre os genes dos indivíduos e faz cortes no mesmo. Depois combina cada uma das partes gerando novos indivíduos, como pode ser visto no exemplo das Figura 43 e Figura 44:

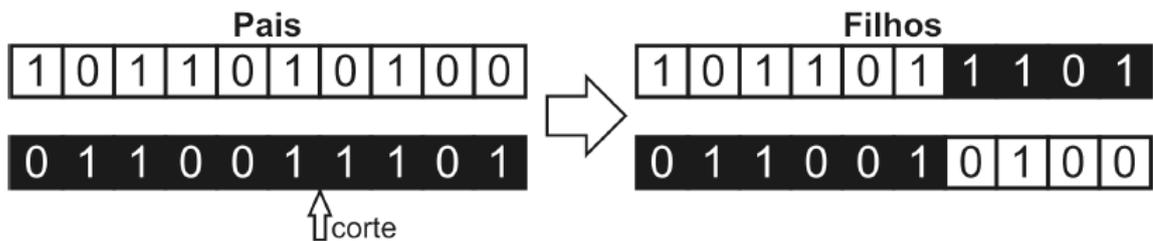


Figura 43 – *Crossover* de um ponto

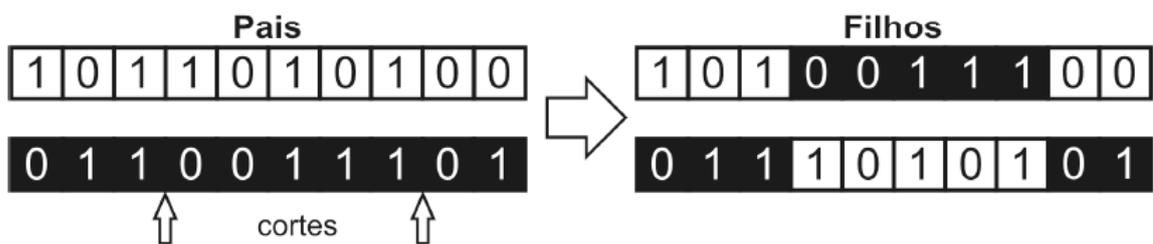
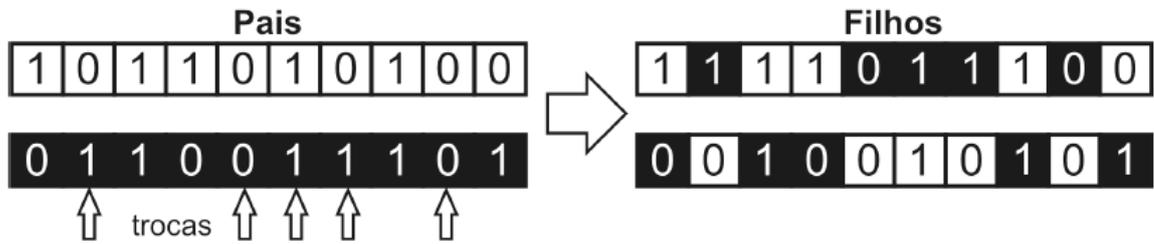


Figura 44 – *Crossover* de dois pontos

No *crossover* uniforme caminha-se pelo cromossomo e escolhe-se aleatoriamente como será feita a troca de material genético. A Figura 45 mostra um exemplo de como o *crossover* uniforme trabalha:

Não é sempre que os indivíduos se combinam, existe uma probabilidade para que a execução dessa etapa seja feita. Caso o *crossover* não ocorra, é feita uma simples clonagem dos indivíduos selecionados.

Figura 45 – *Crossover* uniforme

5.2.6 Operador de mutação

Após o processo de combinação, os novos indivíduos passam pelo operador de mutação. Esse operador é muito importante, pois é um operador exploratório e tenta criar uma maior diversidade na população e também combate a convergência prematura.

Ele trabalha de forma simples e com apenas um indivíduo. De acordo com uma probabilidade, operações pontuais são feitas no cromossomo. Como exemplo em uma codificação binária, passamos por cada gene do cromossomo e com certa probabilidade trocamos seu valor. Esse processo é demonstrado na Figura 46:



Figura 46 – Operador de mutação

Outros métodos também podem ser utilizados, vai depender da estrutura utilizada. Como também a probabilidade de mutação não precisa ser fixa, ela pode mudar de acordo com as gerações. Normalmente quanto mais próximo do final do processo, menos é a intensidade da mutação, pois nessa etapa o AG tenta refinar uma boa solução e não continuar a exploração.

5.3 AMORTECIMENTO EXPONENCIAL

A amortecimento exponencial é um método de previsão, que pode ser visto como uma média ponderada. Porém, nesse caso, o método pondera cada valor anterior com um peso diferente e quanto mais antiga a observação, menor o seu peso.

$$M_T = \alpha_1 Z_T + \alpha_2 Z_{T-1} + \alpha_3 Z_{T-2} + \alpha_4 Z_{T-3} + \dots \quad (5.20)$$

Onde Z_T é o valor atual da série de dados e $\alpha_1 > \alpha_2 > \alpha_3 > \alpha_4 \dots$

Como é trabalhoso escolhermos todos os parâmetros alfas, simplificamos ainda mais o método, onde temos apenas um parâmetro alfa, que calcula pesos de potências sucessivas.

$$M_T = \alpha Z_T + \alpha(1-\alpha)Z_{T-1} + \alpha(1-\alpha)^2 Z_{T-2} + \dots + \alpha(1-\alpha)^k Z_{T-k} + \dots + \alpha(1-\alpha)^T Z_0 \quad (5.21)$$

Como $0 < \alpha < 1$ e $0 < 1 - \alpha < 1$ o maior peso é dado à observação mais recente e os pesos seguintes descresem exponencialmente.

Contudo, ainda simplificamos ainda mais, substituindo o cálculo da média, que na Equação (5.21) leva em consideração todas as observações anteriores, por um cálculo iterativo, onde a média é atualizada a cada instante, pela média anterior e o novo valor de observação.

$$M_T = \alpha Z_T + (1 - \alpha)M_{T-1} \quad (5.22)$$

Considerando que $M_T = \hat{Z}_{T+1}$ e $M_{T-1} = \hat{Z}_T$ podemos escrever a equação como:

$$\hat{Z}_{T+1} = \alpha Z_T + (1 - \alpha)\hat{Z}_T \quad (5.23)$$

Podemos ainda, decompor a equação e ver o método de amortecimento exponencial como um método que vai corrigindo a previsão, utilizando a erro anterior observado (se a previsão foi baixa, será aumentada; se foi alta, será abaixada):

$$\alpha Z_T + (1 - \alpha)\hat{Z}_T = \alpha Z_T + \hat{Z}_T - \alpha\hat{Z}_T = \hat{Z}_T + \alpha(Z_T - \hat{Z}_T) = \hat{Z}_T + \alpha e_T \quad (5.24)$$

Dessa forma, o método implementa uma forma de correção de erro, particularmente útil quando o nível da série sofre oscilações.

5.3.1 Método de Holt

O método anteriormente demonstrado, é o amortecimento exponencial simples, que funciona com séries estacionárias. Para séries com declividades, utilizamos o método de Holt.

O método de Holt, é um método iterativo de previsão, que utiliza duas equações, uma para estimar o nível da série e outra para estimar sua declividade ou tendência:

$$\hat{a}_T = \alpha Z_T + (1 - \alpha)(\hat{a}_{T-1} + \hat{b}_{T-1}) \quad (5.25)$$

$$\hat{b}_T = \beta(\hat{a}_T - \hat{a}_{T-1}) + (1 - \beta)\hat{b}_{T-1} \quad (5.26)$$

$$\hat{Z}_{T+1} = \hat{a}_T + \hat{b}_T \quad (5.27)$$

$$\hat{Z}_{T+k} = \hat{a}_T + \hat{b}_T k \quad (5.28)$$

As Equações (5.25, 5.26) aplicam a ideia dos métodos de amortecimento exponencial, o que permite obter estimativas de forma recursiva.

5.3.2 Método de Holt-Winters (HW)

A sazonalidade de um série temporal pode ser modelada de diversas formas. Um forma é a utilização de fatores sazonais. Eles são representados por uma série, onde temos um valor para cada período do ciclo sazonal. Eles podem ser acrescentados no modelo, tanto constante ou com tendência, de maneira aditiva ou multiplicativa.

Considerando, por exemplo, que o fator sazonal do mês t seja representado por $\rho^{m(t)}$ temos o seguinte modelo multiplicativo:

$$\hat{Z}_{T+k} = (\hat{a}_T + \hat{b}_T k)\hat{p}_T^{m(T+k)} \quad (5.29)$$

Essa formulação é também conhecida como método de Holt-Winters, e vamos explicar seu funcionamento passo a passo:

- Modelo:

$$Z_t = \mu_t \rho^{m(t)} + \epsilon_t \quad (5.30)$$

onde $\mu_t = a + bt$

- Previsão (supondo $T = 0$):

$$\hat{Z}_{T+k} = (\hat{a}_T + \hat{b}_T k)\hat{p}_T^{m(T+k)} \quad (5.31)$$

- Processo de inicialização:

No instante $t = T - 1$, temos as estimativas:

$$\hat{a}_{T-1} \quad (5.32)$$

$$\widehat{b}_{T-1} \quad (5.33)$$

$$\widehat{p}_{T-1}^{m(T)} \quad (5.34)$$

Com base nelas, fazemos as previsões dos parâmetros no instante seguinte, $t = T$:

$$\widehat{a}_{T|T-1} = \widehat{a}_{T-1} + \widehat{b}_{T-1} \quad (5.35)$$

$$\widehat{b}_{T|T-1} = \widehat{b}_{T-1} \quad (5.36)$$

$$\widehat{p}_{T|T-1}^{m(T)} = \widehat{p}_{T-1}^{m(T)} \quad (5.37)$$

Depois, quando temos o valor observado Z_T , obtemos novas estimativas dos parâmetros, com base neste novo valor e nas estimativas dos outros parâmetros:

$$\widehat{a}_{T|T} = \frac{Z_T}{\widehat{p}_{T-1}^{m(T)}} \quad (5.38)$$

$$\widehat{b}_{T|T} = \widehat{a}_T - \widehat{a}_{T-1} \quad (5.39)$$

$$\widehat{p}_{T|T}^{m(T)} = \frac{Z_T}{\widehat{a}_T} \quad (5.40)$$

E enfim, as duas estimativas para cada parâmetro são combinadas, por uma média:

$$\widehat{a}_T = \alpha \widehat{a}_{T|T} + (1 - \alpha) \widehat{a}_{T|T-1} = \alpha \left[\frac{Z_T}{\widehat{p}_{T-1}^{m(T)}} \right] + (1 - \alpha) (\widehat{a}_{T-1} + \widehat{b}_{T-1}) \quad (5.41)$$

$$\widehat{b}_T = \beta \widehat{b}_{T|T} + (1 - \beta) \widehat{b}_{T|T-1} = \beta (\widehat{a}_T - \widehat{a}_{T-1}) + (1 - \beta) \widehat{b}_{T-1} \quad (5.42)$$

$$\widehat{p}_T^{m(T)} = \gamma \widehat{p}_{T|T}^{m(T)} + (1 - \gamma) \widehat{p}_{T|T-1}^{m(T)} = \gamma \left[\frac{Z_T}{\widehat{a}_T} \right] + (1 - \gamma) \widehat{p}_{T-1}^{m(T)} \quad (5.43)$$

- Normalização:

Os fatores devem ser normalizados a cada passo do algoritmo:

$$\sum_{i=1}^S \widehat{p}_T^{m(i)} = S \quad (5.44)$$

onde S é o período sazonal.

$$\widehat{p}_T^{m(j)} = \widehat{p}_T^{m(j)*} \times \frac{S}{\sum_{i=1}^S \widehat{p}_T^{m(i)*}} \text{ para } j = 1, 2, \dots, S \quad (5.45)$$

onde $\widehat{p}_T^{m(j)*}$ indica os fatores sazonais antes da normalização.

- Inicialização:

1) Tendência inicial (b_0):

$$\widehat{b}_0 = \frac{1}{S} \left[\frac{(Z_{S+1} - Z_1)}{S} + \frac{(Z_{S+2} - Z_2)}{S} + \dots + \frac{(Z_{S+S} - Z_S)}{S} \right] \quad (5.46)$$

2) Nível inicial:

$$\widehat{a}_0 = \frac{1}{S}(Z_1 + \dots + Z_S) \quad (5.47)$$

3) Fatores sazonais iniciais:

$$\widehat{p}_0^{m(t)} = \frac{Z_t}{Z_1} \text{ onde } t = 1, 2, \dots, S \quad (5.48)$$

5.4 MÉTODO DE OTIMIZAÇÃO DE BYRD

O método de Byrd ou método L-BFGS-B, tem como base o método de otimização pelo Gradiente Conjugado. Ele é uma modificação ao método L-BFGS que por sua vez é uma modificação do método Broyden–Fletcher–Goldfarb–Shanno (BFGS).

A teoria do gradiente conjugado nos permite obter o mínimo de uma função $f(x)$, a partir da expansão em série de Taylor de sua derivada:

$$f'(x) = f'(x_0) + (x - x_0)f''(x_0) \quad (5.49)$$

Considerando o ponto de $f'(x) = 0$ temos:

$$x = x_0 - \frac{f'(x_0)}{f''(x_0)} \quad (5.50)$$

Para funções de mais de uma dimensão, temos a seguinte notação:

$$X = X_0 - H^{-1}\nabla f \quad (5.51)$$

sendo que H é a matriz *Hessiana*, ou seja, a matriz de derivadas segunda de $f(x)$, e ∇f é o vetor gradiente, que contém as derivadas primeiras da função f :

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 x_n} \\ \frac{\partial^2 f}{\partial x_2 x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 f}{\partial x_N x_1} & \frac{\partial^2 f}{\partial x_N x_2} & \cdots & \frac{\partial^2 f}{\partial x_N^2} \end{bmatrix} \quad (5.52)$$

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_N} \right) \quad (5.53)$$

Como o custo de computação da Hessiana é grande, quando N for grande, normalmente os métodos utilizam uma aproximação da Hessiana. Esses métodos são conhecidos como do tipo *quase-Newton*. O método BFGS é um desses métodos.

O método L-BFGS, faz uma adaptação ao método BFGS, diminuindo o consumo de memória. O método L-BFGS-B, por sua vez, é uma adaptação do método L-BFGS que trabalha com restrições de borda. Mais informações sobre o método de Byrd podem ser encontradas em [114].

REFERÊNCIAS

- [1] “Resenha Mensal do Mercado de Energia Elétrica”, 2013, <http://www.epe.gov.br/> - Acesso em 2015.
- [2] “Demanda de energia elétrica - 10 Anos (informe à imprensa)”, 2012, <http://www.epe.gov.br/> - Acesso em 2015.
- [3] GROSS, G., GALIANA, F., “Short-term load forecasting”, *Proceedings of the IEEE*, v. 75, pp. 1558–1573, 1987.
- [4] HAGAN, T. M., BEHR, S., “The Time Series Approach to Short Term Load Forecasting”, *IEEE Trans Power Systems*, v. 2, pp. 785–791, 1987.
- [5] HUANG, S.-J., SHIH, K.-R., “Short-Term Load Forecasting Via ARMA Model Identification Including Non-Gaussian Process Considerations”, *IEEE Trans Power Systems*, v. 18, pp. 673, May 2003.
- [6] CANCELO, J. R., ESPASA, A., GRADE, R., “Forecasting the electricity load for one day to one week ahead for the Spanish system operator”, *International Journal of Forecasting*, pp. 588–602, 2008.
- [7] HARVEY, A., KOOPMAN, S. J., “Forecasting hourly electricity demand using time-varying splines”, *Journal of the American Statistical Association*, v. 88, pp. 1228–1236, 1993.
- [8] INFIELD, D. G., HILL, D. C., “Optimal smoothing for trend removal in short term electricity demand forecasting”, *IEEE Transactions on Power Systems*, v. 13, pp. 1115–1120, 1998.
- [9] AL-HAMADI, H. M., SOLIMAN, S. A., “Short-term electric load forecasting based on Kalman filtering algorithm with moving window weather and load model.” *Electric Power Systems Research*, v. 64, pp. 47–59, 2004.
- [10] MARTIN-RODRIGUEZ, G., CACERES-HERNANDEZ, J. J., “Modelling the hourly Spanish electricity demand”, *Economic Modelling*, v. 22, pp. 551–569, 2005.
- [11] DORDONNAT, V., KOOPMAN, S. J., OOMS, M., DESSERTAINE, A., COLLET, J., “An hourly periodic state space model for modelling French national electricity load”, *International Journal of Forecasting*, v. 24, pp. 566–587, 2008.
- [12] RAHMAN, S., BHATNAGAR, R., “An Expert System Based Algorithm for Short Term Load Forecast”, *IEEE Transactions on Power Systems*, v. 3, pp. 392–399, 1988.
- [13] RAHMAN, S., HAZIM, O., “A Generalized Knowledge-based Short-term Load Forecasting Technique”, *IEEE Transactions on Power Systems*, v. 8, pp. 508–514, 1993.
- [14] LI, F., “Singular value decomposition expansion for electrical demand analysis”, *IMA Journal of Mathematics Applied in Business and Industry*, v. 11, pp. 37–48, 2000.

- [15] ABU-SHIKHAH, N., ELKARMI, F., “Medium-term electric load forecasting using singular value decomposition”, *Energy*, v. 36, pp. 4259–4271, 2011.
- [16] PAPALEXOPOULOS, A. D., HESTERBERG, T., “A regression-based approach to short-term system load forecasting”, *IEEE Transactions on Power Systems*, v. 5, pp. 1535–1547, 1990.
- [17] RAMANATHAN, R., ENGLE, R., GRANGER, C., VAHID-ARAGHI, F., BRACE, C., “Short-run forecasts of electricity loads and peaks”, *International journal of forecasting*, v. 13, pp. 161–174, 1997.
- [18] RAMANATHAN, R., GRANGER, C., ENGLE, R., “Two-step modelling for short-term forecasting”, In: BUNN, D. W., FARMER, E. (eds), *Comparative Models for Electrical Load Forecasting*, 1st ed., John Wiley and Sons Inc., 1985.
- [19] ACKERMAN, G., “Short-term load prediction for electric-utility control of generating units.” In: BUNN, D. W., FARMER, E. (eds), *Comparative Models for Electrical Load Forecasting*, 1st ed., John Wiley and Sons Inc., 1985.
- [20] LIU, K., SUBBARAYAN, S., SHOULTS, R. R., MANRY, M. T., KWAN, C., LEWIS, F. I., NACCARINO, J., “Comparison of very short-term load forecasting techniques”, *IEEE Transactions on Power Systems*, v. 11, pp. 877–882, 1996.
- [21] PAPADAKIS, S. E., THEOCHARIS, J. B., BAKIRTZIS, A. G., “A load curve based fuzzy modeling technique for short-term load forecasting”, *Fuzzy Sets and Systems*, v. 135, pp. 279–303, 2003.
- [22] MAMLOOK, R., BADRAN, O., ABDULHADI, E., “A fuzzy inference model for short-term load forecasting”, *Energy Policy*, v. 37, pp. 1239–1248, 2009.
- [23] CHEN, B., CHANGE, M. L. C., LIN, C. J., “Load Forecasting Using Support Vector Machines: A Study on EUNITE Competition 2001”, *IEEE Transactions on Power Systems*, v. 19, pp. 1821–1830, 2004.
- [24] FAN, S., CHEN, L., “Short-term Load Forecasting Based on an Adaptive Hybrid Method”, *IEEE Transactions on Power Systems*, v. 21, pp. 392–401, 2006.
- [25] NIU, D., WANG, Y., WU, D. D., “Power load forecasting using support vector machine and ant colony optimization”, *Expert Systems With Applications*, v. 37, pp. 2531–2539, 2010.
- [26] CHEN, Y., YANG, Y., LIU, C., LI, C., LI, L., “A hybrid application algorithm based on the support vector machine and artificial intelligence: An example of electric load forecasting”, *Applied Mathematical Modelling*, v. 39, pp. 2617–2632, 2015.
- [27] YU, L., KIM, C., SONG, Y., “A novel short-term load forecasting technique using wavelet transform analysis”, *Electric Machines And Power Systems*, v. 28, pp. 537–549, 2000.
- [28] BAHRAMI, S., HOOSHMAND, R.-A., PARASTEGARI, M., “Short term electric load forecasting by wavelet transform and grey model improved by PSO (particle swarm optimization) algorithm”, *Energy*, v. 72, pp. 434–442, 2014.

- [29] METAXIOTIS, K., KAGIANNAS, A., ASKOUNIS, D., PSARRAS, J., “Artificial intelligence in short term electric load forecasting: a state-of-the-art survey for the researcher”, *Energy Conversion and Management*, v. 44, pp. 1525–1534, 2003.
- [30] ALFARES, H. K., NAZEERUDDIN, M., “Electric load forecasting: literature survey and classification of methods”, *International Journal of Systems Science*, v. 33, pp. 23–34, 2002.
- [31] HIPPERT, H. S., *Previsão de cargas a curto prazo - uma avaliação da viabilidade do uso de redes neurais artificiais*, Tese de doutorado, PUC-Rio, 2001.
- [32] HAHN, H., MEYER-NIEBERG, S., PICKL, S., “Electric load forecasting methods: Tools for decision making”, *European Journal of Operational Research*, v. 199, pp. 902–907, 2009.
- [33] RAZA, M. Q., KHOSRAVI, A., “A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings”, *Renewable and Sustainable Energy Reviews*, v. 50, pp. 1352–1372, 2015.
- [34] HONG, T., FAN, S., “Probabilistic electric load forecasting: A tutorial review”, *International Journal of Forecasting*, v. 32, n. 3, pp. 914–938, 2016.
- [35] YAO, S. J., SONG, Y. H., ZHANG, L. Z., CHENG, X. Y., “Wavelet transform and neural networks for short-term electrical load forecasting”, *Energy Conversion & Management*, v. 41, pp. 1975–1988, 2000.
- [36] REIS, A. J. R., SILVA, A. P. A., “Feature Extraction via Multiresolution Analysis for Short-Term Load Forecasting”, *IEEE Transactions on Power Systems*, v. 20, pp. 189–198, 2005.
- [37] LIU, Z., LI, W., SUN, W., “A novel method of short-term load forecasting based on multiwavelet transform and multiple neural”, *Neural Computing and Applications*, v. 22, pp. 271–277, 2013.
- [38] GHAYEKHLOO, M., MENHAJ, M., GHOFRANI, M., “A hybrid short-term load forecasting with a new data preprocessing framework”, *Electric Power Systems Research*, v. 119, pp. 138–148, 2015.
- [39] GHOFRANI, M., GHAYEKHLOO, M., ARABALI, A., GHAYEKHLOO, A., “A hybrid short-term load forecasting with a new input selection framework”, *Energy*, v. 81, pp. 777–786, 2015.
- [40] KHOTANZAD, A., ZHOU, E., ELRAGAL, H., “A Neuro-Fuzzy Approach to Short-Term Load Forecasting in a Price-Sensitive Environment”, *IEEE Transactions on Power Systems*, v. 17, pp. 1273–1282, 2002.
- [41] ABRAHAM, A., NATH, B., “A neuro-fuzzy approach for modelling electricity demand in Victoria”, *Applied Soft Computing Journal*, v. 1, pp. 127–138, 2001.
- [42] ZAHEDI, G., AZIZI, S., BAHADORI, A., ELKAMEL, A., ALWI, S., “Electricity demand estimation using an adaptive neuro-fuzzy network: A case study from the Ontario province - Canada”, *Energy*, v. 49, pp. 323–328, 2013.

- [43] ÇEVİK, H. H., ÇUNKAŞ, M., “Short-term load forecasting using fuzzy logic and ANFIS”, *Neural Computing and Applications*, v. 26, pp. 1355–1367, 2015.
- [44] CHONG-XIN, L. L., RECAI, L. K., “Application of Chaos and Neural Network in Power Load Forecasting”, *Discrete Dynamics in Nature and Society*, v. 2011, pp. 12, 2011.
- [45] LIU, Y., LEI, S., SUN, C., ZHOU, Q., REN, H., “A multivariate forecasting method for short-term load using chaotic features and RBF neural network”, *European Transactions on Electrical Power*, v. 21, pp. 1376–1391, 2011.
- [46] KOUHI, S., KEYNIA, F., RAVADANEGH, S. N., “A new short-term load forecast method based on neuro-evolutionary algorithm and chaotic feature selection”, *International Journal of Electrical Power & Energy Systems*, v. 62, pp. 862–867, 2014.
- [47] AWAN, S. M., ASLAM, M., KHAN, Z. A., SAEED, H., “An efficient model based on artificial bee colony optimization algorithm with Neural Networks for electric load forecasting”, *Neural Computing and Applications*, v. 25, pp. 1967–1978, 2014.
- [48] GERMI, M., MIRJAVADI, M., BAZIAR, A., GERMI, M., “A hybrid model for daily peak load power forecasting based on SAMBA and neural network”, *Journal of Intelligent and Fuzzy Systems*, v. 27, pp. 913–920, 2014.
- [49] ULAGAMMAI, M., VENKATESH, P., KANNAN, P. S., PADHY, N. P., “Application of bacterial foraging technique trained artificial and wavelet neural networks in load forecasting”, *Neurocomputing*, v. 70, pp. 2659–2667, 2007.
- [50] KAVOUSHI-FARD, A., NIKNAM, T., GOLMARYAMI, M., “Short term load forecasting of distribution systems by a new hybrid modified FA-backpropagation method”, *Journal of Intelligent and Fuzzy Systems*, v. 26, pp. 517–522, 2014.
- [51] XIAO, L., WANG, J., HOU, R., WU, J., “A combined model based on data pre-analysis and weight coefficients optimization for electrical load forecasting”, *Energy*, v. 82, pp. 524–549, 2015.
- [52] EL-TELBANY, M., EL-KARMI, F., “Short-term forecasting of Jordanian electricity demand using particle swarm optimization”, *Electric Power Systems Research*, v. 78, pp. 425–433, 2008.
- [53] SHAYEGHI, H., SHAYANFAR, A. H., AZIMI, G., “STLF based on optimized neural network using PSO”, *International Journal of Electrical and Computer Engineering*, v. 4, pp. 1190–1199, 2009.
- [54] QUAN, H., SRINIVASAN, D., KHOSRAVI, A., “Uncertainty handling using neural network-based prediction intervals for electrical load forecasting”, *Energy*, v. 73, pp. 916–925, 2014.
- [55] LING, S. H., LEUNG, F. H. F., LAM, H. K., YIM-SHU, L., TAM, P. K. S., “A novel genetic-algorithm-based neural network for short-term load forecasting”, *IEEE Transactions on Industrial Electronics*, v. 50, pp. 793–799, 2003.

- [56] AMJADY, N., KEYNIA, F., “Short-term load forecasting of power systems by combination of wavelet transform and neuro-evolutionary algorithm”, *Energy*, v. 34, pp. 46–57, 2009.
- [57] CHATURVEDI, D., SINHA, A., MALIK, O., “Short term load forecast using fuzzy logic and wavelet transform integrated generalized neural network”, *International Journal of Electrical Power & Energy Systems*, v. 67, pp. 230–237, 2015.
- [58] JUNGUO, W., JIANZHONG, Z., BING, P., “A hybrid neural genetic method for load forecasting based on phase space reconstruction”, *Kybernetes*, v. 39, n. 8, pp. 1291–1297, 2010.
- [59] LING, S. H., LEUNG, F. H. F., LAM, H. K., TAM, P. K. S., “Short-term electric load forecasting based on a neural fuzzy network”, *IEEE Transactions on Industrial Electronics*, v. 50, pp. 1305–1316, 2003.
- [60] LI, P., LI, Y., XIONG, Q., CHAI, Y., ZHANG, Y., “Application of a hybrid quantized Elman neural network in short-term load forecasting”, *International Journal of Electrical Power & Energy Systems*, v. 55, pp. 749–759, 2014.
- [61] SHEIKHAN, M., MOHAMMADI, N., “Neural-based electricity load forecasting using hybrid of GA and ACO for feature selection”, *Neural Computing and Applications*, v. 21, pp. 1961–1970, 2012.
- [62] CHARLTON, N., SINGLETON, C., “A refined parametric model for short term load forecasting”, *International Journal of Forecasting*, v. 30, pp. 364–368, 2014.
- [63] HONG, T., PINSON, P., FAN, S., “Global energy forecasting competition 2012”, *International Journal of Forecasting*, v. 30, pp. 357–363, 2014.
- [64] YANG, H. T., HUANG, C. M., “A new short-term load forecasting approach using self-organizing fuzzy ARMAX models”, *IEEE Transactions Power Systems*, v. 13, pp. 217–225, 1998.
- [65] WANG, B., TAI, N., ZHAI, H., YE, J., ZHU, J., QI, L., “A new ARMAX model based on evolutionary algorithm and particle swarm optimization for short-term load forecasting”, *Electric Power Systems Research*, v. 78, pp. 1679–1685, 2008.
- [66] MASTOROCOSTAS, P. A., THEOCHARIS, J. B., KIARTZIS, S. J., BAKIRTZIS, A. G., “A hybrid fuzzy modeling method for short-term load forecasting”, *Mathematics And Computers In Simulation*, v. 51, pp. 221–232, 2000.
- [67] SONG, K., BAEK, Y. H. D. J. G., “Short-term Load Forecasting for the Holidays Using Fuzzy Linear Regression Method”, *IEEE Transactions on Power Systems*, v. 20, pp. 96–101, 2005.
- [68] AZADEH, A., SABERI, M., SERAJ, O., “An integrated fuzzy regression algorithm for energy consumption estimation with non-stationary data: a case study of Iran”, *Energy*, v. 35, pp. 2351–2366, 2010.
- [69] VILLALBA, S. A., BEL, C. L., “Hybrid Demand Model for Load Estimation and Short Term Load Forecasting in Distribution Electric Systems”, *IEEE Transactions on Power Delivery*, v. 15, pp. 764–769, 2000.

- [70] FILIK, U., KURBAN, M., FILIK, U., “A New Approach for the Short-Term Load Forecasting with Autoregressive and Artificial Neural Network Models”, *International Journal of Computational Intelligence Research*, v. 3, pp. 66–71, 2007.
- [71] YADAV, V., SRINIVASAN, D., “A SOM-based hybrid linear-neural model for short-term load forecasting”, *Neurocomputing*, v. 74, pp. 2874–2885, 2011.
- [72] BALLINI, R., YAGER, R., “OWA filters and forecasting models applied to electric power load time series”, *Evolving Systems*, v. 5, pp. 159–173, 2014.
- [73] YANG, Y., WU, J., CHEN, Y., LI, C., “A new strategy for short-term load forecasting”, *Abstract and Applied Analysis*, v. 2013, 2013.
- [74] BENNETT, C., STEWART, R., LU, J., “Autoregressive with exogenous variables and neural network short-term load forecast models for residential low voltage distribution networks”, *Energies*, v. 7, pp. 2938–2960, 2014.
- [75] WANG, J., WANG, J., LI, Y., ZHU, S., ZHAO, J., “Techniques of applying wavelet de-noising into a combined model for short-term load forecasting”, *International Journal of Electrical Power and Energy Systems*, v. 62, pp. 816–824, 2014.
- [76] CAI, G., WANG, W., LU, J., “A Novel Hybrid Short Term Load Forecasting Model Considering the Error of Numerical Weather Prediction”, *Energies*, v. 9, n. 12, pp. 994, 2016.
- [77] TAYLOR, J. W., “Short-term Electricity Demand Forecasting Using Double-seasonal Exponential Smoothing”, *Journal of Operational Research Society*, v. 54, pp. 799–805, 2003.
- [78] TAYLOR, J. W., “Discussion”, *International Journal of Forecasting*, pp. 671–672, 2006.
- [79] TAYLOR, J. W., MCSHARRY, P. E., “Short-term load forecasting methods: An evaluation based on european data”, *IEEE Transactions on Power Systems*, pp. 2213–2219, 2007.
- [80] TAYLOR, J. W., “Triple seasonal methods for short-term electricity demand forecasting”, *European Journal of Operational Research*, v. 204, pp. 139–152, 2010.
- [81] TAYLOR, J. W., “Exponentially weighted methods for forecasting intraday time series with multiple seasonal cycles”, *International Journal of Forecasting*, v. 26, pp. 627–646, 2010.
- [82] TAYLOR, J. W., “Short-Term Load Forecasting With Exponentially Weighted Methods”, *IEEE Transactions on Power Systems*, v. 27, pp. 458–464, 2012.
- [83] TAYLOR, J. W., SNYDER, R., “Forecasting Intraday Time Series with Multiple Seasonal Cycles Using Parsimonious Seasonal Exponential Smoothing”, *Omega*, v. 40, pp. 748–757, 2012.
- [84] ARORA, S., TAYLOR, J. W., “Short-term Forecasting of Anomalous Load Using Rule-based Triple Seasonal Methods”, *IEEE Transactions on Power Systems*, v. 28, pp. 3235–3242, 2013.

- [85] SUDHEER, G., SUSEELATHA, A., “Short term load forecasting using wavelet transform combined with Holt–Winters and weighted nearest neighbor models”, *International Journal of Electrical Power & Energy Systems*, v. 64, pp. 340–346, 2015.
- [86] DA SILVA CARNEIRO, A. C. M., *Previsão do consumo de energia elétrica a curto prazo, usando combinações de métodos univariados*, Dissertacao de mestrado, Programa de Pós-Graduação em Modelagem Computacional - UFJF, 2014.
- [87] SHUKUR, O. B., FADHIL, N. S., LEE, M. H., AHMAD, M. H., “Electricity load forecasting using hybrid of multiplicative double seasonal exponential smoothing model with artificial neural network”, *Jurnal Teknologi*, v. 69, pp. 65–70, 2014.
- [88] NETO, G. G., *Métodos univariados e multivariados para previsão de demanda de energia elétrica em curto prazo: Um estudo comparativo*, Dissertacao de mestrado, Programa de Pós-Graduação em Modelagem Computacional - UFJF, 2014.
- [89] “Periódicos da CAPES”, 2017, <http://www.periodicos.capes.gov.br/> - Acesso em 01/12/2017.
- [90] HIPPERT, H. S., PEDREIRA, C. E., SOUZA, R. C., “Neural networks for short-term load forecasting: A review and evaluation”, *IEEE Transactions on power systems*, v. 16, pp. 44–55, 2001.
- [91] KHOTANZAD, A., AFKHAMI-ROHANI, R., MARATUKULAM, D., “ANNSTLF-artificial neural network short-term load forecaster-generation three”, *IEEE Transactions on Power Systems*, v. 13, n. 4, pp. 1413–1422, 1998.
- [92] BUNN, D. W., “Forecasting loads and prices in competitive power markets”, *Proceedings of the IEEE*, v. 88, n. 2, pp. 163–169, 2000.
- [93] HYNDMAN, R. J., ATHANASOPOULOS, G., *Forecasting: principles and practice*. OTexts, 2014.
- [94] ALPAYDIN, E., *Introduction to machine learning*. 2nd ed. MIT Press, 2010.
- [95] HASTIE, T., TIBSHIRANI, R., FRIEDMAN, J., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics, Springer New York, 2009.
- [96] PRINCIPE, J. C., EULIANO, N. R., LEFEBVRE, W. C., *Neural And Adaptative Systems*. John Wiley & Sons Inc., 2000.
- [97] ARORA, S., TAYLOR, J. W., “Rule-based autoregressive moving average models for forecasting load on special days: A case study for France”, *European Journal of Operational Research*, 2017.
- [98] TAYLOR, J. W., MENDES, L. M. P. D., “A Comparison of Univariate Methods for Forecasting Electricity Demand Up to a Day Ahead”, *International Journal of Forecasting*, v. 22, pp. 1–16, 2006.
- [99] “Dados competição EUNITE 2001”, 2001, <http://neuron.tuke.sk/competition/> - Acesso em 07/01/2009.

- [100] CHANG, M.-W., CHEN, B.-J., LIN, C.-J., “EUNITE network competition: Electricity load forecasting”. In: *National Taiwan University*, 2001.
- [101] CHARLTON, N., SINGLETON, C., “A refined parametric model for short term load forecasting”, *International Journal of Forecasting*, v. 30, n. 2, pp. 364–368, 2014.
- [102] LLOYD, J. R., “GEFCom2012 hierarchical load forecasting: Gradient boosting machines and Gaussian processes”, *International Journal of Forecasting*, v. 30, n. 2, pp. 369–374, 2014.
- [103] DA S. CARNEIRO, A. C. M., NETO, G. G., DEFILIPPO, S. B., HIPPERT, H. S., “Previsão de carga a curto prazo por combinações de métodos lineares”. In: *2014 SIMPEP Anais do XXI SIMPEP - Simpósio de Eng. de Produção*, 2014.
- [104] BISHOP, C. M., *Neural networks for pattern recognition*. Oxford Univ Press, 1995.
- [105] HAYHIN, S., *Redes Neurais - Princípios e prática*. Bookman, 2001.
- [106] MACKAY, D. J., “Bayesian interpolation”, *Neural computation*, v. 4, n. 3, pp. 415–447, 1992.
- [107] BLUM, A. L., LANGLEY, P., “Selection of relevant features and examples in machine learning”, *Artificial intelligence*, v. 97, n. 1, pp. 245–271, 1997.
- [108] HIPPERT, H. S., TAYLOR, J. W., “An evaluation of Bayesian techniques for controlling model complexity and selecting inputs in a neural network for short-term load forecasting”, *Neural networks*, v. 23, n. 3, pp. 386–395, 2010.
- [109] HOLLAND, J. H., *Adaptation in Natural and Artificial systems*. University of Michigan Press, 1992.
- [110] CHATURVEDI, D., SINHA, A., MALIK, O., “Short term load forecast using fuzzy logic and wavelet transform integrated generalized neural network”, *International Journal of Electrical Power & Energy Systems*, v. 67, pp. 230–237, 2015.
- [111] SHEIKHAN, M., MOHAMMADI, N., “Time series prediction using PSO-optimized neural network and hybrid feature selection algorithm for IEEE load data”, *Neural computing and applications*, v. 23, n. 3-4, pp. 1185–1194, 2013.
- [112] BOX, G. E. P., JENKINS, G. M., REINSEL, G. C., *Linear Stationary Models*. A John Wiley & Sons, Inc., Publication, 2008.
- [113] BATES, J. M., GRANGER, C. W., “The combination of forecasts”, *Or*, pp. 451–468, 1969.
- [114] BYRD, R. H., LU, P., NOCEDAL, J., ZHU, C., “A limited memory algorithm for bound constrained optimization”, *SIAM Journal on Scientific Computing*, v. 16, pp. 1190–1208, 1995.
- [115] DOUGLAS, A. P., BREIPOHL, A. M., LEE, F. N., ADAPA, R., “The impacts of temperature forecast uncertainty on Bayesian load forecasting”, *IEEE Transactions on Power Systems*, v. 13, n. 4, pp. 1507–1513, 1998.

- [116] VALENZUELA, J., MAZUMDAR, M., KAPOOR, A., “Influence of temperature and load forecast uncertainty on estimates of power generation production costs”, *IEEE Transactions on Power Systems*, v. 15, n. 2, pp. 668–674, 2000.
- [117] TAYLOR, J. W., MAJITHIA, S., “Using combined forecasts with changing weights for electricity demand profiling”, *Journal of the Operational Research Society*, pp. 72–82, 2000.
- [118] VAPNIK, V. N., CHERVONENKIS, A. Y., “On the uniform convergence of relative”, *Theory of probability and its applications*, v. 16, pp. 264–280, 1971.
- [119] WIDROW, B., HOFF, J. R., “Adaptative Switching Circuits”, *M. E. In Ire Western Electric Show And Convention Record*, v. 4, pp. 96–104, 1960.
- [120] ROSENBLATT, F., “The Perceptron: A probabilistic model for information storage and organization in the brain”, *Psychological Review*, v. 65, pp. 386–408, 1958.
- [121] MINSKY, M. L., PAPERT, S. A., *Perceptrons*. MIT Press, 1969.
- [122] KOLMOGOROV, A. N., “On the representation of continuous functions of several variables by superposition of continuous functions of one variable and addition.” *oklady Akademiia Nauk*, v. 114, pp. 953–956, 1957.
- [123] FUNAHASHI, K., “On the approximate realization of continuous mappings by neural networks”, *Neural Networks*, v. 2, pp. 183–192, 1989.
- [124] CYBENKO, G., “Approximation by superpositions of a sigmoidal function”, *Mathematics of Control. Signals and Systems*, v. 2, pp. 304–314, 1989.
- [125] WERBOS, P. J., “Beyond regression: New tools for prediction and analysis in the behavioral sciences”, *Doctoral Dissertation, Applied Mathematics, Harvard University, MA*, 1974.
- [126] RUMELHARD, D., HILTON, G., WILLIAMS, R., “Learning representations by backpropagation errors”, *Nature*, v. 323, pp. 533–566, 1986.
- [127] DARWIN, C., *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. John Murray, 1859.
- [128] FOGEL, L. J., OWENS, A. J., WALSH, M. J., *Artificial Intelligence through Simulated Evolution*. John Wiley, 1966.
- [129] RECHENBERG, I., *Cybernetic Solution Path of an Experimental Problem*. Hants, 1965.
- [130] HOLLAND, J. H., “Genetic algorithms and the optimal allocation of trials”, *SIAM J. Comput.*, v. 2, pp. 88–105, 1973.
- [131] HOLLAND, J. H., *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- [132] BORGES, F. P., *Otimização via Algoritmo Genético do Processo Construtivo de Estruturas de Concreto Submetidos à Retração Restringida*. Universidade Federal do Rio de Janeiro, 2002.

- [133] LINDEN, R., *Algoritmos Genéticos: Uma Importante Ferramenta da Inteligência Computacional*. Brasport, 2006.
- [134] BLICKLE, T., THIELE, L., *A comparison of selection schemes used in genetic algorithms*. TIK-Report, 1995.