UNIVERSIDADE FEDERAL DO PIAUÍ

CENTRO DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

MESTRADO ACADÊMICO EM ENGENHARIA ELÉTRICA

EDUARDO HENRIQUE COSTA BARBOSA

APLICAÇÃO DE ALGORITMOS DE AGRUPAMENTO DE DADOS E DE REDES NEURAIS DINÂMICAS PARA A PREVISÃO DE DEMANDA EM SISTEMAS DE DISTRIBUIÇÃO DE ENERGIA

TERESINA

2019

EDUARDO HENRIQUE COSTA BARBOSA

APLICAÇÃO DE ALGORITMOS DE AGRUPAMENTO DE DADOS E DE REDES NEURAIS DINÂMICAS PARA A PREVISÃO DE DEMANDA EM SISTEMAS DE DISTRIBUIÇÃO DE ENERGIA

Dissertação de Mestrado submetida à coordenação do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Piauí, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica. Área de Concentração: Sistemas de Energia Elétrica

Orientador: Prof. Dr. Hermes Manoel Galvão Castelo Branco

TERESINA

2019

FICHA CATALOGRÁFICA
Universidade Federal do Piauí
Biblioteca Comunitária Jornalista Carlos Castello Branco
Serviço de Processamento Técnico

B238a Barbosa, Eduardo Henrique Costa.
Aplicação de algoritmos de agrupamento de dados e de redes neurais dinâmicas para a previsão de demanda em sistemas de distribuição de energia / Eduardo Henrique Costa Barbosa. – 2019.
136 f.

Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal do Piauí, Teresina, 2019.
"Orientador: Prof. Dr. Hermes Manoel Galvão Castelo Branco".

1. Combinação de agrupamentos. 2. Previsão de demanda. 3. Rede FTDNN. 4. Rede LSTM. 5. Rede NARX. I. Título.

CDD 621.3

EDUARDO HENRIQUE COSTA BARBOSA

APLICAÇÃO DE ALGORITMOS DE AGRUPAMENTO DE DADOS E DE REDES NEURAIS DINÂMICAS PARA A PREVISÃO DE DEMANDA EM SISTEMAS DE DISTRIBUIÇÃO DE ENERGIA

Dissertação de Mestrado submetida à coordenação do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Piauí, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica. Área de Concentração: Sistemas de Energia Elétrica

Aprovada em: 20 de novembro de 2019.

BANCA EXAMINADORA

Prof. Dr. Hermes Manoel Galvão Castelo Branco (Orientador)
Universidade Federal do Piauí (UFPI)

Prof. Dr. José Maria Pires de Menezes Júnior
Universidade Federal do Piauí (UFPI)

Prof. Dr. Guilherme de Alencar Barreto
Universidade Federal do Ceará (UFC)

Prof. Dr. Kelson Romulo Teixeira Aires
Universidade Federal do Piauí (UFPI)

À minha amada mãe, Maria de Fátima, por acreditar em meus sonhos e amor incondicional.

AGRADECIMENTOS

Agradeço a Deus, por me conceder paciência e resiliência.

À minha mãe, Maria de Fátima, pelo amor e por estar sempre ao meu lado, incentivando-me em todos os momentos.

Ao meu pai, Marcelino, pelo amor e por sempre acreditar em seus filhos.

Ao meu irmão, Caio, pela amizade e convivência. Aos meus avós, Creusa e Bento, e ao meu padrasto, João, pelo constante apoio e admiração.

Ao meu orientador, Professor Hermes Manoel Galvão Castelo Branco, a quem agradeço por todos os conselhos, competência, confiança, paciência e orientações, que ajudaram a tornar possível realizar este estudo.

À minha namorada, Aline Beatriz, por todo o carinho, companheirismo e por estar sempre presente nos momentos difíceis.

Agradeço aos meus professores, por todo o conhecimento que me proporcionaram, contribuindo direta ou indiretamente para a realização deste trabalho. Especialmente ao Professor José Maria, pelas sugestões e excelente conhecimento repassado durante a disciplina de Redes Neurais e ao Professor Adriano, pelo auxílio durante a obtenção dos dados e esclarecimentos em relação às características dos mesmo.

Aos colegas da Universidade Federal do Piauí. Em especial ao meu amigo Ênio, pela parceria e amizade.

À FAPEPI (Fundação de Amparo à Pesquisa do Estado do Piauí) e à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo suporte financeiro.

A todos, sou grato.

"A satisfação está no esforço e não apenas na realização final."

(Mahatma Gandhi)

RESUMO

A previsão de demanda de curto prazo é um processo importante que auxilia no planejamento da operação de sistemas elétricos e gerenciamento dos recursos energéticos. Um dos grandes problemas ao se trabalhar com a previsão de curvas demanda a partir de modelos baseados em redes neurais reside na escolha apropriada dos padrões de entrada que possam permitir às redes a captação da dinâmica do sistema. Outro problema decorre da arquitetura de rede a ser escolhida para esta finalidade, em que muitos dos estudos relacionados utilizam redes estáticas. Nesta dissertação é proposta uma metodologia para previsão de curvas diárias de demanda por meio da utilização de algoritmos de agrupamento de dados e redes neurais artificiais dinâmicas. Os algoritmos com paradigma de aprendizagem não-supervisionada são utilizados para realizar os agrupamentos das curvas de demanda. Os grupos obtidos são utilizados como insumos para o treinamento das redes dinâmicas. Para realizar os agrupamentos, foi utilizada uma estratégia que consistiu em combinar, por meio de função consenso, os resultados obtidos a partir de dois algoritmos: o *k-means* e o *minCEntropy*. As redes neurais, por sua vez, são responsáveis por realizarem a previsão das séries temporais de maneira recursiva em um horizonte de previsão de 288 passos à frente, totalizando 24 horas. Para esta finalidade é proposta a utilização de duas redes neurais dinâmicas baseadas na arquitetura da rede MLP, as redes FTDNN e NARX. Além de uma rede recorrente, que utiliza mecanismos de memória denominados "portas" (*gates*), conhecida como rede LSTM. O treinamento realizado com a utilização dos grupos obtidos se mostrou capaz de fornecer às redes neurais dados cujas características refletem as sazonalidades e periodicidades intrínsecas ao tipo de séries temporais estudadas. Além disso, foi utilizada uma metodologia que contempla a etapa de pré-processamento, evitando que possíveis dados corrompidos por eventuais problemas relacionados a distúrbios ou falhas nos equipamentos de medição/aquisição dos dados possam interferir no desempenho dos algoritmos. Para a realização do estudo, foram utilizados dados reais de uma concessionária de distribuição de energia elétrica. Os índices de desempenho NMSE e MAPE mostraram que os modelos propostos foram capazes de fornecer erros de previsão bem menores quando comparados com uma estratégia em que não se utiliza o uso de agrupamentos, podendo vir a ser uma alternativa eficiente frente a outros modelos de previsão comumente empregados.

Palavras-chave: combinação de agrupamentos. previsão de demanda. rede NARX. rede LSTM. rede FTDNN. redes neurais artificiais dinâmicas.

ABSTRACT

Short-term load forecasting is an important process that assists in planning the operation of electrical systems and managing energy resources. One of the major problems when working with forecasting demand curves from neural network-based models, is the appropriate choice of input patterns that can allow networks to capture system dynamics. Another problem stems from the network architecture to be chosen for this purpose, where many of the related studies use static networks. This dissertation proposes a methodology for forecasting daily demand curves by combining the use of data clustering algorithms and dynamic artificial neural networks. Algorithms with unsupervised learning paradigm are used to perform the demand curve clustering. The obtained groups are used as allowance for the dynamic neural networks trainning. To perform the clustering, a strategy was used to combine, by means of a consensus function, the results obtained from two algorithms: *k-means* and *minCEntropy*. The neural networks are responsible for recursively forecasting the time series, with feedback of the predicted values to the input layer, in a forecast horizon of 288 steps ahead, totaling 24 hours. For this purpose it is proposed the use of two dynamic neural networks based on the MLP neural network architecture, the FTDNN and NARX networks. In addition, a recurrent neural network that uses memory mechanisms called gates, known as LSTM network. The training performed using the obtained clusters was able to provide neural networks with data that reflect the intrinsic seasonality and periodicity of the time series studied. In addition, a methodology that contemplates the whole preprocessing step was used, avoiding that possible data corrupted by eventual problems related to disturbances or failures in the data measurement/acquisition equipment could interfere in the performance of the algorithms. To perform the study, real data from an electricity distribution utility were used. The NMSE and MAPE performance indices showed that the proposed models were able to provide much smaller prediction errors when compared to a strategy that does not use clustering, and may become an efficient alternative to other commonly predicted models employed.

Keywords: clustering ensemble. load forecasting. NARX network. LSTM network. FTDNN network. dynamic artificial neural networks.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ONS	Operador Nacional do Sistema
PMO	Programa Mensal de Operação
RNAs	Redes Neurais Artificiais
SIN	Sistema Interligado Nacional
TDNN	*Time Delay Neural Network*
FTDNN	*Focused Time Delay Neural Network*
NARX	*Nonlinear Autorregressive with Exogenous Inputs*
LSTM	*Long Short-Term Memory*
SGD	*Sthocastic Gradient Descent*
RMSprop	*Root Mean Square Propagation*
minCEntropy	*minimum conditional entropy*
MLP	*Multilayer Perceptron*
SOM	*Self-Organizing Maps*
SVM	*Support Vector Machine*
AR	*Autorregressive*
MA	*Moving Average*
ARMA	*Autorregressive Moving Average*
ARIMA	*Integrated Moving Average*
TW	*Transformada Wavelet*
TWD	*Transformada Wavelet Discreta*
TWC	*Transformada Wavelet Contínua*
MAD	*Median Absolute Deviation*

BPTT	*Back Propagation Through Time*
AMR	Análise Multi-Resolução
MSE	*Mean Squared-Error*
NMSE	*Normalised Mean Squared-Error*
MAPE	*Mean Absolute Percentage Error*
SVR	*Support Vector Regression*
LS-SVR	*Least Square - Support Vector Regression*

LISTA DE SÍMBOLOS

n	índice tempo discreto
H	horizonte de previsão
$\mathbf{x}$	vetor regressão
f	função matemática
$\widehat{f}$	aproximação da função f
$\widehat{x}$	valor escalar estimado
e	resíduo
ψ	função *wavelet*
a_0	passo fixo de dilatação
b_0	parâmetro de localização
$D_{m,n}$	coeficientes de Detalhe
$A_{m,n}$	coeficientes de Aproximação
$\phi_{m,n}$	função escala
W_L	janela deslizante
δ_n	desvio padrão
$P_{consenso}$	partição consenso
$v_j(n)$	campo local induzido do neurônio j
$w_{i,j}$	peso sináptico que conecta o neurônio i ao neurônio j
δ_j	gradiente local do neurônio j
η	taxa de aprendizagem
d_E	dimensão de imersão
τ	atraso de imersão

d_e	ordem da memória de entrada da rede NARX
d_y	ordem da memória de saída da rede NARX
f_t	*forget gate*
i_t	*input gate*
c_t	*cell state*
h_t	*hidden state*
g_t	*candidate state*

SUMÁRIO

1 INTRODUÇÃO

A previsão de demanda (também conhecida como previsão de carga) é uma ferramenta importante, usada para garantir que a energia fornecida pelas concessionárias atenda à carga, mesmo considerando as perdas de energia do sistema. Para este fim, é necessária uma equipe especializada para realizar esta função. E pode ser definida como, basicamente, a ciência ou arte de prever a carga futura em um determinado sistema, por um determinado período de tempo à frente. Essas previsões podem ser apenas por uma fração de uma hora à frente para fins de operação ou até vários anos no futuro para fins de planejamento (SOLIMAN; AL-KANDARI, 2010).

De maneira geral, a previsão de carga pode ser classificada em três tipos: curto, médio e longo prazo. Previsões de curto prazo têm por finalidade estimar o consumo de energia elétrica levando em consideração um horizonte que pode englobar desde minutos até alguns dias. As previsões de médio prazo buscam estimar o consumo que será feito em algumas semanas ou até meses. Enquanto nas de longo prazo, o objetivo é ter uma estimativa da energia que será consumida durante o período de um ou mais anos. Cada um dos tipos possui suas peculiaridades e são utilizadas de diferentes maneiras, dependendo do objetivo a ser alcançado (FEINBERG; GENETHLIOU, 2005).

As previsões de longo prazo, que podem chegar a alcançar décadas, são muito importantes do ponto de vista do planejamento da expansão dos sistemas de distribuição, como é observado no Módulo 2 do manual de Procedimentos de Distribuição de Energia Elétrica - PRODIST (ANEEL, 2016). As de médio prazo servem como base para programas de manutenção, programar a compra de energia e servir como pesquisa de mercado para os produtores e revendedores negociarem contratos com outras empresas, diminuindo os riscos financeiros. Enquanto as previsões de curto prazo desempenham papel importante na orientação do planejamento da operação, transferência de energia e gerenciamento de demanda. E são necessárias para as empresas de energia tomarem decisões relacionadas ao planejamento da geração, transmissão e distribuição da energia elétrica (MORDJAOUI *et al.*, 2017a; FEINBERG; GENETHLIOU, 2005).

Devido à inviabilidade em se armazenar a energia produzida pelas usinas geradoras, é necessário que haja a todo momento um fornecimento de energia compatível com a energia que é consumida. A existência de desequilíbrio entre geração e demanda ocasiona variações de frequência com um impacto negativo no desempenho de sistemas elétricos. Por esta razão,

desvios de frequência podem ser considerados um indicativo de desbalanço entre a energia que é gerada e a que é consumida. A fim de manter a relação de produção e consumo de acordo com os diferentes padrões existentes e garantir operações lucrativas do sistema de energia, o consumo de carga elétrica deve ser previsto e controlado a todo instante, havendo, portanto, a necessidade de se lançar mão de modelos precisos de previsão de carga (WEISSBACH; WELFONDER, 2009; MORDJAOUI *et al.*, 2017a).

Neste contexto, a previsão de curto prazo desempenha um papel orientador de modo a aperfeiçoar a utilização dos recursos de geração e transmissão do Sistema Interligado Nacional - SIN (SILVEIRA, 2010). Ainda, de acordo com o submódulo 5.4 do manual de Procedimentos de Rede, que trata da consolidação da previsão de carga para a programação diária da operação eletroenergética e para a programação de intervenções em instalações da rede de operação, os agentes devem enviar ao Operador Nacional do Sistema - ONS dados diários da previsão de carga. Além disso, no submódulo 5.1 é explicado que as concessionárias são obrigadas a fornecer o Programa Mensal de Operação - PMO ao ONS, sendo responsável por estabelecer as metas energéticas do mês, que, por sua vez, permite verificar a necessidade da utilização de usinas térmicas, a fim de balancear o uso dos recursos energéticos existentes (ONS, 2009; ONS, 2010).

Portanto, a previsão de carga é um processo importante que permite obter maior eficiência e aumentar as receita das empresas de geração e distribuição de energia elétrica, ajudando-as a planejar suas operações a fim de garantir de forma confiável o fornecimento necessário de energia a todos os consumidores. Uma previsão imprecisa pode aumentar os custos operacionais. A sobrestimação da demanda de carga resulta em uma reserva giratória desnecessária. Por outro lado, a subestimação pode causar a interrupção do suprimento de energia.

Muitos estudos têm sido desenvolvidos com a finalidade de prever curvas de carga de sistemas de energia elétrica. Modelos matemáticos tradicionais, tais como regressão linear, método de Box-Jenkins e regressão não paramétrica costumam ser tradicionalmente utilizados. Entretanto, a previsão de carga é muitas vezes difícil devido à existência de diversos fatores externos que exercem influência sobre a carga desses sistemas. Tais como condições climáticas, geográficas, sócio-econômicas e fatores aleatórios. Além disso, deve-se atentar ao fato de que as curvas de carga possuem características não-lineares e não-estacionárias, tudo isso faz com estas curvas estejam sujeitas a erros de previsão. Portanto, é de se esperar que a previsão obtida por modelos matemáticos simples possam ser imprecisas. Devido a isso, técnicas de inteligência

computacional têm ganhado importância nestes tipos de estudo ao longo dos últimos anos (ZOR; TIMUR; TEKE, 2017; HIPPERT; PEDREIRA; SOUZA, 2001a).

Estas técnicas têm mostrado eficiência em lidar com as relações não-lineares complexas, que muitas vezes são difíceis de modelar nos métodos matemáticos tradicionais e, por isso, têm se tornado cada vez mais populares. Nos estudos que envolvem a previsão de carga de curto prazo, que é foco desta dissertação, destacam-se o emprego de Redes Neurais Artificiais (RNAs) (ZHUANG *et al.*, 2016), Máquinas de Vetor de Suporte (MA; ZHOU; LIN, 2008), Lógica Fuzzy (HSU; HO, 1992) ou abordagens híbridas que combinam estas e outras técnicas (SRIVASTAVA; PANDEY; SINGH, 2016).

Em particular, as redes neurais têm sido muito utilizadas na área de previsão de séries temporais de demanda de curto prazo, devido à sua capacidade de detectar implicitamente relações complexas entre variáveis dependentes e independentes, serem aproximadores universais de funções, possuírem capacidade de aprendizado, paralelismo e robustez à presença de ruídos (DUDEK, 2016). As redes neurais empregadas para esta finalidade, geralmente consistem em redes do tipo *feedforward* com algoritmo de treinamento *backpropagation*. Entretanto, como explicado em Haykin (2009), redes neurais com algoritmo de treinamento baseado no gradiente descendente, como é o caso do *backpropagation*, apresentam dificuldade em relacionar uma resposta desejada em um instante de tempo específico e que seja dependente de outras respostas temporalmente distantes fornecidas no passado. Este problema é conhecido como *vanishing-gradient* e faz com que o aprendizado de memórias de longo prazo por tais tipos de algoritmos seja difícil e, em certas situações, até mesmo impossível (HAYKIN, 2009).

Outro aspecto que pode ser verificado nestes estudos, é a necessidade da escolha adequada de padrões de entrada que permitam à rede neural detectar a influência de outras variáveis no comportamento dinâmico, tal como a temperatura. Entretanto, neste tipo de abordagem, além de ser necessário haver a disponibilidade de uma outra variável, no caso a temperatura, o usuário também passa a depender da previsão de temperatura para o dia em que se deseja realizar a previsão de carga. Devido às sazonalidades em que são geradas as curvas de carga, a escolha de padrões adequados de treinamento se torna uma tarefa difícil. Sendo, portanto, necessário encontrar uma forma de repassar tal informação à rede neural.

A escolha adequada de padrões de entrada para uma rede neural é uma tarefa relativamente complexa, tendo em vista que tais padrões devem fornecer informações dinâmicas que tornem a rede apta a realizar, de maneira satisfatória, a previsão da curva para o dia ou dias

seguintes. Ou seja, é necessário que os dados fornecidos possuam um comportamento dinâmico semelhante à curva de carga que se deseja prever. Uma abordagem que pode ser utilizada, é a escolha de um conjunto de treinamento composto por curvas históricas correspondentes ao mesmo dia que se deseja realizar a previsão. Entretanto, tal abordagem está sujeita à escolha de padrões que podem não refletir o comportamento da série que se deseja prever, devido por exemplo: a ocorrência de eventos comemorativos ou feriados que podem alterar o comportamento do consumidor e, portanto, modificar a dinâmica da curva de carga, condições climáticas e, também, a existência de curvas poluídas por fatores aleatórios como faltas, desligamentos, etc. Portanto, a escolha de curvas sujeitas a tais tipos de problemas pode influenciar negativamente na capacidade de previsão do modelo utilizado. Soma-se a isso o fato de, não necessariamente, a curva a ser prevista possuir comportamento semelhante às curvas históricas de dias temporalmente próximos, como por exemplo aquelas geradas durante uma mesma semana ou mês.

Levando-se em consideração tais aspectos, seria útil dispor de uma ferramenta capaz de fornecer às RNAs um conjunto de dados de treinamento que possuíssem maior semelhança com os padrões de demanda a serem previstos. Portanto, nesta dissertação, é utilizada uma metodologia que consiste em fornecer às RNAs dados de treinamento obtidos a partir do agrupamento das curvas históricas de carga, de forma que, curvas com dinâmicas semelhantes à ser prevista possam ser utilizadas para se ajustar os pesos sinápticos das redes neurais na etapa de treinamento. Ou seja, é escolhido um conjunto reduzido de curvas (amostras de treinamento) que, por sua vez, compõem um dos grupos obtidos pelos algoritmos de agrupamento. Soma-se a isso, o fato destes algoritmos serem capazes de fornecer informações muito relevantes quanto às características e dinâmica dos dados sob estudo.

Embora não seja o foco deste trabalho, seria possível criar, a partir da metodologia proposta, um sistema especialista capaz de escolher o melhor grupo, capaz de prover os melhores dados de treinamento ao modelo preditor utilizado, por meio de uma escolha adequada de atributos de entrada que reflitam as peculiaridades em comum entre o grupo de treinamento e a possível curva de demanda a ser prevista.

Do ponto de vista da qualidade dos dados históricos das curvas de carga, ressalta-se que em diversas situações, alguns ou muitos desses dados podem estar corrompidos devido a problemas oriundos dos equipamentos de medição, distúrbios elétricos, ou existência de faltas que fazem com que a curva diária registrada esteja poluída por fenômenos que não refletem a dinâmica original da carga ao longo do dia. Em uma situação em que se deseja utilizar de

algoritmos de aprendizagem, como no caso das RNAs, é necessário que haja uma correção ou remoção destas curvas do conjunto utilizado durante a aprendizagem das redes, de forma que as mesmas não estejam suscetíveis a incorporar tais problemas durante o ajuste de seus pesos sinápticos.

Com a finalidade de se evitar que tais dados possam prejudicar a capacidade de aprendizagem das redes neurais e o desempenho dos algoritmos de agrupamento, também é utilizada uma metodologia com base em Transformada *Wavelet* e Identificador de Hampel capaz de filtrar a maioria destas curvas, evitando que as mesmas venham a fazer parte do conjunto de dados sob análise. Outros estudos também utilizam metodologia semelhante como meio de encontrar variações anormais de sinais, incluindo aqueles cujo objetivo principal é o tratamento de dados de curvas de carga, como pode ser observado nos trabalhos de Gonçalves (2018), Costa (2010), Oliveira (2013), Silva, Souza e Brito (2006) e Andrade (2017)

Além disso, é importante destacar o uso de redes neurais dinâmicas aplicadas ao problema da previsão de carga em curto prazo, visto que uma grande parte dos estudos realizados visam a aplicação de redes do tipo Perceptron Multicamadas (*MultiLayer Perceptron* - MLP). Portanto, é proposto o uso da rede NARX (*Nonlinear Autorregressive with Exogenous Inputs*) adaptada por Menezes Júnior e Barreto (2008) que consiste em uma versão modificada da arquitetura original que, por sua vez, utiliza uma variável exógena como entrada. No modelo utilizado nesta dissertação, a rede NARX é adaptada para trabalhar com séries univariadas, contendo regressores que refletem memórias de curto e longo prazo, conforme proposto em Menezes Júnior e Barreto (2008). Outro modelo utilizado, consiste em uma rede LSTM (*Long Short-Term Memory*), que é capaz de reter informações relevantes em relação ao sistema a ser previsto por meio de unidades internas conhecidas como *gates* (portas), sejam estas informações de curta ou longa dependência (HOCHREITER; SCHMIDHUBER, 1997).

Ambas as arquiteturas, apesar de possuírem grande aplicação na área de previsão de séries temporais, foram pouco utilizadas na previsão de curvas de carga. Em relação ao modelo NARX utilizado, não verificou-se na literatura sua aplicação neste tipo de problema. Enquanto o uso da rede LSTM vem ganhando algum destaque em alguns estudos recentes. Além do uso destas duas arquiteturas, também é averiguado o desempenho da rede FTDNN (*Focused Time Delay Neural Network*), composta por atrasos de tempo na camada de entrada. Todas as redes são utilizadas na tarefa de previsão recursiva, em um horizonte de 288 passos à frente, com instantes de tempo separados por intervalos de cinco minutos, o que corresponde a uma curva de

demanda diária.

Assim, espera-se que os algoritmos de agrupamento utilizados sejam capazes de fornecer às RNAs utilizadas dados que reflitam as características sazonais das curvas de carga, facilitando o ajuste de seus pesos e evitando a possibilidade das mesmas serem submetidas a dados de treinamento que não possuem relação com a curva a ser prevista, facilitando até mesmo a previsão de dias atípicos, como feriados.

O grande potencial existente nas RNAs dinâmicas em tarefas que envolvem previsões de séries temporais, e a verificação de melhorias que ainda podem ser acrescentadas à esta área, motivaram a realização desta dissertação cujos objetivos são descritos a seguir.

1.1 Objetivos do Estudo

O objetivo geral desta dissertação consiste em propor uma metodologia capaz de realizar, com acurácia e precisão, a previsão recursiva de 288 passos à frente de curvas de demanda de energia elétrica com discretização de 5 minutos, totalizando 24 horas, por meio da utilização de redes neurais dinâmicas e algoritmos de agrupamento.

1.1.1 Objetivos Específicos

Visando-se alcançar o objetivo descrito acima, uma série de objetivos específicos foram alcançados:

- Filtrar os sinais para que fosse evitado que curvas corrompidas por graves problemas de medição ou distúrbios elétricos fizessem parte do conjunto de dados sob análise e pudessem comprometer o desempenho dos algoritmos de agrupamento e das RNAs. Para esta finalidade foram utilizados o Identificador de Hampel e Transformada *Wavelet*;
- Realizar o agrupamento das séries temporais por meio dos algoritmos *k-means* e *minCEntropy* e obtenção das partições base;
- Aplicar algoritmo de combinação de agrupamentos baseado em votação nas partições base oriundas dos algoritmos *k-means* e *minCEntropy*;
- Realizar estudo estatístico dos agrupamentos obtidos em relação à distribuição das séries temporais levando-se em consideração grupos, dias, meses e anos em que as séries foram geradas;
- Fazer a previsão das curvas de carga por redes neurais dinâmicas utilizando-se como

critérios de treinamento e previsão grupos previamente compostos pelos algoritmos de agrupamento;

- Comparar as três arquiteturas de redes neurais dinâmicas aplicadas ao problema de previsão recursiva de curvas de carga;
- Comparar a metodologia proposta frente a outras duas alternativas de previsão: redes neurais sem a utilização de algoritmos de agrupamento e a média simples de curvas geradas em dias anteriores.

1.2 Trabalhos Relacionados

Nesta seção são discutidos alguns trabalhos recentes que envolvem a previsão e/ou agrupamento de curvas de carga.

No estudo de Bento *et al.* (2019), os dados submetidos à rede neural são previamente selecionados com base em critérios de similaridade e filtrados por meio do uso de Transformada *Wavelet*. Os parâmetros da rede neural, por sua vez, são otimizados por meio do uso de um algoritmo bio-inspirado (algoritmo de morcegos). A previsão é feita para um intervalo de um dia, composto por 24 amostras.

Rana e Koprinska (2016) utilizam a Transformada *Wavelet Packet* para decompor as curvas de carga em componentes de baixa e alta frequência. Os dados são submetidos aos treinamentos das redes neurais de forma que sejam capazes de prever cada uma destas componentes de forma separada e, a partir destas componentes, o sinal predito possa ser reconstruído. Também é utilizada a Informação Mútua para extração de características. Os resultados mostraram que o método ofereceu bom desempenho frente a outras técnicas, com a previsão variando de 1 a 12 passos à frente. Aplicações semelhantes de Transformada *Wavelet* também podem ser verificadas nos estudos de Reis e Silva (2005), Chen *et al.* (2009) e Amjady e Keynia (2009).

Um outro exemplo da aplicação da Transformada *Wavelet* foi feito por Pandey, Singh e Sinha (2010), em que os autores decompõem as curvas de carga em componentes de baixa e alta frequência, em seguida estas últimas são retiradas, resultando em uma versão suavizada do sinal. Segundo os autores, as curvas de carga sem estas componentes resultam em previsões mais precisas com o uso redes neurais.

Sadaei *et al.* (2019) utilizam redes neurais convolucionais para a previsão de carga de curto prazo, utilizando séries multivariadas, que incluem valores horários de carga e temperatura e versões *fuzzyficadas* das séries de curva de carga. Estes valores sequenciados foram convertidos

em imagens multicanais e utilizados como dados de treinamento pela rede neural. Os resultados mostraram que diante do conjunto de dados sob análise, a metodologia proposta ofereceu baixos erros de previsão quando comparada a outros tipos de modelos.

Saviozzi, Massucco e Silvestro (2019) utilizaram uma combinação de resultados de previsão oriundos de redes neurais do tipo MLP. O horizonte de previsão é de 24 horas, amostradas a cada 15 minutos. Singh e Dwivedi (2018) também utilizam a MLP, mas neste caso os autores propõem um algoritmo evolucionário para realizar a otimização dos pesos sinápticos da rede a fim de minimizar os erros de previsão. O algoritmo proposto é testado em curvas horárias de consumo de energia. Amjady e Keynia (2009) também fizeram uso de um algoritmo evolucionário para ajuste dos pesos sinápticos da rede para que as previsões possam se aproximar de um ótimo global.

Cabe destacar que uma parte considerável destes estudos utilizam a MLP como modelo empregado para previsão. Como outros exemplos é possível citar os trabalhos de Ding *et al.* (2015), Ekonomou (2010) e Katsatos e Moustris (2019).

O uso de redes FTDNN tem sido observado em outras aplicações. Vargas, Paredes e Bustos (2010) fizeram uma previsão focada na potência gerada por parques eólicos e os valores futuros são previstos utilizando-se uma rede FTDNN, obtendo os melhores resultados de previsão dentre os outros modelos analisados. A FTDNN também é utilizada por Al-Messabi *et al.* (2012) para se prever a potência gerada por módulos fotovoltaicos, os resultados mostraram boa performance para previsões de até 6 passos à frente, equivalente a um período de 60 minutos. E no estudo de Menezes Júnior (2006) é visto que esta rede também é utilizada na modelagem de séries temporais caóticas e comparada junto a outras arquiteturas de redes neurais.

Alguns trabalhos utilizam variáveis exógenas, como temperatura, para se realizar a previsão de valores futuros de curvas de carga. Abordagens multivariadas muitas vezes utilizam a rede NARX como modelo, que possibilita a utilização de duas ou mais séries temporais como variáveis de entrada. Buitrago e Asfour (2017) utilizaram uma rede NARX para a previsão de carga de curto prazo. A rede é treinada utilizando dados reais de carga e temperatura. Embora outros trabalhos utilizem variáveis exógenas como auxílio na previsão de carga, verifica-se que as arquiteturas utilizadas consistem em redes do tipo MLP, onde identifica-se uma tarefa semelhante à identificação de padrões, em que a rede não se comporta de maneira dinâmica, estando a rede apta apenas a trabalhar em função de valores estáticos, não realizando as previsões em função dos valores previstos pela própria rede (PANAPAKIDIS, 2016b).

Nesta dissertação, as redes FTDNN, NARX e LSTM se comportam de maneira autônoma, estando os valores previstos em função de valores passados e previamente estimados pelas redes neurais. Ou seja, as próprias saídas das redes são utilizadas como entradas. Destaca-se, ainda, que no caso da rede NARX, a mesma é utilizada de forma a trabalhar com séries temporais univariadas, de acordo como o proposto por Menezes Júnior e Barreto (2008), portanto, não existe a presença de uma variável exógena, aplicada ao problema de previsão de curvas de carga.

De maneira geral, poucos trabalhos encontrados na revisão bibliográfica realizada no desenvolvimento desta pesquisa utilizam redes dinâmicas com a finalidade de prever valores futuros de curvas de carga de sistemas elétricos e isso também inclui a rede LSTM. Somente recentemente alguns estudos têm voltado sua atenção para esta aplicação com o uso deste tipo de rede neural. Nos estudos de Kong *et al.* (2018) e Wang *et al.* (2019) a LSTM é utilizada para se realizar a previsão de curvas de carga de unidades residenciais. Por outro lado, Qing e Niu (2018) utilizaram a LSTM na previsão horária de irradiação solar, enquanto Sagheer e Kotb (2019) a aplicaram como modelo de previsão de produção de petróleo. Em todos os casos a rede apresentou bom desempenho.

Outros estudos visam o agrupamento de curvas de carga com a finalidade de verificar os perfis de consumo apresentados com base na similaridade das curvas de carga. Panapakidis, Alexiadis e Papagiannis (2013) utilizaram métodos que permitem identificar a quantidade ótima de grupos em que curvas de carga de consumidores podem ser divididas. Para a tarefa de agrupamento foram utilizados os algoritmos *Fuzzy C-means* e *Minimum Variance Method*.

Panapakidis, Alexiadis e Papagiannis (2015) realizaram um estudo sobre os padrões de consumo de um consumidor industrial de alta tensão. O processo envolve o uso de algoritmos de agrupamento bem conhecidos na literatura e avaliação considerando-se diversos índices de desempenho. Verificou-se que o algoritmo *minCEntropy* (*Minimum Conditional Entropy*) foi capaz de apresentar os melhores resultados na maioria dos casos.

Jiang *et al.* (2018) propuseram o uso de um algoritmo baseado em Transformada *Wavelet*. Esta última foi utilizada para realizar a diminuição da dimensionalidade e, ao mesmo tempo, manter a informação original dos dados, decompondo os sinais em coeficientes de aproximação e detalhe. Em seguida, agrupamentos foram identificados tanto nas aproximações dos sinais, quanto nos coeficientes de detalhe. Posteriormente, pares de grupos oriundos dos dois agrupamentos foram combinados com a finalidade de se obter um agrupamento otimizado.

O *k-means* foi utilizado para a tarefa de agrupamento. Já Mets, Depuydt e Develder (2016) aplicaram a Transformada *Wavelet* Rápida para reduzir a dimensão dos vetores de entrada, enquanto *g-means* foi utilizado como algoritmo de agrupamento. Também é possível observar o uso da Transformada *Wavelet* em tarefas de agrupamento de carga nos trabalhos de Li, Li e Smith (2016) e Gonçalves (2018).

Lin *et al.* (2017) realizaram uma classificação de curvas de carga baseada na entropia, aproximação agregada por partes (*Piecewise Aggregate Approximation* - PAA) e agrupamento espectral. Os dados de carga diária foram modelados por meio da técnica de resolução temporal variável, em seguida o agrupamento espectral foi aplicado.

Outros estudos propõem a utilização conjunta de algoritmos de agrupamento e algoritmos supervisionados, como é o caso da maioria das redes neurais, com a finalidade de se obter modelos de previsão mais precisos. Pois as instâncias contidas em um mesmo grupo possuem maior similaridade entre si, quando comparadas com o restante dos dados. De modo que possa haver um algoritmo de previsão especializado para cada grupo existente. Ou seja, os algoritmos tornam-se aptos a prever determinados padrões de consumo.

Como exemplo, Mares, Mercado *et al.* (2017) utilizaram Mapas Auto-organizáveis de Kohonem (*Self-Organizing Maps* - SOM) para categorizar as curvas de carga a serem previstas por redes neurais do tipo MLP. Nagi *et al.* (2011) aplicaram a SOM em conjunto com Máquina de Vetor de Suporte (*Support Vector Machine* - SVM) para prever valores de pico de carga de médio prazo, para o mês seguinte. Wu e Peng (2017) agruparam curvas de potência de geradores eólicos por meio do algoritmo *k-means*, levando em consideração dados meteorológicos e histórico das curvas de potência. A previsão foi obtida por meio da combinação de resultados obtidos por redes neurais do tipo MLP. Nos estudos de Panapakidis (2016a) e Panapakidis (2016b), o algoritmo *minCEntropy* é utilizado para se realizar o agrupamento das curvas de carga, a previsão é feita por meio de redes MLP. E Fu *et al.* (2018) empregou o algoritmo *Fuzzy C-Means* para se realizar o agrupamento de curvas de carga residenciais, a previsão por sua vez foi realizada com redes neurais *fuzzy* auto-organizáveis (*Self-organising Fuzzy Neural Networks* - SOFNN).

1.3 Contribuições da Pesquisa

As principais contribuições desta dissertação consistem em:

- fornecer uma metodologia que contempla desde a etapa de pré-processamento dos dados até a previsão das curvas de carga;

- realizar a previsão com base em similaridades existentes entre os padrões das curvas de carga, de forma a refletir as sazonalidades, periodicidades e outras relações existentes entre os dados;
- fornecer uma metodologia de agrupamento de curvas de carga que consiste na combinação de dois ou mais algoritmos, já que não existe na literatura um consenso sobre qual o melhor método a ser utilizado;
- mostrar a eficiência que redes neurais dinâmicas podem oferecer frente ao problema de previsão de carga de curto prazo, já que grande parte dos trabalhos existentes utilizam a MLP como algoritmo mais empregado para esta tarefa;
- realizar uma previsão com bom nível de precisão, mesmo considerando um horizonte de previsão de 288 passos à frente, característica ainda não explorada na literatura.

1.3.1 Produção Científica

Durante o período de desenvolvimento desta dissertação o artigo intitulado *Critical analysis of pattern recognition load curves using multi-layer perceptron neural network* foi publicado no *13th IEEE International Conference on Industry Applications* - INDUSCON (BARBOSA *et al.*, 2018). Neste artigo foi utilizada uma rede neural do tipo MLP para a classificação de curvas de carga. Diferentemente de outros estudos, foi utilizada uma abordagem com treinamento supervisionado. Verificou-se que mesmo ao se estipular as classes a que cada instância do conjunto de dados deveria pertencer, o algoritmo as classificava de acordo com as suas similaridades, chegando-se à conclusão que curvas de carga geradas em um mesmo dia da semana podem possuir características completamente distintas. Podendo esta ser uma metodologia que poderia auxiliar na validação de resultados obtidos por algoritmos não-supervisionados ou ser utilizada em conjunto com os mesmos.

1.4 Organização da Dissertação

Além desta introdução, esta dissertação encontra-se dividida em outros sete capítulos. No Capítulo 2 são introduzidos alguns conceitos sobre séries temporais e curvas de carga e como é abordado o problema da previsão de séries temporais.

No Capítulo 3 são apresentados os dois algoritmos de processamento de sinais utilizados durante a etapa de tratamento dos dados. São descritos alguns conceitos essenciais

sobre a Transformada *Wavelet* e Identificador de Hampel.

Em seguida, no Capítulo 4, são introduzidos e descritos os algoritmos de agrupamento utilizados: o *k-means* e *minCEntropy*. Ressaltando algumas vantagens da combinação de algoritmos de agrupamento e descrição da função consenso utilizada.

O Capítulo 5 faz uma descrição geral sobre alguns aspectos importantes das redes neurais artificiais. Explicando a rede MLP, da qual deriva-se as arquiteturas das redes FTDNN e NARX. Em seguida dá-se ênfase às redes FTDNN, NARX e LSTM, que consistem em redes neurais dinâmicas com maior aptidão a captar a dinâmica de séries temporais.

Os Capítulos 6 e 7 descrevem, respectivamente, a metodologia adotada e os resultados obtidos nesta dissertação. Incluindo como é feita toda a etapa de pré-processamento dos sinais, agrupamento e previsão das curvas de carga. No Capítulo 7 são explicitados os resultados oriundos tanto do processo de agrupamento, quanto da etapa de previsão propriamente dita.

Por último, no Capítulo 8, são apresentadas as conclusões obtidas e as possibilidades para trabalhos futuros.

2 CURVAS DE CARGA NO CONTEXTO DA ANÁLISE DE SÉRIES TEMPORAIS

2.1 Introdução

A previsão de curvas de carga consiste em um problema de análise de séries temporais. Devido a isso, neste capítulo são abordados alguns aspectos fundamentais que envolvem este tipo de análise e que, devido à natureza do problema, podem ser utilizados na tarefa de previsão de carga de curto prazo. O modo como é feita a abordagem do problema da previsão de séries temporais descrita neste capítulo é de fundamental importância, pois os modelos de RNAs utilizados nesta dissertação possuem como base a informação temporal presente nas curvas de carga e atuam em função dos valores passados observados.

Uma série temporal pode ser considerada como um conjunto de observações sequencialmente ordenadas no tempo. Séries temporais podem ser frequentemente encontradas em diversos âmbitos, como negócios, economia, indústria, engenharia e ciência. Alguns exemplos bastante comuns são: consumo diário de energia elétrica, crescimento populacional anual e temperatura horária ao longo de um dia. Como pode ser observado, todos os exemplos citados possuem em comum o fato das variáveis sob análise se darem em função do tempo. Entretanto, como aponta Morettin e Toloi (2006), exitem séries em função de outros parâmetros físicos, como espaço ou volume. Desta forma, a relevância existe na regularidade em que as variáveis são medidas.

Uma série temporal que contenha apenas observações de uma única variável é definida como uma série temporal univariada. Por outro lado, quando a análise envolve mais de uma variável, coletadas durante um mesmo intervalo de tempo, diz-se que a série é multivariada. Dependendo da forma como são medidas, podem ser classificadas como de natureza discreta ou contínua. No primeiro caso, as observações são coletadas em intervalos discretos de tempo, enquanto no segundo, são medidas a cada instante de tempo, de maneira ininterrupta durante um intervalo de tempo qualquer. Na prática, uma série discreta é obtida a partir de uma série temporal contínua, por meio de sua amostragem em intervalos de tempo, Δt, iguais. Entretanto, existem casos em que o valor da série discreta, em um determinado instante, é obtido por meio do acúmulo de valores, da série contínua, em intervalos de tempos iguais (MORETTIN; TOLOI, 2006; BROCKWELL; DAVIS; CALDER, 2002). Como exemplos de séries temporais contínuas é possível citar: leituras de temperatura, do fluxo de um rio, da concentração de um processo químico, etc. Todas podem ser registradas de maneira contínua.

Diferentemente das séries provenientes de dados sintéticos, as variáveis que constituem as séries observadas no mundo real podem ser influenciadas principalmente por quatro tipos de fatores, que podem ser separados dos dados observados, sendo eles: tendência, componentes cíclicos, sazonais e irregulares (ADHIKARI; AGRAWAL, 2013; HEIZER; RENDER, 2008).

A propensão de uma série temporal em aumentar, diminuir ou estagnar durante um intervalo de tempo considerado longo é chamada tendência secular ou, apenas, tendência. Assim, pode-se considerar que a tendência é uma dinâmica de longo prazo em uma série temporal. Por exemplo, séries relacionadas ao crescimento populacional, número de casas em uma cidade, etc., mostram uma tendência ascendente, enquanto a tendência de queda pode ser observada em séries relacionadas a taxas de mortalidade, epidemias, etc. (ADHIKARI; AGRAWAL, 2013).

A variação cíclica descreve as mudanças de médio prazo na série, causadas por circunstâncias, que se repetem em ciclos. A duração de um ciclo se estende por um período maior, geralmente dois ou mais anos. A maioria das séries econômicas e financeiras mostra algum tipo de variação cíclica. Por outro lado, as variações sazonais são flutuações que ocorrem durante um período de tempo específico devido a determinados tipos de fatores que exercem influência sobre a variável observada. As condições climáticas, hábitos tradicionais e estações do ano são alguns exemplos destes fatores. Por exemplo, é comum a venda de aparelhos de ar-condicionado aumentar durante as épocas do ano em que as temperaturas são mais elevados, o mesmo ocorre em relação ao consumo de energia, devido ao aumento do número destes aparelhos em funcionamento. A variação sazonal é um fator importante para empresários, lojistas e produtores pois possibilita a realização de planos futuros adequados, com base nas variações sazonais esperadas (ADHIKARI; AGRAWAL, 2013).

Por último, as variações irregulares ou aleatórias em uma série temporal são causadas por influências difíceis ou muitas vezes impossíveis de serem previstas, que não são regulares e geralmente não demonstram um padrão de repetição específico. Tais variações são causadas por diversos tipos de incidências, como guerras, greves, terremotos, inundações, falhas de equipamentos, etc. Não há técnica estatística definida para medir flutuações aleatórias em uma série temporal (ADHIKARI; AGRAWAL, 2013).

Nas próximas seções são descritos alguns aspectos importantes sobre curvas de carga, a previsão de séries temporais e alguns modelos matemáticos típicos empregados como ferramentas para esta tarefa.

2.2 Curvas de Carga

As curvas de carga variam de acordo com diversos fatores controláveis ou não, como por exemplo o comportamento do consumidor, fatores econômicos e climáticos. Sendo assim, esses fatores devem ser estudados e levados em consideração no estudo das curvas de carga. Segundo a resolução ANEEL nº 800 (2017), os consumidores são definidos conforme a seguir:

- residencial: aquela unidade com fim residencial, dividida em diferentes subclasses de acordo com as condições sócio-econômicas apresentadas;
- industrial: unidade em que se desenvolve atividade industrial, inclusive transporte de matéria-prima, insumo ou produto resultante do seu processamento;
- comercial, serviços e outras utilidades: unidade em que seja exercida atividade comercial ou de prestação de serviços;
- rural: unidade localizada em área rural e que executa atividade rural.
- poder público: unidade que independentemente da atividade a ser desenvolvida é de pessoa jurídica de direito público;
- iluminação pública: iluminação de ruas, praças, avenidas, túneis, e demais logradores de domínio público;
- serviço público: fornecimento para equipamentos e cargas essenciais à operação de serviços públicos.

Este tipo de classificação é importante em estudos de planejamento, pois torna possível a identificação de hábitos de consumo e instantes em que podem haver maiores variações de tensão (como as causadas por partidas de motores) ou de maior demanda, por exemplo (KAGAN; OLIVEIRA; ROBBA, 2005).

Pode-se ainda considerar a classificação dos consumidores de acordo com a área de localização atendida pelo sistema de distribuição em zonas, tais como: zona urbana, suburbana e rural. Nas áreas centrais da zona urbana caracterizadas geralmente pelo grande volume de edificações e, portanto, grande densidade de carga, os consumidores são geralmente constituídos por escritórios e estabelecimentos comerciais, em que os períodos de funcionamento e hábitos de consumo são, na maioria das vezes, comuns a todos eles. Como a maior parte destas áreas é completamente edificada, é difícil o surgimento de novos consumidores, resultando apenas no aumento do número de equipamentos elétricos utilizados, sendo, portanto, um crescimento vegetativo. Por outro lado, nas áreas mais distantes da área central, a densidade de carga é geralmente menor, com hegemonia de consumidores residenciais, embora ainda possa existir

a ocorrência de consumidores comerciais e industriais. E, na áreas rurais, nota-se a presença de baixa densidade de carga, de predominância residencial e agro-industrial, cujos hábitos de consumo costumam diferir dos demais tipos de consumidores (KAGAN; OLIVEIRA; ROBBA, 2005).

Outro comportamento que interfere nas curvas de carga, são aqueles ligados às condições climáticas, ou seja, sazonais. Para o setor comercial, são dois os fatores que determinam a sazonalidade: a variação da produção industrial ao longo do ano e a estrutura do parque industrial (OLIVEIRA; SILVEIRA; BRAGA, 2000).

Para o setor comercial, as evidências mostram que são duas as variáveis explicativas principais da sazonalidade: a temperatura e a atividade econômica; causas reforçadas pelo consumo ser maior no verão, devido tanto às maiores temperaturas quanto às festas de final de ano, às férias e ao turismo; e menor no inverno, pelas menores temperaturas e diminuição da atividade econômica (OLIVEIRA; SILVEIRA; BRAGA, 2000).

Já no setor residencial, o fator preponderante é o fato de haver consumos maiores no verão e menores no inverno. Podendo-se observar também que em regiões onde as variações de temperatura não são significativas, como por exemplo o Nordeste, as curvas também não tem variância significativa durante o ano todo. Porém, no sul, sudeste e centro-oeste a variação é bem significativa (OLIVEIRA; SILVEIRA; BRAGA, 2000).

Diante de tais aspectos, e em conformidade com normas técnicas, a demanda de uma instalação é a carga nos terminais receptores medida em valor médio em um determinado intervalo de tempo. Neste caso, entende-se por "carga" a grandeza que é medida, podendo estar em valores de potência aparente, ativa ou reativa, ou, até mesmo, em valor eficaz de corrente, de acordo com a conformidade. O período levado em consideração para medir o valor médio é definido como intervalo de demanda e à medida em que o mesmo tende a zero é possível se obter a demanda instantânea. Desta forma, dependendo da aplicação, é possível se obter, em um determinado período (como, por exemplo, um dia), a curva da demanda instantânea em função do tempo, também chamada de curva de carga diária. Caso a demanda seja dada em termos de potência ativa, a área sob a curva será correspondente à energia diária consumida (KAGAN; OLIVEIRA; ROBBA, 2005).

2.3 Previsão de Séries Temporais

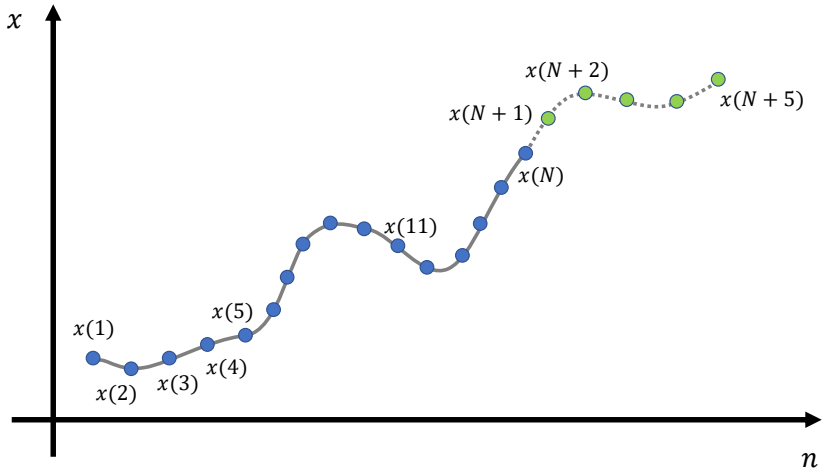
A análise de séries temporais tem por objetivo investigar os seguintes aspectos (MORETTIN; TOLOI, 2006):

- investigar o mecanismo gerador da série temporal, ou seja, a partir da observação de uma dada variável procurar saber que tipo de fenômeno ou situação a gerou;
- descrever o comportamento da série, útil quando existe a necessidade de entender a dinâmica apresentada pela série, assim como os fatores de característica cíclicas, sazonais e aleatórias que podem exercer influência sobre a série;
- procurar periodicidades importantes nos dados;
- fazer previsão de valores futuros da série, que podem ser de curto ou de longo prazo dependendo do tipo de série que se deseja prever.

Dentre os aspectos citados, a previsão de valores futuros é de grande importância devido à sua aplicação em diversas áreas, e envolve aquisição de dados históricos sobre a variável em estudo para, a partir disso, realizar a projeção de seus valores no futuro, por meio do uso de algum modelo matemático adequado (HEIZER; RENDER, 2008). Na prática, um modelo é ajustado a uma determinada série temporal e os parâmetros correspondentes são estimados usando os valores de dados conhecidos (HIPEL; MCLEOD, 1994). Por meio desta análise é possível encontrar modelos que tentam entender a natureza da série e são frequentemente úteis para previsão e simulações futuras. Os eventos futuros são então previstos usando este modelo. Esta abordagem é particularmente útil quando não há muito conhecimento sobre o padrão estatístico seguido pelas observações sucessivas ou quando há falta de um modelo explicativo satisfatório (ADHIKARI; AGRAWAL, 2013).

Menezes Júnior (2012) define bem o problema de previsão de séries temporais, alguns de seus principais aspectos são descritos a seguir. Matematicamente uma série temporal é representada por uma sequência de valores referentes a uma variável $x \in \mathbb{R}$, em que cada intervalo de tempo existe uma amostra n, $\{x(1),\ x(2),\ x(3),\ ...,\ x(N)\}$, sendo N o número total de amostras observadas, tal que $\{x(n)\}_{n=1}^{N}$. A previsão, portanto, consiste em se encontrar as amostras correspondentes a instantes de tempo futuros $\{x(N+1),\ x(N+2),\ ...,\ x(N+H)\}$ a partir dos N valores passados, em que H é o horizonte de previsão que se deseja alcançar. Como dito anteriormente, é necessária a construção de um modelo que dará origem às observações futuras, de forma semelhante ao que ocorre na identificação de sistemas. Entretanto, neste último caso é necessária adicionalmente uma variável exógena, para se estimar um instante de tempo à

Figura 1 – Exemplo de série temporal e previsão de seus valores futuros.



Fonte – O autor.

frente $\{x(N+1)\}$. A Figura 1 ilustra um exemplo de uma série temporal da qual se dispõe de N amostras, por meio destas é possível estimar seus próximos cinco valores futuros ($x(N+5)$).

Desta forma, a previsão de valores futuros de uma série temporal pode ser obtida a partir de um vetor regressão

$$\mathbf{x}(n) = [x(n),\ x(n-1),\ x(n-2),\ \ldots,\ x(n-d+1)], \tag{2.1}$$

em que d é o número de observações pelo qual é formado o regressor $\mathbf{x}(n)$, assim um instante futuro pode ser obtido em função de $\mathbf{x}(n)$

$$x(n+1) = f(\mathbf{x}(n)), \tag{2.2}$$

sendo $f(\cdot)$ uma função linear ou não-linear. Portanto um modelo empregado para a finalidade de encontrar $x(n+1)$ deve ser capaz de encontrar uma aproximação de $f(\cdot)$, fornecendo, portanto um valor aproximado de $x(n+1)$, ou seja:

$$\widehat{x}(n+1) = \widehat{f}(\mathbf{x}(n)), \tag{2.3}$$

sendo $\widehat{f}(\mathbf{x}(n))$ e $\widehat{x}(n+1)$ aproximações da função $f(\cdot)$ e do valor futuro $x(n+1)$, respectivamente. Por se tratar de uma aproximação, existe uma diferença entre os valores de $x(n+1)$ e $\widehat{x}(n+1)$, esta diferença é chamada de erro de previsão ou resíduo:

$$e(n) = x(n+1) - \widehat{x}(n+1). \tag{2.4}$$

Basicamente existem dois tipos de previsão. Na do tipo um passo adiante se está interessado em prever apenas o próximo valor da série temporal. Ou seja, como foi descrito,

a partir de $\mathbf{x}(n)$ obtém apenas o valor da série no instante de tempo posterior à amostra n, $x(n+1)$. Desta forma, um modelo que utiliza deste tipo de previsão prevê o próximo valor apenas baseado em valores atuais, já observados da série temporal, e não há a realimentação dos valores estimados para a entrada do modelo. De outro modo, o segundo tipo é chamado previsão de múltiplos passos adiante, ou recursiva. Neste caso, é feita a previsão H passos a diante, a partir da amostra atual, sempre com a realimentação dos valores previstos para a entrada do modelo preditor. Por exemplo, considerando-se uma série temporal qualquer, em que se dispõe única e exclusivamente de N amostras observadas, deseja-se prever os seus próximos H valores, onde $H \geq 1$ é o número de amostras no futuro que se deseja prever. Para se alcançar a H-ésima amostra é necessário realimentar os próprios valores anteriores estimados pelo preditor à sua entrada, recursivamente. A Tabela 1 ilustra o processo de previsão recursiva aqui descrito. Percebe-se que se H tende ao infinito, em certo momento, a previsão é feita somente com base nos valores estimados, e o modelo se torna portanto um sistema autônomo (MENEZES JÚNIOR, 2012).

Tabela 1 – Previsão recursiva (H-passos adiante).

Instante	Regressor	Saída
n	$x(n),\ x(n-1),\ x(n-2),\ ...,\ x(n-(d_e-1))$	$\widehat{x}(n+1)$
$n+1$	$\widehat{x}(n+1),\ x(n),\ x(n-1),\ ...,\ x(n-(d_e-2))$	$\widehat{x}(n+2)$
$n+2$	$\widehat{x}(n+2),\ \widehat{x}(n+1),\ x(n),\ ...,\ x(n-(d_e-3))$	$\widehat{x}(n+3)$
$\vdots$	$\vdots$	$\vdots$

Fonte – Adaptada de Menezes Júnior (2012)

Destaca-se que a previsão recursiva é uma tarefa muito mais complexa do que a previsão de um passo adiante, visto que até modelos ruins são capazes de realizar a previsão de um passo adiante, resultando sempre em baixos resíduos. Portanto, a previsão recursiva é um bom modo de verificar se o modelo adotado foi capaz de captar a dinâmica apresentada pelo sistema que se deseja prever (MENEZES JÚNIOR, 2012).

Nos últimos anos, as redes neurais têm sido cada vez mais empregadas como modelos eficazes para a tarefa de previsão de séries temporais, incluindo as do tipo analisadas nesta dissertação, referentes às curvas de carga (DUDEK, 2016; MORDJAOUI *et al.*, 2017b).

As redes neurais consistem em uma excelente ferramenta para a modelagem na análise de séries temporais devido à sua característica inerentemente não-linear, o que facilita sua adaptação a séries que possuam tal tipo de característica, mesmo sem haver, a princípio,

nenhum conhecimento estatístico sobre a distribuição das observações que compõem a série temporal. O modelo apropriado é adaptativamente formado com base nos dados fornecidos. Por esta razão, as RNAs são orientadas por dados e auto-adaptáveis por natureza (ADHIKARI; AGRAWAL, 2013). Muitas pesquisas envolvendo aplicação de redes neurais para modelagem e previsão de séries temporais têm sido realizadas (LEI *et al.*, 2009; HIPPERT; PEDREIRA; SOUZA, 2001b). Embora este trabalho utilize redes neurais como método principal de previsão, muitos estudos e empresas ainda utilizam modelos tradicionais como método de previsão, apesar de suas limitações envolvendo a capacidade de lidar com não-linearidades. Na próxima seção são discutidos brevemente alguns destes métodos. Posteriormente, no Capítulo 5 é dado ênfase às redes neurais e sua utilização no problema de previsão de séries temporais.

2.4 Métodos Baseados em Modelos Matemáticos

2.4.1 Média Móvel Simples

O método das médias móveis utiliza uma certa quantidade de dados históricos para gerar as previsões. Matematicamente pode ser expresso por

$$\widehat{x}(n+1) = \frac{x(n) + x(n-1) + x(n+2),\ \ldots,\ x(n-R+1)}{R}, \tag{2.5}$$

em que R indica quantidade disponíveis de amostras mais recentes para se prever $\widehat{x}(n+1)$. Quanto maior for o valor de R, maior será o amortecimento e caso R seja igual a 1, o valor da previsão será igual ao último valor observado. Este último caso é o tipo de previsão mais simples que existe, e é também conhecido como método ingênuo. Entretanto, quando R é igual ao número total de amostras observadas N, a previsão se torna a média aritmética de todos os valores observados, sendo indicada apenas caso a série seja composta por muitos valores aleatórios.

A cada período, a observação mais antiga é substituída pela mais recente, daí o nome média móvel. Presume-se, então, que todos os valores futuros são dados em termos da média móvel calculada previamente. A simplicidade de aplicação e a possibilidade de ser aplicável mesmo em situações em que se dispõe de poucas amostras são suas principais vantagens. Entretanto, deve-se atentar ao fato que é aconselhável o seu uso apenas para previsão de séries estacionárias (MORETTIN; TOLOI, 2006).

2.4.2 Média Simples

A média simples, também conhecida como média aritmética, é definida como o quociente da divisão do somatório dos valores das amostras de uma determinada variável pela quantidade de observações disponíveis (MAGALHÃES; LIMA, 2002).

No contexto das séries temporais, a média simples entre duas ou mais séries, seria o somatório entre cada uma das amostras de um mesmo instante tempo, dividido pelo número de séries que compõem o somatório. Considerando-se uma curva de demanda X composta por N amostras, ou seja: $X = \{x(1),\ x(2),\ x(3),\ ...,\ x(N)\}$. A previsão realizada por meio da média simples entre uma quantidade S de curvas contidas em um conjunto de observações é dada por

$$\widehat{X} = \frac{\sum_{s=1}^{S} \{x_s(1),\ x_s(2),\ x_s(3),\ ...,\ x_s(N)\}}{S}. \tag{2.6}$$

2.4.3 Suavização Exponencial Simples

Neste tipo de previsão o valor $\widehat{x}(n+1)$ previsto consiste em um valor exponencialmente suavizado obtido a partir da equação

$$\widehat{x}(n+1) = \alpha x(n) + (1-\alpha)\widehat{x}(n), \tag{2.7}$$

onde α é definida como uma constante de suavização. A Equação (2.7) também pode ser escrita na forma expandida como

$$\widehat{x}(n+1) = \alpha x(n) + \alpha(1-\alpha)x(n-1) + \alpha(1-\alpha)^2 x(n-2) + ... \tag{2.8}$$

Na prática a suavização exponencial simples é uma média ponderada cujos pesos referentes às observações mais recentes são maiores. Isto é uma das vantagens deste método em relação ao das médias móveis, em que tanto as amostras antigas como as recentes possuem a mesma influência sobre o valor a ser previsto.

Em relação ao parâmetro α, quanto maior seu valor, menor peso é dados às observações passadas. Valores menores, por outro, lado implicam em maior peso nestas observações, resultando em previsões finais mais estáveis, em casos em que existam flutuações aleatórias no valor observado atual (presente). Devido a esta flexibilidade, a suavização exponencial é mais utilizada que as médias móveis. Além disso ao se utilizar

$$\alpha = \frac{2}{R-1}, \tag{2.9}$$

é possível se obter resultados semelhantes aos da média móvel (MORETTIN; TOLOI, 2006; MONTGOMERY; JOHNSON; GARDINER, 1990). Entretanto, uma de suas desvantagens reside na dificuldade em se ajustar o valor de α.

2.4.4 Modelos ARMA e ARIMA

Existem dois tipos de modelos amplamente utilizados em previsões lineares de séries temporais. O ARMA (*Autorregressive Moving Average*) e ARIMA (*Integrated Moving Average*). Ambos são combinações de modelos autorregressivos (*Autorregressive* - AR) e de médias móveis (*Moving Average* - MA), desenvolvidos por Box, Jenkins e Reinsel (1994) (MORETTIN; TOLOI, 2006; MENEZES JÚNIOR, 2012).

Um modelo autorregressivo pode ser descrito como

$$x(n+1) = \phi_0 + \sum_{i=1}^{p} \phi_i x(n-i+1), \tag{2.10}$$

em que ϕ_i são os coeficientes e p é a ordem de regressão. O método dos mínimos quadrados é a forma mais comum de se encontrar os coeficientes ϕ_i deste tipo de modelo.

O processo de médias móveis de ordem q é descrito por

$$x(n) = a(n) + \theta_1 a(n-1) + \theta_2 a(n-2) + ... + \theta_q a(n-q), \tag{2.11}$$

sendo θ_i os coeficientes do modelo, calculados pelo método de verossimilhança e os termos $a(n-i)$ são variáveis aleatórias com média zero e variância σ^2 contante, denominados ruído branco. Portanto, conceitualmente, um modelo de média móvel é uma regressão linear da observação atual das séries temporais contra os choques aleatórios de uma ou mais observações anteriores. Ajustar um modelo MA a uma série temporal é mais complicado do que ajustar um modelo AR porque, no primeiro, os termos de erro aleatório não são previsíveis (ADHIKARI; AGRAWAL, 2013).

Modelos AR e MA podem ser combinados para formar os modelos ARMA, que são matematicamente representados por

$$x(n) = \sum_{i=1}^{p} \phi_i x(n-1) + a(n) \sum_{i=1}^{q} \theta_i a(n-i), \tag{2.12}$$

onde os coeficientes ϕ_i e θ_i e as ordens p e q são os mesmos definidos previamente.

A principal limitação do modelo ARMA é sua aplicabilidade limitada apenas às séries temporais estacionárias, entretanto, em aplicações reais, muitas das séries, como as de

dados sócio-econômicos, financeiros ou aquelas suscetíveis a tendência ou sazonalidades, não apresentam tal característica. O ARIMA é uma generalização do modelo anterior, que, por sua vez, pode ser aplicado a séries com características não-estacionárias. Isto é feito pela aplicação de diferenças finitas entre suas amostras consecutivas (ADHIKARI; AGRAWAL, 2013).

Considerando-se uma série $\{x(n)\}_{n=1}^{N}$ não-estacionária, é possível torná-la estacionária por meio de uma quantidade d de sucessivas aplicações da equação

$$w(n) = \delta x(n) = x(n) - x(n-1), \tag{2.13}$$

e, a partir de então, é possível a aplicação de um modelo linear de Box-Jenkins. Destaca-se também, que os modelos AR, ARMA e ARIMA não são indicados para a captura de dependência temporal de longa duração, retendo apenas memórias de curta duração (MENEZES JÚNIOR, 2012).

3 PROCESSAMENTO DE SINAIS

3.1 Introdução

Neste capítulo é dada ênfase às duas ferramentas de processamentos de sinais utilizadas nesta dissertação: A Transformada *Wavelet* (TW) e o Identificador de Hampel.

A análise *Wavelet* fornece representação de tempo e frequência (análise de escala de tempo) simultaneamente. A TW é aplicada para decompor o sinal original do domínio do tempo em várias outras escalas com diferentes níveis de resolução no que é chamado de decomposição multiresolução (CHUI, 2016).

Isso permite extrair a informação irregular do sinal original que é mais provável que seja produzida pelos resíduos e tem alterações de alta frequência. Tecnicamente, os dados de carga são transformados em coeficientes baixos e altos. Os coeficientes baixos são uma versão aproximada associada à filtragem de baixa passagem, enquanto a última é uma versão detalhada associada à filtragem de alta frequência. A seleção do tipo de *wavelet* tem um efeito significativo nos resultados obtidos (BASHIR; EL-HAWARY, 2009).

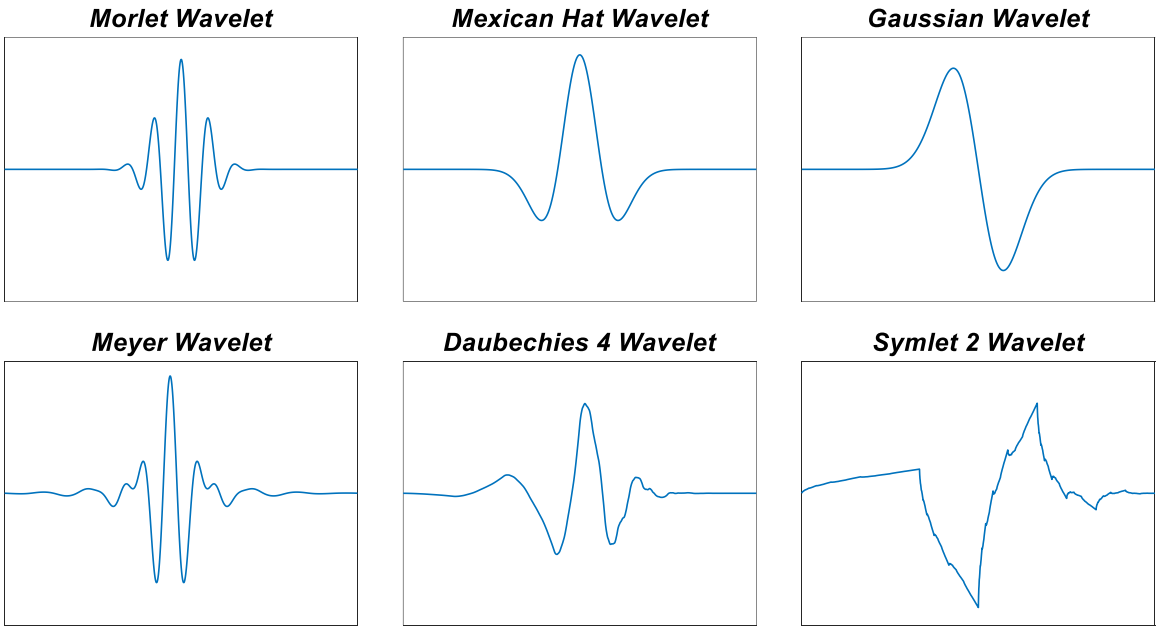
Neste trabalho, a TW é utilizada durante a etapa de pré-processamento dos dados. Neste capítulo são apresentados alguns conceitos básicos sobre TW com base nas referências Addison (2002) e Jensen e Cour-Harbo (2001). Posteriormente, é abordado de maneira concisa o Identificador de Hampel, que, por sua vez, utiliza a mediana como uma maneira de substituir eventuais *outliers* contidos em amostras de um determinado sinal.

3.2 *Wavelets*

A análise de TW usa pequenas funções de onda conhecidas como *wavelets*. Estas funções são usadas para transformar o sinal sob investigação em outra representação que apresenta a informação do sinal de uma forma mais útil. De maneira sucinta, isto é feito de tal forma que uma *wavelet* seja deslocada (convolução) e escalonada sobre um sinal em análise. Essa transformação do sinal é conhecida como Transformada *Wavelet*.

A transformada é calculada em várias localizações do sinal e para várias escalas da *wavelet*, preenchendo assim o plano de transformação: isto é feito de uma forma contínua suave para a transformada de onda contínua (TWC) ou em passos discretos para a transformada discreta de *wavelets* (TWD).

Figura 2 – Exemplos de *wavelets*.



Fonte – O autor

A escolha de uma *wavelet* para uma aplicação em particular depende tanto da natureza do sinal quanto do que exigimos da análise (ou seja, que tipo de fenômenos físicos ou processo se deseja investigar, ou como se deseja manipular o sinal). A Figura 2 ilustra alguns tipos de *wavelets* comumente utilizadas.

A seguir são discutidos alguns dos principais aspectos da TWC e da TWD, assim como a utilização da TWD a partir de bancos de filtros, que consiste na Análise Multirresolução (AMR), utilizada nesta dissertação.

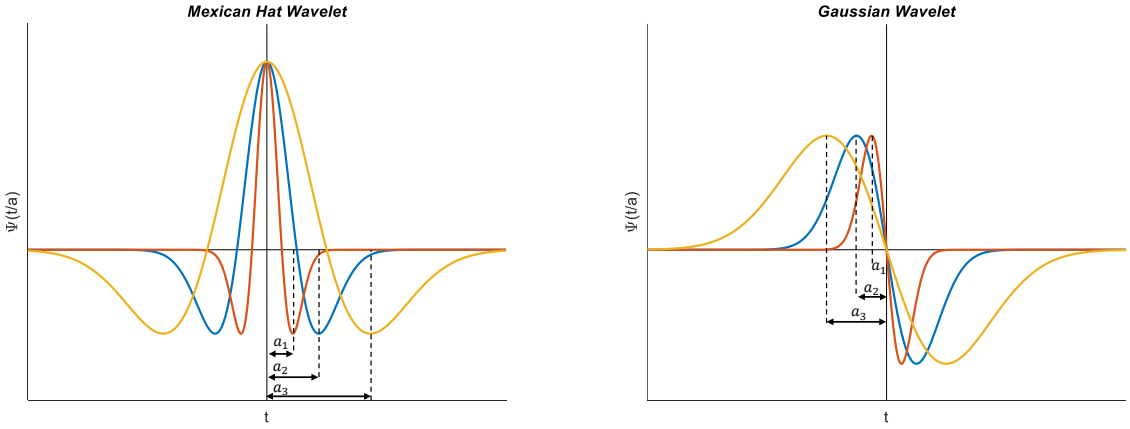
3.3 A Transformada *Wavelet* Contínua

A forma básica de uma *wavelet*, da qual são derivados os movimentos de translação e escalonamento, é conhecida como *wavelet* mãe. Na TWC, o movimento de dilatação é governado pelo parâmetro a, enquanto o movimento da *wavelet* em relação ao tempo é determinado pelo parâmetro b. Desta forma, a *wavelet* $\psi(t)$ pode ser escrita em termo destes dois parâmetros, como mostra a Equação (3.1),

$$\psi_{a,b}\left(\frac{t-b}{a}\right). \tag{3.1}$$

Portanto, em uma *wavelet* mãe, os parâmetros a e b recebem, respectivamente, valor igual a 1 e 0, mantendo-se o formato original da *wavelet*.

Figura 3 – Ilustração do movimento de escalonamento (dilatação e contração) das *wavelets* Gaussiana e Chapéu Mexicano.



Fonte – O autor.

Na Figura 3 é possível observar o procedimento de escalonamento aplicado às *wavelets* Chapéu Mexicano (*Mexican Hat Wavelet*) e Gaussiana (*Gaussian Wavelet*), onde são utilizadas as seguintes relações para o parâmetro a: $a_1 = a_2/2$ e $a_3 = 2 \cdot a_2$.

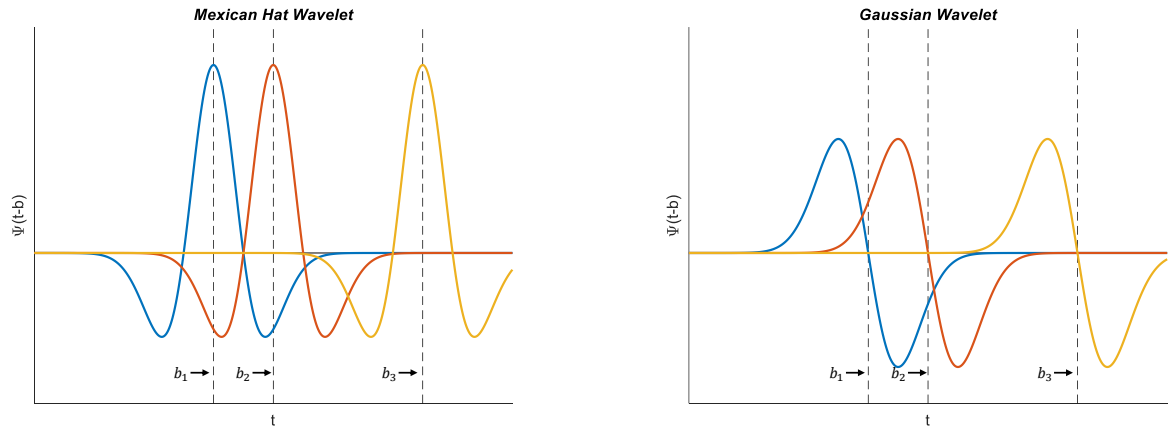
Por outro lado o processo de convolução destas mesmas *wavelets* pode ser visualizado na Figura 4, em que o parâmetro b obedece as seguintes relações: $b_1 = 2 + b_2$ e $b_3 = 7 + b_2$.

Levando-se em consideração a Equação (3.1), é possível obter a TWC de um sinal $x(t)$ utilizando-se uma faixa de valores para os parâmetros a e b, definida como

$$T(a,b) = w(a) \int_{-\infty}^{+\infty} x(t)\, \psi\left(\frac{t-b}{a}\right) dt, \tag{3.2}$$

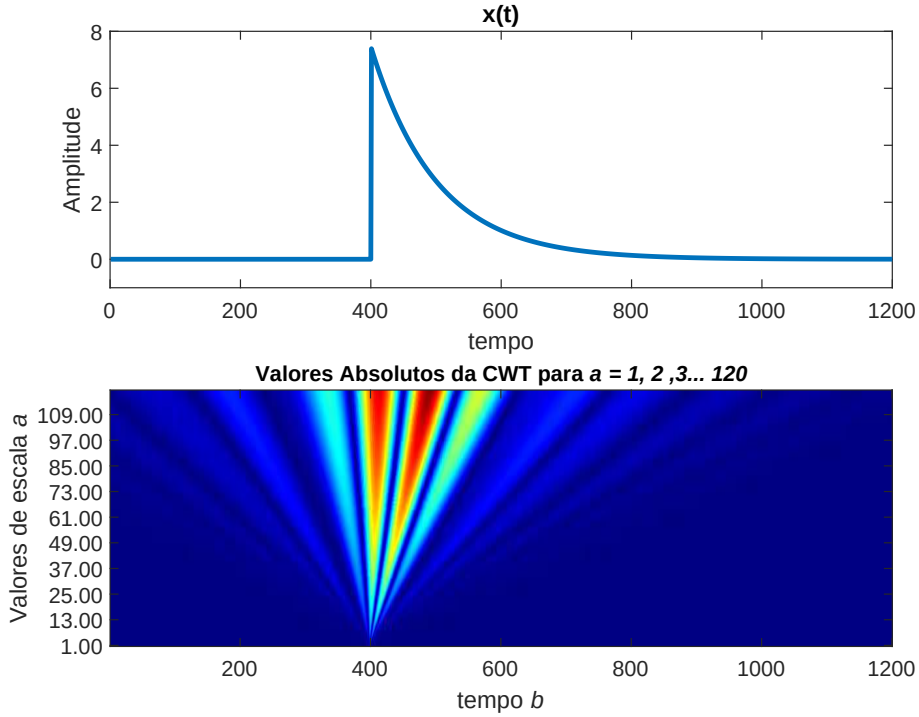
em que $w(a)$ é uma função cujo valor é definido geralmente como $1/\sqrt{a}$, assegurando que, independentemente da escala utilizada pelas *wavelets*, todas possuam a mesma energia Addison

Figura 4 – Ilustração do movimento de convolução (translação) das *wavelets* Gaussiana e Chapéu Mexicano.



Fonte – O autor.

Figura 5 – Exemplo de Aplicação da TWC em sinal $x(t)$ composto por uma descontinuidade. Abaixo, a TWC é calculada em função dos parâmetros a e b.



Fonte – O autor.

(2002). Na Equação(3.2), percebe-se que o produto entre a *wavelet* e o sinal são integrados ao longo da faixa de valores de $x(t)$. Matematicamente, este processo é conhecido como convolução.

Por meio da movimentação de uma *wavelet* ao longo de um sinal $x(t)$, com sucessivos incrementos no valor de b, é possível encontrar estruturas no sinal relacionadas a valores específicos de a. Este processo é, portanto, a TWC e é repetido até que todas estas estruturas do sinal possam ser distinguidas.

A Figura 5 ilustra a aplicação da TWC na detecção de um ponto de descontinuidade que consiste em uma subida abrupta do sinal, seguida por um decaimento exponencial. O gráfico consiste nos valores absolutos da TWC, calculados em função de a, b. O valores de b variam de acordo com o tamanho do sinal $x(t)$, enquanto a foi definido no intervalo $[1, 120]$. Os coeficientes da TWC de maior amplitude (denotados pelas cores mais quentes) são concentrados em uma região estreita no plano de escala e tempo centradas ao redor do ponto 400, que é onde ocorre a descontinuidade do sinal. Nota-se que quanto menor o valor de a, mais próximo do ponto específico de descontinuidade a TWC "aponta". Para gerar a Figura 5 foi utilizada a *wavelet* Meyer, resultados diferentes e até mais precisos podem ser obtidos com o uso de outras *wavelets*.

A partir dos valores obtidos pela Equação (3.2) é possível reconstruir o sinal original $x(t)$, por meio da Transformada *Wavelet* Inversa, descrita na Equação (3.3), utilizando-se todos

os valores de escala (a) e localização (b) previamente utilizados.

$$x(t) = \frac{1}{C_g} \int_{-\infty}^{+\infty} \int_{0}^{\infty} T(a,b) \psi_{a,b}(t) \frac{da\,db}{a^2}, \tag{3.3}$$

em que C_g é a condição de admissibilidade (ADDISON, 2002).

3.4 A Transformada *Wavelet* Discreta

Como discutido previamente, uma *wavelet* mãe é utilizada para análise de sinais por meio de dois procedimentos: o de escalonamento e o de convolução. Uma função *wavelet*, representada por $\psi(t)$, pode ser descrita em termos destes dois parâmetros, como mostra a Equação (3.4).

$$\psi_{m,n}(t) = \frac{1}{\sqrt{a_0^m}} \psi \left(\frac{t - nb_0 a_0^m}{a_0^m} \right). \tag{3.4}$$

O parâmetro m é responsável pelos processos de dilatação e contração da *wavelet* e o parâmetros n é responsável pelo movimento de convolução da *wavelet* ao longo do sinal em análise. As constantes a_0 e b_0 são, respectivamente, um passo fixo de dilatação, com valor maior que 1, e o parâmetro de localização, cujo valor é maior que zero. É visto em Addison (2002) que valores adequados para estes parâmetros são 2 e 1, respectivamente.

Levando-se isso em consideração, a TW de um sinal contínuo $x(t)$, utilizando-se uma *wavelet* discretizada, como descrito na Equação (3.4), é definida por

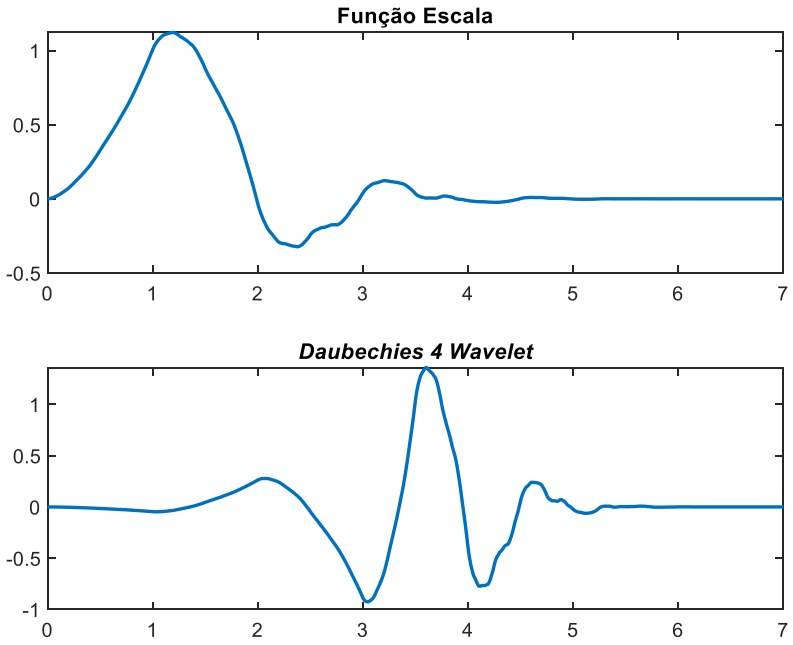
$$D_{m,n} = \int_{-\infty}^{+\infty} x(t)\, \psi_{m,n}(t)\ dt. \tag{3.5}$$

Os valores oriundos da *wavelet* caracterizam as altas frequências do sinal em análise e são conhecidos como coeficientes de Detalhe (D) da TW. Tais coeficientes muitas vezes representam a existência de ruídos ou descontinuidades em um sinal.

Por outro lado, as componentes de baixa frequência estão associadas a uma função escala que, por sua vez, está associada a uma *wavelet* discretizada. Assim, a função escala esta associada à suavização do sinal e é dada pela Equação (3.6). A Figura 6 ilustra, como exemplo, a função escala associada à *wavelet* Daubechies 4.

$$\phi_{m,n}(t) = 2^{\frac{-m}{2}} \phi(2^{-m}t - n) \tag{3.6}$$

Figura 6 – Função escala associada à *wavelet* Daubechies 4.



Fonte – O autor.

Portanto, a convolução da função escala com um sinal produz os coeficientes de Aproximação (A) da TWD, como descrito na Equação (3.7).

$$A_{m,n} = \int_{-\infty}^{+\infty} x(t)\,\phi_{m,n}(t)\;dt \tag{3.7}$$

Os coeficientes de aproximação em uma escala m são conhecidos como aproximação discretizada. A aproximação contínua do sinal na escala m pode ser gerada pela combinação das funções escala com os coeficientes de aproximação, de acordo com a Equação 3.8:

$$x_m(t) = \sum_{n=-\infty}^{+\infty} A_{m,n}\phi_{m,n}(t), \tag{3.8}$$

em que $x_m(t)$ é uma versão suavizada do sinal original $x(t)$, reconstruído a partir dos seus coeficientes de aproximação em uma determinada escala m. Assim, a aproximação (versão suavizada) do sinal na Equação (3.8) consiste numa sequência de funções escala colocadas lado a lado, cada uma fatorada por seu respectivo coeficiente de aproximação.

É possível reconstruir o sinal $x(t)$ a partir da combinação de sua aproximação $x_m(t)$ com sucessivos sinais de detalhe $d_m(t)$, em que $d_m(t)$ está definido na Equação (3.9). Desta forma, $x(t)$ pode ser reconstruído em uma escala arbitrária m_0, de acordo com a Equação (3.10). Percebe-se que, no caso da TWD, é possível reconstruir completamente o sinal original usando somatórios infinitos de coeficientes discretos das *wavelets* em vez de integrais contínuas, como

ocorre na TWC. O que resulta em um cálculo mais rápido da TW e sua inversa (ADDISON, 2002).

$$d_m(t) = \sum_{n=-\infty}^{+\infty} D_{m,n}\psi_{m,n}(t) \tag{3.9}$$

$$x(t) = x_{m_0}(t) + \sum_{m=-\infty}^{m_0} d_m(t) \tag{3.10}$$

Ainda, como demonstra Addison (2002), a aquisição de sucessivos níveis de aproximação e detalhes do sinal original representa a Análise Multirresolução (AMR). Além disso, a adição dos coeficientes de detalhe com os coeficientes de aproximação de um sinal, ambos em um mesmo nível m, resulta na aproximação do sinal um nível acima, ou seja:

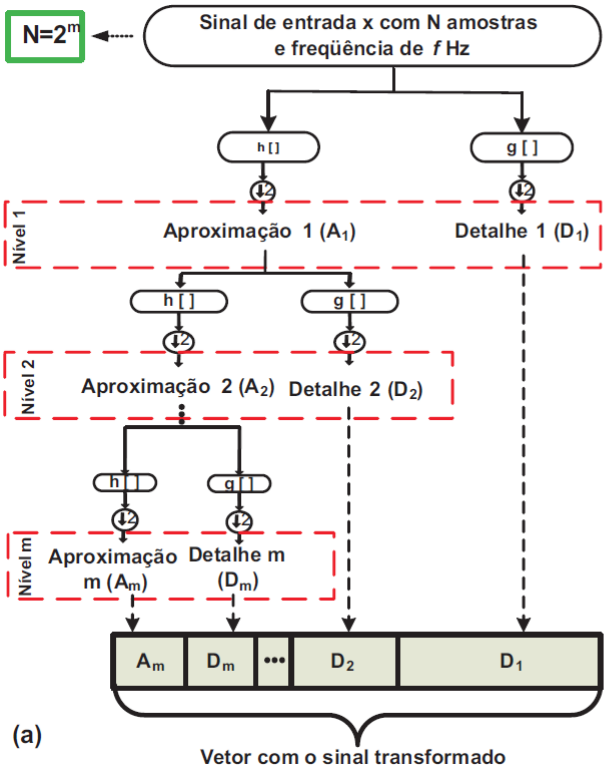
$$x_{m-1}(t) = x_m(t) + d_m(t) \tag{3.11}$$

A partir do exposto nesta seção e o que é descrito nas referências citadas, a aplicação das funções ψ e ϕ em um sinal $x(t)$ equivale à utilização de um filtro passa-altas e passa-baixas, respectivamente. Observa-se, também, que em situações práticas, o sinal $x(t)$ é uma versão discretizada no tempo. Portanto, um sinal composto por N amostras deve ter seu comprimento N determinado a partir de uma potência de 2, ou seja, $N = 2^M$. Desta forma, o número de níveis obtidos a partir de uma AMR deve estar compreendido no intervalo $0 < m < M$.

A obtenção de sucessivos níveis de decomposição de um sinal por meio de AMR, assim como faixa de frequência e amostragem do sinal em nível m são ilustrados na Figura 7, em que h e g representam, respectivamente, filtros passa-baixas e passa-altas e na saída de cada filtro é aplicado o operador *downsampling*, que reduz o número de amostras do sinal pela metade. Conforme (JENSEN; COUR-HARBO, 2001), a TWD consiste em uma filtragem digital no domínio do tempo, através de convolução discreta com aplicação do operador *downsampling*, o que torna possível sua implementação por meio de bancos de filtros. Assim, é possível fazer uma analogia entre o filtro passa-baixas e a função ϕ e entre o filtro passa-altas e a função ψ, visto que os dois primeiros são responsáveis por fornecer Aproximação e os dois últimos fornecem o Detalhe do sinal.

A Figura 8 ilustra um sinal Doppler ruidoso. Como exemplo de aplicação da TWD, é possível realizar uma filtragem no sinal de forma a retirar uma parte considerável do ruído existente. Desta forma, o sinal é decomposto em três níveis, conforme ilustrado na Figura 9. O

Figura 7 – a) Ilustração da Análise Multirresolução; b) Amostragem do sinal em cada nível *m* de decomposição.



(b)

Sinal	Faixas de Freqüência (Hz)	Nº de Amostras
x	0 ~ f	N
A_1	0 ~ f/2	N/2
D_1	f/2 ~ f	N/2
A_2	0 ~ f/4	N/4
D_2	f/4 ~ f/2	N/4
A_m	0 ~ $f/2^m$	1
D_m	$f/2^m$ ~ $f/2^{m-1}$	1

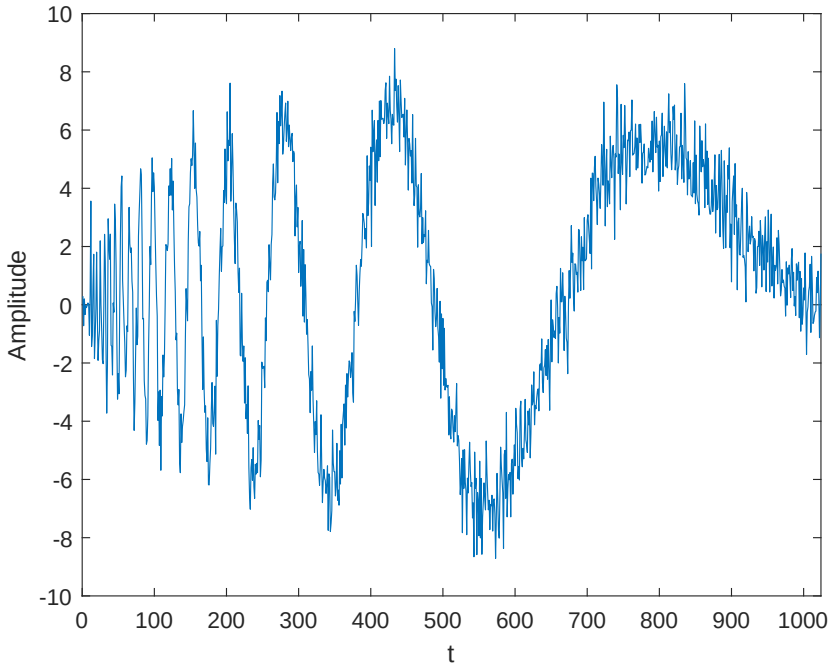
Fonte – Branco (2009).

processo é feito de acordo com o descrito no diagrama da Figura 7. Ou seja, a TWD é aplicada obtendo-se sucessivos níveis de Aproximação e Detalhe a partir dos coeficientes de Aproximação do nível anterior. Percebe-se, por meio da Figura 9, que tanto os sinais de Aproximação quanto os de Detalhe possuem metade da quantidade de amostras do sinal no nível imediatamente anterior.

As Figuras 9(a), 9(b) e 9(c) são referentes às componentes de baixa frequência, enquanto as Figuras 9(d), 9(e) e 9(f) são referentes às componentes de alta frequência. Estas últimas, por sua vez, indicam a presença de ruído no sinal. À medida em que mais níveis de decomposição são obtidos, o sinal de Aproximação se assemelha cada vez mais à uma versão suavizada do sinal original, pois a cada nível, mais sinais de Detalhe (ruído) são retirados. Isso acontece por que a TWD é aplicada apenas ao sinal de Aproximação do nível anterior, fazendo com que o sinal do nível subsequente seja cada vez mais suavizado, mas com uma amostragem menor.

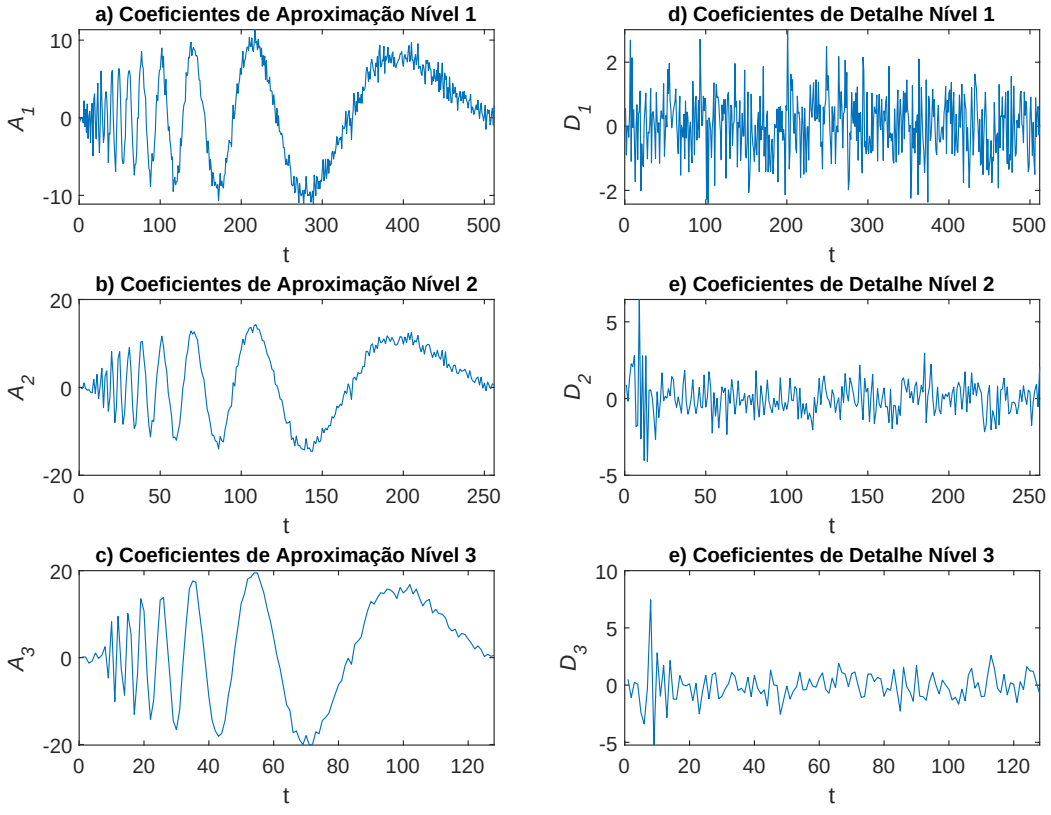
A filtragem do sinal é feita, substituindo os valores dos coeficientes de Detalhe (alta frequência) por zero em todos os três níveis exemplificados na Figura 9, ou seja, removendo-se as componentes de alta frequência durante a reconstrução do sinal por meio da Transformada

Figura 8 – Sinal com presença de ruído.



Fonte – Adaptada de MathWorks (2017).

Figura 9 – Aplicação da TWD no sinal ruidoso da Figura 8.

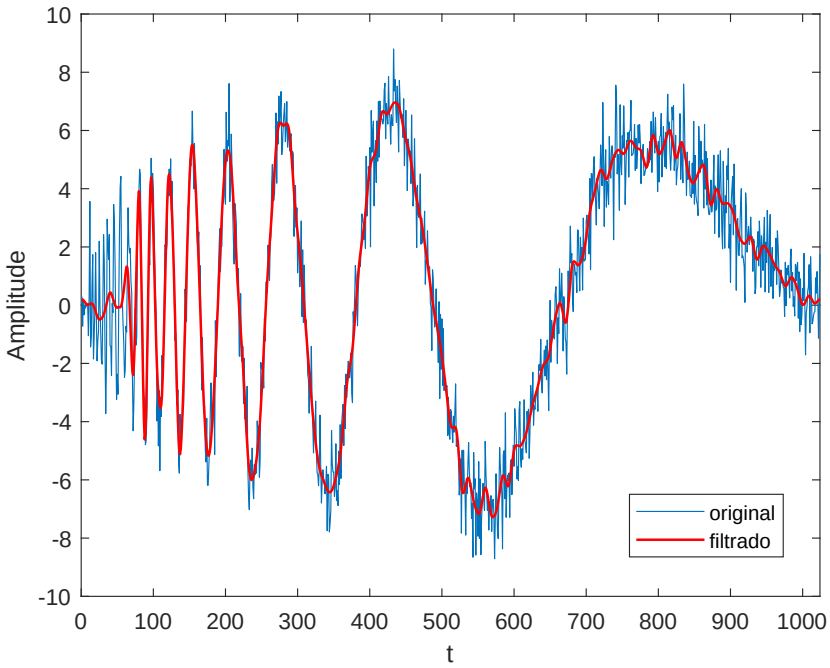


Fonte – O autor.

Wavelet Inversa. O resultado final é ilustrado na Figura 10. Percebe-se que uma parte considerável do ruído originalmente presente no sinal foi removido. O resultado final pode depender do tipo de *wavelet* (filtro) utilizada, assim como até qual nível o sinal foi decomposto. Níveis de decomposição muito elevados e substituição de todos os coeficientes de Detalhe por zero podem fazer com que haja muita perda de informação, resultando em uma reconstrução imprecisa do sinal.

O processo estabelecido na AMR também pode ser utilizado para detecção de anomalias em um sinal, como é utilizado neste trabalho e explicado mais adiante.

Figura 10 – Sinal filtrado por meio da aplicação da TWD.



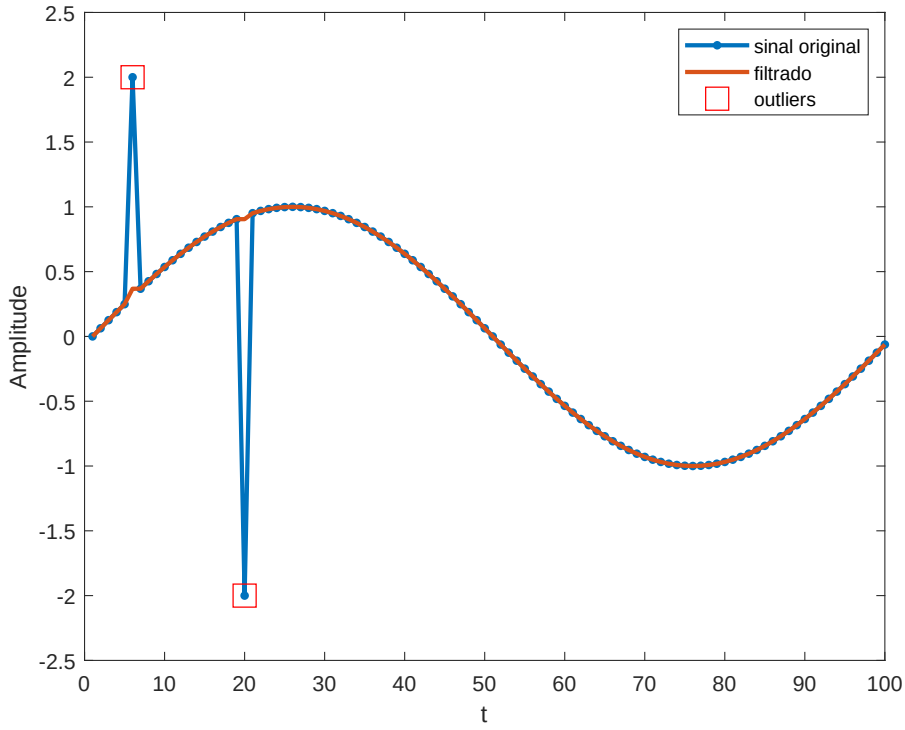
Fonte – O autor

3.5 Identificador de Hampel

Outliers são observações que não seguem a distribuição estatística dos dados. Em um conjunto de amostras, podem ser considerados como observações que desviam significativamente da maioria. Podem ser gerados por diferentes fatores, tais como presença de ruído em sensores, distúrbios, degradação de instrumentos, erros de natureza humana,etc. Dentre os diversos métodos de detecção de *outliers*, destaca-se o Identificador de Hampel devido a sua robustez e eficiência (LIU; SHAH; JIANG, 2004; PEARSON, 2002).

Considerando-se uma sequência de amostras $\mathbf{x} = \{x_1, x_2, x_3, ..., x_N\}$ e uma janela

Figura 11 – Exemplo de aplicação do Identificador de Hampel em senoide com dois *outliers*.



Fonte – Adaptada de MathWorks (2015).

deslizante que percorre as N amostras de $\mathbf{x}$, definida por

$$W_L = \{x_{n-L},\ ...,\ x_n,\ ...,\ x_{n+L}\}, \tag{3.12}$$

em que L é o número de vizinhos que estão ao redor da amostra x_n. E m_n é a mediana de W_L:

$$m_n = mediana(x_{n-L},\ ...,\ x_n,\ ...,\ x_{n+L}). \tag{3.13}$$

O identificador de Hampel irá declarar uma amostra $x(n)$ como *outlier*, para um determinado limiar t caso

$$|x_n - m_n| > t\,\delta_n, \tag{3.14}$$

onde δ_n é uma estimativa de escala do desvio absoluto mediano (*Median Absolute Deviation* - MAD):

$$\delta_n = \kappa \times mediana_{l\in[-L,L]}\{x_{n-l} - m_n\}, \tag{3.15}$$

em que $\kappa \approx 1,4826$. A relação δ_n/κ é conhecida como MAD. O fator κ é definido para que o valor esperado de δ_n seja igual ao desvio padrão para dados distribuídos normalmente (LIU; SHAH; JIANG, 2004; DAVIES; GATHER, 1993; PEARSON, 2002; PEARSON *et al.*, 2016).

A Figura 11 ilustra um exemplo de aplicação do Identificador de Hampel em um sinal em que duas de suas amostras possuem valores destoantes dos demais. A filtragem deste sinal permite a substituição destes valores pelo valor da mediana da janela deslizante W_L em cada uma das amostras consideradas *outliers*.

4 ALGORITMOS DE AGRUPAMENTO

4.1 Introdução

A análise de agrupamento, ou *clustering*, consiste em um conjunto de técnicas computacionais em que o objetivo é separar objetos de um determinado conjunto de dados em grupos homogêneos a partir de uma determinada métrica de dissimilaridade, com base nas características existentes em cada um desses objetos. Em outras palavras, objetos que possuem alta similaridade são designados a um mesmo grupo e, concomitantemente, objetos que pertencem a grupos distintos possuem maior dissimilaridade de acordo com algum critério pré-estabelecido. Assim, pode-se afirmar que o agrupamento se faz necessário quando existe a necessidade de se conhecer a estrutura e padrões presentes em um conjunto de dados sob análise e não há, a princípio, o conhecimento sobre qual grupo cada um dos objetos pertence, ou seja, as instâncias não possuem rótulos que permitam sua identificação (LINDEN, 2009; LIAO, 2005).

Os algoritmos de agrupamento mais utilizados são os particionais e os hierárquicos. No primeiro caso os grupos são gerados por meio da otimização de uma função objetivo e do aprimoramento, a cada iteração, da qualidade das partições estimadas inicialmente. Por outro lado, nos algoritmos hierárquicos utiliza-se uma estrutura de dados baseada em árvores binárias conhecida como dendograma. Esta estrutura é utilizada para realizar a escolha da quantidade de grupos por meio de sua divisão em diferentes níveis. Dependendo do nível escolhido uma quantidade diferente de grupos é gerada, sem que haja a necessidade de executar novamente o algoritmo. Outra diferença existente entre ambos, é a necessidade de se fornecer um conjunto inicial de grupos, no caso dos algoritmos particionais, enquanto que na outra abordagem não existe esta necessidade (REDDY; VINZAMURI, 2013; RAJARAMAN; ULLMAN, 2011).

O papel da métrica de distância também é diferente em ambos os casos. No agrupamento hierárquico, a métrica de distância é aplicada inicialmente em cada dado no nível base e, em seguida, aplicados progressivamente nos subgrupos, através da escolha de pontos representativos absolutos. No entanto, no caso dos algoritmos baseados em métodos particionais, em geral, os pontos representativos escolhidos em diferentes iterações podem ser pontos "virtuais", como o centróide do grupo. (REDDY; VINZAMURI, 2013).

Devido a esta habilidade em se identificar e organizar as estruturas contidas em conjunto de dados, os algoritmos de agrupamento são uma abordagem bastante eficiente na análise exploratória de dados. A informação processada pelo algoritmo pode ser considerada

estática, caso os dados não variem em relação ao tempo, ou dinâmica, como é o caso das séries temporais, em que os pontos que compõem cada objeto se dão em função do tempo.

Com a grande quantidade de dados cada vez mais presente devido à constante presença de sensores e novos conceitos como computação em nuvem e *big data*, assim como suas aplicações, muitas informações passaram a ser armazenadas em forma de séries temporais. Como por exemplo dados de venda, taxa de câmbio em finanças, dados biométricos, rastreamento de partículas atômicas, etc. Resultando em diversas pesquisas que visam aperfeiçoar as técnicas de agrupamento empregadas, assim como a extração de informações relevantes dos diversos tipos de sistemas analisados. Alguns exemplos podem ser vistos nas referências de Aghabozorgi, Shirkhorshidi e Wah (2015) e Liao (2005).

Nesta dissertação, algoritmos particionais são utilizados para realizar o agrupamento de curvas de carga, com finalidade de se extrair padrões e comportamentos dinâmicos das séries temporais que constituem o conjunto de dados sob análise. Nas próximas seções são apresentadas as técnicas de agrupamento utilizadas nesta dissertação.

4.2 O Algoritmo *k-means*

O *k-means* é um dos métodos de agrupamento convencionais mais utilizados. Embora muito simples, o *k-means* é rápido, eficiente e depende pouco da sequência de amostras e, pode ser facilmente implementado na resolução de muitos problemas práticos, especialmente para aqueles com grandes conjuntos de dados (JIAXI *et al.*, 2008).

A fase inicial do algoritmo consiste em escolher k pontos representativos como centroides iniciais. Cada uma das instâncias presentes no conjunto de dados é atribuída ao centroide mais próximo, a partir do cálculo de uma medida de proximidade (distância euclidiada é geralmente a mais popular). O conjunto formado pelas instâncias que foram atribuídas a um mesmo centroide constitui um grupo k. À medida que novas instâncias são atribuídas à estes grupos, à cada iteração do algoritmo, novos centroides são calculados e, novamente, calcula-se a proximidade entre estes e todas as instâncias. Este passos são repetidos iterativamente até que não haja mais mudança nos centroides ou que outro critério de convergência seja alcançado. O *k-means* visa minimizar a soma do quadrado das distâncias entre as instâncias e o centroide dos k grupos (REDDY; VINZAMURI, 2013).

Esse método possui como principal limitação depender da seleção das instâncias que irão agir como centroides iniciais e diferentes seleções levam a diferentes resultados de

agrupamento, o que pode gerar em alguns casos baixa confiabilidade. Uma outra questão é que a quantidade de grupos deve ser previamente definida e fornecida ao *k-means*. Entretanto, muitas vezes esta informação não se encontra facilmente disponível, sendo necessária a análise por meio de observações ou aplicação de índices de validação para se obter uma estimativa da quantidade de grupos a ser definida, considerando-se uma determinada faixa de valores de k.

Algumas adaptações do *k-means* tradicional, que visam minimizar o efeito da inicialização dos centroides e melhorar o qualidade dos agrupamentos, têm sido propostas ao longo dos anos, como é o caso dos métodos de Hartigan e Wong (1979), Milligan (1981), Bradley e Fayyad (1998) e do *k-means++* proposto em Arthur e Vassilvitskii (2007).

4.3 O Algoritmo *minCEntropy*

O minCEntropy é um algoritmo de agrupamento que, comparado a outras abordagens, se destaca por não colocar qualquer suposição sobre a distribuição de dados, pois é baseado em uma medida generalizada de Entropia (entropia estrutural α de Havrda-Charvat) combinado à estimativa de densidade de janela de Parzen, resultando em um método de agrupamento conceitualmente claro, simples e fácil de implementar. Além disso, diferentemente do *k-means*, a proximidade ponto-a-cluster para o *minCEntropy* é julgada pela semelhança total entre um ponto e todos os outros pontos de dados nos clusters (VINH; EPPS, 2010).

Assim como o *k-means*, o *minCEntropy* pertence à categoria de algoritmos particionais e também é baseado na minimização de um função objetivo. A diferença entre ambos reside principalmente na forma como a função objetivo opera. Enquanto no primeiro, a soma do quadrado das distâncias é minimizada, ou seja, a distância entre as instâncias e os respectivos centroides, no *minCEntropy* o objetivo consiste em minimizar o critério de entropia condicional. Considera-se um conjunto de dados $X = \{x_1, x_2, ..., x_N\}$ composto por N instâncias e $C = \{c_1, c_2, ..., c_K\}$ uma forma de dividir X em K subconjuntos que não se sobrepõem. O espaço que contém todas os possíveis subconjuntos (partições) em que X pode ser dividido é dado por Γ. O problema consiste em identificar o agrupamento C^* em Γ que maximize a informação mútua entre o conjunto de dados X e o agrupamento (VINH; EPPS, 2010). Ou seja, encontrar o agrupamento C contido em Γ que possua a maior informação mútua (I) em relação a X, conforme a equação

$$C^* = \underset{C \in \Gamma}{arg\,max} \{I(C;X)\}, \tag{4.1}$$

ou, de forma equivalente minimize a entropia condicional (H):

$$C^* = \underset{C \in \Gamma}{arg\,min}\{H(X;C)\}. \tag{4.2}$$

Desta forma, a informação mútua pode ser obtida através da entropia de $H(X)$: $I(C;X) = H(X) - H(X;C)$, sendo $H(X)$ constante. A função objetivo é obtida aplicando-se a entropia de Havrsa-Charvat (HAVRDA; CHARVÁT, 1967) na Equação (4.2) e combinando-se o resultado com o método da janela de Parzen, obtendo-se

$$CE(C) = \sum_{k=1}^{K} \frac{\sum_{x_i, xj \in c_k} exp\left(\frac{-\left\|x_i - x_j\right\|^2}{4\sigma^2}\right)}{n_k}, \tag{4.3}$$

pela maximização de CE na Equação (4.3) a entropia condicional C^* é, então, minimizada,

$$C^* = \underset{C \in \Gamma}{arg\,max}\{CE(C)\}, \tag{4.4}$$

em que n_k é a quantidade de instâncias que fazem parte do grupo c_k e σ é o tamanho do *kernel* Gaussiano. O parâmetro CE é uma medida da qualidade do grupo. Portanto, enquanto no *k-means* a soma do quadrado das distâncias é minimizada, no critério de entropia condicional mínima, o somatório da similaridade intra-cluster em termos do *kernel* Gaussiano é maximizado, ou seja, as distâncias entre pares dos membros do mesmo grupo. Detalhes sobre o desenvolvimento do algoritmo e de cada uma destas equações podem ser encontrados em Vinh e Epps (2010).

4.4 Combinação de Agrupamentos

Ambos algoritmos discutidos anteriormente possuem paradigma de aprendizagem não-supervisionada, úteis quando não existe conhecimento prévio sobre classes associadas a cada um dos dados, além de desempenharem um papel fundamental na análise inicial dos dados. No decorrer dos anos, diversas técnicas de agrupamento têm sido propostas (JAIN; MURTY; FLYNN, 1999; HRUSCHKA *et al.*, 2009; NAGPAL; JATAIN; GAUR, 2013). Entretanto, considerando-se o problema de agrupamento de séries temporais, mais especificamente o de curvas de carga, não existe nenhuma abordagem que seja universalmente aceita, visto que cada algoritmo demonstra performance diferente de acordo com as medidas de adequação e indicadores de agrupamento.

No trabalho de Panapakidis, Alexiadis e Papagiannis (2015) diversos algoritmos como *k-means*, *minCEntropy*, SOM, *Fuzzy c-means* e algoritmos de agrupamento hierárquico são utilizados para a tarefa de agrupar séries temporais que consistem em curvas diárias de carga

de consumidores industriais. No estudo, diversas métricas de validação são utilizadas e, na maioria delas, o *minCEntropy* e *k-means* obtiveram bons resultados. Entretanto, como visto nas duas seções anteriores, cada algoritmo possui suas próprias características, o que pode ocasionar uma grande variedade de soluções a depender do ajuste de certos parâmetros.

Na Seção 4.2 observou-se que o *k-means*, embora seja um algoritmo rápido de boa eficácia, é sensível à inicialização dos centroides, o que pode ocasionar soluções diferentes à medida que novas execuções do algoritmo são feitas em um mesmo conjunto de dados. Apesar do algoritmo *minCEntropy*, não utilizar a mesma abordagem baseada em distâncias das instâncias aos centroides, o mesmo também necessita de um conjunto de grupos iniciais para que só então os critérios de similaridade entre instâncias e grupos possam ser calculados. Uma alternativa é utilizar uma iteração do algoritmo *k-means* e então utilizá-la como solução inicial do *minCEntropy*, o que também pode ocasionar soluções diferentes, a dependerem da posição inicial dos centroides, levando à convergência para mínimos locais.

Neste contexto, a maioria dos métodos focam em encontrar um agrupamento ótimo ou sub-ótimo com base em alguns critérios específicos, o que pode se tornar uma tarefa tediosa e que pode ocasionar uma grande quantidade de testes afim de verificar qual técnica oferece os melhores resultados, visto que cada uma oferece uma série de vantagens e desvantagens.

A combinação de agrupamentos, também conhecida como *Ensemble Clustering* ou *Consensus Clustering* tem sido utilizada como solução para este problema, fornecendo resultados que muitas vezes superam as soluções obtidas por um único método de agrupamento, como pode ser visto em Dimitriadou, Weingessel e Hornik (2001) e Ghosh e Acharya (2013). Trata-se de uma abordagem em que se utiliza um conjunto de agrupamentos, conhecidos como partições base, e a partir destas partições obtém-se uma partição consenso, cujo resultado costuma ser melhor quando comparada com as partições base individualmente. Além disso, diferentemente do que ocorre em tarefas de classificação ou regressão, em que o intuito da combinação de resultados é primariamente o melhoramento da precisão dos resultados obtidos, outras razões, além da melhora na qualidade das soluções, podem instigar o uso de combinações de agrupamentos (GHOSH; ACHARYA, 2013), tais como:

- **robustez:** Uma combinação de agrupamentos pode fornecer um modelo mais robusto, ao ser capaz de oferecer bons resultados mesmo considerando-se uma ampla variedade de conjuntos de dados. Estando, portanto, menos suscetível a ruídos, *outliers* e quantidade de atributos;

- **seleção de modelo:** Podem fornecer uma nova abordagem para o problema de seleção de modelos considerando a correspondência entre as soluções base para determinar o número final de agrupamentos a serem obtidos;
- **reaproveitamento de conhecimento sobre os dados:** Em certas aplicações, o conhecimento prévio acerca da variedade de agrupamento do conjunto de dados em estudo pode já existir devido a projetos e experimentações anteriores. Uma solução de consenso pode integrar essas informações para obter um agrupamento mais consolidado.

Na próxima seção são discutidos três dos principais métodos de combinação existentes na literatura, assim como mais detalhes do método utilizado nesta dissertação.

4.4.1 Função Consenso

Uma função consenso deve ser capaz de lidar corretamente com três problemas: a combinação de diferentes soluções de agrupamento, resolver o problema de correspondência entre grupos de partições base diferentes e garantir o consenso entre estas partições de forma simétrica e imparcial. Em meio a diversas abordagens, Yang (2016) elenca três métodos que levam em conta a simplicidade de implementação, complexidade computacional e qualidade do agrupamento. Tais métodos são fundamentados no estudo de Ghaemi *et al.* (2009), em que os autores abordam as funções consenso utilizadas em diversos outros trabalhos, citando vantagens e desvantagens. Em geral, todas as técnicas utilizadas podem ser divididas em cinco classes. De acordo com Yang (2016), as que melhor satisfazem as três características citadas acima são: função consenso baseada em particionamento de hiper-grafos, função baseada em co-associação e função consenso com base em votação.

No primeiro caso, os grupos são representados por hiper-arestas (*hyper-edges*) em um grafo onde cada vértice corresponde às instâncias que se deseja agrupar. Cada hiper-aresta representa um conjunto de instâncias que pertencem a um mesmo grupo, desta forma a bordagem consiste em encontrar um ponto de corte mínimo do hiper-grafo. Em relação ao segundo caso, algoritmos hierárquicos são aplicados a uma matriz de co-associação, que é gerada pela medida de similaridade entre múltiplas partições de entrada, e assim obtêm-se a partição consenso. Enquanto no último caso, as instâncias são designadas a um determinado grupo por meio da maior quantidade de vezes em que são assinaladas como pertencentes a um grupo, com base em um conjunto de partições geradas previamente (GHAEMI *et al.*, 2009).

Devido à facilidade de implementação e comprovada eficácia em outros trabalhos

como o de Dimitriadou, Weingessel e Hornik (2001) e nos trabalhos elencados por Ghaemi *et al.* (2009), optou-se por esta última abordagem no desenvolvimento desta dissertação.

Entretanto, um dos principais problemas encontrados no sistema de votação é o problema de correspondência entre os grupos que compõem cada partição base. Em virtude dos algoritmo de agrupamento possuírem aprendizagem não-supervisionada, não existe um índice (valor que indica a qual grupo cada instância pertence) fixo que possa identificar cada uma das instâncias presentes no conjunto de dados em relação ao grupo em que está designada. Ou seja, independentemente de serem utilizados um ou múltiplos algoritmos de agrupamento em um mesmo conjunto de dados, as instâncias em cada uma das vezes em que os algoritmos são executados podem possuir índices diferentes. Desta forma, como se aplicar um sistema de votação em um conjunto composto por T partições se não se sabe exatamente, a que grupo corresponde os índices utilizados em cada uma delas? A ideia básica seria permutar os índices dos k grupos até que a melhor concordância seja obtida. Entretanto, em problemas cuja quantidade de grupos é elevada, este pode se tornar um problema de complexidade $k!$ (GHAEMI *et al.*, 2009), (DIMITRIADOU; WEINGESSEL; HORNIK, 2001).

É importante esclarecer que uma partição base corresponde a uma solução qualquer gerada por um ou mais algoritmos de agrupamento. Ou seja, uma solução gerada, contendo uma quantidade específica de grupos. Para realizar o processo de votação, duas ou mais partições base, são geradas. Em seguida, é verificado para qual grupo cada instância é designada, baseando-se na maior quantidade de vezes em que cada uma é indicada (indexada) como pertencente a um grupo específico, de acordo com as partições base previamente geradas. Por exemplo: em um conjunto de cinco partições base, cada uma composta por dez grupos (enumerados de 1 a 10), uma determinada instância foi designada em quatro das cinco partições como pertencente ao grupo número 7 e uma vez como pertencente ao grupo número 2. Sendo assim, na partição consenso esta instância será pertencente ao grupo número 7 (maioria), sendo este o resultado obtido a partir da votação.

Em Yang (2016) é descrito um método simples de votação que utiliza o algoritmo de Munkres (WINSTON; GOLDBERG, 2004), (YANG, 2016) para contornar o problema de reatribuição dos índices de maneira mais rápida e eficaz. Primeiramente, todas as partições obtidas por meio das execuções de um ou mais algoritmos de agrupamento devem possuir a mesma quantidade de grupos e uma destas T partições deve ser escolhida como partição de referência. Assim, todo o procedimento envolve basicamente duas etapas: (i) utilizar o

algoritmo de Munkres para realizar a correspondência entre os índices dos grupos das partições base, levando-se em consideração uma partição de referência. (ii) Aplicar, em seguida, o voto majoritário nas partições de entrada e assim gerar a partição consenso. O pseudocódigo deste algoritmo é descrito a seguir.

Algoritmo 1: Pseudocódigo do sistema de votação

Entrada:

- *Um inteiro K (número de grupos das partições base)*
- *Um inteiro N (número de instâncias do conjunto de dados)*
- *Um conjunto de partições de entrada* $\{P_1, P_2, ..., P_T\}$, com T partições
- *Uma partição de referência* $P^{'} = P_1$
- *O algoritmo de Munkres (Hungarian Algorithm)*

for $t = 1 : T$ **do**
  Realizar correspondência dos índices da partição de entrada: $P_t^{'} = HUNGARIAN(P^{'}, P_t)$
end for
$P^{'} = \left\{ P_t^{'} \right\}_t^T$
for $t = 1 : T$ **do**
  for $n = 1 : N$ **do**
    for $k = 1 : K$ **do**
      $H_t^{n,k} = 1$, *se a instância n está atribuída ao grupo j em* $P_t^{'}$,
      $H_t^{n,k} = 0$, *caso contrário*
    end for
  end for
end for
for $n = 1 : N$ **do**
  $P_{consenso}(x_n) = arg\,max \sum_t^T w_t H_t^{n,k}$, *onde* $w_t = \frac{1}{T}$, $\forall t$
end for
Saída: *o agrupamento final* $P_{consenso}$.

A aplicação direta do algoritmo de *Munkres* é detalhadamente descrita em Yang (2016) por meio de exemplos. A partição $P_{consenso}$ representa o resultado final do agrupamento obtido através de voto majoritário com base nas partições de entrada fornecidas ao algoritmo.

Os algoritmos descritos neste capítulo são utilizados para realizar o agrupamento de séries temporais, especificamente curvas de demanda. Os algoritmos *k-means* e *minCEntropy* são utilizados para gerar um conjunto de possíveis soluções, denominadas partições base. Estas partições são submetidas ao algoritmo de votação para que seja obtida a versão final do agrupamento das séries temporais.

Os grupos obtidos a partir deste procedimento são utilizados como dados de treinamento pelas RNAs descritas no capítulo seguinte. Detalhes sobre como é realizado o agrupamento e a previsão das RNAs são dados no Capítulo 6.

5 REDES NEURAIS ARTIFICIAS

5.1 Introdução

Uma das principais características das redes neurais artificiais é a habilidade de aprendizado por meio de um processo de treinamento que, basicamente, consiste em fornecer à uma rede neural um conjunto de amostras que reflitam o comportamento de um sistema. São compostas por um conjunto de neurônios artificiais, conectados uns aos outros por meio de sinapses artificiais representadas por vetores ou matrizes de pesos sinápticos. As redes neurais são projetadas de forma a modelar o comportamento do cérebro humano na realização de uma tarefa específica ou função de interesse. Durante a fase de treinamento, o algoritmo de aprendizagem realiza a modificação e ajuste dos pesos sinápticos de uma RNA iterativamente, até que seja alcançado um objetivo de projeto desejado para o qual a rede foi designada (HAYKIN, 2009).

Basicamente, dois paradigmas de aprendizagem são comumente utilizados para o treinamento de redes neurais: aprendizado supervisionado e o não-supervisionado. O primeiro modelo requer que o conhecimento acerca do sistema trabalhado seja repassado para a rede neural, isso é feito disponibilizando-se um conjunto de pares de entrada-saída, ou seja, para cada amostra dos sinais de entrada é atribuída uma respectiva saída desejada. A resposta desejada, por sua vez, representa a atuação ótima da rede frente a um determinado sinal de entrada (vetor de treinamento). Por outro lado, no aprendizado não-supervisionado não é fornecida nenhuma informação sobre a resposta da rede frente a um determinado conjunto de entradas. Em outras palavras, não existem saídas desejadas e a rede deve se organizar de forma independente, detectando e ajustando seus pesos sinápticos às regularidades contidas nos conjuntos de amostras, realizando representações internas que codificam o comportamento do sistema de uma maneira mais explicita e simples (BECKER, 1991).

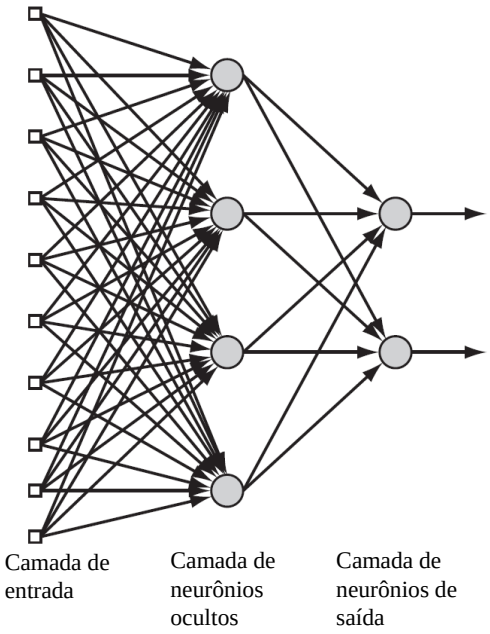
Neste capítulo são abordados alguns dos principais aspectos sobre redes neurais artificiais, tais como tipos de arquitetura, redes Perceptron Multicamadas e o algoritmo de aprendizagem *backpropagation*. Em seguida é apresentada uma breve discussão sobre as RNAs dinâmicas utilizadas nesta dissertação.

5.2 Redes Neurais *Feedforward*

Basicamente, as RNAs são compostas por um conjunto de neurônios artificiais, conectados uns aos outros por meio de sinapses artificiais, representadas por vetores ou matrizes de pesos sinápticos. A maneira como os diversos neurônios que compõem a rede estão arranjados define a arquitetura de uma RNA. De maneira geral, uma rede neural é formada por uma ou mais camadas de neurônios.

Em arquiteturas do tipo *feedforwar*d a informação segue um fluxo único ao longo destas camadas em direção à saída da rede. Na prática, uma rede *feedforward* deve possuir pelo menos uma camada de neurônios, as redes de múltiplas camadas se distinguem pela presença de uma ou mais camadas de neurônios ocultas, estas, por sua vez, ficam localizadas entre a camada de entrada e a camada de saída da rede e recebem esse nome devido a informação não se demonstrar de forma explícita como ocorre nas camadas de entrada e saída da rede. É importante observar que a quantidade de camadas ocultas, assim como a de neurônios que as constituem dependem de alguns aspectos tais como a complexidade do problema a ser mapeado, quantidade e qualidade dos dados disponíveis (SILVA; SPATTI; FLAUZINO, 2010). A Figura 12 ilustra uma rede *feedforward* com uma camada oculta. Percebe-se que cada camada é interconectada, de forma que a saída da primeira camada da RNA serve como entrada para a camada oculta, e as saídas da camada oculta alimentam a camada de saída da rede neural, as setas em negrito

Figura 12 – Rede Neural com Arquitetura *Feedforward* com 10 entradas, uma camada oculta com 4 neurônios e uma camada de saída com 2 neurônios.



Fonte – Adaptada de Haykin (2009).

ilustram o sentido que a informação é processada neste tipo de arquitetura, cada um dos nós (entradas e neurônios) que compõem a rede neural é conectado a cada um dos outros nós da camada subsequente, nota-se que nenhuma informação é processada na camada de entrada, devido a inexistência de neurônios nesta região.

Uma das principais arquiteturas de redes com arquitetura *feedforward* é a do tipo Perceptron Multicamadas (*MultiLayer Perceptron* – MLP), discutida na próxima sessão.

5.2.1 Redes Perceptron Multicamadas

As redes do tipo MLP consistem em RNAs com uma ou mais camadas ocultas, além das camadas de entrada e saída. Por se tratar de uma arquitetura do tipo *feedforward*, os sinais inseridos na entrada se propagam camada por camada, até a saída, representando uma função de sua entrada atual, parametrizada pelos seus pesos sinápticos (NORVIG; RUSSELL, 2014).

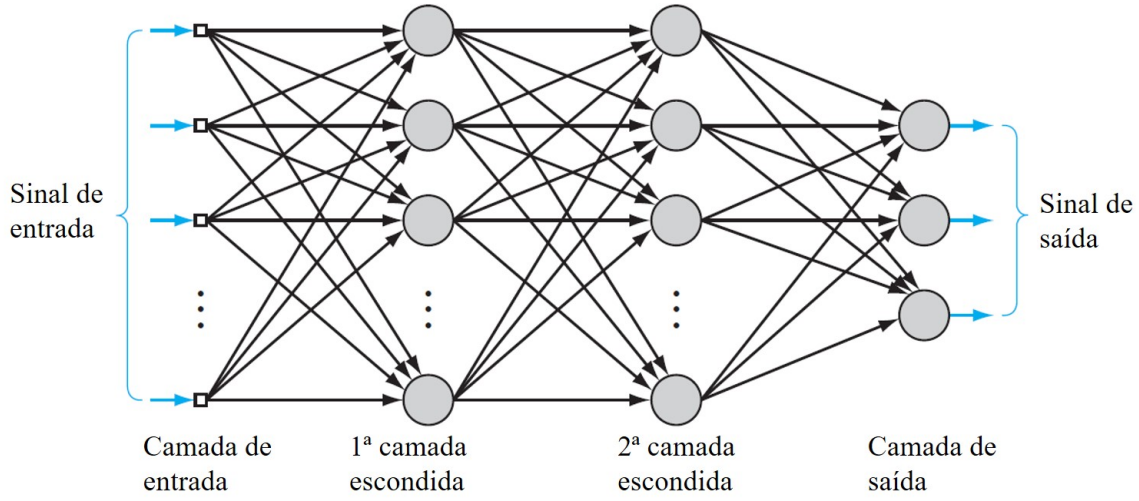
A existência de camadas ocultas não-lineares fornece à MLP a capacidade de resolver problemas complexos, pois tais camadas têm a função de permitir alterações subsequentes na representação dos dados originais até que o problema possa ser resolvido pela última camada de neurônios (camada de saída) (MENEZES JÚNIOR, 2006). Devido a isso, as redes MLP são reconhecidas por demonstrarem uma grande versatilidade quanto a sua aplicabilidade em diversos tipos de problemas em diferentes áreas do conhecimento. Dentre as áreas de potencial utilização é possível citar: aproximação universal de funções, reconhecimento de padrões, identificação e controle de processos, previsão de séries temporais e otimização de sistemas (SILVA; SPATTI; FLAUZINO, 2010).

O termo ‘rede neural’ tem origem na tentativa de encontrar uma representação matemática de como um neurônio biológico processa a informação. No decorrer dos anos diversos pesquisadores fizeram suas contribuições para o desenvolvimento da área. MacCuloch e Pitts (1990) foram os primeiros a apresentarem o conceito de redes neurais como modelos computacionais. Posteriormente, Hebb (1949) postulou a primeira estratégia de treinamento supervisionado. E, em 1958, Rosenblatt (1958) propôs o Perceptron como o primeiro e mais simples modelo de RNA, que consistia em um único neurônio artificial com sinapses ajustáveis por meio de treinamento supervisionado. A restrição deste modelo somente a aplicações de classificação de padrões com duas classes linearmente separáveis, consistia em uma grande limitação, visto que a maioria dos problemas tratados no mundo real não possui este tipo de comportamento (BISHOP, 2006; SILVA; SPATTI; FLAUZINO, 2010).

Embora as redes do tipo MLP representem uma generalização do modelo proposto por Rosenblat, as mesmas conseguem superar as limitações práticas impostas pelo modelo anterior. Nas MLPs cada neurônio possui uma função de ativação não-linear suave e, portanto, diferenciável em qualquer ponto. A não linearidade das funções de ativação utilizadas neste tipo de RNA é um fator de grande importância, visto que, do contrário, a relação existente entre as entradas e saídas poderia obter comportamento semelhante ao observado no Perceptron de Rosenblat. Ao contrário do que ocorre nas camadas intermediárias (ocultas), a camada de saída pode possuir funções de ativação lineares.

Outra característica da rede MLP é alto grau de conectividade existente entre seus neurônios. Ou seja, cada neurônio de uma camada está conectado a cada um dos neurônios da camada anterior, com cada interconexão associada a um determinado peso sináptico. Desta forma, também em contraste com o modelo de Rosenblat, o conhecimento relacionado ao comportamento entrada/saída do sistema é distribuído por todos os neurônios da MLP. A Figura 13 ilustra uma rede MLP composta pela camada de entrada, duas camadas ocultas e a camada de saída.

Figura 13 – Rede Perceptron Multicamadas com duas camadas ocultas.



Fonte – Adaptada de Haykin (2009).

Após definidos os aspectos de topologia da rede, os ajustes de pesos sinápticos e limiares de cada neurônio são realizados por meio de um processo de treinamento supervisionado denominado *backpropagation* ou algoritmo de retropropagação do erro, cujo funcionamento é discutido a seguir.

5.2.2 Algoritmo Backpropagation

Resumidamente, o algoritmo *backpropagation* consiste na retropropagação do erro através das camadas que compõem a RNA. Basicamente, quando a informação é propagada pela rede, um conjunto de pesos e limiares é definido, quando caminha em sentido reverso, baseado na diferença entre a saída desejada e a saída fornecida pela rede, os pesos são ajustados, levando-se em conta o erro obtido na saída da rede. Este processo contínuo permite, a cada iteração, que os pesos e limiares dos neurônios adquiram valores que possibilitam que a resposta fornecida pela rede seja cada vez mais próxima da resposta desejada, isso é equivalente ao decaimento do erro a cada iteração (HAYKIN, 1999; HAYKIN, 2009).

Percebe-se que a aplicação do algoritmo consiste em duas diferentes etapas. A primeira delas é referente à propagação adiante, nesta etapa os pesos e limiares permanecem inalterados e a saída de cada um dos j neurônios da rede neural é computada da seguinte forma:

$$y_j(n) = \phi\left(v_j(n)\right), \tag{5.1}$$

em que $v_j(n)$ é o campo local induzido do neurônio j, definido por

$$v_j(n) = \sum_{i=1}^{m} w_{ji}(n) y_i(n), \tag{5.2}$$

sendo m o número total de entradas (excluindo o bias) aplicadas ao neurônio j, $w_{ji}(n)$ é o peso sináptico que conecta o neurônio i de uma camada anterior ao neurônio j e $y_i(n)$ é o sinal de entrada do neurônio j (ou a saída de um neurônio i). Entende-se que quando se trata de uma camada imediatamente posterior à camada de entrada, o termo $y_i(n)$ da equação acima representa apenas o i-ésimo sinal de entrada da rede, podendo, neste caso, $y_i(n) = x_i(n)$. Após as entradas serem apresentadas e cada um dos neurônios apresentar sua respectiva saída conforme a Equação (5.1), o erro é calculado na camada de saída da rede, para cada neurônio j:

$$e_j(n) = d_j(n) - y_j(n), \tag{5.3}$$

em que $d_j(n)$ é a resposta que se deseja que o j-ésimo neurônio obtenha.

Na segunda etapa, retropropagação, o erro previamente calculado na saída é propagado no sentido reverso ao longo das camadas que precedem a camada de saída, camada por camada, de forma que o gradiente local de cada neurônio possa ser calculado. Para um neurônio localizado na camada de saída, o gradiente δ é calculado simplesmente pela multiplicação do erro pela derivada de sua função de ativação φ':

$$\delta_j(n) = e_j(n)\varphi'_j(v_j(n)). \tag{5.4}$$

Os neurônios das camadas escondidas são calculados por meio da equação

$$\delta_j(n) = \varphi'_j(v_j(n)) \sum_k \delta_k(n) w_{kj}(n), \tag{5.5}$$

sendo que k se refere ao k-ésimo neurônio, da camada subsequente, que está conectado ao neurônio j.

A atualização dos pesos sinápticos para os neurônios da camada de saída é dada por

$$w_{ji}(n+1) = w_{ji} + \eta \delta_j(n) y_i(n), \tag{5.6}$$

em que η é a taxa de aprendizagem.

Cabe enfatizar que, na Equação (5.6), o valor do gradiente local δ_j depende de qual camada está localizado o neurônio j, caso seja na camada de saída, δ_j é calculado de acordo com a Equação (5.4), caso seja um neurônio pertencente a uma camada oculta, δ_j é calculado seguindo a fórmula apresentada na Equação (5.5).

A aprendizagem não-supervisionada realizada em redes do tipo MLP pode ser interpretada como um problema de otimização numérica, em que a superfície do erro da MLP é uma função altamente não linear do vetor de pesos sinápticos $\mathbf{w}$. O método do gradiente descendente, exemplificado no algoritmo *backpropagation*, opera de acordo com uma aproximação linear da função custo na vizinhança local do ponto de operação $\mathbf{w}(n)$, baseando-se no vetor gradiente como a única fonte de informação local sobre a superfície do erro. Como resultado, o método possui relativa facilidade de implementação, mas em contrapartida oferece uma lenta taxa de convergência, podendo vir a ser um aspecto exaustivo, sobretudo em problemas de larga escala (HAYKIN, 2009).

Sendo assim, para promover uma melhoria no processo de convergência das redes MLP, é necessária a utilização de métodos de segunda ordem no processo de treinamento. O método de Levemberg-Marquardt é uma aproximação de segunda ordem, que leva em consideração aspectos presentes no método de Newton e no método do gradiente descendente, sendo um algoritmo de rápida convergência em redes neurais de tamanho moderado (HAYKIN, 2009).

Outro método de segunda ordem, conhecido como gradiente conjugado, que pode ser visto como um intermediário entre o método da descida mais íngreme e de Newton. O uso do gradiente conjugado é justificado pelo desejo de se aumentar a velocidade de convergência do gradiente descendente e, ao mesmo tempo, evitar as exigências computacionais concernentes ao cálculo, armazenamento e inversão da matriz hessiana, no método de Newton (HAYKIN, 2009; GOODFELLOW; BENGIO; COURVILLE, 2016). Dentre os métodos de segunda ordem,

é reconhecida a capacidade do método do gradiente conjugado de ser aplicável em problemas de larga escala, ou seja, problemas com centenas ou milhares de parâmetros ajustáveis. Desta forma, é um método muito adequado para treinamento de redes MLP e aplicado em problemas como reconhecimento de padrões, análise de séries temporais, controle e aproximação de funções (HAYKIN, 1999).

Devido à grande dimensionalidade dos dados utilizados nesta dissertação, o que aumenta bastante a quantidade de parâmetros ajustáveis (pesos) das redes neurais utilizadas, as redes NARX e FTDNN, que são, por natureza, variantes da rede MLP, utilizaram o *backpropagation* baseado no método do gradiente conjugado escalonado (HAYKIN, 2009). Esta escolha é justificada pelo fato deste método oferecer uma boa velocidade de convergência, ao passo que utiliza menos recursos computacionais. Embora seja comprovada a eficiêcia do algoritmo de Levemberg-Marquardt, sua escolha se mostrou computacionalmente ineficiente, pois à medida em que a topologia das redes aumentava, o treinamento se mostrava cada vez mais cruciante, com utilização de grandes níveis de recursos computacionais, tornando inviável o treinamento das redes neurais.

5.3 Redes Neurais Dinâmicas

As redes MLP possuem vasta aplicabilidade nos mais diversos tipos de problemas e áreas do conhecimento, com destaque para aqueles que envolvem classificação de padrões, aproximação de funções e sistemas dinâmicos. Um problema que envolva reconhecimento de padrões consiste em associar certos padrões a um determinado objeto ou classe, tais como reconhecimento de fala (PARK *et al.*, 2011), assinaturas harmônicas (BARBOSA *et al.*, 2017), de doenças cardíacas (JANG *et al.*, 2019), etc. Por sua vez, a aproximação funcional permite o mapeamento de uma função, quando não há a disponibilidade da função original, com base em um conjunto de entradas (pontos) e respostas obtidas pela rede que reproduzem o comportamento de um determinado sistema. Diferentemente dos dois tipos de problemas anteriores, os sistemas dinâmicos dependem da variável tempo. Ou seja, a saída de um sistema dinâmico em um instante de tempo t, qualquer, é dependente dos valores em instantes anteriores.

Dentro deste contexto, costuma-se classificar as redes neurais como estáticas ou dinâmicas. Nas redes estáticas, não existem laços de realimentação ou atrasadores e são caracterizadas por equações não-lineares e inexistência de memória, ou seja, as saídas se dão em função apenas das entradas atuais, sem dependência de valores passados. Portanto, para um

determinado padrão de entrada, uma saída instantânea é obtida por meio de um mapeamento linear ou não-linear, características, estas, intrínsecas aos problemas de classificação e aproximação funcional. Por outro lado, nas redes dinâmicas, as saídas não dependem somente de suas respectivas entradas atuais, mas também de entradas ou saídas em instantes de tempo passados, o que as tornam capazes de modelar as propriedades dinâmicas de um processo (GUPTA; JIN; HOMMA, 2004).

As redes MLP, discutidas anteriormente, são originalmente consideradas redes estáticas. Entretanto, de acordo com Haykin (1999), é possível fornecer uma representação implícita da informação temporal a uma rede estática, suprindo-a com propriedades dinâmicas. Em outras palavras, é possível fazer com que uma rede estática como uma MLP se torne uma rede dinâmica. Para que isso ocorra, é necessária a incorporação de estruturas de memória de curta ou longa duração na arquitetura da rede neural através de atrasos de tempo ou a inserção de laços de realimentação, tornando-a, neste caso, uma rede recorrente.

Em relação a estas últimas, a realimentação pode assumir diversos tipos de configurações e são basicamente classificadas em realimentação do tipo global ou do tipo local. No primeiro caso, as conexões são feitas entre os neurônios de uma camada para neurônios de uma camada anterior, fazendo com que a saída de um neurônio também se torne a entrada de outro, respectivamente. Por outro lado, em uma realimentação local, um neurônio realimenta a si próprio (MENEZES JÚNIOR, 2012). Ao mapear o espaço de entrada em um espaço de saída, a rede recorrente responde de maneira temporal a um sinal de entrada aplicado externamente, o que a torna apta a previsões não lineares, como as tratadas neste trabalho. Outras aplicações incluem: modelagem, equalização adaptativa de canais de comunicação, processamento de voz, controle de instalações industriais e diagnósticos de motores automotivos. Portanto, devido aos efeitos produzidos pela inserção dos laços de realimentação, este tipo de rede neural é capaz de fornecer resultados mais satisfatórios em tais tipos de aplicações do que as redes que possuem apenas mecanismos de atraso.

É importante ressaltar que, tanto as redes com atrasadores como as redes recorrentes são exemplos de redes neurais dinâmicas. Além disso, por mais que estes tipos de redes possuam diversos tipos de configurações em relação às suas plantas arquiteturais, a grande maioria delas são derivadas de redes estáticas, mais especificamente as MLP, de forma a explorar sua capacidade de mapeamento não-linear.

As próximas seções descrevem as três arquiteturas dinâmicas utilizadas neste traba-

lho, cada uma com uma forma específica de processar a informação temporal.

5.3.1 Rede Neural com Atrasos de Tempo na Entrada

A rede com atrasos de tempo (*Time Delay Neural Network* – TDNN) é uma arquitetura derivada da MLP, originalmente proposta no trabalho de Waibel (1989) para o reconhecimento de sinais de voz. Na forma como foi concebida, a TDNN consiste em uma arquitetura com alimentação adiante, em que atrasos de tempo são inseridos em todas as camadas da rede, fazendo com que todos os neurônios ocultos e da camada de saída sejam replicados temporalmente (WAIBEL, 1989).

Uma forma diferente de se utilizar os atrasos é através de sua utilização apenas na camada de entrada de uma rede estática, neste caso a MLP, tornando-a sensível a estrutura temporal dos sinais. A inserção de tais atrasadores fornece à rede memórias de curta duração e são representados por vetores compostos pela entrada atual $x(n)$ e de p valores passados $x(n-1)$, $x(n-2)$, ..., $x(n-p)$ representando uma linha de atraso de ordem p de acordo com a Equação (5.7).

$$\mathbf{x}(n) = [x(n),\, x(n-1),\, x(n-2),\, \ldots,\, x(n-p)]^T. \tag{5.7}$$

Desta forma, os pesos sinápticos da rede são ajustados para que o erro médio quadrático entre a saída da rede, $y(n)$, e a resposta deseja, $d(n)$, seja minimizado. O próximo valor da série temporal $x(n+1)$ é uma função de $\mathbf{x}(n)$, tal que

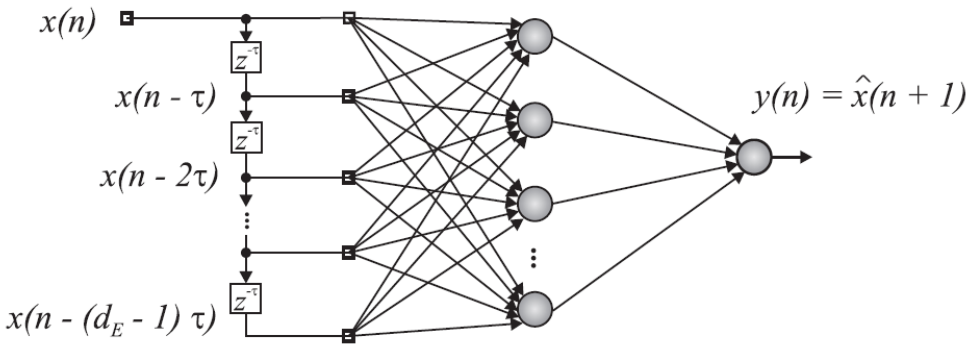
$$x(n+1) = f[\mathbf{x}(n)], \tag{5.8}$$

e a saída da rede fornece uma aproximação de $x(n+1)$, ou seja,

$$y(n) = \hat{x}(n+1). \tag{5.9}$$

A rede neural dinâmica cujos atrasadores estão presentes na camada de entrada é, portanto, um caso específico da TDNN, chamada de *Focused Time Delay Neural Network* – FTDNN, em que o termo "*focused*" indica que os atrasos estão focados somente na entrada (PRINCIPE; EULIANO; LEFEBVRE, 2000). A Figura 14 ilustra a arquitetura de uma FTDNN em que suas entradas e saídas são definidas de acordo com Equação (5.7) e (5.9), respectivamente. Ressalta-se que a janela de tempo representada pela Equação (5.7) é convolucionada ao longo de todo sinal em questão.

Figura 14 – Rede FTDNN com os atrasos de tempo na entrada.



Fonte – Menezes Júnior (2006).

Cabe lembrar que uma das vantagens em se utilizar redes dinâmicas derivadas da MLP é a capacidade de se utilizar o algoritmo *backpropagation* para seu treinamento, como é o caso da FTDNN e da rede NARX apresentada a seguir.

5.3.2 Rede Neural com Variáveis Exógenas

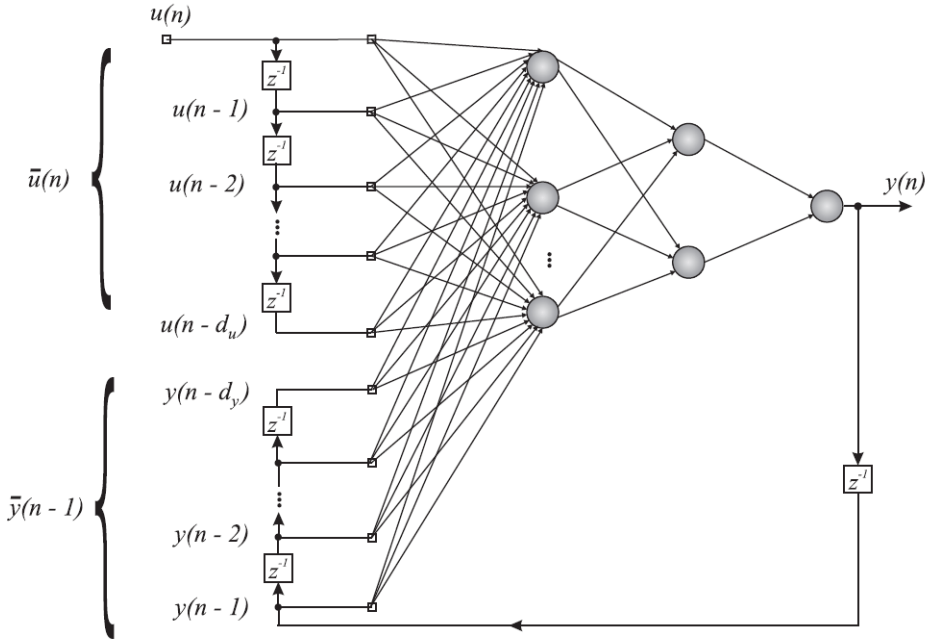
O modelo Autorregressivo não-linear com entradas exógenas (*Nonlinear Autoregressive with Exogenous Inputs* – NARX) é um importante tipo de rede neural dinâmica recorrente. Vários trabalhos já demonstraram a capacidade deste modelo em modelar sistemas não lineares. O comportamento dinâmico da rede NARX é descrito por

$$y(n+1) = f[y(n),\ \ldots,\ y(n-d_y+1); u(n),\ \ldots,\ u(n-d_u+1)], \tag{5.10}$$

sendo $f(.)$ uma função não-linear de ambos os regressores, $u(n)$ e $y(n)$. Portanto, as entradas deste tipo de modelo são constituídas de dois componentes em um determinado instante n: os valores passados e presentes da entrada, de uma variável exógena, advinda de fora da rede, representados por $u(n), u(n-1), ..., u(n-du+1)$; e os valores de saída da rede atrasados no tempo representados por $y(n), y(n-1), ..., y(n-dy+1)$. Estes valores, por sua vez, são realimentados para entrada da rede (HAYKIN, 2009). As variáveis du e dy representam a ordem de memória das componentes de entrada e saída, respectivamente. A Figura 15 ilustra um modelo NARX composto por uma única camada oculta. Nota-se que apenas um dos regressores é realimentado, enquanto o outro é apenas atrasado no tempo.

Verifica-se, portanto que a saída do modelo NARX é considerada uma estimativa da resposta de algum sistema dinâmico não-linear, o qual se deseja modelar. A saída é realimentada para a entrada da rede MLP como parte da configuração normal da NARX. Devido a isso, durante o treinamento, a saída estimada da rede está sempre disponível na entrada, fazendo com que os

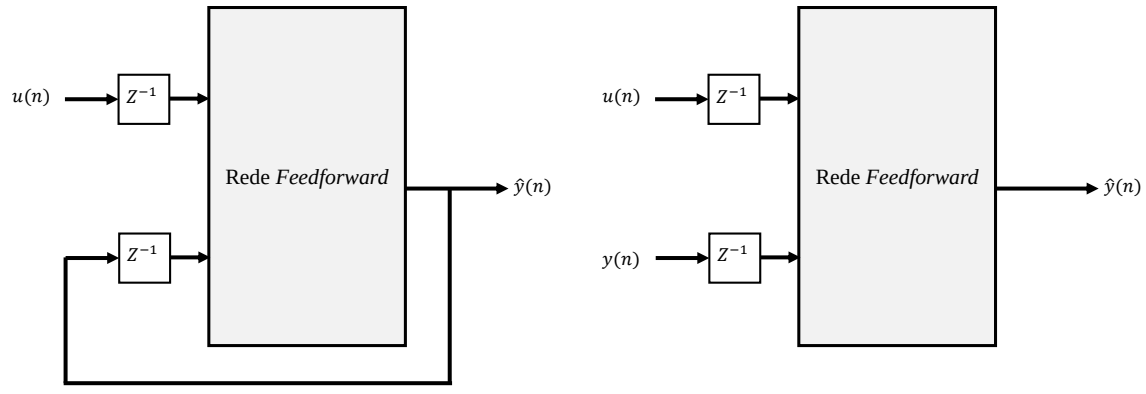
Figura 15 – Rede Neural com variáveis Exógenas (NARX).



Fonte – Menezes Júnior (2006).

erros de previsão sejam também realimentados. Este modo de treinamento é conhecido como configuração paralela. Uma outra forma de se realizar o treinamento, chamada de configuração série-paralela, consiste em realimentar para a entrada da rede somente a resposta desejada, em vez da resposta estimada. A vantagem em se utilizar a configuração de treinamento série-paralela consiste em fornecer um sinal de entrada mais preciso à RNA, sem a existência de erros de previsão. Neste trabalho, este modo de treinamento é sempre utilizado. Durante a fase de teste, entretanto, a previsão se torna recursiva utilizando-se, portanto, arquitetura no modo paralelo. Ambas as arquiteturas são ilustradas na Figura 16

Figura 16 – Arquitetura nos modos Paralelo e Série-Paralelo da rede NARX, respectivamente.



Arquitetura Paralela Arquitetura Série-Paralela

Fonte – O autor.

É interessante notar que em diversos trabalhos a NARX tem sido aplicada como uma ferramenta de modelagem de entrada-saída em identificação de sistemas não-lineares (SAHOO; DASH; RATH, 2013). Para a tarefa de previsão de séries temporais a NARX é comumente utilizada em séries temporais multivariadas, em que uma destas séries consiste em uma variável exógena. Ou seja, o modelo aproxima um valor futuro de uma série temporal com base nos valores passados da própria série que se deseja prever e valores passados e atuais de uma outra série (exógena) que exerce influência sobre a série que se deseja prever.

Como o foco desta dissertação consiste na previsão de séries temporais univariadas sem o uso de variáveis exógenas, é proposta a utilização de um modelo NARX, desenvolvido por Menezes Júnior e Barreto (2008) em que a rede é utilizada na previsão recursiva de séries univariadas de natureza caótica e com dependência temporal de longa duração. Nesta dissertação, tal modelo é utilizado na previsão de curvas de carga de energia de uma concessionária de energia elétrica. O modelo NARX proposto em Menezes Júnior e Barreto (2008) é brevemente discutido a seguir.

A principal diferença ocorre na concepção dos regressores $u(n)$ e $y(n)$. O regressor de entrada, $u(n)$, passa a ser composto por coordenadas de atraso da própria série que se deseja prever, e não mais por valores passados de uma série exógena como é descrito na forma original da NARX. Desta forma, o regressor de entrada passa a ser composto por d_e amostras da série temporal amostradas a cada τ unidades de tempo, como na Equação (5.11),

$$u(n) = [x(n),\ x(n-\tau),\ x(n-2\tau),\ \ldots,\ x(n-(d_e-1)\,\tau]. \tag{5.11}$$

Como a NARX possui dois modos distintos de treinamento, o regressor $y(n)$ no modo de operação série-paralelo (SP) e paralelo (P) é descrito, respectivamente, pelas Equações (5.12) e (5.13), a seguir,

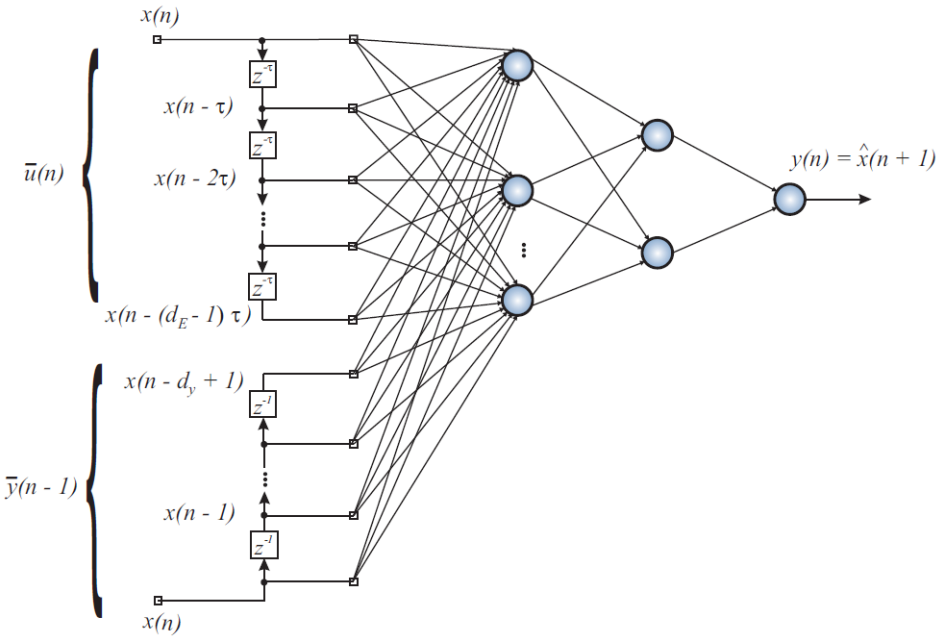
$$y_{sp}(n) = [x(n),\ \ldots,\ x(n-d_y+1)], \tag{5.12}$$

$$y_p(n) = [\hat{x}(n),\ \ldots,\ \hat{x}(n-d_y+1)]. \tag{5.13}$$

Enfatiza-se que no modo série-paralelo o regressor y_{sp} contém d_y valores passados da série temporal real, enquanto no modo paralelo o regresor y_p possui d_y valores passados das estimativas da série temporal. O valor do parâmetro d_y pode ser determinado a partir da Equação $d_y = \tau \times d_e$. Os parâmetros d_y e d_e são denominados atraso de imersão dos regressores de entrada

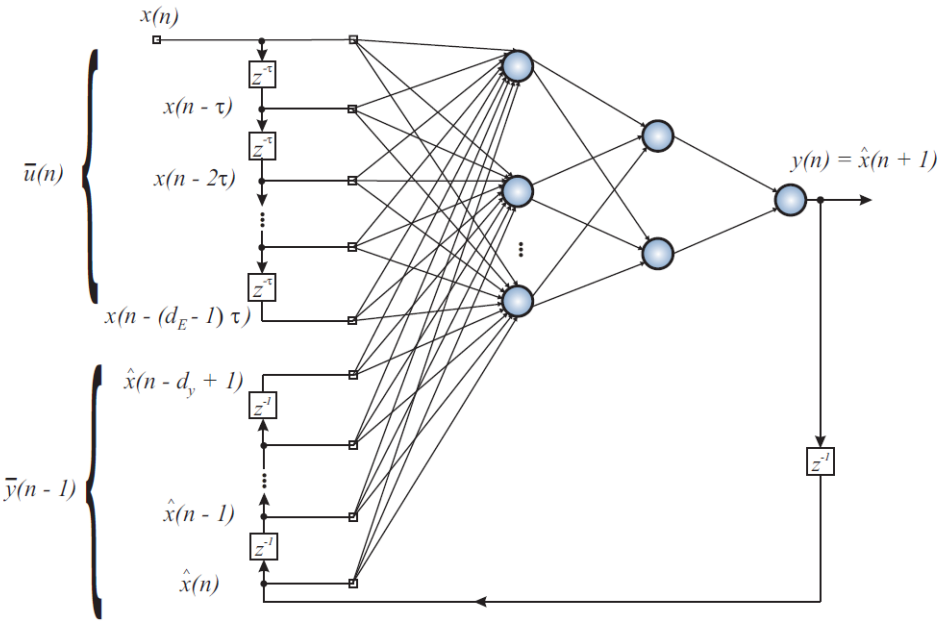
e saída, respectivamente, enquanto o parâmetro τ é denominado atraso de imersão. Nota-se, portanto, que apenas o regressor de entrada u_n é dependente do parâmetro τ, o que significa que o regressor de saída é composto apenas de amostras em intervalos de tempo consecutivos. As Figuras 17 e 18 ilustram os modos de operação no treinamento do modelo NARX utilizado.

Figura 17 – Rede NARX com arquitetura em modo de operação Série-Paralelo.



Fonte – Menezes Júnior (2012).

Figura 18 – Rede NARX com arquitetura em modo de operação Paralelo.

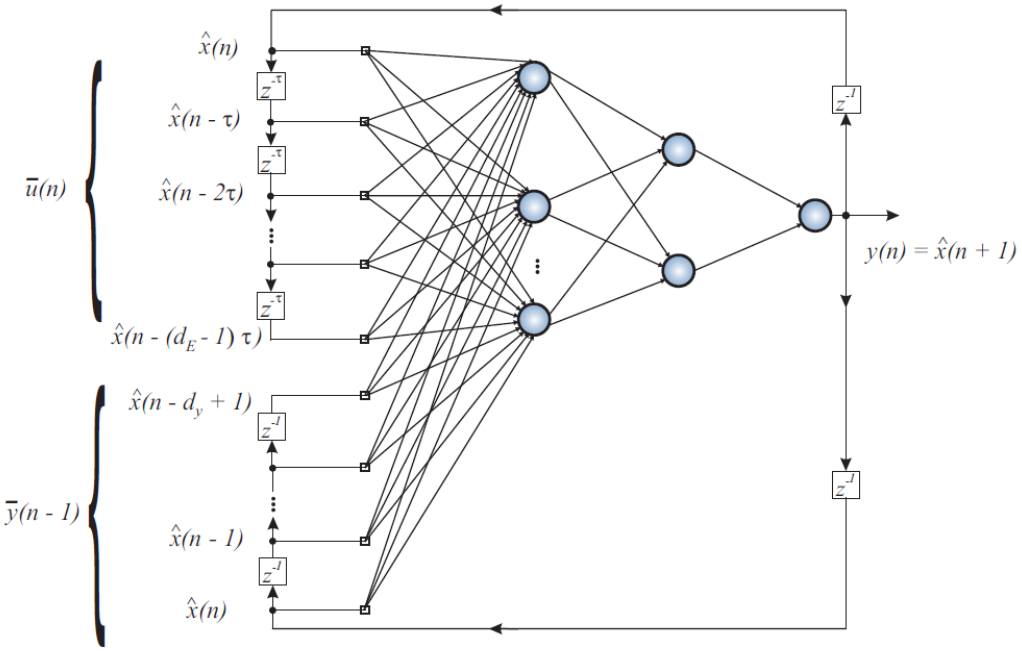


Fonte – Menezes Júnior (2012).

Independentemente do modo de operação escolhido, o modelo adaptado da NARX permite a previsão de uma série univariada, dependente apenas dos valores assumidos pela própria série, sem a necessidade de uma variável exógena. A partir desta configuração, é possível perceber que o regressor de entrada $u(n)$ supre informações de médio e longo prazo sobre a dinâmica da série temporal, enquanto o regressor de saída $y(n)$ provê informações de curto prazo (MENEZES JÚNIOR, 2006).

Esta característica também é apresentada durante a fase de teste da rede. À medida que o número de previsões de longo prazo sejam requeridas, os valores previstos devem realimentar tanto o regressor de entrada $u(n)$ quanto o de saída $y(n)$, simultaneamente, resultando em dois laços de realimentação, como ilustra a Figura 19.

Figura 19 – Rede NARX durante a fase de teste (predição recursiva).

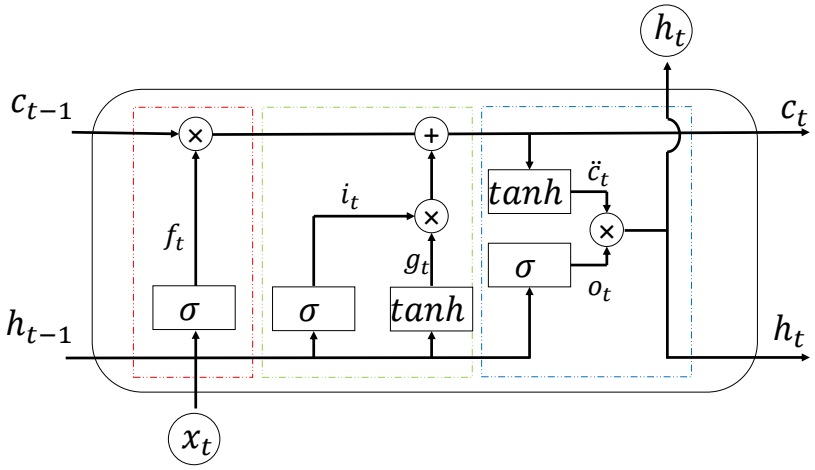


Fonte – Menezes Júnior (2012).

5.3.3 Rede Neural com Memória de Longo Prazo

A rede neural com memória de longo prazo (*Long Short-Term Memory* - LSTM) foi primeiramente introduzida por Hochreiter e Schmidhuber (1997). A arquitetura de uma LSTM consiste em um mecanismo de portas (*gates*) que permitem que dependências temporais de longo prazo sejam captadas de maneira mais fácil durante o *backpropagation*, regulando o fluxo da informação. Tais portas permitem escolher quais informações são importantes e quais devem ser descartadas (GREFF *et al.*, 2016).

Figura 20 – Célula de uma LSTM. Em vermelho: porta de esquecimento; verde: porta de entrada; azul: porta de saída.



Fonte – Adaptada de Olah (2015).

Diferentemente das RNAs convencionais, cujas unidades internas são conhecidas como neurônios, nas LSTM existe uma estrutura chamada célula, que reúne o conjunto de portas e um mecanismo de memória denominado estado de célula (*cell state*), responsável por armazenar as informações oriundas das portas. A Figura 20 ilustra os componentes de uma célula da LSTM.

O problema do *Vanishing Gradient* é conhecido como uma das causas frequentes do baixo desempenho de redes neurais recorrentes em certos problemas de previsão de séries temporais, fazendo com que somente informações com dependências de curto prazo sejam capturadas pela RNA, em detrimento das informações com dependência de longo prazo, que se perdem (HAYKIN, 2009). A rede LSTM contorna o problema do *Vanishing Gradient* por meio da utilização de portas que retem ou descartam seletivamente informações. Além das portas, um conceito importante de uma LSTM consiste no estado de célula $\mathbf{c}_t$, que age como a "memória" da rede neural e é o responsável por levar informações importantes acerca do processo. Em teoria, a existência de tais mecanismos permite à LSTM um melhor desempenho quando utilizada para análise de dados sequenciais, quando comparada com outras RNAs mais simples (HOCHREITER; SCHMIDHUBER, 1997; GREFF *et al.*, 2016).

Cada uma das portas presentes na Figura 20 são definidas pelas seguintes equações:

$$\mathbf{f}_t = \sigma_g(\mathbf{W}_f \mathbf{x}_t + \mathbf{R}_f \mathbf{h}_{t-1} + b_f) \tag{5.14}$$

$$\mathbf{i}_t = \sigma_g(\mathbf{W}_i \mathbf{x}_t + \mathbf{R}_i \mathbf{h}_{t-1} + b_i) \tag{5.15}$$

$$\mathbf{g}_t = \sigma_c(\mathbf{W}_g\mathbf{x}_t + \mathbf{R}_g\mathbf{h}_{t-1} + b_g) \tag{5.16}$$

$$\mathbf{o}_t = \sigma_g(\mathbf{W}_o\mathbf{x}_t + \mathbf{R}_o\mathbf{h}_{t-1} + b_o), \tag{5.17}$$

em que σ_g e σ_c correspondem, respectivamente, às funções logística e tangente hiperbólica.

Na célula, o primeiro lugar por onde a informação segue, é pela porta de esquecimento $\mathbf{f}_t$ (*forget gate*), Equação (5.14), responsável por decidir qual informação deve ser descartada do estado de célula anterior $\mathbf{c}_{t-1}$. Esta decisão é feita por meio de uma função sigmóide, geralmente do tipo logística, que leva em conta tanto o estado de célula anterior, quanto a entrada atual $\mathbf{x}_t$, e retorna um número entre 0 e 1, que indica o grau de relevância da informação.

Ao mesmo tempo em que a informação de entrada segue por $\mathbf{f}_t$, é necessário escolher qual nova informação deve ser inserida no estado de célula. Esta etapa é feita na parte destacada em verde na Figura 20. Primeiramente, tanto a entrada $\mathbf{x}_t$ quanto a saída anterior $\mathbf{h}_{t-1}$ passam pela porta de entrada $\mathbf{i}_t$ (*input gate*), que também consiste em função logística, esta mesma informação é submetida a uma função tangente hiperbólica, que são os estados de célula candidatos, $\mathbf{g}_t$, Equação (5.16). O resultado de ambas as funções $\mathbf{i}_t$ e $\mathbf{g}_t$ são, então, multiplicados.

Os resultados obtidos previamente são utilizados para atualizar o estado de célula,

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t, \tag{5.18}$$

sendo que $\odot$ é o produto de Hadamard. Portanto, o estado de célula $\mathbf{c}_t$ é atualizado tanto utilizando o valor oriundo da porta de esquecimento $\mathbf{f}_t$ como o produto entre $\mathbf{i}_t$ e $\mathbf{g}_t$, este último decide o quanto o novo estado de célula $\mathbf{c}_t$ deve ser atualizado pelos estados candidatos $\mathbf{g}_t$, portanto $\mathbf{i}_t \odot \mathbf{g}_t$ é um escalonamento de $\mathbf{i}_t$ em $\mathbf{g}_t$.

A saída da célula, definida por $\mathbf{h}_t$, é dada em função do produto entre a Equação (5.17) e da tangente hiperbólica do estado de célula atualizado $\mathbf{c}_t$,

$$\mathbf{h}_t = \mathbf{o}_t \odot \sigma_c(\mathbf{c}_t). \tag{5.19}$$

Percebe-se que em todos os casos, as portas dependem sempre da entrada atual $\mathbf{x}_t$ e da saída anterior da célula $\mathbf{h}_{t-1}$. Além disso, como visto na Equação (5.19), a saída $\mathbf{h}_t$ é sempre dada em função do estado de célula, que, por sua vez, age como um meio que "transporta" as

informações relevantes armazenadas. As Equações (5.14) a (5.19) não contém o uso de termos referentes às conexões *peepholes* (uma variação da rede LSTM proposta por Gers e Schmidhuber (2000)), como pode ser visto em Greff *et al.* (2016). Os termos **W**, **R** e **b** são matrizes que denotam, respectivamente os pesos sinápticos de entrada, os pesos sinápticos das conexões recorrentes e os limiares de cada componente, organizadas da seguinte forma:

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_f \\ \mathbf{W}_i \\ \mathbf{W}_g \\ \mathbf{W}_o \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \mathbf{R}_f \\ \mathbf{R}_i \\ \mathbf{R}_g \\ \mathbf{R}_o \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_f \\ \mathbf{b}_i \\ \mathbf{b}_g \\ \mathbf{b}_o \end{bmatrix}. \tag{5.20}$$

Assim como outros tipos de redes neurais recorrentes, as LSTMs são treinadas por meio de uma adaptação do algoritmo *backpropagation* denominada *backpropagation through time* - BPTT (HAYKIN, 2009; GREFF *et al.*, 2016), e utiliza algumas versões do gradiente descendente estocástico (*Sthocastic Gradient Descent* - SGD) para cálculo dos gradientes durante a retropropagação do erro. A cada iteração, o SGD avalia o gradiente e atualiza os parâmetros da rede utilizando um subconjunto dos dados de treinamento, em vez do conjunto inteiro. O subconjunto é denominado mini-lote (*mini-batch*). Assim que todos os mini-lotes são apresentados à rede neural, tem-se uma época de treinamento. A LSTM utilizada nesta dissertação utiliza um tipo específico do SGD, conhecido como *root mean square propagation* - RMSProp (cujo uma das características é uma taxa de aprendizagem adaptativa), que pode ser visto em Tieleman e Hinton (2012) e Goodfellow, Bengio e Courville (2016) e encontrado na *toolbox* de *Deep Learning* do software MATLAB, utilizada para implementação da LSTM nesta dissertação. Outros detalhes sobre a LSTM podem ser encontrados em Graves e Schmidhuber (2005), Goodfellow, Bengio e Courville (2016), Greff *et al.* (2016) e Graves (2012).

Também é importante mencionar que a mesma *toolbox* foi utilizada para implementação das redes NARX e FTDNN. Neste caso, foram utilizadas redes MLP e realizou-se a adaptação e inserção de laços de realimentações nestas redes para que, assim, fosse possível obter as arquiteturas das redes NARX e FTDNN.

No próximo capítulo, é discutida a metodologia empregada neste estudo. Detalhando como foi feita a utilização das ferramentas computacionais discutidas até o presente capítulo.

6 METODOLOGIA DE PROJETO

6.1 Introdução

Este capítulo tem por objetivo apresentar de maneira sucinta a metodologia adotada durante o estudo. Primeiramente, é dada ênfase ao cenário de estudo e ao conjunto de dados utilizados. Após isso, o processo de filtragem dos dados é explicado, para que então se obtenha o conjunto de dados a ser de fato trabalhado junto às outras ferramentas utilizadas. A primeira etapa pós-filtragem, que consistiu no agrupamento das séries temporais, é descrita logo em seguida. Na segunda etapa, que consistiu na previsão das séries temporais, é explicada a forma como os dados foram apresentados às redes neurais, assim como suas configurações de parâmetros e topologias. Por último, são apresentadas as métricas de avaliação aplicadas aos resultados obtidos.

É importante salientar que antes e durante a realização das etapas descritas a seguir, foi realizada uma extensa revisão bibliográfica sobre as possíveis ferramentas que poderiam ser empregadas neste estudo, assim como uma busca sobre diversos trabalhos relacionados e a forma como pesquisadores têm abordado o problema da previsão de curvas de demanda.

Todos os algoritmos necessários para a realização desta dissertação foram implementados pelo autor. As simulações foram feitas em microcomputador com processador de quatro núcleos, com frequência de até 4 GHz e 8 GB de memória RAM. Por conveniência, algumas simulações foram realizadas com auxílio da unidade de processamento gráfico, quando possível.

6.2 Cenário de estudo

O cenário de estudo consiste em curvas de carga cedidas por uma distribuidora de energia elétrica localizada no estado do Piauí. Os sinais correspondem a curvas diárias de demanda de energia elétrica, medidas e armazenadas em intervalos de tempo de cinco minutos. Assim, cada curva corresponde a um intervalo de 24 horas, composto por 288 amostras. As curvas foram medidas durante os anos de 2012 a 2018, em uma linha de distribuição de 69 kV. A carga alimentada, possui característica predominantemente comercial e residencial, composta por 36.803 consumidores, como mostra a Tabela 2

Tabela 2 – Quantidade e tipos de consumidores supridos pela linha de distribuição.

Alimentador	Residencial	Comercial	Industrial	Rural	Outros
1	1817	1757	30	0	54
2	2376	260	11	0	41
3	2541	515	15	1	49
4	1470	939	18	0	54
5	3442	364	24	3	35
6	4226	716	27	2	44
7	9537	745	48	5	103
8	4075	1329	37	0	93

6.3 Pré-processamento dos Dados

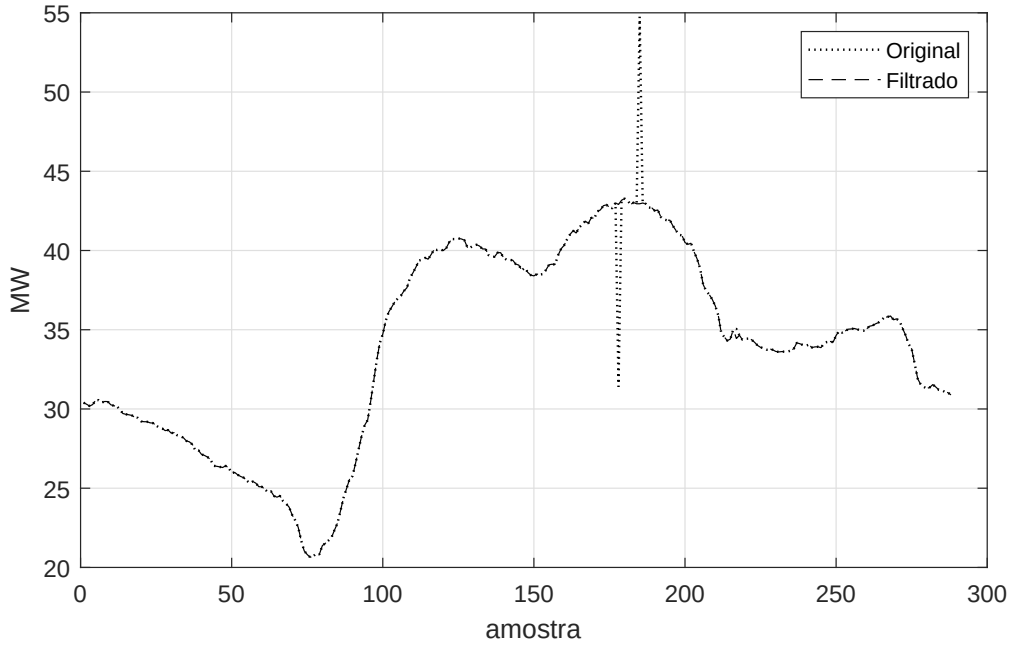
A etapa de pré-processamento consistiu em dois procedimentos. No primeiro, alguns dados que apresentavam pequenos problemas decorrentes de falhas na medição, pequenas faltas ou indisponibilidade, foram recuperados por meio da utilização do Identificador de Hampel. Com o segundo procedimento, os dados em que não houve possibilidade de resolver tais problemas por meio da utilização do Identificador de Hampel, devido a limitações do próprio algoritmo, foram removidos do conjunto de dados. Isto foi feito, pois a permanência de tais dados poderia interferir de maneira negativa tanto no desempenho dos algoritmos de agrupamento quanto no das RNAs utilizadas para previsão. Cada um dos procedimentos é descrito a seguir.

6.3.1 Filtragem por Identificador de Hampel

Como discutido na Seção 3.5, o Identificador de Hampel é utilizado como um identificador de *outlier*. Considerando uma curva de carga típica, um *outlier* seria qualquer instante de tempo cuja a amplitude correspondesse à uma mudança abrupta do sinal, o que seria equivalente à uma falta, ou falha durante a aquisição dos dados.

Muitos dos sinais utilizados neste estudo possuem um ou mais pontos em que é possível se perceber tais anomalias. Para evitar que tais dados fossem meramente descartados, optou-se pela tentativa de corrigir estes trechos defeituosos, quando possível, e reintegrá-los ao conjunto de dados sob estudo. A Figura 21 ilustra uma curva originalmente composta por dois picos que não deveriam fazer parte do comportamento dinâmico da curva em circunstâncias normais. Após a aplicação do Identificador de Hampel, observa-se que ambos os picos foram removidos, pois foram identificados como *outliers*, o restante do sinal permanece idêntico ao original.

Figura 21 – Curva de carga com dois possíveis problemas de medição filtrada pelo Identificador de Hampel.

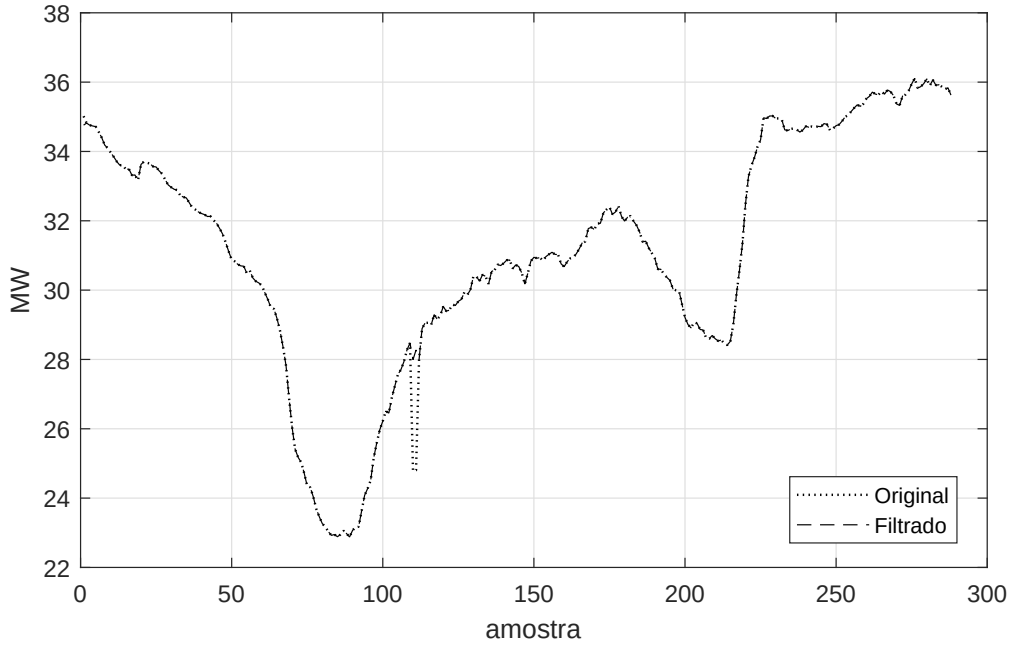


Para que fosse obtido o mesmo resultado em todas as curvas que continham problemas semelhantes, diversas combinações de parâmetros foram testados, de forma que, ao se realizar a aplicação do filtro, trechos com dinâmicas normais não fossem alterados. Considerando-se tal aspecto, e os parâmetros previamente explicados na Seção 3.5, ajustou-se o filtro para que a mediana de quatro amostras que precedem e outras quatro que sucedem uma determinada amostra $x(n)$ de um sinal qualquer $\mathbf{x} = [x(1),\ x(2),\ ...,\ x(N)]$, em que N é o número de amostras que compõem o sinal, fosse calculada. Caso a diferença entre os valores da amostra $x(n)$ e da mediana fosse maior que três desvios padrões, conforme Equação (3.14), a amostra era, então, substituída pela mediana, caso contrário o valor continuava o mesmo. Outro exemplo dos resultados obtidos pode ser visto na Figura 22.

Embora seja possível perceber a remoção dos trechos corrompidos, em alguns casos, quando existiam várias falhas sucessivas e muito próximas entre sí, não era possível se obter uma boa reconstrução do sinal, pois como a mediana é calculada entre pontos adjacentes, a presença de muitos *outliers* próximos fazia com que o valor obtido não fosse condizente com a dinâmica da curva naquele instante.

Desta forma, era necessário haver outra forma de impedir que tais dados fizessem parte do conjunto de dados, isto foi feito por meio do uso da Transformada *Wavelet*.

Figura 22 – Curva de carga com problema de continuidade e remoção de *outlier* por Identificador de Hampel.

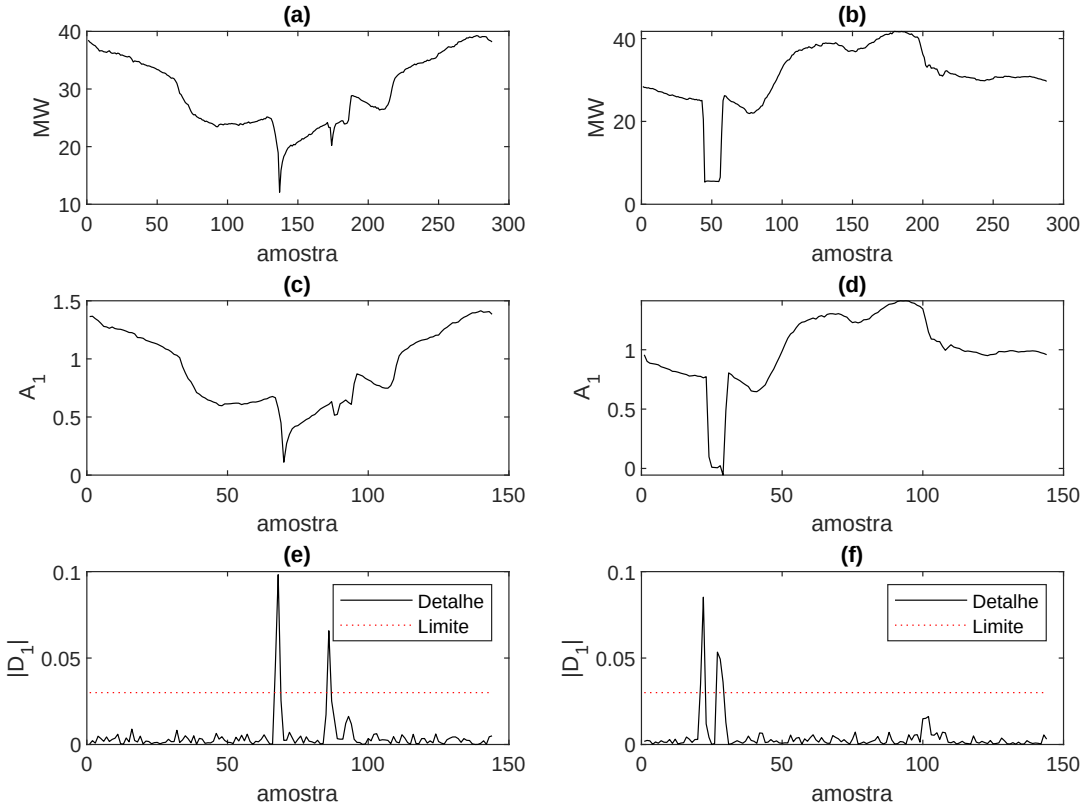


6.3.2 Filtragem por Transformada Wavelet

Muitos dos sinais existentes no conjunto de dados, possuíam problemas intratáveis pelo identificador de Hampel. Devido a isso, todos os dados são novamente filtrados por um algoritmo que tem como base o uso da TW. O principal intuito em se utilizar o Identificador de Hampel antes da etapa descrita no decorrer desta seção, era evitar que curvas que possuíam problemas de menor significância e que fossem tratáveis, fossem excluídas do conjunto de dados, visto que a etapa seguinte visa justamente a remoção dos dados que apresentavam mudanças abruptas de comportamento. Curvas de carga, geralmente, não costumam apresentar grandes mudanças de amplitude em curtos intervalos de tempo, principalmente quando esta mudança ocorre quase que instantaneamente. Na maioria das vezes, tal comportamento tem a ver com algum tipo de falha no sistema elétrico ou falha no sistema de medição/aquisição. E, como citado anteriormente, estas falhas podem prejudicar a capacidade de aprendizagem de um sistema baseado em inteligência computacional como o proposto nesta dissertação.

A Figura 23 ilustra dois exemplos de curvas em que ocorreram mudanças repentinas de seus comportamentos e cujo o problema não foi possível resolver a partir do método utilizado na seção anterior. Como explicado no Capítulo 3, uma das formas de se implementar a TW é por meio da técnica de AMR, em que o sinal pode ser decomposto em componentes de alta e baixa frequência, equivalente à uma filtragem por filtros passa-altas e passa-baixas. Na TW, os

Figura 23 – Duas curvas com mudanças abruptas de amplitude (Figuras (a) e (b)). Abaixo, constam os coeficientes de aproximação A_1 e o módulo dos coeficientes D_1, junto ao limite fixado.



coeficientes de detalhe representam as componentes de alta frequência, enquanto os coeficientes de aproximação representam as componentes de baixa frequência.

Nos sinais mostrados na Figura 23, as amostras nas quais houveram queda abrupta de energia foram indicadas com valores elevados dos coeficientes de detalhe, em virtude dos mesmos estarem associados às altas frequências do sinal. Para eliminar estes sinais do conjunto de dados estabeleceu-se um limite, para o qual os coeficientes de detalhe não deveriam ultrapassar. Este limite foi estabelecido tomando como base a análise de diversos testes e verificando-se um valor ideal a ser adotado. Desta forma, para este conjunto de dados, especificamente, ficou estabelecido um valor de 0,03 como limiar fixo. É importante observar que, antes de passarem pela TW, cada sinal foi normalizado com base em seus respectivos valores máximos e mínimos. Durante os testes, verificou-se que a *wavelet* Daubechies-4 foi a que apresentou melhores resultados, em virtude de oferecer maior amplitude de detalhe nos pontos em que ocorriam as descontinuidades, ao passo que pequenas variações no sinal refletiam em baixas amplitudes dos coeficientes.

Cabe destacar que o método de filtragem utilizado serve apenas como meio de manter certas curvas por meio da correção de trechos que apresentem pequenos erros de amplitude, ou a eliminação das curvas cujos problemas não são possíveis serem contornados. Não há, portanto, a substituição destes dados, uma vez que seria necessária a construção de uma metodologia que tornasse possível a reconstrução dos mesmos de uma maneira mais ampla e eficaz que a utilizada pelo Identificador de Hampel.

Após a realização da etapa de pré-processamento dos dados, foram realizados os processos de agrupamento e previsão, discutidos nas próximas duas seções.

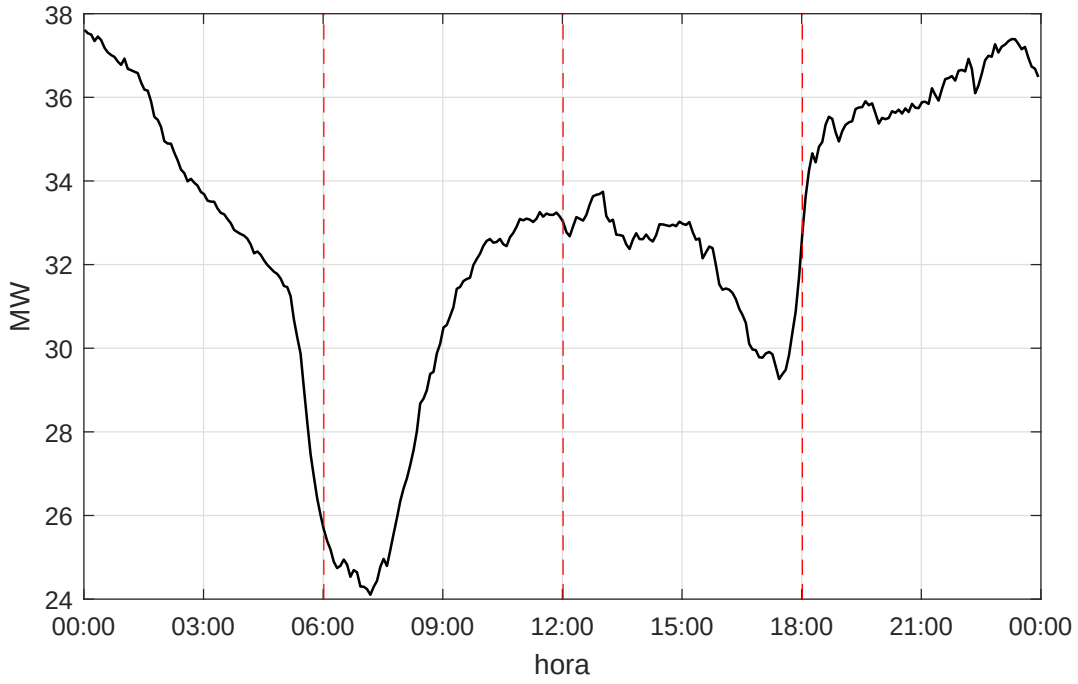
6.4 Agrupamento dos Dados

Com objetivo principal de reduzir os erros de previsão, são utilizados algoritmos de agrupamento que visam a formação de grupos de treinamento com alto grau de correlação. Por meio desta metodologia é possível obter conjuntos de treinamento cujos dados possuem grande semelhança entre si, facilitando o ajuste dos pesos sinápticos das RNAs.

Nesta dissertação são utilizados dois tipos de algoritmos particionais: *k-means* e o *minCEntropy*. Como descrito no Capítulo 4, uma das principais desvantagens de tais métodos consiste na suscetibilidade que os resultados apresentados por estes algoritmos possuem às suas condições de inicialização. Ou seja, dependendo da posição em que os centroides gerados inicialmente se encontram, o algoritmo muitas vezes pode convergir para resultados diferentes, chegando, em alguns casos, a convergir para ótimos locais que se encontram demasiadamente distantes do ótimo global. Com a finalidade de mitigar este problema, obter maior robustez e combinar as melhores características apresentadas por cada algoritmo, é proposta a utilização da combinação dos resultados obtidos por estes dois algoritmos por meio de função consenso baseada em votação. Detalhes sobre esta abordagem são dados na Seção 4.4.

Antes de prosseguir com a tarefa de agrupamento, foi necessário definir, antes, os atributos que serviriam como entradas dos algoritmos. Estes atributos foram verificados tanto no *k-means*, como no *minCEntropy*, até que fosse encontrada uma combinação que fosse capaz de fornecer os melhores resultados em ambos algoritmos, considerando-os individualmente. A princípio, optou-se por utilizar apenas as curvas de carga propriamente ditas como atributos. No entanto, foi verificado que dependendo do horário e dia, as curvas podem fornecer comportamentos distintos. Devido a isso, foram extraídos atributos estatísticos que tornavam mais evidente estas diferenças.

Figura 24 – Divisão de intervalos para extração de atributos. Cada intervalo corresponde a uma duração de 6h.

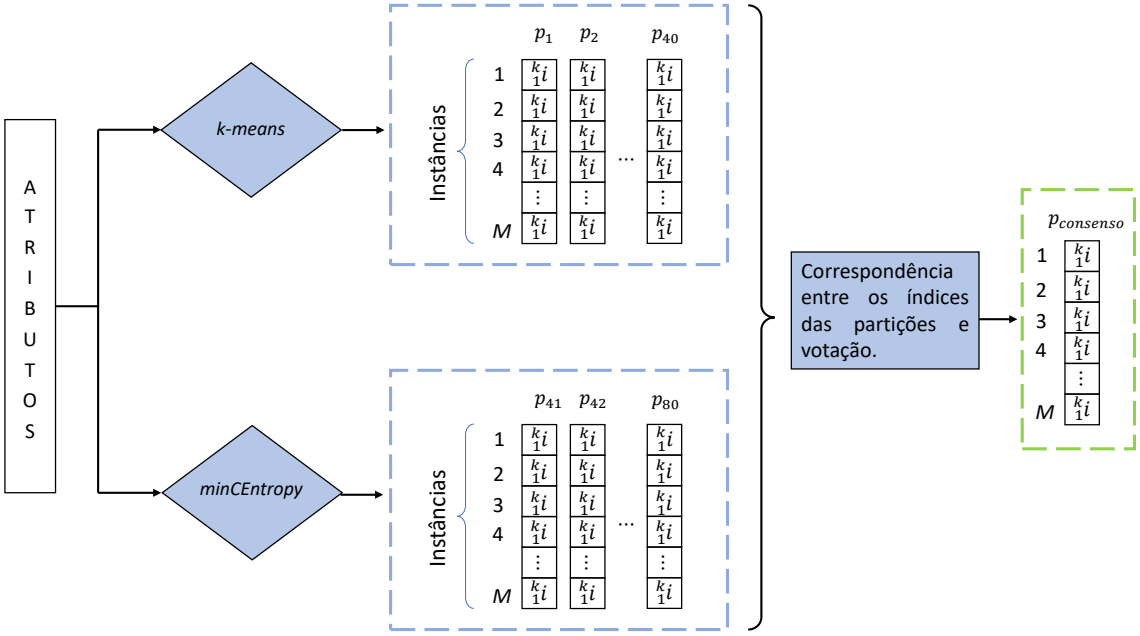


A partir da Figura 24, percebe-se que cada instância presente no conjunto de dados foi dividida em quatro partes bem específicas, cada uma com padrões de consumo distintos, das quais foram extraídas as seguintes características: máximo, mínimo, média, desvio padrão e variância. Tal abordagem permite verificar diferenças com base em quatro intervalos horários diferentes, permitindo que mesmo curvas com dinâmicas semelhantes ao longo da maior parte do tempo sejam diferenciadas com base em mudanças dinâmicas que podem ocorrer em um ou mais destes intervalos. Levando-se em conta este aspecto, os atributos de entrada foram definidos como sendo um vetor $\mathbf{V}_k$ composto pela curva de carga $\mathbf{x}_k = [x(1),\ x(2),\ x(3),\ ...,\ x(N)]$, e pelo máximo, mínimo, desvio padrão e variância de cada um dos i intervalos de $\mathbf{x}_k$. Em que N é o número de amostras da série temporal e k é a k-ésima série temporal do conjunto de dados. Como mostram as Equações (6.1) e (6.2), onde $\mathbf{g}_k^i$ é o vetor com as medidas estatísticas do intervalo i de $\mathbf{x}_k$:

$$\mathbf{g}_k^i = [min(\mathbf{x}_k^i),\ max(\mathbf{x}_k^i),\ med(\mathbf{x}_k^i),\ dp(\mathbf{x}_k^i),\ var(\mathbf{x}_k^i)], \tag{6.1}$$

$$\mathbf{V}_k = [\mathbf{x}_k,\ \mathbf{g}_k^1,\ \mathbf{g}_k^2,\ \mathbf{g}_k^3,\ \mathbf{g}_k^4]. \tag{6.2}$$

Figura 25 – Diagrama do processo de agrupamento. Cada uma das M instâncias recebe um índice i que pode variar de 1 a K, indexando cada curva a um determinado grupo nas partições. Após a reatribuição dos índices é realizada a votação.



Fonte – O autor.

Considerando-se que cada curva é composta por 288 amostras e são construídas 5 medidas estatísticas de seus 4 subintervalos, cada vetor $\mathbf{V}_k$ é composto por $(288 + 5 \times 4)$ 308 atributos de entrada. Com a finalidade de tornar evidente as diferenças de amplitude apresentadas pelas curvas, que, por sua vez, refletem a sazonalidade em que as mesmas estão submetidas, de acordo com a época do ano, os valores dos atributos foram normalizados, considerando-se os valores máximos e mínimos. Como os atributos são compostos por grandezas diferentes, considerou-se cada um dos cinco componentes do vetor $\mathbf{V}_k$ separadamente. Ou seja, cada curva foi normalizada, considerando-se os valor máximo e mínimo de amplitude de todo o conjunto de dados. O mesmo foi feito para os subintervalos, separadamente. Os dados foram escalados em faixa de valores compreendidos entre $[-1,\ 1]$.

Após definidos e normalizados, os atributos foram submetidos aos dois algoritmos de agrupamento, previamente definidos: *k-means* e *minCEntropy*. Cada algoritmo foi executado sucessivamente 40 vezes, fazendo com que diversas soluções fossem geradas levando-se em consideração as diferentes condições iniciais dos algoritmos. A Figura 25 ilustra o processo de agrupamento. Percebe-se que o resultado proveniente de cada uma das execuções correspondia a um vetor $\mathbf{p}_t$, denominado partição base, resultando, portanto, em conjunto composto por 80 $(t = [1,\ 80])$ destas partições de tamanho $M \times 1$. Sendo M o número de instâncias existente

no conjunto de dados. Cada instância recebe um índice i cujo valor varia de 1 a K e designa a qual grupo cada instância de uma partição $\mathbf{p}_t$ pertence, K é a quantidade de grupos em que cada partição base $\mathbf{p}_t$ foi dividida. Observa-se que após formadas pelos respectivos algoritmos, as partições passam por um processo de correspondência entre seus índices com base em uma partição de referência, como explicado na Seção 4.4.1. E em seguida, é feita uma votação majoritária a fim de verificar a qual grupo cada uma das instâncias foi, na maioria das vezes, designada pelas partições base. A partir deste procedimento, obtém-se uma partição consenso $\mathbf{p}_{consenso}$ que, por sua vez, representa o resultado final do agrupamento obtido.

Embora tenham sido realizados experimentos com diversos valores para K, estabeleceu-se 16 como uma quantidade adequada de grupos para o conjunto de dados em questão. Resultados mostraram que a partir deste valor, os algoritmos eram capazes de fornecer grupos mais homogêneos. Valores maiores também forneceram bons resultados, entretanto estes dados também são utilizados como treinamento pelas RNAs, necessitando, portanto, que os grupos de treinamento contenham quantidade razoáveis de instâncias. Percebe-se, portanto, que existe um *trade-off* entre qualidade e quantidade, que depende fortemente das características e tamanho do conjunto de dados a ser trabalhado.

Uma das principais vantagens desta metodologia é a possibilidade de se alcançar resultados mais próximos a ótimos globais, sem a necessidade de recorrer à uma técnica específica, possibilitando o aproveitamento de características distintas apresentadas por cada um destes algoritmos. Entretanto, é importante se atentar ao fato de que o custo computacional envolvido é maior, uma vez que mais algoritmos são necessários para chegar a este fim, assim como uma maior quantidade de suas execuções, responsáveis por gerarem as partições base, das quais é extraído o resultado final.

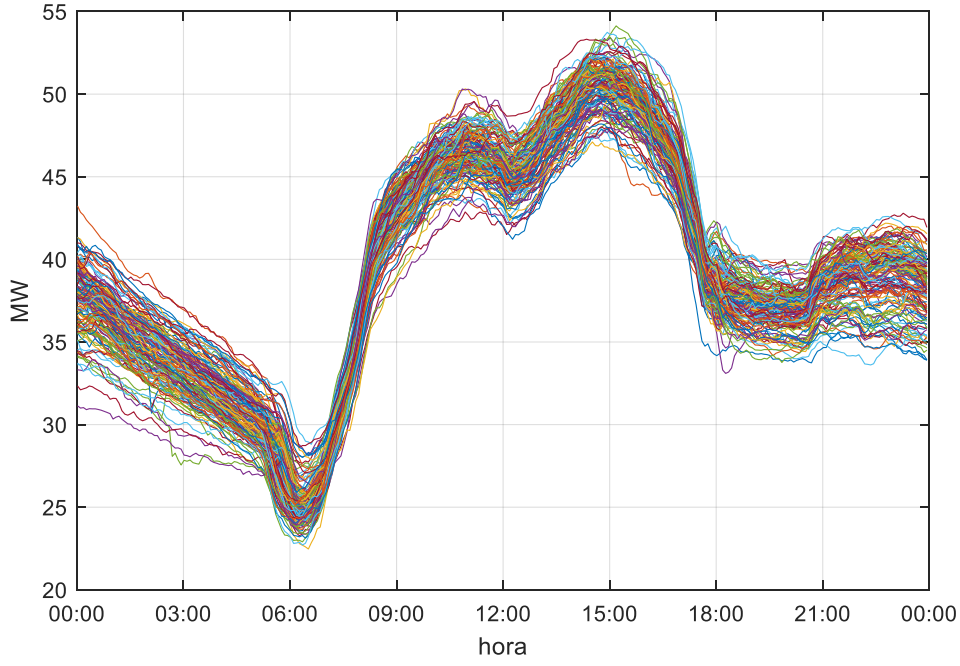
6.5 Algoritmos de Previsão

A última etapa consiste na previsão recursiva de curvas de carga, amostradas a cada cinco minutos. Assim, o problema consiste em uma previsão de 288 passos à frente. O grande desafio existente em grandes horizontes de previsão, é o acúmulo do erro que é realimentado para a entrada dos modelos à medida que as previsões são feitas. Esta dissertação também mostra que cada uma das RNAs empregadas para esta tarefa aborda a inserção de memórias de curto e longo prazo de maneiras diferentes. Na Seção 5.3 foi descrita a forma como cada uma das RNAs empregadas lidam com estes mecanismos.

A FTDNN consiste em uma MLP, cujas estruturas de memória são atrasos de tempo sucessivos na camada de entrada. Em relação a NARX, que também é uma MLP, é adicionada uma estrutura de memória a mais que caracteriza as memórias de longa dependência. Por outro lado, a rede LSTM possui uma arquitetura que difere completamente das outras duas, pois apresenta laços de recorrência em todas as suas unidades, além de mecanismos que facilitam a identificação de termos de longa dependência. Os mecanismos existentes nos dois últimos modelos citados buscam lidar com um problema muito comum chamado *vanishing gradient*, que dificulta a capacidade de aprendizagem dos modelos baseados no método do gradiente descendente.

O treinamento das RNAs é feito a partir dos agrupamentos gerados previamente. Para melhor elucidar este processo, considera-se, por exemplo, o agrupamento de curvas ilustrados na Figura 26. Cada uma das curvas que compõem este agrupamento foi gerada em determinado dia da semana, mês e ano, e pertencem a um mesmo agrupamento devido às características em comum que compartilham entre si. No treinamento das RNAs, este conjunto de curvas é utilizado para ajustar os parâmetros livres da rede. Ou seja, neste processo as RNAs consideram uma curva por vez contidas num mesmo grupo (como o mostrado no exemplo ilustrado na Figura 26), em que o objetivo é aproximar o valor fornecido (saída das RNAs) para cada amostra da série temporal, ao valor real destas amostras.

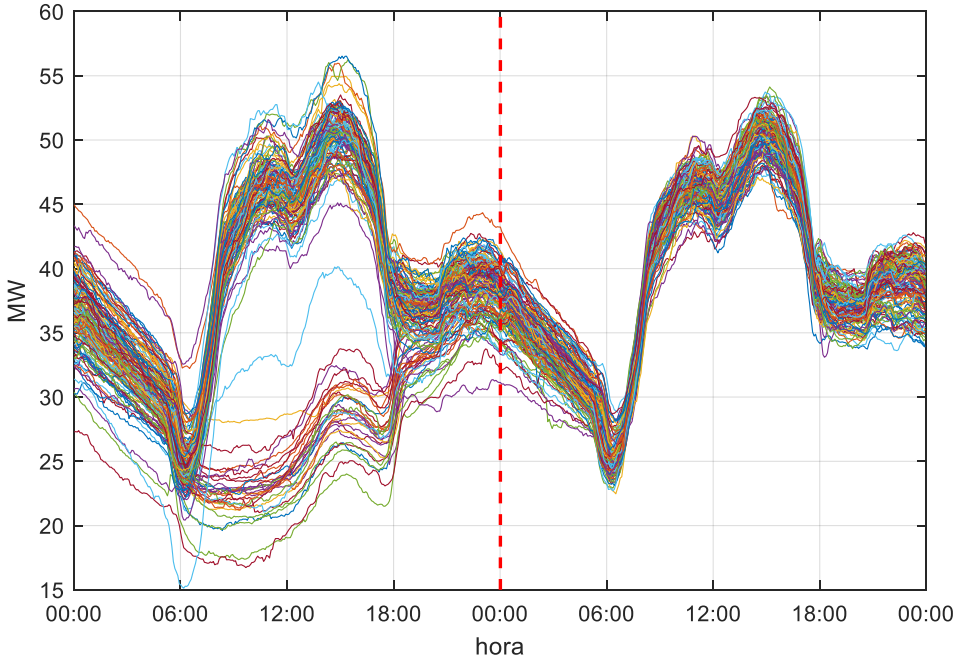
Figura 26 – Exemplo de agrupamento para treinamento das redes neurais.



Um ponto importante a ser observado, é que a curva de demanda de um determinado dia está relacionada com a curva gerada no dia anterior. Portanto, durante o treinamento é necessário que amostras da curva do dia anterior também sejam utilizadas para prever as amostras do dia desejado. Na Figura 27, as curvas mostradas no agrupamento da Figura 26, são ilustradas em conjunto com as curvas de demanda que as precedem. Aquelas que estão à esquerda da linha vertical vermelha são as curvas precedentes, às curvas mostradas na Figura 26, que por sua vez estão ilustradas novamente na Figura 27, ao lado direito da reta vertical vermelha. Em situações em que não havia disponibilidade da curva precursora devido à falta deste dado em virtude de problemas ou falhas decorrentes de faltas no sistema ou nos equipamentos de medição, foi utilizada a curva referente ao dia temporalmente mais próximo e que continham características semelhantes.

Como dito anteriormente, uma RNA considera uma curva por vez, em um agrupamento, no processo de ajuste dos pesos sinápticos. Considera-se, por exemplo, o processo de treinamento para uma única curva contida no agrupamento da Figura 26, em que a Figura 28 ilustra esta curva, juntamente com a que a precede. A curva cheia, com algumas amostras destacadas em cinza, é a precursora. Enquanto a tracejada, com algumas amostras destacadas em verde é a curva alvo do treinamento, ou seja, aquela cujo valor de cada uma das suas amostras a RNA irá tentar aproximar iterativamente. Para simplificar a visualização e o entendimento,

Figura 27 – Agrupamento das curvas com suas respectivas curvas precursoras.



optou-se por explicitar apenas 12 amostras de cada uma das curvas, em vez das 288 que cada uma possui.

Os padrões de entrada da RNA são vetores compostos por uma determinada quantidade de amostras. Estes vetores servem como meios para que a RNA seja capaz de aproximar o valor da amostra exatamente subsequente ao vetor de entrada. Considerando-se a Figura 28 e supondo-se que o tamanho do vetor de entrada (regressor) seja igual a 7, ou seja, composto por 7 amostras, no primeiro instante de tempo o regressor será formado pelas amostras $\{x(12), x(11), x(10), x(9), x(8), x(7), x(6)\}$. O mesmo é fornecido à camada de entrada da RNA e o objetivo da RNA será fornecer em sua saída o valor mais próximo possível da amostra subsequente, ou seja, a amostra $x(13)$. Na próxima iteração, o vetor de entrada irá "deslizar" uma unidade ao longo da curva, como se fosse uma janela deslizante, fazendo com que a amostra temporalmente mais distante ($x(6)$) seja descartada, e a amostra $x(13)$ seja incorporada ao regressor da camada de entrada, assim, nesta iteração, o objetivo da RNA é fornecer um valor o mais próximo possível da amostra seguinte, neste caso a amostra $x(14)$. O regressor é deslocado ao longo da curva, até que todas as amostras da curva a qual se deseja prever sejam aproximados pela RNA. Todas as iterações para o exemplo mostrado na Figura 28 estão contidas na Tabela 3, em que é possível verificar tanto os regressores de entrada, quanto a resposta desejada pela RNA em cada iteração. Quando o processo estiver finalizado, o mesmo é repetido para a curva

Figura 28 – Exemplo de curva precursora e curva a ser prevista (aproximada) pelas RNAs. Por simplicidade, a maioria das amostras que compõem cada curva foram omitidas.

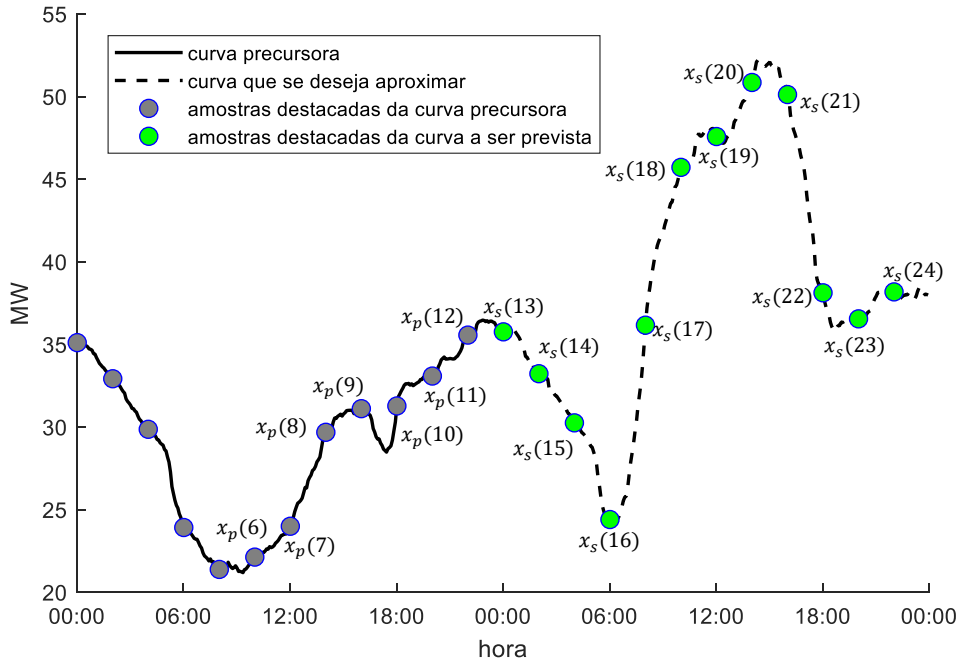


Tabela 3 – Relação de Entradas e Saídas Desejadas durante treinamento, considerando-se uma curva do agrupamento.

Instante	Regressor de Entrada	Saída Desejada
1	$x(12), x(11), x(10), x(9), x(8), x(7), x(6)$	$x(13)$
2	$x(13), x(12), x(11), x(10), x(9), x(8), x(7)$	$x(14)$
3	$x(14), x(13), x(12), x(11), x(10), x(9), x(8)$	$x(15)$
4	$x(15), x(14), x(13), x(12), x(11), x(10), x(9)$	$x(16)$
5	$x(16), x(15), x(14), x(13), x(12), x(11), x(10)$	$x(17)$
6	$x(17), x(16), x(15), x(14), x(13), x(12), x(11)$	$x(18)$
7	$x(18), x(17), x(16), x(15), x(14), x(13), x(12)$	$x(19)$
8	$x(19), x(18), x(17), x(16), x(15), x(14), x(13)$	$x(20)$
9	$x(20), x(19), x(18), x(17), x(16), x(15), x(14)$	$x(21)$
10	$x(21), x(20), x(19), x(18), x(17), x(16), x(15)$	$x(22)$
11	$x(22), x(21), x(20), x(19), x(18), x(17), x(16)$	$x(23)$
12	$x(23), x(22), x(21), x(20), x(19), x(18), x(17)$	$x(24)$

seguinte contida no mesmo grupo, até que todas as curvas tenham sido apresentadas à RNA da mesma maneira.

Considerando-se que na situação real uma única curva é composta por 288 amostras, são gerados 288 pares de entrada-saída. O procedimento descrito no parágrafo anterior e na Tabela 3 é repetido para todas as outras curvas contidas no mesmo agrupamento. Após todas as curvas serem submetidas, obtém-se então uma época de treinamento.

Destaca-se que em nenhum momento as amostras da curva precursora são os objetivos de saída da RNA. Sendo apenas utilizadas nos seus regressores de entrada como auxílio para a previsão das amostras da curva seguinte (a qual se deseja prever). Por meio do exemplo citado, percebe-se que durante a fase de treinamento a RNA realiza uma previsão de um passo adiante, pois o objetivo neste caso é fazer a RNA aproximar o valor de uma amostra por vez, sendo que os valores que incorporam os regressores de entrada são sempre os valores reais observados das séries temporais, e não o valor fornecido pela saída da RNA. Na prática, este procedimento permite que os erros de previsão não sejam realimentados para a camada de entrada, fazendo com os pesos sinápticos da RNA sejam ajustados de acordo apenas com os valores desejados, sem a existência de erros de previsão.

Embora este procedimento seja utilizado durante o treinamento das RNAs, o mesmo não acontece na etapa de testes, pois durante esta etapa a previsão acontece de maneira recursiva, ou seja as próprias amostras previstas pelas RNAs são realimentadas para o regressor de entrada, fazendo com que exista um erro de previsão (mesmo que pequeno em alguns casos), que, por sua vez, passa a exercer influência nas outras saídas produzidas pela RNA. Ressalta-se, ainda, que em

Tabela 4 – Relação de Entradas e Saídas previstas durante fase de teste de uma RNA. Nota-se que, na primeira iteração, apenas as amostras referentes à curva precursora são fornecidas à RNA.

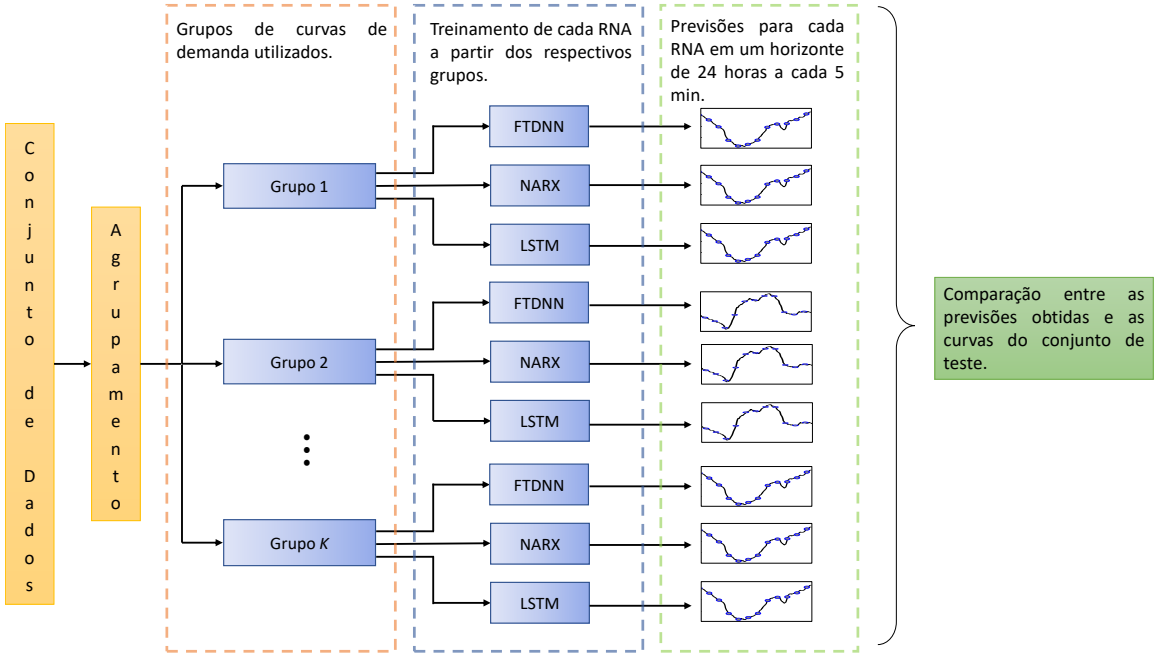
Instante	Regressor de Entrada	Valor Previsto
1	$x(12), x(11), x(10), x(9), x(8), x(7), x(6)$	$\hat{x}(13)$
2	$\hat{x}(13), x(12), x(11), x(10), x(9), x(8), x(7)$	$\hat{x}(14)$
3	$\hat{x}(14), \hat{x}(13), x(12), x(11), x(10), x(9), x(8)$	$\hat{x}(15)$
4	$\hat{x}(15), \hat{x}(14), \hat{x}(13), x(12), x(11), x(10), x(9)$	$\hat{x}(16)$
5	$\hat{x}(16), \hat{x}(15), \hat{x}(14), \hat{x}(13), x(12), x(11), x(10)$	$\hat{x}(17)$
6	$\hat{x}(17), \hat{x}(16), \hat{x}(15), \hat{x}(14), \hat{x}(13), x(12), x(11)$	$\hat{x}(18)$
7	$\hat{x}(18), \hat{x}(17), \hat{x}(16), \hat{x}(15), \hat{x}(14), \hat{x}(13), x(12)$	$\hat{x}(19)$
8	$\hat{x}(19), \hat{x}(18), \hat{x}(17), \hat{x}(16), \hat{x}(15), \hat{x}(14), \hat{x}(13)$	$\hat{x}(20)$
9	$\hat{x}(20), \hat{x}(19), \hat{x}(18), \hat{x}(17), \hat{x}(16), \hat{x}(15), \hat{x}(14)$	$\hat{x}(21)$
10	$\hat{x}(21), \hat{x}(20), \hat{x}(19), \hat{x}(18), \hat{x}(17), \hat{x}(16), \hat{x}(15)$	$\hat{x}(22)$
11	$\hat{x}(22), \hat{x}(21), \hat{x}(20), \hat{x}(19), \hat{x}(18), \hat{x}(17), \hat{x}(16)$	$\hat{x}(23)$
12	$\hat{x}(23), \hat{x}(22), \hat{x}(21), \hat{x}(20), \hat{x}(19), \hat{x}(18), \hat{x}(17)$	$\hat{x}(24)$

um determinado teste qualquer realizado nas redes neurais, apenas as amostras associadas à curva precursora são fornecidas inicialmente à RNA, portanto todas as amostras que correspondem ao dia a ser previsto são fornecidos iterativamente pelas saídas da RNA, ou seja, não há, em nenhum momento o fornecimento dos valores reais das amostras referentes à curva a ser prevista para a camada de entrada da RNA. A Tabela 4 mostra como seria uma etapa de teste realizada, em que o termo $\hat{x}$ é referente ao valor aproximado fornecido pela saída da rede neural. Nota-se que na primeira iteração, apenas as amostras da curva precursora estão presentes no regressor de entrada e à medida que as saídas são produzidas pela RNA, são incorporadas ao regressor de entrada na iteração seguinte.

Outro ponto importante, diferentemente das redes FTDNN e LSTM, a rede NARX utiliza dois regressores em sua camada de entrada, como discutido anteriormente no Capítulo 5. Entretanto as amostras contidas em um dos regressores também estão contidas no outro, mas espaçadas em um intervalo temporal maior. É importante mencionar novamente tal aspecto, pois da forma como foi exemplificado nas Tabelas 3 e 4, o leitor pode ter a impressão que todas as RNAs utilizaram apenas um regressor, o que não é verdade para o caso da rede NARX. Ainda assim, os processos de treinamento e teste exemplificados anteriormente são os mesmos utilizados em todas as RNAs utilizadas.

Os grupos fornecidos pelos algoritmos de agrupamento e combinados por função consenso com base em votação são um ponto preponderante durante o processo de treinamento de todas as RNAs consideradas nesta dissertação. Pois cada curva de teste é treinada com base

Figura 29 – Diagrama ilustrando a metodologia de previsão.



Fonte – O autor.

em grupos específicos. Além disso, foi feito um estudo estatístico que indica a disposição das curvas em relação aos grupos formados, tomando como base aspectos como dia, mês e ano em que as curvas foram geradas. Tal estudo gera informações importantes sobre a escolha de qual grupo deve ser utilizado na etapa de treinamento das RNAs, ao indicar a tendência que as curvas de carga possuem em ocupar determinados grupos a partir da época do ano que foram geradas. Na prática, esta informação torna possível a escolha dos dados de treinamento mais adequados e que tenham maior correlação com os dados a serem previstos, a partir da hipótese de que estes últimos teriam uma grande possibilidade de possuírem as mesmas características dos dados de treinamento. O processo de previsão é ilustrado na Figura 29. Percebe-se que cada grupo gera um conjunto de treinamento diferente que, por sua vez, são submetidos à etapa de treinamento de cada uma das RNAs.

Com a finalidade de verificar até que ponto os grupos obtidos poderiam influenciar positivamente o desempenho das RNAs, as curvas escolhidas para teste (previsão) são removidas de seus respectivos grupos. E para cada curva de teste a ser prevista, utiliza-se como conjunto de treinamento das RNAs as curvas remanescentes pertencentes ao mesmo grupo em que se encontrava a curva de teste. A eficácia do modelo e das respectivas RNAs é verificada por meio de índices de desempenho. Por meio da validação desta hipótese seria possível a criação de métodos que tornassem possível a escolha adequada dos grupos existentes para a etapa de

treinamento, tomando como base as características do dia a ser previsto, e as características dos grupos obtidos.

Dentre as possíveis maneiras de escolha dos grupos adequados para treinamento é possível citar algumas. A primeira delas seria por meio da probabilidade de uma das curvas gerada em um determinado dia estar presente em um ou mais grupos. Por exemplo: a probabilidade em que uma curva gerada em uma quarta-feira do mês de outubro estar presente em determinado(s) grupo(s). Feita esta verificação a previsão seria uma média ponderada das previsões geradas por uma RNA quando treinada com o conjunto de dados oriundo destes agrupamentos por vez.

Uma segunda maneira seria treinar a RNA com um conjunto de treinamento formado por todos os grupos em que o dia ser previsto possui probabilidade de estar inserido. Semelhante ao que foi descrito no parágrafo anterior, mas em vez de se considerar apenas um grupo para treinamento por vez, seriam considerados todos os possíveis grupos em que a curva a ser prevista poderia estar inserida, ou seja, a união de dois ou mais grupos para se formar o conjunto de treinamento.

Uma outra alternativa sugerida seria a utilização de um sistema especialista alimentado por características diversas que tornariam possível a escolha do melhor agrupamento a ser utilizado no treinamento da RNA. Algumas destas características poderiam ser a temperatura do dia que precede e a previsão de temperatura ambiente do dia a ser previsto, assim como precipitação e umidade relativa do ar, a existência de feriado ou véspera de feriado, a época do ano em ocorrerá a previsão (mês, estação do ano, dia da semana), verificação se a curva de demanda do dia anterior consistiu em padrões anômalos, etc.

As três formas de escolha até aqui elencadas não fazem parte do escopo deste trabalho. Ao contrário disso, busca-se mostrar que grupos de treinamento com padrões semelhantes à curva que se deseja prever fornecem subsídios para uma melhor previsão, pois como os métodos de previsão utilizados baseiam-se em técnicas de inteligência computacional, a escolha de dados ruins ou que não condizem com as informações a serem previstas poderiam gerar modelos imprecisos ou até mesmo incapazes de realizar uma previsão satisfatória. Por outro lado, dados de boa qualidade podem levar à criação de modelos com bons índices de previsão. E como explicado anteriormente, para verificar a existência de tal potencial os dados utilizados na etapa de treinamento são escolhidos levando-se em consideração o grupo a qual pertence a curva utilizada na etapa de testes. Para isto, os dados testados são retirados de seus respectivos grupos e utilizados apenas para verificar o quão boa foi a previsão dos modelos utilizados. Os treinamentos

das RNAs, foram realizados com os dados remanescentes destes grupos, sem a existência dos dados utilizados para a etapa de testes da RNA.

A metodologia utilizada permite verificar se estes agrupamentos são realmente capazes de fornecer meios para uma boa previsão das RNAs, além de ser possível verificar a eficiência das redes neurais dinâmicas empregadas, comparando os resultados de previsão obtidos entre si e, também, comparando-os com um método de previsão tradicional de boa eficiência: a média simples.

Ainda em relação às RNAs, é importante destacar alguns aspectos devido às dificuldades de se encontrar um modelo adequado, capaz de se adequar a um determinado sistema sob estudo. Devido a isso, é comum despender bastante tempo até que uma arquitetura apropriada de RNA seja encontrada. Além disso, na prática, diversos testes e ajustes devem ser realizados para que se possa definir parâmetros como, por exemplo, o número de camadas escondidas, número de neurônios, algoritmo de treinamento, dimensão dos vetores de entrada, etc.

Diante da imensa quantidade de combinações de ajustes destes parâmetros, é recomendável que se dê preferência ao modelo mais simples capaz de fornecer bons resultados, sobretudo em relação à quantidade de camadas e neurônios da RNA. Em outras palavras, quando têm-se dois modelos que se ajustam igualmente a uma determinada sequência de dados, seleciona-se o que é o "mais simples" no sentido de permitir o uso de uma descrição mais sucinta dos dados, permitindo o menor uso de recursos computacionais. Este método também é conhecido como princípio da navalha de Occam (*Occam's Razor*) (HAYKIN, 2009).

Devido a isso, muitos dos testes realizados com as RNAs utilizadas neste trabalho foram feitos com a finalidade de se encontrar uma certa quantidade de camadas e neurônios capazes de fornecer bons resultados, assim, também, como o tamanho dos regressores de entrada adotados. Embora não seja possível afirmar, veementemente, que os parâmetros utilizados nesta dissertação sejam os melhores possíveis, as análises realizadas mostraram que os valores encontrados para os mesmos foram capazes de fornecer bons resultados, em meio a alguns intervalos de valores previamente definidos.

Observou-se, por exemplo, que as redes NARX e FTDNN foram capazes de fornecer melhores resultados com o uso de duas camadas escondidas, ao passo que para a rede LSTM foi necessário o uso de apenas uma, sendo que o acréscimo de outra camada não demonstrou melhora de desempenho. Devido às características compartilhadas pelas redes NARX e FTDNN, optou-se por utilizar a mesma quantidade de neurônios em suas camadas ocultas, sendo 80

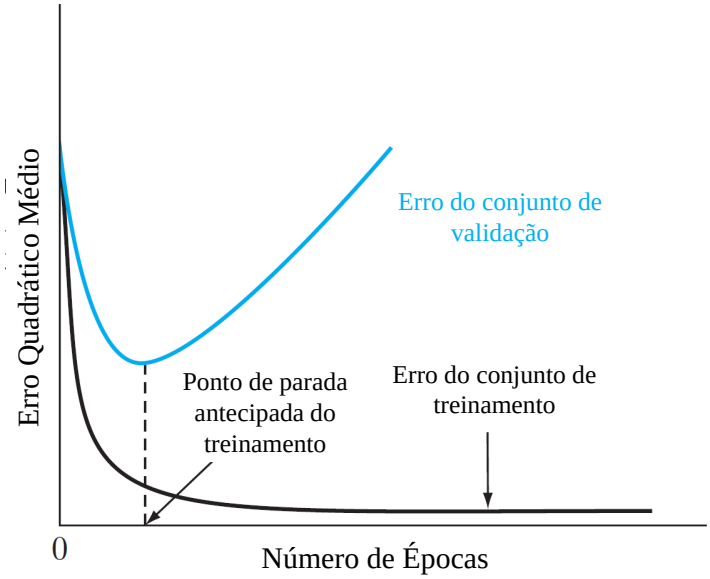
e 9, respectivamente, em meio a outros valores analisados, uma quantidade que se mostrou capaz de fornecer bons resultados em ambas. Foi verificado que o aumento da quantidade de neurônios não possibilitou melhorias nas previsões, ao passo que proporcionava aumento do custo computacional durante o treinamento das RNAs. Por outro lado, verificou-se que 200 neurônios na única camada oculta da rede LSTM se mostrou suficiente. Embora testes com quantidade reduzida de neurônios distribuídos em duas camadas ocultas tenham sido realizados, como nos casos das redes NARX e FTDNN, constatou-se que melhores performances eram obtidas com o uso de uma única camada escondida e maior quantidade de neurônios. Resultados, estes, que podem estar atrelados às peculiaridades existentes na rede LSTM e brevemente discutidas no Capítulo 5.

Em relação aos regressores, foi utilizada metodologia semelhante, analisando-se, primeiramente, regressores com pequenas quantidades de amostras das séries temporais. Todavia, foi constatado que todas as RNAs demonstraram-se incapazes de emular a dinâmica das séries temporais observadas de maneira satisfatória para o horizonte de previsão de 288 passos, fornecendo uma boa previsão para os primeiros instantes de tempo, mas com erros demasiadamente elevados para os instantes de tempo posteriores. À medida em que a dimensão dos regressores era aumentada, foi possível perceber melhorias gradativas nos seus desempenhos. Em geral, pode-se afirmar que regressores contendo acima de 150 amostras eram capazes de melhorar substancialmente os resultados obtidos pelas RNAs.

Ressalta-se que todos os valores discutidos até o momento podem ser considerados subótimos, uma vez que foram encontrados por meio de testes experimentais. Uma análise mais profunda e exata demandaria uma busca exaustiva por estes parâmetros, sendo, portanto, uma perspectiva que não pertence ao propósito deste trabalho. Entretanto, verificou-se que dentre diversas configurações analisadas, as utilizadas nesta dissertação foram as que apresentaram melhores desempenho.

Durante o treinamento, um certo conjunto de dados é apresentado às RNAs em forma de vetores de entrada e saída. Uma época de treinamento é definida como a apresentação de todos esses pares à uma RNA. A escolha da quantidade adequada de épocas de treinamento é um aspecto importante, visto que é durante este processo que ocorre o ajuste dos pesos sinápticos da rede. Para o treinamento das redes NARX e FTDNN, considerou-se 5000 como a quantidade máxima de épocas permitidas. Ao alcançar este valor, o treinamento das redes estaria finalizado. Não obstante, para evitar que estas redes estivessem submetidas a ajuste excessivo dos pesos,

Figura 30 – Ilustração do erro quadrático médio em função do número de épocas para os conjuntos de treinamento e validação.



Fonte – Adaptada de Haykin (2009).

optou-se por utilizar um critério de parada antecipada do treinamento. Para isto, um conjunto aleatório de pares entrada-saída era escolhido e extraído do conjunto de treinamento. À medida que as épocas eram processadas durante o treinamento, o erro médio quadrático das RNAs era calculado tanto para o conjunto de treinamento, quanto para o de validação (HAYKIN, 2009).

A Figura 30 ilustra conceitualmente as curvas de aprendizado de ambos os casos. Verifica-se que a partir de certo ponto, o erro médio quadrático do conjunto de validação começa a aumentar, enquanto o do conjunto de treinamento continua a diminuir. Embora isto ocorra, pode-se considerar que a partir deste ponto a rede começa a incorporar em seu aprendizado apenas o ruído contido nos dados de treinamento, sendo, portanto, possível antecipar a parada do processo de treinamento das RNAs na época em que ocorre o menor erro quadrático de validação (HAYKIN, 2009). Na prática, o que ocorre é que apesar de se definir uma quantidade máxima de épocas, as redes nunca chegaram a atingi-la devido ao procedimento de parada antecipada. Em virtude disso, em média, o treinamento de ambas as redes era finalizado entre 3000 e 3500 épocas. Em relação a este aspecto, a rede LSTM se mostrou mais eficiente, sendo necessária a definição de apenas 150 épocas para que a mesma pudesse finalizar seu treinamento, com taxa de aprendizagem inicial igual a 0,001. Embora outros valores tenham sido verificados, este último se mostrou capaz de fornecer os melhores resultados.

Antes de efetuar os treinamentos das RNAs, foi necessário realizar a normalização dos valores de entrada. Sendo assim, todas as amostras do conjunto de treinamento e teste foram

escalonadas no intervalo [-1, 1], levando-se em consideração a seguinte equação:

$$x'_n = 2 \cdot \left(\frac{x_n - x_{min}}{x_{max} - x_{min}} \right) - 1. \tag{6.3}$$

Tal estratégia é fundamentada no princípio dos segmentos proporcionais e leva em consideração a faixa de variação das funções de ativação das RNAs (SILVA; SPATTI; FLAUZINO, 2010). Tal estratégia de normalização foi utilizada nas redes FTDNN e NARX que por sua vez utilizam função de ativação tangente hiperbólica na camada oculta e função linear na saída. Entretanto, o mesmo esquema de normalização não se mostrou eficiente quando utilizado na rede LSTM. Uma estratégia eficiente para a normalização de dados para RNAs como a LSTM e redes convolucionais (*Convolution Neural Networks* - CNNs), é normalizar os dados para que tenham média zero e variância unitária, de acordo com a Equação (6.4), em que μ é a média dos dados e σ é o desvio padrão (GOODFELLOW; BENGIO; COURVILLE, 2016; IOFFE; SZEGEDY, 2015).

$$x'_n = \frac{x_n - \mu}{\sigma} \tag{6.4}$$

Deve-se ressaltar que devido ao escalonamento realizado por meio das Equações (6.3) e (6.4), as saídas produzidas pelas redes neurais também estará no intervalo de valores definido pela normalização. Portanto, é necessário que seja feito o redimensionamento dos valores preditos para a faixa de valores original, para que, posteriormente, a saída das redes possa ser avaliada por meio dos índices de desempenho.

Para que fosse possível analisar o desempenho das RNAs quando submetidas ao conjunto de teste, (ou seja, dados diferentes daqueles apresentados durante o treinamento) as redes são treinadas e testadas diversas vezes. Em cada treinamento as redes são iniciadas com pesos sinápticos aleatórios, ao final do treino as redes são, então testadas. Ao final de cada ciclo de treinamento\teste foram utilizados índices de desempenho para verificar a precisão dos valores estimados pelas RNAs na etapa de teste. Considerando-se todos os ciclos para uma determinada série-temporal de teste, foi possível obter-se cálculos estatísticos da precisão das predições realizadas pelas redes neurais. Na próxima seção, são discutidos os índices de desempenho utilizados nesta dissertação.

6.6 Índices de Desempenho

A qualidade da previsão obtida é um dos principais requisitos ao se comparar a eficiência de algoritmos e modelos empregados na tarefa de previsão. Ambos os índices de desempenho utilizados nesta dissertação utilizam como base o resíduo ou erro de previsão que consiste na diferença entre o valor real observado $x(n)$ e o valor estimado $\hat{x}(n)$,

$$e(n) = x(n) - \hat{x}(n). \tag{6.5}$$

O resíduo é calculado em cada uma das N amostras que compõem a série temporal. O erro médio de previsão ou erro quadrático médio (*Mean Squared-Error* - MSE) é definido por

$$\varepsilon^2 = \frac{\sum_{n=1}^{H}(x(n) - \hat{x}(n))^2}{H}, \tag{6.6}$$

em que H é o tamanho da sequencia a ser prevista. Quanto maior o valor de ε^2, menor é a qualidade da previsão e vice-versa.

Embora a Equação (6.6) forneça uma medida de qualidade da previsão, a mesma não fornece um indicativo do quão elevado é o erro de predição, pois seu resultado é um número absoluto. Portanto, para decidir se o erro de previsão fornecido por um modelo é alto ou não, é necessário compará-lo com algum valor de referência, tornando possível que modelos diferentes possam ser comparados entre si . Por conseguinte, o erro médio quadrático normalizado (NMSE) é uma alternativa que permite verificar a eficiência de um modelo, tendo como base um valor de referência, definido pela Equação (6.7) (MENEZES JÚNIOR, 2012).

$$NMSE(H) = \frac{\sum_{n=1}^{H}(x(n) - \hat{x}(n))^2}{\sum_{n=1}^{H}(x(n) - \overline{x}(n))^2} \tag{6.7}$$

A Equação (6.7) pode ser deduzida a partir das seguintes expressões:

$$\varepsilon_N^2 = \frac{\varepsilon^2}{\hat{\sigma}_x^2}, \tag{6.8}$$

em que $\hat{\sigma}_x^2$ representa a variabilidade amostral dos dados em relação à média,

$$\hat{\sigma}_x^2 = \frac{\sum_{n=1}^{H}(x(n) - \overline{x})^2}{H}, \tag{6.9}$$

e $\overline{x}$ é a média da série temporal. Substituindo-se as Equações (6.6) e (6.9) na Equação (6.8), obtém-se (6.7). Como explica Menezes Júnior (2012), a Equação (6.7) é equivalente a comparar o erro de previsão de um determinado modelo, com o erro fornecido pelo modelo mais trivial, o

de média móvel. Ou seja, valores próximos a zero indicam uma previsão considerada boa. Por outro lado, valores próximos a 1 mostram que o resultado é tão ruim quanto o fornecido pelo modelo que gera previsões pelo valor médio.

Um outro indicador comumente utilizado como métrica em problemas de previsão de séries temporais é o erro percentual médio absoluto (*Mean Absolute Percentage Error* - MAPE), que representa a média percentual da divisão entre o erro dos dados previstos e os dados reais. Na prática, o MAPE é frequentemente utilizado devido a sua fácil interpretação ao ser simplesmente expresso em termos percentuais. Formalmente, o MAPE pode ser definido como a média da diferença absoluta entre o valor real e o valor estimado, expresso como um percentual do valor observado (HEIZER; RENDER, 2008), de acordo com a Equação (6.10).

$$MAPE = \frac{1}{H}\sum_{n=1}^{H}\frac{|x(n)-\hat{x}(n)|}{x(n)} \times 100 \tag{6.10}$$

Uma das limitações em se utilizar somente o MSE como métrica de avaliação, é sua sensibilidade a largos desvios, devido o termo quadrático na Equação (6.6). Fazendo com que em certas situações seja preferível vários pequenos desvios, do que a presença de um único desvio de alta amplitude (HEIZER; RENDER, 2008). Devido à inexistência do termo quadrático na Equação (6.10), o MAPE é menos sensível a este problema. Entretanto, devido à esta característica, deve-se apontar para o fato de que, por meio do NMSE, é possível apontar diferenças mais marcantes entre dois ou mais modelos analisados, diferente do que ocorre no MAPE. Ambas as métricas são utilizadas na análise dos modelos utilizados nesta dissertação.

Como as redes neurais são submetidas a ciclos de treinamento/teste, com a finalidade de se observar a precisão estatística dos modelos, são extraídos valores referentes às medianas, máximos, mínimos e desvios padrões dos índices obtidos. Além disso, o uso da mediana fornece uma boa forma de análise, pois se trata de uma medida estatística que possui menos sensibilidade a *outliers* (HUBER, 2011; MAGALHÃES; LIMA, 2002).

7 RESULTADOS

7.1 Introdução

Nos capítulos anteriores foram discutidas todas as ferramentas utilizadas durante o desenvolvimento desta dissertação, desde a etapa de pré-processamento dos dados, alguns dos principais aspectos teóricos sobre os algoritmos de agrupamento e redes neurais utilizadas, até a metodologia de projeto adotada durante este estudo.

Este capítulo tem como finalidade mostrar e analisar os principais resultados obtidos por meio da combinação entre algoritmos de agrupamento e redes neurais para a tarefa de previsão de curvas de carga. Primeiramente, são abordados os resultados oriundos da metodologia de agrupamento utilizada, que consiste na combinação de agrupamentos por meio de função consenso por sistema de votação. Em seguida, são analisados os resultados das previsões de múltiplos passos adiante, obtidos para os três modelos de RNAs dinâmicas empregadas.

O desempenho destas RNAs é analisado principalmente sob a ótica dos índices apontados na Seção 6.6, observando-se aspectos como capacidade de generalização do modelo, precisão e sensibilidade das redes à mudança da dimensão dos seus regressores. As curvas utilizadas como teste para as RNAs são referentes ao mês de novembro de 2017. Desta forma, além dos aspectos citados anteriormente, é observado a capacidade das redes em captar o comportamento dinâmico destas curvas, levando-se em consideração as peculiaridades atreladas aos dias e horários em que ocorrem o consumo de energia elétrica.

Convém ressaltar que a metodologia de treinamento utilizada pelas RNAs é fundamentada nos grupos obtidos previamente. Resultando em uma abordagem híbrida que mescla paradigmas de aprendizagem não-supervisionada (algoritmos de agrupamento) e supervisionada (redes neurais dinâmicas).

Embora não tenha sido feita uma busca exaustiva para se encontrar os parâmetros que maximizassem a capacidade preditiva das RNAs sob análise, é mostrada as potencialidades do uso conjunto de ferramentas de agrupamento e RNAs dinâmicas, possibilitando a previsão recursiva de demanda energia elétrica em um horizonte de 288 passos à frente, considerando-se apenas a série-temporal univariada, sem a existências de qualquer outra variável que possa explicar a dinâmica do sistema.

7.2 Resultados dos Algoritmos de Agrupamento

Após todo o processo de agrupamento descrito na Seção 6.4, foi feita uma análise da distribuição das curvas de carga em relação aos grupos em que foram alocadas, aos meses do ano e dias em que foram geradas. Apenas para exemplificar como ficou estabelecida a distribuição das séries temporais, as Tabelas 5, 6 e 7 mostram o percentual de distribuição obtido para todos os meses de janeiro, outubro e novembro, respectivamente, considerando-se todos os anos em que foram feitas as medições. Sendo assim, as três tabelas mostram a tendência que as curvas possuem em ocupar certos grupos, dependendo do mês em que foram geradas e independentemente do ano. Isto se deve ao fato de haver padrões de consumo intrínsecos a determinados períodos do ano.

Tabela 5 – Distribuição (%) das curvas de carga em relação aos meses de janeiro.

Grupo	Domingo	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
1	0	40	12,50	16,67	0	20	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	10	30
5	0	30	12,50	41,67	70	40	0
6	0	0	0	0	0	0	0
7	66,67	0	0	8,33	0	0	0
8	0	0	0	0	0	0	0
9	33,33	10	12,50	0	10	0	20
10	0	0	0	0	0	0	0
11	0	10	62,50	33,33	10	20	0
12	0	10	0	0	10	10	0
13	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0
15	0	0	0	0	0	0	50
16	0	0	0	0	0	0	0

Comparando-se as Tabelas 5 e 6 percebe-se que a tendência de ocupação dos grupos pelas curvas em relação aos dias úteis é bastante diferente. Enquanto, no primeiro caso, estes dias ocupam majoritariamente os Grupos 1, 5, 9 e 11, no segundo caso percebe-se que as curvas ocupam de maneira bem concentrada os Grupos 3, 8 e 12. Isto se deve, provavelmente, ao fatores relativos a condições climáticas. Na cidade Teresina, os meses de janeiro registram temperaturas mais amenas que o mês de outubro, fazendo com que os padrões de consumo possam ser diferentes nestas épocas do ano, refletindo na maneira como as curvas são agrupadas em determinados grupos.

É possível perceber, por exemplo, na Tabela 5 que 70% das curvas geradas nas quintas-feiras dos meses de janeiro, possuem a tendência de pertencerem ao Grupo 5, independentemente do ano em que foram geradas. Enquanto no mês de outubro, não existe nenhuma curva gerada em quintas-feiras pertencente a este grupo. Esta é uma característica interessante, visto que demonstra certas diferenças e peculiaridades em comum que curvas geradas em uma determinadas épocas do ano podem possuir. Ressalta-se, novamente, que os resultados contidos nas Tabelas 5 a 7, servem para exemplificar esta distribuição, sendo que analises semelhantes puderam ser feitas para os demais meses. Além disso, estes resultados contemplam os meses, independentemente do ano. Ou seja, são considerados todos os meses de janeiro, fevereiro, etc. contidos em todo o conjunto de dados, portanto não são referentes a um ano em específico.

Tabela 6 – Distribuição (%) das curvas de carga em relação aos meses de outubro.

Grupo	Domingo	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	12,50	37,50	30,77	45,45	33,33	0
4	0	0	0	0	0	0	0
5	0	6,25	0	0	0	0	0
6	0	0	0	0	0	0	0
7	61,54	0	0	0	9,09	0	0
8	0	43,75	43,75	46,15	36,36	41,67	0
9	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0
12	0	18,75	12,50	7,69	0	16,67	0
13	0	6,25	6,25	0	9,09	0	0
14	0	12,50	0	0	0	0	54,55
15	0	0	0	0	0	0	27,27
16	38,46	0	0	15,38	0	8,33	18,18

Para este estudo, optou-se em dividir todo o conjunto de dados em 16 grupos. É importante salientar que esta opção não reflete uma quantidade de grupos absoluta. Na realidade, o número de grupos em que deve ser feito o agrupamento depende de alguns fatores como tamanho e complexidade do conjunto de dados. Sendo, portanto, um parâmetro que deve ser ajustado de acordo com cada caso, baseando-se em testes em experimentações que possibilitem uma escolha adequada. Neste trabalho, foi verificado que 16 grupos forneceu agrupamentos com boa consistência, levando-se em consideração grupos com quantidade suficiente de instâncias e relativamente uniformes. Sendo assim, bons resultados também poderiam ser obtidos utilizando-

se uma quantidade de grupos ligeiramente diferente.

Outras característica que é possível perceber ao se analisar as três tabelas, é a segregação existente entre grupos compostos por curvas geradas em dias úteis e aqueles cuja composição consiste majoritariamente em curvas geradas durante os finais de semana. Em todos os casos colocados como exemplo, o Grupo 7 é o que possui maior quantidade de curvas geradas em domingos. Uma característica interessante, é que para o mês de janeiro, o restante das curvas de domingo está alocado no Grupo 9, enquanto para os meses de outubro e novembro, o restante está alocado no Grupo 16. Isto indica que, em alguns casos, o consumo de energia elétrica relativo ao início do ano, durante os dias de domingo, é diferente em relação aos meses mais próximos ao final do ano. Em relação aos sábados, observa-se que nos meses de janeiro e novembro, a maioria das curvas está alocada no Grupo 15, enquanto no mês de outubro, a maioria está designada ao Grupo 14. Ainda assim, é possível perceber que a tendência de ocupação das curvas geradas aos sábados é mais semelhante entre os meses de novembro e outubro, pois em ambos, estes dias possuem maior probabilidade de estarem alocados nos Grupos 14, 15 e 16. Por outro lado, nos meses de janeiro, as curvas estão distribuídas entre os Grupos 4, 9 e 15.

Tabela 7 – Distribuição (%) das curvas de carga em relação aos meses de novembro.

Grupo	Domingo	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
1	0	0	0	0	0	0	0
2	0	9,09	0	0	0	0	0
3	0	72,73	33,33	33,33	38,46	50,00	0
4	0	0	0	0	0	0	0
5	0	0	11,11	25,00	7,69	6,25	0
6	0	0	0	0	0	0	0
7	54,55	0	0	0	7,69	6,25	8,33
8	0	9,09	33,33	33,33	30,77	12,50	0
9	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0
12	0	9,09	22,22	8,33	15,38	18,75	0
13	0	0	0	0	0	0	0
14	0	0	0	0	0	0	33,33
15	0	0	0	0	0	0	50,00
16	45,45	0	0	0	0	6,25	8,33

É importante enfatizar, que poderia se pensar, a princípio, que os dados referentes aos meses de outubro e novembro poderiam apresentar uma grande semelhança ao ponto de boa parte destes dados estarem ocupando os mesmos grupos. De fato isto acontece, mas não

necessariamente em todos os casos. Por exemplo, em ambos os meses, as curvas de dias úteis possuem uma forte tendência de ocupação dos Grupos 3, 8 e 12, assim como ocorre com as curvas de domingo, que ocupam os Grupos 7 e 16 em ambos os meses. Entretanto, existem algumas peculiaridades que devem ser notadas. Enquanto no mês de novembro as curvas oriundas das segundas-feiras estão, em sua maioria (72%), presentes no Grupo 3, no mês de outubro a maioria está contida no Grupo 8 e outra parcela significativa dividida entre os Grupos 3, 12 e 14.

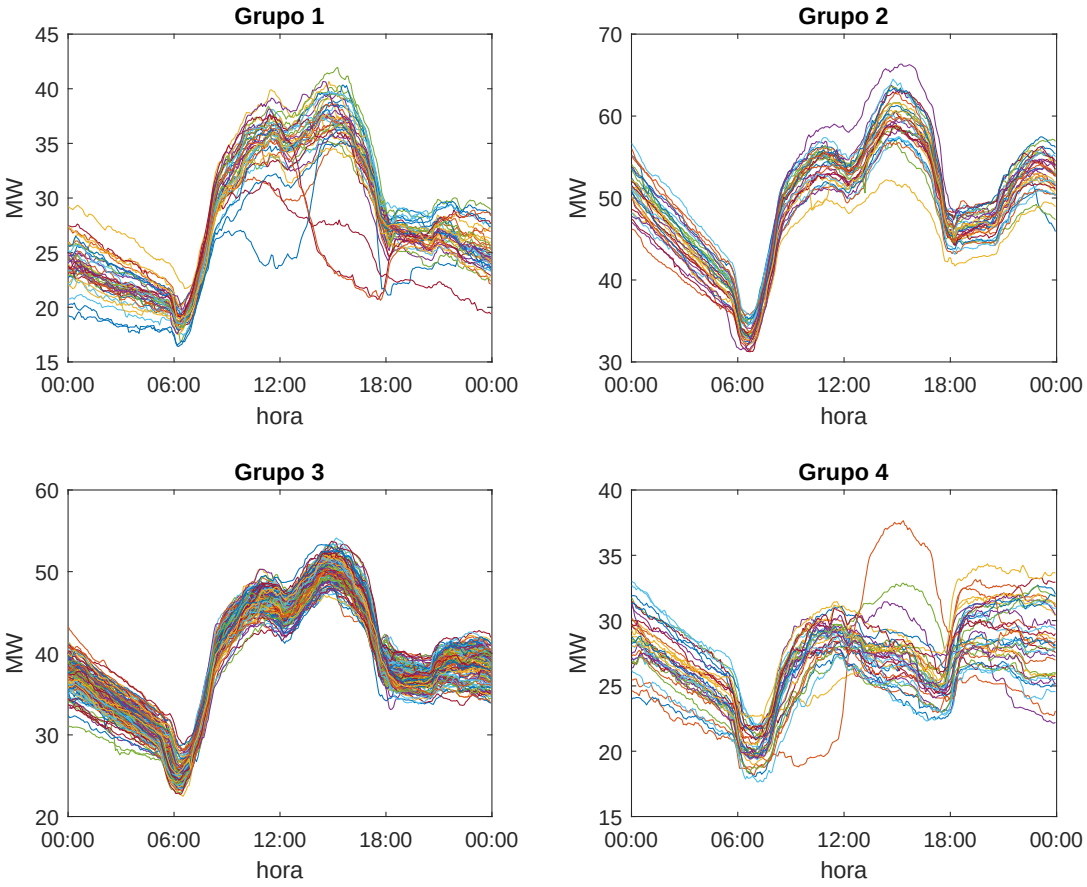
Quando realizada esta mesma análise, comparando o mês de janeiro com os meses de outubro e novembro, fica claro como os dois últimos possuem características bem distintas em relação ao primeiro. Embora haja semelhanças, principalmente em relação aos finais de semana, é possível perceber as grandes diferenças existentes entre a disposição das curvas dos dias úteis, pois, em janeiro, a tendência é que estas curvas ocupem principalmente os Grupos 1, 5, 9 e 11, como foi discutido anteriormente.

A partir destas análises é possível notar uma das vantagens em se utilizar técnicas de agrupamento: a possibilidade de se conhecer as características dos dados em estudo. Outro ponto relevante, o agrupamento permite a obtenção de grupos uniformes formados por séries temporais que possuem dinâmicas semelhantes. A capacidade dos algoritmos em captar tais semelhanças é independente do dia e mês em que as curvas foram geradas. Exatamente devido a este aspecto que curvas referentes a um mesmo dia e mês podem estar distribuídas em grupos diferentes. Ao passo que curvas geradas em dias e meses distintos podem ocupar um mesmo grupo, indicando que, apesar da diferença temporal em que foram geradas, possuem dinâmicas semelhantes. Em outras palavras, o fato das curvas de carga pertencerem a um mesmo mês e aos mesmos dias da semana, não indica, em muitas situações, que estes dados terão dinâmicas semelhantes.

Caso uma estratégia de agrupamento como a que foi feita, não fosse realizada, a escolha do conjunto de dados necessário para realizar o treinamento das RNAs poderia estar comprometida, pois não seria possível saber, a princípio, se estes dados teriam dinâmicas com características em comum, o que poderia aumentar os erros de previsão obtidos. A estratégia de agrupamento é utilizada, principalmente para reduzir esta possibilidade, ao fornecer um meio para reduzir o conjunto de treinamento, reduzir presença de *outliers* e possibilitar a escolha de padrões semelhantes para treinamento, tornando as RNAs aptas a captarem as características mais específicas do conjunto de dados.

Os agrupamentos podem ser visualizados por meio das Figuras 31 a 34. Nota-se que os algoritmos conseguiram extrair bem as características existentes na maioria das curvas,

Figura 31 – Ilustração dos Grupo 1, 2, 3 e 4.



gerando grupos uniformes cujos padrões dinâmicos são semelhantes. Ao observá-los atentamente, é possível identificar algumas curvas que não se enquadram completamente no formato típico apresentado pelo grupo. Isto ocorre sobretudo devido a existência de dias atípicos, tais como feriados, eventos festivos, interrupções do fornecimento de energia elétrica e entre outros fatores. Alguns exemplos podem ser visualizados nos Grupos 1, 4 (Figura 31) e 5 (Figura 32). E nestas situações, os algoritmos distribuem tais curvas nos grupos que supostamente possuem maiores características em comum.

A Tabela 8 exibe o percentual de ocupação de cada grupo por dias úteis e finais de semana, considerando-se todo o conjunto de dados. Percebe-se que os grupos mais heterogêneos são aqueles nos quais a maioria de seus componentes são referentes a curvas de finais de semana. Como explicado no parágrafo anterior, isto se deve principalmente à presença de feriados em dias úteis, fazendo com que curvas geradas de segunda a sexta possam ser destinadas a grupos cujos padrões existentes sejam de curvas de finais de semana.

Figura 32 – Ilustração dos Grupo 5, 7, 8 e 9.

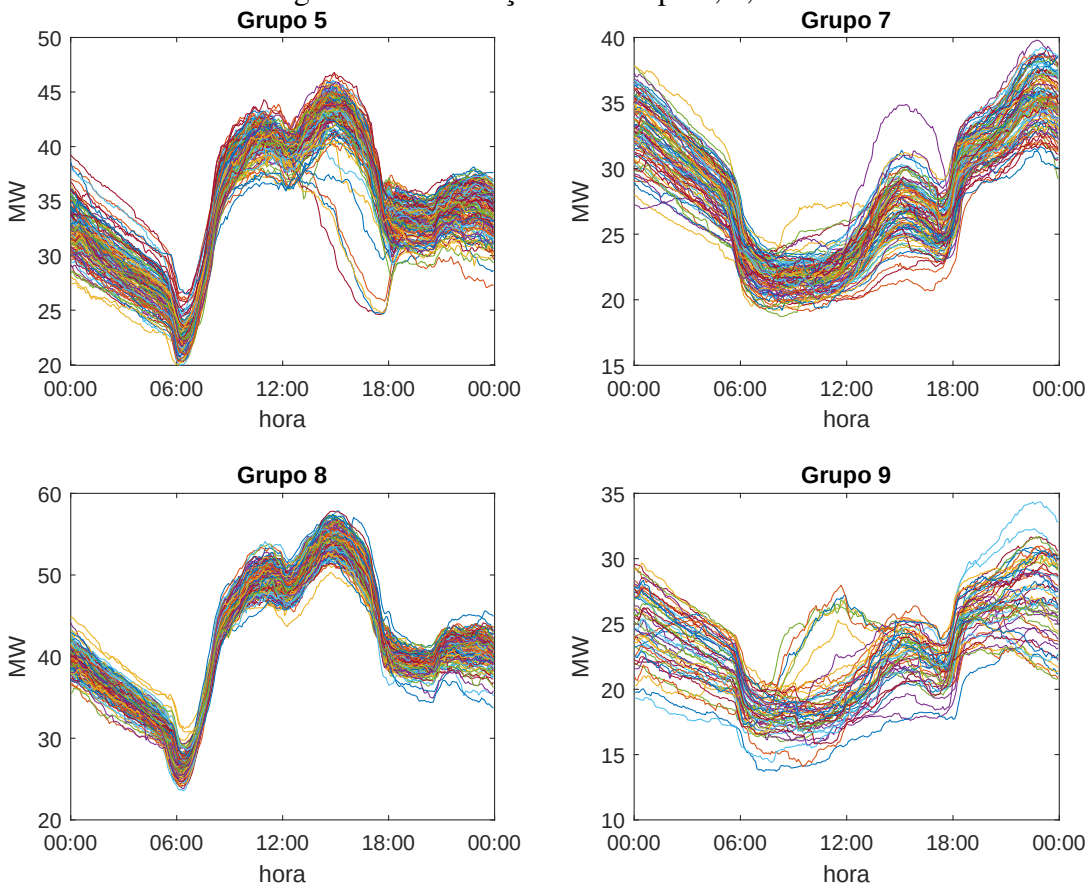
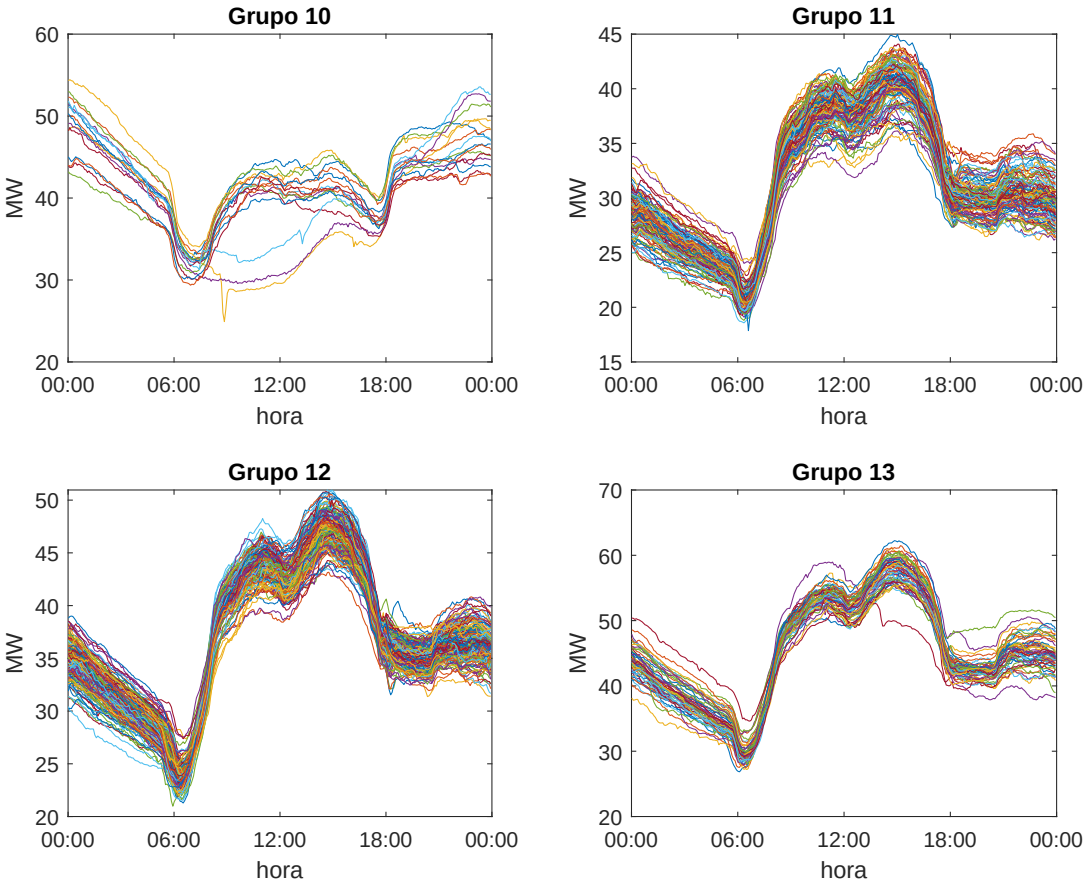


Tabela 8 – Distribuição (%) de dias úteis e finais de semana ao longo dos grupos

Grupo	Dias úteis	Sábados	Domingos
1	100	0	0
2	100	0	0
3	100	0	0
4	8,82	91,18	0
5	100	0	0
6	0	0	0
7	10,99	3,30	85,71
8	100	0	0
9	32	18	50
10	0	81,25	18,75
11	100	0	0
12	100	0	0
13	100	0	0
14	5,71	94,29	0
15	9,59	89,04	1,37
16	22	6	72

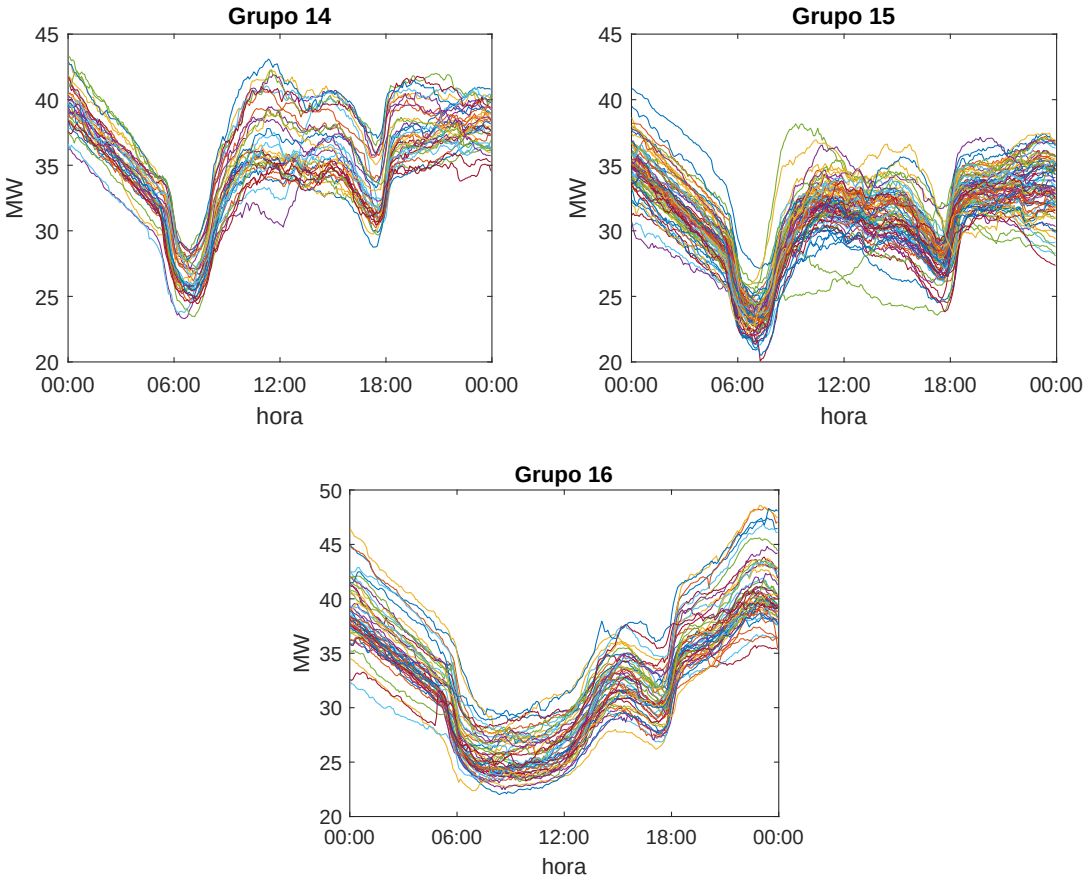
Figura 33 – Ilustração dos Grupo 10, 11, 12 e 13.



Por outro lado, curvas de dias úteis, em que não houve a ocorrência de feriados, possuem dinâmica bem delimitada, fazendo com que grupos compostos por estes dias não possuam curvas de nenhuma outra natureza, como é o caso dos Grupos 1, 2, 3, 5, 8, 11, 12 e 13. Além disso, cabe notar que o Grupo 6 não possui instâncias. Embora as partições geradas pelos algoritmos *minCEntropy* e *k-means* fossem designados a fornecer 16 grupos, a pouca ocorrência de instâncias designadas ao Grupo 6, durante a combinação por votação dos agrupamentos gerados pelas partições base, fez com que as instâncias pertencentes a este grupo passassem a pertencer a outro, devido a existência de voto majoritário entre as partições. Ou seja, algumas instâncias foram tão poucas vezes apontadas como pertencentes ao Grupo 6 e por isso o mesmo acabou sendo sobrepujado por outros grupos durante o processo de votação. Sendo assim, efetivamente, foram obtidos apenas 15 grupos.

Como citado anteriormente, uma análise por meio de agrupamentos permite conhecer as características dos dados sob estudo. Um ponto que chamou atenção foi o motivo das curvas

Figura 34 – Ilustração dos Grupo 14, 15 e 16.



contidas no Grupo 1 possuírem uma certa diferença de padrão durante os horários de maior consumo, quando comparadas aos demais grupos formados por curvas de dias úteis. Verificou-se que este grupo é quase 95% formado somente por curvas geradas durante o ano de 2018. Após uma análise mais minuciosa junto à concessionária de energia elétrica, foi constatado que houve uma mudança no alimentador sob estudo no início do ano de 2018, e o mesmo passou a alimentar novas cargas. Este acontecimento fez com que a dinâmica de consumo fosse alterada, gerando padrões com características diferentes e que foram segregadas em um único grupo.

A Tabela 9 mostra a distribuição das curvas nos grupos em relação ao ano em que foram geradas. Nota-se que, assim como ocorreu no Grupo 1, os Grupos 2 e 10 também separaram do restante dos dados, curvas cujas características foram apresentadas quase que totalmente durante um único ano. No Grupo 2, quase 97% das curvas foram geradas somente em dias úteis no ano de 2015. Enquanto no Grupo 10, as curvas foram geradas durante sábados do ano de 2015. Neste último caso, nota-se a presença de três curvas cujo formato destoa das

demais, sendo as únicas que não foram geradas durante aquele ano. Embora exista a ocorrência deste fato, foi verificado que o ano de 2015 possui distribuição bem homogênea ao longo dos outros grupos, o que pode indicar apenas mudanças no padrão de consumo em determinados dias e meses, fazendo com que as curvas relativas aos mesmos fossem separadas em grupos a parte, como é o caso dos Grupos 2 e 10.

Tabela 9 – Distribuição (%) das curvas de carga em relação aos anos em que foram geradas

Grupo	2012	2013	2014	2015	2016	2017	2018
1	0	0	2,63	2,63	0	0	94,74
2	0	0	0	96,88	0	3,125	0
3	6,99	29,37	19,58	2,80	12,59	28,67	0
4	0	17,65	14,71	8,82	0	14,71	44,12
5	3,62	31,88	36,96	8,70	5,07	7,25	6,52
6							
7	3,30	38,46	26,37	5,49	9,89	16,48	0
8	7,25	17,39	22,46	4,35	22,46	26,09	0
9	0	14	12	14	0	4	56
10	0	0	0	81,25	0	18,75	0
11	0	13,33	12,38	11,43	1,90	7,62	53,33
12	5,30	29,80	25,83	3,31	14,57	21,19	0
13	0	0	1,85	37,04	3,70	57,41	0
14	0	17,14	28,57	5,71	11,43	37,14	0
15	8,22	30,14	24,66	5,48	12,33	19,18	0
16	8	12	14	14	16	36	0

Convém salientar, ainda, que embora alguns grupos possuam propriedades muito semelhantes à de outros, é necessário se atentar a um fator muito importante: a amplitude das séries temporais em cada um dos quatro intervalos de seis horas, como fora descrito na Seção 6.4. Como exemplo, pode-se comparar os Grupos 14 e 15 na Figura 34. Mesmo que ambos sejam predominantemente compostos por curvas de sábado, apresentando formato característico para este tipo dia, nota-se que no Grupo 15 a faixa de amplitudes que as curvas ocupam é maior do que no Grupo 14. Sendo esta mais uma característica captada pelos algoritmos de agrupamento e que, entre outras coisas, refletem as sazonalidades a que estas curvas estão submetidas. Como, por exemplo, aumento ou diminuição da temperatura ambiente diária ou fatores econômicos que implicam em maior ou menor consumo de energia.

Em relação ao conjunto de curvas utilizadas na etapa de teste das RNAs, pode-se verificar a distribuição mostrada na Tabela 10. Desta forma, a escolha dos grupos de treinamento para as RNAs obedece a disposição elencada nesta tabela, tornando possível verificar a hipótese

de que a utilização dos agrupamentos é capaz de fornecer subsídios para que as RNAs possam realizar uma previsão satisfatória e com baixos índices de erros.

Tabela 10 – Distribuição das curvas de demanda em relação aos dias utilizados na etapa de teste das RNAs (*feriado).

Grupo	**Data**	**Dia**
Grupo 3	06/11/2017	segunda-feira
	10/11/2017	sexta-feira
	13/11/2017	segunda-feira
	14/11/2017	terça-feira
	16/11/2017	quinta-feira
	30/11/2017	quinta-feira
Grupo 5	08/11/2017	quarta-feira
Grupo 7	02/11/2017	quinta-feira*
	05/11/2017	domingo
	12/11/2017	domingo
Grupo 12	01/11/2017	quarta-feira
	03/11/2017	sexta-feira
	09/11/2017	quinta-feira
	17/11/2017	sexta-feira
Grupo 15	04/11/2017	sábado
	11/11/2017	sábado

Ademais, a escolha de se dividir o conjunto de dados em 16 grupos se mostrou eficiente, resultando em grupos com boa quantidade de instâncias e que ao mesmo tempo mantiveram a qualidade, apresentando baixa heterogeneidade. Ainda que em alguns testes realizados tenha sido verificado a formação de grupos cada vez mais homogêneos, constatava-se também grupos compostos por menos instâncias e maior fragmentação dos dados ao longo destes grupos, o que poderia prejudicar a escolha adequada de grupos de treinamento para as RNAs. Portanto, embora tenham sido realizados experimentos com maior e menor quantidade foi verificado que uma menor quantidade resultou em grupos mais heterogêneos, ao passo que uma maior quantidade de grupos resultou em grupos mais homogêneos, mas com menor quantidade de instâncias. Como o objetivo desta fase do trabalho era gerar grupos que figurassem relativamente bem diante destas duas características (homogeneidade/quantidade de instâncias), os 15 grupos efetivamente obtidos pelo processo de votação se mostraram eficientes para o propósito deste trabalho. Ainda assim, aponta-se que esta não é uma quantidade absoluta, e foi utilizada com base em experimentações. Desta forma, não seria surpresa se uma quantidade de

grupos ligeiramente diferente da utilizada neste trabalho também fosse capaz de fornecer bons resultados de previsão para as redes neurais.

Na próxima seção são descritos os resultados de previsão obtidos por meio das redes neurais analisadas.

7.3 Resultados de Previsão para as RNAs

Diante dos resultados obtidos para os algoritmos de agrupamento, o passo seguinte consistiu em verificar o desempenho dos três tipos de redes neurais dinâmicas previamente discutidos nesta dissertação. Como mencionado anteriormente, a escolha de um modelo adequado de rede neural geralmente consiste em uma tarefa de grande dificuldade, sobretudo devido à imensa variedade de combinações de parâmetros que uma arquitetura pode fornecer, cabendo ao projetista a escolha e ajuste destes parâmetros de maneira adequada. Diversas combinações de parâmetros foram testadas em relação ao número de camadas escondidas, dimensão dos regressores, número de neurônios e, também, algoritmos de treinamento (Capítulo 5). Chegando-se à conclusão que dentre os parâmetros testados, os utilizados para gerar os resultados presentes neste capítulo foram os melhores.

Tais aspectos foram discutidos no Capítulo 6, incluindo topologias e explicações sobre como é feita a organização dos dados para o processo de treinamento das RNAs, com base nos agrupamentos previamente obtidos. Os resultados mostrados nesta seção são referentes às predições recursivas, ou seja, à medida que as redes neurais estimam os valores em cada instante das séries temporais em suas camadas de saída, estes valores são realimentados para as suas respectivas camadas de entrada, permitindo que os modelos propostos possam realizar uma previsão de curto prazo em um horizonte de 24 horas a cada cinco minutos (288 passos).

Com o objetivo de verificar o desempenho das redes neurais frente a diferentes dimensões dos regressores de entrada, foram realizadas diversas etapas de treinamento e teste com o intuito de verificar qual configuração seria capaz de fornecer os melhores resultados para cada RNA. A precisão dos modelos é medida por meio dos índices de desempenho discutidos na Seção 6.6. A partir de ciclos de treinamentos/testes em que as RNAs são submetidas, os índices são aplicados, tornando possível a avaliação dos modelos por meio da comparação entre os valores estimados das séries temporais e os valores reais observados.

Os resultados para os índices de desempenho NMSE e MAPE podem ser vistos nas Tabelas 11 e 12. Para cada um dos 16 dias analisados, as RNAs foram testadas e treinadas 10

vezes, a fim de se obter um valor de precisão estatística.

Tabela 11 – Mediana e desvio padrão dos NMSE obtidos. Valores destacados indicam o melhor resultado (*feriado).

Dia	Data	NARX		LSTM		FTDNN	
		Mediana	Desvio	Mediana	Desvio	Mediana	Desvio
qua.	01/nov/17	**0,0236**	0,008	0,0241	0,007	0,0276	0,005
qui.*	02/nov/17	0,0666	0,037	**0,0431**	0,036	0,0843	0,110
sex.	03/nov/17	0,0254	0,012	**0,0156**	0,004	0,0199	0,053
sáb.	04/nov/17	0,0996	0,043	**0,0693**	0,016	0,1287	0,041
dom.	05/nov/17	0,0303	0,012	**0,0193**	0,008	0,0336	0,021
seg.	06/nov/17	0,0218	0,015	**0,0142**	0,005	0,0284	0,019
qua.	08/nov/17	0,1562	0,065	**0,0779**	0,036	0,1498	0,062
qui.	09/nov/17	0,0613	0,052	**0,0323**	0,013	0,0539	0,056
sex.	10/nov/17	**0,0156**	0,009	0,0361	0,010	0,0231	0,010
sáb.	11/nov/17	0,0513	0,019	**0,0457**	0,008	0,0535	0,019
dom.	12/nov/17	0,0448	0,013	0,0448	0,011	**0,0363**	0,022
seg.	13/nov/17	**0,0110**	0,005	0,0120	0,008	0,0118	0,004
ter.	14/nov/17	**0,0106**	0,005	0,0217	0,015	0,0134	0,009
qui.	16/nov/17	0,0410	0,008	**0,0357**	0,008	0,0426	0,009
sex.	17/nov/17	0,0444	0,008	0,0421	0,006	**0,0419**	0,018
qui.	30/nov/17	0,0177	0,007	0,0128	0,006	**0,0114**	0,011

Em geral, nota-se que os três tipos de redes neurais conseguiram oferecer bons desempenhos. Com exceção de apenas três casos, em todos os outros a mediana do NMSE obtido ficou abaixo de 0,1. Ressalta-se que, quanto menor o valor do NMSE, melhor é o desempenho da previsão, valores próximos ou maiores que 1 indicam uma previsão ruim.

Em relação ao erro percentual absoluto médio (MAPE), as redes NARX e FTDNN obtiveram em dois casos erro superior a 5%. Entretanto em todos os outros casos observados, o valor do MAPE esteve abaixo deste valor.

Embora seja possível constatar a boa precisão obtida pelas redes neurais, nota-se que, a rede LSTM, conseguiu melhores índices de exatidão na maioria dos dias considerados. Sendo este um aspecto notável, principalmente nos casos em que as redes NARX e FTDNN obtiveram os maiores erros, como ocorre nos dias 02, 04, 08 e 09 de novembro, em que a LSTM consegue índices de precisão razoavelmente melhores que as outras duas redes. Não obstante, deve-se observar que existem casos em que tanto a NARX quanto a FTDNN superam a LSTM, indicando que, embora sejam arquiteturas de redes neurais mais simples, as mesmas também são capazes de oferecer resultados equiparáveis aos de modelos mais complexos, como é o caso da LSTM.

Tabela 12 – Mediana e desvio padrão dos MAPE obtidos (%). Valores destacados indicam o melhor resultado (*feriado).

Dia	Data	NARX		LSTM		FTDNN	
		Mediana	Desvio	Mediana	Desvio	Mediana	Desvio
qua.	01/nov/17	**2,27**	0,34	2,49	0,26	2,42	0,19
qui.*	02/nov/17	4,04	0,93	**3,15**	0,99	4,06	1,84
sex.	03/nov/17	2,22	0,45	**1,87**	0,30	1,96	1,54
sáb.	04/nov/17	2,37	0,46	**1,90**	0,25	2,74	0,51
dom.	05/nov/17	2,35	0,50	**1,92**	0,32	2,51	0,65
seg.	06/nov/17	2,35	0,80	**2,15**	0,45	2,74	0,87
qua.	08/nov/17	5,84	1,23	**4,04**	0,88	5,48	1,02
qui.	09/nov/17	4,58	1,78	**2,82**	0,39	4,07	1,35
sex.	10/nov/17	**1,78**	0,55	2,91	0,52	2,19	0,49
sáb.	11/nov/17	1,92	0,40	**1,74**	0,18	1,86	0,37
dom.	12/nov/17	2,97	0,49	2,88	0,41	**2,60**	0,72
seg.	13/nov/17	1,86	0,36	1,90	0,49	**1,76**	0,32
ter.	14/nov/17	**1,41**	0,33	2,08	0,41	1,59	0,45
qui.	16/nov/17	3,07	0,35	**2,89**	0,43	3,22	0,42
sex.	17/nov/17	**2,94**	0,30	3,12	0,19	3,17	0,66
qui.	30/nov/17	2,07	0,40	1,87	0,38	**1,70**	0,59

Os baixos valores de desvios padrões observados mostram baixa variabilidade de resultados entre os valores médios e os máximos e mínimos obtidos. Estes dois últimos podem ser vistos nas Tabelas 13 e 14, para o NMSE e MAPE, respectivamente. Observando-se a Tabela 14, é possível perceber que mesmo considerando-se apenas os valores máximos de erro, as redes neurais foram capazes de fornecer percentuais de erro baixos. Neste aspecto, a rede LSTM também sobressai-se, sendo que apenas nos dias 02 e 08 de novembro, o erro obtido foi superior a 5%. Analisando-se concomitantemente os dados contidos nas Tabelas 13 e 14, constata-se novamente a melhor capacidade da rede LSTM em obter os menores erros de previsão.

Desta forma, embora haja uma maior precisão por parte da rede LSTM, é possível apontar que mesmo havendo uma inicialização aleatória dos pesos sinápticos em cada ciclo de treinamento/teste, as redes neurais conseguem, na maioria das vezes, chegar a um ótimo local satisfatório, mostrando a capacidade das mesmas em captar o comportamento dinâmico das séries temporais. Destaca-se, também, o bom desempenho mesmo diante de um feriado, como ocorreu no dia 02 de novembro de 2017. Além disso, constata-se que, em relação à curva do dia 08 de novembro, as redes neurais obtiveram maior dificuldade em obter uma boa precisão, o que é evidenciado nos valores relativamente maiores das medianas dos erros obtidos neste dia. Para efeito de comparação entre as RNAs, os resultados mostrados nas Tabelas 11 a 14

Tabela 13 – Valores mínimos e máximos dos NMSE obtidos. Valores destacados indicam o melhor resultado (*feriado).

Dia	Data	NARX		LSTM		FTDNN	
		Mínimo	Máximo	Mínimo	Máximo	Mínimo	Máximo
qua.	01/nov/17	**0,0148**	0,0399	0,0157	0,0384	0,0231	**0,0364**
qui.*	02/nov/17	0,0496	0,1543	**0,0245**	**0,1230**	0,0355	0,3411
sex.	03/nov/17	0,0129	0,0555	**0,0087**	**0,0220**	0,0122	0,1850
sáb.	04/nov/17	0,0571	0,2127	**0,0454**	**0,1037**	0,0696	0,1765
dom.	05/nov/17	0,0142	0,0467	**0,0135**	**0,0366**	0,0178	0,0909
seg.	06/nov/17	0,0127	0,0518	**0,0060**	**0,0231**	0,0149	0,0715
qua.	08/nov/17	0,0389	0,2548	**0,0259**	**0,1500**	0,0916	0,3013
qui.	09/nov/17	0,0240	0,1950	0,0216	**0,0616**	**0,0215**	0,1775
sex.	10/nov/17	**0,0081**	**0,0335**	0,0154	0,0454	0,0155	0,0489
sáb.	11/nov/17	0,0344	0,0900	**0,0328**	**0,0553**	0,0336	0,0991
dom.	12/nov/17	0,0239	0,0658	0,0235	**0,0598**	**0,0214**	0,0890
seg.	13/nov/17	0,0060	**0,0195**	0,0079	0,0275	**0,0059**	0,0204
ter.	14/nov/17	**0,0043**	**0,0181**	0,0147	0,0578	0,0054	0,0313
qui.	16/nov/17	0,0319	0,0530	**0,0160**	**0,0453**	0,0320	0,0620
sex.	17/nov/17	0,0388	0,0607	0,0340	**0,0519**	**0,0266**	0,0829
qui.	30/nov/17	0,0070	**0,0258**	**0,0065**	**0,0258**	0,0077	0,0432

são referentes aos melhores resultados medianos obtidos pelas mesmas para cada dia analisado, independentemente da dimensão dos regressores utilizados. O desempenho das RNAs em função da dimensão dos regressores é discutido no decorrer deste capítulo.

As Figuras 35 a 38 ilustram a previsão obtida para o melhor modelo de cada uma das três RNAs. Na Figura 35 percebe-se que na maioria dos casos, as três RNAs conseguiram realizar uma boa previsão, com as curvas estimadas conseguindo acompanhar de maneira bastante próxima o comportamento das séries temporais originais. O bom desempenho das RNAs em prever o feriado do dia 02 de novembro pode ser visualizado na Figura 35(b), ratificando os baixos índices de erros mostrados nas Tabelas 11 a 14. Embora seja possível perceber um maior desvio da rede FTDNN em relação à série observada no intervalo compreendido entre as 15 e 18 horas do dia.

Em relação a este aspecto, todos os anos, no dia 02 de novembro ocorre o feriado nacional de finados. No ano de 2017, este feriado ocorreu durante uma quinta-feira. Portanto, mesmo se tratando de um dia útil, a ocorrência do feriado ocasionou grandes mudanças no padrão de consumo para este dia, fazendo com que a curva de demanda apresentasse um formato semelhante ao apresentado por uma curva típica de domingo. Portanto, é necessário apontar que a estratégia de agrupamento das séries temporais permitiu às redes neurais realizarem uma

Tabela 14 – Valores (%) mínimos e máximos dos MAPE obtidos. Valores destacados indicam o melhor resultado (*feriado).

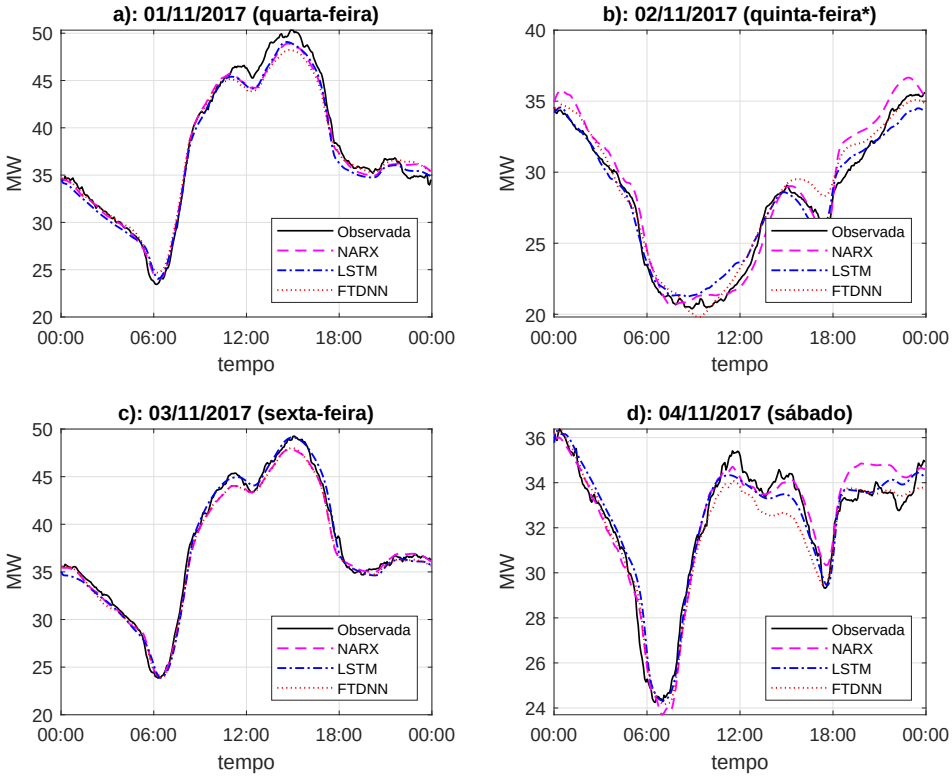
Dia	Data	NARX		LSTM		FTDNN	
		Mínimo	Máximo	Mínimo	Máximo	Mínimo	Máximo
qua.	01/nov/17	**1,83**	3,00	2,11	2,93	2,20	**2,86**
qui.*	02/nov/17	3,07	**5,38**	**2,40**	5,43	2,73	7,91
sex.	03/nov/17	1,57	3,19	**1,45**	**2,47**	1,55	6,66
sáb.	04/nov/17	1,87	3,55	**1,58**	**2,44**	1,91	3,29
dom.	05/nov/17	**1,64**	3,12	1,70	**2,63**	1,83	4,09
seg.	06/nov/17	1,60	4,09	**1,25**	**2,80**	1,79	4,45
qua.	08/nov/17	2,73	7,10	**2,04**	**5,35**	4,49	7,88
qui.	09/nov/17	2,78	8,59	**2,41**	**3,76**	3,03	7,21
sex.	10/nov/17	**1,32**	**2,98**	1,90	3,38	1,64	3,31
sáb.	11/nov/17	1,46	2,66	**1,41**	**2,10**	1,46	2,71
dom.	12/nov/17	2,21	3,62	2,05	**3,47**	**2,01**	4,28
seg.	13/nov/17	1,30	**2,31**	1,54	2,90	**1,27**	2,39
ter.	14/nov/17	**0,91**	**1,92**	1,78	3,02	1,04	2,18
qui.	16/nov/17	2,67	3,69	**1,86**	**3,39**	2,67	3,96
sex.	17/nov/17	2,82	3,66	2,84	**3,41**	**2,27**	4,43
qui.	30/nov/17	**1,32**	2,56	1,33	**2,47**	1,39	3,30

boa previsão para este dia, visto que, de outro modo, as redes neurais teriam previsto a curva de demanda da quinta-feira como se fosse a de um dia útil comum. Portanto, verificar que esta quinta-feira se trata de um feriado e submetê-la a um conjunto de treinamento (grupo) adequado mostra-se um fator preponderante para uma boa previsão em casos semelhantes a este.

Embora as quintas-feiras sejam consideradas dias úteis e em situações normais possuam curvas de demanda com formato característico semelhantes às observadas nas Figuras 35(a) e 35(c), a referente ao dia 02 se encontrava alocada em um grupo composto somente por curvas de demanda de domingos e feriados, este foi um fator determinante para que as redes conseguissem fazer uma previsão adequada, mesmo considerando que os dias que precedem os domingos são curvas de carga de sábados, enquanto a que precede a curva de quinta-feira em questão, seja uma curva de carga típica de dia útil, no caso uma quarta-feira. Como o agrupamento utilizado na etapa de teste para a previsão desta curva era composto por ambos os tipos de curvas, foi possível fornecer este "conhecimento" às RNAs, fazendo com que as mesmas incorporassem tal dinâmica aos seus pesos sinápticos.

Outra curva que merece destaque, é a referente ao dia 08 de novembro (Figura 36). Por algum motivo desconhecido, o padrão de consumo para este dia apresentou certa diferença em relação a outros dias úteis do mês, fazendo com que as curvas apresentassem dois picos

Figura 35 – Previsão para os dias 01 a 04/11/2017.



de amplitudes praticamente iguais, antes e após as 12 horas. Observando-se também a baixa amplitude de demanda próxima a este horário e, também a partir das 18 horas. Tendência, esta, que se manteve até as primeiras horas do de seguinte (09 de novembro).

Como observado anteriormente por meio dos índices de desempenho, nestes dois dias as RNAs mostraram uma dificuldade maior em estimá-los, resultando em maiores erros de previsão devido a algumas características atípicas que ambos apresentaram. Embora haja a presença de tais características, as três RNAs foram capazes de acompanhar adequadamente as curva observadas, o mesmo ocorreu para as curvas dos outros dias ilustradas na Figura 36.

Embora seja possível verificar o bom desempenho das RNAs propostas juntamente com a utilização dos algoritmos de agrupamento, é preciso investigar até que ponto os agrupamentos podem favorecer a capacidade das RNAs fornecerem uma boa previsão. Antes disso, é necessário discutir brevemente os regressores utilizados para treinamento e teste das RNAs. Na rede NARX, a dimensão do regressor d_y é definida a partir de $d_e \times \tau$, ou seja, esta rede possui dois regressores, d_y e d_e, sendo τ o atraso de imersão do regressor d_e. Além disso, foi utilizado o modo de treinamento série-paralelo, conforme discutido na Seção 5.3. As redes FTDNN e LSTM, por outro lado, possuem apenas um regressor na camada de entrada.

Figura 36 – Previsão para os dias 05 a 09/11/2017

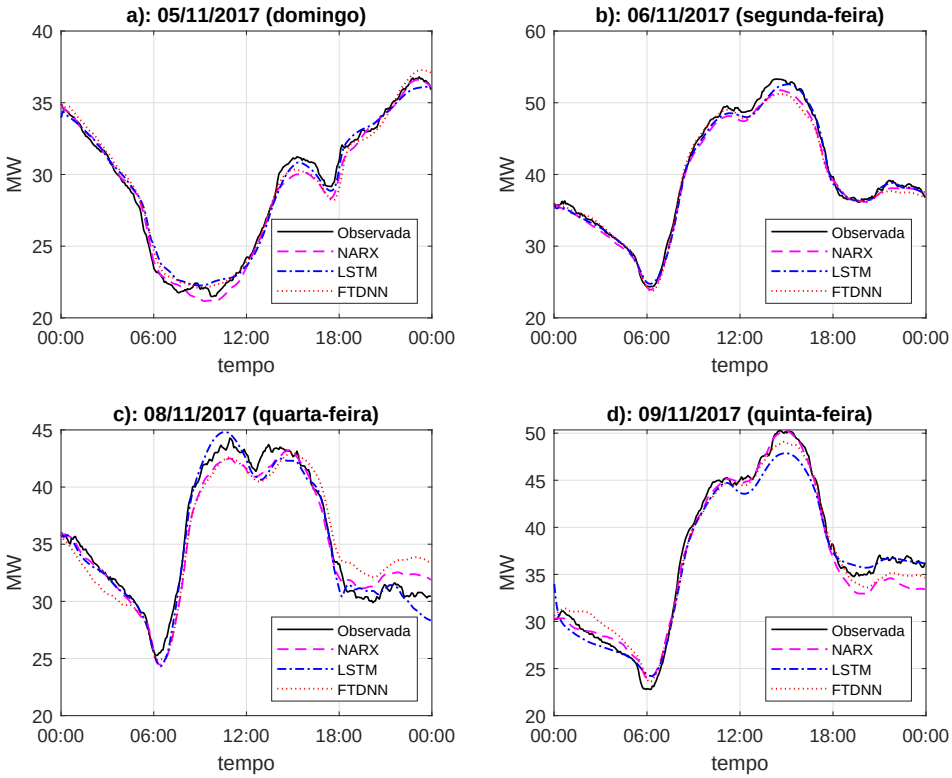


Figura 37 – Previsão para os dias 10 a 13/11/2017

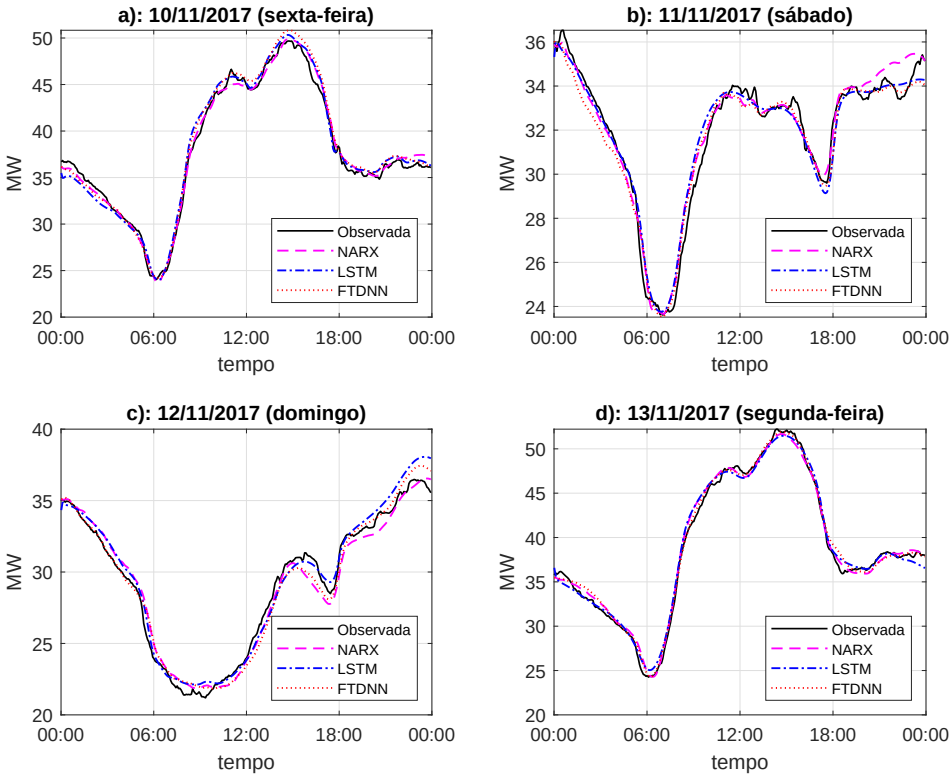
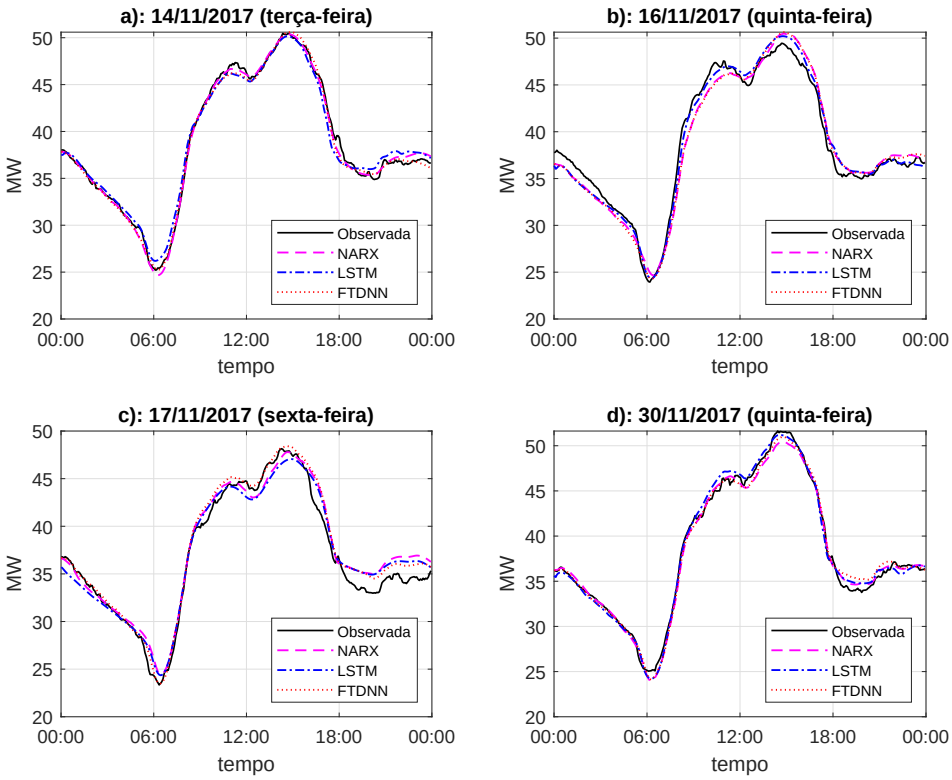


Figura 38 – Previsão para os dias 14 a 30/11/2017

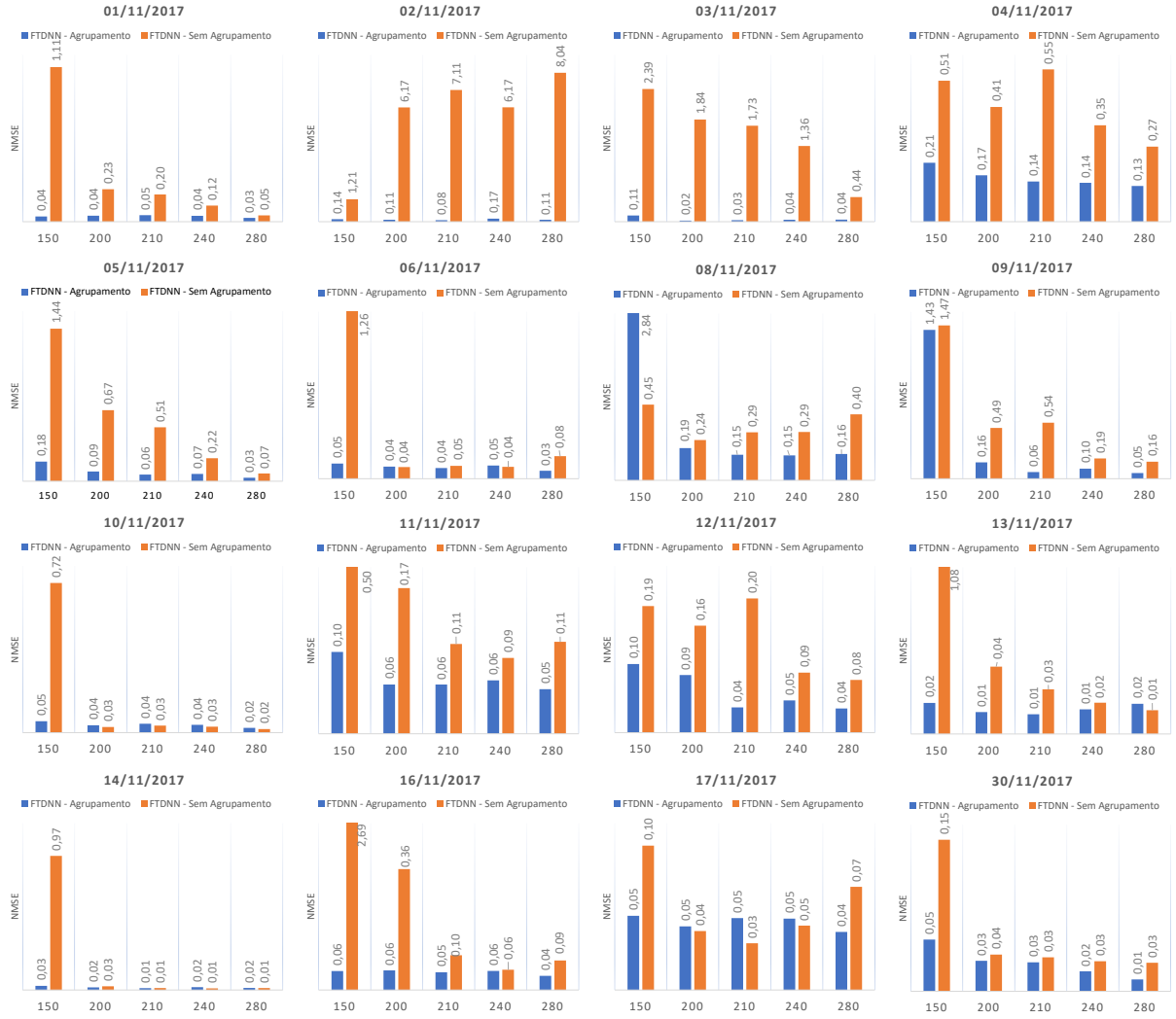


Para que fosse possível verificar o desempenho da metodologia proposta, as RNAs também foram submetidas a treinamento sem a existência dos agrupamentos obtidos por meio dos algoritmos não-supervisionados, considerando-se diferentes tamanhos de regressores. Os resultados provenientes das etapas de teste, e mensurados em termos do NMSE, foram comparados com os resultados obtidos a partir da combinação de RNAs/Agrupamento, considerando-se, também, diferentes valores para os regressores das RNAs.

Foram escolhidas duas faixas de valores para τ e d_e. $\tau = [5,\ 7]$ e $d_e = [30,\ 40]$, considerando-se a rede NARX, para a rede FTDNN foram feitos experimentos para os valores de $d_y = [150,\ 200,\ 210,\ 240,\ 280]$, enquanto para a rede LSTM foram testados $d_y = [200,\ 270]$. Embora outros testes experimentais tenham sido realizados com outros valores de dimensão para os regressores, foi verificado que, diante do sistema sob análise, os valores apresentados foram capazes de fornecer os melhores resultados. Os resultados que constam nas Tabelas 11 a 14 foram obtidos a partir destas configurações, considerando-se, os regressores capazes de fornecer menores erros de previsão em cada dia analisado.

O treinamento alternativo ao qual as RNAs foram submetidas consistiu em escolher conjuntos de dados de treinamento com base no dia da semana no qual se deseja realizar a

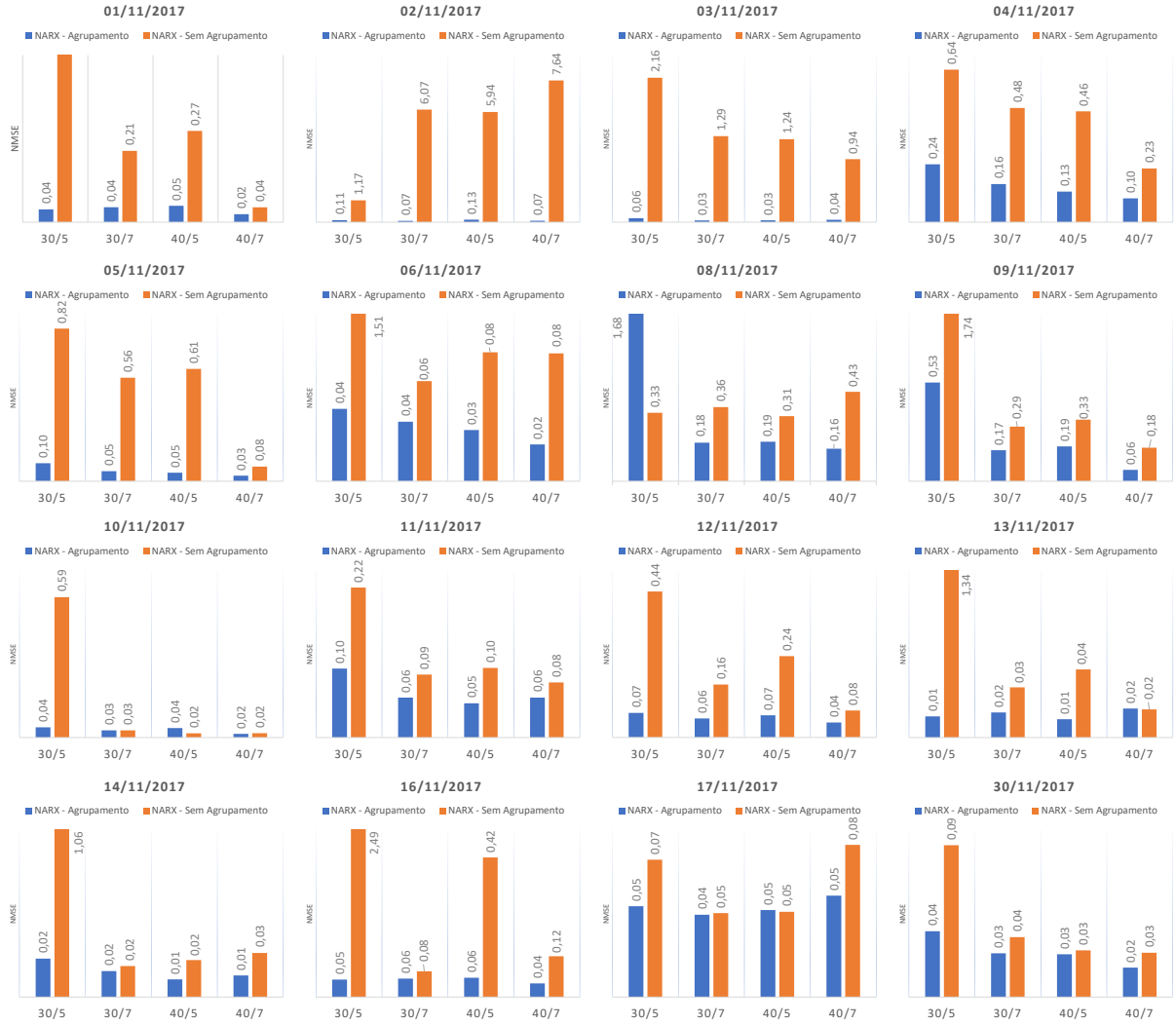
Figura 39 – Medianas do NMSE em função do tamanho do regressor d_y, obtidas com a rede FTDNN, com treinamentos realizados com e sem uso de agrupamentos.



previsão. Como consequência, se a curva a ser prevista era relativa uma segunda-feira, eram escolhidas a partir do conjunto de dados, curvas anteriores que também foram geradas durante segundas-feiras de meses e anos anteriores para comporem o conjunto de treinamento das RNAs. O mesmo raciocínio era válido para outras previsões.

Na Figura 39 é avaliada a precisão obtida para a rede FTDNN treinada com e sem o uso dos algoritmos de agrupamento. Pode-se perceber que, nos casos em que os treinamentos da RNA foram realizados sem a utilização dos agrupamentos obtidos preliminarmente pelos algoritmos não-supervisionados, o desempenho foi menos satisfatório. Demonstrando uma maior robustez dos modelos que utilizam agrupamentos. Nota-se que nos casos em que foram utilizados regressores com tamanho igual 150, sempre houve uma diferença de erro consideravelmente alta. Na maioria das vezes em que ocorreram resultados semelhantes entre ambos os modelos, foram

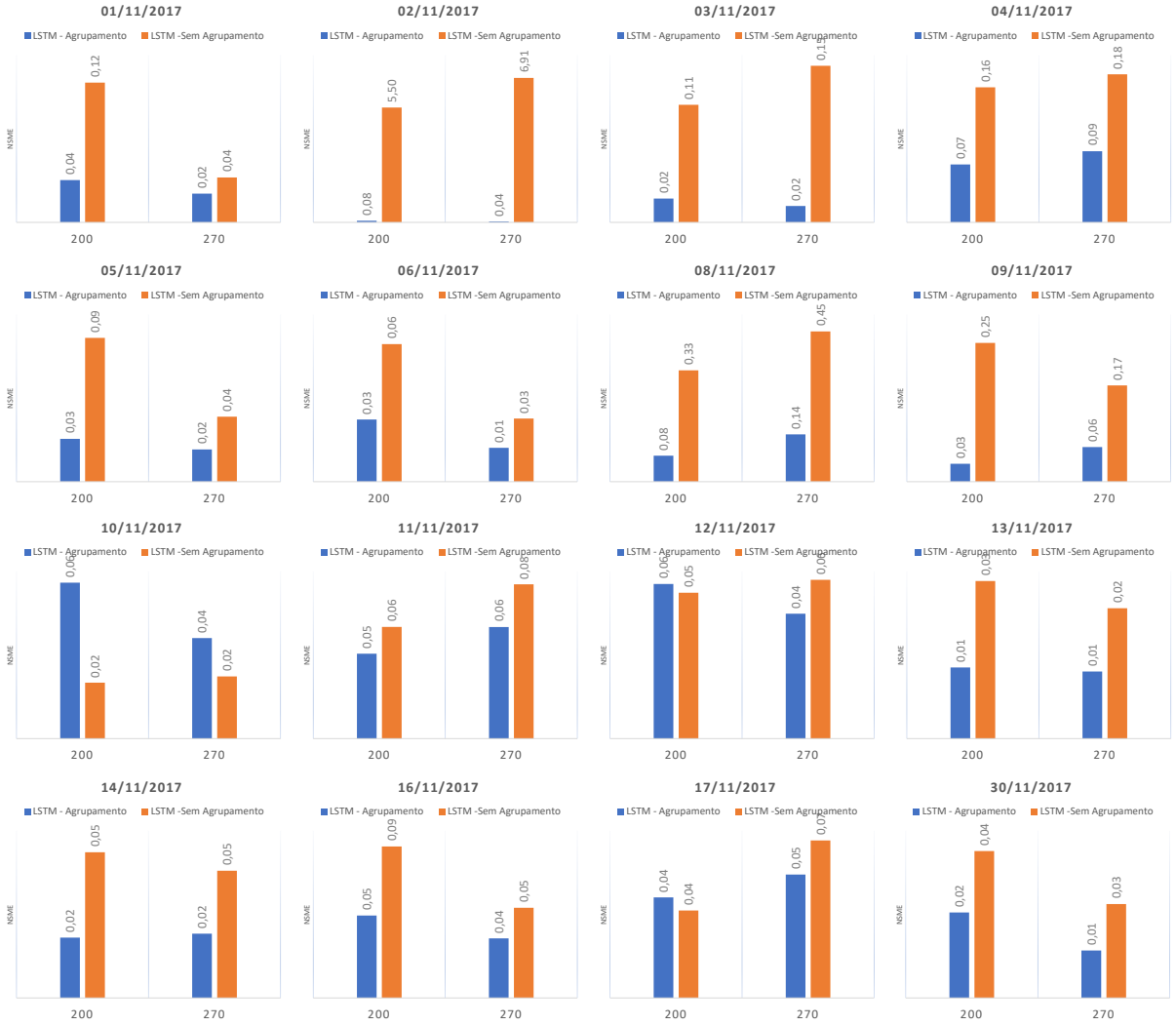
Figura 40 – Medianas do NMSE em função de τ e d_e, obtidas com a rede NARX, com treinamentos realizados com e sem uso de agrupamentos.



nas situações em que foram utilizados regressores com dimensões maiores que 150, mas mesmo levando-se isto em consideração, existem casos em que existe uma diferença significativamente perceptível de desempenho entre os dois casos analisados.

O mesmo tipo de comparação é feita para a rede NARX na Figura 40. Assim como ocorreu para o caso da rede FTDNN, os erros obtidos pela NARX com a utilização de agrupamentos foi consideravelmente menor para as previsões dos cinco primeiros dias analisados. A tendência em se produzir altos valores de erros de previsão, quando se utiliza regressores com tamanho $d_y = \tau \times d_e = 150$ com o modelo sem a utilização de agrupamentos, também se repete para a rede NARX. Outra característica que pode ser notada tanto no caso da Figura 39, quanto no caso da Figura 40, é que na maioria das situações em que o modelo sem agrupamento obteve melhor desempenho, a diferença do NMSE em relação aos modelos com agrupamento não foi

Figura 41 – Medianas do NMSE em função do tamanho do regressor d_y, obtidas com a rede LSTM, com treinamentos realizados com e sem uso de agrupamentos.



significativa. Isto pode ser observado, por exemplo, no dia 10 de novembro das Figura 39 e 40 e no dia 14 de novembro da Figura 39.

Os resultados para a rede LSTM utilizando as duas formas de treinamento é visto na Figura 41. Análise de desempenho da rede LSTM permite corroborar os resultados obtidos anteriormente para as redes FTDNN e NARX. Mostrando o melhor desempenho da RNA quando considerado o treinamento com agrupamentos, na maioria das situações. Um aspecto comum entre os três tipos de RNAs é a diminuição dos erros de previsão à medida em que os regressores assumem valores maiores que 150, sendo, em muitas situações a maior dimensão considerada a que permitiu obter os melhores níveis de precisão, sobretudo em relação aos treinamentos realizados com a utilização dos agrupamentos. Entretanto, é possível verificar que também existem casos em que ocorreram pouca ou nenhuma melhora das previsões, independentemente

da dimensão dos regressores utilizados, como é possível verificar, por exemplo, no dia 14 de novembro das Figuras 39, 40 e 41.

De fato, o procedimento alternativo para escolha dos conjuntos de treinamento das RNAs também pode ser interpretado como uma forma de agrupamento. Entretanto, o agrupamento é realizado de maneira empírica, em que são levadas em consideração pressupostas semelhanças existentes entre curvas de demanda geradas em mesmos dias. Embora esta suposição não seja incorreta, é necessário atentar para a possibilidade destas não possuírem dinâmicas tão semelhantes em algumas situações, como discutido na Seção 7.2. Sobretudo devido à interferência de fatores climáticos, econômicos, ocorrência de feriados ou padrões de consumo atípicos. Por outro lado, algoritmos de agrupamento, como os utilizados neste trabalho, realizam uma análise mais meticulosa de cada uma das séries temporais, verificando a existência de semelhanças por meio de características que vão além do simples dia da semana em que os dados são gerados. Característica que se reflete em conjuntos de treinamentos mais coesos, e que, portanto, podem permitir uma melhor capacidade de aprendizado das redes neurais.

A capacidade dos algoritmos de agrupamento em melhorar a capacidade de redes neurais também pode ser observada no estudo de Panapakidis (2016b), em que verifica-se que o uso de redes do tipo *feedforward* obtém melhor de desempenho de previsão quando são utilizados conjuntos de treinamento obtidos a partir da utilização de algoritmo não supervisionado para a geração de agrupamentos. Entretanto, nesta dissertação, em vez de se utilizar apenas um algoritmo, é feito um consenso entre dois algoritmos, por meio da geração de diversas partições, possibilitando maior robustez nos agrupamentos gerados.

Uma outra diferença observada entre o estudo realizado nesta dissertação e o proposto por Panapakidis (2016b), é que neste último as redes neurais utilizadas são do tipo *feedfoward*, ou seja, redes estáticas, sem a presença de estruturas de realimentação. Desta forma a estimação dos valores de saída são definidas apenas em função dos valores de entrada das RNAs, não levando em consideração, valores em instantes de tempo passados gerados pelas RNAs, ao contrário do que ocorre com os modelos de RNAs utilizadas nesta dissertação, que possuem natureza dinâmica. É possível verificar que os erros obtidos para conjuntos de testes para previsões em um horizonte de 24 horas, ficaram, na maioria dos casos, acima de 4 ou 5% de erro. Todavia, é necessário ressaltar que os dados utilizados nesta dissertação e os utilizados em Panapakidis (2016b) são de naturezas diferentes. Além disso, neste último, foi utilizada outra metodologia para realizar as previsões e o uso de variáveis exógenas. Ademais, o horizonte de previsão

é de 24 horas (24 passos), ou seja, cada série temporal é composta por apenas 24 amostras. Característica também notada em outros estudos mencionados na Seção 1.2, ao contrário das 288 amostras que compõem cada uma das séries temporais utilizadas no estudo proposto nesta dissertação. Pode-se afirmar também, que os sistemas utilizados em cada um dos estudos citados, incluindo o presente estudo, podem diferir quanto a dinâmica apresentada, devido às diferentes cargas que alimentam e outros fatores relacionados a diferenças geográficas e socioeconômicas.

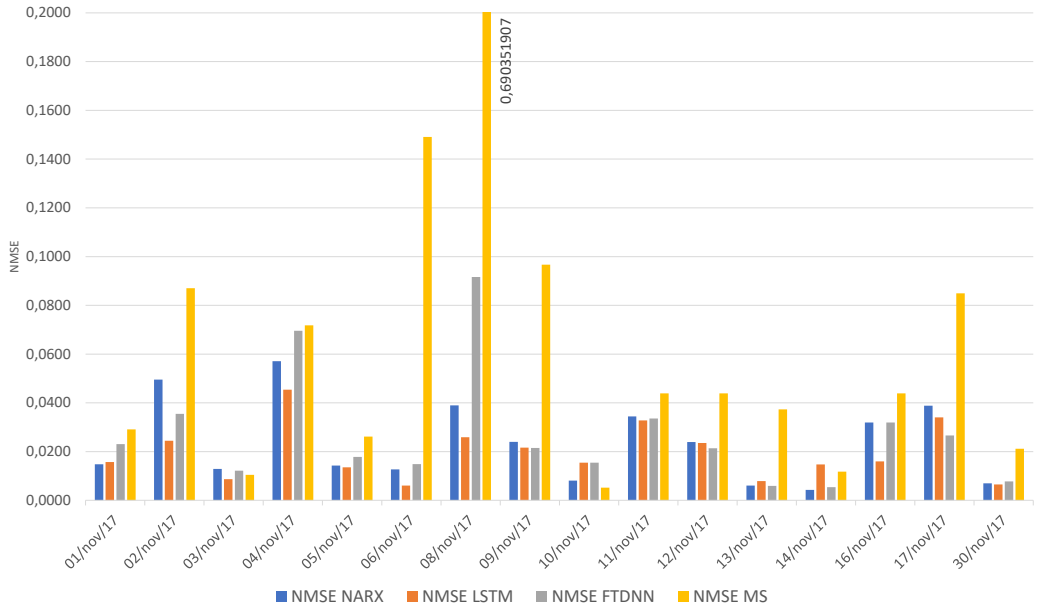
Sendo assim, devido às diferenças observadas, não é possível afirmar que uma metodologia é capaz de se sobrepor à outra em termos de qualidade. Entretanto, é importante salientar que a boa precisão obtida até o momento para as as três redes dinâmicas analisadas, sem o uso de qualquer variável exógena e contemplando um horizonte de previsão de 24 horas em 288 passos, com realimentação dos valores estimados, indica a capacidade desses modelos em lidar de maneira eficiente com o problema de previsão em curto prazo. Podendo ser uma boa alternativa em relação à utilização de RNAs estáticas como as utilizadas em (PANAPAKIDIS, 2016a) e outros estudos.

Embora tenha sido possível observar o bom desempenho obtido, levando-se em consideração o uso de agrupamentos e redes neurais dinâmicas. É possível afirmar que a utilização da média simples também é capaz de fornecer uma boa previsão, por meio de semelhanças existentes entre curvas de demanda geradas, considerando-se um determinado período de tempo. Desta forma, a média simples é a soma de duas ou mais séries temporais, somadas ponto a ponto (ou seja, soma-se as amostras das séries pertencentes a um mesmo instante de tempo), divididas pelo número de séries temporais que compõem a soma, como descrito na Seção 2.4.2.

A Figura 42 ilustra o desempenho obtido com os melhores modelos de RNAs treinadas, em relação às previsões obtidas por meio de média simples. Para a composição da média, foram consideradas quatro curvas de mesmo dia geradas anteriormente nas últimas semanas que antecedem o dia em que é realizada a previsão. Nota-se que esta estratégia também forneceu bons desempenhos em determinados casos. Todavia é perceptível a melhor precisão obtida pelos modelos neurais na maior parte dos testes realizados.

Isto é ainda mais perceptível quando observados os dias 02, 06, 08, 09, 12, 13 e 17 de novembro, casos em que uma ou mais das RNAs obtiveram erros consideravelmente menores que a média simples. Observando-se os outros casos, percebe-se ainda que pelo menos um dos modelos de RNAs é capaz de superar os resultados obtidos pela média simples, embora seja visto que, em relação ao dia 10 de novembro, a média simples obteve o melhor resultado.

Figura 42 – NMSE obtido para as redes NARX, LSTM e FTDNN e para a Média Simples.

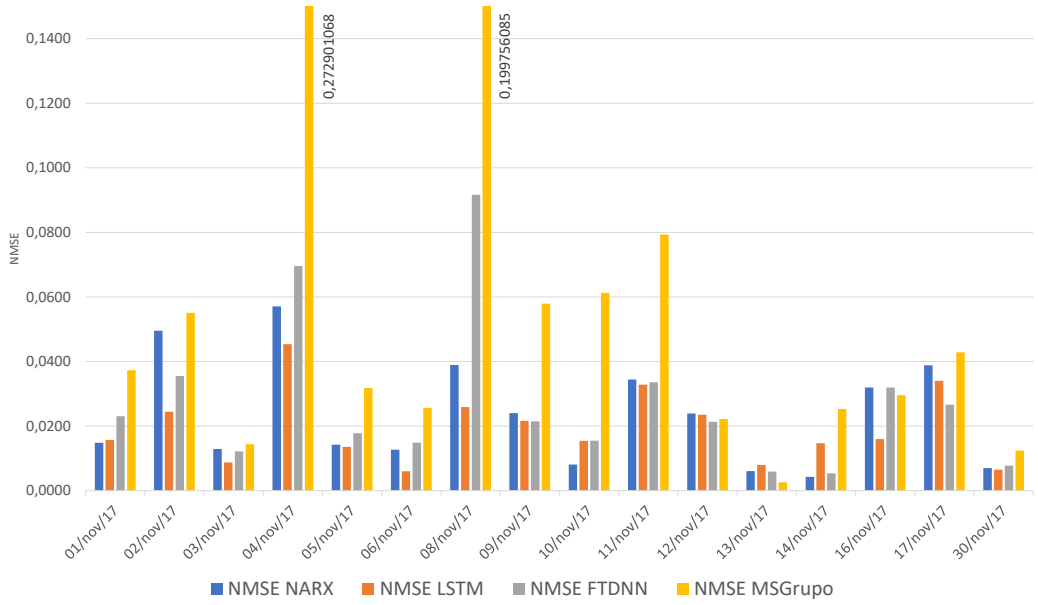


Um ponto a ser destacado é que a previsão para o dia 02 de novembro não foi feita a partir da média das últimas quintas-feiras observadas, mas sim com a média dos últimos feriados do dia 02 de novembro observados no conjunto de dados. Visto que, no primeiro caso, os erros ficam demasiadamente elevados (NMSE = 5,7865) devido às características completamente diferentes que o feriado apresenta em relação a dias úteis. Utilizando-se o segundo modo o NMSE foi reduzido para apenas 0,08703. Ressalta-se, também, que as médias obtidas para os outros dias não levou em consideração a curva gerada no dia 02 ou de qualquer outro feriado, em virtude dos seus comportamentos atípicos, e que poderiam prejudicar o desempenho oferecido pela média simples.

Diante deste aspecto, é importante fazer uma observação. Como visto na Figura 42, a média simples de fato consegue oferecer resultados muito bons em determinados casos. Principalmente em situações em que se dispõe de dados com grande nível de semelhança, possibilitando previsões com grande nível de assertividade. Entretanto, é preciso notar que situações como esta podem não ocorrer com frequência, ou seja, podem acontecer situações em que o conjunto de curvas anteriores que compõem a média simples possuam comportamentos dinâmicos que diferem sobremaneira dos demais, impactando, consequentemente, na previsão obtida. Em situações como esta, é preferível que estes dados não façam parte dos demais que compõem a média.

Uma hipótese que pode ser levantada é: se o mesmo procedimento de agrupamento realizado para as redes neurais poderia melhorar as previsões realizadas por meio de média

Figura 43 – NMSE obtido pas as redes NARX, LSTM e FTDNN e para a Média Simples com auxílio dos agrupamentos obtidos.



simples. Esta hipótese é factível, uma vez que os agrupamentos obtidos são compostos por instâncias com grande nível de semelhança entre si. Sendo assim, realizar a previsão com base na média das curvas temporalmente mais próximas à curva a ser prevista e contidas em um mesmo grupo poderia fazer com que os erros de previsão obtidos fossem reduzidos significativamente.

Os resultados obtidos para esta estratégia e, novamente comparados com as RNAs, podem ser observados na Figura 43. Embora seja possível notar a redução do erro das previsões obtidas por média simples em algumas situações, é possível verificar que ainda assim as RNAs são equiparáveis ou, em outras situações, conseguem sobrepujar a média simples. Mesmo diante do processo de agrupamento que, em teoria, poderia beneficiar este último modelo. Isto se deve à capacidade adaptativa das RNAs, adquirida por meio de aprendizado experimental extraído do ambiente de treinamento (HAYKIN, 2009), enquanto a média simples se trata apenas de uma medida estatística de tendência central. Portanto, para que seja possível obter previsões precisas com a mesma, é necessário que os dados possuam perfis muito semelhantes em relação à curva prevista, em virtude de eventuais desvios causados na dinâmica de uma ou mais das curvas que compõem a média, poderem causar previsões com elevados índices de erro.

Diante dos resultados expostos, é possível perceber que o uso das RNAs juntamente com algoritmos de agrupamento se mostrou uma estratégia competitiva frente a outras estratégias de previsão de curvas de demanda. As conclusões finais deste estudo são discutidas no capítulo seguinte.

8 CONCLUSÕES E TRABALHOS FUTUROS

A previsão de carga é um assunto importante e fonte de muitos estudos em pesquisas atuais. Isso se deve às vantagens que um processo de previsão robusto pode oferecer no planejamento e operação de sistemas de energia. Uma revisão sobre alguns pontos importantes sobre curvas de carga no contexto das séries temporais foi dada no Capítulo 2.

O estudo realizado nesta dissertação consistiu na aplicação de RNAs como modelos de previsão de curvas de demanda diárias de energia elétrica. Foram utilizadas três redes dinâmicas, a NARX, LSTM e FTDNN, cada uma possuindo suas próprias características quanto à forma de processar e armazenar a informação temporal das séries temporais sob estudo. Detalhes sobre estas redes foram dados no Capítulo 5.

Devido a existência de alguns fatores que podem prejudicar a qualidade dos dados coletados, foi necessário, antes, realizar um processo de filtragem, para que apenas os dados que não apresentassem problemas fossem mantidos no conjunto de dados a ser posteriormente analisado. Isto foi feito para que dados corrompidos não prejudicassem a performance e eficiência dos resultados obtidos pelos outros algoritmos utilizados. Para esta tarefa foram utilizados o Identificador de Hampel e Transformada *Wavelet*, descritos no Capítulo 3

Após realizada a etapa de filtragem os dados foram submetidos aos algoritmos de agrupamento *minCEntropy* e *k-means*, cujos modos de operação foram abordados no Capítulo 4. Estes algoritmos foram responsáveis por gerar um conjunto de partições que foram utilizadas como entrada de um algoritmo de combinação de agrupamentos baseado em função consenso por votação. A partir deste último, foi possível obter-se a partição consenso, que representava o agrupamento final obtido através da utilização destes três algoritmos. A metodologia utilizada nesta etapa e para as RNAs foi descrita no Capítulo 6.

Os resultados para os agrupamentos e as previsões realizadas pelas RNAs foram mostrados no Capítulo 7. Foi possível perceber que o agrupamento, composto por 15 grupos, demonstrou resultados razoavelmente uniformes, e cada grupo foi formado por séries temporais com características bem semelhantes, tanto para as curvas geradas em dias úteis, quanto para aquelas geradas em finais de semana ou até mesmo feriados. Na maior parte dos casos, foi possível separar padrões que possuíam características desconexas com o restante dos dados utilizados. Como ocorreu com as curvas de carga do ano de 2018, em que verificou-se uma mudança de topologia da linha de distribuição das quais foram medidas as curvas de carga. O mesmo foi verificado para algumas curvas do ano de 2015, em que foi possível separar algumas

curvas geradas em determinados sábados em um único grupo (Grupo 10), indicando padrões de consumo específicos.

As previsões das curvas de demanda diárias foram feitas com base nos agrupamentos obtidos, as três RNAs foram submetidas a treinamentos e testes, e 16 dias do mês de novembro de 2017 foram então previstos. Uma das vantagens em se utilizar esta metodologia de agrupamento de cargas consistiu em fornecer às RNAs, conjuntos de treinamento reduzidos e que ao mesmo tempo refletiam as sazonalidades existentes nestes tipos de dados, uma vez que a separação das curvas em grupos com diferentes faixas de amplitude pode ser considerada equivalente à relação existente entre o aumento de temperatura decorrente de determinadas épocas do ano, e aumento do consumo de energia elétrica. A eficiência deste modelo de previsão foi comparada junto a uma estratégia em que não se utilizou os agrupamento obtidos previamente pelos algoritmos não-supervisionados. E foi verificado que a utilização dos agrupamentos demonstrou desempenho bem superior na maioria dos casos.

As previsões obtidas por meio da metodologia proposta também foram comparadas aos resultados obtidos por meio de média simples, que é conhecidamente uma boa estratégia para realizar a previsão do tipo de série temporal de que trata este trabalho. Embora tenha sido percebido desempenhos bastante parecidos em certas ocasiões entre as duas formas de previsão, verificou-se que em certas ocasiões as redes neurais foram capazes de fornecer resultados mais satisfatórios.

As previsões realizadas pelas RNAs foram, em geral, satisfatórias. Demonstrando a eficiência dos algoritmos e da metodologia utilizada neste estudo. Considerando-se os valores apresentados pelos índices de desempenho, quando foram comparadas as três RNAs entre si, verificou-se que, na maioria das vezes, a rede LSTM obteve os melhores resultados. Entretanto, deve-se salientar que as redes NARX e FTDNN, embora possuam algoritmos de treinamento e arquiteturas mais simples, também foram capazes fornecer baixos índices de erro, e em certas ocasiões forneceram resultados melhores que a LSTM. Outro ponto a ser ressaltado foi a capacidade que a metodologia proposta obteve em prever de forma satisfatória o feriado do dia 02 de novembro. Geralmente um dos grandes desafios dos estudos que envolvem previsão de carga, estão relacionados com a capacidade de oferecer boas previsões em dias de consumo atípicos como são, geralmente, os feriados.

Dois outros pontos merecem destaque, a capacidade das RNAs preverem de forma recursiva as séries temporais 288 passos adiante, com baixos erros de previsão, e utilização de

uma abordagem univariada, mostrando que as próprias séries temporais são capazes de se auto-explicar, sem a necessidade da utilização de variáveis exógenas, como a temperatura ambiente, por exemplo.

Entretanto, mais estudos devem ser realizados para o desenvolvimento efetivo de uma metodologia que possibilite a melhor seleção possível dos grupos de treinamento das RNAs, pois como foi visto, os agrupamentos são capazes de fornecer meios para que as RNAs forneçam níveis de acurácia equiparáveis ou até mesmo melhores que outras metodologias.

Por fim, é possível afirmar que o estudo proposto permite a utilização de uma metodologia alternativa, baseada em técnicas de inteligência computacional para o problema de previsão de curvas de demanda. Isso contribui, entre outras coisas, para a otimização dos recursos energéticos, confiabilidade do sistema e aumento dos lucros das concessionárias de geração, transmissão e distribuição de energia elétrica.

Algumas perspectivas de trabalhos futuros incluem:

- Mesclar as arquiteturas de uma NARX e LSTM, oferecendo à LSTM a mesma recorrência obtida entre as entradas e saídas da rede NARX;
- Realizar uma busca exaustiva por hiperparâmetros ótimos de treinamento das redes analisadas;
- Verificar a influência das componentes de alta frequência das séries temporais na capacidade de previsão das redes, visto que as séries previstas consistem em versões suavizadas das séries originais. Para isto, poderia se utilizar a TW para realizar a filtragem destas componentes (processo semelhante ao exemplo mostrado na Seção 3.4, em que se realiza a filtragem de um sinal ruidoso), e utilizar versões suavizadas das curvas de demanda durante o processo de treinamento das redes neurais;
- Utilizar a TW para realizar a redução da dimensão das séries temporais, decompondo-as em componentes de baixa e alta frequência. E a partir disto utilizar redes neurais dinâmicas para realizar a previsão destas componentes. Uma vez obtidas, seria possível realizar a reconstrução do sinal previsto por meio da TWI. A redução da dimensionalidade das curvas poderia resultar em menor esforço computacional durante o treinamento das RNAs e previsões com maior nível de exatidão;
- Realizar a previsão utilizando como base a distribuição percentual da probabilidade da curva a ser prevista estar presente em um ou mais grupos. E após esta verificação, a previsão seria uma média ponderada das previsões geradas por uma RNA quando treinada

com os grupos em que houve a possibilidade da curva ser prevista estar atrelada;

- Fornecer como previsão um consenso das três redes neurais utilizadas. Visto que em certos trechos das curvas de demanda, uma ou outra RNA pode obter um nível de acurácia ligeiramente melhor que as demais;
- A formulação e aplicação de um sistema especialista capaz de reconhecer de maneira automática o melhor ou melhores grupos que podem ser utilizados, dentro de um conjunto de possibilidades, para treinamento das RNAs, levando-se em consideração a inferência de possíveis características do dia em que será realizada a previsão;
- Utilizar *Support Vector Regression* - SVR e *Least Squares - Support Vector Regression* - LS-SVR para se realizar a previsão de curto prazo e comparar os resultados com as ferramentas propostas nesta dissertação;
- Aplicar as redes NARX e LSTM em problemas de curvas de demanda com comportamento mais aleatório, como aquelas observadas diretamente em medições realizadas em alimentadores de distribuição;
- Aplicar as redes NARX e LSTM em problemas de previsão de longo prazo, contemplando um horizonte de 2 ou mais anos, como ferramenta de auxílio na expansão de sistemas de distribuição.

REFERÊNCIAS

ADDISON, P. S. **The illustrated wavelet transform handbook: introductory theory and applications in science, engineering, medicine and finance**. UK: Institute of Physics Publishing, 2002.

ADHIKARI, R.; AGRAWAL, R. K. An introductory study on time series modeling and forecasting. **arXiv preprint arXiv:1302.6613**, 2013.

AGÊNCIA NACIONAL DE ENERGIA ELÉTRICA. **Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional – PRODIST**: Módulo 2 – planejamento da expansão do sistema de distribuição. [S.l.], 2016. 24 p. Disponível em: <www.aneel.gov.br/modulo-2>. Acesso em: 05 jun. 2019.

AGÊNCIA NACIONAL DE ENERGIA ELÉTRICA. **Resolução Normativa nº 800 de 19 de dezembro de 2017**. [S.l.], 2017. 19 p. Disponível em: <http://www2.aneel.gov.br/cedoc/ren2017800.pdf2>. Acesso em: 19 jun. 2019.

AGHABOZORGI, S.; SHIRKHORSHIDI, A. S.; WAH, T. Y. Time-series clustering–a decade review. **Information Systems**, Elsevier, v. 53, p. 16–38, 2015.

AL-MESSABI, N. *et al.* Forecasting of photovoltaic power yield using dynamic neural networks. In: IEEE. **The 2012 International Joint Conference on Neural Networks (IJCNN)**. [S.l.], 2012. p. 1–5.

AMJADY, N.; KEYNIA, F. Short-term load forecasting of power systems by combination of wavelet transform and neuro-evolutionary algorithm. **Energy**, Elsevier, v. 34, n. 1, p. 46–57, 2009.

ANDRADE, L. C. M. d. **Transformada Wavelet e técnicas de inteligência computacional aplicadas à identificação, compressão e armazenamento de sinais no contexto de qualidade da energia elétrica**. Tese (Doutorado) — Universidade de São Paulo, 2017.

ARTHUR, D.; VASSILVITSKII, S. k-means++: The advantages of careful seeding. In: SOCIETY FOR INDUSTRIAL AND APPLIED MATHEMATICS. **Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms**. [S.l.], 2007. p. 1027–1035.

BARBOSA, E. *et al.* Aplicação de redes neurais artificiais para o reconhecimento de assinaturas harmônicas de equipamentos eletromédicos. **Revista de Sistemas e Computação-RSC**, v. 7, n. 1, 2017.

BARBOSA, E. H. C. *et al.* Critical analysis of pattern recognition load curves using multi-layer perceptron neural network. In: IEEE. **2018 13th IEEE International Conference on Industry Applications (INDUSCON)**. São Paulo-SP, 2018. p. 91–98.

BASHIR, A. Z.; EL-HAWARY, E. M. Applying wavelets to short-term load forecasting using pso-based neural networks. **IEEE transactions on power systems**, v. 20-27, n. 1, p. 3335–3343, 2009.

BECKER, S. Unsupervised learning procedures for neural networks. **International Journal of Neural Systems**, v. 1-2, p. 17–33, 1991.

BENTO, P. *et al.* Optimization of neural network with wavelet transform and improved data selection using bat algorithm for short-term load forecasting. **Neurocomputing**, Elsevier, 2019.

BISHOP, C. M. **Pattern recognition and machine learning**. [S.l.]: Springer, 2006.

BOX, G.; JENKINS, G.; REINSEL, G. **Time series analysis, forecasting and control. Englewood Clifs**. [S.l.]: NJ: Prentice-Hall, 1994.

BRADLEY, P. S.; FAYYAD, U. M. Refining initial points for k-means clustering. In: CITESEER. **ICML**. [S.l.], 1998. v. 98, p. 91–99.

BRANCO, H. M. G. C. **Uma estratégia para a detecção e classificação de transitórios em transformadores de potência pela utilização da transformada *Wavelet* e da lógica *Fuzzy***. Dissertação (Dissertação de Mestrado) — Universidade de São Paulo, 2009.

BROCKWELL, P. J.; DAVIS, R. A.; CALDER, M. V. **Introduction to time series and forecasting**. New York, NY: Springer-Verlag New York, 2002. v. 2.

BUITRAGO, J.; ASFOUR, S. Short-term forecasting of electric loads using nonlinear autoregressive artificial neural networks with exogenous vector inputs. **Energies**, Multidisciplinary Digital Publishing Institute, v. 10, n. 1, p. 40, 2017.

CHEN, Y. *et al.* Short-term load forecasting: Similar day-based wavelet neural networks. **IEEE Transactions on Power Systems**, IEEE, v. 25, n. 1, p. 322–330, 2009.

CHUI, C. K. **An introduction to wavelets (1992)**. San Diego: Academic Press, 2016.

COSTA, F. B. **Uma técnica de diagnóstico em tempo real de distúrbios transitórios baseada na transformada *wavelet* para uso em registradores digitais de perturbação**. Tese (Doutorado) — Universidade Federal de Campina Grande, Campina Grande - PB, 2010.

DAVIES, L.; GATHER, U. The identification of multiple outliers. **Journal of the American Statistical Association**, Taylor & Francis Group, v. 88, n. 423, p. 782–792, 1993.

DIMITRIADOU, E.; WEINGESSEL, A.; HORNIK, K. Voting-merging: An ensemble method for clustering. In: SPRINGER. **International Conference on Artificial Neural Networks**. [S.l.], 2001. p. 217–224.

DING, N. *et al.* Neural network-based model design for short-term load forecast in distribution systems. **IEEE transactions on power systems**, IEEE, v. 31, n. 1, p. 72–81, 2015.

DUDEK, G. Neural networks for pattern-based short-term load forecasting: A comparative study. **Neurocomputing**, Elsevier, v. 205, p. 64–74, 2016.

EKONOMOU, L. Greek long-term energy consumption prediction using artificial neural networks. **Energy**, Elsevier, v. 35, n. 2, p. 512–517, 2010.

FEINBERG, E. A.; GENETHLIOU, D. Load forecasting. In: **Applied mathematics for restructured electric power systems**. [S.l.]: Springer, 2005. p. 269–285.

FU, X. *et al.* Clustering-based short-term load forecasting for residential electricity under the increasing-block pricing tariffs in china. **Energy**, Elsevier, v. 165, p. 76–89, 2018.

GERS, F. A.; SCHMIDHUBER, J. Recurrent nets that time and count. In: IEEE. **Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium**. [S.l.], 2000. v. 3, p. 189–194.

GHAEMI, R. *et al.* A survey: clustering ensembles techniques. **World Academy of Science, Engineering and Technology**, v. 50, p. 636–645, 2009.

GHOSH, J.; ACHARYA, A. **Cluster Ensembles: Theory and Applications.** [S.l.]: Citeseer, 2013.

GONÇALVES, J. de A. **Algoritmo Híbrido para Pré-Processamento e Agrupamento de Curvas de Carga**. Dissertação (Dissertação de Mestrado) — Universidade Federal do Piauí, 2018.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. [S.l.]: MIT press, 2016.

GRAVES, A. Supervised sequence labelling with recurrent neural networks. 2012. **URL http://books. google. com/books**, 2012.

GRAVES, A.; SCHMIDHUBER, J. Framewise phoneme classification with bidirectional lstm and other neural network architectures. **Neural Networks**, Elsevier, v. 18, n. 5-6, p. 602–610, 2005.

GREFF, K. *et al.* Lstm: A search space odyssey. **IEEE transactions on neural networks and learning systems**, IEEE, v. 28, n. 10, p. 2222–2232, 2016.

GUPTA, J. M.; JIN, L.; HOMMA, N. **Static and dynamic neural networks: from fundamentals to advanced theory**. New Jersey: John Wiley and Sons, 2004.

HARTIGAN, J. A.; WONG, M. A. Algorithm as 136: A k-means clustering algorithm. **Journal of the Royal Statistical Society. Series C (Applied Statistics)**, JSTOR, v. 28, n. 1, p. 100–108, 1979.

HAVRDA, J.; CHARVÁT, F. Quantification method of classification processes. concept of structural a-entropy. **Kybernetika**, Institute of Information Theory and Automation AS CR, v. 3, n. 1, p. 30–35, 1967.

HAYKIN, S. **Neural networks: a comprehensive foundation**. New Jersey: Pretience-Hall 2nd edition, 1999.

HAYKIN, S. **Neural networks and learning machines**. New York: Pretience Hall, 2009. v. 3.

HEBB, D. O. **The organization of behavior: a neuropsychological theory**. New York, NY, USA: Willey, 1949.

HEIZER, J. H.; RENDER, B. **Operations management**. [S.l.]: Pearson Education India, 2008. v. 1.

HIPEL, K. W.; MCLEOD, A. I. **Time series modelling of water resources and environmental systems**. [S.l.]: Elsevier, 1994. v. 45.

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**, IEEE, v. 16, n. 1, p. 44–55, 2001.

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**, IEEE, v. 16, n. 1, p. 44–55, 2001.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.

HRUSCHKA, E. R. *et al.* A survey of evolutionary algorithms for clustering. **IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, IEEE, v. 39, n. 2, p. 133–155, 2009.

HSU, Y.-Y.; HO, K.-L. Fuzzy expert systems: an application to short-term load forecasting. In: IET. **IEE Proceedings C (Generation, Transmission and Distribution)**. [S.l.], 1992. v. 139, n. 6, p. 471–477.

HUBER, P. J. **Robust statistics**. [S.l.]: Springer, 2011.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. **arXiv preprint arXiv:1502.03167**, 2015.

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: a review. **ACM computing surveys (CSUR)**, Acm, v. 31, n. 3, p. 264–323, 1999.

JANG, D.-H. *et al.* Developing neural network models for early detection of cardiac arrest in emergency department. **The American journal of emergency medicine**, Elsevier, 2019.

JENSEN, A.; COUR-HARBO, A. **Ripples in mathematics: the discrete wavelet transform**. UK: Springer, 2001.

JIANG, Z. *et al.* A fused load curve clustering algorithm based on wavelet transform. **IEEE Transactions on Industrial Informatics**, IEEE, v. 14, n. 5, p. 1856–1865, 2018.

JIAXI, Y. *et al.* Research on time process-oriented power system static security analysis. In: IEEE. **2008 Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies**. [S.l.], 2008. p. 1516–1521.

KAGAN, N.; OLIVEIRA, C. C. B. d.; ROBBA, E. J. **Introdução aos sistemas de distribuição de energia elétrica**. [S.l.]: Edgard Blücher, 2005.

KATSATOS, A.; MOUSTRIS, K. Application of artificial neuron networks as energy consumption forecasting tool in the building of regulatory authority of energy, athens, greece. **Energy Procedia**, Elsevier, v. 157, p. 851–861, 2019.

KONG, W. *et al.* Short-term residential load forecasting based on lstm recurrent neural network. **IEEE Transactions on Smart Grid**, IEEE, 2018.

LEI, M. *et al.* A review on the forecasting of wind speed and generated power. **Renewable and Sustainable Energy Reviews**, Elsevier, v. 13, n. 4, p. 915–920, 2009.

LI, R.; LI, F.; SMITH, N. D. Multi-resolution load profile clustering for smart metering data. **IEEE Transactions on Power Systems**, IEEE, v. 31, n. 6, p. 4473–4482, 2016.

LIAO, T. W. Clustering of time series data—a survey. **Pattern recognition**, Elsevier, v. 38, n. 11, p. 1857–1874, 2005.

LIN, S. *et al.* Clustering load profiles for demand response applications. **IEEE Transactions on Smart Grid**, IEEE, 2017.

LINDEN, R. Técnicas de agrupamento. **Revista de Sistemas de Informação da FSMA**, v. 4, p. 18–36, 2009.

LIU, H.; SHAH, S.; JIANG, W. On-line outlier detection and data cleaning. **Computers & chemical engineering**, Elsevier, v. 28, n. 9, p. 1635–1647, 2004.

MA, L.; ZHOU, S.; LIN, M. Support vector machine optimized with genetic algorithm for short-term load forecasting. In: IEEE. **2008 International Symposium on Knowledge Acquisition and Modeling**. [S.l.], 2008. p. 654–657.

MACCULOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **Bulletin of mathematical biology**, v. 52, p. 99–115, 1990.

MAGALHÃES, M. N.; LIMA, A. C. P. de. **Noções de probabilidade e estatística**. [S.l.]: Editora da Universidade de São Paulo, 2002. v. 5.

MARES, J. J.; MERCADO, K. D. *et al.* A methodology for short-term load forecasting. **IEEE Latin America Transactions**, IEEE, v. 15, n. 3, p. 400–407, 2017.

MATHWORKS. **Hampel - Outlier removal using Hampel identifier**. 2015. Disponível em: <https://www.mathworks.com/help/signal/ref/hampel.html#>. Acesso em: 10 jul. 2019.

MATHWORKS. **Wavelet signal denoising**. 2017. Disponível em: <https://www.mathworks.com/help/wavelet/ref/wdenoise.html>. Acesso em: 7 jul. 2019.

MENEZES JÚNIOR, J. M. P.; BARRETO, G. A. Long-term time series prediction with the narx network: An empirical evaluation. **Neurocomputing**, v. 71, n. 16-18, p. 3335–3343, 2008.

MENEZES JÚNIOR, J. M. P. de. **Redes Neurais Dinâmicas para Predição e Modelagem Não-linear de Séries Temporais**. Dissertação (Dissertação de Mestrado) — Universidade Federal do Ceará, Fortaleza - CE, 2006.

MENEZES JÚNIOR, J. M. P. de. **Contribuições ao problema de predição recursiva de séries temporais univariadas usando redes neurais recorrentes.** Tese (Doutorado) — Universidade Federal do Ceará, Fortaleza - CE, 2012.

METS, K.; DEPUYDT, F.; DEVELDER, C. Two-stage load pattern clustering using fast wavelet transformation. **IEEE Transactions on Smart Grid**, IEEE, v. 7, n. 5, p. 2250–2259, 2016.

MILLIGAN, G. W. A monte carlo study of thirty internal criterion measures for cluster analysis. **Psychometrika**, Springer, v. 46, n. 2, p. 187–199, 1981.

MONTGOMERY, D. C.; JOHNSON, L. A.; GARDINER, J. S. **Forecasting and time series analysis**. [S.l.]: McGraw-Hill Companies, 1990.

MORDJAOUI, M. *et al.* Electric load forecasting by using dynamic neural network. **International Journal of Hydrogen Energy**, Elsevier, v. 42, n. 28, p. 17655–17663, 2017.

MORDJAOUI, M. *et al.* Electric load forecasting by using dynamic neural network. **International Journal of Hydrogen Energy**, Elsevier, v. 42, n. 28, p. 17655–17663, 2017.

MORETTIN, P. A.; TOLOI, C. **Análise de séries temporais**. 2. ed. São Paulo: Egard Blucher, 2006.

NAGI, J. *et al.* A computational intelligence scheme for the prediction of the daily peak load. **Applied Soft Computing**, Elsevier, v. 11, n. 8, p. 4773–4788, 2011.

NAGPAL, A.; JATAIN, A.; GAUR, D. Review based on data clustering algorithms. In: IEEE. **2013 IEEE Conference on Information & Communication Technologies**. [S.l.], 2013. p. 298–303.

NORVIG, P.; RUSSELL, S. **Inteligência Artificial: Tradução da 3a edição**. Brasil: Elsevier, 2014.

OLAH, C. **Understanding LSTM Networks**. 2015. Disponível em: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. Acesso em: 01 jun. 2019.

OLIVEIRA, A. d.; SILVEIRA, G. B. d.; BRAGA, J. d. M. Diversidade sazonal do consumo de energia elétrica no brasil. Instituto de Pesquisa Econômica Aplicada (Ipea), 2000.

OLIVEIRA, L. A. A. de. **Tratamento de dados de curvas de carga via análise de agrupamentos e transformada *wavelets***. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, 2013.

OPERADOR NACIONAL DO SISTEMA - ONS. **Procedimentos de Rede**: Submódulo 5.1 - consolidação da previsão de carga: visão geral. [S.l.], 2009. 9 p. Disponível em: <http://ons.org.br/paginas/sobre-o-ons/procedimentos-de-rede/vigentes>. Acesso em: 05 jun. 2019.

OPERADOR NACIONAL DO SISTEMA - ONS. **Procedimentos de Rede**: Submódulo 5.4 - consolidação da previsão de carga para a programação diária da operação eletroenergética e para a programação de intervenções em instalações da rede de operação. [S.l.], 2010. 10 p. Disponível em: <http://ons.org.br/paginas/sobre-o-ons/procedimentos-de-rede/vigentes>. Acesso em: 05 jun. 2019.

PANAPAKIDIS, I.; ALEXIADIS, M.; PAPAGIANNIS, G. Evaluation of the performance of clustering algorithms for a high voltage industrial consumer. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 38, p. 1–13, 2015.

PANAPAKIDIS, I. P. Application of hybrid computational intelligence models in short-term bus load forecasting. **Expert Systems with Applications**, Elsevier, v. 54, p. 105–120, 2016.

PANAPAKIDIS, I. P. Clustering based day-ahead and hour-ahead bus load forecasting models. **International Journal of Electrical Power & Energy Systems**, Elsevier, v. 80, p. 171–178, 2016.

PANAPAKIDIS, I. P.; ALEXIADIS, M. C.; PAPAGIANNIS, G. K. Deriving the optimal number of clusters in the electricity consumer segmentation procedure. In: IEEE. **2013 10th International Conference on the European Energy Market (EEM)**. [S.l.], 2013. p. 1–8.

PANDEY, A. S.; SINGH, D.; SINHA, S. K. Intelligent hybrid wavelet models for short-term load forecasting. **IEEE Transactions on Power Systems**, IEEE, v. 25, n. 3, p. 1266–1273, 2010.

PARK, J. *et al.* The efficient incorporation of mlp features into automatic speech recognition systems. **Computer Speech & Language**, Elsevier, v. 25, n. 3, p. 519–534, 2011.

PEARSON, R. K. Outliers in process modeling and identification. **IEEE Transactions on control systems technology**, IEEE, v. 10, n. 1, p. 55–63, 2002.

PEARSON, R. K. *et al.* Generalized hampel filters. **EURASIP Journal on Advances in Signal Processing**, Springer, v. 2016, n. 1, p. 87, 2016.

PRINCIPE, J. C.; EULIANO, N. R.; LEFEBVRE, W. C. **Neural and adaptive systems: fundamentals through simulations**. New York: Wiley, 2000.

QING, X.; NIU, Y. Hourly day-ahead solar irradiance prediction using weather forecasts by lstm. **Energy**, Elsevier, v. 148, p. 461–468, 2018.

RAJARAMAN, A.; ULLMAN, J. D. **Mining of massive datasets**. [S.l.]: Cambridge University Press, 2011.

RANA, M.; KOPRINSKA, I. Forecasting electricity load with advanced wavelet neural networks. **Neurocomputing**, Elsevier, v. 182, p. 118–132, 2016.

REDDY, C. K.; VINZAMURI, B. A survey of partitional and hierarchical clustering algorithms. In: **Data Clustering**. [S.l.]: Chapman and Hall/CRC, 2013. p. 87–110.

REIS, A. R.; SILVA, A. A. D. Feature extraction via multiresolution analysis for short-term load forecasting. **IEEE Transactions on power systems**, IEEE, v. 20, n. 1, p. 189–198, 2005.

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, v. 65, n. 6, p. 286–408, 1958.

SADAEI, H. J. *et al.* Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series. **Energy**, Elsevier, 2019.

SAGHEER, A.; KOTB, M. Time series forecasting of petroleum production using deep lstm recurrent networks. **Neurocomputing**, Elsevier, v. 323, p. 203–213, 2019.

SAHOO, H.; DASH, P.; RATH, N. Narx model based nonlinear dynamic system identification using low complexity neural networks and robust h filter. **Applied Soft Computing**, Elsevier, v. 13, n. 7, p. 3324–3334, 2013.

SAVIOZZI, M.; MASSUCCO, S.; SILVESTRO, F. Implementation of advanced functionalities for distribution management systems: Load forecasting and modeling through artificial neural networks ensembles. **Electric Power Systems Research**, Elsevier, v. 167, p. 230–239, 2019.

SILVA, I. N. da; SPATTI, D. H.; FLAUZINO, R. F. **Redes neurais artificiais para engenharia e ciências aplicadas**. São Paulo: Artliber, 2010.

SILVA, K.; SOUZA, B. A.; BRITO, N. S. Fault detection and classification in transmission lines based on wavelet transform and ann. **IEEE Transactions on Power Delivery**, IEEE, v. 21, n. 4, p. 2058–2063, 2006.

SILVEIRA, T. M. A. da. **Modelos de previsão de carga elétrica em curto prazo desenvolvidos com redes neurais artificiais e lógica Fuzzy considerando a variável temperatura**. Dissertação (Dissertação de Mestrado) — Universidade de São Paulo, 2010.

SINGH, P.; DWIVEDI, P. Integration of new evolutionary approach with artificial neural network for solving short term load forecast problem. **Applied energy**, Elsevier, v. 217, p. 537–549, 2018.

SOLIMAN, S. A.-h.; AL-KANDARI, A. M. **Electrical load forecasting: modeling and model construction**. [S.l.]: Elsevier, 2010.

SRIVASTAVA, A.; PANDEY, A. S.; SINGH, D. Short-term load forecasting methods: A review. In: IEEE. **2016 International Conference on Emerging Trends in Electrical Electronics & Sustainable Energy Systems (ICETEESES)**. [S.l.], 2016. p. 130–138.

TIELEMAN, T.; HINTON, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. **COURSERA: Neural networks for machine learning**, v. 4, n. 2, p. 26–31, 2012.

VARGAS, L.; PAREDES, G.; BUSTOS, G. Data mining techniques for very short term prediction of wind power. In: IEEE. **2010 IREP Symposium Bulk Power System Dynamics and Control-VIII (IREP)**. [S.l.], 2010. p. 1–7.

VINH, N. X.; EPPS, J. mincentropy: A novel information theoretic approach for the generation of alternative clusterings. In: IEEE. **2010 IEEE International Conference on Data Mining**. [S.l.], 2010. p. 521–530.

WAIBEL, A. Modular construction of time-delay neural networks for speech recognition. **Neural computation**, v. 1, n. 1, p. 39–46, 1989.

WANG, Y. *et al.* Probabilistic individual load forecasting using pinball loss guided lstm. **Applied Energy**, Elsevier, v. 235, p. 10–20, 2019.

WEISSBACH, T.; WELFONDER, E. High frequency deviations within the european power system: Origins and proposals for improvement. In: IEEE. **2009 IEEE/PES Power Systems Conference and Exposition**. [S.l.], 2009. p. 1–6.

WINSTON, W. L.; GOLDBERG, J. B. **Operations research: applications and algorithms**. [S.l.]: Thomson Brooks/Cole, 2004. v. 3.

WU, W.; PENG, M. A data mining approach combining *k*-means clustering with bagging neural network for short-term wind power forecasting. **IEEE Internet of Things Journal**, IEEE, v. 4, n. 4, p. 979–986, 2017.

YANG, Y. **Temporal Data Mining Via Unsupervised Ensemble Learning**. [S.l.]: Elsevier, 2016.

ZHUANG, L. *et al.* Comparison of forecasting methods for power system short-term load forecasting based on neural networks. In: IEEE. **2016 IEEE International Conference on Information and Automation (ICIA)**. [S.l.], 2016. p. 114–119.

ZOR, K.; TIMUR, O.; TEKE, A. A state-of-the-art review of artificial intelligence techniques for short-term electric load forecasting. In: IEEE. **2017 6th International Youth Conference on Energy (IYCE)**. [S.l.], 2017. p. 1–7.