

UNIVERSIDADE ESTADUAL DO OESTE DO PARANÁ

CAMPUS DE FOZ DO IGUAÇU

PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA E COMPUTAÇÃO

DISSERTAÇÃO DE MESTRADO

**REDES NEURAIS ARTIFICIAS BIGRU_CNN
APLICADAS À PREVISÃO DE DEMANDA DE ENERGIA
ELÉTRICA DE CURTO PRAZO**

LUCAS DUARTE SOARES

FOZ DO IGUAÇU

2021

Lucas Duarte Soares

Redes Neurais Artificias BiGRU_CNN Aplicadas à Previsão de Demanda de Energia Elétrica de Curto Prazo

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Computação da Universidade Estadual do Paraná como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica e Computação. Área de concentração: Sistemas Elétricos e Computação.

Orientador: Edgar Manuel Carreño Franco

Foz do Iguaçu
2021

Ficha de identificação da obra elaborada através do Formulário de Geração Automática do Sistema de Bibliotecas da Unioeste.

Soares, Lucas Duarte
Redes Neurais Artificiais BiGRU_CNN Aplicadas à Previsão de
Demanda de Energia Elétrica de Curto Prazo / Lucas Duarte
Soares; orientador Edgar Manuel Carreño Franco. -- Foz do
Iguaçu, 2021.
118 p.

Dissertação (Mestrado Acadêmico Campus de Foz do Iguaçu) --
Universidade Estadual do Oeste do Paraná, Centro de
Engenharias e Ciências Exatas, Programa de Pós-Graduação em
Engenharia Elétrica e Computação, 2021.

1. Engenharia Elétrica. 2. Inteligência Artificial. I.
Franco, Edgar Manuel Carreño, orient. II. Título.

Redes Neurais Artificias BiGRU_CNN Aplicadas à Previsão de Demanda de Energia Elétrica de Curto Prazo

Lucas Duarte Soares

Esta Dissertação de Mestrado foi apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Computação e aprovada pela Banca Examinadora assim constituída:

Prof. Dr. **Edgar Manuel Carreño Franco** - (Orientador)

Universidade Estadual do Oeste do Paraná - UNIOESTE

Prof. Dr. **Renato Bobsin**

Universidade Estadual do Oeste do Paraná - UNIOESTE

Prof. Dr. **Marcos Ricardo Müller**

Faculdade de Ensino Superior de São Miguel do Iguaçu - FAESI

Data da defesa: 03/09/2021.

Resumo

O presente trabalho analisou o comparativo entre redes neurais artificiais *feedforwards*, recorrentes, convolucionais e bidirecionais baseadas em camadas de diferentes arquiteturas como ferramenta preditiva de demanda de energia elétrica de curto prazo. Esses modelos de previsões podem servir como instrumento de apoio relacionados à tomada de decisão de empresas do setor energético, em virtude da demanda de energia ser estabelecida um dia antes de sua transmissão em boa parte do mundo. O código das redes neurais artificiais foi programado em *Python* fazendo uso do pacote *Keras*. As previsões de todas as redes foram realizadas 10 vezes até se chegar a uma amostra estatística aceitável para que valores futuros de demanda de energia sejam os mais próximos possíveis da realidade. O melhor modelo de previsão foi o da rede BiGRU_CNN, onde os erros médios atribuídos às suas previsões em um horizonte de 24 horas foram de 3,42% para o erro MAPE, 100,75 MW para a métrica de acurácia MAE e 122,2 MW para o erro RMSE.

Palavras-chave: Inteligência Artificial, Aprendizado de Máquina, Redes Neurais Recorrentes, Séries Temporais.

Abstract

The present work analyzed the comparison between feedforwards, recurrent, convolutional and bidirectional artificial neural networks based on different layers architectures as a predictive tool for short-term load forecasting. These forecasting models can serve as a support instrument related to the decision making of companies in the energy sector, as the demand for energy is requested one day before its transmission in much of the world. The code of the artificial neural networks was programmed in Python using the Keras package. Forecasts for all networks have been performed 10 times until an acceptable statistical sample is reached so that future values demand for energy are as close as possible to reality. The best forecasting model was the BiGRU_CNN network where the average errors attributed to its predictions in a 24-hour horizon was 3.42% for the MAPE error, 100.75 MW for the MAE accuracy metric and 122.2 MW for the RMSE error.

Keywords: Artificial Intelligence, Machine Learning, Recurrent Neural Network, Time Series.

Este trabalho é inteiramente dedicado à minha mãe, Edina,
pelo apoio e incentivo incondicional em novas conquistas.

Agradecimentos

Para essa dissertação de mestrado atingir o presente resultado houve a colaboração e apoio fundamental de diversas pessoas.

Quero agradecer, inicialmente, ao meu orientador, Professor Doutor Edgar Manuel Carreño Franco, pelas correções sempre que necessárias; além da paciência, competência e maestria com que sempre me orientou ao longo deste trabalho o qual não seria realizado sem sua presença, muito obrigado.

Agradeço aos Professores Doutores do departamento de matemática da UTFPR, onde tudo começou. Jairo Marlon Corrêa, Samuel Bellido Rodrigues e Tássia Hickmann que gentilmente aceitaram minha participação no projeto inicial, pelos trabalhos em conjuntos, por todo conhecimento compartilhado e pelas conversas e conselhos quando nos reuníamos em Medianeira.

Ao Professor Doutor Levi Lopes Teixeira um dos responsáveis pelo meu ingresso no projeto inicial, obrigado por ter sido o primeiro a acreditar em mim, sem o seu apoio e incentivo tenho certeza que não chegaria neste ponto.

Gostaria de agradecer a todos os meus colegas do PGEEC da UNIOESTE onde a colaboração e o companheirismo estiveram presentes em todos os momentos, desde as aulas até a finalização deste trabalho.

Por último, agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e ao Centro de Estudos Avançados em Segurança de Barragens (CEASB), pelo apoio financeiro do presente trabalho.

"O sucesso é ir de fracasso em fracasso sem perder entusiasmo."

Winston Churchill

Sumário

Lista de Figuras	11
Lista de Símbolos	13
Lista de Siglas e Abreviaturas	14
1 Contexto	15
1.1 Objetivos	17
1.1.1 Objetivo Geral	17
1.1.2 Objetivo Específicos	17
1.2 Estrutura do Trabalho	17
2 Referencial Teórico	18
2.1 Neurônio Biológico	18
2.2 Neurônio Artificial	20
2.3 Rede Neural Artificial	22
2.4 Aprendizado de Redes Neurais <i>Feedforward</i>	25
2.5 Rede Neural Recorrente	28
2.6 Rede Neural LSTM	31
2.6.1 <i>Vanishing Gradient</i> e a Rede LSTM	32
2.7 Rede Neural GRU	34
2.8 Rede Neural CNN	37
2.9 Revisão Bibliográfica	39
3 Materiais e Métodos	53
3.1 Séries Temporais	53
3.2 Metodologia	54
3.2.1 Previsão de Demanda de Energia de Curto Prazo	54
3.2.2 Código das Redes	56
4 Resultados e Discussões	60

5 Conclusão	69
Referências Bibliográficas	71
A Códigos em <i>Python</i>	79
B Gráficos das IMFs da Decomposição CEENDAM	84
C Artigo Aceito para Publicação na Revista <i>Production</i>	91
D Artigo Submetido à <i>Journal of Control Automation and Electrical Systems</i>	106

Lista de Figuras

Figura 2.1: Neurônio biológico com seus componentes principais	19
Figura 2.2: Diagrama esquemático simplificado de dois neurônios biológicos	19
Figura 2.3: Neurônio artificial	20
Figura 2.4: Transformação afim	21
Figura 2.5: Funções de ativação	22
Figura 2.6: Rede neural artificial	23
Figura 2.7: <i>Perceptron</i>	24
Figura 2.8: Exemplos de classificação perceptron	25
Figura 2.9: Problemas linearmente inseparáveis	25
Figura 2.10: Neurônio alimentado por vários sinais de entrada	26
Figura 2.11: Rede neural recorrente	28
Figura 2.12: Rede neural recorrente desdobrada no tempo	29
Figura 2.13: <i>Vanishing gradient</i> em redes recorrentes simples	32
Figura 2.14: Diagrama de uma célula LSTM	33
Figura 2.15: Preservação de informações de gradiente por redes LSTM	34
Figura 2.16: Unidade estrutural da rede neural artificial GRU	35
Figura 2.17: Unidade estrutural da rede neural BiGRU	37
Figura 2.18: Estrutura da rede neural CNN	38
Figura 2.19: Processo convolucional a partir de uma entrada 1D	38
Figura 3.1: Visão geral da série histórica da demanda de energia intervalada em 1 hora	54
Figura 3.2: Fluxograma das etapas realizadas na metodologia	59
Figura 4.1: Comparação dos erros das médias das redes	60
Figura 4.2: Comparação do modelo Ceemdan-LSTM-GRU com variação nas épocas	63
Figura 4.3: Tempo de simulação das redes neurais artificiais	65
Figura 4.4: Previsões efetivas de demanda de energia de curto prazo de todas as simulações referentes à rede neural BiGRU_CNN	66
Figura 4.5: Comparação dos erros das simulações da rede BiGRU_CNN	67
Figura 4.6: Previsões acompanhadas dos erros MAPE, MAE e MSE na amostra de teste da rede BiGRU_CNN da segunda simulação	68
Figura A.1: Primeira parte do código da rede BiGRU_CNN	79

Figura A.2: Segunda parte do código da rede BiGRU_CNN	80
Figura A.3: Terceira parte do código da rede BiGRU_CNN	81
Figura A.4: Quarta parte do código da rede BiGRU_CNN	82
Figura A.5: Quinta parte do código da rede BiGRU_CNN	83
Figura A.6: Código do gráfico das previsões efetivas	83
Figura B.1: IMF 2	84
Figura B.2: IMF 3	85
Figura B.3: IMF 4	85
Figura B.4: IMF 5	86
Figura B.5: IMF 6	86
Figura B.6: IMF 7	87
Figura B.7: IMF 8	87
Figura B.8: IMF 9	88
Figura B.9: IMF 10	88
Figura B.10: IMF 11	89
Figura B.11: IMF 12	89
Figura B.12: IMF 13	90

Lista de Símbolos

x_i	i -ésimo sinal de entrada da rede neural
w_{ik}	i -ésimo peso sináptico do neurônio k
b_k	Bias do neurônio k
v_k	Campo induzido do neurônio k
ϕ	Função de ativação
y_k	Saída do neurônio k
u_k	Sinais de entrada ponderados do neurônio k
y_j	Saída do neurônio j
d_j	Saída desejada do neurônio j
e_j	Erro entre a saída e saída desejada do neurônio j
n	Intervalo de tempo na aplicação do algoritmo
m	Número total de entradas aplicadas ao neurônio j
ξ	Função de perda da rede neural
Δw_{ij}	Correção do peso sináptico correspondente entre o neurônio i e j
η	Parâmetro de taxa de aprendizado
x_i^t	Entrada i no instante de tempo t
a_j^t	Entrada no neurônio j no instante de tempo t
b_j^t	Ativação do neurônio j no instante de tempo t
θ_h	Função de ativação diferenciável e não linear
δ_j	Derivada parcial da função de perda diferenciável em relação a saída da rede
x_t	i Entrada da célula LSTM
i_t	<i>Input gate</i> da célula LSTM
f_t	<i>Forget gate</i> da célula LSTM
o_t	<i>Output gate</i> da célula LSTM
\tilde{c}_t	<i>Candidate state</i> da célula LSTM
c_t	<i>Cell state</i> da célula LSTM
h_t	<i>Output</i> da célula LSTM
σ	Função de ativação a <i>sigmoid</i>
W	Peso sináptico atribuído a entrada da célula LSTM
U	Pesos sinápticos das conexões recorrentes da rede LSTM
h	Horizonte de previsão
ψ	Rede treinada e validada

Lista de Siglas e Abreviaturas

UNIOESTE	Universidade Estadual do Oeste do Paraná
LSTM	<i>Long Short-Term Memory</i>
MAPE	<i>Mean Absolute Percent Error</i>
MAE	<i>Mean Absolute Error</i>
RMSE	<i>Root Mean Square Error</i>
UTFPR	Universidade Tecnológica Federal do Paraná
PGEEC	Programa de Pós-Graduação em Engenharia Elétrica e Computação
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CEASB	Centro de Estudos Avançados em Segurança de Barragens
MLP	<i>Multilayer Perceptron</i>
BPTT	<i>Backpropagation Through Time</i>
tanh	Tangente hiperbólica
ARIMA	<i>Autoregressive Integrated Moving Average</i>
kV	kiloVolt
GWh	<i>Gigawatt hours</i>
BP	<i>BackPropagation</i>
RPE	<i>Recursive Prediction Error</i>
MRPE	<i>Modified Recursive Prediction Error</i>
ALRPE	<i>Adaptive Learning Recursive Prediction Error</i>
kW	kiloWatts
MSE	<i>Mean Squared Error</i>
RNN	<i>Recurrent Neural Network</i>
SVR	<i>Support Vector Regressive</i>
GRU	<i>Gated Recurrent Unit</i>
iForest	<i>Isolation Forest</i>
GRNN	<i>Generalized Regression Neural Network</i>
ELM	<i>Extreme Learning Machine</i>
EMD	<i>Empirical Mode Decomposition</i>
PSO	<i>Particle Swarm Optimization</i>
Adam	<i>Adaptive Moment Estimation</i>
ReLU	<i>Rectified Linear Unit</i>

Capítulo 1

Contexto

O desenvolvimento econômico ao redor do planeta depende, quase exclusivamente, da disponibilidade de energia elétrica nas indústrias, pois a maioria delas a utiliza para efetuar suas atividades produtivas vitais (Soliman & Al-Kandari, 2010). Logo a previsão de demanda de energia elétrica é uma ferramenta de apoio à tomada de decisões que deve ser levada em consideração por empresas do setor elétrico para garantir um serviço eficiente (Hahn, Meyer-Nieberg & Pickl, 2009), visto que a falta de eletricidade possui impacto direto na economia e saúde financeira em todo o mundo.

Quando o assunto é previsão de demanda de energia elétrica, existem quatro tipos de horizontes de previsões para atingir diferentes objetivos de planejamento e, também, para auxiliar no monitoramento de condições críticas do sistema elétrico de empresas (Setiawan, Koprinska & Agelidis, 2009). Os horizontes de previsões supracitados, ainda nas palavras de Setiawan et al. (2009), podem ser classificados em: curtíssimo prazo; curto prazo; médio prazo e longo prazo.

As previsões de curtíssimo prazo de demanda de energia fornecem valores futuros entre um minuto e uma hora visando determinar a melhor estratégia para o uso de recursos durante a geração de energia (Charytoniuk & Chen, 2000). Já no curto prazo as previsões são realizadas entre uma hora e uma semana para auxiliar no planejamento operacional, visto que a demanda de energia é definida um dia antes de sua transmissão em grande parte do mundo (Chapagain, Kittipiyakul & Kulthanavit, 2020). Em relação ao médio prazo as previsões são realizadas entre uma semana e um mês com o objetivo de buscar maiores lucros no mercado de energia elétrica por parte das empresas atuantes do ramo (Pan & Lee, 2012). Enquanto as previsões de longo prazo, acima de um ano, são utilizadas como ferramenta de apoio no dimensionamento de novas instalações de empresas de geração, transmissão e distribuição de energia elétrica (Kandil, El-Debeiky & Hasanien, 2002).

Sabendo da grande importância das previsões de demanda de energia para o setor elétrico, de acordo com Ghalekhondabi, Ardjmand, Weckman & Young (2017), artigos publicados relacionados ao tema cresceram exponencialmente nos últimos anos. Esses trabalhos utilizam técnicas preditivas que, segundo Singh & Khatoun (2013), podem ser divididas em três grupos majoritários, sendo eles: modelos tradicionais, modelos tradicionais modificados e modelos

computacionais. Os dois primeiros grupos possuem modelos de predições lineares, consequentemente, nas palavras de Carpinteiro & Alves Da Silva (2000), esses apresentam dificuldades na previsão da demanda de energia uma vez que sua relação com variáveis exógenas é complexa e não linear. Já o terceiro grupo, também conhecido como modelos de inteligência artificial, segundo Singh & Khatoon (2013), ganhou notoriedade por desempenho satisfatório nesses cenários onde os modelos lineares apresentam certa dificuldade.

Alguns modelos baseados em inteligência artificial são utilizados para efetuar previsão de demanda de energia, sendo eles, de acordo com Shahidehpour, Yamin & Li (2002): sistemas especialistas; computação evolutiva; sistemas *fuzzy*; redes neurais artificiais e modelos híbridos. Em relação a estes modelos, as redes neurais artificiais tem recebido maior atenção pelo fato dos seus modelos serem mais precisos, serem de fácil implementação e possuírem boa performance (Shahidehpour et al., 2002). Por causa dessas especificidades é encontrado um número considerável de trabalhos científicos que utilizam esses modelos neurais preditivos para estimar a necessidade energética.

As redes recorrentes (RNN) são um tipo peculiar de redes neurais artificiais que tornaram-se o foco de muitos estudos devido sua capacidade de processar informações sequenciais e temporais (Medsker & Jain, 2001). Segundo Hagan, Demuth & Beale (2014), as redes neurais recorrentes são potencialmente mais poderosas que as redes neurais *feedforward*, por exemplo a *multilayer perceptron* (MLP). Todavia essas redes apresentam dificuldades no treinamento devido ao fenômeno do *vanishing* gradiente que acarreta em resultados não satisfatórios (Hagan et al., 2014). Para solucionar esse problema é utilizado arquiteturas recorrentes baseadas em portas como a *Long Short-Term Memory* (LSTM) e *Gated Recurrent Unit* (GRU), no qual podem explorar informações de longo prazo (Abbasimehr, Shabani & Yousefi, 2020).

Em vista da possibilidade de utilizar-se valores futuros da necessidade energética como ferramenta de apoio à tomada de decisões, o presente trabalho visa avaliar a previsão de demanda de energia elétrica de uma empresa do setor elétrico com os seguintes modelos propostos baseados em diferentes camadas de redes neurais artificiais: BiLSTM-BiGRU, BiGRU-BiLSTM, CNN-BiGRU, BiGRU-CNN e Ceemdan-LSTM-GRU. Essa avaliação é necessária para indicar em qual condição os resultados preditivos são melhores, pois a precisão da previsão da demanda de energia pode afetar os custos e receitas para geradores de energia elétrica e operadoras de transmissão ou distribuição e, portanto, a lucratividade e sustentabilidade dessas organizações (Islam, Che, Hasanuzzaman & Rahim, 2019). Desse modo os modelos neurais preditivos de camadas distintas propostos foram comparados com as redes neurais artificiais clássicas MLP, RNN, GRU e LSTM para verificar se suas previsões são mais precisas. No qual a série histórica de demanda de energia elétrica da respectiva empresa foi utilizada como alimentação nos modelos neurais para efetuar as previsões de demanda de energia de curto prazo.

1.1 Objetivos

1.1.1 Objetivo Geral

Verificar se redes neurais artificiais recorrentes construídas a partir de camadas de diferentes arquiteturas fornecem resultados mais precisos que redes clássicas nas previsões de demanda de energia elétrica no curto prazo de uma empresa do setor elétrico.

1.1.2 Objetivo Específicos

- Coletar dados temporais disponíveis relacionados à demanda de energia elétrica;
- Analisar a viabilidade da técnica Ceemdan aplicada à série histórica energética;
- Comparar os resultados preditivos das redes estruturadas em diferentes tipos camadas.

1.2 Estrutura do Trabalho

O presente trabalho está segmentado em seis capítulos. O Capítulo 1, este de introdução, apresenta o contexto do problema, justificativas e objetivos que serviram como motivação para a realização desta dissertação. O Capítulo 2 discorre sobre o referencial teórico que engloba conceitos básicos sobre redes neurais artificiais, dentre esses conceitos estão: surgimento histórico; motivação de pesquisa; aplicações; arquiteturas; treinamento e, por último, vantagens e desvantagens. Além disso, no mesmo Capítulo são descritos alguns trabalhos relacionados à previsão de demanda de energia dos últimos 11 anos (2010 até 2021); dando ênfase nos trabalhos que aplicam redes neurais em geral, e finalizando com os que aplicam redes de arquitetura recorrentes. No tocante ao Capítulo 3 é apresentado detalhadamente a série temporal e os métodos utilizados neste estudo para a obtenção dos resultados preditivos de demanda de energia de curto prazo. Com a finalidade de verificar e comparar o desempenho preditivo de curto prazo das redes neurais propostas neste trabalho, o Capítulo 4 foi escrito. E, por último, mas não menos importante, o Capítulo 5 foi redigido com o objetivo de apresentar as considerações finais do presente estudo, bem como a motivação para efetuar trabalhos futuros.

Capítulo 2

Referencial Teórico

Com o intuito de introduzir as redes de arquitetura recorrentes, inicialmente é apresentada uma conceituação básica sobre redes neurais artificiais. Dentre esses conceitos estão: surgimento histórico; motivação de pesquisa; aplicações práticas; arquiteturas; algoritmos de treinamento e, por último, vantagens e desvantagens das respectivas arquiteturas. Posteriormente à sucinta explicação teórica das redes neurais artificiais, são descritos artigos científicos dos últimos 11 anos (2010 até 2021) aplicados no que tange à previsão de demanda de energia elétrica, no qual mostram sua atual relevância como ferramenta de apoio à tomada de decisões nas empresas do setor elétrico. Esses trabalhos são citados com ênfase nos que aplicam redes neurais em geral, e finalizando com os que aplicam redes de arquitetura recorrentes.

2.1 Neurônio Biológico

Os neurônios biológicos são pequenos elementos interligados que constituem o cérebro humano, existindo entre 10^{10} a 10^{12} unidades dessas células no qual cada uma possui em torno de 10^4 a 10^5 conexões sinápticas (Harvey, 1994). Para o propósito de redes neurais artificiais, esses neurônios dispõem de três componentes básicos, sendo eles: o dendrito; o corpo celular e o axônio (Hagan et al., 2014). Segundo Harvey (1994), os dendritos são protuberâncias parecidas com galhos de árvores do corpo celular e os axônios são fibras extensoras do corpo celular. A Figura 2.1 ilustra o neurônio com os seus três (3) componentes principais.

Sinais adentram os neurônios através de sinapses localizadas principalmente nos dendritos neuronais; e, em menor proporção, no corpo celular (Guyton & Hall, 2005). Após isso o axônio transporta o sinal do corpo celular para os dendritos de outros neurônios (Harvey, 1994). O ponto que ocorre o contato entre um axônio de um neurônio e um dendrito de outro neurônio é chamado de sinapse (Hagan et al., 2014). A Figura 2.2 apresenta um diagrama esquemático simplificado de dois neurônios biológicos efetuando a sinapse.

O conjunto estruturado de neurônios biológicos formam as redes neurais biológicas que são, basicamente, nas palavras de Galushkin (2007), sistemas não lineares altamente complexos com graus elevados de liberdade que utilizam princípios de processamento de informações dife-

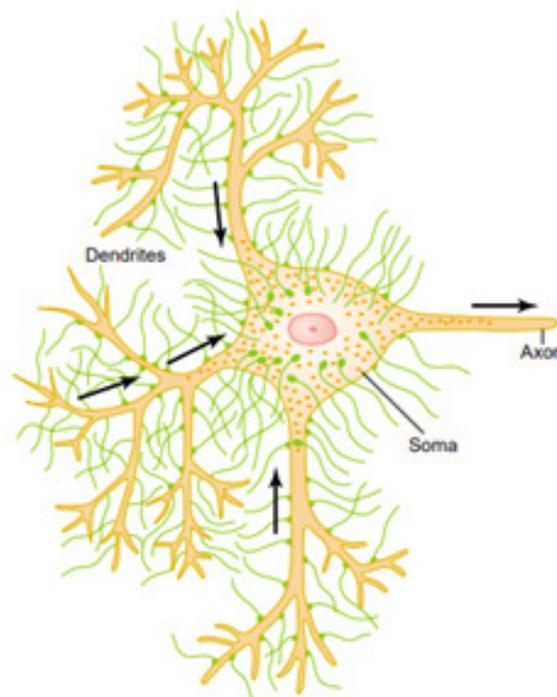


Figura 2.1: Neurônio biológico com seus componentes principais
 Fonte: (Guyton & Hall, 2005).

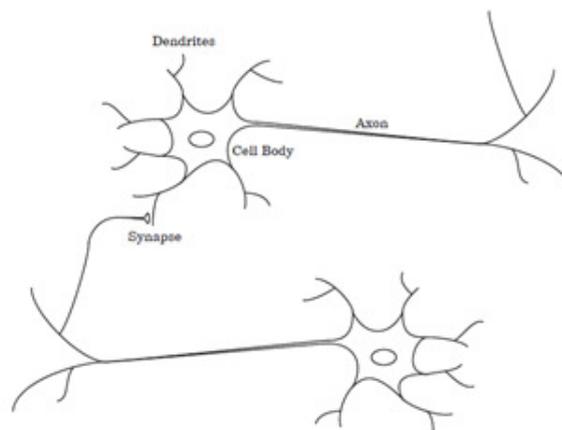


Figura 2.2: Diagrama esquemático simplificado de dois neurônios biológicos
 Fonte: (Hagan et al., 2014).

rentes daqueles de computadores comuns. Mesmo os circuitos elétricos sendo mais rápidos que os neurônios biológicos, o cérebro é capaz de realizar muitas tarefas em menos tempo do que qualquer computador convencional devido, em parte, sua enorme estrutura paralela onde todos os neurônios estão operando simultaneamente (Hagan et al., 2014). A rede neural artificial tem como base a mesma estrutura paralela de neurônios biológicos do cérebro, esse é um dos fatores que deriva o seu poder de computação (Haykin, 1999).

2.2 Neurônio Artificial

O neurônio é a menor unidade de processamento de informações de uma rede neural artificial e é fundamental para o seu funcionamento (Haykin, 1999). A Figura 2.3 mostra a representação matemática de um neurônio artificial onde os sinais de entrada x_1, x_2, \dots, x_m são multiplicados pelos seus respectivos pesos $w_{k1}, w_{k2}, \dots, w_{km}$. Essas entradas ponderadas ($x_1w_{k1}, x_2w_{k2}, \dots, x_mw_{km}$) mais o bias b_k são introduzidos no somador para resultarem na saída v_k que passa pela função de ativação ϕ para gerar a saída y_k do neurônio. De acordo com Haykin (1999), esse modelo de neurônio forma a base para projetar uma grande família de redes neurais.

Fazendo uma analogia entre um neurônio artificial e um neurônio biológico; os pesos sinápticos w correspondem as forças sinápticas, o corpo celular é representado pelo somador Σ junto com a função de ativação ϕ e, por último, a saída do neurônio y_k representa o sinal no axônio (Hagan et al., 2014).

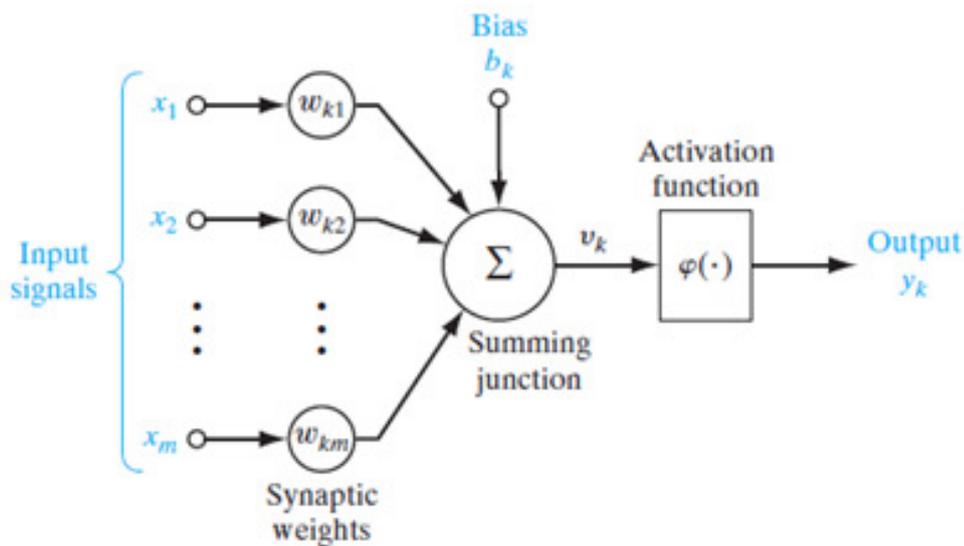


Figura 2.3: Neurônio artificial

Fonte: (Haykin, 1999).

As Equações (2.1) e (2.2) apresentam o comportamento matemático do neurônio artificial da Figura 2.3 (Haykin, 1999). A Equação (2.1) é o somatório das entradas ponderadas do neurônio e a Equação (2.2) é a saída do neurônio dada pelo somatório das entradas ponderadas mais o bias como parâmetros da função de ativação.

$$u_k = \sum_{j=1}^m w_{kj}x_j \quad (2.1)$$

$$y_k = \phi(u_k + b_k) \quad (2.2)$$

O bias tem o efeito de aumentar ou diminuir a entrada líquida da função de ativação e, ainda, aplicar uma transformação afim no somatório das entradas ponderadas do neurônio (Haykin, 1999). Ainda nas palavras de Haykin (1999), dependendo se o bias é positivo ou negativo, a relação entre o campo induzido local v_k -Equação (2.3)- do neurônio e a saída combinada linearmente das entradas ponderadas u_k pode ser vista na Figura 2.4. Observa-se que a transformação afim é aplicada devido a utilização do bias, portanto o gráfico do campo induzido local em função da saída combinada linearmente não é mais interceptado na origem.

$$v_k = u_k + b_k \quad (2.3)$$

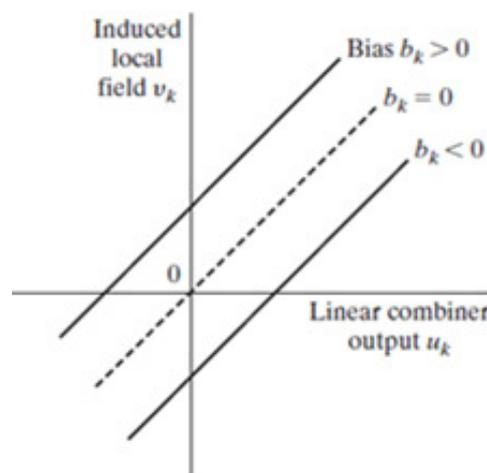


Figura 2.4: Transformação afim
Fonte: (Haykin, 1999).

Já a função de ativação é determinada pela complexidade da rede neural implementada e por sua adaptação à classe de problemas selecionada (Galushkin, 2007). Ela pode ser linear ou não linear dependendo do tipo de problema que o neurônio está tentando resolver (Hagan et al., 2014). Usa-se funções de ativação não lineares nas redes neurais artificiais para que essas funcionem de maneira distinta de um filtro linear (Mandic & Chambers, 2001), ou seja, é a função de ativação não linear que introduz a não linearidade na rede para que essa possa resolver problemas complexos. Por isso que essas funções são empregadas em redes neurais para efetuarem previsão de demanda de energia elétrica, pois, nas palavras de Tjøstheim (1994), é natural assumir o impacto da temperatura no consumo de eletricidade como não linear, no qual as temperaturas baixas e elevadas levam ao aumento do consumo. Algumas funções de ativação podem ser vistas na Figura 2.5.

Name	Input/Output Relation	Icon
Hard Limit	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$	
Symmetrical Hard Limit	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$	
Linear	$a = n$	
Saturating Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n \leq 1$ $a = 1 \quad n > 1$	
Symmetric Saturating Linear	$a = -1 \quad n < -1$ $a = n \quad -1 \leq n \leq 1$ $a = 1 \quad n > 1$	
Log-Sigmoid	$a = \frac{1}{1 + e^{-n}}$	
Hyperbolic Tangent Sigmoid	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$	
Positive Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$	
Competitive	$a = 1 \quad \text{neuron with max } n$ $a = 0 \quad \text{all other neurons}$	

Figura 2.5: Funções de ativação
Fonte: (Hagan et al., 2014).

2.3 Rede Neural Artificial

Existe uma quantidade considerável de definições sobre redes neurais artificiais na literatura. Para Harvey (1994), redes neurais são um sistema dinâmico de interconexões em um único sentido que carregam informações por unidades de processamento denominadas nós que são conectados entre si por elos, esses nós possuem um único sinal de saída que se espalha para outros nós. Para Mandic & Chambers (2001), redes neurais são consideradas como filtros não lineares adaptativos fortemente interconectados. Enquanto Haykin (1999) define uma rede neural como um processador distribuído paralelamente feito por simples unidades de processamento que possuem uma natural propensão para armazenar informação e torná-las acessível para uso.

Uma rede neural artificial com seis neurônios na camada de entrada (primeira camada), três neurônios na segunda camada (primeira camada intermediária), quatro neurônios na terceira camada (segunda camada oculta) e um único neurônio na camada de saída (quarta camada) é mostrada na Figura 2.6. A partir da análise dessa imagem verifica-se que uma rede neural artificial pode ser considerada um conglomerado de neurônios artificiais ordenados paralelamente em camadas que estão ligados entre si por conexões ponderadas. De acordo com Galushkin (2007), o número de neurônios em cada camada deve ser escolhido de maneira suficiente para

que a rede consiga resolver problemas complexos, já o número de camadas é desejável que seja o mínimo possível para diminuir o tempo de solução do problema. A maioria das redes neurais práticas possuem, conforme Hagan et al. (2014), duas ou três camadas intermediárias, mais do que isso são raros os casos.

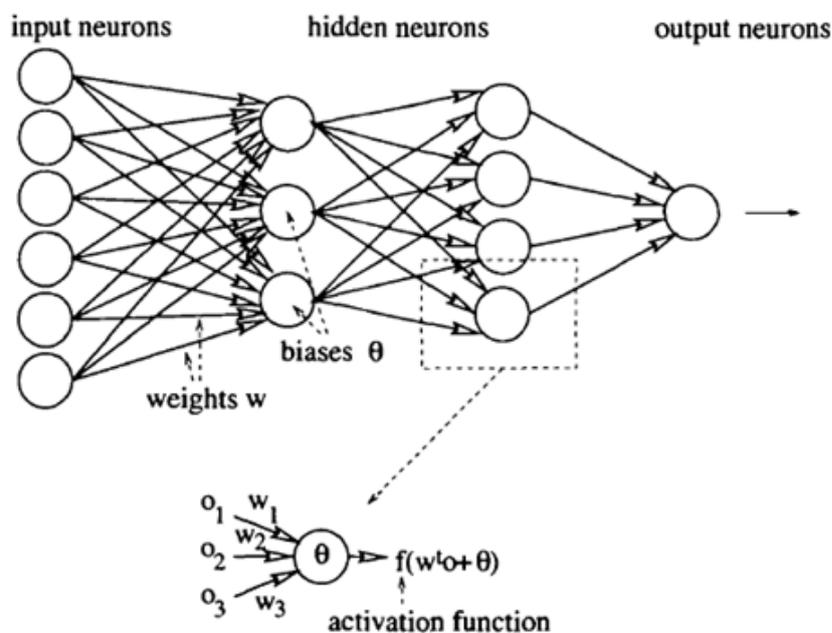


Figura 2.6: Rede neural artificial

Fonte: (Hammer, 2000).

A rede da Figura 2.6 é conhecida como rede neural multicamadas *feedforward*, pois possui várias camadas em sua arquitetura e, segundo Hagan et al. (2014), o sinal de saída é calculado diretamente da entrada em uma única passagem, nenhum *feedback* está envolvido. Uma rede neural artificial sem ciclos é conhecida como rede neural *feedforward* (Graves, 2012) que são usadas para reconhecimento de padrões (Hagan et al., 2014). O cálculo do sinal de entrada da rede multicamadas supracitado é realizado pelos neurônios ocultos presentes nas camadas intermediárias (Haykin, 1999). Algumas arquiteturas de redes *feedforward* bem conhecidas são o *perceptrons*, *radial basis function networks* (RBF) e *Kohonen maps*; sendo que dessa configuração a mais utilizada é a MLP (Graves, 2012). Por esse motivo a rede neural artificial de arquitetura MLP foi usada neste trabalho para efetuar previsões de demanda de energia de curto prazo, embora suas aplicações envolvam mais reconhecimento de padrões. Os seus resultados preditivos do capítulo 4 mostraram que o tempo de simulação, em comparação com as demais redes neurais, foi o menor. Isso se deve ao fato da rede MLP ser do tipo *feedforward*, sem ciclos, que é uma das justificativas para o uso dessas redes na previsão de demanda de energia elétrica até os dias atuais (conforme os trabalho de Massaoudi, Refaat, Chihi, Trabelsi, Oueslati & Abu-Rub (2021), Somu, Raman M R & Ramamritham (2021) e Adewuyi, Aina & Oluwaranti (2020)), apesar de não serem criadas para tal aplicação.

O *perceptron* inventado em 1943 pelo psicólogo Rosenblatt tem um espaço especial no

desenvolvimento histórico das redes neurais, pois foi a primeira rede neural descrita em forma de algoritmo e inspirou engenheiros, físicos e matemáticos a pesquisarem diferentes aspectos dessa área (Haykin, 1999). A rede neural artificial de camada única *feedforward* adaptativa com saída limitada a valores lógicos que possui uma certa capacidade de aprendizagem é conhecida como *perceptron* (Medsker & Jain, 2001). O *perceptron* de Rosenblatt apresentado na Figura 2.7 foi construído baseado no neurônio não linear de McCulloch e Pitts (Haykin, 1999). Fazendo um comparativo entre o neurônio artificial da Figura 2.3 e a rede neural *perceptron* da Figura 2.7 é notório suas semelhanças estruturais. Ambos possuem entradas ponderadas, bias, campo induzindo local, função de ativação e saída. A diferença é que o perceptron possui saída limitada entre zero e um pela função de ativação não linear *Hard limit* (ver Figura 2.5) e sua capacidade de aprendizagem é por intermédio de sua forma estrutural algorítmica.

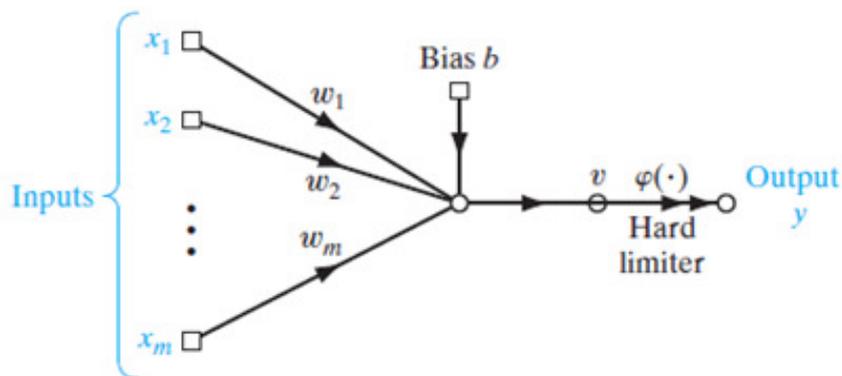


Figura 2.7: *Perceptron*

Fonte: (Haykin, 1999).

Rosenblatt e seus colegas desenvolveram o modelo perceptron e demonstraram sua capacidade para reconhecer padrões que mais tarde, infelizmente, foi verificado que a rede poderia apenas resolver uma classe limitada de problemas (Hagan et al., 2014). Essa classe limitada de problemas é, de acordo com (Medsker & Jain, 2001), classificar padrões linearmente separáveis a partir de um dado conjunto de padrões de treinamento de entrada-saída, no qual o perceptron aprende com os padrões exemplares a adaptar suas representações paramétricas para corresponder a essas classes.

O perceptron é a arquitetura mais simples de rede neural usada para classificação de padrões que são separados linearmente, em outras palavras, padrões que ficam em lados opostos de um hiperplano (Haykin, 1999). A Figura 2.8 apresenta três gráficos que ilustram bem esse cenário, onde uma reta azul divide o hiperplano em dois no qual cada um possui padrões distintos. Nos três casos o hiperplano cinza contém os padrões de pontos pretos e o hiperplano branco envolve os padrões de pontos brancos. A linha azul que segrega o hiperplano para fazer a classificação dos padrões é chamada de, segundo Hagan et al. (2014), fronteira de decisão. Analisando a Figura 2.8 é verificado que a fronteira de decisão é linear pelo fato do perceptron ser uma rede neural utilizada para classificar padrões linearmente separáveis.

A Figura 2.9 mostra três gráficos que contém conjuntos de padrões que não são possí-

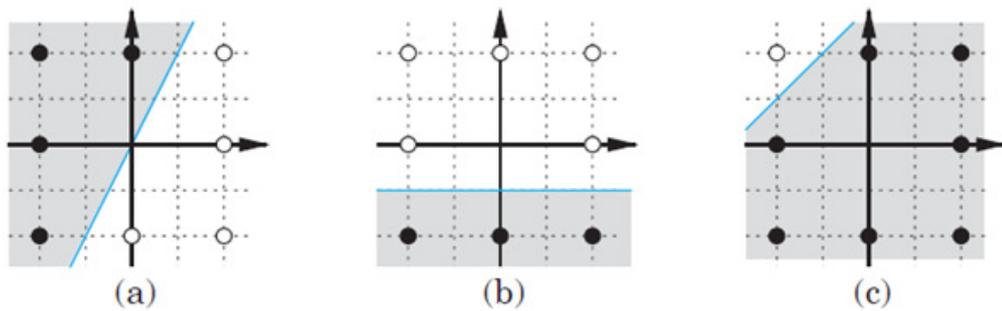


Figura 2.8: Exemplos de classificação perceptron

Fonte: (Hagan et al., 2014).

veis de serem separados linearmente. Em casos como esses o *perceptron* não é capaz de criar uma fronteira de decisão que separe as duas classes corretamente. Outra limitação dessa rede é a incapacidade de implementar funções elementares, por esse motivo foi criada a rede neural *perceptron* multicamadas, mais conhecida pelo seu acrônimo MLP, e os seus respectivos algoritmos de treinamento para superar essa limitação (Hagan et al., 2014).

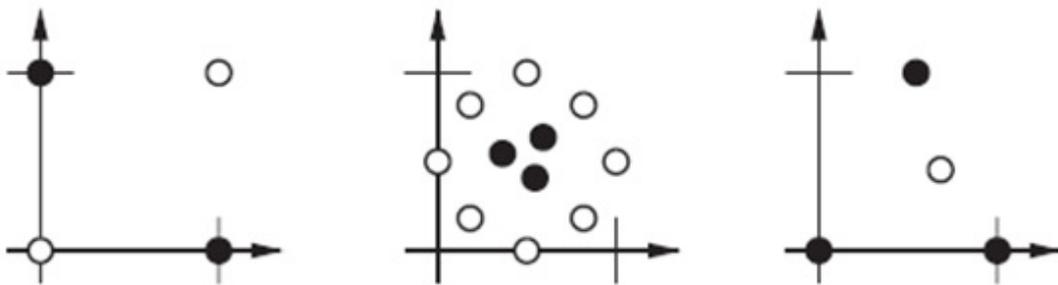


Figura 2.9: Problemas linearmente inseparáveis

Fonte: (Hagan et al., 2014).

A fronteira de decisão será sempre ortogonal em relação a matriz peso das entradas e sua posição pode ser modificada alterando o bias (Hagan et al., 2014), logo a fronteira de decisão é definida pela Equação (2.4). A matriz dos pesos e o bias que classificam corretamente os padrões são encontrados durante o aprendizado da rede, mesmo que esses parâmetros sejam inicializados aleatoriamente (Hagan et al., 2014)

$$wx + b = 0 \quad (2.4)$$

2.4 Aprendizado de Redes Neurais *Feedforward*

O procedimento usado para realizar o processo de aprendizagem de uma rede neural é chamado de algoritmo de aprendizagem, cuja função é modificar os pesos sinápticos de forma ordenada para atingir um objetivo desejado (Haykin, 1999). Na área de previsões de demanda de energia de curto prazo, os pesos sinápticos das redes neurais artificiais são alterados durante o

processo de aprendizagem visando previsões mais realistas. Como a classe de função das redes neurais é parametrizada, um algoritmo de aprendizagem pode ser utilizado para determinar os valores apropriados para esses parâmetros de modo que a função especificada se ajuste aos exemplos dados (Hammer, 2000). Treinar uma rede neural para desempenhar uma determinada tarefa é o propósito do algoritmo de aprendizagem, esses algoritmos podem ser divididos em três grandes grupos: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por esforço (Hagan et al., 2014).

Aprendizado supervisionado é utilizado em problemas de aprendizado de máquina em que um conjunto de pares entrada-destino é fornecido para treinamento com o objetivo de efetuar uma determinada tarefa, esse é o aprendizado definido para treinar as redes neurais artificiais no presente trabalho; aprendizagem por reforço é onde apenas valores escalares de recompensa são fornecidos para o treinamento e aprendizado não supervisionado é onde algoritmo tenta descobrir a estrutura dos dados por inspeção sozinho, pois não existe sinal de treinamento (Graves, 2012).

Para resolver problemas de aprendizado de máquina utilizando o aprendizado supervisionado pode-se recorrer ao algoritmo *backpropagation* que, segundo (Hagan et al., 2014), foi descoberto independentemente por vários pesquisadores e gerou grandes avanços na área por treinar redes *multilayer perceptron* (MLP). A Figura 2.10 é composta por um neurônio alimentado por vários sinais de entrada provenientes de uma camada de neurônios situada à sua esquerda e servirá para descrever o algoritmo de treinamento *backpropagation* (Haykin, 1999). A saída $y_j(n)$ do neurônio j é multiplicada pelo escalar menos um (-1) e, então, subtraída da resposta desejada $d_j(n)$ para gerar o erro $e_j(n)$. Onde n denota o intervalo de tempo na aplicação do algoritmo.

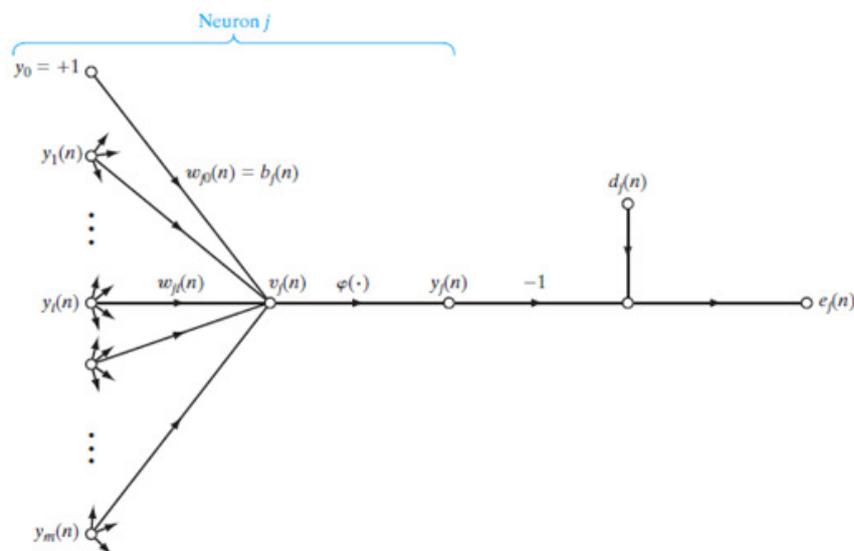


Figura 2.10: Neurônio alimentado por vários sinais de entrada
Fonte: (Haykin, 1999).

O campo local induzido $v_j(n)$ produzido na entrada da função de ativação associada ao neurônio j é, portanto, apresentado na Equação (2.5). Onde m é o número total de entradas

aplicadas ao neurônio j , e w_{j0} é o peso sináptico correspondente à entrada fixada y_0 sendo igual ao bias b_j aplicado ao neurônio j .

$$v_j(n) = \sum_{i=0}^m w_{ji}(n)y_i(n) \quad (2.5)$$

Em vista disso o sinal de função $y_j(n)$ que aparece na saída do neurônio j na iteração n é mostrado na Equação (2.6). Onde ϕ_j é função de ativação do neurônio j .

$$y_j(n) = \phi_j(v_j(n)) \quad (2.6)$$

Backpropagation é um algoritmo de descida de gradiente em que o índice de desempenho é o erro quadrático médio (Hagan et al., 2014). Nas palavras de Mandic & Chambers (2001), *Backpropagation* é um procedimento computacional para obter gradientes necessários para adaptação dos pesos das camadas ocultas de uma rede neural e não é muito diferente de um algoritmo de gradiente geral. De acordo com Graves (2012), *Backpropagation* é simplesmente uma aplicação repetida da regra da cadeia para derivadas parciais, no qual seu primeiro passo é calcular as derivadas da função de perda (ξ) em relação aos pesos sinápticos. O algoritmo do *Backpropagation* aplica uma correção $\Delta w_{ij}(n)$ no peso sináptico $w_{ij}(n)$ que é proporcional à derivada parcial da função perda em relação ao peso sináptico, esse gradiente pode ser expresso com auxílio da regra da cadeia na Equação (2.7) (Haykin, 1999).

$$\frac{\partial \xi(n)}{\partial w_{ij}(n)} = \frac{\partial \xi(n)}{\partial e_i(n)} \frac{\partial e_i(n)}{\partial y_i(n)} \frac{\partial y_i(n)}{\partial v_i(n)} \frac{\partial v_i(n)}{\partial w_{ij}(n)} \quad (2.7)$$

Fazendo algumas manipulações matemáticas na Equação (2.7) obtém-se a Equação (2.8), ver passo a passo em (Haykin, 1999).

$$\frac{\partial \xi(n)}{\partial w_{ij}(n)} = -e_j(n)\phi'(v_i(n))y_i(n) \quad (2.8)$$

A correção $\Delta w_{ij}(n)$ incrementada no peso sináptico $w_{ij}(n)$ é definida pela regra delta ou pela Equação (2.9) (Haykin, 1999), onde η é um parâmetro de taxa de aprendizado proveniente do algoritmo *Backpropagation*. Segundo Hagan et al. (2014), a regra delta é utilizada para ajustar os pesos sinápticos com o intuito de minimizar o erro médio quadrático da rede.

$$\Delta w_{ij}(n) = -\eta \frac{\partial \xi(n)}{\partial w_{ij}(n)} \quad (2.9)$$

De acordo com Haykin (1999), o sinal de negativo da Equação (2.9) indica que o gradiente está descendo até chegar em um mínimo local ou global. Segundo Medsker & Jain (2001), é necessário um valor pequeno para o parâmetro η visando garantir uma performance estável do algoritmo.

2.5 Rede Neural Recorrente

As redes neurais recorrentes (do inglês *recurrent neural network*, com acrônimo RNN) foram um importante foco de pesquisa e desenvolvimento durante a década de 1990, sendo projetadas para aprender padrões sequenciais ou variantes no tempo por intermédio da recorrência do sinal alimentado Medsker & Jain (2001); logo são amplamente empregadas pela literatura no que concerne à previsão de demanda de energia elétrica de curto prazo. Esse tipo de rede preenche a lacuna deixada pela rede neural de arquitetura MLP que não possui capacidade de aprender esse tipo específico de classe por ser do tipo *feedforward*, como visto anteriormente.

As RNN são uma classe de arquitetura de redes neurais artificiais que foram inspiradas pelas conexões cíclicas dos neurônios no cérebro, por isso utilizam funções de realimentação iterativas para armazenar informações (Graves, 2012). Essas arquiteturas de redes são uma ferramenta natural em qualquer domínio onde o tempo desempenha um papel importantíssimo, como reconhecimento e produção de voz, controle ou previsão de série temporal (Hammer, 2000); por isso vários modelos que são baseados em arquiteturas recorrentes foram utilizados no presente trabalho para prever a necessidade energética no curto prazo. Logo seu comportamento é muito mais complexo do que uma rede neural *feedforward* (Hagan et al., 2014). Algumas arquiteturas de redes neurais recorrentes são mencionadas a seguir: *Hopfield*, *Input–Output Recurrent Model*, *State-Space Model*, *Recurrent Multilayer Perceptrons*, *Second-Order Network* (Haykin, 1999); *Elman Recurrent Neural Network*, *Long Short-Term Memory* e *Gated Recurrent Unit* (Bianchi, Maiorino, Kampffmeyer, Rizzi & Jenssen, 2017).

A Figura 2.11 mostra uma rede neural recorrente simples com três neurônios na camada de entrada, quatro neurônios na camada intermediária e dois neurônios na camada de saída. As duas últimas camadas possuem em todos os seus neurônios a função de ativação *sigmoid*, e as conexões cíclicas estão na camada oculta (intermediária).

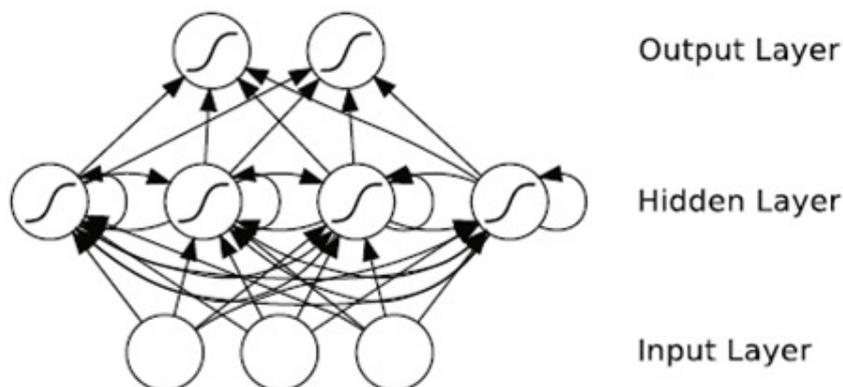


Figura 2.11: Rede neural recorrente

Fonte: (Graves, 2012).

A grande diferença entre uma rede neural recorrente e uma rede neural *feedforward* é que a primeira possui, pelo menos, um *feedback loop* (Haykin, 1999), ou seja, a saída de

um neurônio é utilizada como sua entrada em algum instante futuro. Outra diferença entre as duas redes é, nas palavras de Mandic & Chambers (2001), que a *feedforward* é estática e conseqüentemente não possui memória, enquanto as recorrentes são dinâmicas e possuem capacidade de armazenar informações temporariamente. Esse mecanismo específico possibilita que as redes neurais de arquitetura recorrente possuem a tendência de fornecer previsões de demanda de energia no curto prazo mais próximas da realidade do que as redes *feedforward*.

Uma rede neural recorrente pode ser vista como uma rede neural *feedforward* quando seu *feedback loop* é desdobrado no tempo (Hagan et al., 2014). Esse processo auxilia no entendimento de redes recorrentes com dependências mais complexa (Graves, 2012) e, também, no algoritmo de treinamento backpropagation ao longo do tempo para redes neurais recorrentes (Haykin, 1999). A Figura 2.12 ilustra uma rede neural recorrente desdobrada no tempo, onde cada nó representa uma camada de unidades da rede em um único passo de tempo. As conexões ponderadas da camada de entrada para a camada oculta são identificadas como 'w1', aquelas da camada oculta para ela mesma (ou seja, os pesos recorrentes) são rotuladas como 'w2' e os pesos ocultos para a saída são rotulados 'w3'.

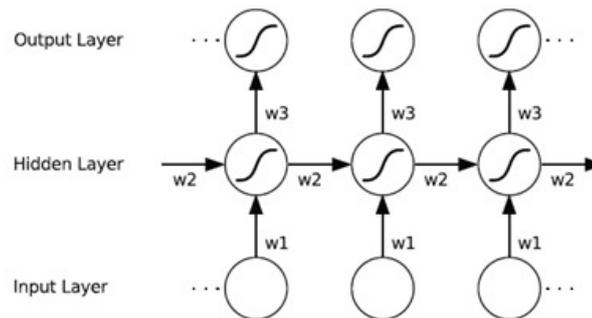


Figura 2.12: Rede neural recorrente desdobrada no tempo
Fonte: (Graves, 2012).

A passagem de informação pela camada de entrada até chegar na camada de saída de uma rede neural recorrente é a mesma de um *perceptron* multicamadas com uma única camada oculta, exceto que as ativações chegam à camada oculta tanto da entrada externa atual quanto das ativações da camada oculta do passo de tempo anterior (Graves, 2012). A Equação (2.10) (Graves, 2012) apresenta o processo supracitado formalmente, onde x é o sinal de entrada inserido em uma rede neural com I neurônios na camada de entrada e com H neurônios na camada oculta. Seja x_i^t o valor da entrada i no instante de tempo t , a_j^t o valor da entrada no neurônio j no instante de tempo t , b_j^t a ativação do neurônio j no instante de tempo t , w_{ih} o peso sináptico entre o neurônio i da camada de entrada e o neurônio h da camada oculta, e $w_{h'h}$ é o peso sináptico entre o neurônio h' da camada oculta no instante de tempo $t - 1$ e o neurônio h da camada oculta no instante de tempo t .

$$a_h^t = \sum_{i=1}^I w_{ih} x_i^t + \sum_{h'=1}^H w_{h'h} x_{h'}^{t-1} \quad (2.10)$$

Uma função de ativação diferenciável e não linear θ_h é, então, aplicada da mesma forma como em uma rede neural perceptron multicamadas conforme a Equação (2.11) (Graves, 2012).

$$b_h^t = \theta_h(a_h^t) \quad (2.11)$$

De acordo com (Graves, 2012), a sequência completa das ativações ocultas pode ser calculada começando com $t = 1$ e aplicando recursivamente as Equações (2.10) e (2.11) incrementando t a cada iteração. Ainda nas palavras do autor, as entradas da rede para as unidades (neurônios) de saída podem ser calculadas ao mesmo tempo como nas ativações ocultas, conforme a Equação (2.12).

$$a_k^t = \sum_{h=1}^H w_{hk} b_h^t \quad (2.12)$$

As redes neurais recorrentes geralmente não são treinadas pelo algoritmo *backpropagation*, pois, segundo Medsker & Jain (2001), esse processo iterativo não é tão efetivo quanto o *Backpropagation Through Time* (BPTT) em que o sinal do erro é propagado de volta no tempo e dependências temporais são aprendidas de melhor forma; consequentemente fornecendo previsões de demanda de energia de curto prazo mais efetivas. O algoritmo *Backpropagation Through Time* é baseado na premissa de que a operação temporal de uma rede recorrente pode ser desdobrado em um perceptron multicamadas para, então, aplicar o algoritmo padrão do *backpropagation* (Medsker & Jain, 2001). Ambos os algoritmos de descida do gradiente consistem na aplicação repetitiva da regra da cadeia para efetuarem o treinamento das redes (Graves, 2012).

Para redes recorrentes, Segundo Graves (2012), a função de perda depende da ativação da camada oculta não apenas por meio de sua influência na camada de saída, mas também por meio de sua influência na camada oculta no próximo passo de tempo. Logo a Equação (2.13) é apresentada, onde δ_j é definido pela Equação (2.14) que é representada pela derivada parcial de alguma função de perda diferenciável em relação a saída da rede.

$$\delta_h^t = \theta'(a_h^t) \left(\sum_{k=1}^K \delta_k^t w_{hk} + \sum_{h'=1}^H \delta_{h'}^{t+1} w_{hh'} \right) \quad (2.13)$$

$$\delta_h^t = \frac{\partial \xi}{\partial a_j^t} \quad (2.14)$$

Por fim, tendo em mente que os mesmos pesos são reutilizados a cada instante de tempo, é somada toda a sequência para obter as derivadas em relação aos pesos sinápticos da rede (Graves, 2012). O resultado é apresentado na Equação (2.15).

$$\frac{\partial \xi}{\partial w_{ij}} = \sum_{t=1}^T \frac{\partial \xi}{\partial a_j^t} \frac{\partial a_j^t}{\partial w_{ij}} = \sum_{t=1}^T \delta_j^t b_i^t \quad (2.15)$$

Um problema que requer atenção em aplicações práticas quando é utilizada redes neurais recorrentes é o *vanishing gradient* que surge durante o treinamento da rede para dados de entrada do passado distante (Haykin, 1999). Logo para essa arquitetura padrão, a gama de contexto que pode ser acessada na prática é bastante limitada (Graves, 2012); incluindo previsões de demanda de energia no curto prazo. Quando as redes neurais recorrentes são treinadas por algoritmos de descida de gradiente, ocorre o esquecimento de informações antigas pelo *vanishing gradient* (Mandic & Chambers, 2001); ou seja, dados de consumo de energia elétrica de anos anteriores são, simplesmente, esquecidos pela rede RNN. Esse problema ocorre pela influência de uma determinada entrada na camada oculta e, posteriormente, na saída da rede decair ou explodir exponencialmente à medida que circula em torno das conexões recorrentes da rede (Graves, 2012).

2.6 Rede Neural LSTM

A rede neural recorrente básica (RNN) descrita anteriormente não possui estrutura suficiente em modelagem sofisticada dinâmica temporal e, devido a isso, na prática, tem se mostrado incapaz de olhar no passado em muitos tipos de sequências de entrada (Yu & Deng, 2014); isso é um problema no momento de efetuar previsões de demanda de energia de curto prazo, pois informações de consumo de energia de períodos distantes não são levados em consideração. Uma solução é adotar arquiteturas com portas, tal como a *long short-term memory* (LSTM), que foram projetadas especificamente para lidar com *vanishing gradients* e permitem que a rede aprenda dependências de alcance temporal muito mais extenso (Bianchi et al., 2017); em razão disso é encontrado na literatura uma extensa gama de trabalhos científicos utilizando-as na previsão de demanda de energia elétrica de curto prazo, pois, geralmente, as séries históricas de consumo de eletricidade contém dados de vários anos.

Segundo Yu & Deng (2014), uma rede neural com estrutura LSTM é uma versão avançada de uma rede neural recorrente básica e, nas palavras de Kamath, Liu & Whitaker (2019), foi desenvolvida por Hochreiter e Schmidhuber para lidar com o problema de dependências de longo prazo durante o treinamento pelo algoritmo do *backpropagation through time*. Esse problema é solucionado pela memória de longo prazo resultante das células *long short-term memory* que compõem a estrutura dessa arquitetura recorrente (Yu & Deng, 2014). Devido a isso, tal arquitetura é amplamente utilizada hoje em dia por causa do seu superior desempenho na modelagem precisa de dependências de dados de curto e longo prazo (Bianchi et al., 2017); principalmente no que diz respeito à aplicação de redes neurais na previsão de demanda de energia no curto prazo.

2.6.1 *Vanishing Gradient* e a Rede LSTM

O problema do *vanishing gradient* ocorre durante o treinamento pelo algoritmo BPTT em redes neurais recorrentes básicas – Figura 2.13. De acordo com (Graves, 2012), o sombreamento dos nós na rede desdobrada indica sua sensibilidade às entradas no instante de tempo 1 (quanto mais escura a tonalidade, maior a sensibilidade). Ainda nas palavras do autor, a sensibilidade diminui com o tempo à medida que novas entradas substituem as ativações da camada oculta e, conseqüentemente, a rede ‘esquece’ as informações das primeiras entradas. No que tange à previsão de demanda de energia elétrica de curto prazo, a rede trabalha apenas com o consumo energético de períodos mais recente, últimas entradas, pois o restante das informações, que são as mais antigas, foram ‘esquecidas’.

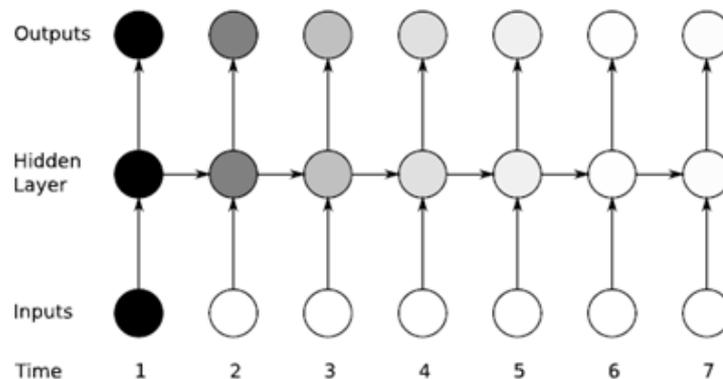


Figura 2.13: *Vanishing gradient* em redes recorrentes simples
Fonte: (Graves, 2012).

A ideia básica por trás da célula LSTM na rede neural recorrente é utilizar três tipos de estruturas de portas para controlar o fluxo de informações da rede (Yu & Deng, 2014). Essas portas são conhecidas como *input*, *output* e *forget gates* que têm como função proteger a memória da célula que armazena o estado oculto quando requisitada no próximo instante de tempo (Kamath et al., 2019). Portanto as células de memória LSTM podem armazenar e acessar informações durante longos períodos de tempo, atenuando, assim, o problema do *vanishing gradient* (Graves, 2012); logo fornecem resultados preditivos de demanda de energia elétrica de curto prazo mais próximos da realidade. Essas células podem ser consideradas como uma unidade de rede complexa e inteligente, capaz de lembrar informações por um longo intervalo de tempo (Yu & Deng, 2014).

A Figura 2.14 apresenta um diagrama de uma célula LSTM de uma rede neural recorrente. Onde x_t é a entrada da célula, i_t é o *input gate*, f_t é o *forget gate*, o_t é o *output gate*, \tilde{c}_t é o *candidate state*, c_t é o *cell state* e h_t é o *output*. De acordo com Bianchi et al. (2017), cada *gate* na célula tem uma funcionalidade específica e única. O *forget gate* f_t decide quais informações devem ser descartadas do *cell state* anterior c_{t-1} . O *input gate* i_t opera no *cell state* anterior c_{t-1} depois desse ser modificado pelo *forget gate* e decide o quanto o *cell state* atual c_t deve ser atualizado com informações do *candidate state* \tilde{c}_t . Para gerar a saída h_t da rede, primeiramente

a célula filtra o *cell state* atual c_t pela função de ativação não linear tangente hiperbólica (\tanh) e, então, o *output gate* o_t seleciona a parte do *cell state* atual c_t para ser retornado como saída h_t . Cada *gate* tem como função de ativação a *sigmoid* (σ) e depende da entrada externa atual x_t e da saída das células anteriores h_{t-1} .

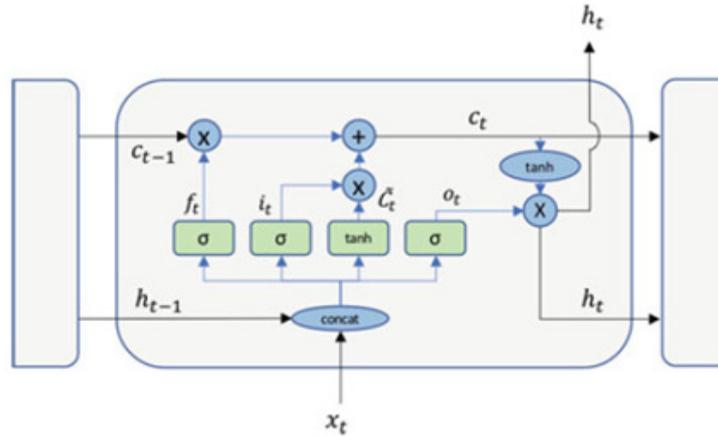


Figura 2.14: Diagrama de uma célula LSTM
Fonte: (Kamath et al., 2019).

As Equações (2.16), (2.17), (2.18), (2.19), (2.20) e (2.21) (Kamath et al., 2019) definem o passo que a entrada da célula percorre até atualizar o estado da célula atual e para calcular a saída. Segundo Bianchi et al. (2017), W_i , W_f , W_o e W_c são os pesos sinápticos atribuídos a entrada da célula LSTM, U_i , U_f , U_o e U_c são os pesos sinápticos das conexões recorrentes da rede neural para diferentes instantes de tempo, b_i , b_f e b_o são os bias e o círculo vazado das Equações (2.20) e (2.21) representa o produto de Hadamard (multiplicação ponto a ponto entre dois vetores ou matrizes).

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (2.16)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (2.17)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (2.18)$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1}) \quad (2.19)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \quad (2.20)$$

$$h_t = o_t \circ \tanh(c_t) \quad (2.21)$$

A Figura 2.15 mostra o resultado do algoritmo *backpropagation through time* em uma rede LSTM. Segundo Graves (2012), todos os gates estão totalmente abertos ('O') ou fechado ('-') o que elimina o problema do *vanishing gradient*, pois a saída do instante de tempo seis é fortemente influenciada pela entrada no instante de tempo um. O contrário ocorre na Figura 2.13, onde a influência da entrada no instante de tempo um vai perdendo força de modo gradual nas saídas temporais posteriores, tornando-se nula no instante de tempo seis.

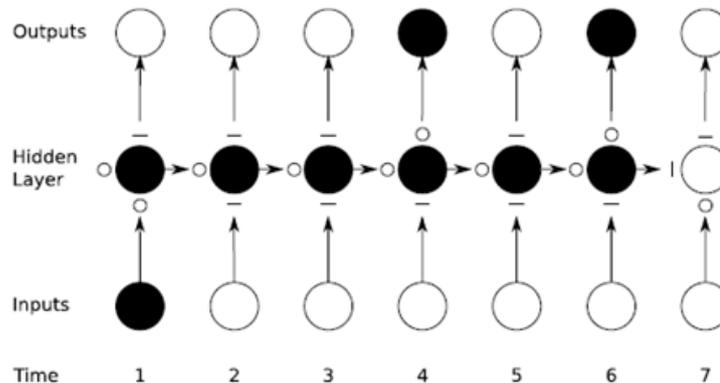


Figura 2.15: Preservação de informações de gradiente por redes LSTM
Fonte: (Graves, 2012).

2.7 Rede Neural GRU

Além da rede LSTM, outra alternativa para superar o obstáculo do *vanishing gradient* é utilizar a rede neural de arquitetura *Gated Recurrent Unit* (GRU) que é, basicamente, sua versão melhorada (Huo, Wu, Miao, Wang, He & Su, 2019). Enquanto a estrutura bastante complexa da rede LSTM é composta por *input gate*, *forget gate* e *output gate* (Xiuyun, Ying, Yang, Chengzhi, Wen & Yimiao, 2018), a rede GRU não possui o *output gate* em sua estrutura e, por isso, seu tempo de treinamento é muito menor devido a quantidade inferior de parâmetros (Wang, Liao & Chang, 2018). Em outras palavras, a rede GRU é mais eficiente em alcançar a convergência e atualizar os pesos internos durante o treinamento pois sua estrutura de portas internas é mais sucinta do que a rede LSTM (Xie, Wang, Wan & Tang, 2020). Esses mecanismos de portas presentes nas redes GRU são simplificados nas células recorrentes para aumentar significativamente a eficiências computacional na tentativa de manter o mesmo desempenho preditivo da rede LSTM (Lv, Liu, Yu, Zheng & Lv, 2020). Com isso é possível obter a mesma performance nas previsões de demanda de energia elétrica de curto prazo da rede LSTM com a rede GRU.

Uma célula ou unidade típica da rede neural de arquitetura GRU é formada com base em duas portas nomeadas de *reset gate* e *update gate* (Lee, Kim, Kim, Hur & Kim, 2018). A primeira porta denominada de *reset gate* tem como função filtrar informações anteriormente irrelevantes nas camadas ocultas (Fengming, Shufang, Zhimin, Bo, Shiming & Mingming,

2017) e quanto menor o seu valor maior é a quantidade de informações ignoradas (Huo et al., 2019). Caso seu valor seja próximo de 0 a informação do momento anterior é desprezada, por outro lado, quando esse valor for próximo de 1, a informação oculta do instante de tempo passado é retida na memória atual (Li, Luo, Liu, Wang, Zhu, Deng & Zhang, 2020). Já a segunda porta, que é a *update gate*, determina a quantidade de informações a serem transferidas para a camada de saída (Fengming et al., 2017) e quanto maior o seu valor, mais informações contidas no estado anterior são utilizadas (Huo et al., 2019). Se o seu valor for próximo de 0 a informação do estado oculto do instante anterior será ignorada, já se esse valor for próximo de 1 a informação é retida no estado oculto atual (Li, Luo, Liu, Wang, Zhu, Deng & Zhang, 2020).

A unidade estrutural da rede neural artificial GRU apresentada pela Figura 2.16 possui duas entradas em instantes de tempo distintos, sendo elas o vetor de entrada atual x_t e o vetor de saída h_{t-1} do instante de tempo anterior, onde a saída de cada porta pode ser obtida através de operações lógicas e transformações não lineares da entrada (Wang et al., 2018). As Equações (2.22), (2.23), (2.24), (2.25) que regem o funcionamento da célula da rede neural GRU da Figura 2.16 são mostradas logo abaixo, no qual \tilde{z}_t é o update gate, r_t é o reset gate, h_t é o estado oculto candidato do nó oculto atual, h_t é o estado oculto atual, x_t é a entrada atual da rede neural artificial e h_{t-1} é o estado oculto do instante de tempo anterior. Dentre essas variáveis, o update gate z_t determina a integração entre a informação de uma nova entrada com as informações históricas e o reset gate r_t estabelece a proporção do estado da informação no modelo (Xiuyun et al., 2018).

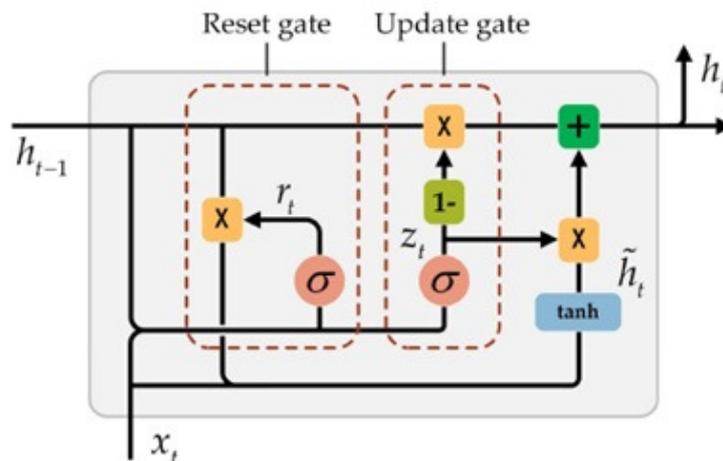


Figura 2.16: Unidade estrutural da rede neural artificial GRU
Fonte: (Li, Luo, Liu, Wang, Zhu, Deng & Zhang, 2020).

$$z_t = \sigma(w_{zx}x_t + u_{zh}h_{t-1}) \quad (2.22)$$

$$r_t = \sigma(w_{rx}x_t + u_{rh}h_{t-1}) \quad (2.23)$$

$$\tilde{h}_t = \tan(w_{hx}x_t + r_t \circ u_{hh}h_{t-1}) \quad (2.24)$$

$$h_t = (1 - z_t) \circ \tilde{h}_t + z_t \circ h_{t-1} \quad (2.25)$$

A rede neural de arquitetura GRU emprega o mecanismo de recorrência para armazenar e recuperar informações por longos períodos de tempo, porém seu desempenho na prática pode não ser tão satisfatório quanto na teoria pelo fato da rede somente acessar informações passadas (Deng, Jia, Li, Tong, Qiu & Li, 2019). Isso é um evento indesejável, pois para muitas tarefas de modelagem sequencial é interessante ter acesso a informações tanto do passado quanto do futuro (Chen, Jiang & Zhang, 2019); para previsões de demanda de energia de curto prazo não poderia ser diferente. Visando superar essa condição foi criada a GRU bidirecional (Bi-GRU) que é capaz de realizar essa operação, pois consiste em duas GRU padrões que processam a sequência de entrada em dos sentidos distintos (cronológica e anticronológica) e, em seguida, mescla essas representações em uma única variável (Tao, Liu, Li & Sidorov, 2019).

A rede Bi-GRU contém uma camada futura em que a sequência dos dados está na direção oposta, logo essa rede utiliza duas camadas ocultas para extrair informações tanto do passado quanto do futuro e ambas estão conectadas na mesma camada de saída (Luo, Zhou, Wang, Zhu & Deng, 2018). Devido a esse mecanismo a rede é capaz de explorar informações passadas assim como futuras que, por sua vez, podem fornecer resultados preditivos mais eficientes (Chen et al., 2019), pois mais informações são extraídas e, conseqüentemente, melhora a performance do processo de aprendizagem (Zhang, Tian, Hong, Han, Ren & Chen, 2018). Por causa dessa característica peculiar o modelo Bi-GRU é utilizado no presente trabalho para que as previsões de demanda de energia de curto prazo possam ser melhores.

A Figura 2.17 apresenta uma rede neural Bi-GRU de duas camadas intermediárias orientada pelas Equações (2.26), (2.27), (2.28), (2.29) e (2.30), onde \vec{h}_t^1 e \vec{h}_t^2 são os vetores saídas das camadas *forward* da primeira e segunda camada da rede no instante de tempo t . Por outro lado, os vetores \overleftarrow{h}_t^1 e \overleftarrow{h}_t^2 representam as saídas da primeira e segunda camada *backward* da rede no mesmo instante de tempo t .

$$\vec{h}_t^1 = f(w_{x\vec{h}_1^1}x_t + w_{\vec{h}_1^1\vec{h}_1^1}\vec{h}_{t-1}^1 + b_{\vec{h}_1^1}) \quad (2.26)$$

$$\overleftarrow{h}_t^1 = f(w_{x\overleftarrow{h}_1^1}x_t + w_{\overleftarrow{h}_1^1\overleftarrow{h}_1^1}\overleftarrow{h}_{t+1}^1 + b_{\overleftarrow{h}_1^1}) \quad (2.27)$$

$$\vec{h}_t^2 = f(w_{\vec{h}_1^1\vec{h}_2^2}\vec{h}_t^1 + w_{\vec{h}_2^2\vec{h}_2^2}\vec{h}_{t-1}^2 + b_{\vec{h}_2^2}) \quad (2.28)$$

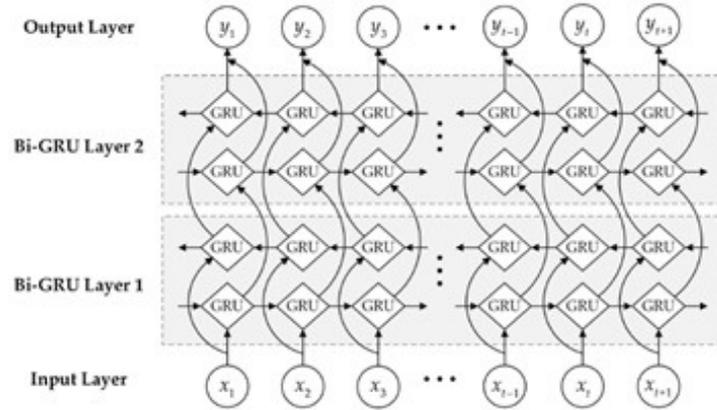


Figura 2.17: Unidade estrutural da rede neural BiGRU
 Fonte: (Li, Luo, Liu, Wang, Zhu, Deng & Zhang, 2020).

$$\overleftarrow{h}_t^2 = f(w_{\overleftarrow{h}^1 \overleftarrow{h}^2} \overleftarrow{h}_t^1 + w_{\overleftarrow{h}^2 \overleftarrow{h}^2} \overleftarrow{h}_{t+1}^2 + b_{\overleftarrow{h}^2}) \quad (2.29)$$

$$y_t = g(w_{\overrightarrow{h}^2 y} \overrightarrow{h}_t^2 + w_{\overleftarrow{h}^2 y} \overleftarrow{h}_t^2 + b_y) \quad (2.30)$$

2.8 Rede Neural CNN

Convolutional neural network (CNN) é um tipo de rede neural artificial profunda aplicada com frequência para resolver problemas nos quais os dados têm alta correlação local; como imagens visuais, previsão de vídeo e categorização de texto; pelo fato dessa rede específica ter a capacidade de capturar o mesmo padrão localizado em regiões distintas (Tian, Ma, Zhang & Zhan, 2018). Ao contrário das redes neurais de arquitetura recorrente, a CNN é um tipo de rede *feedforward* que não é formada por conexões cíclicas e não possui memória como entrada (Lee et al., 2018). Em comparação com os métodos de classificação tradicionais, a rede neural de arquitetura CNN pode não apenas mapear relacionamentos não lineares mais complexos, como também possui boa capacidade de generalização em suas respostas (Gong, Tang, Zhu, Liao & Song, 2020). Buscando melhorar as previsões de demanda de energia de curto prazo a partir dessas especificidades, utilizou-se a rede CNN no presente trabalho.

Embora a rede CNN tenha sido especialmente desenvolvida para resolver tarefas de classificação de imagens, no qual a rede é alimentada por dados de duas dimensões, esse modelo também é usado no campo de análise de séries temporais, no qual são utilizados dados de uma única dimensão, pois o conceito de *weight sharing* é empregado para aumentar o desempenho na solução de problemas não lineares, tal qual é visto nas previsões de demanda de energia elétrica (Sajjad, Khan, Ullah, Hussain, Ullah, Lee & Baik, 2020). De maneira sucinta, o *weight sharing* aplica as traduções de invariâncias na rede CNN com o intuito de ajudar a filtrar o recurso de aprendizagem independentemente das propriedades espaciais (Albawi, Mohammed

& Alzawi, 2017). Além de efetuar classificações, as redes CNN são bastante empregadas na extração de recursos por meio do kernel que é, em poucas palavras, um filtro ou matriz que desliza sobre a entrada para efetuar a operação de convolução e produzir um mapa de recursos, e com isso kernels distintos criam mapas de recursos distintos e todos estes são mesclados para originar a saída da camada de convolução (Madhure, Raman & Singh, 2020).

Por ser do tipo *feedforward*, a rede neural de arquitetura CNN é capaz de codificar informações importantes contidas nos dados de entrada com muito menos parâmetros do que em outros modelos de aprendizado profundo (Zhang et al., 2018) e são compostas por camadas convolucionais, camadas *pooling* e camadas totalmente conectadas (Gong et al., 2020). A Figura 2.18 dispõe essas camadas organizadas de forma genérica para estruturar a rede CNN, no qual a camada de convolução tem a função de extrair recursos efetivos dos dados de entrada por meio de seus múltiplos kernels convolucionais intrínsecos e a camada de *pooling* pode manter os recursos fortes e descartar os recursos fracos para reduzir a complexidade e evitar overfitting. E, por último, a camada totalmente conectada soma todos os recursos locais para formar um recurso global utilizado no cálculo do resultado final (Tian et al., 2018).

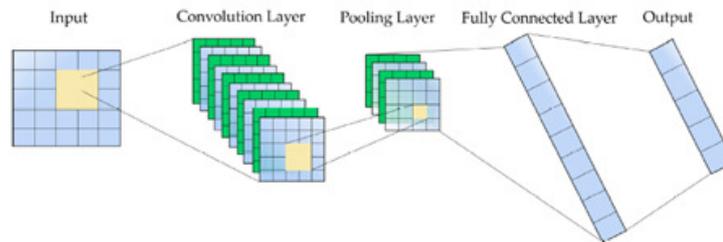


Figura 2.18: Estrutura da rede neural CNN

Fonte: (Wu, Wu, Feng, Yang, Liang, Yang & Zhao, 2021).

A ilustração de um processo convolucional a partir de uma entrada unidimensional é mostrada na Figura 2.19. Esse processo é regido pela Equação 2.31, onde $*$ representa a operação convolucional, I corresponde à entrada unidimensional da camada atual, K denota o Kernel unidimensional e S é a saída da convolução, também conhecida como *feature map*.

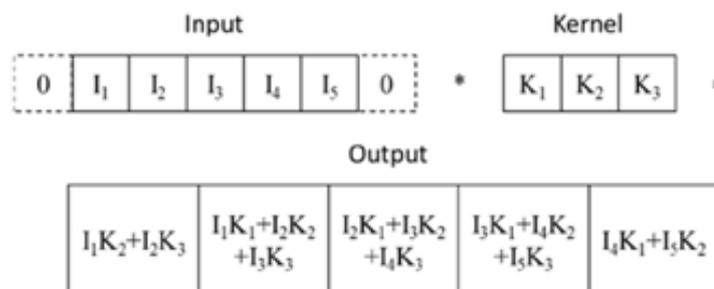


Figura 2.19: Processo convolucional a partir de uma entrada 1D

Fonte: (Tudose, Sidea, Picioroaga, Boicea & Bulac, 2020).

$$S(i) = (I * K)_i = \sum_{m=0}^i I(i-m)K(m) \quad (2.31)$$

2.9 Revisão Bibliográfica

Na literatura existe uma grande quantidade de artigos científicos pertinentes à previsão de demanda de eletricidade. Alguns utilizam métodos de predição estatísticos; por exemplo o trabalho de (Lekshmi & Adithya Subramanya, 2019) no qual foi utilizado o modelo ARIMA (*Autoregressive Integrated Moving Average*) para efetuar previsões de curto prazo de demanda de energia em uma subestação de 400 kV. Além das previsões, esse estudo analisou o impacto da influência da variável exógena temperatura no aumento da acurácia dos resultados preditivos, onde a métrica escolhida para determinar a elevação da acurácia foi a diminuição do erro RMSE (*Root Mean Square Error*). Para o horizonte de previsão de um dia, sem considerar a temperatura ambiente, chegou-se a um erro RMSE de 6.009955 e, com a introdução da variável exógena, o erro RMSE encontrado foi de 5.990357. De maneira similar, agora para o horizonte de previsão de uma semana, sem considerar a temperatura ambiente, chegou-se a um erro RMSE de 4.68697 e, com a introdução da variável exógena, o erro RMSE encontrado foi de 4.682843. Os resultados provam, nas palavras dos autores, que a inserção da variável exógena temperatura e a quantidade de amostras consideradas são fatores determinantes na acurácia das previsões de demanda de energia quando utilizadas séries temporais. A inserção da variável exógena de temperatura na rede foi visando melhorar a capacidade preditiva da demanda de energia (variável endógena), pois, de acordo com os trabalhos de Hekkenberg, Moll & Uiterkamp (2009), de de Cian, Lanzi & Roson (2013) e de De Cian, Lanzi & Roson (2011), a temperatura é um fator que influencia a demanda de energia elétrica.

Outros trabalhos que utilizam modelos ARIMA para realizar previsões de demanda de energia são citados a seguir: Noureen, Atique, Roy & Bayne (2019) analisaram e aplicaram o ARIMA sazonal em um estudo de caso utilizando dados de pequena escala relacionados à agricultura; onde os resultados preditivos foram encorajadores, todavia os autores concluíram que há escopo de pesquisas adicionais para refinar a ideia; já Mitkov, Noorzad, Gabrovska-Evstatieva & Mihailov (2019) selecionaram o modelo estatístico linear ARIMA (1,1,1) para prever o consumo de energia nos anos subsequentes da República Islâmica do Afeganistão, seus resultados mostraram que a demanda esperada poderá crescer anualmente entre 4391 GWh (*Gigawatt hours*) e 8375 GWh até 2024; enquanto (Angelaccio, 2019) realizou um estudo de caso em municípios italianos no qual foi efetuado previsões de consumo público de eletricidade a partir de modelos ARIMA, onde os resultados baseado na remoção de componentes cíclicas dos dados temporais foram 2.3% melhores, quando levado em consideração o erro MAPE (*Mean Absolute Percent Error*), daqueles sem alteração dos dados temporais.

Os métodos de previsões estatísticos fornecem resultados satisfatórios quando os dados relacionados à demanda de energia e temperatura são lineares, por exemplo; quando não existem muitas variações para diferentes instantes de tempo, de acordo com (Upadhaya, Thakur & Singh, 2019). Essa conclusão foi realizada pelos autores a partir do estudo de 126 artigos relacionados a diferentes modelos de previsão de demanda de energia compreendidos em 26 jornais

internacionais distintos. Outras conclusões interessantes do trabalho são descritas a seguir: para dados não lineares de demanda de energia e temperatura os modelos de redes neurais artificiais são mais indicados; seleção de modelos preditivos baseados somente na acurácia é insensato, pois a variável clima desempenha um importante papel; análise e processamento preliminar dos dados podem gerar melhores resultados e recomenda-se utilizar conjuntos de dados completos.

Além dos métodos estatísticos de previsão de demanda de energia, existem os métodos baseados em inteligência artificial. Zor, Timur & Teke (2017) mencionam em seu trabalho que os modelos preditivos da literatura pertencem a uma dessas categorias, onde os modelos de regressão linear, os modelos ARIMA e os modelos não paramétricos de regressão linear pertencem à primeira categoria, enquanto os modelos de redes neurais artificiais, *support vector machine* e *neuro-fuzzy inference system* pertencem à segunda categoria. Os autores comentam que o processo de previsão de demanda de energia é complexo pelo fato da influência de fatores econômicos, sociais e meteorológicos, por isso os métodos estatísticos não fornecem resultados satisfatórios quando utilizam essas variáveis não lineares. Em contrapartida, ainda nas palavras dos autores, os métodos de inteligência artificial ganharam importância notória na redução de erros de estimativa de previsão, pois conseguem extrair informações dessas variáveis a partir de suas componentes estruturais. Eles citam que as vantagens das redes neurais artificiais são: menor necessidade de treino estatístico formal, capacidade de detectar implicitamente relações complexas entre variáveis dependentes e independentes, possui acesso a vários algoritmos de treinamento e capacidade de detectar todas as possíveis iterações entre as variáveis predictoras. Por outro lado, as desvantagens citadas por eles são: processamento que altera os parâmetros internos da rede durante o aprendizado é uma caixa preta; custo computacional demasiado para efetuar o treinamento do modelo neural; a rede tende a *overfitting* quando seus hiperparâmetros não são escolhidos adequadamente e, por último, a natureza empírica de escolha desses hiperparâmetros do modelo neural desenvolvido.

No trabalho de Zhao & Tang (2016) é feita uma revisão abrangente do gerenciamento e previsão de demanda de energia em ambientes de *smart grid* no qual é comparado diferentes modelos preditivos compreendidos entre métodos estatísticos e de inteligência artificial. A comparação foi realizada por intermédio do erro MAPE onde o modelo de rede neural artificial sobressaiu em relação aos demais com 1,59% de acurácia. Enquanto os modelos de lógica *Fuzzy* e *support vector machine* obtiveram 2,21% e 2,71% de acurácia, respectivamente. Já o modelo híbrido composto pelo modelo linear ARIMA e modelo de inteligência artificial *support vector machine* foi responsável por um erro MAPE de 3,86%. O modelo de regressão linear múltipla teve 3,8% de acurácia, o modelo ARIMA teve 4,5% de acurácia e, por último, o modelo ARMA teve 4,77% de acurácia. Nesse trabalho os autores mostraram que a precisão dos métodos modernos é superior aos tradicionais por serem mais adequados para processamento de dados de entrada não lineares.

Os Modelos de redes neurais artificiais fazem parte dos métodos de inteligência artificial. Logo o MLP, que foi bastante explanado no referencial teórico, pode ser utilizado para efetuar

previsões de demanda de energia, apesar de seu desenvolvimento ser baseado em problemas de classificação. O trabalho de Saad, Hazirah, Suziana, Azhar, Yaacob, Ahmad & Yusnita (2017) mostra o modelo de rede neural MLP aplicado em três sistemas elétricos de tensões distintas, sendo essas grandezas físicas de 415V, 11kV e 33kV. As redes neurais foram treinadas com os algoritmos *BackPropagation* (BP), *Recursive Prediction Error* (RPE), *Modified Recursive Prediction Error* (MRPE) e *Adaptive Learning Recursive Prediction Error* (ALRPE) para efetuarem previsões de demanda de energia de curto prazo. Nos três casos o algoritmo de treinamento ALRPE forneceu melhores resultados preditivos que os algoritmos RPE, BP e RPE quando comparados pelo mean square error. Os autores ainda concluíram que o algoritmo BP teve o pior desempenho que os demais em relação a mesma métrica de acurácia. Dragomir, Dragomir, Brezeanu & Mincă (2011) propuseram em seu trabalho a rede neural MLP como ferramenta de previsão de curto prazo da diferença de energia gerada e consumida de um conjunto de painéis fotovoltaico com dimensões mínimas de 0.4kV/10kW, onde kW equivale a kiloWatts. Os resultados dos autores obtidos a partir de simulações e testes mostraram a precisão e acurácia da estrutura da rede neural MLP em captar dependências não lineares entre os dados de entrada e saída.

Outro modelo de rede neural artificial utilizado para prever demanda de curto prazo de energia é as redes convolucionais. O trabalho de Khan, Khalid, Javed, Javaid, Ahmed & Javaid (2020) as utilizou junto com os modelos MLP e um modelo melhorado das redes convolucionais para efetuar a previsão de demanda de curto prazo de energia e de preços em uma área residencial. Antes de efetuar as previsões, os dados passaram por uma *feature extraction* por intermédio do processo de *mutual information* e uma *feature selection* a partir do *random forest* e *recursive feature elimination*. As métricas *Mean Absolute Percentage Error* (MAPE), *Root Mean Square Error* (RMSE), *Mean Absolute Error* (MAE) e *Mean Squared Error* (MSE) foram utilizadas para avaliar os resultados preditivos. Mediante os valores fornecidos pelos modelos, as redes neurais convolucionais aperfeiçoadas desempenharam-se melhor que os demais.

A probabilidade não é uma ferramenta exclusiva dos métodos estatísticos de previsão, ela pode ser usada também nas redes neurais artificiais para efetuar previsões de demanda de curto prazo de energia, como no trabalho de Nair & Joshi (2010). Nesse artigo os autores comentam que se o mecanismo básico do processo gerador dos dados de demanda de energia for incluído na análise de previsão, uma abordagem estatística pode ser o melhor. Por esse motivo a classe de redes neurais artificiais *Radial Basis Function Networks* foi utilizada no trabalho, pois pode fornecer alguma abordagem estatística. Os autores concluíram que o modelo probabilístico de rede neural artificial forneceu resultados suficientemente bons e com uma margem de erro aceitável, mesmo não utilizando variáveis relacionadas ao clima. Ainda nas palavras dos autores, a inserção dessas variáveis exógenas no modelo iria, sem dúvida, melhorar a acurácia das previsões.

As redes neurais profundas podem ser aplicadas na previsão de demanda de energia no curto prazo, é o que mostra o trabalho de Maksut, Karbozov, Myrzaliyeva, Nunna, Jamwal &

Doolla (2019). Para os autores, o aprendizado de redes neurais profundas, aquelas que possuem inúmeras camadas intermediárias, é considerado um método poderoso na previsão de demanda de eletricidade. Todavia o presente estado de aprendizagem profunda ainda está evoluindo, portanto novas ferramentas para melhorar a precisão da previsão de demanda são significativamente importantes. Sabendo disso os autores propuseram o uso do método *restricted boltzmann machine* para o pré-treinamento e a função de ativação *rectifier linear unit* para melhorar o campo atual da *deep neural network*. Foi concluído que as variáveis exógenas de temperatura e velocidade do vento são cruciais nas variações de demanda de energia residencial em Londres. Entretanto, segundo os autores, esse resultado já era esperado, pois essas variáveis climáticas são quase sempre as mais influenciadoras no consumo de energia ao redor do mundo.

Dos vários modelos de redes neurais artificiais existentes na literatura, um merece atenção especial no problema de previsão de demanda de energia de curto prazo. Esse modelo é o de redes neurais recorrentes, pois sua estrutura foi desenvolvida para trabalhar com dados sequencias conforme visto no referencial teórico, Capítulo 2. O trabalho de Ge, Zhang, Sun & Sun (2010) propõem um modelo *fuzzy* de rede neural dinâmica recorrente para efetuar a previsão de demanda de energia de curto prazo em um sistema de potência. O modelo foi proposto com o intuito de resolver o problema de velocidade de convergência e mínimos locais provenientes do algoritmo clássico de treinamento do *backpropagation* em redes *feedforward*. A função de inferência *fuzzy* foi aplicada facilmente na rede neural recorrente por uma operação de produto. Os resultados da simulação indicam que a rede proposta pode superar o limite de métodos de redes estáticas baseadas em *backpropagation* e prever com precisão a demanda de energia de curto prazo. O modelo proposto dos autores teve uma acurácia de 98,5% nas previsões em 74,6 segundos, enquanto a rede clássica teve 92,8% em 72,5 segundos. A partir dos resultados provenientes da simulação, os autores concluíram que ficou provado que a taxa de convergência e a acurácia das previsões do modelo fuzzy de rede neural dinâmica recorrente foram superiores quando comparadas com o modelo tradicional de redes *feedforward*, pois o problema de ótimos parciais é evitado.

Outro trabalho que utiliza redes neurais recorrentes é o de Oduro-Gyimah (2018) que as aplica para efetuar a previsão de pico de demanda de energia em Ashanti Region. Para obter os resultados preditivos foram utilizados três *Recurrent Neural Network* (RNN) com modelos de arquitetura em camadas (RNN-1-10-1, RNN-1-15-1 e RNN-1-20-1). Os resultados do autor mostraram que o melhor modelo foi o RNN-1-10-1 quando levado em consideração como métrica o erro RMSE. O primeiro modelo com 10 neurônios na camada intermediária foi responsável por um erro de $2.46e-03$, já o segundo modelo com 15 neurônios na camada oculta foi responsável por um erro $7.02e-03$ e, por último, o terceiro modelo com 20 neurônios na camada intermediária forneceu um erro de $2.46e-03$. Com esses resultados verifica-se que a quantidade de neurônios na camada oculta desempenha um papel importante na acurácia dos resultados preditivos. Bui, Nguyen, Pham, Kim & Jang (2020) empregaram as redes neurais recorrentes para prever a demanda de energia com horizonte de uma hora. No seu trabalho foi proposto um modelo multiescalar baseado em RNN que possui tanto memória de curto prazo

quanto de longo prazo. Os autores concluíram que o modelo proposto possui reatividade e alta acuracidade, por isso o potencial para ser uma solução robusta para o problema de previsão de demanda de carga é evidente.

A implementação de memória de longo prazo em redes neurais recorrentes é uma alternativa para lidar com o problema causado pelo *vanishing gradient* durante o treinamento, como visto no referencial teórico (Capítulo 2). Por isso existem vários trabalhos que utilizam as redes neurais LSTM (*Long Short-Term Memory*) na solução de problemas relacionados à previsão de demanda de energia de curto prazo. O artigo de Jiang, Zhu, Li & Qing (2018) é um desses trabalhos que utilizam as redes LSTM para prever o consumo de energia de uma empresa elétrica de larga escala da Estônia. Os autores transformaram os dados originais em dados de aprendizado supervisionado de máquina para que a rede neural de arquitetura LSTM pudesse ser treinada e, então, efetuar as previsões de demanda. O experimento demonstrou que as redes LSTM são eficientes e, quando comparadas com os modelos de *Support Vector Regressive* (SVR), podem extrair dos dados energéticos mais informações que melhoram seu desempenho preditivo com o aumento da acurácia. As previsões de demanda de energia do modelo LSTM foram acompanhadas de um erro RMSE de 38.74, ao passo que o modelo SVR forneceu um erro RMSE de 43.16 nas previsões. Além disso, os autores concluíram que os resultados provenientes das redes recorrentes de arquitetura LSTM podem auxiliar no planejamento elétrico nacional. Outro trabalho que faz um comparativo entre redes LSTM e modelo SVR na previsão de demanda de energia é o artigo de Islam, Al Mamun, Sohel, Hossain & Uddin (2020). No qual o modelo LSTM proposto supera o modelo SVM com a menor taxa de erro RMSE e MAE de 127.682 e 92.448 respectivamente, o que resulta em 52,24% e 21,67% na redução nos erros quando comparados com o modelo SVM.

No trabalho de Liu, Jin, Gu & Qiu (2017) é realizada a previsão de demanda de energia de curto prazo com dois modelos de redes neurais artificiais, o primeiro modelo é a rede neural recorrente sem memória (Elman) e o segundo modelo é a rede com memória LSTM. Foram realizadas previsões em dois casos, sendo que a rede LSTM forneceu melhores resultados preditivos em ambos os cenários quando a métrica utilizada foi o erro MAPE. No primeiro caso a rede LSTM teve erro de 2.57%, enquanto a rede Elman teve erro de 6.92%. No segundo caso a rede LSTM teve erro de 2.13%, já a rede Elman teve erro de 5.19%.

No trabalho de Cheng, Xu, Mashima, Thing & Wu (2017) foi utilizada a rede LSTM para prever a demanda de energia de 114 apartamentos localizados na cidade de Massachusetts. A rede utilizada foi alimentada com as séries históricas da demanda de energia, informações do tempo e calendário, seus resultados preditivos foram melhores quando comparados a outros dois métodos de previsão. O artigo de Berriel, Lopes, André Teixeira Rodrigues, Varejão & Oliveira-Santos (2017) comparou três (3) arquiteturas distintas de redes neurais artificiais para previsão da demanda de energia no Brasil, sendo que o modelo que obteve o melhor desempenho preditivo foi o da rede LSTM. Outros trabalhos recentes que utilizaram redes neurais artificiais de arquitetura LSTM para efetuar previsão de demanda de energia podem ser vistos

em: Kumar, Hussain, Banarjee & Reza (2018); Bouktif, Fiaz, Ouni & Serhani (2018); Marino, Amarasinghe & Manic (2016); Rahman, Srikumar & Smith (2018) e Chandramitasari, Kurniawan & Fujimura (2019).

Cui, He, Di, Lu, Dai & Lv (2020) realizaram seu trabalho baseado em redes neurais artificiais de arquitetura LSTM para efetuar previsões de demanda de curto prazo. Os dados foram transformados em um problema de aprendizado de máquina supervisionado, onde as 48 observações anteriores tinham como objetivo prever a atual. Esses dados tinham como intervalo de tempo 30 minutos entre si e foram normalizados para serem treinados e processados pelas redes. As previsões tinham um horizonte de previsão de 48 passos, ou seja, 24 horas e tiveram como métrica de acurácia o erro MAPE. Antes de obter as efetivas previsões, os autores dividiram os dados em três amostras, sendo elas: amostra de treino com 86,5% do total de observações; amostra de validação 3,5% e amostra de teste 10%. As previsões efetivas foram feitas em dois cenários. O primeiro cenário realiza as previsões *one-step* dos próximos 30 minutos baseados nas informações compreendidas nas últimas 24 horas. Enquanto o segundo cenário realiza previsões *mult-step* que utilizam dados do dia anterior completo para prever a demanda de energia da metade do seguinte dia. O erro MAPE da rede LSTM proveniente do primeiro cenário foi de 1.806% e do segundo cenário foi de 2.496%. Dessa forma os autores concluíram, a partir da comparação com a demanda de energia real, que as previsões fornecidas pela rede LSTM proposta teve efeito significativo na melhora da precisão dos resultados quando comparados com a rede neural de arquitetura GRU (*Gated Recurrent Unit*).

Sabendo que o ruído gerado pela medição dos dados de uma rede de distribuição possui impacto na precisão dos modelos preditivos de demanda de energia, Ma, Zhang, Ding, Wang & Ma (2019) propõem em seu trabalho um método baseado em *Isolation Forest* (iForest) e rede neural artificial LSTM para efetuar previsões de demanda de energia de curto prazo. Na primeira etapa o algoritmo iForest é usado para minerar e limpar os dados históricos anormais de carga. Na segunda etapa um modelo de previsão é estabelecido com base na rede neural LSTM com aprendizagem profunda (várias camadas intermediárias). Na terceira etapa o iForest-LSTM é formado e, em seguida, aplicado ao problema de previsão de demanda de energia de curto prazo. Por último, os resultados preditivos do método proposto foram comparados com as previsões da rede neural LSTM padrão e o iForest-BP. A comparação é realizada por intermédio do erro MAPE ao longo de sete dias onde cada um representa um cenário distinto. Em todos os cenários o método proposto iForest-LSTM desempenhou-se melhor que os outros dois modelos preditivos. Em segundo lugar, também em todos os cenários, ficou a rede neural LSTM padrão. Já o método iForest-BP teve o pior desempenho nos sete cenários possíveis. Desse modo os autores concluíram que a limpeza de dados históricos provenientes da demanda de energia combinado com algoritmos de aprendizado profundo podem efetivamente melhorar a acurácia das previsões de curto prazo.

Li, Sun, Li, Ma & Zhang (2019) realizaram um trabalho envolvendo aprendizado profundo em redes neurais LSTM para prever a demanda de energia de um sistema combinado

de resfriamento, aquecimento e energia, pois, segundo os autores, as previsões desempenham um importante papel no controle e otimização desse sistema específico. Além disso, a precisão dos valores previstos afeta diretamente as estratégias de controle e a eficiência energética do sistema. O coeficiente de correlação de Person foi utilizado para verificar que a série temporal multivariada composta pela carga de resfriamento, carga de aquecimento e carga elétrica é tipicamente caótica pelo fato de suas variáveis não serem afetadas somente por elas próprias, mas, também, umas pelas outras. Os erros RMSE e MAE foram utilizados para comparar os resultados preditivos do modelo proposto da rede LSTM multivariada (exógena) com o modelo LSTM univariável, o modelo de rede neural *Radial Basis Function* e o modelo *Support Vector Machine*. O método proposto teve melhor precisão em seus resultados preditivos de demanda de energia, pois, segundo os autores, a rede LSTM com entrada exógena tem a capacidade de modelar dependência de longo prazo de maneira efetiva, extraíndo recursos inerentes importantes de dado históricos automaticamente. Logo essa ferramenta de previsão demonstrou-se eficiente para o sistema combinado de resfriamento, aquecimento e energia.

No trabalho de Hossain & Mahmood (2020) são desenvolvidos dois modelos de previsão baseados em redes neurais de arquitetura LSTM para efetuar previsões de demanda de energia de curto prazo. O primeiro modelo prevê a demanda de forma *one-step ahead* e o segundo modelo realiza a previsão de forma *multi-step ahead* dentro de um dia para diferentes horizontes de previsão. A série temporal de demanda de energia é usada junto com os dados meteorológicos da área geográfica considerada. As informações referentes ao horário das observações, os feriados e os dias da semana foram colocados em um vetor e introduzidas nas redes neurais junto com os dados históricos, o que mostrou um aumento significativo na acurácia das previsões. Os resultados da rede neural LSTM proposta foram comparados com as previsões de duas outras redes neurais artificiais conhecidas como: GRNN (*Generalized Regression Neural Network*) e ELM (*Extreme Learning Machine*). Essa comparação foi feita baseada nos erros MAE, RMSE e MAPE, onde nos dois cenários (*one-step ahead* e *multi-step ahead*) a rede LSTM proposta teve um desempenho melhor que os modelos GRNN e ELM considerado os três erros. Agora, comparando os dois cenários entre si, os três modelos com previsões *one-step ahead* foram muito mais eficientes que os três modelos com previsões *multi-step ahead* para as três métricas de acurácia supracitadas.

Li, Wang, Zhou, Zhang & Zhao (2018) propuseram em seu trabalho um modelo para previsão de demanda de energia de curto prazo baseado em redes neurais artificiais de arquitetura LSTM com decomposição e otimização. A série histórica original foi decomposta em componentes relativamente estacionárias pela *Empirical Mode Decomposition* (EMD) para que a rede LSTM conseguisse extrair mais informações relevantes durante seu treinamento para aumentar a acurácia das previsões. Além disso, os parâmetros da rede neural foram otimizados por intermédio do algoritmo *Particle Swarm Optimization* (PSO). Os resultados preditivos do método proposto denominado EMD-PSO-LSTM foi, então, comparado com as previsões dos seguintes modelos: LSTM; EMD-LSTM; PSO-LSTM; RNN e GRU. Como o método proposto EMD-PSO-LSTM combina as vantagens do EMD e PSO, sua curva de previsão é a mais pró-

xima da demanda de energia real e os erros pontuais de cada previsão estão abaixo de 3%, e o mais baixo foi de 0.0833%. Com uma acurácia de 98.5726% em um horizonte de previsão de 24 horas o método proposto é uma ferramenta poderosa para previsão de demanda de energia de curto prazo.

Outro trabalho que é utilizado a técnica *Empirical Mode Decomposition* (EMD) na série histórica de demanda de energia para efetuar previsões de valores futuros por intermédio de redes neurais de arquitetura LSTM é o de Zheng, Yuan & Chen (2017). O estudo dos autores apresenta um algoritmo híbrido que é composto por *similar days* (SD) *selection*, EMD e redes LSTM para construir um modelo de previsão de energia de curto prazo que é denominado pelas siglas supracitadas, SD-EMD-LSTM. O algoritmo *extreme gradient boosting-based weighted k-means* foi utilizado com o intuito de avaliar as similaridades entre as previsões fornecidas pelo modelo proposto e os dados reais correspondentes à necessidade energética. O método EMD é empregado para decompor a série temporal elétrica SD em uma quantidade considerável de *intrinsic mode functions* (IMFs) e resíduos. De maneira separada, distintas redes neurais LSTM foram utilizadas para prever cada um dos IMFs e resíduos para que, então, os valores previstos de cada modelo LSTM fosse reconstruído. Os autores concluíram que o método SD-EMD-LSTM pode prever demanda de energia elétrica de forma eficiente a partir dos resultados provenientes dos testes numéricos. Zhang, Ma, Li, Ma & Ding (2020) também utilizaram o método EDM para realizar previsões de demanda de energia elétrica utilizando redes neurais de arquitetura *Long Short-Term Memory*. Já no trabalho He, Zhou, kai Feng, Liu & Yang (2019) também foi aplicada as redes recorrentes do tipo LSTM para obter valores preditivos relacionados à demanda de energia de curto prazo; porém, ao invés de utilizar a técnica *Empirical Mode Decomposition* na série temporal, foi aproveitado o seu método derivado conhecido como *Variational Mode Decomposition* (VMD).

Além do *Empirical Mode Decomposition*, a decomposição *Wavelet* é outro método de decomposição de séries históricas relacionadas à demanda de energia muito utilizado em trabalhos de cunho científico. Imani & Ghassemian (2019) empregaram as redes neurais de arquitetura LSTM como um método de aprendizado profundo para criar um modelo preditivo eficiente que é alimentado por características importantes da série temporal. A decomposição *Wavelet* foi utilizada para desempenhar essa função, pois a técnica removeu detalhes redundantes da série histórica original. Com a extração dessas componentes, o modelo de previsão proposto teve uma melhor performance no tocante às previsões de demanda de energia residencial de curto prazo.

Yan, Li, Ji, Qi & Du (2019) realizaram previsões do consumo de energia elétrica em casas residenciais individuais utilizando um modelo híbrido construído a partir das redes LSTM combinada com a técnica de decomposição *stationary wavelet transform* (SWT). Segundo os autores, a SWT reduz a volatilidade da série temporal e aumenta a possibilidade de extrair informações úteis, dessa forma as chances de melhorar o desempenho preditivo das redes LSTM são significativas. Os Experimentos de verificação do trabalho foram realizados com base em um

conjunto de dados de consumo de energia doméstico do mundo real. Os resultados dos autores mostram que, com uma eficiência de treinamento competitiva, o método proposto supera todos os métodos comparados de última geração, incluindo o método *persistent* de previsão, para diferentes *step sizes* de 5, 10, 20 e 30 minutos em três métricas de erro distintas. Esses métodos de última geração são: *support vector regression* (SVR), *long short term memory* (LSTM) e *convolutional neural network* combinada com *long short term memory* (CNN-LSTM). Outros trabalhos que utilizam a decomposição wavelet para realizar previsões de demanda de energia podem ser vistos em Sudheer & Suseelatha (2015), Zhai (2015) e Rafiei, Niknam, Aghaei, Shafie-Khah & Catalao (2018).

Outra técnica de decomposição de séries temporais encontrada na literatura no tocante à previsão de demanda de energia é conhecida como *complete ensemble empirical mode decomposition adaptive noise* (CEEMDAN). Zhang & Hong (2019) a utiliza em seu trabalho para efetuar o pré-processamento dos dados com o intuito de melhorar a acuracidade das previsões. Os autores declaram que o *modelo support vector regression* (SVR) pode ser combinado com novos algoritmos metaheurísticos não apenas para identificar flutuações e as tendências não lineares de cargas elétricas, mas, também, para gerar previsões satisfatórias. Entretanto, ainda nas palavras de Zhang & Hong (2019), muitos desses algoritmos possuem uma quantidade considerável de desvantagens, entre elas uma baixa diversidade populacional e encontro de ótimos locais, que, basicamente, são problemas de convergência precoce. Consequentemente, abordagens para aumentar a precisão da previsão devem ser desenvolvidas. Consequentemente, abordagens para aumentar a precisão da previsão devem ser desenvolvidas. Por essa razão os autores recorreram aos mecanismos da computação quântica para “quantimizar” o comportamento da libélula (*dragonfly*) para aumentar a eficiência da procura do algoritmo *dragonfly*, conhecido como QDA (*Quantum dragonfly algorithm*). Dessa forma um novo modelo de previsão de demanda de energia elétrica é proposto com o objetivo de fornecer previsões precisas, definido como CEEMDAN-SVRQDA. Esse modelo é formado pela combinação do método CEEMDAN e o modelo híbrido SVR e QDA. Dois exemplos numéricos da Tokyo Electric Power Company e da National Grid, a primeira situada no Japão enquanto a segunda é localizada no Reino Unido, demonstram que o modelo proposto supera outros 12 modelos preditivos, sendo 3 destes: ARIMA (*autoregressive integrated moving average*); SARIMA (*Seasonal autoregressive integrated moving average*) e SVR.

Chen, Huang, Wu & Ouyang (2020) também usufruiu do método CEEMDAN para realizar previsões que tangem à demanda de eletricidade no curto prazo por intermédio do modelo preditivo *stacked bidirectional gated recurrent unit* (SBiGRU). A acuracidade preditiva do modelo combinado foi avaliada em dois conjuntos de dados energéticos públicos, no qual os resultados provenientes dos experimentos mostraram que a combinação do modelo foi responsável por uma melhor eficiência e estabilidade quando comparada com o modelo tradicional SBiGRU.

Improved complete ensemble empirical mode decomposition with adaptive noise (ICE-

EMDAN), como o nome já sugere, é uma técnica melhorada baseada na CEEMDAN e pode ser vista no trabalho de Li, Qian & He (2020). Onde é afirmado que pesquisas recentes mostram que métodos conhecidos como empirical mode decomposition são poderosíssimos no que concerne à previsão de demanda de energia. Com o objetivo de aumentar a eficácia dos valores preditivos advindos de modelos de previsão de demanda de energia de curto prazo, os autores propõem uma nova abordagem integrando a técnica ICEEMDAN, *grey wolf optimization* (GWO) e *multiple kernel extreme learning machine* (MKELM), batizada de ICEEMDAN-GWO-MKELM. O método proposto é baseado em 3 estágios distintos; sendo que o primeiro consiste na decomposição dos dados de carga brutos e complexos em algumas componentes relativamente simples mediante a técnica ICEEMDAN; já o segundo estágio é encarregado de usar o modelo MKELM para prever todas as componentes decompostas do passo anterior individualmente, onde foi utilizado o GWO para otimizar os pesos e os parâmetros de cada kernel na máquina de aprendizado extremo para melhorar a capacidade de previsão; e, por último, os resultados de todas as componentes são agregadas como o produto final da previsão. Os extensos experimentos dos autores revelam que o ICEEMDAN-GWO-MKELM pode superar várias abordagens de previsão de última geração em termos de alguns critérios de avaliação, mostrando que o modelo proposto é muito eficaz para previsão de demanda de curto prazo.

Após a verificação da possibilidade de técnicas de decomposição em séries temporais relacionadas ao consumo de energia elétrica aumentar a eficácia de modelos preditivos, conforme os artigos supracitados, foram aplicadas duas delas no conjunto de dados do presente trabalho. As técnicas utilizadas foram a CEEMDAN e a sua versão melhorada, ICEEMDAN. A escolha delas foi baseada na tentativa de criar um modelo “inédito” de previsão de demanda de energia de curto prazo, no qual a série temporal de carga seria decomposta por uma dessas técnicas e cada componente seria alimentada em uma rede neural recorrente de arquitetura LSTM para que, então, as previsões individuais fossem integradas para formar a previsão final. O modelo seria chamado de CEEMDAN-LSTM ou ICEEMDAN-LSTM, dependendo do método de decomposição adotado, e seria “inédito” pelo fato de não ter sido encontrado nenhum trabalho com essa metodologia aplicado à demanda de energia elétrica no *Google Scholar*, embora o método proposto foi usado em uma série histórica financeira, conforme o trabalho de Lin, Sun & Chen (2020). Entretanto, ao aplicar ambas as técnicas de decomposição, uma de cada vez, na série temporal do presente trabalho, o resultado não foi o esperado, pois algumas intrinsic mode functions, componentes criadas a partir das decomposições, tiveram uma volatilidade tão alta que os modelos de previsão LSTM não conseguiram modelá-las. Com isso os resultados preditivos foram ineficientes e muito distantes dos valores reais da série histórica de demanda de energia elétrica.

Outra técnica recente que está sendo utilizada na previsão de demanda de energia elétrica são as redes neurais recorrentes *Bidirectional*. Basicamente essas redes são construídas conectando duas camadas ocultas de direções opostas à mesma saída, que resulta na camada de saída da rede para obter informações dos estados passado e futuro concomitantemente. Du, Cheng, Zhou, Zhang, Zhang & Li (2020) usufruiu dessa técnica para propor seu método deno-

minado de Attention-BiLSTM (*Attention based Bidirectional Long Short-Term Memory*) para fazer previsões precisas de demanda de energia de curto prazo, devido às limitações dos métodos tradicionais de previsão de carga para lidar com dados de séries temporais não lineares em grande escala. Este modelo preditivo é baseado na rede neural recorrente BiLSTM que tem alta robustez no tempo, modelagem de dados em série e mecanismo de atenção que pode destacar os principais recursos desempenhando papéis importantes na previsão de demanda de energia nos dados de alimentação. Os experimentos de verificação dos autores com dados reais em uma determinada área geográfica mostram que o modelo proposto supera outros modelos (A-LSTM, BiLSTM e LSTM) em termos de precisão de previsão e robustez do algoritmo.

A rede BiLSTM pode ser vista, também, em Zhou, Wu & Zhang (2020) na tentativa de prever a demanda de energia de curto prazo. Segundo os próprios autores, essa ferramenta de auxílio na tomada de decisões sempre foi muito importante na operação da rede elétrica, por isso melhorar a precisão das previsões tem grande significado no aprimoramento da eficiência de despacho de empresas do setor elétrico de energia e, conseqüentemente, nos seus benefícios econômicos. Tendo em vista que a rede neural do tipo *feedforward* não tem capacidade para aprender a relação da série temporal de demanda de energia, um modelo de profundidade híbrido baseado em *improved bacterial foraging algorithm* (IBFO) e redes neurais recorrentes de arquitetura *Bidirectional LSTM* é proposto para efetuar previsões de demanda de energia de curto prazo. O trabalho emprega dados históricos da necessidade energética dos consumidores e seus dados associados de temperatura e informações como entrada, os pesos iniciais e o *bias* são otimizados por *improved bacterial foraging algorithm*, e a rede neural LSTM bidirecional multi-camada é usada para extrair a relação oculta da série temporal entre os dados de entrada e, finalmente, produzir predição. O modelo de previsão proposto foi verificado usando os dados históricos de carga de Nantong e seus resultados mostram possuir maior precisão de predição do que os modelos ARIMA, SVM e MLP.

Além de serem utilizadas nas previsões de demanda de energia de curto prazo, a rede *Bidirectional LSTM* pode ser aplicada na estimativa futura da necessidade energética de longo prazo. No trabalho de Petroşanu & Pîrjan (2021) é constatado que a previsão precisa do consumo de eletricidade por hora no mês anterior representa um aspecto muito importante para os consumidores de eletricidade não domésticos e operadores de sistema, e ao mesmo tempo, representa um fator chave no que diz respeito à eficiência energética e ao alcance de operações econômicas, comerciais e de gestão. Neste contexto, os autores idealizaram, desenvolveram e validaram o método de previsão do consumo de eletricidade no estudo com um mês de antecedência. Este método é baseado em uma rede neural artificial (ANN) bidirecional de memória de longo-curto prazo (BiLSTM) aprimorado com uma abordagem de múltiplos atrasos decrescentes simultaneamente, juntamente com o ajuste de função redes neurais (FITNETs). O método desenvolvido pelos autores visa o consumo da eletricidade total por hora do mês anterior no nível de um consumidor do tipo centro comercial e para o consumo do mês de hora em hora seguinte da sala de armazenamento do refrigerador. A abordagem desenvolvida no trabalho oferece excelentes resultados de previsões, destacados pelos resultados do estágio de validação

junto com as métricas de desempenho registradas, a saber, 0,0495 para a métrica de desempenho do erro quadrático médio (RMSE) para o mês do horário total consumo de eletricidade e 0,0284 para a despesa do refrigerador. Os autores conseguiram atingir uma previsão de eletricidade consumida de hora em hora sem experimentar uma queda significativa na precisão da previsão que geralmente tende a ocorrer após as primeiras duas semanas, alcançando assim um método confiável que satisfaça as necessidades do contratante, sendo capaz de aprimorar sua atividade do ponto de vista econômico, empresarial e de gestão. Mesmo se a solução de previsão para o consumo por hora planejado, desenvolvido e validado almeja um consumidor comercial tipo center, com base em sua precisão, esta solução também pode representar uma ferramenta útil para outros consumidores não residenciais de eletricidade devido à sua capacidade de generalização.

A técnica *Bidirectional* aplicadas às redes neurais recorrentes não é exclusiva das arquiteturas LSTM, as redes de memória de longo prazo denominadas de *gated recurrent unit* (GRU) podem, da mesma forma, serem beneficiadas por esse procedimento no que tange às previsões de demanda de energia de curto prazo, conforme o trabalho de Massaoudi, Refaat, Chihi, Trabelsi, Abu-Rub & Oueslati (2020). Os autores comentam que a acurácia da previsão de carga de curto prazo tem sido considerado um tema de extrema importância para gestão de energia, transações confiáveis de energia e despacho de operação em *smart grids*. No entanto, a instabilidade contínua da demanda de energia elétrica devido, essencialmente, à alta volatilidade das condições climáticas e do comportamento da necessidade dos clientes afeta drasticamente a precisão das previsões no curto prazo. A fim de superar este problema, cinco técnicas eficazes de *Deep Learning* (DL) são propostas para séries temporais multivariadas aplicadas às previsões de demanda de energia de curto prazo com base na *Convolutional Neural Network* (CNN), *Gated Recurrent Unit* (GRU), *Long Short-Term Memory* (LSTM), e *stacked Auto-Encoder* (AE). Essas técnicas baseadas em DL são empregadas para construir os seguintes modelos preditivos *stacked Bidirectional GRU* (BiGRU), *Convolutional LSTM* (ConvLSTM), *stacked Bidirectional LSTM-AE* (BiLSTM-AE), *hybrid CNN-LSTM-AE* (CNNLSTM), e *LSTM-AE* (LSTM-AE). Os autores efetuaram estudos baseados em simulações para demonstrar o desempenho superior do modelo BiLSTM-AE em comparação com os outros modelos DL. As principais contribuições do artigo incluem 1) a integração de uma variedade de redes neurais profundas para previsão de demanda de energia de curto prazo; 2) empregar séries temporais como uma referência para comparação entre modelos de DL construídos de arquiteturas heterogêneas; 3) conduzir as análises em conjuntos de dados reais.

O estudo de Wu et al. (2021) é mais um que utiliza redes neurais artificiais de camadas de diferentes arquiteturas para efetuar previsões de demanda de energia elétrica de curto prazo. Os autores afirmam que nos últimos anos diversos tipos de energias foram integradas ao sistema de energia, que constitui um *regional integrated energy system* (IES). No entanto, o acoplamento e complementação de múltiplas fontes de energia tornam a previsão de carga energética mais difícil. Devido a sequência de tempo e as características não lineares da carga de eletricidade, bem como a complementaridade de diferentes energias em IES, o artigo propôs o seguinte modelo

preditivo na demanda de energia de curto prazo de um (IES), *attention-based convolutional neural network* (CNN) combinada com *long short-term memory* (LSTM) e *bidirectional long short-term memory* (BiLSTM). A série histórica de demanda de energia, temperatura, carga de resfriamento e consumo de gás dos últimos 5 dias são usados como os recursos de entrada da rede neural artificial. A rede CNN integrado com bloco de atenção é utilizado para extrair recursos eficazes dos fatores de impacto da carga. Então a carga da próxima hora é prevista pela LSTM combinado com camadas BiLSTM. Finalmente, o modelo foi verificado pelos dados de um parque integrado de energia no norte da China. Os resultados dos autores mostram que o método proposto tem melhor desempenho de previsão do que CNNBiLSTM, CNN-LSTM, BiLSTM, LSTM, rede neural de retropropagação (BPNN), regressão de floresta aleatória (RFR) e regressão de máquina de vetor de suporte (SVR).

O artigo de Massaoudi, Refaat, Abu-Rub, Chihi & Oueslati (2020) propõe uma estrutura de aprendizado profundo eficaz para previsão de demanda de energia no curto prazo de séries temporais multivariadas. O modelo híbrido proposto SG-CBiLSTM consiste em *Convolutional neural network-Bidirectional Long Short-Term Memory* (CBiLSTM) com base no método *Evolution Strategy* (ES) e o filtro *Savitzky-Golay* (SG). A metodologia adotada incorpora a virtude de diferentes blocos atraentes para melhorar o desempenho do modelo CBiLSTM. Em particular, uma estratégia de aumento de dados é empregada para melhorar sinteticamente a representação de recursos do modelo CBiLSTM. Os dados aumentados são encaminhados para o método *Partial Least Square* (PLS) para selecionar os recursos mais informativos acima do limite predefinido. Em seguida, o algoritmo SG é calculado para suavizar a carga para aprimorar os recursos de aprendizagem do sistema subjacente. A estrutura do SG-CBiLSTM para o conjunto de dados da ISO New England é otimizada usando a técnica ES. Finalmente, o modelo CBiLSTM gera previsões de saída. A abordagem proposta pelos autores demonstra uma notável melhoria no desempenho do modelo CBiLSTM original. Além disso, o experimental resultados confirmam fortemente a alta eficácia do modelo SG-CBiLSTM proposto em comparação com as técnicas de última geração.

Já Chitalia, Pipattanasomporn, Garg & Rahman (2020) apresenta em seu artigo uma estrutura robusta de previsão de carga elétrica de curto prazo que pode capturar variações em operação do edifício, independentemente do seu tipo e localização. Nove tipos diferentes de redes neurais recorrentes híbridos e os seus agrupamentos são explorados, sendo eles: LSTM; LSTM with attention; BiLSTM; BiLSTM with attention; CNN + LSTM; CNN + BiLSTM; ConvLSTM; ConvBiLSTM e Encoder-Decoder. Os casos de teste envolvem cinco edifícios comerciais de cinco tipos diferentes, ou seja, acadêmico, laboratório de pesquisa, escritório, escola e supermercado, localizados em cinco locais diferentes em Bangkok Tailândia, Hyderabad Índia, Virgínia-EUA, Nova York-EUA e Massachusetts-EUA. Os resultados da previsão de carga dos autores indicam que os algoritmos de aprendizado profundo implementados no artigo oferecem uma melhoria de 20 a 45% no desempenho da carga prevista em comparação com os resultados atuais do estado da arte, tanto para horas de antecedência quanto para 24 horas de antecedência na previsão de carga. No que diz respeito à análise de sensibilidade, verifica-se

que: (i) o uso de algoritmos de aprendizado profundo híbrido pode levar menos de um mês de dados para fornecer previsão de carga com hora de antecedência satisfatória, (ii) semelhante a técnica de agrupamento, dados de resolução de 15 minutos, se disponíveis, fornecem 30% de melhoria na previsão de carga com horas de antecedência e (iii) os métodos formulados são considerados robustos contra erros de previsão do tempo. Por último, os resultados da previsão em todos os cinco edifícios validam a robustez da estrutura de aprendizagem profunda proposta para as tarefas de previsão de carga elétrica de curto prazo.

Observando a relevância atual das redes neurais artificiais constituídas a partir de arquiteturas heterogêneas, da mesma maneira que a técnica *Bidirectional* aplicadas em redes recorrentes de memória de longo prazo, a presente dissertação visa efetuar previsões de demanda de energia de curto prazo com os seguintes modelos propostos BiLSTM-BiGRU, BiGRU-BiLSTM, CNN-BiGRU e BiGRU-CNN. A escolha das camadas neurais na construção das redes foi baseada em uma pesquisa onde não foram encontrados artigos científicos publicados relacionados ao assunto, pelo menos até o momento das simulações computacionais. Esse procedimento foi adotado para contribuir com a comunidade acadêmica e científica na geração de conhecimento voltados a métodos de previsão de demanda de energia de curto prazo.

Capítulo 3

Materiais e Métodos

Neste capítulo é apresentado detalhadamente a série temporal e os métodos utilizados neste estudo para a obtenção dos resultados preditivos de demanda de energia de curto prazo das redes neurais de arquitetura MLP, RNN, GRU, LSTM, BiLSTM-BiGRU, BiGRU-BiLSTM, CNN-BiGRU, BiGRU-CNN e Ceemdan-LSTM-GRU.

3.1 Séries Temporais

Uma série temporal é qualquer conjunto de dados observados de maneira ordenada no tempo Morettin & Toloí (1987). O presente estudo é composto por uma (1) série histórica que foi utilizada para efetuar previsões de demanda de energia elétrica.

A série temporal de demanda de energia elétrica pertence à empresa *Duke Energy* e é composta por 44.553 observações horárias realizadas entre as 13:00 horas do dia primeiro de outubro de 2012 até a 01:00 hora do dia onze de outubro de 2017. A Figura 3.1 apresenta os dados de demanda de energia ao longo do tempo, no qual o eixo das ordenadas é dado em megawatt (MW) e o eixo das abscissas é cada observação temporal dada em horas. O arquivo em formato *csv* (*Comma-separated values*) que contém os dados provenientes da série temporal foi obtido no seguinte endereço eletrônico: <https://www.kaggle.com/robikscube/hourly-energy-consumption>.

Analisando a Figura 3.1 é possível assumir que a série histórica de demanda de energia é sazonal e/ou cíclica, e nela não existe tendência de crescimento ou decrescimento. Seus dados, com valor máximo de 5308 MW e valor mínimo de 1896 MW, obtidos provenientes do arquivo *csv*, estavam completos, ou seja, todas as medições de potência foram aferidas corretamente nos intervalos de tempo predeterminados, logo não foi necessário recorrer à ferramenta estatística de interpolação para determinar os valores faltantes.

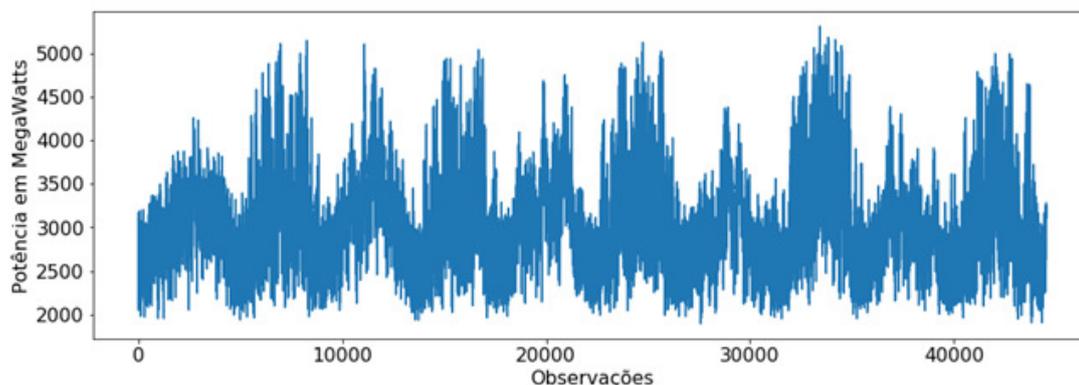


Figura 3.1: Visão geral da série histórica da demanda de energia intervalada em 1 hora
Fonte: (Autor, 2020).

3.2 Metodologia

3.2.1 Previsão de Demanda de Energia de Curto Prazo

Os modelos de redes neurais artificiais apresentados neste trabalho de dissertação foram escolhidos por dois critérios específicos. O primeiro critério é a utilização de redes clássicas já bem conhecidas no que tange à previsão de demanda de energia de curto prazo, sendo elas: MLP, RNN, GRU e LSTM. O segundo critério envolve a mistura de camadas dessas redes clássicas para verificar se esse processo é capaz de melhorar os resultados preditivos quando comparadas com as redes clássicas. Esses novos modelos neurais de previsão, formados por camadas de diferentes arquiteturas, foram construídos a partir de pesquisas no google acadêmico que envolviam a aglutinação dos termos das redes clássicas somada ao termo previsão de demanda de energia.

Os modelos escolhidos através do critério de número 2 são os seguintes: BiLSTM-BiGRU, BiGRU-BiLSTM, CNN-BiGRU, BiGRU-CNN e Ceemdan-LSTM-GRU. Esses modelos e mais os clássicos definidos pelo critério de número 1 foram utilizados para efetuar previsões de demanda de energia de curto prazo em um horizonte preditivo de 24 horas, onde cada previsão foi realizada em um intervalo de uma hora. Logo foram necessárias 24 previsões para se alcançar o horizonte preditivo. Os valores futuros da necessidade energética no curto prazo foram obtidos da seguinte maneira: realizou-se apenas uma simulação para treinar e validar cada um dos diferentes modelos preditivos; após serem treinados e validados, esses modelos realizaram as previsões *multi-step* de forma recursiva. Nesse tipo de previsão os valores futuros são realimentados nas redes neurais artificiais como se fossem observações da amostra de treino ou de validação, com isso é evitado que o modelo seja treinado a cada nova previsão. Em outras palavras, o modelo de previsão recursivo é treinado e validado uma única vez para efetuar 24 previsões efetivas, o que gera uma economia de tempo considerável devido ao tempo necessário no treinamento das redes.

A Equação (3.1) mostra como é realizada a previsão efetiva do primeiro valor da amostra de teste, onde y_t é a resposta da rede treinada e validada ψ para um conjunto de entradas x , e t é um instante de tempo qualquer. Já a Equação (3.2) apresenta como é obtido o segundo valor preditivo da amostra de teste y_{t+1} . Observando-a, é verificado que a previsão efetiva da Equação (3.1), y_t , é parâmetro da rede ψ , a mesma rede da Equação (3.1), nesse momento a recursividade já é utilizada. A Equação (3.3) mostra as previsões efetivas de número um e dois da amostra de teste como argumentos do modelo preditivo ψ para encontrar a terceira previsão. Fazendo esse processo sucessivamente até se obter a última previsão da amostra de teste através da Equação (3.4), verifica-se que o ultimo valor preditivo relacionado á demanda de energia está em função de todas as previsões precedentes. Isso pode se tornar um grande problema quando os primeiros valores futuros não forem eficientes, pois os erros são propagados para as previsões subsequentes. Tendo em vista essa possibilidade, realizou-se 10 vezes cada simulação para que esse cenário seja mitigado, no qual as previsões efetivas finais foram dadas pela média aritmética simples de cada simulação individual. Cada previsão de um valor futuro de demanda de energia elétrica foi baseado em 24 observações anteriores, Cui et al. (2020) e Li et al. (2019) também realizaram previsões de demanda de energia de curto prazo com um janelamento de um dia anterior (24 horas) para séries temporais distintas.

$$y_t = \psi(x_{t-1}, x_{t-2}, x_{t-3}, \dots, x_{t-22}, x_{t-23}, x_{t-24}) \quad (3.1)$$

$$y_{t+1} = \psi(y_t, x_{t-1}, x_{t-2}, \dots, x_{t-21}, x_{t-22}, x_{t-23}) \quad (3.2)$$

$$y_{t+2} = \psi(y_{t+1}, y_t, x_{t-1}, \dots, x_{t-20}, x_{t-21}, x_{t-22}) \quad (3.3)$$

$$y_{t+23} = \psi(y_{t+22}, y_{t+21}, y_{t+20}, \dots, y_{t+2}, y_{t+1}, y_t) \quad (3.4)$$

Após a série temporal ser transformada em um problema de aprendizado de máquina supervisionado, os dados de demanda de energia foram divididos em três amostras distintas. A primeira amostra foi a de treinamento formada pelos primeiros 36.199 dados; a segunda amostra foi a de validação composta pelos 8.330 dados posteriores à amostra de treino; e, por último, a amostra teste foi construída somente pelos 24 dados restantes.

A amostra de treinamento teve como objetivo o aprendizado das redes neurais artificiais dos padrões da série temporal de demanda de energia elétrica. Os parâmetros internos das redes neurais encontrados durante o treinamento foram testados na amostra de validação para verificar a capacidade preditiva da rede em dados não vistos na etapa anterior. Caso o desempenho das redes fosse similar nas amostras de treino e de validação, a amostra teste seria utilizada para avaliar as previsões efetivas de demanda de energia. Entende-se com previsões efetivas aquelas

que são utilizadas para fins práticos, já as previsões realizadas nas amostras de treinamento e validação servem para averiguar se os parâmetros das redes neurais encontrados ao longo do treinamento são capazes de generalizar resultados.

As previsões nas amostras de treino e validação foram efetuadas com o auxílio da Equação (3.1) na modalidade *one step ahead*, onde as previsões eram feitas exclusivamente em função das observações já conhecidas. Por outro lado, as previsões efetivas realizadas na amostra de teste foram obtidas na forma *multi step ahead* por intermédio da recursividade que começa na Equação (3.2) e termina na Equação (3.4), cujas previsões foram baseadas em previsões anteriores. As medidas de acurácia MAPE (*Mean Absolute Percentage Error*), MAE (*Mean Absolute Error*) e RMSE (*Root Mean Square Error*) apresentadas pelas Equações (3.5), (3.6) e (3.7), respectivamente, foram utilizadas para avaliar as previsões fornecidas por todos os modelos de redes neurais artificiais nas amostras de treino, validação e teste. Já a medida de acurácia MSE (*Mean Square Error*) mostrada na Equação (3.8) foi utilizada somente na amostra de teste para avaliar as previsões efetivas individuais. Onde x é o valor desejado, y é o valor previsto e n é a quantidade de elementos da amostra.

$$MAPE = \frac{100\%}{n} = \sum_{i=1}^n \left| \frac{x_i - y_i}{y_i} \right| \quad (3.5)$$

$$MAE = \frac{1}{n} = \sum_{i=1}^n |x_i - y_i| \quad (3.6)$$

$$RMSE = \sqrt{\frac{1}{n} = \sum_{i=1}^n (x_i - y_i)^2} \quad (3.7)$$

$$MSE = \frac{1}{n} = \sum_{i=1}^n (x_i - y_i)^2 \quad (3.8)$$

3.2.2 Código das Redes

As linhas de código utilizadas no presente trabalho foram escritas na linguagem de programação *Python* versão 3.7.6 por intermédio do IDE (*Integrated Development Environment*) *Jupyter Notebook* e rodadas em uma máquina com as seguintes configurações: Processador Intel® Core™ i7-6500U CPU @ 2,5 GHz; memória RAM de 12 GB e sistema operacional Windows 10 de 64 bits.

Python surgiu nas últimas décadas como uma ferramenta de primeira classe para tarefas de computação científica (Vanderplas, 2016). Parte de seu sucesso como plataforma de computação científica é pela facilidade de integração com códigos em C, C++ e *FORTRAN*;

pois é possível utilizar bibliotecas dessas linguagens de programações para resolver problemas de álgebra linear, otimização, integração, transformada rápida de Fourier e outros algoritmos (McKinney, 2018). Python combina o poder das linguagens de programação de uso geral com a facilidade de uso de linguagens de script de domínio específico, como MATLAB e R (Müller & Guido, 2016), esse foi o motivo principal para a linguagem Python ter sido escolhida para a realização deste trabalho. Entretanto Python é uma linguagem de programação interpretada que em geral será executada substancialmente mais lenta do que um código escrito em uma linguagem compilada como Java ou C ++ (McKinney, 2018), essa foi uma das razões para que as redes neurais artificiais demorassem consideravelmente para efetuar as previsões de demanda de energia elétrica após o treinamento e validação dos modelos.

Como já mencionado anteriormente, o uso dos pacotes teve extrema importância na obtenção dos resultados preditivos, sendo eles: *NumPy*, *Matplotlib*, *Pandas*, *Sklearn*, *Keras*, *PyEMD*. O pacote fundamental para computação científica é o *Numerical Python (NumPy)* devido seus recursos rápidos de processamento de matrizes que tornam a manipulação e armazenamento de dados muito mais eficiente que qualquer outra função *built-in* em *Python* (McKinney, 2018). *Matplotlib* é o principal pacote de plotagem científica em *Python* fornecendo funções para realizar visualizações de qualidade como gráfico de dispersão, gráfico de linhas e histogramas (Müller & Guido, 2016). Já o pacote *Pandas* foi construído baseado na biblioteca *NumPy* e fornece uma implementação eficiente de *DataFrames*, que são essencialmente matrizes multidimensionais que possuem rótulos nas colunas e linhas (Vanderplas, 2016). *Keras* é uma API (*Application Programming Interface*) de aprendizado profundo de alto nível que permite criar, treinar, avaliar e executar facilmente muitos tipos de redes neurais artificiais (Geron, 2019). E, por último, o pacote *PyEMD* que, de acordo com Laszuk (2017), é mais uma implementação *Python* do *Empirical Mode Decomposition (EMD)*, o qual contém muitas variações de EMD, algumas delas são o *Ensemble EMD (EEMD)* e *Complete Ensemble EMD (CEEMDAN)*. Além disso o pacote conta com splines diferentes para envelopes, critérios de parada e interpolação extrema.

A série histórica de demanda de energia elétrica do arquivo com extensão *csv* foi lida no código por intermédio da função *read_csv* do pacote *Pandas*. Após a obtenção dos dados temporais no programa, a função *MinMaxScaler* do pacote *sklearn.preprocessing* foi utilizada para normalizar os dados antes de serem introduzidos nas redes neurais artificiais de diferentes arquiteturas. A normalização foi necessária pois os dados de demanda de energia possui uma grande variação que poderia afetar o desempenho do algoritmo durante o treinamento e, dessa maneira, fornecer resultados não condizentes com a realidade; além disso, segundo Upadhaya et al. (2019), o processamento preliminar dos dados podem gerar melhores resultados preditivos relacionados à demanda de energia de curto prazo, por isso a técnica *MinMaxScaler* é usada em Li et al. (2018), Hossain & Mahmood (2020) e Jiang et al. (2018). A Equação (3.9) apresenta como a normalização dos dados foi realizada pela função *MinMaxScaler*.

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (3.9)$$

Após o pré-processamento dos dados históricos, a série temporal foi transformada em um problema de aprendizado supervisionado de máquina, ou seja, criou-se uma sequência de pares de entradas e saídas, da mesma forma que Jiang et al. (2018), Cui et al. (2020) e Dragomir et al. (2011) realizaram para efetuar previsões de demanda de energia de curto prazo com conjunto de dados históricos diferentes. Essas entradas são apresentadas às redes para que uma decisão seja tomada e então comparada com a saída desejada. Os parâmetros internos das redes neurais artificiais são modificados durante o treinamento pelo algoritmo Adam (*Adaptive Moment Estimation*) para que a diferença entre a resposta da rede e a saída desejada para um dado conjunto de entradas seja mínima. O algoritmo de treinamento Adam foi escolhido a partir do trabalho de Kingma & Ba (2015) pelo motivo de ter sido superior a outros algoritmos, além de ser utilizado como otimizador da função objetivo na rede neural para realizar a previsão de demanda de curto prazo em Li et al. (2018) e Bui et al. (2020). A métrica de acurácia utilizada como função objetivo a ser minimizada na rede para realizar essa comparação foi o erro MSE definido pela Equação 3.8, pois os erros são elevados ao quadrado antes de serem incorporados à média. Dessa forma, o MSE atribui um peso relativamente alto para erros grandes, logo é uma boa escolha de métrica quando estes resíduos elevados são indesejáveis, como no caso da previsão de demanda de energia de curto prazo de diferentes séries históricas visto em Bui et al. (2020), Dragomir et al. (2011), Oduro-Gyimah (2018), Ge et al. (2010) e Saad et al. (2017). Os dez neurônios de cada uma das duas camadas intermediárias de todos os modelos preditivos têm a *rectified linear unit* (ReLU), Equação (3.10), como função de ativação do mesmo modo que as redes neurais artificiais de Bui et al. (2020) e Maksut et al. (2019) utilizadas na previsão da necessidade energética no curto prazo de séries temporais diferentes. O motivo da escolha dessa função de ativação é por possuírem, conforme Talathi & Vartak (2015), a capacidade de melhorar a performance preditiva de redes neurais recorrentes. No que tange às camadas de arquitetura CNN das redes neurais, a quantidade do *kernel_size* foi definida em 6 e o número de *filters* foi estabelecido em 8 baseado no trabalho de Lang, Steinborn, Steffens & Lang (2019).

$$f(x) = \max(0, x) \quad (3.10)$$

O treinamento de todas as redes neurais artificiais foi realizado em 150 épocas. Jiang et al. (2018) utilizou 50 épocas, já Maksut et al. (2019) e Khan et al. (2020) necessitaram de 100 épocas para treinar suas redes neurais para conjuntos de dados energéticos distintos. O batch size utilizado no código foi igual a 32 (Islam et al. (2020) recorreu ao mesmo batch size para efetuar previsões de demanda de energia de curto prazo em seu trabalho para uma série temporal diferente). Após o treinamento das redes os parâmetros encontrados foram testados na amostra de validação para garantir a acurácia dos resultados preditivos fora da amostra de treino. Caso os modelos fossem validados, com erros próximos nas duas amostras, as previsões

efetivas seriam realizadas e testadas na amostra de teste. A Figura 3.2 apresenta o fluxograma das etapas realizadas na metodologia.



Figura 3.2: Fluxograma das etapas realizadas na metodologia
Fonte: (Autor, 2020).

Capítulo 4

Resultados e Discussões

Com a finalidade de verificar e comparar o desempenho preditivo de curto prazo das redes neurais MLP, RNN, GRU, LSTM, BiLSTM-BiGRU, BiGRU-BiLSTM, CNN-BiGRU, BiGRU-CNN e Ceemdan-LSTM-GRU, a tabela da da Figura 4.1 mostra os erros MAPE, MAE e RMSE (Todos com a unidade de *Megawatts*) provenientes das médias das 10 previsões de demanda de energia nas amostras de treino, validação e testes. A quantidade de números dentro dos parênteses adjacentes ao cenário em questão da primeira coluna representa o total de camadas intermediárias, e a magnitude desses valores indicam a quantidade de neurônios ocultos. Por exemplo: (10,10) possui 2 camadas com 10 neurônios cada uma, salvo as camadas de arquitetura CNN das redes neurais, onde a quantidade do *kernel_size* foi definida em 6 e o número de *filters* foi estabelecido em 8. O tempo requerido em horas para treinar, validar e efetuar todas a previões de demanda de energia de curto prazo está localizada na última coluna.

Estrutura \ Amostra		Treino			Validação			Teste			Tempo total em horas
		MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	
(10,10)	Ceemdan-LSTM-GRU	5,78%	178,29	251,81	4,40%	131,39	156,74	17,14%	513,59	572,19	11,19
	MLP	1,27%	39,05	53,32	1,24%	37,61	51,82	4,52%	134,36	171,08	0,73
	RNN	1,23%	38,62	52,69	1,22%	37,57	51,73	3,57%	106,05	138,51	3,87
	LSTM	1,19%	36,83	50,19	1,18%	35,98	49,29	4,10%	122,04	158,05	8,31
	GRU	1,19%	37,00	50,34	1,18%	36,19	49,59	4,10%	121,50	153,36	9,19
	Bi-LSTM_Bi-GRU	1,20%	37,28	50,86	1,19%	36,47	50,27	4,36%	129,43	167,65	12,35
	Bi-GRU_Bi-LSTM	1,16%	36,27	49,47	1,17%	35,79	49,30	4,41%	131,61	170,16	12,14
	CNN_Bi-GRU 150 épocas	2,55%	79,82	106,06	2,52%	77,05	102,47	4,81%	142,70	195,20	4,84
	CNN_Bi-GRU 300 épocas	2,44%	75,35	99,76	2,45%	74,25	98,30	4,90%	144,79	197,57	9,53
	Bi-GRU_CNN	1,27%	39,72	53,80	1,26%	38,46	52,33	3,42%	100,75	122,22	8,29

Figura 4.1: Comparação dos erros das médias das redes
Fonte: (Autor, 2021).

As células contendo os erros foram coloridas de acordo com o desempenho das previsões de cada cenário com intuito de melhorar a visualização e entendimento. Em azul estão os menores erros entre todas as redes neurais artificiais formadas por diferentes arquiteturas nas três amostras. Em contrapartida, na coloração vermelha estão os maiores erros nas amostras de treino, validação e teste.

Analisando a tabela da Figura 4.1 verifica-se que os desempenhos preditivos são próximos nas amostras de treino e validação para os três erros, principalmente o erro MAPE, com

exceção do modelo de previsão Ceemdan-LSTM-GRU. Com isso é possível afirmar que os parâmetros internos das redes neurais artificiais encontrados durante o aprendizado por intermédio da amostra de treino foram satisfatórios em relação a generalização das respostas na amostra de validação. Portanto as redes neurais treinadas e validadas estão aptas a realizarem as previsões efetivas de demanda de energia de curto prazo. Essas previsões efetivas são comparadas com a amostra de teste que é composta por 24 observações que não foram vistas em nenhum momento pelas redes neurais artificiais durante o treinamento e validação. As magnitudes dos erros originários das amostras de teste são bem maiores que da amostra de treino e validação. Uma explicação plausível para esse evento é o tamanho da amostra de teste (24 dados) possuir uma quantidade muito inferior de elementos do que as outras duas amostras, 36.199 dados compondo a amostra de treino e 8.330 formando a amostra de validação. Outra explicação, só que dessa vez mais coerente, é que as previsões da amostra de teste foram realizadas de maneira *multi step ahead* por intermédio da recursividade, o que ocasionou a propagação dos erros conforme as previsões eram feitas. Enquanto os valores futuros de demanda de energia na amostra de treino e validação foram obtidos na modalidade *one step ahead*, onde as previsões anteriores não interferem nas previsões subsequentes.

A partir da tabela da Figura 4.1 é possível notar que o modelo preditivo que teve a pior média de desempenho nas 3 amostras foi o Ceemdan-LSTM-GRU. Basicamente, o processo que originou os erros mais altos de todas as redes foi feito da seguinte forma: Em primeiro lugar utilizou-se a decomposição CEEMDAN na série de demanda de energia, de onde originou-se 13 componentes chamadas de *Intrinsic Mode Functions* (IMF) e a sub-série dos resíduos que foi descartada por apresentar magnitude insignificante; em seguida, nessas 13 IMFs foram utilizadas uma rede neural com uma camada LSTM e outra camada GRU para efetuar as previsões nas amostras de treino, validação e teste de maneira isolada em cada uma das componentes; e, por fim, todas as previsões de cada uma das componentes foram somadas para se obter as previsões finais. Até o IMF 8, o modelo LSTM-GRU estava sendo validado pois suas previsões nas amostras de treino e validação estavam aderentes à curva original, e o comportamento dos erros das duas amostras ao longo do treinamento estava decaindo de forma esperada até o ponto de convergência. Porém, a partir do IMF 9 os gráficos começaram a apresentar comportamentos estranhos que indicam uma não convergência do modelo durante o treinamento. Esses resultados são bem inesperados, pois essas componentes possuem uma suavidade muito maior que as primeiras componentes, o que seria tarefa fácil para a rede neural aprender seus padrões. Por isso foram refeitas as simulações várias vezes das componentes para tentar melhorar os resultados, foi até utilizado a rede MLP e RNN, entretanto sem sucesso.

Algumas dessas várias simulações são mostradas na tabela da Figura 4.2, onde integrou-se as componentes de baixa frequência IMF 9, IMF 10, IMF 11, IMF 12 e IMF 13 para formar uma única série temporal para economizar tempo de simulação, pois foram realizados 4 treinamentos a menos. Essa nova série temporal foi treinada em quatro cenários distintos. O primeiro cenário teve 500 épocas; o segundo, 1000 épocas; o terceiro, 1500 épocas; e, por último, o quarto com 2000 épocas. O número de épocas foi aumentado devido à suspeita de a rede não

ter tempo hábil para aprender os padrões da soma dessas IMFs, no teste inicial era de 150 épocas. O primeiro, o segundo, o terceiro e o quarto cenário tiveram, respectivamente; 9,92; 12,57; 15,43 e 17,99 horas. Os resultados na amostra de treino e validação foram bons, no entanto, quando as previsões foram realizadas na amostra de teste, que são as previsões efetivas, os resultados foram ruins. Os erros foram acima de 13,27 %. Esse é um caso clássico de overfitting, onde a rede, devido ao alto número de épocas, decora o padrão dos dados da amostra de treino e validação para diminuir seus respectivos erros, ao invés de aprender suas características intrínsecas. Só que esse padrão decorado não consegue reproduzir bons resultados para dados nunca vistos antes (amostra teste). Foram utilizados, também, as decomposições *Empirical Mode Decomposition* (EMD) e *Ensemble Empirical mode decomposition* (EEMD), porém os resultados foram semelhantes.

Após pesquisas sobre a decomposição CEEMDAN foi verificado que a técnica é eficaz, segundo Chang (2010), apenas em sinais não estacionários e não lineares. Toda série de demanda de energia possui suas componentes lineares e não lineares, porém não existe um método para saber a parcela exata de cada uma na construção da série temporal original. Mas há fortes evidências que as componentes lineares estão em maiores quantidades, pois existem muitos métodos baseados em regressão linear utilizados para efetuar previsão de demanda de energia de curto prazo na literatura, por exemplo: Auto-regressivos, Médias Móveis, ARIMA, SARIMA, Suavização exponencial, Holt-Winters. Em relação a não estacionaridade, fazendo uma inspeção visual verifica-se que há fortes indícios de a série original ser estacionária, onde suas propriedades estatísticas (média, variação, autocorrelação,...) são constantes ao longo do tempo. Essas podem ser as razões para que o método Ceemdan-LSTM-GRU fornecesse resultados preditivos relacionados à demanda de energia não condizentes com as observações da amostra de teste.

Vendo a impossibilidade de utilizar decomposições empíricas na série de demanda de energia original, optou-se por misturar camadas de redes neurais já conhecidas com o objetivo de diminuir os erros MAPE, MAE e RMSE na amostra de teste. As redes propostas foram: BiLSTM_Bi-GRU, BiGRU_BiLSTM, CNN_BiGRU e BiGRU_CNN.

Recorrendo à tabela da Figura 4.1 verifica-se que o modelo BiGRU_BiLSTM foi o que forneceu as previsões mais eficientes nas três métricas de acurácia (MAPE, MAE e RMSE) nas amostras de treino e validação. Embora os resultados preditivos nessas amostras fossem os melhores em relação às demais redes neurais artificiais, a alta acuracidade das previsões não foi perpetuada para a amostra de teste. Esse fenômeno mostra a extrema importância de se trabalhar com 3 amostras de previsão, pois em uma análise preliminar, a partir dos resultados preditivos das amostra de treino e validação, um modelo que fornece previsões não condizentes com a realidade pode ser escolhido em detrimento dos outros mais eficientes. Para reverter essa situação é utilizado uma amostra de teste onde os modelos preditivos são testados para dar uma maior segurança na escolha da rede neural artificial de maior acuracidade. O conjunto de dados dessa amostra não é conhecido pela rede durante o treinamento e validação de seus parâmetros

internos. Esses dados são utilizados apenas para comparar as previsões efetivas na extrapolação da série de demanda de energia pelos modelos preditivos na modalidade *multi step ahead*. Apesar de utilizar uma amostra de teste para aumentar a confiabilidade no momento de escolha da rede neural artificial, ainda assim não é garantido que o modelo preditivo seja o melhor para o conjunto de dados históricos em questão. Pois se tratando de estimar valores futuros, não existe certeza absoluta.

Um exemplo ainda melhor sobre a importância da utilização de uma amostra de teste para escolher o modelo preditivo mais adequado é observado na tabela da Figura 4.2, mas especificamente na rede Ceemdan-LSTM-GRU treinada em 1500 épocas. Na respectiva simulação de número 4, os erros das amostra de treino e validação são extremamente baixos. Para ser mais preciso, as métricas de acurácia MAPE, MAE e RMSE foram de 1,01%; 31,75 MW e 47,92 MW na amostra de treino e 0,94%; 28,23 MW e 35,92 MW na amostra de validação. Todavia a magnitude dos resíduos sofreu um drástico aumento na amostra de teste com os seguintes valores de 13,55% provenientes do MAPE; 413,77 MW no erro MAE e, por fim, 495,98 MW no tocante ao erro RMSE. Caso o modelo preditivo fosse escolhido baseado exclusivamente nos erros da amostra de treino e validação, as previsões efetivas seriam desastrosas, conforme visto na amostra de teste, o que ocasionaria problemas no fornecimento de energia elétrica no curto prazo para o suprimento da demanda. Pois caso o fornecimento seja abaixo do esperado, a disponibilidade e a confiabilidade da energia elétrica fica comprometida, impactando diretamente de forma negativa o consumidor final. Em contrapartida, se o fornecimento de energia estiver acima da capacidade suportada pelo consumo da demanda, a eficiência energética da empresa do setor elétrico é afetada.

simulação	treino			validação			teste			tempo em horas
	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	
1 (150 épocas)	5,78%	178,29	251,81	4,40%	131,39	156,74	17,14%	513,59	572,19	11,19
2 (500 épocas)	1,98%	64,18	93,02	1,69%	52,80	70,59	13,56%	411,26	497,93	9,92
3 (1000 épocas)	1,21%	39,03	62,55	1,01%	31,49	42,74	13,27%	402,41	483,51	12,57
4 (1500 épocas)	1,01%	31,75	47,92	0,94%	28,23	35,92	13,65%	413,77	495,98	15,43
5 (2000 épocas)	1,58%	50,12	65,93	1,51%	46,30	55,21	14,38%	437,65	538,11	17,99

Figura 4.2: Comparação do modelo Ceemdan-LSTM-GRU com variação nas épocas

Fonte: (Autor, 2021).

Voltando a atenção para a rede neural artificial de arquitetura BiGRU_BiLSTM, um motivo para que esse modelo fornecesse os melhores resultados na amostra de treino e validação é devido a memória de longo prazo das camadas intermediárias GRU e LSTM que fazem com que informações muito distantes não sejam afetadas pelo *vanishing gradient* durante o treinamento. Outro motivo é o incremento do método *Bidirectional* (Bi) adicionado as camadas ocultas recorrentes que fazem com que a rede efetue previsões não apenas com informações do passado até o futuro, porém com informações do futuro até o passado. Em outras palavras as previsões são realizadas com informações nos dois sentidos. No entanto essas características não foram capazes de reproduzirem bons resultados nas previsões efetivas da amostra de teste. Nem ao menos invertendo a ordem das camadas intermediárias, dando surgimento ao modelo

BiLSTM_BiGRU, foi possível melhorar os valores futuros de demanda de energia da amostra de teste. Esse modelo necessitou do maior tempo computacional para realizar todas as 10 simulações que totalizou em 12,35 horas. Em seguida o modelo invertido BiGRU_BiLSTM foi responsável pelo segundo maior dispêndio de tempo computacional (12,14 horas). Os dois modelos propostos, apesar de terem sido validados após o treinamento, não forneceram as melhores previsões efetivas, nem mesmo superou as redes clássicas RNN, GRU e LSTM. E mesmo se isso ocorre-se, os erros deveriam ser bastante baixos para compensar o alto tempo de simulação dos modelos.

Após formar uma rede neural com duas camadas de arquiteturas recorrentes *Bidirectional* distintas de memória de longo prazo, optou-se por utilizar uma delas para integrar um modelo com uma camada CNN. Dessa fusão surgiu o modelo CNN_BiGRU que foi treinado por 150 épocas o qual resultou em valores elevados nas três métricas de acurácia nas amostras de treino e validação quando comparados com as demais redes neurais artificiais, salvo a rede Ceemdan-LSTM-GRU, da tabela da Figura 4.1. Dessa forma foi formulada a hipótese de que 150 épocas não era uma quantidade suficiente para treinar esse modelo, por isso a rede foi treinada novamente com o dobro de épocas do cenário anterior, totalizado em 300 épocas. Mesmo com essa alteração realizada, houve uma pequena melhora nos erros MAPE, MAE e RMSE nas amostras de treinamento e validação em relação ao cenário de 150 épocas, porém os erros ainda estavam maiores que os provenientes das outras redes neurais, com exceção, mais uma vez, da Ceemdan-LSTM-GRU. No que concerne à amostra de teste, o cenário de 150 épocas prevaleceu sobre o de 300 no que tange aos três erros. Ambos os cenários da CNN_BiGRU forneceram as piores previsões efetivas na amostra de teste concomitantemente com o modelo Ceemdan-LSTM-GRU, que forneceu os erros mais elevados.

Ainda na busca de encontrar modelos preditivos mais eficientes que os famigerados MLP, RNN, GRU e LSTM na aplicação em séries temporais de demanda de energia elétrica, foram invertidas as camadas intermediárias da rede CNN_BiGRU para compor o modelo BiGRU_CNN. Esse modelo não teve destaque nas previsões efetuadas nas amostras de treinamento e validação, em compensação conquistou o melhor desempenho nas previsões efetivas da amostra de teste, dentre todos os modelos disponíveis, possuindo os menores erros nas três métricas de acurácia. Esses erros foram de 3,42% para o MAPE; 100,75 MW para o MAE e 122,22 MW no que diz respeito ao RMSE. Ao contrário do que ocorreu entre os modelos BiGRU_BiLSTM e BiLSTM_BiGRU, a troca de posição das camadas intermediárias de CNN_BiGRU para BiGRU_CNN alterou substancialmente o desempenho preditivo da rede levando em consideração a baixa dos três erros nas três amostras.

A tabela da Figura 4.1 mostrou o comportamento das métricas de acurácias MAPE, MAE e RMSE provenientes das previsões de demanda de energia de curto prazo das redes de diferentes camadas propostas, bem como as redes neurais artificiais clássicas, nas amostras de treino, validação e teste. Sob outra perspectiva, é interessante analisar o tempo requerido para que todas as redes neurais artificiais efetuassem o treinamento do modelo, validação dos

parâmetros do modelo e previsões efetivas. A Figura 4.3 apresenta o tempo dado em horas (eixo vertical) que foi necessário para completar todas as 10 simulações computacionais nos diferentes modelos preditivos (eixo horizontal). Caso o critério de escolha deixasse de ser os menores erros na amostra de teste e passasse a ser o menor tempo de simulação, o modelo BiGRU_CNN de 8,29 horas daria lugar a rede MLP de 0,73 horas. A economia de tempo foi de 7,56 horas ao custo do aumento dos erros em 1,1% no MAPE; 33,61 MW no MAE e 48,86 para o RMSE. Se o critério de escolha, agora, fosse definido como o melhor custo benefício entre o tempo de simulação da rede e a eficiências das previsões efetivas, o modelo neural RNN seria preferido. Pois o tempo poupado seria de 4,42 horas ao viés da crescente dos erros MAPE, MAE e RMSE na ordem de 0,15%; 5,3 MW e 16,29 MW quando comparados com o modelo BiGRU_CNN. Com essas análises é possível perceber que a escolha da rede neural artificial para efetuar previsões de demanda de energia de curto prazo não fica restritamente baseada nas melhores previsões efetivas da amostra de teste. Na verdade, a decisão cabe ao analista no momento de tomar as decisões da escolha do critério a ser utilizado. Não esquecendo de levar em consideração todos os cenários possíveis e ponderar as possíveis consequências da escolha de uma critério em detrimento de outro.

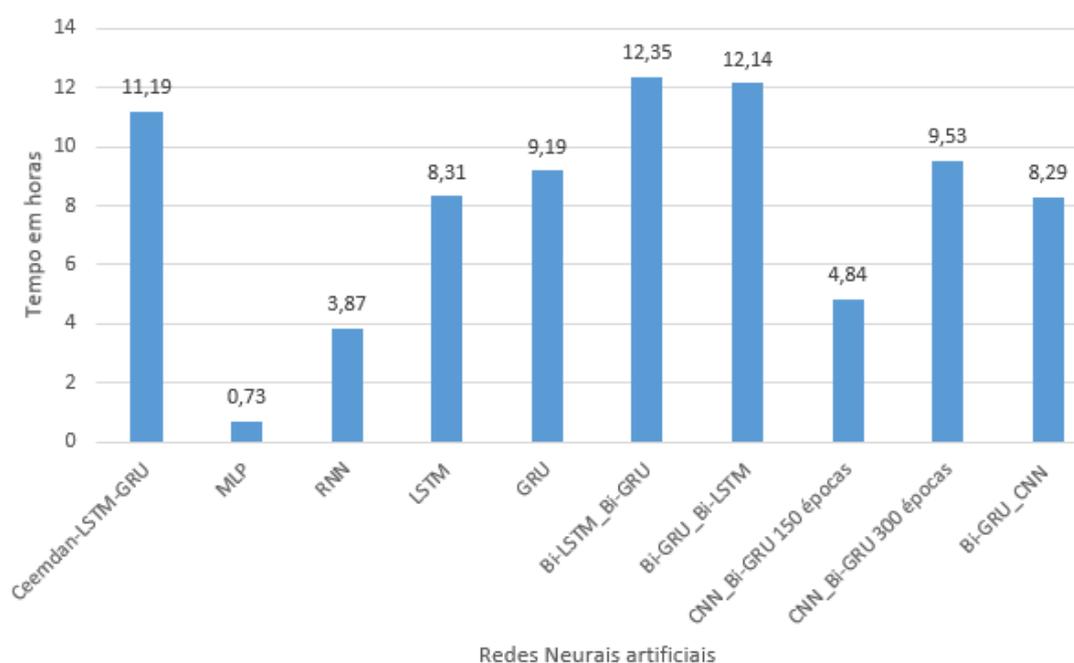


Figura 4.3: Tempo de simulação das redes neurais artificiais
Fonte: (Autor, 2021).

Como o critério de escolha da rede neural artificial foram os menores resíduos das três métricas de acurácia das previsões efetivas de demanda de energia da amostra de teste, todos os valores previstos do modelo BiGRU_CNN nas 10 simulações são apresentados na Figura 4.4, juntamente com as observações reais da série temporal energética (curva em azul). Essas previsões efetivas foram executadas para formar os valores médios da tabela da Figura 4.1, onde as curvas preditivas referentes a cada um deles possuem uma cor distinta e o número posterior à palavra BiGRU_CNN localizado no quadro superior do canto esquerdo é o respectivo cenário

da simulação.

O eixo das ordenadas do gráfico da Figura 4.4 é representado pela demanda de energia em megawatt e o eixo das abscissas corresponde ao horizonte de previsão, em outras palavras, o gráfico expõe o comportamento previsto pelas redes neurais da demanda de energia em função do tempo. Esse tempo foi fragmentado no intervalo de horas compreendido entre 02:00 h da madrugada do dia 31 de outubro de 2017 e 01:00 da madrugada do dia primeiro de novembro do mesmo ano. As observações reais reproduzidas pela curva azul foram aferidas de hora em hora, logo existem 24 dados de demanda de energia observados que formam a amostra de teste. Por consequência as previsões efetivas realizadas pela rede BiGRU_CNN foram de forma horária dentro do horizonte de previsão de 24 horas, significando que 24 previsões foram efetuadas.

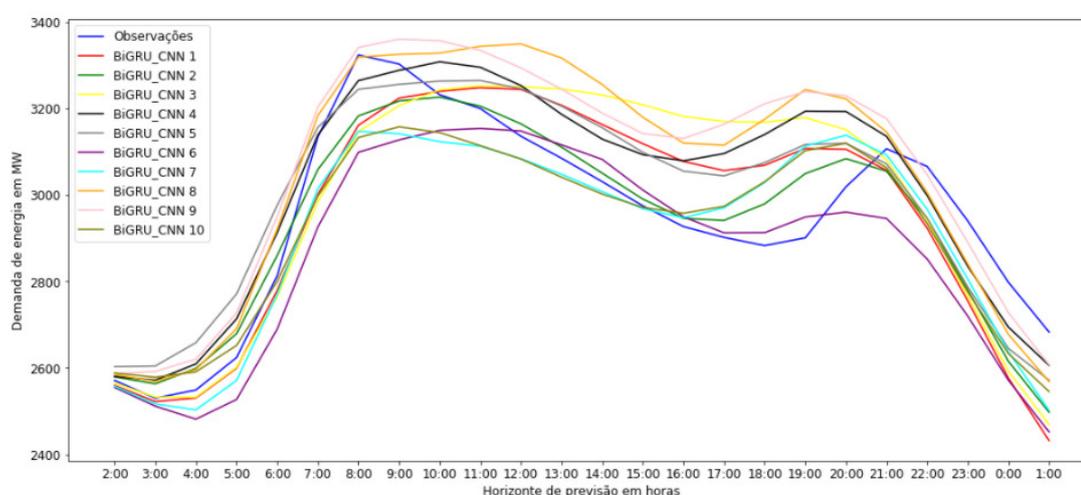


Figura 4.4: Previsões efetivas de demanda de energia de curto prazo de todas as simulações referentes à rede neural BiGRU_CNN

Fonte: (Autor, 2021).

Analisando a Figura 4.4 verifica-se que a curva em azul da série temporal original da demanda de energia da amostra de teste possui um crescimento acentuado entre 03:00 h da madrugada até 08:00 h da manhã, onde a necessidade energética de 3324,00 MW é a maior para o respectivo dia. Do período de tempo compreendido no intervalo de 08:00 h da manhã e 18:00 h da noite, a curva sofre um decaimento suave até o valor de 2883,00 MW. Após esse intervalo de tempo a demanda de energia volta a crescer até as 21:00 h da noite chegando à 3107,00 MW que, em seguida, começa a decair de maneira acentuada até o valor de 2683,00 MW no horário de 01:00 h da madrugada do dia seguinte.

Todas as curvas de previsões realizadas pelas redes BiGRU_CNN não apresentam uma aderência satisfatória com a demanda real de energia entre o período de tempo percebido entre 2:00 h da madrugada até 6:00 h da manhã. Posteriormente a esse intervalo temporal, as previsões das redes não são capazes de acompanhar o comportamento das observações até 21:00 h da noite, principalmente do máximo global (8:00 h) até o máximo local (21:00 h) da função não linear geradora da série histórica de demanda de energia. Do intervalo de tempo entre 6:00 h até 8:00 h, as curvas preditivas não crescem na mesma proporção que a série temporal original e

das 8:00 h até 21:00 h as previsões apresentam um comportamento oscilatório em torno de uma média compreendida no intervalo de 3100,00 MW e 3200,00 MW, não acompanhando a dinâmica natural da demanda de energia que decai suavemente até as 18:00 h e, logo depois, tem uma ascensão significativa até as 21:00 h, uma variação de 224 MW em um espaço de tempo de três horas. No período de tempo restante, todas as previsões das redes possuem comportamento de decaimento similar das observações, onde, após as 22:00 h, todos os valores previstos de demanda de energia ficam abaixo do real.

De acordo com a tabela da Figura 4.5 verifica-se que o cenário que obteve as melhores previsões efetivas da amostra de teste, baseada nos erros MAPE, MAE e RMSE, foi a simulação de número 2 (curva verde da Figura 4.4) com os respectivos valores de 2,38%, 69,47 MW e 89,57 MW. Em contrapartida a simulação que teve o pior desempenho foi a de número 3 (curva amarela da Figura 4.4), ainda na amostra de teste, com os seguintes erros de 4,58%, 135,63 MW e 164,63 MW para as métricas de acurácia MAPE, MAE e RMSE. Mesmo possuindo o pior desempenho nas previsões efetivas da amostra de teste, a terceira simulação da rede neural artificial BiGRU_CNN forneceu os menores erros para as amostras de treinamento e validação. Esses resultados corroboram que o critério de escolha de um modelo preditivo não possa ser baseado nas previsões das duas amostras citadas anteriormente, pois a extrapolação da série temporal de demanda de energia por essa rede pode não condizer com suas características intrínsecas.

simulação	treino			validação			teste			tempo em horas
	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	
1	1,28%	40,25	54,68	1,25%	38,15	51,84	3,76%	111,19	132,38	0,84
2	1,30%	41,30	56,17	1,27%	39,34	53,72	2,38%	69,47	89,57	0,82
3	1,18%	36,74	50,44	1,16%	35,52	48,73	4,58%	135,63	164,63	0,82
4	1,25%	38,38	51,20	1,25%	37,68	50,61	3,47%	101,57	123,44	0,82
5	1,36%	43,04	58,62	1,35%	42,07	57,36	3,74%	108,40	119,65	0,82
6	1,31%	40,86	55,72	1,27%	38,90	53,61	3,39%	100,58	129,26	0,85
7	1,26%	39,50	53,50	1,25%	38,51	52,31	2,94%	87,36	107,06	0,84
8	1,26%	39,18	52,48	1,25%	37,96	51,26	4,44%	131,09	160,06	0,83
9	1,28%	39,67	53,32	1,28%	38,84	52,70	2,50%	73,86	89,37	0,83
10	1,23%	38,32	51,84	1,23%	37,58	51,12	2,96%	88,31	106,74	0,83

Figura 4.5: Comparação dos erros das simulações da rede BiGRU_CNN

Fonte: (Autor, 2021).

As previsões efetivas da rede BiGRU_CNN da simulação de número 2 são comparadas com a amostra de teste que é composta por 24 observações que não foram vistas em nenhum momento pela rede durante o seu treinamento e validação. A tabela da Figura 4.6 confronta os resultados fornecidos pela rede neural com as observações pertencentes à amostra de teste por meio dos erros MAPE, MAE e MSE para os diferentes horizontes de previsão.

Analisando a tabela da Figura 4.6 verifica-se que os três erros aparentam possuir uma tendência de crescimento oscilatória em função do aumento do horizonte de previsão, na verdade seus comportamentos são de difícil interpretação. O horizonte de previsão que teve melhor desempenho nas três métricas de acurácia foi o de número nove com os erros MAPE de 0,17%, MAE de 5,47 MW e MSE de 29,96 MW; os de número 1 e 10 também tiveram desempenho

satisfatório. Em contrapartida o horizonte de previsão de número vinte e quatro teve o pior desempenho nas três métricas de acurácia com os erros MAPE de 6,91%, MAE de 185,52 MW e MSE de 34416,68 MW; seguido pelos horizontes preditivos de posição vinte e três e vinte e dois.

Horizonte de previsão	Amostra teste	Previsão efetiva	MAPE em %	MAE em MW	MAE em RMSE
1	2571	2579,42	0,33%	8,42	70,92
2	2530	2563,27	1,32%	33,27	1106,99
3	2549	2599,05	1,96%	50,05	2505,03
4	2624	2679,04	2,10%	55,04	3029,22
5	2813	2860,06	1,67%	47,06	2214,21
6	3134	3058,69	2,40%	75,31	5671,94
7	3324	3182,96	4,24%	141,04	19892,09
8	3303	3217,42	2,59%	85,58	7323,53
9	3232	3226,53	0,17%	5,47	29,96
10	3200	3205,61	0,18%	5,61	31,44
11	3136	3164,69	0,92%	28,69	823,39
12	3085	3110,84	0,84%	25,84	667,94
13	3031	3050,62	0,65%	19,62	384,92
14	2975	2990,63	0,53%	15,63	244,23
15	2927	2945,96	0,65%	18,96	359,34
16	2902	2941,31	1,35%	39,31	1544,94
17	2883	2979,55	3,35%	96,55	9321,11
18	2901	3049,50	5,12%	148,50	22053,63
19	3019	3084,05	2,15%	65,05	4231,29
20	3107	3055,03	1,67%	51,97	2700,83
21	3066	2944,01	3,98%	121,99	14882,69
22	2940	2779,90	5,45%	160,10	25632,35
23	2798	2615,24	6,53%	182,76	33402,92
24	2683	2497,48	6,91%	185,52	34416,68

Figura 4.6: Previsões acompanhadas dos erros MAPE, MAE e MSE na amostra de teste da rede BiGRU_CNN da segunda simulação
Fonte: (Autor, 2021).

Capítulo 5

Conclusão

Neste trabalho foram testadas e avaliadas redes neurais artificiais de arquitetura BiLSTM-BiGRU, BiGRU-BiLSTM, CNN-BiGRU, BiGRU-CNN e Ceemdan-LSTM-GRU para efetuar previsões de demanda de energia elétrica de curto prazo, pois seus resultados preditivos podem servir como ferramenta de apoio à tomada de decisão no que concerne ao planejamento operacional de empresas do setor elétrico, em virtude da demanda de energia elétrica ser estabelecida no dia anterior a sua transmissão em grande parte do mundo. Essas previsões de demanda de energia foram realizadas com camadas de diferentes arquiteturas com a finalidade de conseguir melhores resultados que as redes neurais artificiais clássicas do tipo MLP, RNN, GRU e LSTM, visto que previsões mais próximas da realidade significam melhores decisões que poderão ocasionar maiores receitas brutas para a empresa do setor energético, além de seus respectivos clientes serem beneficiados pelo aumento da confiabilidade e disponibilidade dos serviços essenciais prestados.

Dos modelos neurais preditivos propostos anteriormente relacionados à demanda de energia elétrica no curto prazo, o único capaz de fornecer resultados melhores que as redes neurais clássicas, para o conjunto de dados utilizados nesta dissertação, foi o BiGRU-CNN com o erro médio MAPE na ordem de 3,42% na amostra de teste que representa as previsões efetivas. Embora os modelos BiLSTM-BiGRU, BiGRU-BiLSTM e CNN-BiGRU não foram capazes de reproduzirem previsões efetivas com os erros médios MAPE menores que as redes neurais MLP, RNN, GRU e LSTM, eles podem, ainda, serem utilizados em aplicações práticas pois as médias dos seus respectivos erros MAPE não foram superiores a 4,90%. Por outro lado, aplicar a técnica de decomposição de modo empírico Ceemdan na série temporal de demanda de energia e, então, alimentar a rede LSTM-GRU com suas componentes não é viável em cenários reais. Uma vez que o modelo de previsão Ceemdan-LSTM-GRU foi responsável pela média do erro MAPE da amostra de teste de 17,14%. Caso a empresa do setor elétrico em questão opta-se por esse modelo preditivo para auxiliá-la na sua gestão e tomada de decisões, certamente problemas de natureza técnica e operacional surgiriam impactando significativamente suas economias e seus clientes.

Além de comparar os resultados preditivos das redes neurais artificiais e verificar a superioridade da modelo proposto BiGRU-CNN sobre os demais, analisou-se o comportamento das redes para diferentes modelos estruturais. Foi verificado no Capítulo anterior que a alteração da

ordem das camadas distintas pode impactar significativamente na acurácia preditiva dos modelos, bem como o tempo requerido para que as simulações computacionais fossem finalizadas. Esse fenômeno foi observado nas redes neurais artificiais formadas pelas camadas de arquitetura *Convolutional Neural Network* (CNN) e *Bidirectional Gated Recurrent Unit* (BiGRU). Contudo a mesma situação não ocorreu nas redes neurais artificiais formadas pelas camadas de arquitetura *Bidirectional Long Short Term Memory* (BiLSTM) e BiGRU.

Quase todas as estruturas neurais dos modelos preditivos necessitaram de um tempo considerável para realizar as 10 simulações até se chegar a uma amostra estatística aceitável, com exceção do modelo Ceemdan-LSTM-GRU que foi simulado uma única vez onde ocorreu 13 previsões correspondentes às suas respectivas componentes. Cada simulação de uma rede específica realizou previsões efetivas dentro do horizonte de 24 horas, o que torna os modelos de difícil acesso para aplicações reais como ferramenta de apoio à tomada de decisão envolvendo demanda de energia de curto prazo em empresas do setor elétrico. Logo a quantidade de simulações a serem realizadas para não gerar resultados tendenciosos deve ser escolhida de maneira parcimoniosa para que os modelos sejam realizáveis. Outra forma de tornar os modelos mais acessíveis para aplicações práticas é efetuar as simulações em máquinas computacionais mais potentes e dedicadas somente para as redes neurais. Entretanto esta alternativa pode torna-se muito dispendiosa, à medida que aquela pode fornecer resultados preditivos menos eficientes. Independentemente da escolha feita pelo gestor ocorrerá um *trade-off* que deve ser analisado minuciosamente para que a saúde financeira da empresa não seja afetada.

Conforme visto nos trabalhos similares da literatura, a inserção da variável exógena de temperatura na rede neural melhora a acurácia das previsões relacionadas à demanda de energia de curto prazo. À vista desse fato, como sugestão para trabalhos futuros fica em utilizar variáveis exógenas na rede BiGRU-CNN que teve o melhor desempenho no presente trabalho para aumentar a eficácia das previsões pertinentes à necessidade energética dos consumidores. Essas variáveis exógenas podem ser meteorológicas, tais como: umidade relativa do ar, velocidade do vento e insolação; e de comportamento dos consumidores, por exemplo: feriados, dias de semana e fins de semana. Além da inserção de mais variáveis exógenas na rede neural de arquitetura BiGRU-CNN, métodos de previsões estatísticos poderiam ser utilizados para formar um modelo híbrido preditivo, pois esse modelo composto poderia captar tanto os padrões não lineares da demanda de energia, quantos os lineares, ocasionando a possibilidade de gerar melhores previsões.

Referências Bibliográficas

- Abbasimehr, H., Shabani, M. & Yousefi, M. (2020). An optimized model using LSTM network for demand forecasting, *Comput. Ind. Eng.* **143**(March): 106435.
URL: <https://doi.org/10.1016/j.cie.2020.106435> Citado na página 16.
- Adewuyi, S. A., Aina, S. & Oluwaranti, A. I. (2020). A deep learning model for electricity demand forecasting based on a tropical data, *Applied Computer Science* **16**(1): 5–17. Citado na página 23.
- Albawi, S., Mohammed, T. A. M. & Alzawi, S. (2017). Understanding of a Convolutional Neural Network, *2017 International Conference on Engineering and Technology (ICET)* pp. 1–6. Citado na página 37.
- Angelaccio, M. (2019). Forecasting Public Electricity Consumption with ARIMA Model: A Case Study from Italian Municipalities Energy Data, *2019 Int. Symp. Adv. Electr. Commun. Technol. ISAECT 2019* pp. 2–4. Citado na página 39.
- Berriel, R. F., Lopes, André Teixeira Rodrigues, A., Varejão, F. M. & Oliveira-Santos, T. (2017). Monthly energy consumption forecast: A deep learning approach, *2017 Int. Jt. Conf. Neural Networks* . Citado na página 43.
- Bianchi, F. M., Maiorino, E., Kampffmeyer, M. C., Rizzi, A. & Jenssen, R. (2017). *Networks for Short-Term Load Forecasting An Overview and Comparative Analysis*, 2 edn, Springer. Citado 4 vezes nas páginas 28, 31, 32 e 33.
- Bouktif, S., Fiaz, A., Ouni, A. & Serhani, M. A. (2018). Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches, *Energies* **11**(7). Citado na página 44.
- Bui, V., Nguyen, V. H., Pham, T. L., Kim, J. & Jang, Y. M. (2020). RNN-based Deep Learning for One-hour ahead Load Forecasting, *2020 Int. Conf. Artif. Intell. Inf. Commun. ICAIIC 2020* pp. 587–589. Citado 2 vezes nas páginas 42 e 58.
- Carpinteiro, O. A. & Alves Da Silva, A. P. (2000). A hierarchical neural model in short-term load forecasting, *Proc. - Brazilian Symp. Neural Networks, SBRN 2000-Janua*: 120–124. Citado na página 16.
- Chandramitasari, W., Kurniawan, B. & Fujimura, S. (2019). Building Deep Neural Network Model for Short Term Electricity Consumption Forecasting, *Proceeding - 2018 Int. Symp. Adv. Intell. Informatics Revolutionize Intell. Informatics Spectr. Humanit. SAIN 2018* pp. 43–48. Citado na página 44.
- Chang, K. M. (2010). Ensemble empirical mode decomposition: A Noise-Assited, *Biomedizinische Technik* **55**(1): 193–201. Citado na página 62.
- Chapagain, K., Kittipiyakul, S. & Kulthanavit, P. (2020). Short-term electricity demand forecasting: Impact analysis of temperature for Thailand, *Energies* **13**(10): 1–29. Citado na página 15.
- Charytoniuk, W. & Chen, M.-S. (2000). Very Short-Term Load Forecasting Using Artificial,

IEEE Trans. Power Syst. **15**(1): 263 – 268. Citado na página 15.

- Chen, J. X., Jiang, D. M. & Zhang, Y. N. (2019). A Hierarchical Bidirectional GRU Model with Attention for EEG-Based Emotion Classification, *IEEE Access* **7**: 118530–118540. Citado na página 36.
- Chen, T., Huang, W., Wu, R. & Ouyang, H. (2020). Short Term Load Forecasting Based on SBiGRU and CEEMDAN-SBiGRU Combined Model, *IEEE Access* **9**. Citado na página 47.
- Cheng, Y., Xu, C., Mashima, D., Thing, V. L. & Wu, Y. (2017). PowerLSTM: Power demand forecasting using long short-term memory neural network, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* **10604 LNAI**: 727–740. Citado na página 43.
- Chitalia, G., Pipattanasomporn, M., Garg, V. & Rahman, S. (2020). Robust short-term electrical load forecasting framework for commercial buildings using deep recurrent neural networks, *Applied Energy* **278**(May): 115410.
URL: <https://doi.org/10.1016/j.apenergy.2020.115410> Citado na página 51.
- Cui, C., He, M., Di, F., Lu, Y., Dai, Y. & Lv, F. (2020). Research on Power Load Forecasting Method Based on LSTM Model, *Proc. 2020 IEEE 5th Inf. Technol. Mechatronics Eng. Conf. ITOEC 2020* (Itoec): 1657–1660. Citado 3 vezes nas páginas 44, 55 e 58.
- De Cian, E., Lanzi, E. & Roson, R. (2011). The Impact of Temperature Change on Energy Demand: A Dynamic Panel Analysis, *SSRN Electron. J.* (46). Citado na página 39.
- de Cian, E., Lanzi, E. & Roson, R. (2013). Seasonal temperature variations and energy demand: A panel cointegration analysis for climate change impact assessment, *Clim. Change* **116**(3-4): 805–825. Citado na página 39.
- Deng, Y., Jia, H., Li, P., Tong, X., Qiu, X. & Li, F. (2019). A deep learning methodology based on bidirectional gated recurrent unit for wind power prediction, *Proceedings of the 14th IEEE Conference on Industrial Electronics and Applications, ICIEA 2019* pp. 591–595. Citado na página 36.
- Dragomir, O. E., Dragomir, F., Brezeanu, I. & Mincă, E. (2011). MLP neural network as load forecasting tool on short- term horizon, *2011 19th Mediterr. Conf. Control Autom. MED 2011* pp. 1265–1270. Citado 2 vezes nas páginas 41 e 58.
- Du, J., Cheng, Y., Zhou, Q., Zhang, J., Zhang, X. & Li, G. (2020). Power Load Forecasting Using BiLSTM-Attention, *IOP Conference Series: Earth and Environmental Science* **440**(3). Citado na página 48.
- Fengming, Z., Shufang, L., Zhimin, G., Bo, W., Shiming, T. & Mingming, P. (2017). Anomaly detection in smart grid based on encoder-decoder framework with recurrent neural network, *The Journal of China Universities of Posts and Telecommunications* **24**(December): 67–73. Citado 2 vezes nas páginas 34 e 35.
- Galushkin, A. (2007). *Neural Networks Theory*, 1 edn, Springer. Citado 3 vezes nas páginas 18, 21 e 22.
- Ge, C., Zhang, J. C., Sun, Y. B. & Sun, L. Y. (2010). Application of dynamic recurrent neural network in power system short-term load forecasting, *2010 Int. Conf. Comput. Control Ind. Eng. CCIE 2010* **1**: 378–381. Citado 2 vezes nas páginas 42 e 58.

- Geron, A. (2019). *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2 edn, O'Reilly. Citado na página 57.
- Ghalekhondabi, I., Ardjmand, E., Weckman, G. R. & Young, W. A. (2017). An overview of energy demand forecasting methods published in 2005–2015, *Energy Syst.* **8**(2): 411–447. Citado na página 15.
- Gong, X., Tang, B., Zhu, R., Liao, W. & Song, L. (2020). Data augmentation for electricity theft detection using conditional variational auto-encoder, *Energies* **13**(17): 1–14. Citado 2 vezes nas páginas 37 e 38.
- Graves, A. (2012). *Supervised Sequence Labelling with Recurrent Neural Network*, 1 edn, Springer, New York. Citado 9 vezes nas páginas 23, 26, 27, 28, 29, 30, 31, 32 e 34.
- Guyton, A. C. & Hall, J. E. (2005). *Textbook of Medical Physiology*, Saunders. Citado 2 vezes nas páginas 18 e 19.
- Hagan, M. T., Demuth, H. B. & Beale, M. H. (2014). *Neural Network Design*, 2 edn, Martin Hagan. Citado 13 vezes nas páginas 16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28 e 29.
- Hahn, H., Meyer-Nieberg, S. & Pickl, S. (2009). Electric load forecasting methods: Tools for decision making, *European Journal of Operational Research* **199**(3): 902–907.
URL: <http://dx.doi.org/10.1016/j.ejor.2009.01.062> Citado na página 15.
- Hammer, B. (2000). *Learning with Recurrent Neural Networks*, Springer London. Citado 3 vezes nas páginas 23, 26 e 28.
- Harvey, R. (1994). *Neural Network Principles*, 1 edn, Prentice Hall, New Jersey. Citado 2 vezes nas páginas 18 e 22.
- Haykin, S. (1999). *Neural networks and learning Machines*, Vol. 127, 2 edn, Pearson, New Jersey. Citado 12 vezes nas páginas 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29 e 31.
- He, F., Zhou, J., kai Feng, Z., Liu, G. & Yang, Y. (2019). A hybrid short-term load forecasting model based on variational mode decomposition and long short-term memory networks considering relevant factors with Bayesian optimization algorithm, *Applied Energy* **237**(January): 103–116.
URL: <https://doi.org/10.1016/j.apenergy.2019.01.055> Citado na página 46.
- Hekkenberg, M., Moll, H. C. & Uiterkamp, A. J. (2009). Dynamic temperature dependence patterns in future energy demand models in the context of climate change, *Energy* **34**(11): 1797–1806.
URL: <http://dx.doi.org/10.1016/j.energy.2009.07.037> Citado na página 39.
- Hossain, M. S. & Mahmood, H. (2020). Short-Term Load Forecasting Using an LSTM Neural Network, *2020 IEEE Power Energy Conf. Illinois, PECEI 2020*. Citado 2 vezes nas páginas 45 e 57.
- Huo, X., Wu, K., Miao, W., Wang, L., He, H. & Su, D. (2019). Research on Network Traffic Anomaly Detection of Source-Network-Load Industrial Control System Based on GRU-OCSVM, *IOP Conference Series: Earth and Environmental Science* **300**(4). Citado 2 vezes nas páginas 34 e 35.
- Imani, M. & Ghassemian, H. (2019). Residential load forecasting using wavelet and collaborative representation transforms, *Applied Energy* **253**(February): 113505.
URL: <https://doi.org/10.1016/j.apenergy.2019.113505> Citado na página 46.

- Islam, M. A., Che, H. S., Hasanuzzaman, M. & Rahim, N. A. (2019). *Energy demand forecasting*, Elsevier Inc.
URL: <http://dx.doi.org/10.1016/B978-0-12-814645-3.00005-5> Citado na página 16.
- Islam, M. R., Al Mamun, A., Sohel, M., Hossain, M. L. & Uddin, M. M. (2020). LSTM-Based Electrical Load Forecasting for Chattogram City of Bangladesh, pp. 188–192. Citado 2 vezes nas páginas 43 e 58.
- Jiang, Q., Zhu, J. X., Li, M. & Qing, H. Y. (2018). Electricity Power Load Forecast via Long Short-Term Memory Recurrent Neural Networks, *Proc. - 2018 4th Annu. Int. Conf. Netw. Inf. Syst. Comput. ICNISC 2018* pp. 265–268. Citado 3 vezes nas páginas 43, 57 e 58.
- Kamath, U., Liu, J. & Whitaker, J. (2019). *Deep Learning for NLP and Speech Recognition*, 1 edn, Springer. Citado 3 vezes nas páginas 31, 32 e 33.
- Kandil, M. S., El-Debeiky, S. M. & Hasanien, N. E. (2002). Long-term load forecasting for fast developing utility using a knowledge-based expert system, *IEEE Trans. Power Syst.* **17**(2): 491–496. Citado na página 15.
- Khan, B., Khalid, R., Javed, M. U., Javaid, S., Ahmed, S. & Javaid, N. (2020). Short-Term Load and Price Forecasting based on Improved Convolutional Neural Network, *2020 3rd Int. Conf. Comput. Math. Eng. Technol. Idea to Innov. Build. Knowl. Econ. iCoMET 2020* . Citado 2 vezes nas páginas 41 e 58.
- Kingma, D. P. & Ba, J. L. (2015). Adam: A method for stochastic optimization, *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.* pp. 1–15. Citado na página 58.
- Kumar, S., Hussain, L., Banarjee, S. & Reza, M. (2018). Energy Load Forecasting using Deep Learning Approach-LSTM and GRU in Spark Cluster, *Proc. 5th Int. Conf. Emerg. Appl. Inf. Technol. EAIT 2018* pp. 1–4. Citado na página 44.
- Lang, C., Steinborn, F., Steffens, O. & Lang, E. W. (2019). Electricity load forecasting - An evaluation of simple 1D-CNN network structures, *arXiv* . Citado na página 58.
- Laszuk, D. (2017). Python implementation of empirical mode decomposition algorithm, <https://github.com/laszukdawid/PyEMD>. Citado na página 57.
- Lee, K., Kim, J. K., Kim, J., Hur, K. & Kim, H. (2018). CNN and GRU combination scheme for Bearing Anomaly Detection in Rotating Machinery Health Monitoring, *1st IEEE International Conference on Knowledge Innovation and Invention, ICKII 2018* pp. 102–105. Citado 2 vezes nas páginas 34 e 37.
- Lekshmi, M. & Adithya Subramanya, K. N. (2019). Short-term load forecasting of 400kV grid substation using R-tool and study of influence of ambient temperature on the forecasted load, *2019 2nd Int. Conf. Adv. Comput. Commun. Paradig. ICACCP 2019* pp. 1–5. Citado na página 39.
- Li, K., Sun, Y., Li, S., Ma, X. & Zhang, C. (2019). Load forecasting method for CCHP system based on deep learning strategy using LSTM-RNN, *Proc. 14th IEEE Conf. Ind. Electron. Appl. ICIEA 2019* pp. 827–831. Citado 2 vezes nas páginas 44 e 55.
- Li, P., Luo, A., Liu, J., Wang, Y., Zhu, J., Deng, Y. & Zhang, J. (2020). Bidirectional gated recurrent unit neural network for Chinese address element segmentation, *ISPRS International Journal of Geo-Information* **9**(11). Citado 2 vezes nas páginas 35 e 37.
- Li, T., Qian, Z. & He, T. (2020). Short-Term Load Forecasting with Improved CEEMDAN and

- GWO-Based Multiple Kernel ELM, *Complexity* **2020**. Citado na página 48.
- Li, T., Wang, B., Zhou, M., Zhang, L. & Zhao, X. (2018). Short-Term Load Forecasting Using Optimized LSTM Networks Based on EMD, *10th Int. Conf. Commun. Circuits Syst. ICCAS 2018* pp. 84–88. Citado 3 vezes nas páginas 45, 57 e 58.
- Lin, H., Sun, Q. & Chen, S. Q. (2020). Reducing exchange rate risks in international trade: A hybrid forecasting approach of CEEMDAN and multilayer LSTM, *Sustainability (Switzerland)* **12**(6). Citado na página 48.
- Liu, C., Jin, Z., Gu, J. & Qiu, C. (2017). Short-term load forecasting using a long short-term memory network, *2017 IEEE PES Innov. Smart Grid Technol. Conf. Eur. ISGT-Europe 2017 - Proc.* **2018-Janua**: 1–6. Citado na página 43.
- Luo, X., Zhou, W., Wang, W., Zhu, Y. & Deng, J. (2018). Attention-Based Relation Extraction with Bidirectional Gated Recurrent Unit and Highway Network in the Analysis of Geological Data, *IEEE Access* **6**(c): 5705–5715. Citado na página 36.
- Lv, P., Liu, S., Yu, W., Zheng, S. & Lv, J. (2020). EGA-STLF: A Hybrid Short-Term Load Forecasting Model, *IEEE Access* **8**: 31742–31752. Citado na página 34.
- Ma, Y., Zhang, Q., Ding, J., Wang, Q. & Ma, J. (2019). Short Term Load Forecasting Based on iForest-LSTM, *14th IEEE Conf. Ind. Electron. Appl.* . Citado na página 44.
- Madhure, R. U., Raman, R. & Singh, S. K. (2020). CNN-LSTM based Electricity Theft Detector in Advanced Metering Infrastructure, *2020 11th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2020* . Citado na página 38.
- Maksut, M., Karbozov, A., Myrzaliyeva, M., Nunna, H. S., Jamwal, P. K. & Doolla, S. (2019). Deep Learning Based Short-Term Load Forecasting for Urban Areas, *2019 IEEE Ind. Appl. Soc. Annu. Meet. IAS 2019* pp. 1–6. Citado 2 vezes nas páginas 41 e 58.
- Mandic, D. P. & Chambers, J. A. (2001). *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*, Vol. 4, 1 edn, Wiley-Blackwell, New Jersey. Citado 5 vezes nas páginas 21, 22, 27, 29 e 31.
- Marino, D. L., Amarasinghe, K. & Manic, M. (2016). Building energy load forecasting using Deep Neural Networks, *IECON Proc. (Industrial Electron. Conf.)* pp. 7046–7051. Citado na página 44.
- Massaoudi, M., Refaat, S. S., Abu-Rub, H., Chihi, I. & Oueslati, F. S. (2020). PLS-CNN-BiLSTM: An end-to-end algorithm-based savitzky-golay smoothing and evolution strategy for load forecasting, *Energies* **13**(20): 1–29. Citado na página 51.
- Massaoudi, M., Refaat, S. S., Chihi, I., Trabelsi, M., Abu-Rub, H. & Oueslati, F. S. (2020). Short-Term Electric Load Forecasting Based on Data-Driven Deep Learning Techniques, *IECON Proceedings (Industrial Electronics Conference) 2020-October*: 2565–2570. Citado na página 50.
- Massaoudi, M., Refaat, S. S., Chihi, I., Trabelsi, M., Oueslati, F. S. & Abu-Rub, H. (2021). A novel stacked generalization ensemble-based hybrid LGBM-XGB-MLP model for Short-Term Load Forecasting, *Energy* **214**: 118874.
URL: <https://doi.org/10.1016/j.energy.2020.118874> Citado na página 23.
- McKinney, W. (2018). *Python Para Análise de Dados: Tratamento de Dados com Pandas*,

- NumPy e IPython*, 1 edn, Novatec Editora. Citado na página 57.
- Medsker, L. & Jain, L. (2001). *RECURRENT NEURAL NETWORKS Design and Applications*, CRC Press. Citado 5 vezes nas páginas 16, 24, 27, 28 e 30.
- Mitkov, A., Noorzad, N., Gabrovska-Evstatieva, K. & Mihailov, N. (2019). Forecasting the energy consumption in Afghanistan with the ARIMA model, *2019 16th Conf. Electr. Mach. Drives Power Syst. ELMA 2019 - Proc.* (June): 1–4. Citado na página 39.
- Morettin, P. & Toloí, C. (1987). *Análise de Séries temporais*, 2 edn, Atual, São Paulo. Citado na página 53.
- Müller, A. C. & Guido, S. (2016). *Introduction to Machine Learning with Python: A Guide for Data Scientists*, 1 edn, O'Reilly Media. Citado na página 57.
- Nair, A. & Joshi, S. K. (2010). Short Term Load Forecasting using probabilistic Neural Network based algorithm, *Proc. - 2010 Int. Conf. Comput. Intell. Commun. Networks, CICN 2010* pp. 128–132. Citado na página 41.
- Noureen, S., Atique, S., Roy, V. & Bayne, S. (2019). Analysis and application of seasonal ARIMA model in Energy Demand Forecasting: A case study of small scale agricultural load, *Midwest Symp. Circuits Syst.* **2019-Augus**: 521–524. Citado na página 39.
- Oduro-Gyimah, F. K. (2018). Application of recurrent neural network model in the analysis of electricity load demand in Ashanti Region of Ghana, *IEEE Int. Conf. Adapt. Sci. Technol. ICASST 2018-Augus*: 1–9. Citado 2 vezes nas páginas 42 e 58.
- Pan, X. & Lee, B. (2012). A comparison of support vector machines and artificial neural networks for mid-term load forecasting, *2012 IEEE Int. Conf. Ind. Technol. ICIT 2012, Proc.* pp. 95–101. Citado na página 15.
- Petroşanu, D. M. & Pîrjan, A. (2021). Electricity consumption forecasting based on a bidirectional long-short-term memory artificial neural network, *Sustainability (Switzerland)* **13**(1): 1–31. Citado na página 49.
- Rafiei, M., Niknam, T., Aghaei, J., Shafie-Khah, M. & Catalao, J. P. (2018). Probabilistic load forecasting using an improved wavelet neural network trained by generalized extreme learning machine, *IEEE Transactions on Smart Grid* **9**(6): 6961–6971. Citado na página 47.
- Rahman, A., Srikumar, V. & Smith, A. D. (2018). Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks, *Appl. Energy* **212**(October 2017): 372–385.
URL: <https://doi.org/10.1016/j.apenergy.2017.12.051> Citado na página 44.
- Saad, Z., Hazirah, A. J. N., Suziana, A., Azhar, M. A. A., Yaacob, Z., Ahmad, F. & Yusnita, M. A. (2017). Short-Term Load Forecasting of 415V, 11kV and 33kV Electrical Systems using MLP Network, *2017 Int. Conf. Robot. Autom. Sci.* . Citado 2 vezes nas páginas 41 e 58.
- Sajjad, M., Khan, Z. A., Ullah, A., Hussain, T., Ullah, W., Lee, M. Y. & Baik, S. W. (2020). A Novel CNN-GRU-Based Hybrid Approach for Short-Term Residential Load Forecasting, *IEEE Access* **8**: 143759–143768. Citado na página 37.
- Setiawan, A., Koprinska, I. & Agelidis, V. G. (2009). Very short-term electricity load demand forecasting using support vector regression, *Proc. Int. Jt. Conf. Neural Networks* pp. 2888–

2894. Citado na página 15.

- Shahidehpour, M., Yamin, H. & Li, Z. (2002). *Market Operations in Electric Power Systems: Forecasting, Scheduling, and Risk Management*, 1 edn, Wiley-IEEE Press, New York. Citado na página 16.
- Singh, A. K. & Khatoun, S. (2013). An Overview of Electricity Demand Forecasting Techniques, *Natl. Conf. Emerg. Trends Electr. Instrum. Commun. Eng.* **3**(3): 38–48. Citado 2 vezes nas páginas 15 e 16.
- Soliman, S. A. & Al-Kandari, A. M. (2010). *Electrical Load Forecasting: Modeling and Model Construction*, 1 edn, Butterworth-Heinemann, Oxônia. Citado na página 15.
- Somu, N., Raman M R, G. & Ramamritham, K. (2021). A deep learning framework for building energy consumption forecast, *Renewable and Sustainable Energy Reviews* **137**(April 2020): 110591.
URL: <https://doi.org/10.1016/j.rser.2020.110591> Citado na página 23.
- Sudheer, G. & Suseelatha, A. (2015). Short term load forecasting using wavelet transform combined with Holt-Winters and weighted nearest neighbor models, *International Journal of Electrical Power and Energy Systems* **64**: 340–346.
URL: <http://dx.doi.org/10.1016/j.ijepes.2014.07.043> Citado na página 47.
- Talathi, S. S. & Vartak, A. (2015). Improving performance of recurrent neural network with relu nonlinearity.
URL: <http://arxiv.org/abs/1511.03771> Citado na página 58.
- Tao, Q., Liu, F., Li, Y. & Sidorov, D. (2019). Air Pollution Forecasting Using a Deep Learning Model Based on 1D Convnets and Bidirectional GRU, *IEEE Access* **7**: 76690–76698. Citado na página 36.
- Tian, C., Ma, J., Zhang, C. & Zhan, P. (2018). A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network, *Energies* **11**(12). Citado 2 vezes nas páginas 37 e 38.
- Tjøstheim, D. (1994). Non-linear time series: a selective review, *Scandinavian Journal of Statistics* **21**(2): 97–130.
URL: <http://www.jstor.org/stable/4616304> Citado na página 21.
- Tudose, A. M., Sidea, D. O., Picioroaga, I. I., Boicea, V. A. & Bulac, C. (2020). A CNN Based Model for Short-Term Load Forecasting: A Real Case Study on the Romanian Power System, *UPEC 2020 - 2020 55th International Universities Power Engineering Conference, Proceedings* (1). Citado na página 38.
- Upadhaya, D., Thakur, R. & Singh, N. K. (2019). A systematic review on the methods of short term load forecasting, *2019 2nd Int. Conf. Power Energy Environ. Intell. Control. PEEIC 2019* pp. 6–11. Citado 2 vezes nas páginas 39 e 57.
- Vanderplas, J. (2016). *Python Data Science Handbook: Essential Tools for Working with Data*, 1 edn, O'Reilly. Citado 2 vezes nas páginas 56 e 57.
- Wang, Y., Liao, W. & Chang, Y. (2018). Gated Recurrent Unit Network-Based Short-Term, *Energies* pp. 1–14. Citado 2 vezes nas páginas 34 e 35.
- Wu, K., Wu, J., Feng, L., Yang, B., Liang, R., Yang, S. & Zhao, R. (2021). An attention-based CNN-LSTM-BiLSTM model for short-term electric load forecasting in integrated energy

- system, *International Transactions on Electrical Energy Systems* **31**(1): 1–15. Citado 2 vezes nas páginas 38 e 50.
- Xie, X., Wang, B., Wan, T. & Tang, W. (2020). Multivariate Abnormal Detection for Industrial Control Systems Using 1D CNN and GRU, *IEEE Access* **8**: 88348–88359. Citado na página 34.
- Xiuyun, G., Ying, W., Yang, G., Chengzhi, S., Wen, X. & Yimiao, Y. (2018). Short-term Load Forecasting Model of GRU Network Based on Deep Learning Framework, *2nd IEEE Conference on Energy Internet and Energy System Integration, EI2 2018 - Proceedings* pp. 1–4. Citado 2 vezes nas páginas 34 e 35.
- Yan, K., Li, W., Ji, Z., Qi, M. & Du, Y. (2019). A Hybrid LSTM Neural Network for Energy Consumption Forecasting of Individual Households, *IEEE Access* **7**: 157633–157642. Citado na página 46.
- Yu, D. & Deng, L. (2014). *Automatic Speech Recognition: A Deep Learning Approach*, 1 edn, Springer. Citado 2 vezes nas páginas 31 e 32.
- Zhai, M. Y. (2015). A new method for short-term load forecasting based on fractal interpretation and wavelet analysis, *International Journal of Electrical Power and Energy Systems* **69**: 241–245.
URL: <http://dx.doi.org/10.1016/j.ijepes.2014.12.087> Citado na página 47.
- Zhang, D., Tian, L., Hong, M., Han, F., Ren, Y. & Chen, Y. (2018). Combining convolution neural network and bidirectional gated recurrent unit for sentence semantic classification, *IEEE Access* **6**(8): 73750–73759. Citado 2 vezes nas páginas 36 e 38.
- Zhang, Q., Ma, Y., Li, G., Ma, J. & Ding, J. (2020). Short-Term Load Forecasting Based on Frequency Domain Decomposition and Deep Learning, *Mathematical Problems in Engineering* **2020**. Citado na página 46.
- Zhang, Z. & Hong, W. C. (2019). Electric load forecasting by complete ensemble empirical mode decomposition adaptive noise and support vector regression with quantum-based dragonfly algorithm, *Nonlinear Dynamics* **98**(2): 1107–1136.
URL: <https://doi.org/10.1007/s11071-019-05252-7> Citado na página 47.
- Zhao, H. & Tang, Z. (2016). The review of demand side management and load forecasting in smart grid, *Proc. World Congr. Intell. Control Autom.* **2016-Septe**: 625–629. Citado na página 40.
- Zheng, H., Yuan, J. & Chen, L. (2017). Short-Term Load Forecasting Using EMD-LSTM neural networks with a xgboost algorithm for feature importance evaluation, *Energies* **10**(8). Citado na página 46.
- Zhou, Z., Wu, G. & Zhang, X. (2020). Short-term Load Forecasting Model Based on IBFO-BILSTM, *IOP Conference Series: Earth and Environmental Science* **440**(3). Citado na página 49.
- Zor, K., Timur, O. & Teke, A. (2017). A state-of-the-art review of artificial intelligence techniques for short-term electric load forecasting, *2017 6th Int. Youth Conf. Energy, IYCE 2017*. Citado na página 40.

Apêndice A

Códigos em *Python*

```
import timeit
start = timeit.default_timer()

# import packages
from math import sqrt
from numpy import concatenate
from matplotlib import pyplot
from pandas import read_csv
from pandas import DataFrame
from pandas import concat
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import mean_squared_error
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Flatten
from keras.layers import LSTM
import numpy

# convert series to supervised learning
def series_to_supervised(data, n_in=1, n_out=1, droppan=True):
    n_vars = 1 if type(data) is list else data.shape[1]
    df = DataFrame(data)
    cols, names = list(), list()
    # input sequence (t-n, ... t-1)
    for i in range(n_in, 0, -1):
        cols.append(df.shift(i))
        names += [('var%d(t-%d)' % (j+1, i)) for j in range(n_vars)]
    # forecast sequence (t, t+1, ... t+n)
    for i in range(0, n_out+n_h):
        cols.append(df.shift(-i))
        if i == 0:
            names += [('var%d(t)' % (j+1)) for j in range(n_vars)]
        else:
            names += [('var%d(t+%d)' % (j+1, i)) for j in range(n_vars)]
    # put it all together
    agg = concat(cols, axis=1)
    agg.columns = names
    # drop rows with NaN values
    if droppan:
        agg.dropna(inplace=True)
    return agg
```

Figura A.1: Primeira parte do código da rede BiGRU_CNN

Fonte: (Autor, 2021).

```

# Load dataset
dataset = read_csv('demanda_energia_temperatura.csv', header=0, index_col=0)
values = dataset.values
# ensure all data is float
values = values.astype('float32')
# normalize features
scaler = MinMaxScaler(feature_range=(0, 1))
scaled = scaler.fit_transform(values)
# specify the number of window size
window_size = 24
# specify the number of features
n_features = 3
# specify the number of prediction horizon
n_h=0
# frame as supervised learning
reframed = series_to_supervised(scaled, window_size, 1)

# split into train and test sets
values = reframed.values
n_trainwindow_size = 36199-window_size-n_h
train = values[:n_trainwindow_size, :]
test = values[n_trainwindow_size:, :]

# split into input and outputs
n_obs = window_size * n_features
train_X, train_y = train[:, :n_obs], train[:, -n_features]
test_X, test_y = test[:, :n_obs], test[:, -n_features]

# reshape input to be 3D [samples, timesteps, features]
train_X = train_X.reshape((train_X.shape[0], window_size, n_features))
test_X = test_X.reshape((test_X.shape[0], window_size, n_features))

```

Figura A.2: Segunda parte do código da rede BiGRU_CNN

Fonte: (Autor, 2021).

```

# design network
model = Sequential()
model.add(Bidirectional(GRU(10, activation='relu', input_shape=(train_X.shape[1], train_X.shape[2]), return_sequences=True)))
model.add(Conv1D(filters=8, kernel_size=6, activation='relu', input_shape=(train_X.shape[1], train_X.shape[2])))
model.add(MaxPooling1D(pool_size=2))
model.add(Flatten())
model.add(Dense(1))
model.compile(loss = 'mean_squared_error', optimizer = 'adam', metrics = ['accuracy'])
# fit network
history = model.fit(train_X, train_y, epochs=150, validation_data=(test_X, test_y), batch_size=32, verbose=2, shuffle=False)

# plot history
plt.figure(figsize=(15,5))
plt.plot(history.history['loss'], label='amostra_treino')
plt.plot(history.history['val_loss'], label='amostra_validação')
plt.title('treino x validação')
#plota o nome do eixo y
plt.ylabel('Erro Médio Absoluto')
#plota o nome do eixo x
plt.xlabel('Época')
#mostra o gráfico
plt.legend()
#mostra o gráfico
plt.show()

# make a prediction
yhat = model.predict(test_X)
test_X = test_X.reshape((test_X.shape[0], window_size*n_features))
# invert scaling for forecast
inv_yhat = concatenate((yhat, test_X[:, -2:]), axis=1)
inv_yhat = scaler.inverse_transform(inv_yhat)
inv_yhat = inv_yhat[:,0]
# invert scaling for actual
test_y = test_y.reshape((len(test_y), 1))
inv_y = concatenate((test_y, test_X[:, -2:]), axis=1)
inv_y = scaler.inverse_transform(inv_y)
inv_y = inv_y[:,0]

# reshape input to be 3D [samples, timesteps, features]
train_X = train_X.reshape((train_X.shape[0], window_size, n_features))

```

Figura A.3: Terceira parte do código da rede BiGRU_CNN

Fonte: (Autor, 2021).

```

# make a prediction
yhat = model.predict(test_X)
test_X = test_X.reshape((test_X.shape[0], window_size*n_features))
# invert scaling for forecast
inv_yhat = concatenate((yhat, test_X[:, -2:]), axis=1)
inv_yhat = scaler.inverse_transform(inv_yhat)
inv_yhat = inv_yhat[:,0]
# invert scaling for actual
test_y = test_y.reshape((len(test_y), 1))
inv_y = concatenate((test_y, test_X[:, -2:]), axis=1)
inv_y = scaler.inverse_transform(inv_y)
inv_y = inv_y[:,0]

# reshape input to be 3D [samples, timesteps, features]
train_X = train_X.reshape((train_X.shape[0], window_size, n_features))

# make a prediction
yhat_train = model.predict(train_X)
train_X = train_X.reshape((train_X.shape[0], window_size*n_features))

# invert scaling for forecast
inv_yhat_train = concatenate((yhat_train, train_X[:, -2:]), axis=1)
inv_yhat_train = scaler.inverse_transform(inv_yhat_train)
inv_yhat_train = inv_yhat_train[:,0]

# invert scaling for actual
train_y = train_y.reshape((len(train_y), 1))
inv_y_train = concatenate((train_y, train_X[:, -2:]), axis=1)
inv_y_train = scaler.inverse_transform(inv_y_train)
inv_y_train = inv_y_train[:,0]

observacao=numpy.concatenate((inv_y_train,inv_y))

y_plot = numpy.empty_like(observacao)
y_ploy = numpy.empty_like(observacao)
y_plot[:] = numpy.nan
y_plot[len(inv_y_train):] = inv_yhat.reshape(-1)

```

Figura A.4: Quarta parte do código da rede BiGRU_CNN

Fonte: (Autor, 2021).

```

n_test= 1
novo=scaled[-1]
novo=concatenate((scaled[-24],scaled[-23],scaled[-22],scaled[-21],scaled[-20],scaled[-19],scaled[-18],scaled[-17],scaled[-16],scaled[-15],scaled[-14],scaled[-13],scaled[-12],scaled[-11],scaled[-10],scaled[-9],scaled[-8],scaled[-7],scaled[-6],scaled[-5],scaled[-4],scaled[-3],scaled[-2],scaled[-1]))
print('-----')
print(novo)
previsao = list()

for i in range (n_test):
    novo = novo.reshape((1, window_size, n_features))
    X1=model.predict(novo)
    print('-----')
    print(X1)
    X1=numpy.array([X1,X1,X1])
    X1=X1.reshape(1,3)
    print('-----')
    print(X1)
    novo = novo.reshape((novo.shape[0], window_size*n_features))
    X2=scaler.inverse_transform(X1)
    print('-----')
    print(X2)
    previsao.append(X2[0][0].tolist())
    novo=concatenate((novo, X1), axis=1)
#    tenha que mexer aqui
    novo=numpy.delete(novo, [0])
    novo=numpy.delete(novo, numpy.s_[-n_features+1:])

plt.figure(figsize=(15,5))
plt.plot(previsao)
plt.title('previsão')
plt.ylabel('Potência')
plt.xlabel('Data')
plt.show()

numpy.savetxt('treino0.txt', inv_yhat_train)
numpy.savetxt('validacao0.txt', inv_yhat)
numpy.savetxt('previsao0.txt', previsao)

stop = timeit.default_timer()
print('Time: ', stop - start)

```

Figura A.5: Quinta parte do código da rede BiGRU_CNN

Fonte: (Autor, 2021).

```

from pandas import read_csv
from matplotlib import pyplot

dataset = read_csv('amostra_teste.csv', header=0, index_col=0)
prev_uni_1 = read_csv('prev_uni_1.csv', header=0, index_col=0,encoding='latin-1')
prev_uni_2 = read_csv('prev_uni_2.csv', header=0, index_col=0,encoding='latin-1')
prev_uni_3 = read_csv('prev_uni_3.csv', header=0, index_col=0,encoding='latin-1')
prev_uni_4 = read_csv('prev_uni_4.csv', header=0, index_col=0,encoding='latin-1')
prev_uni_5 = read_csv('prev_uni_5.csv', header=0, index_col=0,encoding='latin-1')
prev_uni_6 = read_csv('prev_uni_6.csv', header=0, index_col=0,encoding='latin-1')
prev_uni_7 = read_csv('prev_uni_7.csv', header=0, index_col=0,encoding='latin-1')
prev_uni_8 = read_csv('prev_uni_8.csv', header=0, index_col=0,encoding='latin-1')
prev_uni_9 = read_csv('prev_uni_9.csv', header=0, index_col=0,encoding='latin-1')
prev_uni_10 = read_csv('prev_uni_10.csv', header=0, index_col=0,encoding='latin-1')

plt.rcParams.update({'font.size': 12})
plt.figure(figsize=(18,8))
plt.plot(dataset, color = 'blue', label='Observações')
plt.plot(prev_uni_1, color = 'red', label='BiGRU_CNN 1')
plt.plot(prev_uni_2, color = 'green', label='BiGRU_CNN 2')
plt.plot(prev_uni_3, color = 'yellow', label='BiGRU_CNN 3')
plt.plot(prev_uni_4, color = 'black', label='BiGRU_CNN 4')
plt.plot(prev_uni_5, color = 'gray', label='BiGRU_CNN 5')
plt.plot(prev_uni_6, color = 'purple', label='BiGRU_CNN 6')
plt.plot(prev_uni_7, color = 'cyan', label='BiGRU_CNN 7')
plt.plot(prev_uni_8, color = 'orange', label='BiGRU_CNN 8')
plt.plot(prev_uni_9, color = 'pink', label='BiGRU_CNN 9')
plt.plot(prev_uni_10, color = 'olive', label='BiGRU_CNN 10')
plt.ylabel('Demanda de energia em MW')
plt.xlabel('Horizonte de previsão em horas')

plt.legend()
plt.show()

```

Figura A.6: Código do gráfico das previsões efetivas

Fonte: (Autor, 2021).

Apêndice B

Gráficos das IMFs da Decomposição CEENDAM

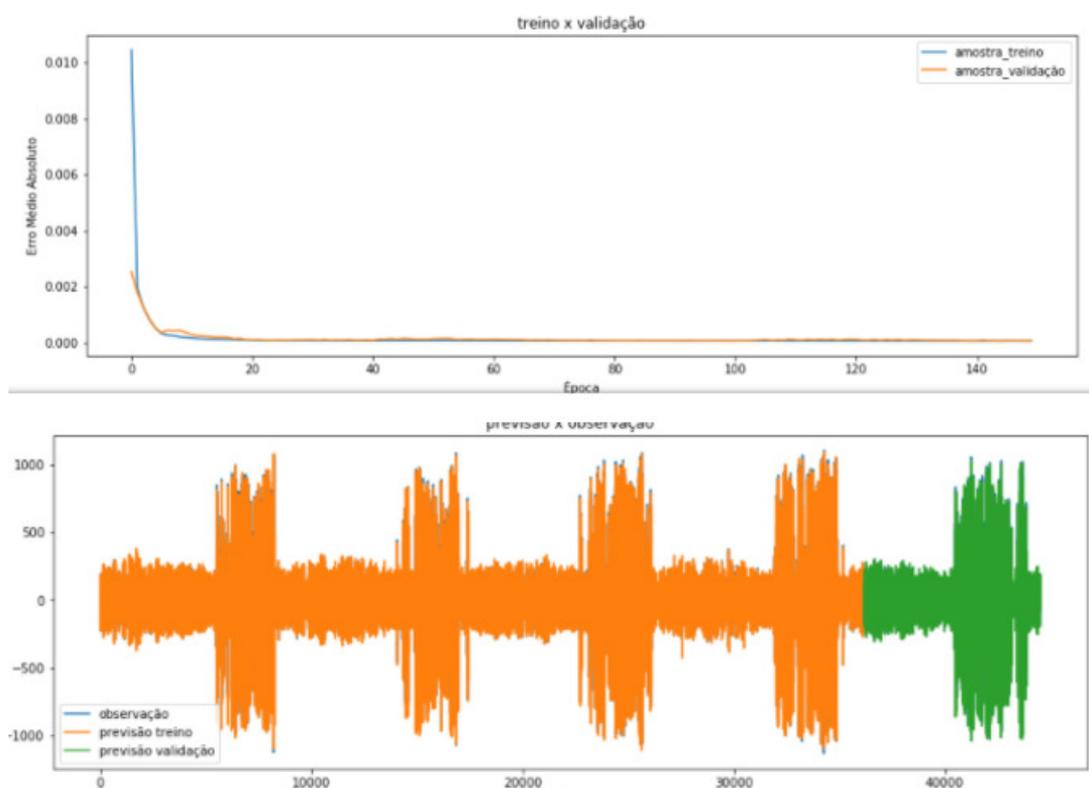


Figura B.1: IMF 2
Fonte: (Autor, 2021).

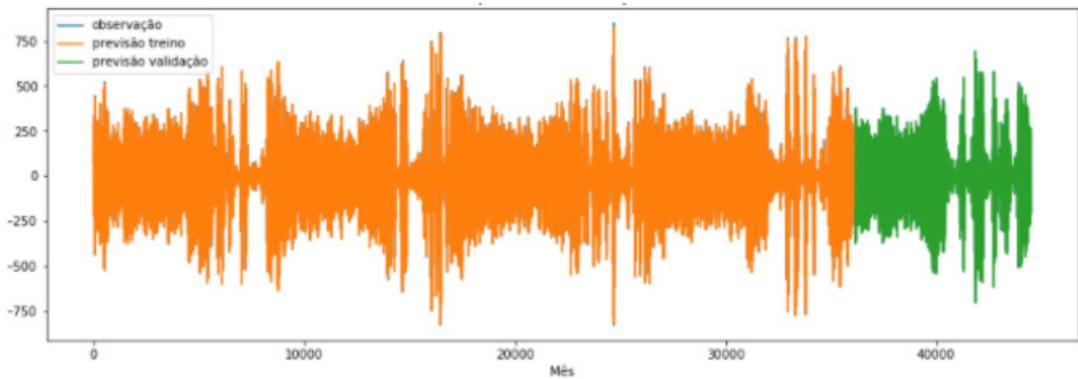
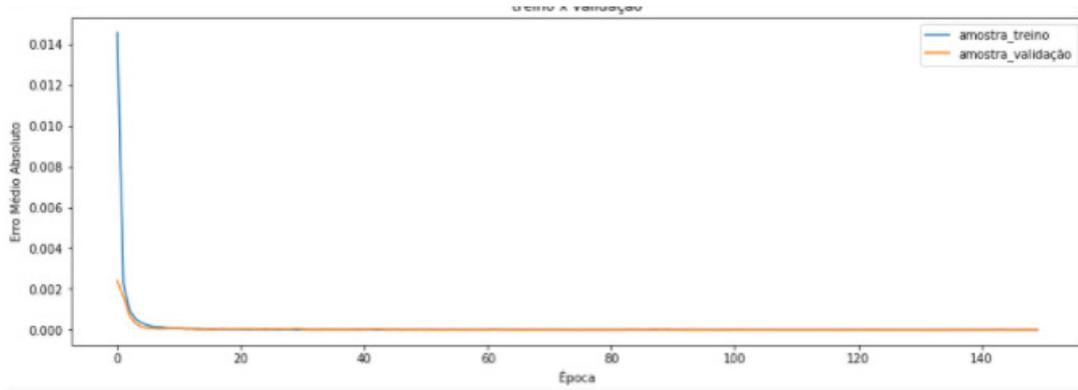


Figura B.2: IMF 3
Fonte: (Autor, 2021).

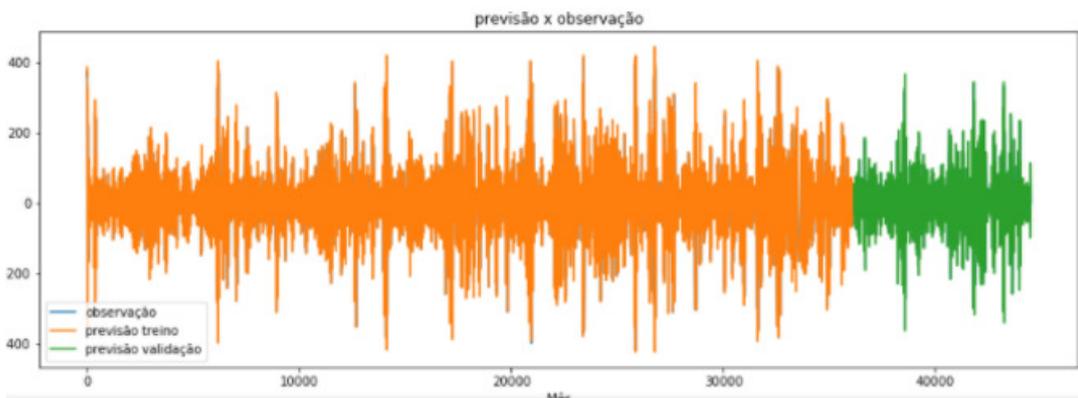
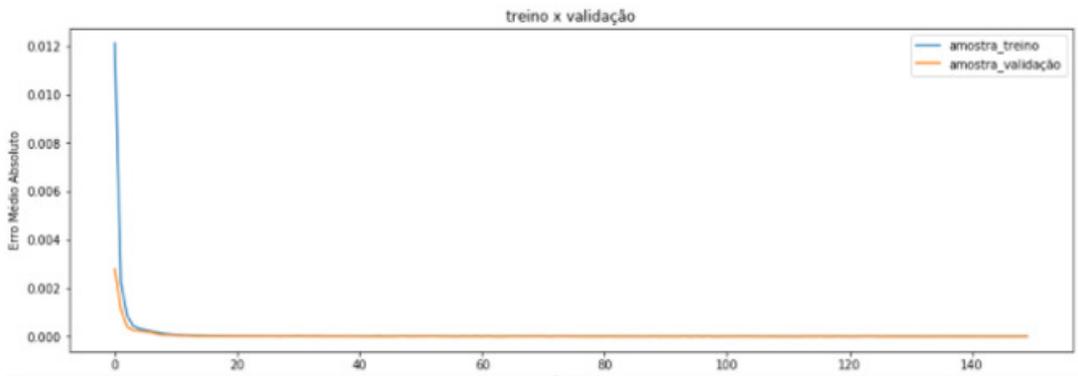


Figura B.3: IMF 4
Fonte: (Autor, 2021).

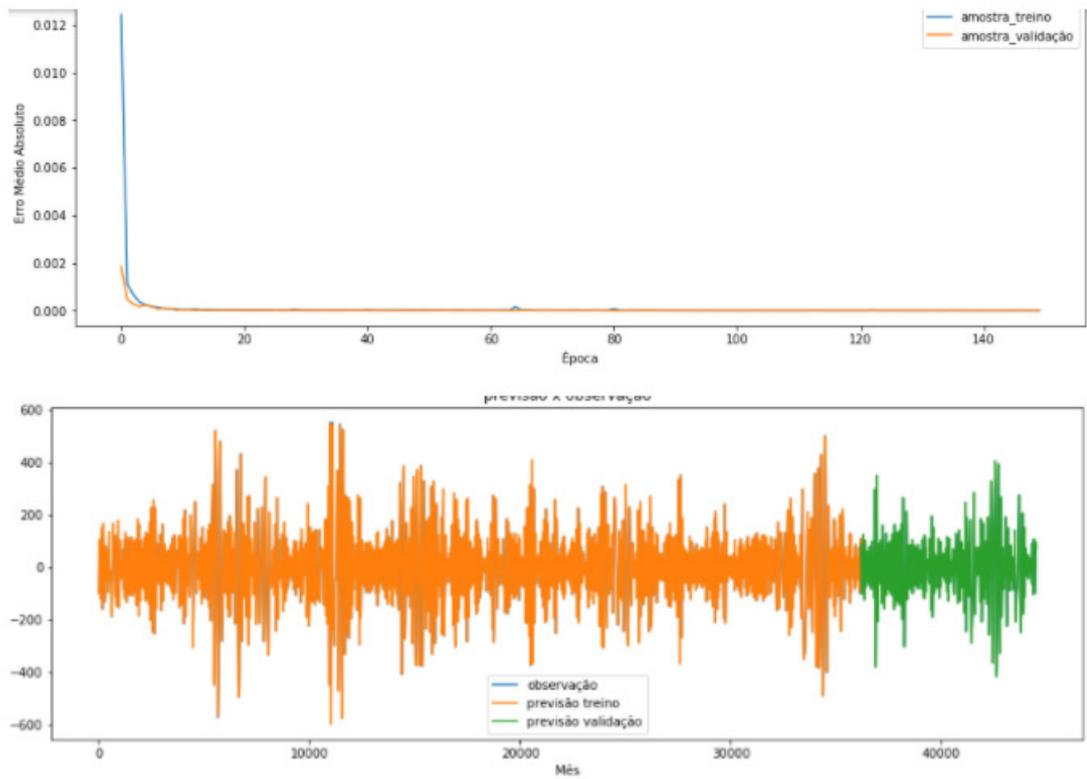


Figura B.4: IMF 5
Fonte: (Autor, 2021).

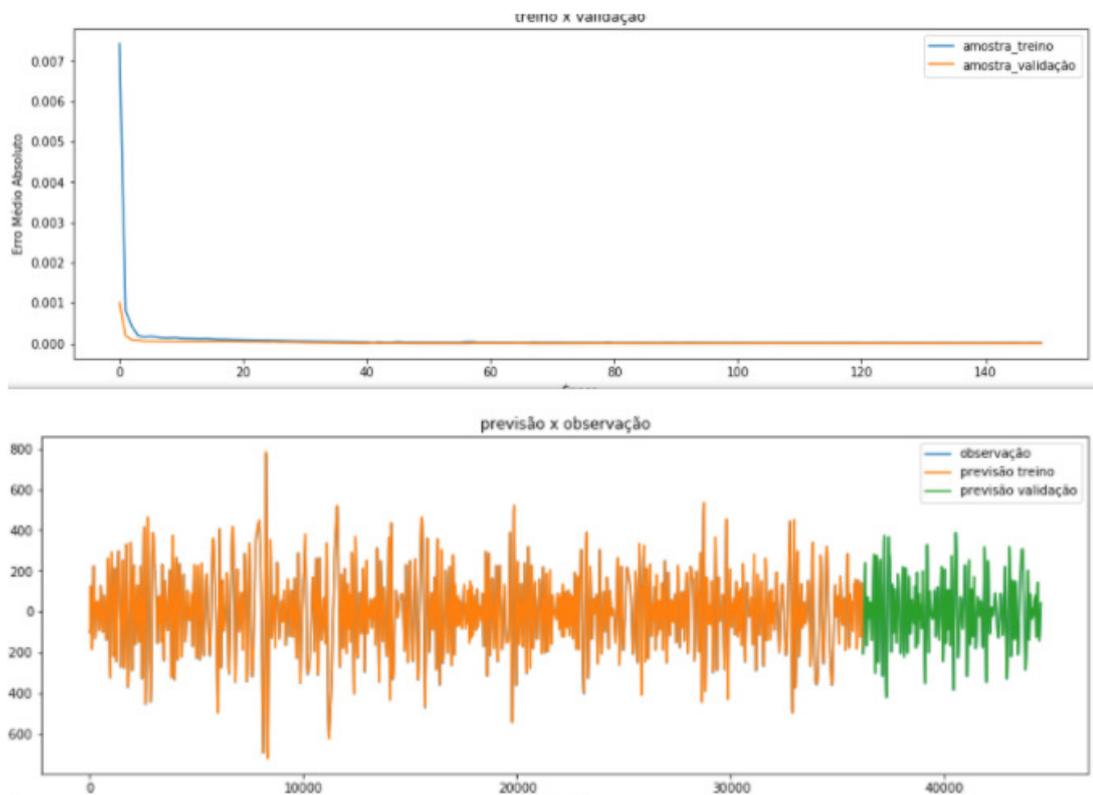


Figura B.5: IMF 6
Fonte: (Autor, 2021).

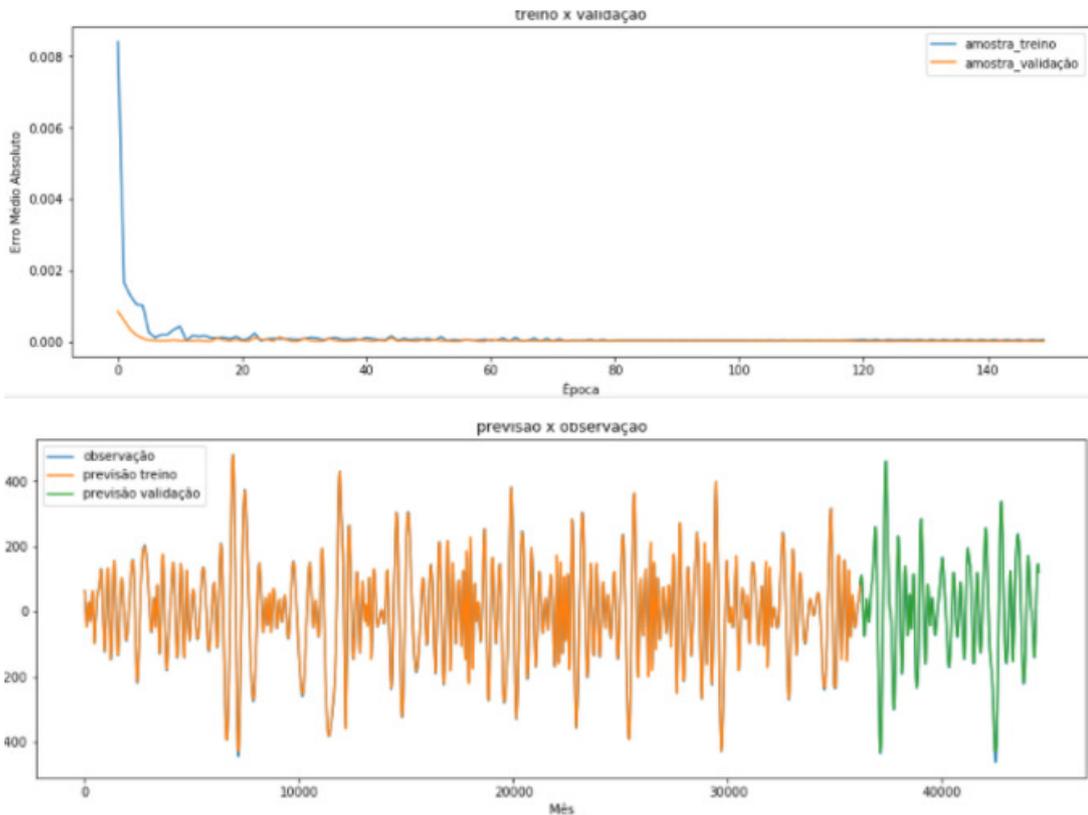


Figura B.6: IMF 7
Fonte: (Autor, 2021).

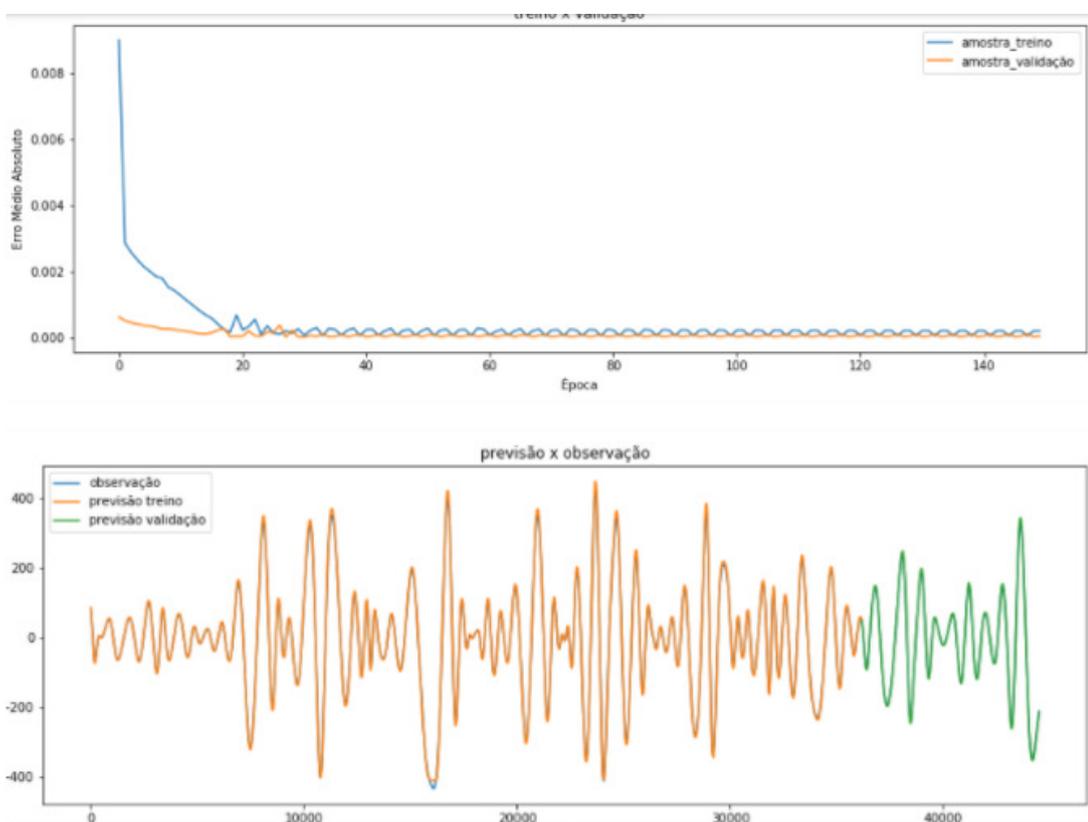


Figura B.7: IMF 8
Fonte: (Autor, 2021).

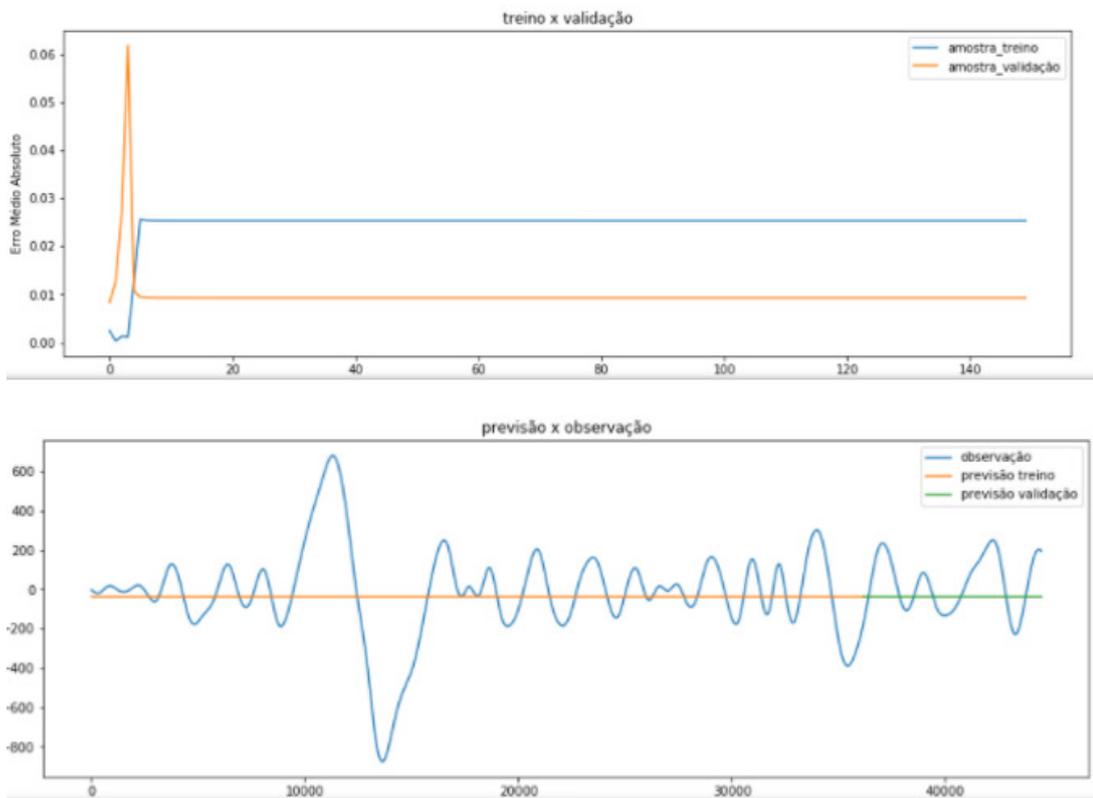


Figura B.8: IMF 9
 Fonte: (Autor, 2021).

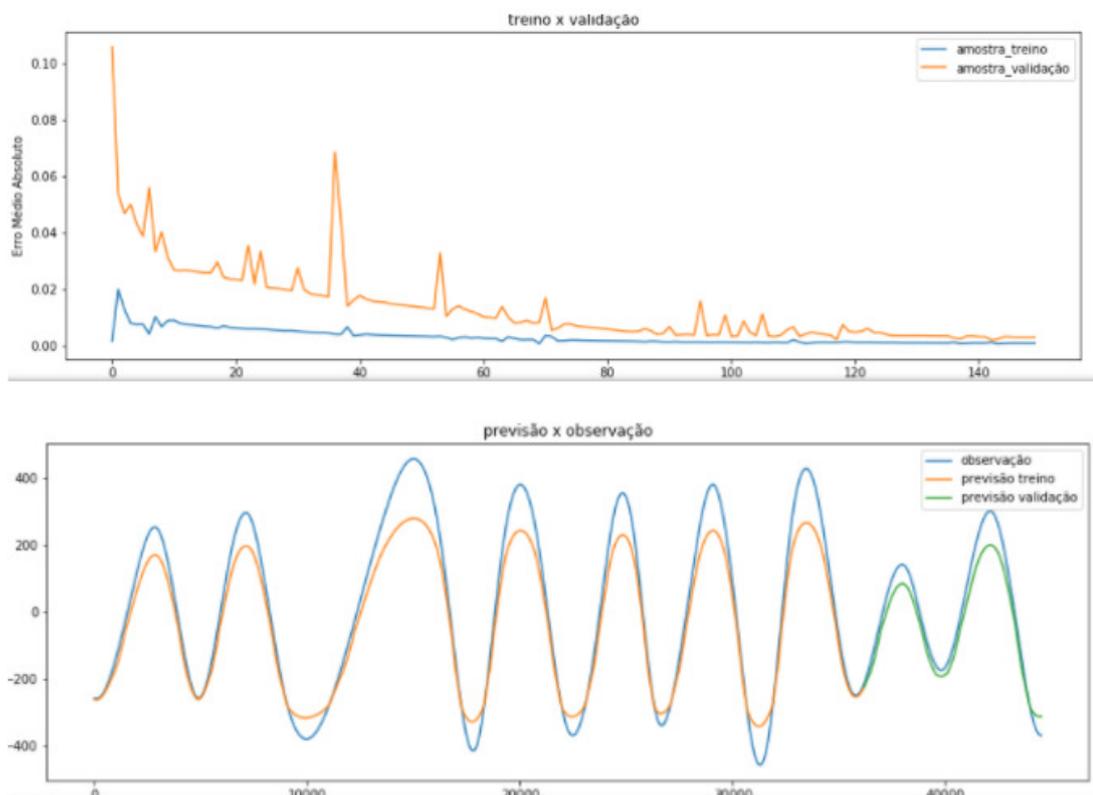


Figura B.9: IMF 10
 Fonte: (Autor, 2021).

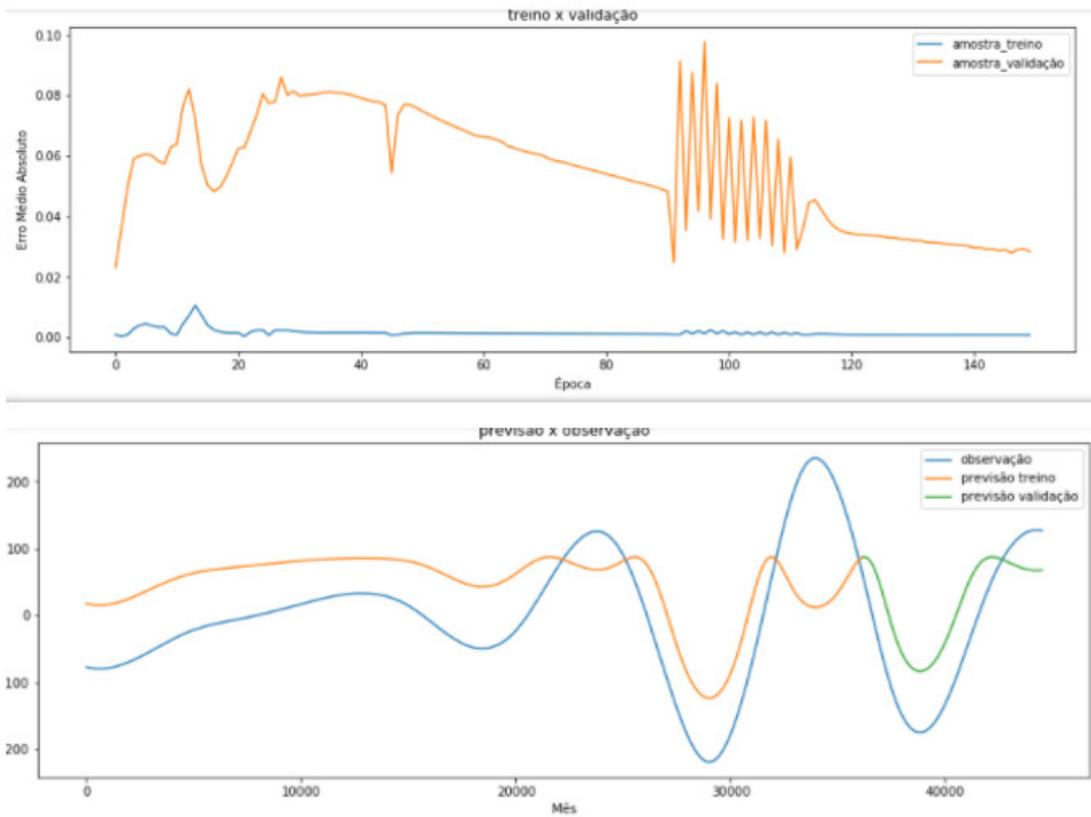


Figura B.10: IMF 11
Fonte: (Autor, 2021).

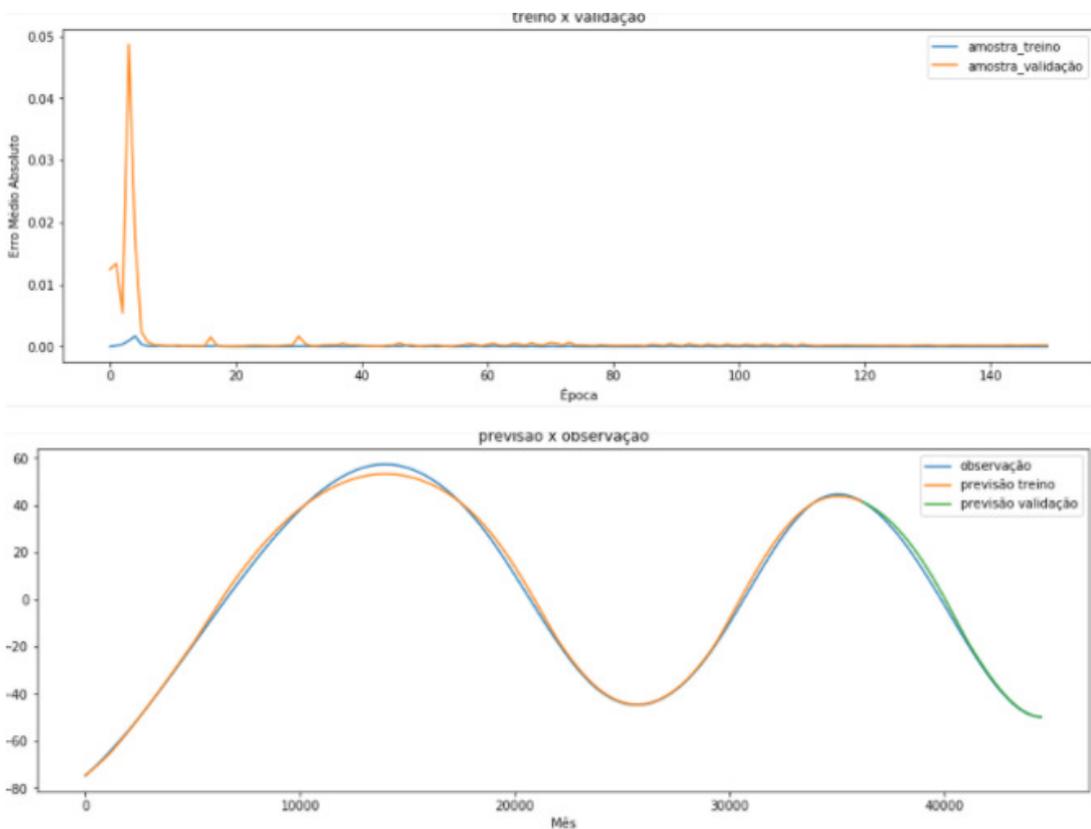


Figura B.11: IMF 12
Fonte: (Autor, 2021).

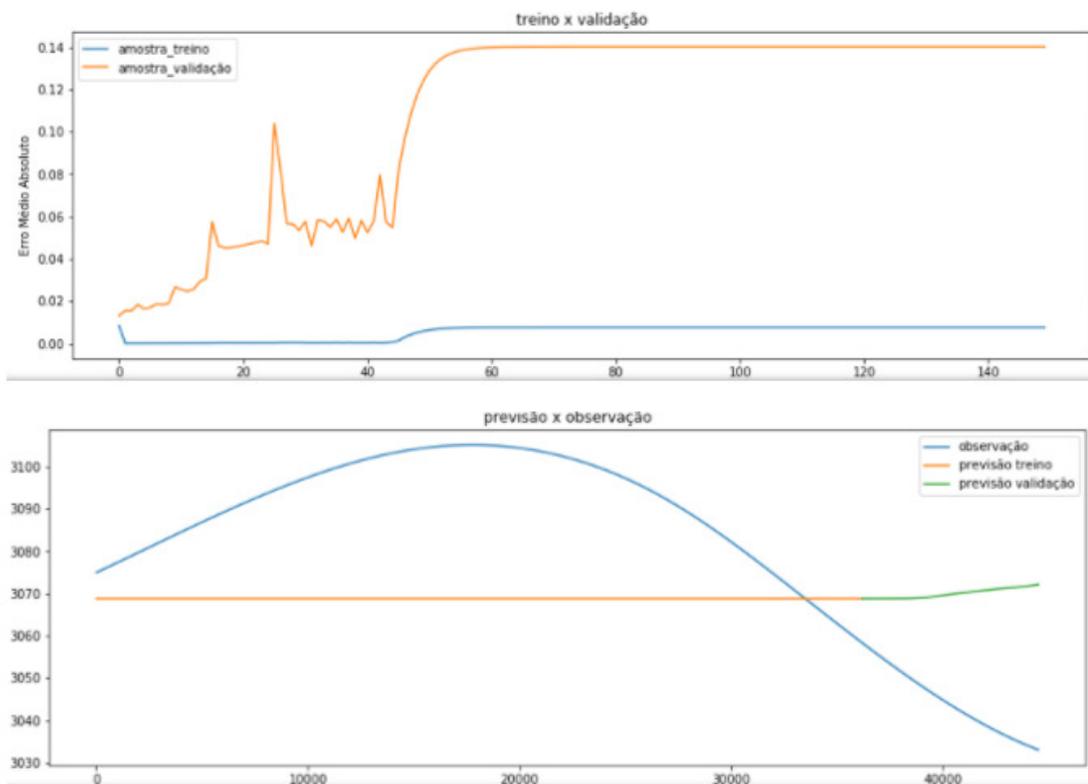


Figura B.12: IMF 13
Fonte: (Autor, 2021).

Apêndice C

Artigo Aceito para Publicação na Revista
Production

BiGRU-CNN neural network applied to short-term electric load forecasting

Lucas Duarte Soares^{a*} , Edgar Manuel Carreño Franco^{a**} 

^aUniversidade Estadual do Oeste do Paraná, Foz do Iguaçu, PR, Brasil

*lucas.2012@alunos.utfpr.edu.br,**emfra.unioeste@gmail.com

Abstract

Paper aims: This study analyzed the feasibility of the BiGRU-CNN artificial neural network as a forecasting tool for short-term electric load. This forecasting model can serve as a support tool related to decision-making by companies in the energy sector.

Originality: Despite a large amount of scientific research in this area, the literature still searches for more assertive forecasting models regarding short-term electric load. Thus, the BiGRU-CNN model, based on layers of BiGRU and CNN architecture networks was tested. This model was already proposed and used for other similar tasks, however, it has not been used on load forecasting.

Research method: The code was programmed in *Python* using the *keras* package. The forecasts of all networks were carried out 10 times until an acceptable statistical sample was reached so that future electric load values are as close as possible to reality.

Main findings: The best forecasting model was the proposed BiGRU-CNN network when compared to classical and some hybrid networks.

Implications for theory and practice: This methodology can be applied to short-term electric load forecasting problems. There is evidence that the combination of different layers of neural networks can provide more efficient forecasting results than classical networks with only one architecture.

Keywords

Time series forecasting. Recurrent neural networks. Artificial intelligence. Machine learning.

How to cite this article: Soares, L. D., & Franco, E. M. C. (2022). Bigru-CNN neural network applied to short-term electric load forecasting. *Production*, 32, e20210087. <https://doi.org/10.1590/0103-6513.20210087>

Received: July 12, 2021; Accepted: Oct. 27, 2021.

1. Introduction

Economic development around the world depends almost exclusively on the availability of electricity in industries, as most of them use it to carry out their vital productive activities (Soliman & Al-Kandari, 2010). Therefore, the electric load forecast is a decision support tool used by companies in the electricity sector to ensure an efficient service (Hahn et al., 2009), as the lack of electricity has a direct impact on the economy and financial health around the world.

There are four types of electric load forecast horizons to achieve different planning objectives and also to assist in the monitoring of critical conditions in the electrical system. According to Setiawan et al. (2009), these forecast horizons can be classified as very short term, short term, medium term, and long term.

Very short-term energy demand forecasts provide future values between one minute and one hour to determine the best strategy for the use of resources during energy generation (Charytoniuk & Chen, 2000). Short-term forecasts are carried out between one hour and one week to assist in operational planning, as electric load is defined one day before its production around the world (Chapagain et al., 2020). Medium-term forecasts are



made between one week and one month aiming to seek higher profits in the electricity market (Pan & Lee, 2012). Finally, long-term forecasts are conducted above one year, being used as a support tool in the dimensioning of new installations of electricity generation, transmission, and distribution companies (Kandil et al., 2002).

According to Ghalekhondabi et al. (2017), knowing the great importance of electric load forecasts for the electricity sector, published articles related to the subject have grown exponentially in recent years, including studies on short-term electric load forecasting. These studies use techniques that, according to Singh & Khatoon (2013), can be divided into three major groups, namely: traditional techniques formed by regression models (Dudek, 2016), multiple regression (Dhaval & Deshpande, 2020; Johannesen et al., 2019; Saber & Alam, 2018), exponential smoothing (Mayrink & Hippert, 2016; Mohammed et al., 2017; Rendon-Sanchez & Menezes, 2019), modified traditional techniques composed by autoregressive integrated moving average models (Alberg & Last, 2018; Amin & Hoque, 2019; Wu et al., 2020a), support vector machine (Chen et al., 2017; Jiang et al., 2018; Li et al., 2018), and computational techniques. According to Carpinteiro & Silva (2000), the traditional and modified traditional techniques present linear forecasting models, thus showing difficulties in forecasting electric load, as their relationship with exogenous variables is complex and non-linear. The computational techniques, also known as artificial intelligence models, has gained notoriety for its satisfactory performance in these scenarios in which linear models present some difficulty (Singh & Khatoon, 2013).

Some models based on artificial intelligence have been used to forecast short-term electric load, namely (Shahidehpour et al., 2002): expert systems (Kandil et al., 2002; Marković & Fraissler, 1993; Rahman & Hazim, 1996), evolutionary computing (Huang & Yang, 1995; Yang et al., 1996), fuzzy systems (Cerne et al., 2018; Coelho et al., 2016; Mukhopadhyay et al., 2018), artificial neural networks (Chandramitasari et al., 2018), and hybrid models (Fallah et al., 2019; Massaoudi et al., 2021; Yan et al., 2019). Regarding these models, artificial neural networks have received higher attention because their models are more accurate and easier to be implemented and have good performance (Shahidehpour et al., 2002). A considerable number of scientific papers using these forecasting neural models to estimate short-term energy requirements are found in the literature because of these specificities.

Recurrent networks are a peculiar type of artificial neural networks that have become the focus of many studies because of their ability to process sequential and temporal information (Medsker & Jain, 2000). Thus, they have been applied to the short-term electricity demand forecast (Bui et al., 2020). According to Hagan et al. (2014), recurrent neural networks are potentially more powerful than feedforward neural networks, but they present difficulties in training due to the vanishing gradient phenomenon, which leads to unsatisfactory forecasting results. Recurrent architectures based on gates have been used to solve this problem, such as the gated recurrent unit (GRU), which can explore essential long-term information in short-term electricity demand forecasting (Dudek, 2020; Gao et al., 2019; Kuan et al., 2017; Niu et al., 2016; Xiuyun et al., 2018).

The combination of a GRU flowing signals in a specific direction with another GRU carrying information in the opposite direction forms the bidirectional gated recurrent unit (BiGRU) (Luo et al., 2018). This process that generated the BiGRU network can provide more efficient short-term electric load forecasts than the original GRU (Lv et al., 2020). On the other hand, the convolutional neural network (CNN) can also be used to generate more efficient results of short-term electric load forecasts (Massaoudi et al., 2020a; Wu et al., 2021). This kind of network has been gaining notoriety for their results in the field related to pattern recognition, ranging from image processing to voice recognition (Albawi et al., 2017).

Different types of artificial neural network architectures can be used to create models with short-term energy demand forecasts closer to reality. Wu et al. (2020b) combined GRU and CNN networks to form the GRU-CNN forecasting model tested in a real-world experiment, in which its MAPE and RMSE were lower than those of the individual GRU and CNN networks, showing that the proposed hybrid model can use the temporal data more completely to obtain a more accurate short-term energy demand forecast. Sajjad et al. (2020) proposed the CNN-GRU model to be an effective alternative to other hybrid short-term energy demand forecasting models in terms of computational complexity and precision of results due to the representative resources of the extraction potential of the CNN network and efficient gate structure of the multilayer GRU network. Since then, other works have used similar approaches with different results (Xuan et al., 2021)

In addition to the hybrid model constituted by the layers of neural networks of GRU and CNN architecture, another hybrid model found in the literature is the one from the LSTM and CNN networks. Boubaker et al. (2021) used the CNN-LSTM and CNN-BiLSTM models to forecast the solar irradiation demand of a photovoltaic system, but their results were worse than the LSTM, BiLSTM, GRU and BiGRU networks when compared by the RMSE and MAPE metrics. On the other hand Massaoudi et al. (2020b) took advantage of the CNN-LSTM

hybrid neural network to perform the short-term electrical energy demand, where its results were superior to the BiGRU and BiLSTM networks regarding the RMSE, MAE and R^2 comparison metrics.

Given the possibility of using future energy demand values as a tool to support decision-making, this study aims to improve the short-term electricity demand forecast of a company in the electricity sector with the following proposed model based on different layers of artificial neural networks named BiGRU-CNN. The precision of short-term energy demand forecasting can affect the costs and revenues for electricity generators and transmission or distribution operators and, therefore, the profitability and sustainability of these organizations (Islam et al., 2019). Thus, the proposed BiGRU-CNN forecasting neural model of distinct layers was compared with the classical artificial neural networks MLP, CNN, RNN, GRU, and LSTM and the hybrid models GRU-CNN e CNN-BiGRU to verify if its results are more accurate. The historical series of the electric load of a company was used as input in the neural models to make short-term energy demand forecasts.

It is important to clarify that the BiGRU-CNN predictive model has already been used in scientific works in other areas of knowledge, such as: Electoral outcomes (Hadi et al., 2019); Chinese question classification (Liu et al., 2019); Aspect Based Opinion Mining (Sindhu et al., 2021a); Sentiment analysis (Sindhu et al., 2021b) and Multilingual named entity recognition (Ayifu et al., 2019). However, to date, there are no works related to short-term electricity demand forecast. Thus, this article seeks to verify whether this model has the capacity to be used in practical applications by companies in the Electric Energy sector in their decision-making.

2. Theoretical framework

2.1. Gated recurrent unit neural networks

A classical recurrent neural network has a memory function suitable for modeling sequential data, but these algorithms cannot deal with long-distance dependency problems due to gradient explosion and gradient disappearance phenomena (Li et al., 2020). The input gate, forget gate, and output gate, which constitute the LSTM neural network, widely used in the field of sequential forecasts, although with a very complex structure, are created to resolve these impasses (Xiuyun et al., 2018). Another disadvantage of this network is its training time, which is much longer than that of other algorithms. Thus, the gated recurrent unit (GRU) network, which is a special LSTM case, takes advantage of this aspect because it has fewer parameters due to the lack of output gate in its structure (Wang et al., 2018).

The GRU neural network is characterized by gate mechanisms that are especially suited to dealing with time-sequential tasks (Deng et al., 2019). These gate mechanisms are simplified in recurrent cells to significantly increase computational efficiencies in an attempt to maintain the same forecasting performance of the LSTM network (Lv et al., 2020). According to Li et al. (2020), the GRU artificial neural network has two control gates called reset gate and update gate, as shown in Figure 1. The first gate (reset gate) determines how much information needs to be forgotten from the hidden state of the previous instant of time. The information from the previous moment is ignored if its value is close to 0. On the other hand, the hidden information from the past time instant is retained in the current memory when the value is close to 1. The update gate, i.e., the second gate, is responsible for the amount of information in the hidden state of the previous time instant that will be brought to the current hidden state. In this case, the information of the hidden state of the previous instant will be ignored if its value is close to 0, but the information is retained in the current hidden state if the value is close to 1.

The GRU artificial neural network structural unit has two inputs at different time instants, being the current input vector x_t and the output vector h_{t-1} of the previous time instant, in which the output of each gate can be obtained through logical operations and non-linear input transformations (Wang et al., 2018). Equations 1, 2, 3 and 4, which control the functioning of the GRU neural network cell in Figure 1 (Li et al., 2020) are shown below, in which z_t is the update gate, r_t is the reset gate, \tilde{h}_t is the candidate hidden state of the current hidden node, h_t is the current hidden state, x_t is the current input of the artificial neural network, and h_{t-1} is the hidden state of the previous time instant. The activation function sigmoid is represented by σ , w represent the weights for each input x_t and u represent the weights for hidden state of the previous time instant h_{t-1} .

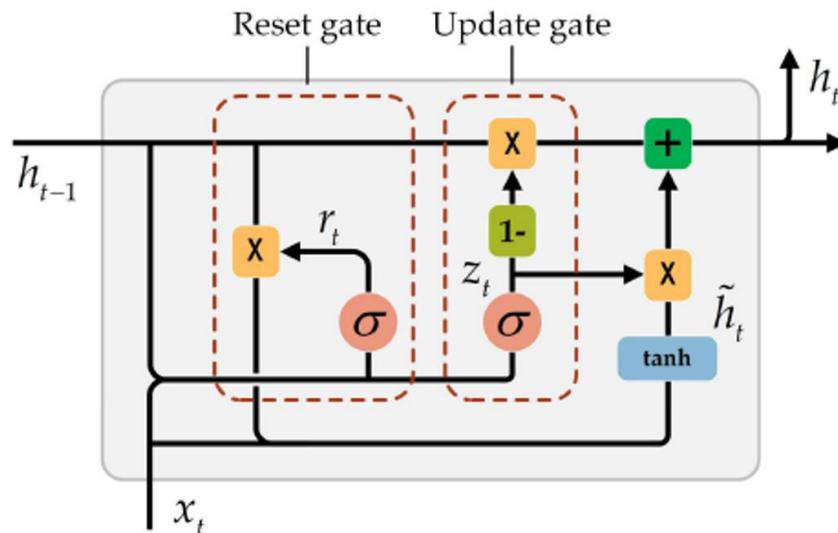


Figure 1. GRU artificial neural network structural unit.

Among these variables, the update gate z_t determines the integration between the information of a new input with the historical information and the reset gate r_t establishes the proportion of the information state in the model (Xiuyun et al., 2018).

$$z_t = \sigma(w_{zx}x_t + u_{zh}h_{t-1}) \quad (1)$$

$$r_t = \sigma(w_{rx}x_t + u_{rh}h_{t-1}) \quad (2)$$

$$\tilde{h}_t = \tanh(w_{hx}x_t + r_t \odot u_{hh}h_{t-1}) \quad (3)$$

$$h_t = (1 - z_t) \odot \tilde{h}_t + z_t \odot h_{t-1} \quad (4)$$

2.2. Bidirectional gated recurrent unit neural networks

The GRU neural network employs the recurrent structure to store and retrieve information for long periods, but its performance in practice may not be as satisfactory as in theory because the network only accesses past information (Deng et al., 2019). The bidirectional GRU (BiGRU) network has a future layer in which the data sequence is in the opposite direction to overcome this problem. Thus, this network uses two hidden layers to extract information from both the past and the future and both are connected in the same output layer (Luo et al., 2018). These characteristics enable the bidirectional structure to assist the recurrent neural networks to extract more information and, consequently, improve the performance of the learning process (Zhang et al., 2018).

Figure 2, taken from Li et al. (2020), shows a BiGRU neural network with two intermediate layers, in which the output layer overlays and normalizes the results of the forward and backward layers at each moment. Its Equations 5, 6, 7, 8 and 9) are shown below, in which \vec{h}_t^1 and \vec{h}_t^2 are the output vectors of forward layers of the first and second layers of the BiGRU artificial neural network at time t . On the other hand, the vectors $\overset{\leftarrow}{h}_t^1$ and $\overset{\leftarrow}{h}_t^2$ represent the outputs of the first and second backward layer of the network at the same instant of time t . f is the GRU neural network processing, g is the activation function and w, b are the weight and bias matrices respectively.

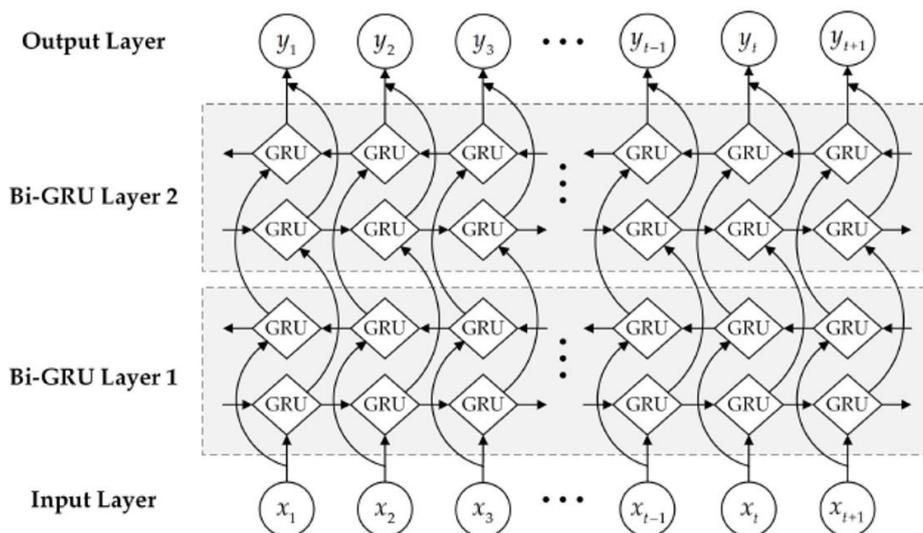


Figure 2. BiGRU neural network structural unit.

.Finally, y_t is the response of the network using past and future information.

$$\bar{h}_t^1 = f\left(w_{xh^1}x_t + w_{h^1h^1}\bar{h}_{t-1}^1 + b_{h^1}\right) \quad (5)$$

$$\bar{h}_t^1 = f\left(w_{xh^1}x_t + w_{h^1h^1}\bar{h}_{t+1}^1 + b_{h^1}\right) \quad (6)$$

$$\bar{h}_t^2 = f\left(w_{h^1h^2}\bar{h}_t^1 + w_{h^2h^2}\bar{h}_{t-1}^2 + b_{h^2}\right) \quad (7)$$

$$\bar{h}_t^2 = f\left(w_{h^1h^2}\bar{h}_t^1 + w_{h^2h^2}\bar{h}_{t+1}^2 + b_{h^2}\right) \quad (8)$$

$$y_t = g\left(w_{h^2y}\bar{h}_t^2 + w_{h^2y}\bar{h}_t^2 + b_y\right) \quad (9)$$

2.3. Convolutional neural networks

Convolutional neural network (CNN) is a type of deep artificial neural network often applied to deal with tasks in which data has high local correlations, such as visual images, video prediction, and text categorization, as this specific network can capture the same pattern located in different regions (Tian et al., 2018). Although the CNN network was specially designed to solve image classification problems, in which the network is fed by two-dimensional data, this algorithm is also applied in the field of time series analysis, in which one-dimensional data is used, as the concept of weight sharing is used to increase performance in solving non-linear problems, as seen in electric load forecasts (Sajjad et al., 2020). Basically, weight sharing applies invariance translations

in the neural network model to assist in filtering the learning resource regardless of the spatial properties (Albawi et al., 2017).

The CNN neural network is a feed-forward network, allowing encoding important information contained in the input data with far fewer parameters than in other deep learning models (Zhang et al., 2018). Its standard structure is formed by convolution layers, pooling layers, and, finally, fully connected layers (Tudose et al., 2020). Figure 3, taken from Wu et al. (2021), shows these layers organized generically to compose the CNN network, in which the convolution layer has the function of extracting effective resources from the input data through its multiple internal convolutional kernels, and the pooling layer, added after the convolution layer, can keep strong features and discard weak features to reduce complexity and avoid overfitting. The fully connected layer integrates all local resources to form a global resource used in the calculation of the final result (Tian et al., 2018).

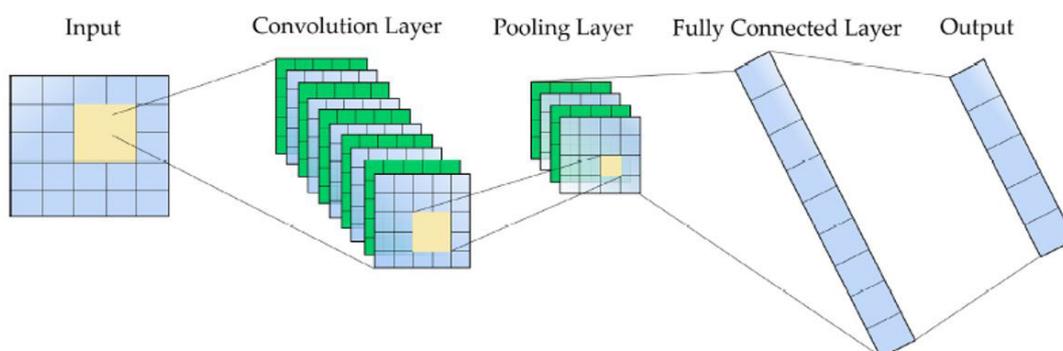


Figure 3. Convolutional neural network structure.

Figure 4 shows a convolutional process from a one-dimensional input (Tudose et al., 2020). This process is controlled by Equation 10, in which $*$ represents the convolutional operation, I corresponds to the one-dimensional input of the current layer, K denotes the one-dimensional Kernel, and S is the convolution output, also known as the feature map.

$$S(i) = (I * K)(i) = \sum_m I(i+m)K(m) \tag{10}$$

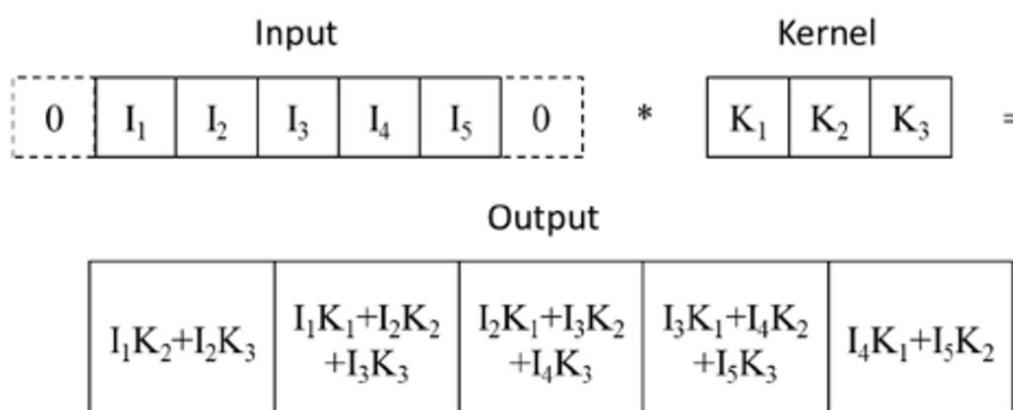


Figure 4. Convolutional process from a 1D input.

3. Proposed method

The BiGRU-CNN model proposed in this study to perform short-term electric load forecasts was built by a BiGRU layer followed by a CNN layer. The input data set referring to the energy demand historical series was manipulated to then feed the BiGRU layer, which performed its processing to extract long-term temporal dependencies. These time-dependent features, which are represented by two hidden state vectors with past and future information, were introduced into the CNN layer so that significant local relationships are captured through the convolution and pooling layers. After this procedure, the data set was structured in several dimensions that had to go through the flatten layer to become one-dimensional again and, therefore, be introduced in the fully connected layer to perform short-term electric load forecasts. The structure of the proposed model mentioned above is shown in Figure 5.

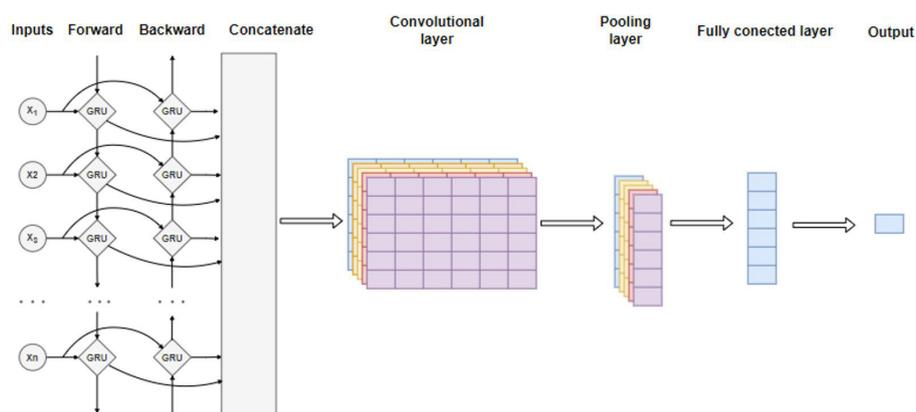


Figure 5. Flowchart of the proposed BiGRU-CNN neural network.

3.1. Time series

A time series is any set of data observed in an orderly manner over time (Morettin & Tolo, 2006). The electric load time series used in this study belongs to the Duke Energy company and is composed of 44,553 hourly observations performed from 1:00 pm on October 1, 2012, to 1:00 am on October 11, 2017. Figure 6 shows the energy demand data over time, in which the ordinate axis is given in megawatts (MW) and the abscissa axis is each time observation given in hours. The file in csv (comma-separated values) format that contained the data from the time series was obtained from <https://www.kaggle.com/robikscube/hourly-energy-consumption>.

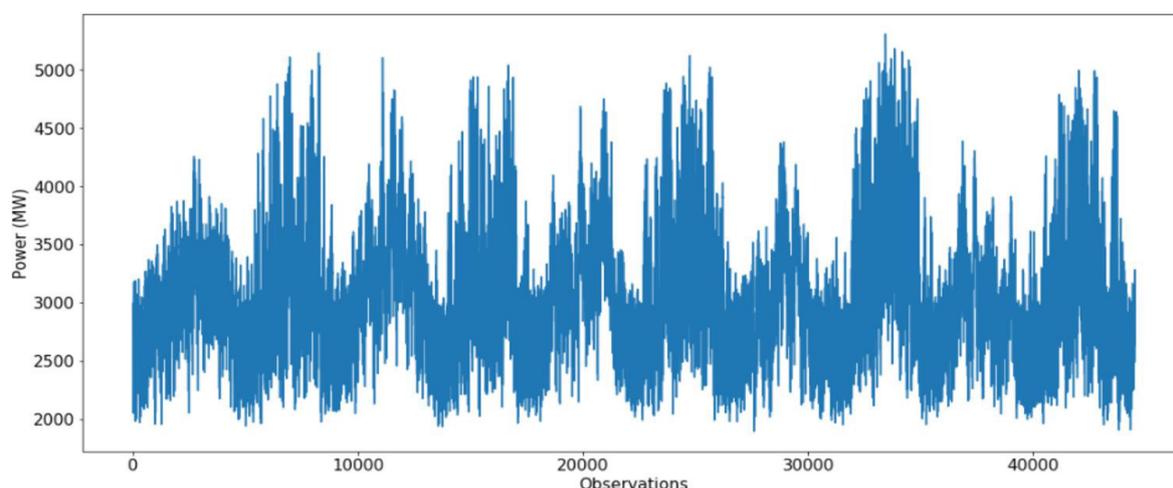


Figure 6. Time series of energy demand.

3.2. Short-term electric load forecast

The proposed BiGRU-CNN model and the well-known classical networks regarding short-term electric load forecast MLP, RNN, GRU, and LSTM were used to forecast in a 24-hour forecasting horizon. Each forecast was performed at an interval of one hour. Thus, 24 forecasts were needed to obtain the forecasting horizon. The future values of short-term electric load requirements were obtained as follows: only one simulation was carried out to train and validate each of the different forecasting models and these models performed multi-step forecasts recursively after being trained and validated. In this type of forecast, future values are fed back into artificial neural networks as if they were observations of the training or validation sample, thus avoiding that the model is trained for each new forecast. In other words, the recursive forecasting model is trained and validated only once to perform 24 effective forecasts, which generates considerable time savings due to the time required to train the networks.

The electric load data were divided into three distinct data set after the time series was transformed into a supervised machine learning problem. The first data set consisted of the training data set, formed by the first 36,199 data; the second data set was that of validation, consisting of 8,330 data after the training data set; and, finally, the test data set was built only by the 24 remaining data.

The training data set aimed to learn the artificial neural networks of the patterns of electric load time series. The internal parameters of the neural networks found during training were tested in the validation data set to verify the forecasting capacity of the network in the data not seen in the previous step. The test data set would be used to evaluate the effective electric load forecasts if the network performance were similar in the training and validation data sets. Effective forecasts are those used for practical purposes, while forecasts made in the training and validation data sets serve to ascertain whether the parameters of the neural networks found during training are capable of generalizing results. The accuracy measures MAPE (mean absolute percentage error), MAE (mean absolute error), and RMSE (root mean square error) presented by Equations 11, 12 and 13, respectively, were used to evaluate the forecasts provided by all artificial neural network models in training, validation, and testing data sets, where x is the desired value, y is the forecasted value, and n is the number of elements in the sample.

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{x_i - y_i}{y_i} \right| \quad (11)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_i - y_i| \quad (12)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2} \quad (13)$$

3.3. Network code

The algorithm was implemented on Python inside the *Google Colab* environment. The `MinMaxScaler` function of the `sklearn.preprocessing` package was used to normalize the data before being introduced into the neural networks after the temporal data were obtained in the program. Normalization was necessary because energy demand data have a high variation, which could affect the algorithm performance during training, thus providing results not consistent with reality. According to Upadhaya et al. (2019), preliminary data processing can generate better forecasting results related to short-term energy demand. Equation 14 shows how data normalization was performed by the `MinMaxScaler` function.

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (14)$$

The time series was transformed into a supervised machine learning problem after pre-processing the historical data, that is, a sequence of input and output pairs was created so that a decision could be made and then compared with the desired output. The internal parameters of artificial neural networks are modified during training by the *Adam* (adaptive moment estimation) algorithm to allow the difference between the network response and the desired output for a given set of inputs to be minimal. The *Adam* training algorithm was chosen considering the study by Kingma & Ba (2015) where it was considered superior to other algorithms. The ten neurons in each of the two middle layers of all forecasting models have the rectified linear unit (ReLU) as their activation function (Equation 15). The reason for this choice is related to the ability to improve the forecasting performance of recurrent neural networks, according to Talathi & Vartak (2015). Regarding the CNN architecture layers of the neural networks, the kernel_size quantity was set at 6 and the number of filters was set at 8. The training of all artificial neural networks was performed in 150 epochs with a batch size of 32.

$$f(x) = \max(0, x) \quad (15)$$

4. Experiments

Analyzing the results in Table 1 from neural networks of classical MLP, CNN, RNN, GRU and LSTM architectures with regard to short-term electricity demand forecasts, it is evident that the CNN model showed the worst forecasting performance over a 24-hour forecast horizon, this was expected because the CNN network is not able to obtain relevant information from the extraction of nonlinear relations between adjacent samples in local regions.

Table 1. MAPE, MAE and RMSE error for all data.

Model	MAPE	MAE	RMSE
MLP	4.52%	134.36	171.08
CNN	5.58%	165.72	228.16
RNN	3.57%	106.05	138.51
GRU	4.10%	121.50	153.36
LSTM	4.10%	122.04	158.05
GRU-CNN	14.51%	428.46	625.36
CNN-BiGRU	5.10%	151.64	196.09
BiGRU-CNN	3.42%	100.75	122.22

The MLP model showed the second worst forecasting performance over a 24-hour forecast horizon, considering the mean of the lowest MAPE, MAE, and RMSE. It can be explained by the fact that this network is not able to extract the temporal dynamic behavior of the energy demand data because its structure does not have information feedback devices. Following are the LSTM and GRU models, which had very similar performances, mainly in the mean of MAPE and MAE, because their structures have analogous gate mechanisms that work as long-term memory to store essential features responsible for the generating process of the time series. The ability of these two networks to learn long-term temporal patterns was not enough to provide the best short-term energy demand forecast results, which leads to the hypothesis that essential information for this purpose is contained in more recent time features. The formulated hypothesis gains more strength when the performance of the RNN network was superior to the LSTM and GRU networks, as the former does not use the mechanisms that constitute long-term memory and, therefore, very distant temporal information is forgotten during training due to the vanishing gradient phenomenon. Therefore, forecasts from the RNN network are based on recent temporal information and its mean was responsible for placing the model in first place.

In an attempt to increase the efficiency, a GRU architecture network layer followed by a CNN architecture network layer was used to compose the GRU-CNN model, however this model provided predictive results worse than those of the individual networks.

Adding a bidirectional GRU layer at the output of a CNN to form the CNN-BiGRU hybrid model was enough to improve the efficiency of the predictive results when compared with the individual CNN model. However, the short-term energy demand forecasts of the CNN-BiGRU model were not superior to those of classical MLP, RNN, GRU, LSTM architecture networks when compared by MAPE, MAE and RMSE errors. The proposed BiGRU-CNN model had the best forecasting performance, considering the simple mean of the three accuracy metrics when compared to the others.

Therefore feeding a CNN layer to extract local trends and then introducing them into a BiGRU layer so that past and future long-term temporal correlations can be obtained was not as satisfactory as the BiGRU-CNN model. The priority of choosing which layer will be the first architecture of a hybrid neural network model is fundamental for the impact on predictive results.

The analysis of the means of the MAPE, MAE, and RMSE shows that the forecasting performance of the RNN and BiGRU-CNN networks was close, especially in the MAPE error. Thus, more computer simulations were performed to verify whether this behavior is perpetuated and if the BiGRU-CNN network is superior to other networks in its capacity to forecast future values of short-term energy demand. The new simulations are divided into two scenarios using the same conditions and hyperparameters in the networks, differing only regarding the time series size. The first and second scenarios have 77.5 and 66.3% of the original energy demand historical series, respectively (Tables 2 and 3).

Table 2. MAPE, MAE and RMSE error for 77.5% data.

Model	MAPE	MAE	RMSE
MLP	4.69%	140.98	171.36
CNN	14.19%	418.35	533.40
RNN	7.16%	214.51	244.32
GRU	8.22%	246.16	276.39
LSTM	7.02%	209.32	249.83
GRU-CNN	14.63%	430.55	592.70
CNN-BiGRU	9.99%	294.19	377.78
BiGRU-CNN	3.97%	118.95	149.25

Table 3. MAPE, MAE and RMSE error for 66.3% data.

Model	MAPE	MAE	RMSE
MLP	9.65%	284.70	342.53
CNN	6.23%	188.05	234.30
RNN	11.24%	333.21	397.03
GRU	7.73%	228.78	283.18
LSTM	7.75%	231.02	287.56
GRU-CNN	14.18%	420.82	562.35
CNN-BiGRU	8.31%	249.59	304.16
BiGRU-CNN	5.83%	173.23	234.04

The analysis of the tables with simulations involving 77.5 and 66.3% of the original historical series shows that the same relationship between the RNN and BiGRU-CNN networks is not maintained, because in the first one the results of MLP were closer to the BiGRU-CNN network and in the second one the performance of the CNN network was the least different. These results show that the most relevant information to generate more efficient short-term electricity demand forecasts are contained in the short-term temporality when 100% of the data is used.

Therefore, the proposed BiGRU-CNN model showed superiority in these three scenarios relative to the MLP, RNN, GRU, LSTM, GRU-CNN and CNN-BiGRU networks. It is important to highlight that the introduction of the bidirectional mechanism in the GRU-CNN network raised its position from the worst model to the best in the 3 available scenarios. Also, most of the errors increased with a reduction in the number of observations related to the electric load feed to the networks. This event is not unusual, as the lesser the data, the lesser the temporal information to be modeled by the networks during training. Thus, the network parameters found in the training are not able to reliably represent the generating process of the electric load time series and, consequently, their forecasts will not be consistent with reality. The aforementioned processes are better understood with the visualization of Figure 7.

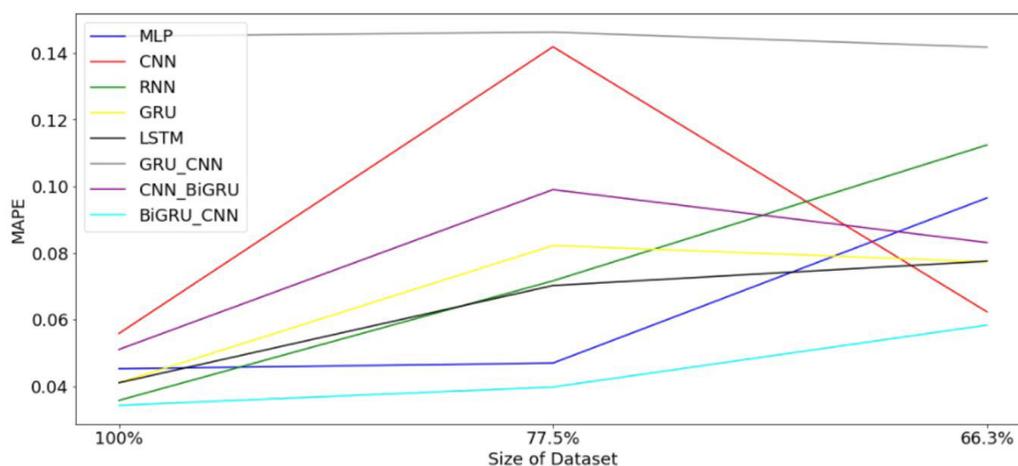


Figure 7. MAPE evolution for all models.

5. Conclusion

This study proposed the BiGRU-CNN model to for short-term electric load forecasts to assist companies in the energy sector in their decision-making. The experimental results showed that feeding a BiGRU layer with the time series to extract its long-term temporal correlations and then introducing these time features into a CNN layer so that local trends can be captured proved to be efficient when compared by the MAPE, MAE, and RMSE errors with the MLP, RNN, GRU, LSTM, GRU-CNN and CNN-BiGRU networks. Further studies need to be conducted by changing the hyperparameters of neural networks and the time series that feed them to ensure that the proposed BiGRU-CNN model can be superior to other forecasting models in terms of short-term electricity demand forecasting.

References

- Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, Aug. 21-23). Understanding of a convolutional neural network. In *International Conference on Engineering and Technology - ICET* (pp. 1-6). Antalya, Turkey: IEEE.
- Alberg, D., & Last, M. (2018). Short-term load forecasting in smart meters with sliding window-based ARIMA algorithms. *Vietnam Journal of Computer Science*, 5(3-4), 241-249. <http://dx.doi.org/10.1007/s40595-018-0119-7>.
- Amin, M. A. A., & Hoque, M. A. (2019, March 13-15). Comparison of ARIMA and SVM for short-term load forecasting. In S. Chakrabarti, & A. Mukherjee (Eds.), *9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference - IEMECON* (pp. 205-210). Jaipur, India: IEEE.
- Ayifu, M., Wushouer, S., & Palidan, M. (2019). Multilingual named entity recognition based on the BiGRU-CNN-CRF hybrid model. *International Journal of Information and Communication Technology*, 15(3), 223-242. <http://dx.doi.org/10.1504/IJICT.2019.102996>.
- Boubaker, S., Benghanem, M., Mellit, A., Lefza, A., Kahouli, O., & Kolsi, L. (2021). Deep neural networks for predicting solar radiation at Hail Region, Saudi Arabia. *IEEE Access: Practical Innovations, Open Solutions*, 9, 36719-36729. <http://dx.doi.org/10.1109/ACCESS.2021.3062205>.
- Bui, V., Nguyen, V. H., Pham, T. L., Kim, J., & Jang, Y. M. (2020, Feb. 19-21). RNN-based deep learning for one-hour ahead load forecasting. In *International Conference on Artificial Intelligence in Information and Communication, ICAIIC 2020* (pp. 587-589). Fukuoka, Japan: IEEE <http://dx.doi.org/10.1109/ICAIIIC48513.2020.9065071>
- Carpinteiro, O. A. S., & Silva, A. P. A. (2000, Nov. 25). A hierarchical neural model in short-term load forecasting. In C. H. C. Ribeiro, & F. M. G. França (Eds.), *Proceedings of Sixth Brazilian Symposium on Neural Networks: Vol. 1* (pp. 120-124). Rio de Janeiro, Brazil: IEEE.

- Cerne, G., Dovzan, D., & Skrjanc, I. (2018). Short-term load forecasting by separating daily profiles and using a single fuzzy model across the entire domain. *IEEE Transactions on Industrial Electronics*, *65*(9), 7406-7415. <http://dx.doi.org/10.1109/TIE.2018.2795555>.
- Chandramitasari, W., Kurniawan, B., & Fujimura, S. (2018, Aug. 29-30). Building deep neural network model for short term electricity consumption forecasting. In A. Pranolo, A. Prahara, A. Azhari, & A. Aktawan (Eds.), *International Symposium on Advanced Intelligent Informatics: Revolutionize Intelligent Informatics Spectrum for Humanity - SAIN* (pp. 43-48). Yogyakarta, Indonesia: IEEE.
- Chapagain, K., Kittipiyakul, S., & Kulthanavit, P. (2020). Short-term electricity demand forecasting: impact analysis of temperature for Thailand. *Energies*, *13*(10), 1-29. <http://dx.doi.org/10.3390/en13102498>.
- Charytoniuk, W., & Chen, M. S. (2000). Very short-term load forecasting using artificial. *IEEE Transactions on Power Systems*, *15*(1), 263-268. <http://dx.doi.org/10.1109/59.852131>.
- Chen, Y., Xu, P., Chu, Y., Li, W., Wu, Y., Ni, L., Bao, Y., & Wang, K. (2017). Short-term electrical load forecasting using the Support Vector Regression (SVR) model to calculate the demand response baseline for office buildings. *Applied Energy*, *195*, 659-670. <http://dx.doi.org/10.1016/j.apenergy.2017.03.034>.
- Coelho, V. N., Coelho, I. M., Coelho, B. N., Reis, A. J. R., Enayatifar, R., Souza, M. J. F., & Guimarães, F. G. (2016). A self-adaptive evolutionary fuzzy model for load forecasting problems on smart grid environment. *Applied Energy*, *169*, 567-584. <http://dx.doi.org/10.1016/j.apenergy.2016.02.045>.
- Deng, Y., Jia, H., Li, P., Tong, X., Qiu, X., & Li, F. (2019, June 19-21). A deep learning methodology based on bidirectional gated recurrent unit for wind power prediction. In *Proceedings of the 14th IEEE Conference on Industrial Electronics and Applications - ICIEA 2019* (pp. 591-595). Xi'an, China: IEEE. <http://dx.doi.org/10.1109/ICIEA.2019.8834205>
- Dhaval, B., & Deshpande, A. (2020). Short-term load forecasting with using multiple linear regression. *Iranian Journal of Electrical and Computer Engineering*, *10*(4), 3911-3917. <http://dx.doi.org/10.11591/ijece.v10i4.pp3911-3917>.
- Dudek, G. (2016). Pattern-based local linear regression models for short-term load forecasting. *Electric Power Systems Research*, *130*, 139-147. <http://dx.doi.org/10.1016/j.epsr.2015.09.001>.
- Dudek, G. (2020). Multilayer perceptron for short-term load forecasting: from global to local approach. *Neural Computing & Applications*, *32*(8), 3695-3707. <http://dx.doi.org/10.1007/s00521-019-04130-y>.
- Fallah, S. N., Ganjkhani, M., Shamshirband, S., & Chau, K. (2019). Computational intelligence on short-term load forecasting: a methodological overview. *Energies*, *12*(3), 393. <http://dx.doi.org/10.3390/en12030393>.
- Gao, X., Li, X., Zhao, B., Ji, W., Jing, X., & He, Y. (2019). Short-term electricity load forecasting model based on EMD-GRU with feature selection. *Energies*, *12*(6), 1-18. <http://dx.doi.org/10.3390/en12061140>.
- Ghalekhondabi, I., Ardjmand, E., Weckman, G. R., & Young, W. A. (2017). An overview of energy demand forecasting methods published in 2005-2015. *Energy Systems*, *8*, 411-447. <http://dx.doi.org/10.1007/s12667-016-0203-y>.
- Hadi, K. A., Lasri, R., & Abderrahmani, A. E. (2019). Social data analytics for forecasting electoral outcomes. *International Journal of Innovative Technology and Exploring Engineering*, *8*(8), 2468-2471.
- Hagan, M. T., Demuth, H. B., & Beale, M. H. (2014). *Neural network design* (2nd ed.). Oklahoma: OSU.
- Hahn, H., Meyer-Nieberg, S., & Pickl, S. (2009). Electric load forecasting methods: tools for decision making. *European Journal of Operational Research*, *199*(3), 902-907. <http://dx.doi.org/10.1016/j.ejor.2009.01.062>.
- Huang, C., & Yang, H. (1995, Nov. 21-23). A time series approach to short term load forecasting through evolutionary programming structures. In *Proceedings of the International Conference on Energy Management and Power Delivery - EMPD* (Vol. 2, pp. 583-588). Singapore: IEEE.
- Islam, M. A., Che, H. S., Hasanuzzaman, M., & Rahim, N. A. (2019). Energy demand forecasting. In M. Hasanuzzaman & N. A. Rahim (Eds.), *Energy for sustainable development: demand, supply, conversion and management*. London: Academic Press/Elsevier.
- Jiang, H., Zhang, Y., Muljadi, E., Zhang, J. J., & Gao, D. W. (2018). A short-term and high-resolution distribution system load forecasting approach using support vector regression with hybrid parameters optimization. *IEEE Transactions on Smart Grid*, *9*(4), 3331-3350. <http://dx.doi.org/10.1109/TSG.2016.2628061>.
- Johannesen, N. J., Kolhe, M., & Goodwin, M. (2019). Relative evaluation of regression tools for urban area electrical energy demand forecasting. *Journal of Cleaner Production*, *218*, 555-564. <http://dx.doi.org/10.1016/j.jclepro.2019.01.108>.
- Kandil, M. S., El-Debeiky, S. M., & Hasani, N. E. (2002). Long-term load forecasting for fast developing utility using a knowledge-based expert system. *IEEE Transactions on Power Systems*, *17*(2), 491-496. <http://dx.doi.org/10.1109/TPWRS.2002.1007923>.
- Kingma, D. P., & Ba, J. L. (2015, May. 7-9). Adam: a method for stochastic optimization. In Y. Bengio & Y. LeCun (Eds.), *3rd International Conference on Learning Representations - ICLR 2015 - Conference Track Proceedings* (pp. 1-15). San Diego: OpenReview.net.
- Kuan, L., Yan, Z., Xin, W., Yan, C., Xiangkun, P., Wenxue, S., Zhe, J., Yong, Z., Nan, X., & Xin, Z. (2017, Nov. 26-28). Short-term electricity load forecasting method based on multilayered self-normalizing GRU network. In F. Gao (Ed.), *IEEE Conference on Energy Internet and Energy System Integration - EI2* (pp. 1-5). Beijing, China: IEEE. <http://dx.doi.org/10.1109/EI2.2017.8245330>
- Li, P., Luo, A., Liu, J., Wang, Y., Zhu, J., Deng, Y., & Zhang, J. (2020). Bidirectional gated recurrent unit neural network for Chinese address element segmentation. *ISPRS International Journal of Geo-Information*, *9*(11), 635. <http://dx.doi.org/10.3390/ijgi9110635>.
- Li, Y., Che, J., & Yang, Y. (2018). Subsampled support vector regression ensemble for short term electric load forecasting. *Energy*, *164*, 160-170. <http://dx.doi.org/10.1016/j.energy.2018.08.169>.
- Liu, J., Yang, Y., Lv, S., Wang, J., & Chen, H. (2019). Attention-based BiGRU-CNN for Chinese question classification. *Journal of Ambient Intelligence and Humanized Computing*, *10*(13), 1-12. <https://doi.org/10.1007/s12652-019-01344-9>
- Luo, X., Zhou, W., Wang, W., Zhu, Y., & Deng, J. (2018). Attention-based relation extraction with bidirectional gated recurrent unit and highway network in the analysis of geological data. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 5705-5715. <http://dx.doi.org/10.1109/ACCESS.2017.2785229>.
- Lv, P., Liu, S., Yu, W., Zheng, S., & Lv, J. (2020). EGA-STLF: a hybrid short-term load forecasting model. *IEEE Access: Practical Innovations, Open Solutions*, *8*, 31742-31752. <http://dx.doi.org/10.1109/ACCESS.2020.2973350>.
- Marković, M. L., & Fraissler, W. F. (1993). Short-term load forecast by plausibility checking of announced demand: An expert system approach. *European Transactions on Electrical Power*, *3*(5), 353-358. <http://dx.doi.org/10.1002/etep.4450030506>.

- Massaoudi, M., Refaat, S. S., Abu-Rub, H., Chihi, I., & Oueslati, F. S. (2020a). PLS-CNN-BiLSTM: an end-to-end algorithm-based savitzky-golay smoothing and evolution strategy for load forecasting. *Energies*, *13*(20), 1-29. <http://dx.doi.org/10.3390/en13205464>.
- Massaoudi, M., Refaat, S. S., Chihi, I., Trabelsi, M., Abu-Rub, H., & Oueslati, F. S. (2020b). Short-term electric load forecasting based on data-driven deep learning techniques. In *IECON - The 46th Annual Conference of the IEEE Industrial Electronics Society* (pp. 2565-2570). Singapore: IEEE.
- Massaoudi, M., Refaat, S. S., Chihi, I., Trabelsi, M., Oueslati, F. S., & Abu-Rub, H. (2021). A novel stacked generalization ensemble-based hybrid LGBM-XGB-MLP model for short-term load forecasting. *Energy*, *214*, 118874. <http://dx.doi.org/10.1016/j.energy.2020.118874>.
- Mayrink, V., & Hippert, H. S. (2016). A hybrid method using exponential smoothing and gradient boosting for electrical short-term load forecasting. In C. Rodríguez, & J. B. Gómez (Eds.), *IEEE Latin American Conference on Computational Intelligence - LA-CCI*. Cartagena, Colombia: IEEE.
- Medsker, L. R., & Jain, L. C. (2000). *Recurrent neural networks: design and applications*. Boca Raton: CRC Press.
- Mohammed, J., Bahadoorsingh, S., Ramsamooj, N., & Sharma, C. (2017, June 18-22). Performance of exponential smoothing, a neural network and a hybrid algorithm to the short term load forecasting of batch and continuous loads. In *IEEE Manchester PowerTech*. Manchester, UK: IEEE.
- Morettin, P. A., & Toloi, C. M. C. (2006). *Análise de séries temporais* (2. ed.). São Paulo, Brazil: Blucher.
- Mukhopadhyay, P., Mitra, G., Banerjee, S., & Mukherjee, G. (2018, Dec. 21-23). Electricity load forecasting using fuzzy logic: Short term load forecasting factoring weather parameter. In *7th International Conference on Power Systems - ICPS* (pp. 812-819). Pune, India: IEEE.
- Niu, M., Sun, S., Wu, J., Yu, L., & Wang, J. (2016). An innovative integrated model using the singular spectrum analysis and nonlinear multi-layer perceptron network optimized by hybrid intelligent algorithm for short-term load forecasting. *Applied Mathematical Modelling*, *40*(5-6), 4079-4093. <http://dx.doi.org/10.1016/j.apm.2015.11.030>.
- Pan, X., & Lee, B. (2012). A comparison of support vector machines and artificial neural networks for mid-term load forecasting. In *IEEE International Conference on Industrial Technology, ICIT* (pp. 95-101). Athens, Greece: IEEE.
- Rahman, S., & Hazim, O. (1996). Load forecasting for multiple sites: development of an expert system-based technique. *Electric Power Systems Research*, *39*(3), 161-169. [http://dx.doi.org/10.1016/S0378-7796\(96\)01114-5](http://dx.doi.org/10.1016/S0378-7796(96)01114-5).
- Rendon-Sanchez, J. F., & Menezes, L. M. (2019). Structural combination of seasonal exponential smoothing forecasts applied to load forecasting. *European Journal of Operational Research*, *275*(3), 916-924. <http://dx.doi.org/10.1016/j.ejor.2018.12.013>.
- Saber, A. Y., & Alam, A. K. M. R. (2018). Short term load forecasting using multiple linear regression for big data. In *IEEE Symposium Series on Computational Intelligence - SSCI* (pp. 1-6). Honolulu, HI, USA: IEEE.
- Sajjad, M., Khan, Z. A., Ullah, A., Hussain, T., Ullah, W., Lee, M. Y., & Baik, S. W. (2020). A novel CNN-GRU-based hybrid approach for short-term residential load forecasting. *IEEE Access: Practical Innovations, Open Solutions*, *8*, 143759-143768. <http://dx.doi.org/10.1109/ACCESS.2020.3009537>.
- Setiawan, A., Koprinska, I., & Agelidis, V. G. (2009). Very short-term electricity load demand forecasting using support vector regression. In *International Joint Conference on Neural Networks* (pp. 2888-2894). Atlanta, GA, USA: IEEE. <http://dx.doi.org/10.1109/IJCNN.2009.5179063>
- Shahidehpour, M., Yamin, H., & Li, Z. (2002). *Market operations in electric power systems: forecasting, scheduling, and risk management* (1st ed.). Hoboken: Wiley. <http://dx.doi.org/10.1002/047122412X>.
- Sindhu, C., Som, B., & Singh, S. P. (2021a). Aspect based opinion mining leveraging weighted bigru and CNN module in parallel. In *International Conference on Intelligent Technologies - CONIT* (pp. 1-7). Hubli, India: IEEE. <http://dx.doi.org/10.1109/CONIT51480.2021.9498441>.
- Sindhu, C., Som, B., & Singh, S. P. (2021b). Aspect-oriented sentiment classification using BiGRU-CNN model. In *5th International Conference on Computing Methodologies and Communication - ICCMC* (pp. 984-989). Erode, India: IEEE..
- Singh, A. K., & Khatoon, S. (2013). An overview of electricity demand forecasting techniques. *National Conference on Emerging Trends in Electrical, Instrumentation & Communications Engineer*, *3*(3), 38-48.
- Soliman, S. A., & Al-Kandari, A. M. (2010). *Electrical load forecasting: modeling and model construction* (1st ed.). Oxford: Butterworth-Heinemann.
- Talathi, S. S., & Vartak, A. (2015). Improving performance of recurrent neural network with relu nonlinearity. *Neural and Evolutionary Computing*, *1*, ArXiv:1511.03771. Retrieved in 2021 November 04, from <http://arxiv.org/abs/1511.03771>
- Tian, C., Ma, J., Zhang, C., & Zhan, P. (2018). A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network. *Energies*, *11*(12), 3493. <http://dx.doi.org/10.3390/en11123493>.
- Tudose, A. M., Sidea, D. O., Picioara, I. I., Boicea, V. A., & Bulac, C. (2020). A CNN based model for short-term load forecasting: a real case study on the Romanian power system. In *55th International Universities Power Engineering Conference - UPEC*. Turin, Italy: IEEE.
- Upadhaya, D., Thakur, R., & Singh, N. K. (2019). A systematic review on the methods of short term load forecasting. In *2nd International Conference on Power Energy Environment and Intelligent Control - PEEIC* (pp. 6-11). Greater Noida, India: IEEE.
- Wang, Y., Liao, W., & Chang, Y. (2018). Gated recurrent unit network-based short-term photovoltaic forecasting. *Energies*, *11*(8), 2163. <https://doi.org/10.3390/en11082163>.
- Wu, F., Cattani, C., Song, W., & Zio, E. (2020a). Fractional ARIMA with an improved cuckoo search optimization for the efficient short-term power load forecasting. *Alexandria Engineering Journal*, *59*(5), 3111-3118. <http://dx.doi.org/10.1016/j.aej.2020.06.049>.
- Wu, K., Wu, J., Feng, L., Yang, B., Liang, R., Yang, S., & Zhao, R. (2021). An attention-based CNN-LSTM-BiLSTM model for short-term electric load forecasting in integrated energy system. *International Transactions on Electrical Energy Systems*, *31*(1), 1-15. <http://dx.doi.org/10.1002/2050-7038.12637>.
- Wu, L., Kong, C., Hao, X., & Chen, W. (2020b). A short-term load forecasting method based on GRU-CNN hybrid neural network model. *Mathematical Problems in Engineering*, *2020*, 1-10. <http://dx.doi.org/10.1155/2020/1428104>.

- Xiuyun, G., Ying, W., Yang, G., Chengzhi, S., Wen, X., & Yimiao, Y. (2018). Short-term load forecasting model of gru network based on deep learning framework. In *2nd IEEE Conference on Energy Internet and Energy System Integration - EI2* (pp. 1–4). Beijing, China: IEEE <http://dx.doi.org/10.1109/EI2.2018.8582419>
- Xuan, Y., Si, W., Zhu, J., Sun, Z., Zhao, J., Xu, M., & Xu, S. (2021). Multi-model fusion short-term load forecasting based on random forest feature selection and hybrid neural network. *IEEE Access : Practical Innovations, Open Solutions*, *9*, 69002–69009. <http://dx.doi.org/10.1109/ACCESS.2021.3051337>.
- Yan, K., Li, W., Ji, Z., Qi, M., & Du, Y. (2019). A hybrid LSTM neural network for energy consumption forecasting of individual households. *IEEE Access : Practical Innovations, Open Solutions*, *7*, 157633–157642. <http://dx.doi.org/10.1109/ACCESS.2019.2949065>.
- Yang, H., Huang, C., & Huang, C. (1996). Identification of ARMAX model for short term load forecasting: an evolutionary programming approach. *IEEE Transactions on Power Systems*, *11*(1), 403–408. <http://dx.doi.org/10.1109/59.486125>.
- Zhang, D., Tian, L., Hong, M., Han, F., Ren, Y., & Chen, Y. (2018). Combining convolution neural network and bidirectional gated recurrent unit for sentence semantic classification. *IEEE Access: Practical Innovations, Open Solutions*, *6*(8), 73750–73759. <http://dx.doi.org/10.1109/ACCESS.2018.2882878>.

Apêndice D

Artigo Submetido à Journal of Control
Automation and Electrical Systems

1 Rede Neural BiGRU-CNN Aplicada à 2 Detecção de Furto de Energia Elétrica

3 **Resumo:** O presente trabalho analisou a
4 viabilidade da utilização da rede neural artificial
5 BiGRU-CNN, baseada em camadas de diferentes
6 arquiteturas das redes bidirecional gated
7 recurrent unit e convolutional neural network,
8 como ferramenta de detecção de furtos de energia
9 elétrica. A utilização desse tipo de ferramenta
10 com este modelo de classificação pode servir
11 como instrumento de apoio à tomada de decisão
12 de empresas do setor energético, através do
13 apontamento de unidades consumidoras
14 suspeitas de fraudes para posteriormente, serem
15 realizadas inspeções manuais. A rede neural
16 artificial proposta foi programada em python,
17 fazendo uso do pacote *keras*. O melhor modelo
18 de detecção foi o da rede BiGRU-CNN quando
19 comparado com as redes multilayer perceptron,
20 recurrent neural network, gated recurrent unit e
21 long short term memory, onde os valores das
22 métricas atribuídas as suas classificações foram
23 de 0,929 para accuracy; 0,885 para precision;
24 0,801 para recall; 0,841 para F1-Score e, por
25 último, 0,966 para Area Under the Receiver
26 Operating Characteristic Curve.

27
28 **Palavras-chave:** Inteligência Artificial,
29 Aprendizado de Máquina, Redes Neurais
30 Recorrentes, Séries Temporais.

31 32 1. Introdução:

33 O progresso econômico dos países em
34 desenvolvimento está diretamente relacionado a
35 utilização de energia elétrica em indústrias de
36 produção manufatureiras. Logo, a falta desse
37 recurso essencial impacta de maneira relevante
38 na economia em geral (Sarwar et al., 2017).

39 São diversas as razões que causam a não
40 disponibilidade adequada de energia elétrica e as
41 causas são classificadas como perdas técnicas e
42 não técnicas (Glauner et al., 2017). As perdas
43 técnicas são aquelas que surgem de forma natural
44 devido à irradiação e como resultado da
45 dissipação inevitável de energia elétrica em sua
46 transmissão e distribuição, envolvendo perdas
47 em dielétricos e sobretudo nos condutores
48 elétricos pelo efeito Joule (Viegas et al., 2017).
49 Já as perdas não técnicas são definidas como
50 qualquer energia consumida ou serviço que não
51 seja cobrado devido à falha de um equipamento
52 de medição ou à sua manipulação fraudulenta.
53 Essas perdas são causadas por avaria ou
54 manipulação ilegal nas instalações do
55 consumidor final e são muito difíceis de prever
56 (Guerrero et al., 2018).

57 Nas empresas de fornecimento de
58 eletricidade, o uso ilegal de energia elétrica
59 mediante fraude por parte do cliente é o principal

60 motivo relacionado às perdas não técnicas
61 (Ahmad, 2017). Esse problema, de longa data,
62 tem sido uma das principais preocupações na
63 gestão do sistema de energia de empresas do
64 setor, dado que, pode afetar o equilíbrio entre
65 demanda e oferta ocasionando problemas de
66 regulação da rede de energia e,
67 conseqüentemente, redução drástica no lucro
68 (Ouyang et al., 2017). Neste sentido, a detecção
69 dos casos de furto nas redes de energia elétrica é
70 fundamental para evitar perdas econômicas e
71 mitigar riscos de segurança. Porém, os métodos
72 convencionais dependem principalmente da
73 verificação humana ou equipamentos específicos
74 para medição que são extremamente lentos, caros
75 e ineficientes (Z. Chen et al., 2020). Para superar
76 esses obstáculos existem na literatura, uma
77 quantidade considerável de técnicas de
78 modelagem para detectar fraudes no consumo de
79 energia elétrica (Ahmad et al., 2017).

80 Diversas são as técnicas de detecção de
81 fraudes nas redes de energia elétrica, onde as
82 técnicas de detecção baseadas em classificação
83 são uma das abordagens mais utilizadas. Esse
84 tipo de técnica tem como ideia principal
85 distinguir padrões anormais de uso de energia de
86 todos os padrões normais de consumo em uma
87 amostra teste contendo exemplos da classe
88 normal e da classe fraudulenta (Jiang et al.,
89 2014). Alguns algoritmos que desempenham
90 essa técnica são: k-nearest neighbors (Aziz et al.,
91 2020) e (Kong et al., 2021); Support vector
92 machine (Toma et al., 2019) e (Jindal et al.,
93 2016); Randon Forest (S. Li et al., 2019) e (Qu et
94 al., 2020); Gradient Boosting (Punmiya & Choe,
95 2019) e (Razavi et al., 2019) e Ensemble
96 Learning (Gunturi & Sarkar, 2021) e (Aslam et
97 al., 2020).

98 Os algoritmos de classificação como K-
99 nearest neighbors, support vector machine,
100 decision tree e logistic regression já foram
101 estabelecidos em diversas aplicações baseadas
102 em problemas relacionados a energia elétrica e
103 também em outros campos de pesquisas (Himeur
104 et al., 2021). Entretanto, a maioria destes, são
105 baseados na extração de recursos artificiais que
106 exigem intervenções manuais e possuem baixa
107 precisão de detecção de roubo de eletricidade
108 (Zheng et al., 2018). É importante ressaltar que,
109 todos os algoritmos supracitados não consideram
110 a natureza sequencial dos dados, assumindo que
111 são independentes em função do tempo (Pereira
112 & Silveira, 2019), porém no mundo real, a
113 situação é contrária, pois o comportamento do
114 consumo de eletricidade é dinâmico (Z. Chen et
115 al., 2020).

116 Neste sentido, para resolver essas
117 limitações Nabil et al. (2018) propôs a utilização
118 de uma rede neural recorrente profunda
119 generalizada com base na detecção de roubo de

120 energia elétrica que pode efetivamente constatar
121 ataques cibernéticos em *smarts grids*. Aplicado
122 em problemas energéticos, esse modelo explora
123 a natureza da série temporal do consumo de
124 eletricidade dos clientes para implementar uma
125 rede neural recorrente de arquitetura *gated*
126 *recurrent unit (GRU)*, melhorando assim, o
127 desempenho de detecção e como consequência,
128 melhores resultados nas simulações de
129 desempenho se comparado com outros métodos
130 clássicos.

131 Nabil et al. (2019) adicionou *Non-*
132 *dominated Sorting Genetic Algorithm* para
133 ajustar os hiperparâmetros da rede GRU, que
134 explora a natureza da série temporal das leituras
135 de consumo de energia, melhorando, dessa
136 forma, o desempenho de detecção em relação aos
137 algoritmos clássicos.

138 As redes neurais recorrentes de
139 arquitetura GRU podem ser utilizadas com outras
140 arquiteturas para formar modelos híbridos de
141 detecção de fraude de energia elétrica. No
142 trabalho de Ullah et al. (2020), é proposto um
143 modelo de rede neural híbrida profunda baseado
144 na combinação das redes GRU e *Convolutional*
145 *Neural Network (CNN)* e do algoritmo *Particle*
146 *Swarm Optimization (PSO)*, onde os dados
147 utilizados eram de consumo de energia elétrica
148 de usuários em tempo real. A seleção e extração
149 de recursos são realizadas usando a rede CNN,
150 que reduzem a dimensionalidade e a redundância
151 presente na série temporal. Já a classificação dos
152 padrões de consumo em normais e fraudulentos é
153 feita usando a rede GRU com o algoritmo PSO.
154 Os resultados da simulação mostram que o
155 modelo proposto supera as técnicas existentes em
156 termos de detecção de roubo de energia. Além
157 disso, o modelo proposto também é mais robusto
158 e preciso do que os métodos de classificação
159 existentes.

160 Gul et al., (2020) empregou pela
161 primeira vez a *Bidirectional Gated Recurrent*
162 *Unit (BiGRU)* para classificar um consumidor
163 como honesto ou fraudulento, utilizando séries
164 históricas em tempo real, onde os experimentos
165 mostraram que esse modelo proposto teve
166 melhor desempenho que as técnicas de
167 classificação tradicionais.

168 Em vista da possibilidade de utilizar-se
169 algoritmos de classificação para detectar fraudes
170 no consumo de energia elétrica, o presente
171 trabalho visa melhorar a forma de detecção de
172 furtos de eletricidade com um modelo baseado
173 em diferentes camadas de redes neurais artificiais
174 denominado de BiGRU-CNN.

175 É importante ressaltar que modelos de
176 classificação mais precisos podem afetar os
177 custos e receitas para empresas do setor
178 energético de forma positiva. Desse modo, o
179 modelo neural classificativo de camadas distintas

180 proposto BiGRU-CNN foi comparado com as
181 redes neurais artificiais clássicas multilayer
182 perceptron (MLP), recurrent neural network
183 (RNN), Long Short Tem Memory (LSTM) e
184 GRU para verificar se suas classificações
185 relativas a furtos de energia são mais precisas ou
186 não. No qual as séries históricas de demanda de
187 energia elétrica de vários consumidores da
188 respectiva empresa foram utilizadas como
189 alimentação nos modelos neurais para efetuar a
190 detecção de fraudes.

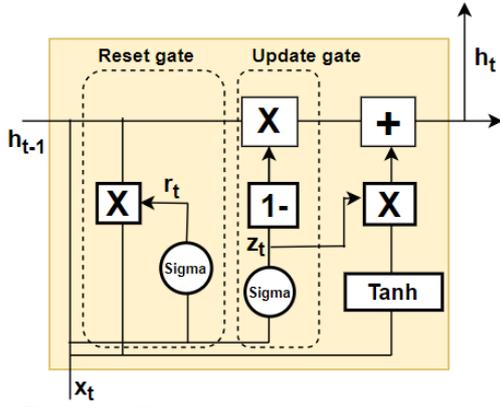
192 2. Referencial teórico

193 A Rede neural recorrente (RNN) é uma
194 espécie de rede neural artificial que emprega a
195 borda de conexão de nós temporais adjacentes e
196 introduz o conceito de tempo no modelo
197 preditivo, fazendo-o adequado para processar
198 séries temporais (Fengming et al., 2017).
199 Entretanto a arquitetura RNN convencional é
200 muito sensível à interferência dos períodos de
201 tempo adjacente, dando origem ao problema de
202 desaparecimento do fluxo de erro (Xie et al.,
203 2020). Uma das alternativas para superar esse
204 obstáculo é utilizar a rede neural de arquitetura
205 GRU que é, basicamente, uma versão melhorada
206 da LSTM (Huo et al., 2019). Geralmente ambas
207 as redes GRU e LSTM são adequadas para
208 resolver o problema do *vanishing gradiente* por
209 intermédio de suas portas multiplicativas, mas a
210 rede GRU é mais eficiente em alcançar a
211 convergência e atualizar os pesos internos
212 durante o treinamento, além de sua estrutura de
213 porta interna ser mais sucinta do que a rede
214 LSTM (Xie et al., 2020).

215 Uma célula ou unidade típica da rede de
216 arquitetura GRU é construída a partir de duas
217 portas denominadas de *reset gate* e *update gate*
218 (Lee et al., 2018). A primeira porta, *reset gate*,
219 filtra informações anteriormente irrelevantes nas
220 camadas ocultas (Fengming et al., 2017) e quanto
221 menor o seu valor, maior é a quantidade de
222 informações ignoradas (Huo et al., 2019). Por
223 outro lado a *update gate*, que é a segunda porta,
224 determina a quantidade de informações a serem
225 transferidas para a camada de saída (Fengming et
226 al., 2017) e quanto maior o seu valor, mais
227 informações contidas no estado anterior são
228 utilizadas (Huo et al., 2019).

229 A Figura 1, adaptada de P. Li et al.,
230 2020, mostra o diagrama estrutural de uma
231 célula da rede neural GRU regida pelas Equações
232 de 1 a 4, onde z_t é o *update gate*, r_t é o *reset gate*,
233 \tilde{h}_t é o estado oculto candidato do nó oculto
234 atual, h_t é o estado oculto atual, x_t é a entrada
235 atual da rede neural artificial e h_{t-1} é o estado
236 oculto do instante de tempo anterior (P. Li et al.,
237 2020).

238



239
240
241 Figura 1 - Unidade estrutural da rede neural artificial GRU

$$z_t = \sigma(w_{zx}x_t + u_{zh}h_{t-1}) \quad (1)$$

$$r_t = \sigma(w_{rx}x_t + u_{rh}h_{t-1}) \quad (2)$$

$$\tilde{h}_t = \tanh(w_{hx}x_t + r_t \odot u_{hh}h_{t-1}) \quad (3)$$

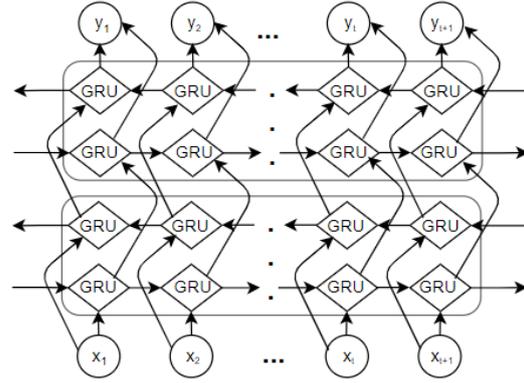
$$h_t = (1 - z_t) \odot \tilde{h}_t + z_t \odot h_{t-1} \quad (4)$$

242

243 Para muitas tarefas de modelagem
244 sequencial é interessante ter acesso a
245 informações tanto do passado quanto do futuro,
246 porém a rede neural GRU padrão processa a
247 sequência temporal em sua ordem cronológica e,
248 sendo assim, não é capaz de obter informações de
249 contextos futuros (J. X. Chen et al., 2019). Já a
250 GRU bidirecional (Bi-GRU) é capaz de realizar
251 essa operação, pois consiste em duas GRU
252 padrões que processam a sequência de entrada
253 em duas direções divergentes (cronológica e
254 anticronológica) e, em seguida, mescla essas
255 representações em uma única variável (Tao et al.,
256 2019). Com isso o modelo é capaz de explorar
257 informações passadas assim como futuras que,
258 por sua vez, podem fornecer resultados
259 preditivos mais eficientes (J. X. Chen et al.,
260 2019).

261 A Figura 2, adaptada de P. Li et al.,
262 2020, apresenta uma rede neural Bi-GRU de duas
263 camadas intermediárias orientada pelas
264 Equações de 5 a 9, onde \vec{h}_t^1 e \vec{h}_t^2 são os vetores
265 saídas das camadas *forward* da primeira e
266 segunda camada da rede no instante de tempo t .

267 Por outro lado, os vetores \overleftarrow{h}_t^1 e \overleftarrow{h}_t^2 representam as
268 saídas da primeira e segunda camada *backward*
269 da rede no mesmo instante de tempo t (P. Li et
270 al., 2020).



271
272
273 Figura 2 - unidade estrutural da rede neural BiGRU

$$\vec{h}_t^1 = f(w_{xh^1}x_t + w_{h^1h^1}\vec{h}_{t-1}^1 + b_{h^1}) \quad (5)$$

$$\overleftarrow{h}_t^1 = f(w_{xh^1}x_t + w_{h^1h^1}\overleftarrow{h}_{t+1}^1 + b_{h^1}) \quad (6)$$

$$\vec{h}_t^2 = f(w_{h^2h^2}\vec{h}_t^1 + w_{h^2h^2}\vec{h}_{t-1}^2 + b_{h^2}) \quad (7)$$

$$\overleftarrow{h}_t^2 = f(w_{h^1h^2}\overleftarrow{h}_t^1 + w_{h^2h^2}\overleftarrow{h}_{t+1}^2 + b_{h^2}) \quad (8)$$

$$y_t = g(w_{h^2y}\vec{h}_t^2 + w_{h^2y}\overleftarrow{h}_t^2 + b_y) \quad (9)$$

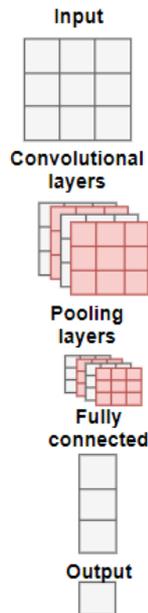
274

275

276 Ao contrário das redes recorrentes de
277 arquitetura GRU e Bi-GRU, a CNN é um tipo de
278 rede *feed-forward* que não é formada por
279 conexões cíclicas e não possui memória como
280 entrada (Lee et al., 2018). Em comparação com
281 os métodos de classificação tradicionais, a CNN
282 pode não apenas mapear relacionamentos não
283 lineares mais complexos, como também possui
284 boa capacidade de generalização (Gong et al.,
285 2020). Além de classificar, as redes CNN são
286 amplamente usadas para extração de recursos por
287 intermédio do kernel que é, em poucas palavras,
288 um filtro ou matriz que desliza sobre a entrada
289 para efetuar a operação de convolução e produzir
290 um mapa de recursos, onde diferentes kernels
291 geram diferentes mapas de recursos e todos estes
292 são mesclados para originar a saída da camada de
293 convolução (Madhure et al., 2020).

294 As redes neurais de arquitetura CNN
295 são compostas por camadas convolucionais,
296 camadas *pooling* e camadas totalmente
297 conectadas, onde a extração das características
298 das curvas de roubo de energia elétrica ficam
299 sobre a responsabilidade das camadas
300 convolucionais e camadas *pooling* (Gong et al.,
301 2020). A Figura 3, adaptada de Wu et al., 2021,
302 apresenta essas camadas organizadas de forma
303 genérica para compor a rede CNN e as Equações
304 10 e 11 definem o seu comportamento; onde x_i é
305 a entrada da i -th camada de convolução, y_i é a
306 saída da i -th camada de convolução, y' é a saída
307 da i -th camada *max-pooling*, f_i é a função de
ativação e, por último, as variáveis b_i e w_i são,

308 respectivamente, o vetor de *offset* e os pesos da
 309 *i*-th camada de convolução (Gong et al., 2020).
 310



311
 312 Figura 3 - Estrutura da rede neural
 313 Convolucional

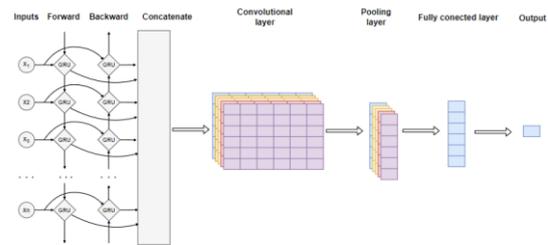
$$314 \quad y_i = f_i(x_i w_i + b_i) \quad (10)$$

$$315 \quad y' = \max(y_{i,j}) \quad (11)$$

315 3. Método proposto

316 O modelo BiGRU-CNN proposto no
 317 presente trabalho para detectar roubos de energia
 318 elétrica foi construído por uma camada BiGRU
 319 seguida de uma camada CNN. O conjunto de
 320 dados de entrada referente às séries históricas de
 321 demanda de energia dos consumidores foi
 322 manipulada para, então, alimentar a camada
 323 BiGRU que realizou o seu processamento com o
 324 objetivo de extrair dependências temporais de
 325 longo prazo. Essas características em função do
 326 tempo, que são representadas por dois vetores de
 327 estado oculto que possuem informações passadas
 328 e futuras, foram introduzidas na camada CNN
 329 para que relações locais significativas sejam
 330 capturadas por intermédio das camadas de
 331 convolução e de pooling.

332 Após esse procedimento, o conjunto de
 333 dados foi estruturado em várias dimensões que
 334 tiveram que passar pela camada *flatten* para
 335 voltarem a ser unidimensionais e, com isso,
 336 serem introduzidos na camada totalmente
 337 conectada para realizar a classificação de um
 338 consumidor de eletricidade em fraudulento ou
 339 honesto. A estrutura do modelo proposto
 340 supracitado é mostrada na Figura 4.



341
 342 Figura 4 - Fluxograma da rede neural BiGRU-
 343 CNN proposta

344 4. Dados

345 A base de dados utilizada no presente
 346 trabalho, também utilizada nos trabalhos de
 347 López et.al., (2014), Queiroz et.al., (2015) e
 348 Queiroz et.al., (2016), a qual foi fornecida por
 349 uma concessionária Colombiana de energia
 350 elétrica, que por motivos de sigilo de
 351 informações não pode ter seu nome divulgado.
 352 Os dados englobam o consumo real de energia
 353 elétrica em kWh de 462.433 usuários no ano de
 354 2011, onde o consumo foi aferido mensalmente.

355 Para complementar as informações dos
 356 consumidores, a empresa concessionária também
 357 forneceu uma base de dados com as revisões
 358 manuais realizadas em todos os clientes
 359 cadastrados, bem como as anomalias encontradas
 360 no momento de tais revisões.

361 Durante a vistoria manual foram
 362 encontrados diversos padrões de consumo com
 363 anomalias, sendo que as principais causas são
 364 fios emendados de forma clandestina, medidor
 365 furado, ligação anterior à caixa de medição e
 366 caixa de medição sem selo de segurança. Vale
 367 mencionar que, outras anomalias foram
 368 encontradas durante as verificações, mas pode-se
 369 afirmar que a maioria delas são provenientes de
 370 furtos de energia elétrica.

371 Como o padrão de consumo de energia
 372 elétrica dos usuários já está rotulado como
 373 fraudulento ou normal, as redes neurais serão
 374 treinadas por meio do aprendizado
 375 supervisionado. Onde a classificação fornecida
 376 pelos modelos será comparada com a classe real
 377 do consumidor, com isso é possível verificar com
 378 maior precisão e confiabilidade se o modelo
 379 proposto é capaz de classificar corretamente um
 380 cliente como fraudulento ou honesto.

381 É evidente que a maioria dos trabalhos
 382 relacionados ao tema encontrados na literatura
 383 não dispõem da real classificação do consumidor
 384 devido à complexidade em fazer as verificações
 385 de fraudes manualmente, seja por motivos
 386 financeiros ou pelo consumo de tempo e mão de
 387 obra, como pode ser visto em Queiroz et.al.,
 388 (2016).

389 Para contornar esta dificuldade, podem
 390 ser considerados todos os consumidores como
 391 honestos e criar dados fictícios de fraudes para

392 realizarem o treinamento de forma
393 supervisionada. Ambas as formas parecem não
394 criar modelos de classificação confiáveis por
395 falta de informações vitais dos dados.

396

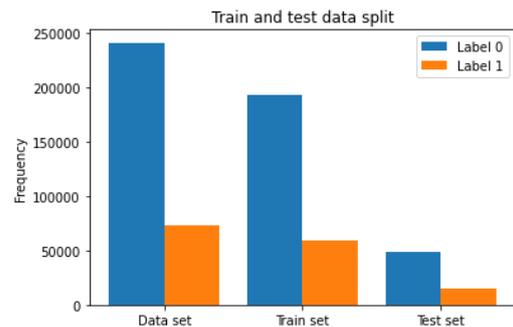
397 5. Pré-processamento dos Dados

398 Inicialmente, foi realizado uma limpeza
399 no banco de dados com o objetivo de eliminar os
400 registros com dados incompletos e retirar as
401 informações irrelevantes para a verificação de
402 furtos na curva de consumo dos usuários. Após
403 esse pré-processamento houve uma redução dos
404 dados que foi de 462.433 para 314.023.

405 Esse novo banco de dados reduzidos foi
406 inserido em um código de linguagem de
407 programação python no google colab. Após a
408 obtenção dos dados de consumo de energia
409 elétrica dos usuários pelo programa, a função
410 MinMaxScaler do pacote sklearn.preprocessing
411 foi utilizada para normalizar os dados antes de
412 serem introduzido nas redes neurais. A
413 normalização foi necessária pois os dados de
414 consumo de energia possuem uma grande
415 variação que poderia afetar o desempenho do
416 algoritmo durante o treinamento e, dessa
417 maneira, forneceria classificações equivocadas.
418 A Equação 12 apresenta como a normalização
419 dos dados foi realizada pela função
420 MinMaxScaler.

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (12)$$

421 O Dataset pré-processado contém
422 314.023 consumidores. Destes, 240.774
423 (76,674%) não cometem furto de energia elétrica
424 e são classificados como label 0. Já o restante dos
425 consumidores, equivalente a 73.249 (23,326%),
426 cometem esse crime e são rotulados de label 1. O
427 Dataset pré-processado foi fragmentado para
428 criar a amostra de treino e a amostra de teste. A
429 amostra de treino compreende 80% da totalidade
430 dos consumidores e serve para auxiliar as redes
431 neurais artificiais a ajustarem os seus respectivos
432 parâmetros internos durante treinamento. Esses
433 parâmetros encontrados são, portanto, avaliados
434 na amostra de teste, representada pelos 20%
435 restante dos consumidores totais, para verificar
436 se realmente são eficazes na classificação de
437 ocorrência de furto (label 1) ou consumo normal
438 de eletricidade (Label 0). Para que as redes
439 neurais não forneçam resultados viesados, as
440 amostras de treino e teste possuem a mesma
441 proporção de consumidores normais e
442 fraudulentos do Dataset pré-processado, ou seja,
443 76,674% e 23,326%. A Figura 5 ilustra o gráfico
444 dessas amostras, bem como suas composições.



445

446

447

Figura 5 – Composição das amostras de treino e teste

448

448 6. Redes neurais

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

Após o pré-processamento dos dados históricos de consumo de energia elétrica dos usuários, as séries temporais foram transformadas em um problema de aprendizado supervisionado de máquina, em outras palavras, criou-se uma sequência de pares de entradas e saídas para que uma decisão seja tomada e então comparada com a saída correta. Os parâmetros internos das redes neurais artificiais são modificados durante o treinamento pelo algoritmo Adam (Adaptive Moment Estimation) para que a taxa de classificações corretas por parte da rede seja a maior possível. Os dez neurônios de cada uma das duas camadas intermediárias de todos os modelos preditivos têm a rectified linear unit (ReLU), Equação 13, como função de ativação, enquanto o único neurônio na camada de saída possui a função de ativação binary crossentropy que é responsável por classificar um consumidor em fraudulento ou honesto. No que tange às camadas de arquitetura CNN das redes neurais, a quantidade do kernel_size foi definida em 6 e o número de filters foi estabelecido em 8. O treinamento de todas as redes neurais artificiais foi realizado em 150 épocas com batch size de 32.

$$f(x) = \max(0, x) \quad (13)$$

475

476

476 7. Métricas de comparação

477

478

479

480

481

482

483

484

485

486

487

Após o treinamento das redes, os modelos neurais foram alimentados pelas informações dos consumidores contidas na amostra de teste para verificar se são capazes de fornecer respostas precisas. Essas previsões foram organizadas em forma de matriz de confusão (Figura 6) para melhorar a capacidade de compreensão do desempenho individual de cada rede neural no que tange a classificação correta de fraude ou consumo normal de eletricidade.

		Prediction	
		1	0
Actual	1	True Positive (TP)	False Negative (FN)
	0	False Positive (FP)	True Negative (TN)

488

489

Figura 6 – matrix de confusão

490

491 A matriz de confusão ilustrada na
 492 Figura 6 é composta de quatro classes, onde o
 493 eixo das ordenadas representa a resposta correta
 494 desejada e o eixo das abscissas indica a previsão
 495 da rede neural. A classe **True Positive (TP)**
 496 abrange a resposta correta da rede para o evento
 497 de interesse. Neste caso a rede acerta que uma
 498 fraude de consumo de energia elétrica de fato
 499 ocorreu. Por outro lado, a classe **False Positive**
 500 **(FP)** corresponde ao total de respostas erradas da
 501 rede para o evento de interesse, ou seja, a rede fez
 502 uma previsão errônea no momento que indicou a
 503 ocorrência de uma fraude quando, na verdade,
 504 era um consumo normal. Já a classe **True**
 505 **Negative (TN)** compreende a classificação exata
 506 realizada pela rede neural no que diz respeito ao
 507 evento de não interesse, neste caso a rede
 508 classifica corretamente um consumidor que não
 509 comete fraudes. Por último, a classe **False**
 510 **Negative (FN)** apresenta os casos em que a rede
 511 indicou que não havia fraude por parte de um
 512 consumidor quando, na realidade, havia furto de
 513 energia elétrica.

514 A matriz de confusão representa de
 515 maneira abrangente o desempenho individual de
 516 cada modelo de previsão no que concerne à
 517 classificação de um consumidor como
 518 fraudulento. Entretanto, comparar os
 519 desempenhos preditivos de diferentes modelos a
 520 partir dessas matrizes não é suficiente, por conta
 521 disso, são extraídas as seguintes métricas:
 522 **Accuracy**, **Precision**, **Recall**, **F1-score** e **ROC**
 523 **AUC**. **Accuracy**: indica a quantidade de acertos
 524 da rede neural, classificando corretamente os
 525 consumidores que cometem fraudes e os que não.

526 **Precision**: é a razão das previsões que
 527 de fato eram verdadeiras no que diz respeito a
 528 consumidores fraudulentos, e todas as previsões
 529 que consideraram os clientes como fraudulentos,
 530 mesmo não sendo. Já **Recall**, também conhecida
 531 como sensibilidade, é a proporção entre as
 532 previsões assertivas dos consumidores que
 533 realizam fraudes e todos os consumidores que
 534 realmente cometeram furto de eletricidade.

535 A média ponderada da métrica
 536 **Precision** e **Recall** é definida como **F1-score**.
 537 Por último, a **ROC AUC** é representada pela área
 538 debaixo da curva formada pela fração de **False**
 539 **Positive** no eixo horizontal, e da razão **True**
 540 **Positive** que forma o eixo vertical. Além de
 541 calcular a métrica AUC, a curva ROC também é
 utilizada para definir um *threshold* ótimo que

542

543

544

545

546

547

548

549

550

551

552

553

554

consiga balancear a relação de **True Positive** e
False Positive, normalmente o *threshold* é
definido por padrão em 0,5.

Todas as métricas indicam resultados
satisfatórios quando estão próximas de 1, e
baixos resultados preditivos quando se
aproximam de 0, no que corresponde a
classificação correta de consumidores
fraudulentos.

As Equações 14, 15, 16 e 17 definem as
métricas **Accuracy**, **Precision**, **Recall** e **F1-score**,
respectivamente.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (14)$$

$$Precision = \frac{TP}{TP + FP} \quad (15)$$

$$Recall = \frac{TP}{TP + FN} \quad (16)$$

$$Accuracy = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}} \quad (17)$$

555

8 Experimentos

556

557

558

559

560

561

562

563

564

565

566

567

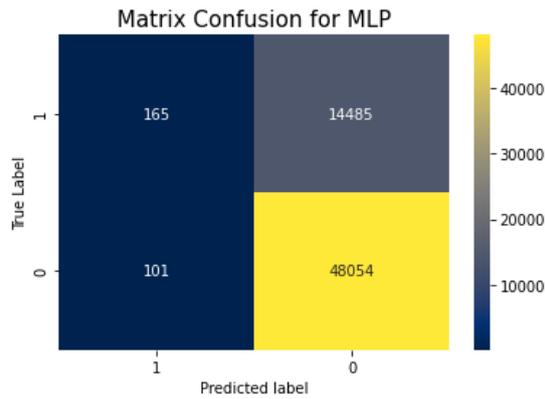
568

569

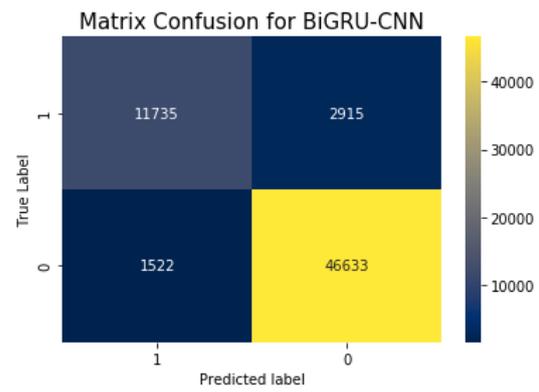
570

Após o treinamento das redes neurais
por intermédio da amostra de treino, os
parâmetros internos encontrados foram testados
para verificar a capacidade de generalizar os
mesmos resultados para dados não vistos antes,
que estão contidos na amostra de teste.

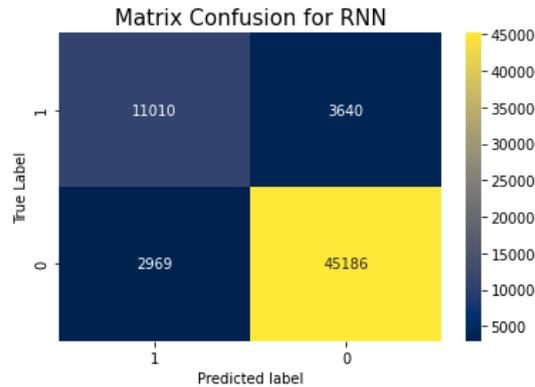
Os resultados individuais de cada rede
neural artificial são ilustrados nas matrizes de
confusão na Figura 7. Além disso, foi construída
a Tabela 1 que possui métricas baseadas nos
valores de cada matriz, para que a comparação
dos modelos distintos se torne mais fácil. Além
das métricas, a tabela possui o tempo necessário
de simulação que cada rede demandou para obtê-
las.



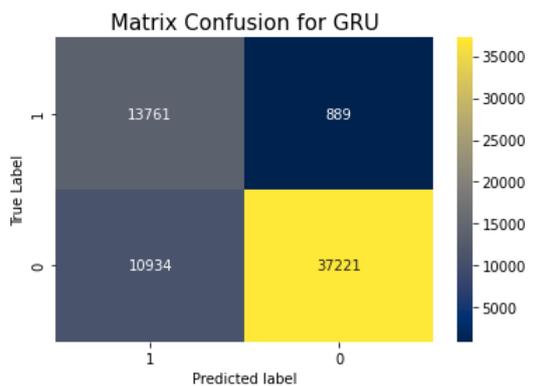
571



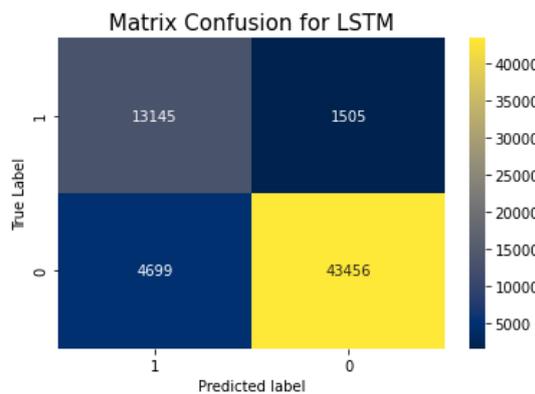
575



572



573



574

Algoritmo	Accuracy	Precision	Recall	F1-Score	ROC AUC	Temp (min)
MLP	0,767	0,620	0,011	0,022	0,660	45,25
RNN	0,894	0,787	0,751	0,769	0,936	51,17
GRU	0,811	0,557	0,939	0,699	0,912	164,45
LSTM	0,901	0,736	0,897	0,809	0,946	157,47
BiGRU-CNN	0,929	0,885	0,801	0,841	0,966	224,55

576

Tabela 1 – Métrica de Comparação

577

578 Como pode ser observado na Tabela 1,
 579 o modelo MLP foi o que teve o pior desempenho
 580 em classificar corretamente um consumidor
 581 como fraudulento ou honesto, levando em
 582 consideração as métricas de *Accuracy*, *Recall*,
 583 *F1-Score* e *ROC AUC*. Isso pode ser explicado
 584 pelo fato dessa rede não conseguir extrair o
 585 comportamento dinâmico temporal dos dados de
 586 demanda de energia dos diferentes
 587 consumidores, pois sua estrutura não possui
 588 dispositivos de retroalimentação das
 589 informações. Em contrapartida, por não possuir
 590 tais mecanismos que processem a temporalidade
 591 dos dados de consumo de energia dos usuários, a
 592 rede possui menos parâmetros a serem
 593 modificados durante o treinamento e,
 594 conseqüentemente, menor tempo de simulação
 595 que as redes recorrentes.

596

597 Já a rede recorrente de arquitetura GRU
 598 teve o pior desempenho na métrica *Precision* e a
 599 melhor performance para a métrica de *Recall*.
 600 Quando a GRU padrão foi utilizada para criar o
 601 modelo proposto BiGRU-CNN, as métricas de
 602 comparação sofreram modificações
 603 significativas. Essa nova rede formada a partir de
 604 camadas de diferentes arquiteturas com o
 605 mecanismo bidirecional teve o melhor
 606 desempenho em 4 das 5 métricas, sendo elas
 607 *Accuracy*, *Precision*, *F1-Score* e *ROC AUC*.

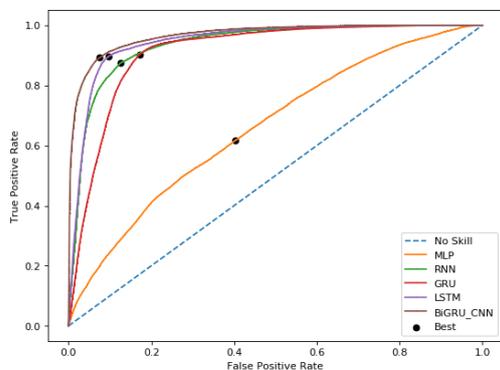
608

609 Esses resultados demonstram que os
 610 procedimentos efetuados na GRU padrão
 611 aumentaram consideravelmente sua capacidade
 612 em classificar de forma correta o consumo de um
 613 usuário em fraudulento ou honesto. Além do
 614 mais, mostram sua superioridade aos modelos de
 615 redes neurais clássicos. Por outro lado, o seu
 616 tempo de simulação foi longo quando comparado
 617 aos demais, por possuírem um número maior de

615 parâmetros a serem modificados durante o
616 treinamento.

617 Pode-se observar na Figura 8 as curvas
618 ROC de todos os modelos neurais que foram
619 utilizadas para calcular a métrica AUC da Tabela
620 1, bem como o ponto ideal que aumenta a
621 proporção de *True Positive* ao mesmo tempo que
622 diminui a razão de *False Positive*. Com essas
623 coordenadas é possível determinar a *G-mean* que
624 posteriormente é utilizada para encontrar o
625 *threshold* ótimo, responsável por melhorar o
626 poder de classificação das redes. A Tabela 2
627 apresenta a *G-mean* associada ao seu *threshold*
628 ideal.

629 Analisando as curvas ROC construídas
630 a partir das classificações do tipo de consumo dos
631 usuários de energia elétrica feitas pelas redes
632 neurais, é possível verificar que a curva da rede
633 MLP é a mais próxima da curva que indica um
634 modelo de classificação ineficiente. Não
635 importando o *threshold* escolhido, seu
636 desempenho de classificação sempre será o pior
637 quando comparado com os demais modelos que
638 possuem a capacidade de processar a
639 autocorrelação temporal do consumo de energia
640 elétrica. Agora, observando as redes recorrentes
641 é possível verificar que suas performances são
642 similares e a superioridade do modelo BiGRU-
643 CNN comprovada pelo maior distanciamento da
644 curva do modelo sem habilidade de classificação.
645



646

647 Figura 7 - curva ROC de todos as redes neurais

Algoritmo	Best Threshold	G-Mean
MLP	0,236	0,607
RNN	0,186	0,874
GRU	0,605	0,864
LSTM	0,501	0,899
BiGRU-CNN	0,330	0,908

648

649 9 Conclusão

650 O presente trabalho propôs o modelo
651 BiGRU-CNN para classificar usuários de energia
652 elétrica como fraudulentos ou honestos a partir
653 dos padrões de seus consumos. Essa classificação
654 tem como objetivo auxiliar empresas do setor

655 energético em sua tomada de decisões no
656 momento de fazer ou não a inspeção manual da
657 unidade consumidora de energia elétrica.

658 Os resultados experimentais mostraram
659 que alimentar uma camada BiGRU com a série
660 histórica para extrair suas correlações temporais
661 de longo prazo e, em seguida, introduzir essas
662 características de tempo em uma camada CNN
663 para que tendências locais possam ser captadas,
664 mostrou-se eficiente quando comparados pelas
665 métricas *Accuracy*, *Precision*, *F1-Score* e ROC
666 AUC com as redes MLP, RNN, GRU e LSTM.
667 Para garantir que o modelo proposto BiGRU-
668 CNN possa ser, de fato, superior aos demais
669 modelos de classificação de consumidores que
670 cometem furto de energia elétrica, trabalhos
671 futuros necessitam ser realizados alterando os
672 hiperparâmetros das redes neurais, bem como a
673 série temporal dos consumidores que as
674 alimentam.

675 Referências

676 Ahmad, T. (2017). Non-technical loss analysis
677 and prevention using smart meters.
678 *Renewable and Sustainable Energy*
679 *Reviews*, 72(November 2015), 573–589.
680 <https://doi.org/10.1016/j.rser.2017.01.100>

681 Ahmad, T., Chen, H., Wang, J., & Guo, Y.
682 (2017). Review of various modeling
683 techniques for the detection of electricity
684 theft in smart grid environment.
685 *Renewable and Sustainable Energy*
686 *Reviews*, October, 1–18.
687 <https://doi.org/10.1016/j.rser.2017.10.040>

688 Aslam, Z., Javaid, N., Ahmad, A., Ahmed, A.,
689 & Gulfam, S. M. (2020). A combined
690 deep learning and ensemble learning
691 methodology to avoid electricity theft in
692 smart grids. *Energies*, 13(21), 1–24.
693 <https://doi.org/10.3390/en13215599>

694 Aziz, S., Naqvi, S. Z. H., Khan, M. U., &
695 Aslam, T. (2020). Electricity Theft
696 Detection using Empirical Mode
697 Decomposition and K-Nearest Neighbors.
698 *2020 International Conference on*
699 *Emerging Trends in Smart Technologies,*
700 *ICETST 2020.*
701 <https://doi.org/10.1109/ICETST49965.2020.9080727>

703 Chen, J. X., Jiang, D. M., & Zhang, Y. N.
704 (2019). A Hierarchical Bidirectional GRU
705 Model with Attention for EEG-Based
706 Emotion Classification. *IEEE Access*, 7,
707 118530–118540.
708 <https://doi.org/10.1109/ACCESS.2019.2936817>

- 710 Chen, Z., Meng, D., Zhang, Y., Xin, T., & Xiao, 766 16601
711 D. (2020). Electricity Theft Detection
712 Using Deep Bidirectional Recurrent
713 Neural Network. *2020 22nd International*
714 *Conference on Advanced Communication*
715 *Technology (ICACT)*, 401–406.
- 716 Fengming, Z., Shufang, L., Zhimin, G., Bo, W.,
717 Shiming, T., & Mingming, P. (2017).
718 Anomaly detection in smart grid based on
719 encoder-decoder framework with
720 recurrent neural network. *The Journal of*
721 *China Universities of Posts and*
722 *Telecommunications*, 24(December), 67–
723 73. [https://doi.org/10.1016/S1005-](https://doi.org/10.1016/S1005-8885(17)60243-7)
724 [8885\(17\)60243-7](https://doi.org/10.1016/S1005-8885(17)60243-7)
- 725 Glauner, P., Meira, J. A., Valtchev, P., State, R.,
726 & Bettinger, F. (2017). The challenge of
727 non-technical loss detection using
728 artificial intelligence: A survey.
729 *International Journal of Computational*
730 *Intelligence Systems*, 10(1), 760–775.
731 <https://doi.org/10.2991/ijcis.2017.10.1.51>
- 732 Gong, X., Tang, B., Zhu, R., Liao, W., & Song,
733 L. (2020). Data augmentation for
734 electricity theft detection using
735 conditional variational auto-encoder.
736 *Energies*, 13(17), 1–14.
737 <https://doi.org/10.3390/en13174291>
- 738 Guerrero, J. I., Monedero, I., Biscarri, F.,
739 Biscarri, J., Millan, R., & Leon, C. (2018).
740 Non-Technical Losses Reduction by
741 Improving the Inspections Accuracy in a
742 Power Utility. *IEEE Transactions on*
743 *Power Systems*, 33(2), 1209–1218.
744 <https://doi.org/10.1109/TPWRS.2017.272>
745 1435
- 746 Gul, H., Javaid, N., Ullah, I., & Qamar, A. M.
747 (2020). applied sciences Detection of
748 Non-Technical Losses Using SOSTLink
749 and Bidirectional Gated Recurrent Unit to
750 Secure Smart Meters. *Applied Sciences*,
751 10. <https://doi.org/10.3390/app10093151>
- 752 Gunturi, S. K., & Sarkar, D. (2021). Ensemble
753 machine learning models for the detection
754 of energy theft. *Electric Power Systems*
755 *Research*, 192(October), 106904.
756 [https://doi.org/10.1016/j.epr.2020.10690](https://doi.org/10.1016/j.epr.2020.106904)
757 4
- 758 Himeur, Y., Ghanem, K., Alsalemi, A.,
759 Bensaali, F., & Amira, A. (2021).
760 Artificial intelligence based anomaly
761 detection of energy consumption in
762 buildings: A review, current trends and
763 new perspectives. *Applied Energy*,
764 287(February), 116601.
765 <https://doi.org/10.1016/j.apenergy.2021.1>
- 767 Huo, X., Wu, K., Miao, W., Wang, L., He, H.,
768 & Su, D. (2019). Research on Network
769 Traffic Anomaly Detection of Source-
770 Network-Load Industrial Control System
771 Based on GRU-OCSVM. *IOP Conference*
772 *Series: Earth and Environmental Science*,
773 300(4). [https://doi.org/10.1088/1755-](https://doi.org/10.1088/1755-1315/300/4/042043)
774 [1315/300/4/042043](https://doi.org/10.1088/1755-1315/300/4/042043)
- 775 Jiang, R., Lu, R., Wang, Y., Luo, J., Shen, C., &
776 Shen, X. S. (2014). Energy-Theft
777 Detection Issues for Advanced Metering
778 Infrastructure in Smart Grid. *Tsinghua*
779 *Science and Technology*, 19(2), 105–120.
- 780 Jindal, A., Dua, A., Kaur, K., Singh, M., Kumar,
781 N., & Mishra, S. (2016). Decision Tree
782 and SVM-Based Data Analytics for Theft
783 Detection in Smart Grid. *IEEE*
784 *Transactions on Industrial Informatics*,
785 12(3), 1005–1016.
786 <https://doi.org/10.1109/TII.2016.2543145>
- 787 Kong, X., Zhao, X., Liu, C., Li, Q., Dong, D. L.,
788 & Li, Y. (2021). Electricity theft detection
789 in low-voltage stations based on similarity
790 measure and DT-KSVM. *International*
791 *Journal of Electrical Power and Energy*
792 *Systems*, 125(February 2020).
793 [https://doi.org/10.1016/j.ijepes.2020.1065](https://doi.org/10.1016/j.ijepes.2020.106544)
794 44
- 795 Lee, K., Kim, J. K., Kim, J., Hur, K., & Kim, H.
796 (2018). CNN and GRU combination
797 scheme for Bearing Anomaly Detection in
798 Rotating Machinery Health Monitoring.
799 *1st IEEE International Conference on*
800 *Knowledge Innovation and Invention,*
801 *ICKII 2018*, 102–105.
802 [https://doi.org/10.1109/ICKII.2018.85691](https://doi.org/10.1109/ICKII.2018.8569155)
803 55
- 804 Li, P., Luo, A., Liu, J., Wang, Y., Zhu, J., Deng,
805 Y., & Zhang, J. (2020). Bidirectional
806 gated recurrent unit neural network for
807 Chinese address element segmentation.
808 *ISPRS International Journal of Geo-*
809 *Information*, 9(11).
810 <https://doi.org/10.3390/ijgi9110635>
- 811 Li, S., Han, Y., Yao, X., Yingchen, S., Wang, J.,
812 & Zhao, Q. (2019). Electricity Theft
813 Detection in Power Grids with Deep
814 Learning and Random Forests. *Journal of*
815 *Electrical and Computer Engineering*,
816 2019.
817 <https://doi.org/10.1155/2019/4136874>
- 818 Lopéz, G. P., Rider, M. J. & Zapata-Grisales, C.
819 J. (2014). Detecção de Fraudes nas Redes
820 de Distribuição de Energia Elétrica

- 821 Utilizando Árvore de Decisão. *Simpósio*
822 *Brasileiro de Sistemas Elétricos* pp.1-6.
823 Anais 2014 - Foz do Iguaçu -PR
- 824 Madhure, R. U., Raman, R., & Singh, S. K.
825 (2020). CNN-LSTM based Electricity
826 Theft Detector in Advanced Metering
827 Infrastructure. *2020 11th International*
828 *Conference on Computing,*
829 *Communication and Networking*
830 *Technologies, ICCCNT 2020.*
831 <https://doi.org/10.1109/ICCCNT49239.20>
832 [20.9225572](https://doi.org/10.1109/ICCCNT49239.20.9225572)
- 833 Nabil, M., Ismail, M., Mahmoud, M., Shahin,
834 M., Qaraqe, K., & Serpedin, E. (2018).
835 Deep Recurrent Electricity Theft
836 Detection in AMI Networks with Random
837 Tuning of Hyper-parameters. *Proceedings*
838 *- International Conference on Pattern*
839 *Recognition, 2018-Augus, 740–745.*
840 <https://doi.org/10.1109/ICPR.2018.85457>
841 48
- 842 Nabil, M., Mahmoud, M., Ismail, M., &
843 Serpedin, E. (2019). Deep recurrent
844 electricity theft detection in ami networks
845 with evolutionary hyper-parameter tuning.
846 *Proceedings - 2019 IEEE International*
847 *Congress on Cybermatics: 12th IEEE*
848 *International Conference on Internet of*
849 *Things, 15th IEEE International*
850 *Conference on Green Computing and*
851 *Communications, 12th IEEE International*
852 *Conference on Cyber, Physical and So,*
853 *1002–1008.*
854 <https://doi.org/10.1109/iThings/GreenCo>
855 [m/CPSCoM/SmartData.2019.00175](https://doi.org/10.1109/iThings/GreenCom/CPSCoM/SmartData.2019.00175)
- 856 Ouyang, Z., Sun, X., & Yue, D. (2017).
857 Hierarchical Time Series Feature
858 Extraction for Power Consumption
859 Anomaly Detection. *Advanced*
860 *Computational Methods in Energy,*
861 *Power, Electric Vehicles, and Their*
862 *Integration, 3, 267–275.*
863 [https://doi.org/10.1007/978-981-10-6364-](https://doi.org/10.1007/978-981-10-6364-0)
864 0
- 865 Pereira, J., & Silveira, M. (2019). Unsupervised
866 Anomaly Detection in Energy Time Series
867 Data Using Variational Recurrent
868 Autoencoders with Attention.
869 *Proceedings - 17th IEEE International*
870 *Conference on Machine Learning and*
871 *Applications, ICMLA 2018, 1275–1282.*
872 <https://doi.org/10.1109/ICMLA.2018.002>
873 07
- 874 Punmiya, R., & Choe, S. (2019). Energy theft
875 detection using gradient boosting theft
876 detector with feature engineering-based
preprocessing. *IEEE Transactions on*
877 *Smart Grid, 10(2), 2326–2329.*
878 <https://doi.org/10.1109/TSG.2019.289259>
879 5
- 880
- 881 Qu, Z., Li, H., Wang, Y., Zhang, J., Abu-Siada,
882 A., & Yao, Y. (2020). Detection of
883 electricity theft behavior based on
884 improved synthetic minority oversampling
885 technique and random forest classifier.
886 *Energies, 13(8).*
887 <https://doi.org/10.3390/en13082039>
- 888 Queiroz, A. S., Franco, E. M. C., López, P. G.
889 (2015). Algoritmos de Inteligência
890 Computacional Utilizados na Detecção de
891 Fraudes nas Redes de Distribuição de
892 Energia Elétrica. *XI Latin- American*
893 *Congress Electricity Generation and*
894 *Transmission.* São José dos Campos - SP
- 895 Queiroz, A. S., Franco, E. M. C., López, P. G.
896 (2016). Detecção de Fraudes nas Redes de
897 Distribuição de Energia Elétrica Utilizado
898 Técnicas de Inteligência Computacional.
899 *VI Simpósio Brasileiro de Sistemas*
900 *Elétricos.* Natal - RN
- 901 Razavi, R., Gharipour, A., Fleury, M., & Akpan,
902 I. J. (2019). A practical feature-
903 engineering framework for electricity
904 theft detection in smart grids. *Applied*
905 *Energy, 238(January), 481–494.*
906 <https://doi.org/10.1016/j.apenergy.2019.0>
907 1.076
- 908 Sarwar, S., Chen, W., & Waheed, R. (2017).
909 Electricity consumption, oil price and
910 economic growth: Global perspective.
911 *Renewable and Sustainable Energy*
912 *Reviews, 76(February 2016), 9–18.*
913 <https://doi.org/10.1016/j.rser.2017.03.063>
- 914 Tao, Q., Liu, F., Li, Y., & Sidorov, D. (2019).
915 Air Pollution Forecasting Using a Deep
916 Learning Model Based on 1D Convnets
917 and Bidirectional GRU. *IEEE Access, 7,*
918 *76690–76698.*
919 <https://doi.org/10.1109/ACCESS.2019.29>
920 21578
- 921 Toma, R. N., Hasan, M. N., Nahid, A. Al, & Li,
922 B. (2019). Electricity Theft Detection to
923 Reduce Non-Technical Loss using
924 Support Vector Machine in Smart Grid.
925 *1st International Conference on Advances*
926 *in Science, Engineering and Robotics*
927 *Technology 2019, ICASERT 2019,*
928 *2019(Icasert).*
929 <https://doi.org/10.1109/ICASERT.2019.8>
930 934601
- 931 Ullah, A., Javaid, N., Samuel, O., Imran, M., &

- 932 Shoaib, M. (2020). CNN and GRU based
933 Deep Neural Network for Electricity Theft
934 Detection to Secure Smart Grid. *2020*
935 *International Wireless Communications*
936 *and Mobile Computing (IWCMC)*, 1598–
937 1602.
- 938 Viegas, J. L., Esteves, P. R., Melício, R.,
939 Mendes, V. M. F., & Vieira, S. M. (2017).
940 Solutions for detection of non-technical
941 losses in the electricity grid: A review.
942 *Renewable and Sustainable Energy*
943 *Reviews*, *80*(August 2016), 1256–1268.
944 <https://doi.org/10.1016/j.rser.2017.05.193>
- 945 Wu, K., Wu, J., Feng, L., Yang, B., Liang, R.,
946 Yang, S., & Zhao, R. (2021). An
947 attention-based CNN-LSTM-BiLSTM
948 model for short-term electric load
949 forecasting in integrated energy system.
950 *International Transactions on Electrical*
951 *Energy Systems*, *31*(1), 1–15.
952 <https://doi.org/10.1002/2050-7038.12637>
- 953 Xie, X., Wang, B., Wan, T., & Tang, W. (2020).
954 Multivariate Abnormal Detection for
955 Industrial Control Systems Using 1D
956 CNN and GRU. *IEEE Access*, *8*, 88348–
957 88359.
958 <https://doi.org/10.1109/ACCESS.2020.29>
959 93335
- 960 Zheng, Z., Yang, Y., Niu, X., Dai, H. N., &
961 Zhou, Y. (2018). Wide and Deep
962 Convolutional Neural Networks for
963 Electricity-Theft Detection to Secure
964 Smart Grids. *IEEE Transactions on*
965 *Industrial Informatics*, *14*(4), 1606–1615.
966 <https://doi.org/10.1109/TII.2017.2785963>
- 967