



**UNIVERSIDADE FEDERAL DO OESTE DO PARÁ
PRÓ-REITORIA DE PESQUISA, PÓS-GRADUAÇÃO E INOVAÇÃO TECNOLÓGICA
PROGRAMA DE PÓS-GRADUAÇÃO EM SOCIEDADE, NATUREZA E DESENVOLVIMENTO**

DAVI GUIMARÃES DA SILVA

**PREVISÃO DE SÉRIES TEMPORAIS DE CONSUMO DE ENERGIA ELÉTRICA
COM APRENDIZAGEM PROFUNDA PARA UM SISTEMA IOT**

**SANTARÉM-PA
2023**

DAVI GUIMARÃES DA SILVA

**PREVISÃO DE SÉRIES TEMPORAIS DE CONSUMO DE ENERGIA ELÉTRICA
COM APRENDIZAGEM PROFUNDA PARA UM SISTEMA IOT**

Tese apresentada ao Programa de Pós-graduação em Sociedade, Natureza e Desenvolvimento, da Universidade Federal do Oeste do Pará – UFOPA, como requisito parcial para a obtenção do título de Doutor em Ciências Ambientais. Área de Concentração: Gestão do Conhecimento e Inovação para o Desenvolvimento Sustentável.

Orientador: Prof. Dr. Anderson Alvarenga de Moura Meneses.

**SANTARÉM-PA
2023**

Dados Internacionais de Catalogação-na-Publicação (CIP)
Sistema Integrado de Bibliotecas – SIBI/UFOPA

S586p Silva, Davi Guimarães da
Previsão de séries temporais de consumo de energia elétrica com aprendizagem profunda para um sistema IoT./ Davi Guimarães da Silva. – Santarém, 2023.
155 p. : il.
Inclui bibliografias.

Orientador: Anderson Alvarenga de Moura Menezes.
Tese (Doutorado) – Universidade Federal do Oeste do Pará, Pró-Reitoria de Pesquisa, Pós-Graduação e Inovação Tecnológica, Programa de Pós-Graduação em Sociedade, Natureza e Desenvolvimento.

1. Aprendizagem profunda. 2. Internet das coisas. 3. Redes neurais profundas. 4. Séries temporais. 5. Consumo – Energia Elétrica. I. Menezes, Anderson Alvarenga de Moura, *orient.*. II. Título.

CDD: 23 ed. 621.042

DAVI GUIMARÃES DA SILVA

**PREVISÃO DE SÉRIES TEMPORAIS DE CONSUMO DE ENERGIA ELÉTRICA
COM APRENDIZAGEM PROFUNDA PARA UM SISTEMA IOT**

Tese apresentada ao Programa de Pós-graduação em Sociedade, Natureza e Desenvolvimento, da Universidade Federal do Oeste do Pará – UFOPA, como parte dos requisitos à obtenção do título de Doutor em Ciências Ambientais. Área de Concentração: Gestão do Conhecimento e Inovação para o Desenvolvimento Sustentável.

Orientador: Prof. Dr. Anderson Alvarenga de Moura Meneses.

Data da Aprovação: 03/04/2023

Prof. Dr. Anderson Alvarenga de Moura Meneses – PPGSND/UFOPA (Presidente)

Prof. Dr. Roberto Schirru – UFRJ (Examinador Externo)

Prof. Dra. Marla Teresinha Barbosa Geller - ULBRA (Examinadora Externo)

Prof. Dra. Andressa dos Santos Nicolau – UFRJ (Examinadora Externo)

Prof. Dr. Omar Andres Carmona Cortes – IFMA (Examinador Externo)

Prof. Dra. Helaine Cristina Moraes Furtado – IEG/UFOPA

Aos meus pais, irmãs, esposa e filhos.

AGRADECIMENTO

Em primeiro lugar quero agradecer a Deus, meu grande mestre, que sempre guiou meus passos, dando sabedoria e entendimento.

Aos meus pais, Abiezer Pereira da Silva e Helena Guimarães da Silva, pelo esforço incondicional durante toda a minha caminhada. Pelas suas orações e palavras de incentivo em todos os momentos. Amo vocês!

À minha esposa Nathalia Araújo por dividir comigo momentos de alegria e alguns também difíceis. À minha filha Heloísa e o bebê (na barriga da mãe). Amo vocês!

À minhas irmãs Ábida, Elane e Laíz, pelo apoio e todos os momentos que precisei. Meus cunhados, Ronildo (*in memoriam*), Marques Jr. e Jaime Júnior. Meus sobrinhos Lucas, Emanuely, Esther, Giovanna e Isabella, que tanto amo. Ao meu sogro Amarilson Ferreira e minha sogra Antônia Morais.

Ao meu orientador prof. Dr. Anderson A. M. Meneses, pela valiosa condução de todo o processo de orientação, por acreditar, apoiar, incentivar desde o início e pelo profissionalismo exemplar. Aos demais pesquisadores do Laboratório de Inteligência Computacional (LABIC-UFOPA) pelas valiosas contribuições (Marla Geller, Mauro Sérgio, Paulo Guilherme, Yana Pereira, Kemuel, dentre outros).

Ao Instituto Federal do Pará (IFPA) Campus Itaituba, por possibilitar aos servidores a formação continuada.

À UFOPA, por nos proporcionar cursar um Programa de Doutorado de muita relevância, que fomenta o desenvolvimento de pesquisas de interesse universal. Aos demais professores do PPGSND, por colaborarem com minha formação de pesquisador.

Enfim, meus mais sinceros agradecimentos a todos que me apoiaram durante esta longa caminhada. Deus os abençoe!

RESUMO

Consumo de energia e eficiência energética são tópicos que têm atraído a atenção dos pesquisadores nos últimos anos, a fim de buscar soluções científicas e tecnológicas para a produção de energia e redução de custos. Uma das alternativas que têm obtido resultados satisfatórios é o uso de tecnologias baseadas em Internet das Coisas (*Internet of Things*, IoT) e Aprendizagem Profunda (AP). Com base nisso, propõe-se avaliar o desempenho de Redes Neurais Profundas (RNPs) *Long Short-Term Memory* (LSTM) e LSTM Bidirecional (BiLSTM) para predição de Séries Temporais (STs) univariadas de consumo de energia elétrica. Para este objetivo foi utilizada a validação cruzada para Séries Temporais (VC-ST). Os resultados indicaram que os modelos LSTM obtiveram uma tendência de melhor desempenho em comparação com os algoritmos Extreme Gradient Boost (XGBoost) e Random Forest (RF) e que por sua vez os modelos BiLSTM obtiveram melhor desempenho que os modelos LSTM, com diferença estatisticamente significativa de acordo com os testes de Friedman ($p = 0,0455$) considerando quatro *datasets*. Desse modo, corroboram-se os resultados experimentais comparativos e análises estatísticas que a AP pode ser utilizada para predição obtendo melhores resultados e que apesar de ter um tempo maior de treinamento, a BiLSTM foi estatisticamente superior a LSTM. Por fim, pode-se destacar que uma das principais vantagens da predição é a possibilidade de, a partir da integração da AP com IoT, traçar estratégias eficazes de curto, médio e longo prazo, para promover soluções apropriadas para cada situação.

Palavras-Chave: Aprendizagem Profunda. Internet das Coisas. Redes Neurais Profundas. Séries Temporais. Consumo de Energia Elétrica.

ABSTRACT

Energy consumption and energy efficiency are topics that have attracted the attention of researchers in recent years, in order to seek scientific and technological solutions for energy production and cost reduction. One of the alternatives that have obtained satisfactory results is the use of technologies based on Internet of Things (IoT) and Deep Learning (DL). Based on this, it is proposed to evaluate the performance of Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM) Deep Neural Networks (DNNs) for the prediction of univariate Time Series (STs) of electricity consumption. Cross-validation for Time Series (CV-TS) was used for this purpose. The results indicated that the LSTM models tended to perform better in comparison with the Extreme Gradient Boost (XGBoost) and Random Forest (RF) algorithms, and that in turn the BiLSTM models performed better than the LSTM models, with a statistically significant difference according to Friedman's tests ($p = 0.0455$) considering four *datasets*. Thus, the comparative experimental results and statistical analyses corroborate that AP can be used for prediction obtaining better results, and that despite having a longer training time, BiLSTM was statistically superior to LSTM. Finally, it can be emphasized that one of the main advantages of prediction is the possibility, based on the integration of DL with IoT systems, to design effective short, medium and long term strategies to promote appropriate solutions for each situation.

Keywords: Deep Learning. Internet of Things. Deep Neural Networks. Time Series. Electricity Consumption.

LISTA DE FIGURAS

Figura 1: Resultado da string de busca.	29
Figura 2: Resultado da string de busca.	30
Figura 3: Resultado da string de busca.	30
Figura 4: Diferenças entre IA, AM e AP.	46
Figura 5: Três ondas históricas do aprendizado profundo.	48
Figura 6: Camadas das Redes Neurais.	52
Figura 7: Modelo de um neurônio artificial.	52
Figura 8: Funções de Ativação de uma RNA.	57
Figura 9: Estrutura de um módulo de uma rede LSTM.	62
Figura 10: Módulo LSTM com destaque para o portão de esquecimento.	63
Figura 11: Módulo LSTM com destaque para o portão de entrada.	64
Figura 12: Módulo LSTM com destaque para o portão de saída.	65
Figura 13: Estrutura da célula de uma rede BiLSTM.	66
Figura 14: Modelo da rede BiLSTM.	67
Figura 15: Algoritmo RF.	72
Figura 16: Esquema conceitual do sistema EnergySaver.	75
Figura 17: Módulo LSTM proposto.	76
Figura 18: Modelo de processo DSR.	79
Figura 19: Fluxo metodológico DSR para a presente pesquisa.	80
Figura 20: Exemplo treinamento-teste com janela deslizante.	88
Figura 21: Exemplo de validação cruzada com 5-folds de ST com <i>TimeSeriesSplit</i>	89
Figura 22: Divisão dos dados de cada <i>dataset</i> em treinamento-teste e validação.	91
Figura 23: Amostras dos dados de consumo elétrico do <i>dataset</i> UCI-Household.	97
Figura 24: <i>Boxplots</i> de consumo de energia elétrica do <i>dataset</i> UCI-Household.	97
Figura 25: Função de autocorrelação no <i>dataset</i> UCI-Household.	98
Figura 26: Decomposição clássica da ST para o <i>dataset</i> UCI-Household.	98
Figura 27: Amostras dos dados de consumo elétrico do <i>dataset</i> LABIC-Building.	99
Figura 28: <i>Boxplots</i> de consumo de energia elétrica do <i>dataset</i> LABIC-Building.	99
Figura 29: Função de autocorrelação no <i>dataset</i> LABIC-Building.	100
Figura 30: Decomposição clássica da ST para o <i>dataset</i> LABIC-Building.	100
Figura 31: Amostras dos dados de consumo elétrico do <i>dataset</i> Singapore.	101
Figura 32: <i>Boxplots</i> de consumo de energia elétrica do <i>dataset</i> Singapore.	101

Figura 33: Função de autocorrelação no <i>dataset</i> Singapore.....	102
Figura 34: Decomposição clássica da ST para o <i>dataset</i> Singapore.....	103
Figura 35: Amostras dos dados de consumo elétrico do <i>dataset</i> Tetouan-Zones. ...	104
Figura 36: <i>Boxplots</i> de consumo de energia elétrica do <i>dataset</i> Tetouan-Zones. ...	104
Figura 37: Função de autocorrelação no <i>dataset</i> Tetouan-Zones.	105
Figura 38: Decomposição clássica da ST para o <i>dataset</i> Tetouan-Zones.....	105
Figura 39: Resumo da Análise Exploratória nos quatro <i>datasets</i>	106
Figura 40: <i>Boxplots</i> dos resultados de RMSE obtidos para o <i>dataset</i> UCI-Household.	107
Figura 41: <i>Boxplots</i> dos resultados de RMSE obtidos para o <i>dataset</i> LABIC-Building.	107
Figura 42: Resultado da execução da rede LSTM (observações do <i>dataset</i> LABIC-Building não usado para treinamento - agosto de 2019).....	109
Figura 43: <i>Boxplots</i> do RMSE obtido com os modelos LSTM, XGBoost e RF para o UCI-Household.....	111
Figura 44: <i>Boxplots</i> do RMSE obtido com os modelos LSTM, XGBoost e RF para o LABIC-Building.	114
Figura 45: Funcionamento do módulo com rede LSTM.	116
Figura 46: <i>Boxplots</i> do NRMSE da LSTM e BiLSTM para o <i>dataset</i> UCI-Household.	119
Figura 47: <i>Boxplots</i> do NRMSE obtido com LSTM e BiLSTM para o <i>dataset</i> LABIC-Building.....	121
Figura 48: <i>Boxplots</i> do NRMSE obtido com os modelos LSTM e BiLSTM para o <i>dataset</i> Singapore.	122
Figura 49: <i>Boxplots</i> do NRMSE obtido com os modelos LSTM e BiLSTM para o <i>dataset</i> Tetouan-Zones.	124
Figura 50: Tempo de execução em segundos da LSTM e BiLSTM para todos os <i>datasets</i>	126
Figura 51: Resultado da predição com LSTM e BiLSTM para o <i>dataset</i> UCI-Household.	127
Figura 52: Resultado da predição com LSTM e BiLSTM para o <i>dataset</i> LABIC-Building.....	128
Figura 53: Resultado da predição com LSTM e BiLSTM para o <i>dataset</i> Singapore.	128

Figura 54: Resultado da predição com LSTM e BiLSTM para o <i>dataset</i> Tetouan-Zones.....	129
Figura 50: Gráfico de dados de consumo recebidos pelo EnergySaver em tempo real.....	155

LISTA DE TABELAS E QUADROS

Quadro 1: Fases para elaboração da Revisão Sistemática	28
Tabela 1: Resultado da seleção obtida em cada base de dados.....	32
Tabela 2: Síntese dos artigos selecionados para a revisão sistemática.....	33
Tabela 3: Principais hiperparâmetros de uma RNA e sua função.....	56
Tabela 4: Principais hiperparâmetros do XGBoost e sua função.....	70
Tabela 5: Principais hiperparâmetros do RF e sua função.....	72
Tabela 6: Configurações das redes LSTM e BiLSTM.	86
Tabela 7: Configurações do XGBoost e RF.....	87
Tabela 8: Métricas médias obtidas pelos modelos de rede em cada <i>dataset</i>	108
Tabela 9: Resultados da rede LSTM para o <i>dataset</i> UCI-Household.	110
Tabela 10: Resultados do XGBoost para o <i>dataset</i> UCI-Household.....	110
Tabela 11: Resultados do RF para o <i>dataset</i> UCI-Household.	111
Tabela 12: Resultados da LSTM para o <i>dataset</i> LABIC-Building.....	112
Tabela 13: Resultados do XGBoost para o <i>dataset</i> LABIC-Building.....	113
Tabela 14: Resultados do RF para o <i>dataset</i> LABIC-Building.	113
Tabela 15: Teste <i>post-hoc</i> de Dunn para o <i>dataset</i> LABIC-Building.....	114
Tabela 16: Tabela consolidada com pontuações médias de teste dos algoritmos LSTM, XGBoost e RF para os <i>datasets</i> UCI-Household e LABIC-Building.	115
Tabela 17: Resultados da rede LSTM para o <i>dataset</i> UCI-Household.	118
Tabela 18: Resultados da rede BiLSTM para o <i>dataset</i> UCI-Household.....	119
Tabela 19: Resultados da LSTM para o <i>dataset</i> LABIC-Building.....	120
Tabela 20: Resultados da BiLSTM para o <i>dataset</i> LABIC-Building.	120
Tabela 21: Resultados da LSTM para o <i>dataset</i> Singapore.....	121
Tabela 22: Resultados da BiLSTM para o <i>dataset</i> Singapore.	122
Tabela 23: Resultados da LSTM para o <i>dataset</i> Tetouan-Zones.....	123
Tabela 24: Resultados da BiLSTM para o <i>dataset</i> Tetouan-Zones.	123
Tabela 25: Tabela consolidada com pontuações médias das métricas RMSE e R ² <i>score</i> obtidas no testes para todos algoritmos e <i>datasets</i>	124
Tabela 26: Tempo de execução em horas da LSTM e BiLSTM para todos os <i>datasets</i>	125
Quadro 2: Resultados Alcançados.....	129

LISTA DE ABREVIATURAS, SIGLAS E SÍMBOLOS

ACF	<i>Autocorrelation Function</i>
AM	Aprendizagem de Máquina
AP	Aprendizagem Profunda
ARIMA	<i>Autoregressive Integrated Moving Average</i>
AVAC	Aquecimento, Ventilação e Ar Condicionado
BiLSTM	Long Short-Term Memory Bidirecional
CNN	<i>Convolutional Neural Network</i>
CV	<i>Cross-Validation</i>
DS	<i>Design Science</i>
DSR	<i>Design Science Research</i>
EPE	Empresa Brasileira de Pesquisa Energética
GPU	<i>Graphics Processing Unit</i>
GRU	<i>Gated Recurrent Unit</i>
IA	Inteligência Artificial
IoT	<i>Internet of Things</i>
IP	Intervalo de Previsão
LabIC	Laboratório de Inteligência Computacional
LGBM	<i>Light Gradient Boosting Machine</i>
LSTM	<i>Long Short-Term Memory</i>
MAE	<i>Mean Average Error</i>
MAPE	<i>Mean Absolute Percentage Error</i>
MLP	<i>Multi-layer Perceptron</i>
MPC	<i>Model Predictive Control</i>
NRMSE	<i>Normalized Root Mean Square Error</i>
RF	<i>Random Forest</i>
RMSE	<i>Root Mean Square Error</i>
RN	Rede Neural
RNA	Rede Neural Artificial
RNP	Redes Neurais Profundas
RNR	Rede Neural Recorrente
SARIMA	<i>Seasonal Autoregressive Integrated Moving Average</i>
ST	Série Temporal
SVM	<i>Support Vector Machine</i>
UFOPA	Universidade Federal do Oeste do Pará
UML	<i>Unified Modeling Language</i>
VC-ST	Validação Cruzada de Séries Temporais
XGBoost	<i>Extreme Gradient Boost</i>

SUMÁRIO

CAPÍTULO 1	17
1 INTRODUÇÃO	17
1.1 Séries temporais e previsão	20
1.2 Problematização e Justificativa	22
1.3 Questão da Pesquisa	23
1.4 Objetivo Geral.....	23
1.5 Objetivos Específicos	23
1.6 Resultados Esperados	24
1.7 Relevância, Interdisciplinaridade e Originalidade da Tese	24
1.8 Visão Geral dos Capítulos.....	25
CAPÍTULO 2	27
2 REVISÃO SISTEMÁTICA DA LITERATURA	27
2.1 Metodologia da revisão sistemática	27
2.2 Resultado da revisão sistemática	32
2.3 Principais trabalhos relacionados	36
CAPÍTULO 3	42
3 FUNDAMENTAÇÃO TEÓRICA	42
3.1 Inteligência Artificial.....	42
3.2 Aprendizagem Profunda.....	46
3.3 Redes Neurais Artificiais	51
3.3.1 Treinamento de RNAs	53
3.3.2 <i>Underfitting e Overfitting</i>	54
3.3.3 Configuração de Hiperparâmetros em Redes Neurais	55
3.3.4 Funções de Ativação de uma RNA	57
3.4 Redes Neurais Recorrentes	59
3.5 Redes Neurais LSTM	61
3.6 Redes Neurais BiLSTM.....	65
3.7 Algoritmo XGBoost.....	68
3.8 Algoritmo Random Forest	71
3.9 IoT e Redes neurais profundas aplicadas a sistema de monitoramento de consumo de energia elétrica.....	73
3.10 O Sistema EnergySaver	74

3.10.1 Esquema módulo de predição proposto para o EnergySaver	76
CAPÍTULO 4	78
4 METODOLOGIA PROPOSTA.....	78
4.1 Caracterização da Pesquisa	78
4.2 Abordagem Metodológica	78
4.3 Fluxo Metodológico	80
4.4 Estudo de caso e <i>Datasets</i>	82
4.4.1 <i>Dataset</i> UCI-Household.....	82
4.4.2 <i>Dataset</i> LABIC-Building.....	83
4.4.3 <i>Dataset</i> Singapore	83
4.4.4 <i>Dataset</i> Tetouan-Zones	84
4.5 Decomposição Automática de Séries Temporais	84
4.6 Treinamento e Testes	86
4.6.1 Configurações do computador e dos algoritmos.....	86
4.6.2 Janela Deslizante	87
4.6.3 Validação cruzada de séries temporais	89
4.7 Pré-processamento dos dados	90
4.8 Métricas de desempenho para avaliação dos resultados.....	92
4.9 Projeto computacional.....	93
4.10 Análise estatística.....	93
4.10.1 Análise estatística para os experimentos preliminares	93
4.10.2 Análise estatística para os experimentos com redes neurais LSTM e BiLSTM.....	94
CAPÍTULO 5	96
5 RESULTADOS	96
5.1 Análise Exploratória de Dados	96
5.1.1 <i>Dataset</i> UCI-Household.....	96
5.1.2 <i>Dataset</i> LABIC-Building.....	99
5.1.3 <i>Dataset</i> Singapore	101
5.1.4 <i>Dataset</i> Tetouan-Zones	103
5.1.5 Resumo da Análise Exploratória nos quatro <i>datasets</i>	106
5.2 Resultados preliminares para dois <i>datasets</i>	106
5.2.1 Análise do número de camadas da rede LSTM.....	107
5.2.2 Predição da rede LSTM em dados não usados para treinamento ...	108

5.3	Resultados entre LSTM, XGBoost e RF com dados de validação	109
5.3.1	Treinamento de Teste com o <i>dataset</i> UCI-Household	109
5.3.2	Treinamento de Teste com o <i>dataset</i> LABIC-Building	112
5.4	Implantação do módulo de predição com LSTM para o EnergySaver	116
5.5	Comparação estatística entre as redes neurais LSTM e BiLSTM com dados de validação	118
5.5.1	Dados de validação com o <i>dataset</i> UCI-Household	118
5.5.2	Dados de validação com o <i>dataset</i> LABIC-Building.....	119
5.5.3	Dados de validação com o <i>dataset</i> Singapore.....	121
5.5.4	Dados de validação com o <i>dataset</i> Tetouan-Zones.....	122
5.5.5	Resumo dos Resultados validação em todos os <i>datasets</i>	124
5.5.6	Análise Estatística dos Resultados obtidos pelas redes LSTM e BiLSTM.....	125
5.5.7	Tempo de execução dos algoritmos por <i>dataset</i>	125
5.5.8	Predição com as redes LSTM e BiLSTM em dados não usados para treinamento para todos os <i>datasets</i>	127
5.6	Resultados Científicos, Técnicos e Acadêmicos	129
CAPÍTULO 6	131
6 DISCUSSÕES	131
6.1	O módulo de Predição com a rede LSTM.....	131
6.2	Comparação entre modelos LSTM	132
6.3	Comparação com outros algoritmos de AM.....	133
6.4	Comparação entre LSTM e BiLSTM	134
6.5	Propostas para o uso Eficiente de Energia Elétrica.....	135
CAPÍTULO 7	138
7 CONCLUSÃO E TRABALHOS FUTUROS	138
7.1	Trabalhos Futuros	141
REFERÊNCIAS	142
APÊNDICE A – IMPLANTAÇÃO DE SISTEMAS COM IOT E AP	154

CAPÍTULO 1

1 INTRODUÇÃO

A energia elétrica tem grande importância para a vida das pessoas, empresas, instituições e está diretamente ligada ao desenvolvimento dos países. Nos dias atuais, é indiscutível a extrema dependência deste recurso, uma vez que o desempenho de diversas atividades econômicas e o bem-estar social estão diretamente ligados ao consumo de energia elétrica. Porém, a conquista de uma boa qualidade de vida não deve comprometer a integridade do planeta, no sentido de manter tanto a comodidade social adquirida quanto minimizar os danos aos recursos naturais, valendo-se de ações que promovam a eficiência energética, tornando perfeitamente viável obter a economia de energia sem reduzir o conforto, o bem-estar e a segurança da comunidade envolvida no processo.

Para Irrek e Thomas (2008), uma importante definição de eficiência energética surgiu a partir de um debate realizado no Wuppertal Institut¹, onde a mesma foi definida como a relação entre o benefício obtido e a energia utilizada, ou seja, a eficiência energética será maior quando, para uma mesma quantidade de energia tivermos mais benefícios, ou ainda, se pudermos desempenhar a mesma atividade com menor consumo de energia. No mesmo sentido, Gardner e Stern (2002), apontam que esse conceito refere-se a adoção de uma tecnologia específica que visa reduzir o consumo de energia sem alterar o comportamento em causa. Conforme a ISO 50001 (2011), é a relação quantitativa entre uma saída de desempenho, serviços, mercadorias ou energia e uma entrada de energia.

Outro problema relacionado a energia elétrica que também demanda o uso de tecnologias e previsão são os surtos elétricos, que de acordo com a fabricante de DPS CLAMPER², trata-se de uma onda transitória de tensão, corrente ou potência caracterizada por uma elevada taxa de variação durante um período muito curto de tempo. Leite e Leite (2011), destacam que os surtos elétricos podem ocorrer em qualquer modalidade de sistema elétrico, seja pela queda de raios ou por possíveis manobras de rede determinadas pelas concessionárias de fornecimento de energia

¹ <https://wupperinst.org/>

² <https://clamper.com.br/>

elétrica, e até mesmo dentro da própria rede local devido ao uso de motores como por exemplo da máquina de lavar roupas. Segundo dados do Grupo de Eletricidade Atmosférica³, as descargas atmosféricas de raios, causam um prejuízo de cerca US\$1 bilhão anual à economia brasileira, tanto para o setor elétrico, que acumula as maiores perdas, quanto os setores de seguro, eletroeletrônicos, construção civil, aviação, agricultura, dentre outros.

Para se ter uma ideia da abrangência e importância do tema, de acordo com a Empresa Brasileira de Pesquisa Energética (EPE)⁴, em sua publicação sobre o balanço energético nacional até 2020, os edifícios industriais, comerciais, residenciais e públicos foram as principais demandas de energia elétrica do país, sendo responsáveis por o consumo de cerca de 79,4% do total de energia consumida no país (EPE, 2020). Outro fator que deve ser observado é o desperdício, pois de acordo com uma pesquisa realizada pela Associação Brasileira das Empresas de Serviços de Conservação de Energia (ABESCO) em 2018, somente nos últimos três anos foram desperdiçados cerca de R\$52,17 bilhões de Reais em energia elétrica, que equivalem a metade da produção de energia elétrica de Hidrelétrica de Itaipu no mesmo período (ABESCO, 2018).

Diversos estudos apontam que a redução do consumo de energia pode ser obtido por meio ações concretas que não afetem a qualidade de vida das populações, como por exemplo, medidas de racionalização, estratégias e ações conducentes à adoção de comportamentos para reduzir o desperdício de energia, bem como as novas tecnologias (FREITAS *et al.*, 2011; RODRIGUES *et al.*, 2009).

Dentre várias destas tecnologias que buscam viabilizar o uso eficiente da energia elétrica, destaca-se o uso de sistemas baseados em *Internet of Things* (IoT). Segundo Serpanos e Wolf (2018), a IoT se caracteriza por fazer uso de tecnologias heterogêneas e, à medida que os dispositivos vão sendo inseridos, novos requisitos de escalabilidade, interoperabilidade e conectividade são adicionados. Para Wong e Kim (2017) a IoT é um novo ecossistema industrial que combina máquinas inteligentes e autônomas, análise preditiva avançada e colaboração homem-máquina para melhorar a produtividade e a eficiência.

³ <http://www.inpe.br/webelat/homepage/>

⁴ <https://www.epe.gov.br/>

Outro conceito importante a ser destacado que pode ser aplicado ao uso eficiente da energia elétrica em conjunto com a IoT é a AP (LECUN *et al.*, 2015; CHOLLET, 2018). AP é um conjunto de técnicas de Inteligência Artificial (IA) que utiliza principalmente as RNPs, com muitas camadas e unidades (neurônios) entre a camada de entrada e saída, processadas com hardware de computação de alto desempenho (como unidades de processamento gráfico).

Atualmente, os modelos de AP têm sido usados para tarefas como classificação e segmentação de imagens, previsão, reconhecimento de fala e detecção de objetos. Vários sistemas usam AP, como por exemplo a Siri (assistente virtual da Apple) e Cortana (assistente virtual da Microsoft). No presente trabalho, as redes neurais LSTM (proposta por Hochreiter e Schmidhuber em 1997) e a BiLSTM (proposta por Graves *et al.*, 2009; Graves e Schmidhuber, 2005, 2009) são investigadas para previsão de consumo dentro de um framework IoT. Tais RNPs são capazes de aprender dependências de curto e longo prazo, sendo comumente utilizadas para processamento e previsão de Séries Temporais (STs).

Dentre diversos exemplos de novas tecnologias aplicadas a eficiência energética, destaca-se a abordagem de Aswani *et al.* (2012), onde os autores apresentaram um modelo de sistema para gerenciamento de ar-condicionado visando proporcionar um uso eficiente da energia elétrica através da redução do seu consumo. O sistema proposto funciona da seguinte forma: microcontroladores com sensores são usados para obter dados, que são armazenados em um servidor local em sMAP (*Simple Measurement and Actuation Profile*). Um computador de controle acessa a internet para obter previsões meteorológicas atualizadas, bem como dados obtidos dos sensores da sala, em seguida executa um esquema MPC, baseado em AP que calcula uma entrada de controle, a qual é enviada do servidor local para o termostato. Este, por sua vez, transmite um sinal correspondente ao equipamento ajustando assim a temperatura ambiente. Os resultados dos experimentos mostraram que houve uma redução de 30% a 70% no consumo de energia, mantendo uma temperatura ambiente confortável. A economia de energia se deve ao controle compensando a ocupação variável, considerando o consumo elétrico transitório e estável.

Outro exemplo foi a proposta de um modelo baseado em IoT por Ruano *et al.* (2018), a partir de uma abordagem para prototipagem e validação de dois componentes de um sistema integrado, como parte de uma solução para controle preditivo de sistemas de aquecimento, ventilação e ar condicionado (*Heating,*

Ventilation and Air Conditioning - HVAC). Eles apresentaram os resultados de uma implementação e teste de sensores sem fio autoalimentados (*self-powered wireless sensors* - SPWS), projetados especificamente para aplicações de gerenciamento de energia residencial, bem como o uso de uma plataforma IoT e seu uso em uma aplicação real. Os resultados apontaram que o uso de pequenos dispositivos, permite integração perfeita em aplicações de eficiência energética predial, além de fácil configuração, instalação e baixo custo em comparação com as alternativas disponíveis, fornecendo alertas para diagnósticos e, o mais importante, permitindo operação autônoma perpétua em condições razoáveis e em condições de iluminação comuns em edifícios e residências.

No mesmo sentido, os autores Lee *et al.* (2019), apresentaram a proposta de um sistema de previsão de consumo de energia baseado em rede neural LSTM que utiliza computação de borda (*Edge Computing*). Para isso, ao longo de quatro meses foram coletados dados reais de uso de energia em ambiente de escritório dentro do edifício (treinamento e verificação) e a análise foi conduzida durante dois meses (testes para verificar o modelo preditivo). Foram implementados dois modelos de previsão, sendo um para previsão de 1 hora e o outro para 1 dia. Como resultados a predição de consumo por dia apresentou melhor desempenho que o consumo por hora. Por fim, conclui-se que além de obter o erro médio de 27% na precisão do modelo, os usuários também puderam detectar anormalidades e fornecer serviço de análise do padrão de consumo de energia.

Desse modo, novas tecnologias de monitoramento e predição têm obtido resultados satisfatórios em busca de formas que auxiliem na promoção da eficiência energética, em particular para análise de séries temporais.

1.1 Séries temporais e previsão

De acordo com Frank *et al.* (2001), uma ST é uma sequência de vetores $\mathbf{x}(t)$, $t = 0, 1, \dots$, onde t representa o tempo decorrido. Para uma ST multivariada $\mathbf{x}(t)$ conterà valores de consumo de energia, temperatura, umidade, etc. Para uma ST univariada será considerada uma série de observações de uma variável x ordenadas pelo tempo t . Na prática, para qualquer sistema físico, x será amostrado para fornecer uma série de pontos de dados discretos, igualmente espaçados no tempo.

Para Morettin e Tolo (2006), uma ST é qualquer coleção de observações ordenadas no tempo, em que a sequência dos dados é fundamental. Assim sendo, o estudo do comportamento de uma variável ao longo do tempo por meio de uma ST torna-se importante. Ao longo dos anos, a previsão de ST de energia de curto e longo prazo tornou-se uma atividade fundamental para indústrias e políticas públicas. Nesse sentido, Hong (2013) discute as origens da previsão de eletricidade, bem como seu presente e futuro.

Hippert *et al.* (2001) em seu artigo afirmaram que “os relatórios sobre o desempenho de Redes Neurais (RNs) na previsão não convenceram inteiramente os pesquisadores nesta área” e revisaram 40 artigos sobre a aplicação de RNs na previsão de consumo de curto prazo. Vale lembrar que, naquela época, RNs como o *Multi-layer Perceptron* (MLP) convencional e o mapa de Kohonen (HAYKIN, 1998), estavam em seu apogeu em diversas áreas de pesquisa. No entanto, Hippert *et al.* (2001) concluem que mais pesquisas com padrões mais rigorosos eram necessárias para uma discussão mais sólida. Para a previsão do consumo de energia em edifícios, Zhao e Magoulès (2012) revisaram os artigos com as técnicas utilizadas, classificando-as em métodos de engenharia (que calcula a dinâmica térmica de acordo com princípios da física), métodos estatísticos (principalmente métodos de regressão que correlacionam o consumo de energia ou índice de energia com outras variáveis) e métodos de AM como RNs e *Support Vector Machine* (SVM).

Hong *et al.* (2020) revisaram trabalhos mais recentes sobre domínios de previsão de energia, como sistemas de energia eólica e solar, bem como preços e demandas de eletricidade. Os autores também discutiram as fronteiras de pesquisa em previsão de energia que são a combinação de previsão de conjunto, previsão hierárquica, previsão probabilística, bem como técnicas de Inteligência Artificial (IA) e Aprendizagem de Máquina (AM), que são nossos focos no presente trabalho.

Arjomandi-Nezhad *et al.* (2022) e Al Mamun *et al.* (2020) adotam as seguintes categorias de previsão, dependendo do horizonte de tempo: previsão de consumo de muito curto prazo, variando de alguns segundos a alguns minutos; previsão de consumo de curto prazo, variando de alguns minutos a algumas horas; previsão de consumo de médio prazo, variando de alguns dias a alguns meses; e previsão de consumo de longo prazo, variando de algumas semanas a vários anos. Mais especificamente, no domínio do presente artigo, que utiliza a previsão de consumo de curto prazo, Gross e Galiana (1987) apresentaram o estado da arte na década de

1980, quando a literatura compreendia, por exemplo, ARMA (*Auto-Regressive Moving Average*) e modelos de espaço de estados.

Com o avanço das RNPs e metodologias mais rigorosas de desenvolvimento, bem como a adoção de métodos estatísticos para avaliar o desempenho de algoritmos de AM (DEMŠAR, 2006; García *et al.*, 2010a; García *et al.*, 2010b), as redes neurais tornaram-se não apenas mais aceitáveis e confiáveis, mas também mais reprodutíveis e precisas.

Enfim, tendo em vista a problemática do monitoramento e predição em sistemas de energia elétrica, por exemplo para aumento da eficiência energética e prevenção de surtos, o conhecimento de novas técnicas se torna imprescindível. Nos próximos tópicos do presente Capítulo serão abordados a problematização e justificativa, a questão de pesquisa e os objetivos.

1.2 Problematização e Justificativa

Apesar da crescente complexidade em modelos baseados em AP para previsão de consumo, nosso foco foi a avaliação de um modelo de rede neural profunda para um framework IoT, que realiza desde a coleta de dados de consumo em equipamento individual ou um edifício, até a exibição dos dados recebidos por meio de uma página web em tempo real. Nesse sentido, avaliou-se o desempenho das redes neurais LSTM e BiLSTM em comparação a algoritmos de AM comumente utilizados na literatura, aplicadas à previsão de consumo de STs de energia, para, por fim, adicioná-la como módulo de predição em um sistema IoT. Uma das funcionalidade desse modulo é exibir uma mensagem ao usuário do sistema com o resultado da predição realizada.

Devido às recentes aplicações bem-sucedidas de algoritmos de AM baseados em árvores, escolhemos dois deles para inicialmente comparar o desempenho RNA LSTM, a saber, o XGBoost e o *Random Forest* (RF). Por exemplo, Ahmadi *et al.* (2020) compararam algoritmos baseados em árvore, incluindo XGBoost e RF para previsão de energia eólica de longo prazo, demonstrando uma aplicação bem-sucedida de tais algoritmos. Recentemente, Arjomandi-Nezdah *et al.* (2022) também compararam vários algoritmos baseados em árvore aplicados à demanda em nível de país na previsão do dia anterior da Alemanha para analisar o consumo de energia STs durante a pandemia de covid-19. Por apresentar melhores resultados que os outros

algoritmos, a LSTM foi usada para testes posteriores em comparação à BiLSTM, sendo que a BiLSTM apresentou resultados estatisticamente superiores.

1.3 Questão da Pesquisa

A partir do contexto apresentado, a presente pesquisa busca responder principalmente a seguinte questão-problema: Para realizar a previsão de STs univariadas de consumo de energia elétrica utilizando AP em um sistema IoT, qual entre as RNPs LSTM e BiLSTM apresenta melhor desempenho estatístico considerando múltiplos *datasets* simultaneamente?

1.4 Objetivo Geral

Desenvolver, testar e validar modelos de RNPs dos tipos LSTM e BiLSTM para previsão de STs de Consumo de Energia Elétrica e compará-las.

1.5 Objetivos Específicos

- a) Estudar conceitos e aplicações de RNRs dos tipos LSTM visando encontrar soluções simples para sua implementação;
- b) Propor uma metodologia para desenvolvimento, testes e validação de diferentes arquiteturas LSTM, normalizações e números de camadas;
- c) Analisar os resultados obtidos através de testes estatísticos visando verificar se há diferença estatisticamente significativa entre as arquiteturas, para determinar e validar o modelo com melhor desempenho estatístico para a previsão;
- d) Comparar estatisticamente os resultados obtidos pela LSTM com algoritmos de AM.
- e) Avaliar os resultados da LSTM em comparação a BiLSTM.
- f) Divulgar os resultados.

1.6 Resultados Esperados

a) Científicos:

- Validar um modelo de rede neural LSTM para STs;
- Adicionar o modelo de rede LSTM a um sistema IoT de coleta de dados de consumo de energia elétrica;
- Publicar artigos científicos em revistas indexadas;
- Apresentar trabalhos em congressos.

b) Técnicos:

- Desenvolver material didático com manual e tutorial de um sistema IoT contendo um módulo de rede neural LSTM.

c) Acadêmicos:

- Ministrando minicurso/Palestra sobre RNPs para previsão de STs;
- Realizar enquadramento de projetos de iniciação científica e/ou trabalhos de conclusão de curso em subáreas relacionadas ao projeto.

1.7 Relevância, Interdisciplinaridade e Originalidade da Tese

A originalidade deste trabalho refere-se à aplicação das RNPs LSTM e BiLSTM à previsão de STs de eletricidade, bem como a avaliação de seu desempenho através de testes estatísticos em quatro *datasets*, com dados reais de consumo de eletricidade utilizando VC-ST, possibilitando a integração de diversos conhecimentos envolvidos para monitoramento de energia em indústrias, estabelecimentos comerciais e prédios públicos, considerando o uso da AP, visto que não há sistema similar sendo desenvolvido na região amazônica de acordo com a pesquisa bibliográfica realizada. Além disso, os resultados atuais da LSTM e BiLSTM formarão uma linha de base para testes futuros, com arquiteturas mais complexas para sistemas IoT.

Desse modo, o presente trabalho difere dos demais artigos da literatura (descritos no Capítulo 2, Seção 2.3), justamente porque investigou-se as RNPs LSTM e BiLSTM para realizar a predição de consumo usando diferentes *datasets* em

diferentes escalas e características, além de acoplá-la como módulo em um sistema IoT que realiza o monitoramento de consumo de energia elétrica em tempo real (em um equipamento individual ou um edifício, por exemplo).

Assim, as principais contribuições deste trabalho são: (i) análise estatística do desempenho de arquiteturas da RNP LSTM com diferentes números de camadas em diferentes *datasets*; (ii) previsão de STs de dados reais coletados em quatro padrões de consumo diferentes: residência, edificação pública, Cidade e País; ou seja, localidades, países e hábitos de consumo diferentes. (iii) previsão de STs de consumo de energia elétrica com dados reais coletados em uma cidade da Amazônia brasileira, com características climáticas e meteorológicas regionais, uma vez que estudos desse tipo nesta região são escassos; (iv) testes estatísticos utilizando VC-ST para comparar os resultados obtidos pela RNP LSTM em relação aos algoritmos XGBoost e RF; (v) Comparação estatística entre as RNPs LSTM e BiLSTM, com dados de validação usando VC-ST.

1.8 Visão Geral dos Capítulos

Este trabalho está organizado em 7 capítulos, conforme descrito a seguir:

Capítulo 1 – Introdução. Apresenta-se uma abordagem geral do trabalho, a descrição dos resultados esperados, bem como a relevância, interdisciplinaridade e originalidade da Tese.

Capítulo 2 - Trabalhos Relacionados. São mencionados os principais trabalhos encontrados na literatura atual que também procuram lidar com o problema de predição de STs e possuem características relacionadas com o que é apresentado neste trabalho.

Capítulo 3 - Fundamentação Teórica. Apresenta-se uma visão geral dos conceitos teóricos em quais este trabalho se baseia, como Inteligência Artificial (IA), AP, Rede Neurais artificiais (RNAs), RNRs, LSTM e BiLSTM. Os algoritmos de AM XGBoost e RF, que foram utilizados para comparação. Aborda-se também o uso de sistemas IoT com RNPs, além de uma visão geral do sistema IoT EnergySaver e a proposta de acoplamento do módulo de RNP ao sistema IoT.

Capítulo 4 – Metodologia Proposta. Detalha-se a caracterização da pesquisa, o procedimento metodológico baseado na metodologia *Design Science Research* (DSR), o fluxo metodológico proposto, o estudo de caso e *datasets*, treinamento e

testes, pré-processamento dos dados, métricas de desempenho para avaliar os resultados, o projeto experimental e a análise estatística.

Capítulo 5 – Resultados. É apresentada a análise exploratória dos dados; os resultados preliminares para dois *datasets*; os resultados entre LSTM, XGBoost e RF com dados de validação; comparação estatística entre as redes neurais LSTM e BiLSTM com dados de validação; e os resultados científicos, técnicos e acadêmicos.

Capítulo 6 - Discussões. São discutidos detalhes sobre a implantação de um módulo de predição de STs para um sistema IoT. Em seguida os testes realizados se subdividiram entre números de camadas para a rede LSTM, algoritmos baseados em AM em relação a LSTM e comparação entre a LSTM e BiLSTM; além destas discussões, exemplificamos propostas para uso eficiente de energia elétrica

Capítulo 7 – Conclusão e Trabalhos Futuros. São apresentadas as conclusões obtidas a partir do desenvolvimento da pesquisa e os trabalhos futuros.

Finaliza-se com as Referências Bibliográficas.

CAPÍTULO 2

2 REVISÃO SISTEMÁTICA DA LITERATURA

De acordo com Boote e Beile (2004), o levantamento do estado da arte por meio da revisão sistemática da literatura é atividade obrigatória para realizar pesquisas científicas de qualidade, que abrange desde o delineamento de um novo projeto de pesquisa até a escrita uma Monografia, Dissertação, Tese ou Artigo Científico. Para isso, os pesquisadores devem realizar um levantamento da literatura atual a respeito de determinado objeto de pesquisa, tendo em vista que a qualidade de uma pesquisa é diretamente proporcional à qualidade do processo de condução do levantamento do estado da arte.

Neste sentido, optou-se por utilizar esta metodologia de revisão sistemática visando identificar um conjunto de estudos da literatura atual que abordam os conceitos envolvidos no escopo da presente pesquisa.

2.1 Metodologia da revisão sistemática

Para executar a revisão sistemática da literatura, deve-se inicialmente definir um protocolo de pesquisa. Assim sendo, optou-se por utilizar a abordagem proposta por Kitchenham e Charters (2007), que apresentam um protocolo com diretrizes amplamente utilizadas em pesquisas na área da computação, sendo agrupadas em três fases principais: 1) planejamento: define a questão da pesquisa; 2) condução: busca e seleção dos estudos primários, avaliação de qualidade, extração dos dados, sumarização e síntese dos resultados e, ainda, interpretação dos resultados; 3) relatório: escrita do relatório/artigo (publicação dos resultados).

Assim sendo, para o escopo deste trabalho, o protocolo proposto por Kitchenham e Charters (2007) foi adaptado conforme Quadro 1 a seguir.

Quadro 1: Fases para elaboração da Revisão Sistemática.

Protocolo	Processo
Planejamento	Definição da questão da pesquisa que foi elaborada com base na problematização e justificativa descrita na Seção 1.2.
Condução	<p>Busca e seleção dos estudos primários: Por meio de bases digitais e motores de busca comuns da área de estudo: <i>IEEEEXPLORE</i>, <i>SCIENCEDIRECT</i>, e <i>Google Acadêmico</i>. Utilizou-se também buscas manuais na <i>web</i>.</p> <p>Avaliação de qualidade: Foram considerados os critérios de inclusão e exclusão, além da identificação de pesquisas experimentais, com a presença de variáveis pertencentes à questão de pesquisa.</p> <p>Extração dos dados: A partir da análise dos resultados das abordagens.</p> <p>Sumarização e síntese dos resultados: Foram identificadas as pesquisas experimentais com palavras-chave pertencentes à questão de pesquisa e elaborada uma tabela com a síntese dos artigos selecionados, contendo: ano de publicação, modelos e principais contribuições.</p> <p>Interpretação dos resultados: Análise das principais abordagens que realizam previsão de STs bem como sistemas baseados em IoT.</p>
Relatório	Estruturação do relatório em formato de artigo científico conforme normas específicas de cada revista/congresso (Descrito no Capítulo 7, em resultados alcançados).

Fonte: Elaboração própria, 2022.

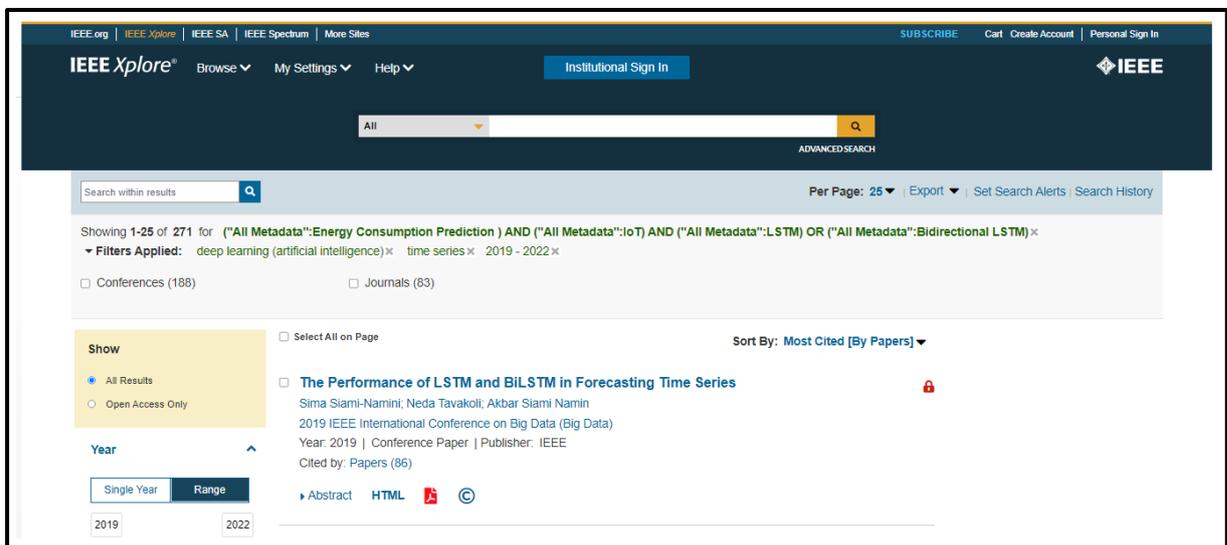
Para fazer a seleção de uma publicação é essencial levar em consideração a sua relevância, ou seja o seu potencial como fonte de estudo primário ou secundário. Assim sendo, para as questões de investigação, serão considerados inicialmente os pontos referentes a análise do título, resumo, palavras-chave, metodologia e conclusão, da seguinte forma:

- 1) O primeiro filtro teve como objetivo fazer a seleção preliminar das publicações, por meio da aplicação das expressões de busca “*Energy Consumption Prediction AND LSTM OR Bidirectional LSTM*”, em três bases de dados, bem como uso de filtros específicos de cada base;
- 2) No segundo filtro, levou-se em consideração a análise de critérios de inclusão e exclusão estabelecidos nos subitens abaixo:
 - a) Critérios de Inclusão: (i) publicações que abordem primária ou secundariamente estudos relacionados a predição de consumo de energia elétrica; (ii) publicações que utilizam redes neurais LSTM e BiLSTM; (iii) publicações que em sua abordagem utilizem aplicações de redes neurais para sistemas IoT.

- b) Critérios de Exclusão: (i) publicações cujas palavras-chaves estão ausentes da publicação e que não há variação destas palavras-chaves (exceto os casos de plural); (ii) publicações em que as palavras-chaves não aparecem no título, resumo, e/ou texto da publicação; (iii) publicações que abordam o tema de maneira superficial; (iv) publicações que estejam fora do intervalo dos períodos referentes aos anos de 2019 a 2022 para as três bases de dados e para busca manual.
- 3) Como terceiro filtro, levou-se em consideração a quantidade de citações na literatura, além da leitura dos resumos/abstract, metodologia e conclusão de cada artigo selecionado na etapa anterior.

A primeira base de dados para realização do levantamento inicial da literatura foi a IEEEXPLORE, realizado no dia 10 de Outubro de 2022, onde foram encontrados 271 (duzentos e setenta e um) trabalhos. Também foram aplicados os seguintes filtros adicionais: *deep learning (artificial intelligence)*, *time series*, ano 2019-2022 (Figura 1).

Figura 1: Resultado da string de busca.



Fonte: www.ieeeexplore.ieee.org

A segunda base de dados foi a SCIENTIFIEDIRECT, realizada no dia 10 de Outubro de 2022, onde foram encontrados 132 (cento e trinta e dois) trabalhos. Também foram aplicados os seguintes filtros adicionais: *Article type: Research articles*, *Publication title: Applied Energy, Energy, Neural Networks*. *Subject areas: Energy*. Ano 2019 a 2022 (Figura 2).

Figura 2: Resultado da string de busca.

The screenshot shows the ScienceDirect search interface. At the top, there is a search bar with the query "Energy Consumption Prediction AND IoT AND LSTM OR Bidirect". Below the search bar, it indicates "132 results" and "sorted by relevance | date". On the left side, there are filters for "Refine by:", "Years" (2022 (71), 2021 (39), 2020 (17)), and "Article type" (Research articles (132)). The main content area displays two research articles:

- Research article**: A multi-step predictive deep reinforcement learning algorithm for HVAC control systems in smart buildings. *Energy*, 20 July 2022, ... Authors: Xiangfei Liu, Mifeng Ren, ... Chengke Wu.
- Research article**: Intelligent autonomous street lighting system based on weather forecast using LSTM. *Energy*, 12 May 2021, ... Authors: Didar Tukymbekov, Ahmet Saymbetov, ... Yeldos Svanbayev.

Fonte: www.sciencedirect.com

A terceira base de dados foi o Google Acadêmico, realizado no dia 10 de Outubro de 2022, onde foram encontrados 475 (quatrocentos e setenta e cinco) trabalhos. Também foram aplicados os seguintes filtros adicionais: ordenar por relevância, em qualquer idioma, qualquer tipo, incluir citações. Ano 2019 - 2022 (Figura 3):

Figura 3: Resultado da string de busca.

The screenshot shows the Google Acadêmico search interface. The search bar contains the query "Energy Consumption Prediction" AND "IoT" AND "LSTM" OR "Bidirectional L". It indicates "Aproximadamente 475 resultados (0,05 s)". On the left side, there are filters for "Artigos", "Período específico..." (2019 - 2022), "Pesquisar", "Ordenar por relevância", "Ordenar por data", "Em qualquer idioma", "Pesquisar páginas em Português", "Qualquer tipo", and "Artigos de revisão". The main content area displays four search results:

- [HTML] Performance evaluation of LSTM neural networks for consumption prediction**. DG da Silva, MTB Geller, MS dos Santos Moura... - e-Prime-Advances in ..., 2022 - Elsevier. ... -Term Memory (LSTM) neural networks in time series electric energy consumption prediction, for a forecasting module of an IoT system. Three time series were used and we compared ...
- Improved LSTM data analysis system for IoT-based smart classroom**. Y Li, L Chen - Journal of Intelligent & Fuzzy Systems, 2020 - content.iospress.com. ... Since the acquired classroom equipment data is a time series, LSTM is used here for energy consumption prediction. Finally, using data visualization, the data is displayed in the form of ...
- Short-term prediction of residential power energy consumption via CNN and multi-layer bi-directional LSTM networks**. FUM Ullah, A Ullah, IU Haq, S Rho, SW Baik - IEEE Access, 2019 - ieeeexplore.ieee.org. ... State-of-the-art Energy Consumption Prediction (ECP) methods are limited in terms of ...-constrained devices for the Internet of Things (IoT) environment by reducing the cost and compu...
- A two-layer water demand prediction system in urban areas based on micro-services and LSTM neural networks**.

Fonte: www.scholar.google.com.br.

Nas buscas manuais na *web* foram selecionados, inicialmente, 40 artigos que mostraram relação com a abordagem proposta e que não foram listados nos motores de busca utilizados, mas que julgou-se importante uma análise mais aprofundada sobre os mesmos. Nestas buscas também foram pesquisados alguns trabalhos que abordam o uso da predição com sistemas IoT, bem como previsão com intervalos de predição.

Assim sendo, após o levantamento inicial para seleção de trabalhos relacionados utilizando a *string* de busca, foram encontrados 894 artigos nas três bases de dados e busca manual. Em seguida, aplicou-se o segundo filtro, que está relacionado a leitura dos resumos/abstract, levando em consideração critérios de inclusão e exclusão. No critério de inclusão para cada base de dados e busca manual foram selecionados: (i) IEEXPLORE (13), (ii) SCIEDIRECT (6), (iii) Google Acadêmico (23); (iv) busca manual (10). Já no critério de exclusão, foram suprimidos os artigos: (i) com abordagens relacionadas a previsão de consumo usando técnicas sem relação com as RNPs (198); (ii) artigos que realizavam predição de consumo com RNPs que não envolvessem LSTM ou BiLSTM (290); (iii) uso de redes neurais LSTM e BiLSTM em contexto diferente de predição de consumo de energia elétrica (357).

Por fim, aplicou-se o terceiro filtro onde buscou-se selecionar, dentre os 52 artigos obtidos no segundo filtro, os artigos mais relevantes em relação a quantidade de citações na literatura, além da leitura dos resumos, metodologia e conclusão. Assim sendo, os dados foram interpretados analisando a coerência interna dos artigos, a partir da identificação dos experimentos descritos, da relação entre objetivos propostos e os resultados obtidos e da relação entre os elementos metodológicos e os dados obtidos.

Desse modo, após a aplicação do terceiro filtro, foram selecionados 12 artigos nas três bases de dados e busca manual, onde os mesmos serão detalhados na Subseção 2.3, bem como serão apresentadas as principais diferenças entre o trabalho proposto e os selecionados.

A Tabela 1, apresenta o resumo dos resultados da seleção obtida em cada base de dados, bem como a quantidade de artigos selecionados em cada filtro descrito.

Tabela 1: Resultado da seleção obtida em cada base de dados.

<i>Base de dados</i>	Total de Publicações		
	Primeiro Filtro	Segundo Filtro	Terceiro Filtro
IEEEXPLORE	271	13	3
SCIENCEDIRECT	132	6	3
Google Acadêmico	475	23	2
Busca Manual	40	10	4
Total	918	52	12

Fonte: Elaboração própria, 2022.

2.2 Resultado da revisão sistemática

A Tabela 2 mostra cronologicamente a síntese dos artigos escolhidos a partir dos critérios descritos na Seção 2.1.

Tabela 2: Síntese dos artigos selecionados para a revisão sistemática.

Autores	Ano de publicação	Modelos	Principais contribuições
<i>Kaur et al.</i>	2019	LSTM	Apresentaram um esquema unificado baseado em rede neural LSTM para gerenciamento de energia em redes inteligentes, visando analisar e extrair padrões de energia importantes que podem ser usados no gerenciamento de resposta à demanda, previsão de carga e redução de pico.
<i>Schirmer et al.</i>	2019	LSTM	Avaliaram o desempenho de diferentes métodos de regressão na tarefa de previsão de consumo de energia residencial, concluindo que a rede neural LSTM supera todos os outros algoritmos testados por eles.
<i>Mellouli et al.</i>	2019	LSTM	Avaliaram quatro arquiteturas de RNPs derivadas da arquitetura LSTM. Os resultados apontam que o modelo Stacked LSTM parece ser a arquitetura de aprendizado profundo eficiente que fornece previsões aceitáveis.
<i>Das et al.</i>	2020	LSTM, GRU e BiLSTM	Realizaram uma análise comparativa entre as redes neurais LSTM, BiLSTM e GRU, para a previsão de cargas elétricas diversas, concluindo que os modelos BiLSTM e GRU foram significativamente melhores em horizonte de predição maior.
<i>De Sousa et al.</i>	2020	MLP, ELMAN, NAR, AIW, LUBE	Utilizaram as RNAs MLP, ELMAN, NAR, AIW e LUBE com Intervalos de Previsão para prever a irradiância solar. Como resultado a LUBE foi o melhor método testado para a tomada de decisão, devido à precisão das informações de IP.
<i>Hou et al.</i>	2021	LSTM	Aplicaram a rede neural LSTM no processo de previsão de consumo residencial de um método de agregação de carga adaptativa com um número de domicílios selecionados. Os resultados mostraram que o modelo foi preciso na Predição.
<i>Rafi et al.</i>	2021	CNN-LSTM	Realizaram a previsão de consumo de curto prazo do sistema elétrico de Bangladesh, integrando os modelos LSTM e CNN, considerando as

			vantagens de cada um. O modelo CNN-LSTM foi comparado com a LSTM, rede funcional de base radial (RBFN) e XGBoost. Os resultados do modelo CNN-LSTM foram superiores aos modelos testados em todos os casos de validação.
Farsi <i>et al.</i>	2021	CNN-LSTM	Foi proposto um modelo de integração LSTM e CNN para previsão de consumo na Malásia (consumo horário) e na Alemanha (consumo diário). Segundo os autores, a rede integrada que chamaram de PLCNet, melhorou a precisão de 83,17% para 91,18% para os dados da Alemanha e 98,23% de precisão para os dados da Malásia.
Hadri <i>et al.</i>	2021	LSTM, XGBoost e SARIMA	Avaliaram os algoritmos LSTM, XGBoost e SARIMA para previsão de consumo visando descobrir qual deles têm o melhor desempenho na previsão do consumo de eletricidade online. Os resultados obtidos mostraram que nenhum deles supera eficientemente as três estratégias de previsão (ou seja, univariada, multivariada e multipasso).
Moura <i>et al.</i>	2021	WSN (<i>Wireless sensor network</i>)	Plataforma IoT baseada em infraestrutura de rede de sensores sem fio (WSN), que suporta um sistema inteligente para controlar aquecimento, ventilação e ar condicionado (HVAC).
Geller e Meneses	2021	UML para um sistema IoT/IA	Propôs o uso de extensões de <i>Unified Modeling Language</i> (UML) para modelar um sistema de monitoramento de energia de inteligência artificial para previsão de consumo de eletricidade.
Da Silva <i>et al.</i>	2021a	LSTM	Um módulo de predição com AP para o sistema EnergySaver baseado em IoT, que visa monitorar o consumo de energia elétrica, desde a captura de dados até a previsão de consumo para o mês seguinte.
Ozer <i>et al.</i>	2021	LSTM (XCORR <i>cross-correlation</i>)	Uso da rede neural LSTM em uma abordagem de aprendizagem de transferência baseada na correlação cruzada XCORR.
Da Silva <i>et al.</i>	2022	LSTM, RF e XGBoost	Avaliaram os algoritmos LSTM, RF e XGBoost, para previsão de consumo de energia elétrica em três <i>datasets</i> usando VC-ST. Como resultado, a

			LSTM apresentou o melhor desempenho de previsão em todos os testes nos <i>datasets</i> .
Fernández-Martínez e Jaramillo-Morán	2022	LSTM e GRU	Apresentaram testes com as redes LSTM e GRU para criar modelo híbrido visando prever os consumos de 24 horas (um dia inteiro à frente) de um hospital. Os resultados apontaram que ambos os modelos forneceram desempenhos semelhantes em todos os casos.
Shin e Woo	2022	LSTM, RF e XGBoost	Avaliaram os algoritmos LSTM, RF e XGBoost, para previsão de consumo de energia na Coreia. Esses algoritmos foram aplicados para previsão no Período 1 (antes do início da COVID-19) e no Período 2 (após o início da COVID-19). Como resultado, a LSTM apresentou o melhor desempenho de previsão Período 1 e RF no Período 2.
Shaqour <i>et al.</i>	2022	RNP, BI-GRU-FCL, GRU-FCL, BI-LSTM-FCL e CNN	Usam agregação de consumo de energia elétrica e as redes RNP, BI-GRU-FCL, GRU-FCL, BI-LSTM-FCL e CNN para o processo de previsão. Os resultados apontaram que a RNP alcançou a maior porcentagem de MAPE para a maioria dos níveis de agregação e o Bi-GRU-FCL indicou menores erros de RMSE.

Fonte: Elaboração própria, 2022.

Para possibilitar uma análise mais aprofundada de cada artigo selecionado, será descrito na Seção 2.3 o resumo geral de cada abordagem proposta pelos autores bem como a principal diferença deles para o nosso trabalho.

2.3 Principais trabalhos relacionados

Kaur *et al.* (2019) desenvolveram um esquema unificado baseado em redes neurais LSTM para um sistema de rede inteligente (*Smart Grid*), com o objetivo de criar uma abordagem integrada de pré-processamento e análise de dados com taxas mínimas de erro e maior precisão. O estudo de caso utilizou dados coletados de 112 casas inteligentes, onde os mesmos foram pré-processados e decompostos usando-se o modelo de decomposição de valor singular de alta ordem (*High-Order Singular Value Decomposition* - HOSVD) e, em seguida, aplicou-se a previsão com três modelos de rede, o modelo proposto RNA-LSTM, uma variante de RNAs e *Autoregressive Integrated Moving Average* (ARIMA). Para avaliação estatística, os autores utilizaram as seguintes métricas: RMSE e erro percentual médio absoluto (*Mean Absolute Percentage Error* - MAPE). Os resultados apontaram que a RNA-LSTM obteve menores erros (RMSE = 3,35 e MAPE = 5,21%), em relação à variante de RNA (RMSE = 4,613 e MAPE 17,312%) e ARIMA (RMSE = 4,27 e MAPE 29,18%).

Schirmer *et al.* (2019) avaliaram o desempenho dos métodos de regressão para previsão de consumo de energia elétrica, a saber: Regressão Linear (RL), Árvores de Decisão (AD), RNPs, Redes Neurais Recorrentes (RNR), *Gated Recurrent Unit* (GRU) e LSTM. Para isso, utilizaram o *dataset Smart Meters in London* (SMinL) que possui dados referente a 5.567 residências em Londres, medidas entre novembro 2011 e fevereiro de 2014. Experimentos foram realizados e foi demonstrado que a LSTM supera todos os outros algoritmos de regressão, melhorando a precisão do preditor e reduzindo o erro absoluto médio em até 26,7% quando comparado ao preditor de Regressão Linear.

Mellouli *et al.* (2019) apresentaram um abordagem que utiliza quatro arquiteturas de RNPs derivadas da LSTM para previsão de STs de temperatura interna e consumo de energia em uma câmara fria, que são: LSTM, *Convolutional LSTM*, *Stacked LSTM* e BiLSTM. Foi utilizado um *dataset* que possui dados de demanda elétrica coletados em uma câmara frigorífica, dividido em cinco casos de uso para treinamento e testes das redes. Os resultados dos experimentos apontaram que

a *Stacked LSTM* foi a arquitetura mais eficiente de para previsões aceitáveis no contexto dos dados utilizados.

Das *et al.* (2020) descreveram um estudo que fornece uma análise do uso da rede neural LSTM para a previsão de cargas elétricas diversas (*Miscellaneous Electric Loads - MEL*), realizando também uma comparação com os modelos BiLSTM e GRU. O *dataset* possui dados de coleta correspondente aos meses de abril a novembro de 2017, e foram coletados em uma instalação de pesquisa localizada na cidade de Abu Dhabi, EAU, local composto por oito mesas individuais, que poderiam ser ocupadas por até 6 estudantes de pós-graduação de cada vez. Os testes apontaram que mediante a análise comparativa em termos das métricas RMSE e MAE (*Mean Average Error*), os três modelos alcançaram bons resultados, dependendo dos dispositivos considerados e da duração da previsão necessária. No entanto, os autores concluíram que a rede BiLSTM provou ser o modelo mais estável dentre os três, em previsões com um dia e uma semana de antecedência.

De Sousa *et al.* (2020) utilizaram RNAs convencionais com Intervalos de Previsão (IPs) para prever irradiância solar, fornecendo uma avaliação da incerteza futura em diferentes cenários. Os IPs consistem em limites inferiores e superiores dentro dos quais um valor futuro desconhecido se situará, com uma probabilidade predeterminada, chamada de nível de confiança, tornando-se mais confiável e informativo para os tomadores de decisão, dando suporte para selecionar a melhor ação em condições incertas.

Hou *et al.* (2021) aplicaram a rede neural LSTM no processo de previsão de consumo residencial de um método de agregação de carga adaptativa com um número de domicílios selecionados. Os Erros Percentuais Médios Absolutos (*Mean Absolute Percentage Error, MAPE*) da previsão de consumo de energia elétrica foram de 14,3% e 10,2% para 50 e 100 domicílios, respectivamente. E quando o número de domicílios chega a 150, o MAPE tanto do método tradicional quanto no método proposto ficam abaixo de 0,1, o que atende ao requisito de precisão da previsão de consumo.

Rafi *et al.* (2021) realizaram a previsão de consumo de curto prazo do sistema elétrico de Bangladesh, integrando os modelos LSTM e CNN, considerando as vantagens de cada um. A questão motivadora citada pelos autores é que muitas vezes um modelo sozinho não funciona bem para previsões de consumo de energia elétrica. O modelo híbrido (CNN-LSTM) é composto por um módulo CNN, um módulo LSTM e

um módulo de fusão de recursos (*feature-fusion*). A validação do modelo foi realizada com as métricas de avaliação média do erro médio MAE, erro quadrático médio RMSE, MAPE e o coeficiente de determinação (R^2). O modelo CNN-LSTM foi comparado com abordagens LSTM, rede funcional de base radial (*Radial Basis Functional Network*, RBFN) e *Extreme Gradient Boosting* (XGBoost). De acordo com a pesquisa, os resultados do modelo CNN-LSTM foram superiores aos modelos testados em todos os casos de validação.

Um modelo de integração LSTM e CNN também foi utilizado por Farsi *et al.* (2021) para previsão de consumo na Malásia (consumo horário) e na Alemanha (consumo diário). Segundo os autores, a rede integrada que chamaram de PLCNet, melhorou a precisão de 83,17% para 91,18% para os dados da Alemanha e 98,23% de precisão para os dados da Malásia, reforçando que o modelo tem bom desempenho para ser utilizado para previsão de consumo de curto prazo.

Hadri *et al.* (2021) investigaram três abordagens principais para a previsão do consumo de eletricidade utilizando os algoritmos XGBoost, LSTM e *Seasonal Autoregressive Integrated Moving Average* (SARIMA), a fim de avaliar empiricamente a precisão da previsão e o tempo/complexidade computacional. Para isso, os autores utilizaram o *dataset* de dados de energia residencial holandesa (*Dutch Residential Energy Database - DRED*), que possui dados de consumo de eletricidade em nível agregado e em nível de eletrodomésticos (TV, geladeiras, máquinas de lavar, micro-ondas, aquecimento central, laptops). Os resultados obtidos mostraram que nenhum deles foi melhor para as três estratégias de previsão (univariada, multivariada e *multistep*). Por exemplo, o XGBoost supera os outros algoritmos para a previsão univariada e *multistep*, já o ARIMA apresentou melhor desempenho para multivariada. A LSTM apresentou o pior comportamento nas três estratégias. Desse modo, tendo em vista o XGBoost apresentar melhor desempenho em duas estratégias, ele foi escolhido para ser implantado em uma plataforma IoT para previsão de consumo de eletricidade em tempo real.

Moura *et al.* (2021) projetaram e instalaram uma plataforma IoT baseada em infraestrutura WSN, que suporta um sistema inteligente para controlar sistemas de aquecimento, ventilação e ar condicionado (AVAC), bem como iluminação em edifícios. O objetivo era encontrar opções para alcançar usos mais sustentáveis de energia, considerando a integração de fontes renováveis de geração e o uso de um sistema inteligente de controle de consumo de energia elétrica. Como resultado, a

abordagem aponta como fornecer a tecnologia certa para edifícios antigos existentes, contribuindo para um consumo de energia próximo de zero (nZEB) a baixo custo.

Geller e Meneses (2021) apresentaram algumas propostas de uso de extensões da Linguagem de modelagem unificada (*Unified Modeling Language - UML*) para modelar sistemas IoT já disponíveis na literatura (Reggio (2018), Thramboulidis e Christoulakis (2016), e Robles-Ramirez *et al.* (2017)). Um sistema de monitoramento de energia com inteligência artificial para previsão de consumo foi utilizado como estudo de caso e foi modelado com recursos UML. Foi possível concluir que os diagramas UML e suas extensões podem representar as diferentes visões de uma aplicação IoT (estática, comportamental, de segurança, etc.), motivando pesquisadores a utilizá-los, dando origem a diversas propostas.

Da Silva *et al.* (2021a) apresentaram uma abordagem para a criação de um módulo de predição com AP para o sistema EnergySaver baseado em IoT, que visa monitorar o consumo de energia elétrica desde a captura de dados até a previsão de consumo para o mês seguinte. Os resultados experimentais indicaram que o software obteve resultados satisfatórios desde a coleta de dados com IoT até a previsão de consumo utilizando um módulo com a rede neural LSTM, em testes de casos de uso preliminares.

Em Da Silva *et al.* (2022), os autores avaliaram o desempenho de redes neurais LSTM para previsão de consumo de energia elétrica em dois passos: 1) análise estatística da RNR LSTM em relação ao número de camadas, onde realizaram experimentos do modelo com 2, 4, 6 e 8 camadas, usando VC-ST em dois *datasets*, que apontou não haver diferença estatisticamente significativa entre elas, usando assim para testes posteriores a rede com 2 camadas; 2) comparação estatística entre a LSTM e os algoritmos XGBoost e RF usando VC-ST para três *datasets*. Como resultado, os testes apontaram que o modelo LSTM apresentou uma tendência de melhor desempenho do RMSE no primeiro *dataset*, e melhores resultados estatisticamente significativos nos outros dois *datasets*, de acordo com o teste de Kruskal-Wallis ($p < 0,0001$ em ambos os casos).

Ozer *et al.* (2021) usaram a rede neural LSTM em uma abordagem de aprendizagem por transferência baseada na correlação cruzada nominada XCORR. Esse modelo foi composto por seis etapas. Inicialmente, ocorreu a normalização dos dados reais de um edifício (que será estimado e que será transferido). Em seguida, o XCORR é aplicado entre os dados do edifício a estimar e os dados de cada edifício a

transferir. O próximo passo é determinar qual dos dados é mais adequado para ser usado para aprendizagem por transferência. Após isso, o modelo LSTM é treinado com os dados normalizados deste edifício. Na sequência, os pesos do modelo resultante são transferidos e retreinados na forma normalizada dos dados originais. Finalmente, os valores de desempenho do modelo são obtidos testando com os dados de teste. Os resultados obtidos com as métricas RMSE, MAPE e média do erro médio (*Mean Average Error* – MAE), mostraram que o modelo proposto usando LSTM com dados de aprendizado de transferência foi bem-sucedido em comparação com modelos que funcionam bem em pequenos *datasets*, como algoritmo Random Forest, XGBoost e *Light Gradient Boosting Machine* (LGBM).

Fernández-Martínez e Jaramillo-Morán (2022) apresentaram uma abordagem para previsão de consumo de energia em um edifício de assistência médica, utilizando redes neurais LSTM e GRU, juntamente com um algoritmo de pré-processamento de decomposição em modo empírico (*Empirical Mode Decomposition* – EMD). O *dataset* utilizado para os experimentos possui dados de consumo de energia elétrica e variáveis meteorológicas, fornecidas pela empresa Emececuadrado (Badajoz, Espanha), coletados no período de setembro de 2016 a julho de 2021. A partir da obtenção desses dados, foram conduzidos dois estudos de caso diferentes para prever o consumo horário: um cenário de previsão univariado baseado no consumo de energia ativa e outro multivariado que usa as quatro STs (potência ativa, potência reativa, temperatura e umidade). Experimentos foram realizados e os resultados alcançados apontaram que os modelos híbridos (rede neural com pré-processamento) superaram claramente os simples (somente a rede neural) em todos os casos. Apesar de ambos os modelos (LSTM e GRU) fornecerem desempenhos semelhantes em todos os casos, os melhores resultados (MAPE = 3,51% e RMSE = 55,06) foram obtidos pela rede LSTM com pré-processamento, no cenário para previsão com dados multivariados.

Shin e Woo (2022) descreveram um estudo que comparou os algoritmos XGBoost, LSTM e RF para previsão de consumo de energia elétrica na Coreia, em dois períodos da seguinte forma: (i) período 1 (antes do início da pandemia de COVID-19); (ii) período 2 (após o início da pandemia de COVID-19). A pesquisa utilizou os dados de demanda de energia entre o período de janeiro de 1996 a junho de 2021 para análise, baseado no fornecimento total de energia (*Total Energy Supply* - TES), contendo variáveis referentes ao consumo total de energia e variáveis altamente

correlacionadas (preço do petróleo, população, geração de energia, índice de produção industrial, temperatura) para confirmar a adequação e usabilidade da previsão de cada modelo testado. Além disso, ao prever o consumo total de energia, os autores separam o período de análise de mercado estável antes da pandemia de COVID-19 e no mercado instável subsequente. Os resultados obtidos a partir das métricas RMSE e MAPE mostraram que a rede LSTM realizou a previsão com mais precisão no período 1, enquanto o modelo RF tendeu a produzir os menores RMSE e MAPE no período 2. Já o XGBoost apresentou os maiores erro de previsão nas métricas.

Shaqour *et al.* (2022), usam agregação de consumo de energia elétrica e as RNPs (*Deep Neural Network- DNN*), BI-GRU-FCL (*Bi-directional Gated Recurrent Unit with fully connected layers*), GRU-FCL (*Gated Recurrent Unit with fully connected layers*), BI-LSTM-FCL (*Bi-directional Long Short-Term Memory with fully connected layers*) e CNN (*Convolutional Neural Network*) para o processo de previsão. Os resultados apontaram que a RNP alcançou a maior porcentagem de MAPE para a maioria dos níveis de agregação e o Bi-GRU-FCL indicou menores erros de RMSE, apresentando velocidade de treinamento 15% mais rápida e 40% menos parâmetros em comparação com o RNPs.

O trabalho aqui proposto difere dos demais artigos da literatura uma vez que propomos o uso das RNPs LSTM e BiLSTM para realizar a previsão de STs de consumo de energia elétrica, acoplando-a como módulo a um sistema IoT que realiza monitoramento do consumo de energia elétrica de um equipamento específico ou um edifício, por exemplo. Também realizamos a avaliação de seu desempenho através de testes estatísticos usando diferentes *datasets* em diferentes escalas e características, a partir da VC-STs com dados reais. É importante destacar que outro ponto importante que diferencia este trabalho dos supracitados é que grande parte deles não realizam esses testes estatísticos para validar o seu resultado.

CAPÍTULO 3

3 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta uma visão geral dos principais conceitos nos quais esta Tese se baseia, abordando conceitos de IA, AP, RNAs, RNRs, LSTM e BiLSTM. Os algoritmos de AM XGBoost e RF, que foram utilizados para comparação no presente trabalho também serão descritos. Aborda-se também o uso de sistemas IoT com RNPs, além de uma visão geral do sistema IoT EnergySaver e a proposta de acoplamento do módulo de RNP ao sistema IoT.

3.1 Inteligência Artificial

Nos dias atuais a IA pode ser considerada a tecnologia mais importante, tendo em vista sua utilização em diversas áreas do conhecimento, particularmente no que se refere a sistemas baseados em AM e RNPs. Grandes avanços nos campos de IA são observados principalmente nas áreas de percepção e cognição, especificamente em: reconhecimento de voz, de imagem, predição de STs, e solução de problemas específicos.

Conforme Russell e Norvig (2013), a IA pode ser definida a partir de duas dimensões: as que se relacionam ao processo de pensamento e raciocínio, e as que se referem ao comportamento. Em seu livro “Inteligência Artificial” lançado em 2013, os autores apresentaram um panorama histórico da evolução da IA, que descreve desde a concepção dos seus conceitos até a data de sua publicação. De forma resumida, a evolução cronológica ocorreu da seguinte forma:

- 1943: foi apresentado o primeiro trabalho que introduziu os conceitos iniciais de IA, por Warren McCulloch e Walter Pitts, baseando-se em três fontes: 1) o conhecimento da fisiologia básica e da função dos neurônios no cérebro; 2) uma análise formal da lógica proposicional criada por Russell e Whitehead; 3) a teoria da computação de Alan Turing. McCulloch e Pitts propuseram um modelo de neurônios artificiais, no qual cada neurônio se caracteriza por estar “ligado” ou “desligado”, com a troca para “ligado” ocorrendo em resposta à estimulação por um número suficiente de neurônios

vizinhos. Também sugeriram que redes definidas adequadamente seriam capazes de aprender.

- 1949: Uma regra simples de atualização para transformar as intensidades de conexão entre neurônios, foi demonstrada por Donald Hebb.
- 1950: Neste ano, o matemático inglês Alan Turing articulou a primeira visão de IA em seu artigo "*Computing Machinery and Intelligence*".
- 1951: Construção do primeiro computador de rede neural por dois alunos de Harvard, Marvin Minsky e Dean Edmonds.
- 1952: Os primeiros anos da IA foram repletos de sucessos, mas de uma forma limitada. Considerando-se os primitivos computadores, as ferramentas de programação da época e o fato de que apenas alguns anos antes os computadores eram vistos como objetos capazes de efetuar operações aritméticas e nada mais, causava surpresa o fato de um computador realizar qualquer atividade remotamente inteligente.
- 1956: Foi realizada conferência no Dartmouth College em New Hampshire, nos Estados Unidos, com participação de matemáticos e lógicos, para discutir os fatores envolvidos na construção de computadores que simulassem aspectos do pensamento. Neste evento o matemático John McCarthy introduziu o termo "*inteligência artificial*". No entanto, apesar de promissores, nenhum grande resultado foi obtido, mas o encontro de pesquisadores gerou interesse e material para os anos seguintes.
- 1957: Desde o início, os pesquisadores da IA eram ousados nos prognósticos de seus sucessos futuros. Herbert Simon, por exemplo, apresentou ao mundo uma visão de presente e futuro, com base em suas pesquisas, em que declarou que: "Não é meu objetivo surpreendê-los ou chocá-los, mas o modo mais simples de resumir tudo isso é dizer que agora existem máquinas que pensam, aprendem, criam e realizam atividades extremamente rápidas para resolução de uma variedade de problemas".
- 1958: McCarthy publicou o artigo "*Programs with common sense*", em que descrevia o *Advice Taker*, um programa hipotético que pode ser considerado o primeiro sistema de IA completo.

- 1959: Herbert Gelernter construiu o *Geometry Theorem Prover*, que podia demonstrar teoremas que seriam considerados bastante complicados por muitos alunos de matemática.
- 1960-1965: O trabalho pioneiro baseado nas redes neurais de McCulloch e Pitts também prosperou. Winograd e Cowan (1963) apresentaram uma pesquisa onde mostrou que grande número de elementos podia representar coletivamente um conceito individual, com aumento correspondente na robustez e no paralelismo. Bernie Widrow (Widrow e Hoff, 1960; Widrow, 1962) aperfeiçoou os métodos de aprendizado de Hebb, que denominou suas redes adalines, e por Frank Rosenblatt (1962) com seus perceptrons. O teorema da convergência do perceptron (Block *et al.*, 1962) determina que o algoritmo de aprendizagem podia ajustar os pesos de conexão de um perceptron para corresponderem a quaisquer dados de entrada, desde que existisse tal correspondência.
- 1967: O livro de Minsky e Papert, *Perceptrons* (1967), provou que, embora os perceptrons (uma forma simples de rede neural) pudessem aprender tudo o que eram capazes de representar, eles podiam representar muito pouco. Tal publicação gerou uma controvérsia relacionada ao desestímulo das pesquisas em IA, que está relacionada ao chamado “inverno da IA” que é o período de pouco interesse de financiamento de pesquisas em IA.
- 1969: Em 1969 Buchanan *et al.* (1969) desenvolveram o programa DENDRAL que tinha por objetivo resolver o problema de inferir a estrutura molecular a partir das informações fornecidas por um espectrômetro de massa, tornando-se importante porque representou o primeiro sistema bem-sucedido de conhecimento intensivo.
- Durante a década de 70 a IA estava praticamente restrita ao ambiente acadêmico e os objetivos da pesquisa eram, principalmente, a construção de teorias e o desenvolvimento de programas que verificassem estas teorias para alguns poucos exemplos, por isso esse período foi chamado de “inverno da IA”.
- 1980: Os sistemas baseados em IA fizeram tanto sucesso que se tornaram uma indústria lucrativa. Assim, a maioria das corporações importantes dos

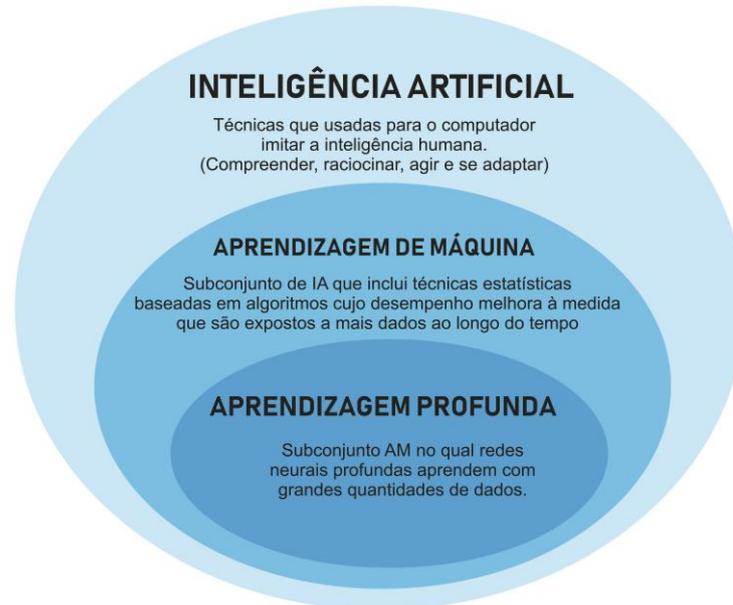
Estados Unidos tinham seu próprio grupo de pesquisas em IA e estavam usando ou investigando sistemas especialistas.

- 1986: Ocorreu a popularização das RNAs com estudos baseados no algoritmo de aprendizado por retroprogramação, também foram introduzidos os chamados “modelos conexionistas” para sistemas inteligentes. Ocorreu também, a separação entre a IA e a ciência cognitiva, onde as novas pesquisas relacionadas a RNAs foram divididas em dois campos: 1) criação de algoritmos e arquiteturas de rede eficazes; 2) modelagem cuidadosa das propriedades empíricas de neurônios reais e conjuntos de neurônios.
- 1987-2000 (também na atualidade): IA se torna uma ciência, revolucionando as formas de trabalho em IA, em relação ao seu conteúdo e metodologia. Assim, ao invés de propor teorias inteiramente novas, passou-se a usar as teorias existentes como bases, e em termos de metodologia, a IA adotou o método científico.
- 2001-2013 e dias atuais: Disponibilidade de *datasets* muito grandes (*Bigdata*). Atualmente há uma explosão de pesquisas e utilização de IA, sendo totalmente presente em nosso cotidiano, a partir de *smartphones*, computador, *tablets*, automóveis, televisão, dentre outros dispositivos.

Conforme Haykin (2009), o propósito da IA é construir padrões por meio de algoritmos utilizando as máquinas para desempenhar trabalhos de compreensão e aprendizado da mesma forma que os humanos. Assim sendo, um sistema de IA precisa ter habilidade para armazenar e aplicar o conhecimento salvo na resolução de problemas.

A IA pode ser considerada um “guarda-chuva” abrangente, pois refere-se a programas de computadores que têm por objetivo de simular o comportamento humano. Dentro dessa abrangência destacam-se as RNP e a AM. A AM é um conjunto de métodos matemáticos e estatísticos associados a programação de computadores, a fim de solucionar problemas através de classificação, regressão e segmentação (MOHRI, 2018). Para melhor entendimento da relação entre IA, AM e AP, deve-se visualizá-las como círculos concêntricos em que a IA surgiu primeiro, em seguida, com base em seus conceitos, surgiu a AM, e mais tarde a AP, conforme Figura 4.

Figura 4: Diferenças entre IA, AM e AP.



Fonte: Adaptado de LeCun *et al.* (2015).

De acordo com Chollet (2018), principal diferença entre a AM e AP é que a AM é, tecnicamente, a busca de representações úteis de alguns dados de entrada, em um espaço predefinido de possibilidades, usando a orientação de um sinal de feedback; já a AP é um subcampo da AM que apresenta uma nova abordagem de aprendizagem de representações de dados que enfatiza a aprendizagem de camadas sucessivas de representações cada vez mais significativas. Muitas vezes, envolve dezenas ou até centenas de camadas sucessivas de representações, e todas elas são aprendidas automaticamente com a exposição a dados de treinamento.

Ainda sobre essa diferença, Chollet (2018), destaca que algumas abordagens de AM tendem a se concentrar no aprendizado de apenas uma ou duas camadas de representações dos dados, sendo normalmente chamadas de aprendizado superficial; enquanto na AP essas representações em camadas são (quase sempre) aprendidas por meio de modelos chamados redes neurais, estruturados em camadas literais empilhadas umas sobre as outras.

3.2 Aprendizagem Profunda

O cérebro humano é altamente complexo pois possui a capacidade de aprender e processar uma grande quantidade de informações em tempo mínimo. Para vários

autores, a aprendizagem humana é uma reação do cérebro às informações contidas no ambiente, tornando-se um processo baseado na experiência, que resulta em uma mudança de comportamento relativamente permanente (GERRIG e ZIMBARDO, 2005; LEFRANÇOIS, 2008; MYERS, 2006; NOBRE, 2004).

As unidades principais do cérebro são os neurônios onde as informações são transmitidas e processadas, e de acordo com Carvalho *et al.* (2011), eles são responsáveis pelo aprendizado e raciocínio humano. De forma mais específica, os autores apontam que os neurônios são formados pelos dendritos (conjunto de terminais de entrada) e pelos axônios (longos terminais de saída), servindo como base conceitual para o desenvolvimento de algoritmos que simulem esses processos.

Existem abordagens propostas na literatura baseados em AM, com o objetivo de criar algoritmos computacionais que simulem o funcionamento do cérebro humano. Dentre os campos da AM, destaca-se o AP, o qual é baseado em um conjunto de algoritmos que visam moldar abstrações de altíssimo nível de dados a partir de um grafo profundo que possui inúmeras camadas utilizadas para processamento dos dados, sendo compostas por várias alterações lineares e não lineares.

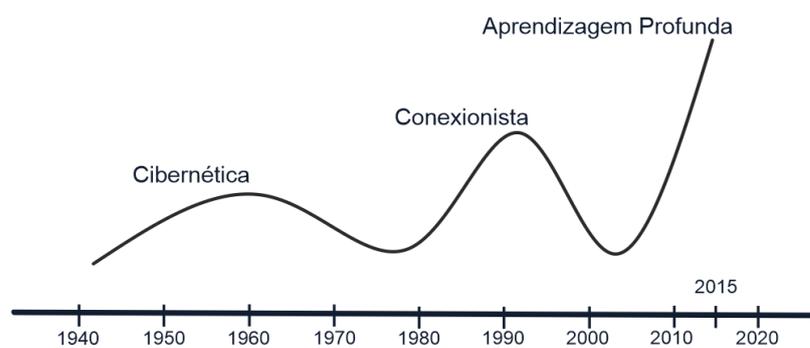
De acordo com Goodfellow (2016), a AP parece ser uma tecnologia nova, porém, possui uma longa e rica história, que surge a partir ano de 1943, tendo sido reformulado várias vezes, refletindo a influência de diferentes pesquisadores e diferentes perspectivas, tendo pequenos avanços nas décadas de 1950, 1960, 1970 e 1980. Em 1943 os cientistas Warren McCulloch (neurofisiologista) e Walter Pitts (matemático), criaram um modelo computacional em conceitos matemáticos e algoritmos chamados *threshold logic* ou lógica de limiar, o qual permitiu pesquisas sobre a rede neural dividida em duas vertentes: a primeira com foco nos processos biológicos cerebrais e a segunda com foco na aplicação dessas redes neurais voltadas a IA, usada para imitar o processo de pensamento humano.

Nessa perspectiva, em 1950, o matemático e cientista da computação britânico Alan Turing previu a existência futura de um supercomputador com inteligência semelhante à humana. O desenvolvimento dos fundamentos de um modelo com retropropagação contínua é creditada a Henry J. Kelley em 1960. Stuart Dreyfus, em 1962, apresentou uma versão mais simples baseada apenas na regra da cadeia. Apesar do conceito de retropropagação existir desde o início dos anos 1960, apenas em 1985 tornou-se amplamente utilizada por pesquisadores. Desde então, essa técnica tem passado por constante evolução, trazendo inúmeras contribuições.

Para Goodfellow (2016), houve três ondas de desenvolvimento de AP: a primeira onda foi denominada cibernética abrangendo os anos de 1940 a 1960, com o desenvolvimento de teorias de aprendizagem biológica e implementações dos modelos iniciais como o *perceptron*, permitindo o treinamento de um único neurônio; a segunda onda começou com a abordagem conexionista nos anos 1980 a 1990, com retropropagação para treinar uma rede neural com uma ou duas camadas ocultas; e a terceira onda faz referência ao seu ressurgimento atual sob o nome de AP a partir de 2006, que permitiu realizar o treinamento de redes muito profundas.

A Figura 5 apresenta uma linha do tempo básica para melhor compreensão em relação as três ondas.

Figura 5: Três ondas históricas do aprendizado profundo.



Fonte: Adaptado de Goodfellow (2016).

Na Figura 5, em ordem cronológica pode-se observar as três ondas históricas: a cibernética, a conexionista e AP. Entretanto, há de se destacar que na onda cibernética a definição do *Perceptron* mostrou-se útil para resolução de problemas linearmente separáveis. No entanto, na década de 1970, a partir de pesquisas, foi demonstrado que esse conceito possuía algumas limitações. Nesse contexto, surge a onda conexionista pautada em um novo conceito chamado algoritmo de retropropagação, onde permitiu-se treinar redes neurais a partir de poucas camadas ocultas. Além disso, o problema da perda de gradiente apoiou a ideia de que treinar redes neurais com mais do que poucas camadas, era um desafio complexo. A terceira onda (AP) surgiu a partir de pesquisas que mostraram ser possível treinar redes neurais artificiais com muitas camadas ocultas, conceito que vem sendo utilizadas até os dias atuais (GOODFELLOW, 2016).

Os primeiros modelos de AP tinham múltiplas camadas com características não lineares baseados em modelos profundos com funções de ativação polinomial.

Segundo Schmidhuber (2015), a AP teve sua origem a partir das pesquisas de Alexey Grigorevich Ivakhnenko (que criou o método de grupo para manipulação de dados - *Group Method of Data Handling*, GMDH), e que junto com Valentin Lapa e McDonough, escreveram "*Cybernetics and Forecasting Techniques*" (Ivakhnenko *et al.*, 1967). Modelos finos, porém profundos eram usados com funções de ativação polinomiais, para fazer análise de métodos estatísticos. Em cada camada, eles selecionavam os melhores recursos através de métodos estatísticos e encaminhavam para a próxima camada. Eles utilizaram mínimos quadrados camada por camada, onde as camadas anteriores foram independentemente instaladas em camadas posteriores, a partir de um processo lento e manual (NVIDIA, 2015).

As primeiras redes neurais convolucionais foram usadas por Kunihiko Fukushima, e possuíam múltiplas camadas de agrupamento e convoluções. Em 1979, ele desenvolveu uma RNA a qual denominou *Neocognitron*, que usava um design hierárquico e multicamadas. Este design permitiu ao computador "aprender" a reconhecer padrões visuais. Essas redes eram semelhantes a versões modernas, no entanto foram treinadas a partir de uma estratégia de reforço de ativação recorrente em múltiplas camadas, ganhando força ao longo do tempo. Outro ponto importante de se destacar é que esse design permitiu que alguns recursos fossem ajustados manualmente aumentando o "peso" de certas conexões (NVIDIA, 2015).

A retropropagação foi utilizada no início dos anos 1960 de uma forma ineficiente e incompleta. A forma moderna foi derivada primeiro por Linnainmaa (1970) em sua dissertação de mestrado, onde incluía o código Fortran para retropropagação, no entanto era relativamente desconhecida e existiam poucas aplicações documentadas. Os pesquisadores Rumelhart, Hinton e Williams (1986) revelaram que a retropropagação em redes neurais poderia produzir representações distribuídas interessantes, tornando-se um resultado importante na área da psicologia cognitiva, onde a questão era se a cognição humana pode ser pensada como baseada em representações distribuídas (conexionismo) ou lógica simbólica (computacionalismo). A primeira aplicação bem sucedida da retropropagação foi apresentada por LeCun *et al.* (1989). Ele usou redes convolucionais em combinação com retropropagação para classificar dígitos manuscritos (MNIST) e este sistema foi usado posteriormente para ler um grande número de cheques manuscritos nos Estados Unidos (NVIDIA, 2015).

Apesar dos resultados promissores, os financiamentos para pesquisas em redes neurais eram escassos. O termo inteligência artificial quase caiu para o status de

pseudociência durante um período de tempo. Alguns avanços importantes foram obtidos nesta época, como por exemplo, as Redes Neurais LSTM apresentadas por Hochreiter e Schmidhuber (1997), a RNN Bidirecional no mesmo ano por Schuster e Paliwal (1997) e a LSTM Bidirecional por Graves e Schmidhuber (Graves et al., 2009; Graves e Schmidhuber, 2005, 2009). No entanto, apesar de as pesquisas continuarem apresentando resultados promissores, as mesmas passaram despercebidas, pois foram ofuscadas pelo desenvolvimento da SVM por Cortes e Vapnik (1995), que apresentavam bons resultados (NVIDIA, 2015).

A grande mudança em relação a utilização do aprendizado profundo se deu com a introdução de unidades de processamento gráfico (*graphics processing unit* - GPUs), que tornaram os computadores com processamento mais rápidos. Desse modo, a medida que a velocidade das GPUs aumentavam, era possível treinar redes profundas, como redes convolucionais, sem a ajuda do pré-treinamento. Outro aspecto importante a ser destacado se refere a programação paralela em GPUs, onde as unidades de processamento gráfico, que possibilitam a realização de operações matemáticas de modo paralelo, especialmente aquelas com matrizes e vetores, os quais estão presentes em modelos de RNAs, permitiram a evolução atual, isto é, o somatório de *Big Data* (grande volume de dados); processamento paralelo e os modelos de AM (NVIDIA, 2015).

No ano de 2009, no Workshop de Processamento de Sistemas de Redes Neurais sobre aprendizado profundo para Reconhecimento de Voz, verificou-se que mesmo com um grupo de dados extenso as redes neurais não precisam de treinamento prévio e os índices de falha caem significativamente. Em 2012, as pesquisas proporcionaram algoritmos de identificação de padrões artificiais com desempenho humano em algumas tarefas, como por exemplo o algoritmo do Google que identificava felinos. Entre os anos de 2012 e 2014, Google, Facebook e Microsoft perceberam essa tendência e fizeram grandes aquisições de startups de aprendizado profundo e equipes de pesquisa. Em 2015, o Facebook utilizou a AP para marcar e reconhecer automaticamente os usuários em fotografias. Os algoritmos executam tarefas de reconhecimento facial utilizando RNPs, dentre outras aplicações (CHAGAS, 2019). A partir do ano de 2017 a AP passou a ter maior notoriedade por fatores como o aperfeiçoamento da capacidade de processamento de chips, a maior disponibilidade de dados para treinamentos, além dos grandes avanços obtidos por pesquisas com

métodos cada vez mais eficazes para funções complexas e não lineares (CHUNG *et al.*, 2015; LECUN *et al.*, 2010).

Os modelos de aprendizado profundo baseados em RNAs profundas utilizam variados tipos de algoritmos, onde cada modelo é adequado para executar tarefas específicas. Eles conseguem encontrar bons resultados a partir de vários tipos de dados, porém, necessitam de grande capacidade de poder computacional e informações para resolver problemas complicados.

3.3 Redes Neurais Artificiais

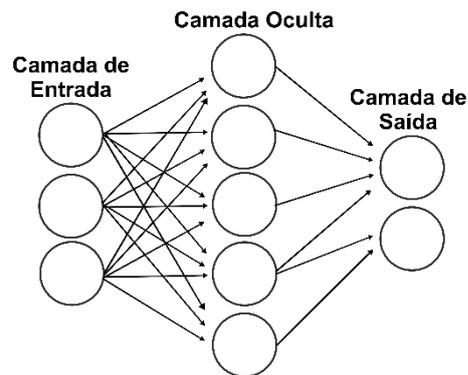
Desde que as RNAs começaram a surgir, várias definições foram formuladas por diferentes pesquisadores e em todas elas prevaleceram as semelhanças com sistema nervoso biológico. Isso porque a principal motivação do seu surgimento foi a premissa de que o cérebro humano é semelhante a um computador com processamento de informações altamente complexas. As RNAs por meio de algoritmos, podem reconhecer padrões ocultos, realizar correlações em dados brutos, agrupar, classificar, e aprender com os erros gerando resultados melhor em testes futuros (HAYKIN, 1998).

Conforme Nelson e Illingworth (1990) as RNAs são técnicas de processamento de informações, com objetivo de simular em computadores o funcionamento do sistema nervoso biológico. Para, Fausett (1994) as RNAs são sistemas de processamento de informações com características similares às das redes neurais biológicas. Zhang *et al.* (1998), destaca que as RNAs são projetadas para identificar tendências em *datasets* e fazer uma generalizações a partir dela.

Segundo Haykin (1998), uma RNA é um sistema de processamento maciçamente paralelo, sendo constituído por unidades simples que possuem capacidade natural para armazenar conhecimento experimental adquirido e disponibilizá-lo para uso futuro, denominadas de neurônios artificiais.

As RNAs são compostas de camadas de muitos processadores simples chamados neurônios, com conexões em diferentes camadas, visando transformar dados para que possam classificá-los como uma saída. Os neurônios são ativados por meio de conexões ponderadas de neurônios previamente ativos, conforme Figura 6 (HAYKIN,1998).

Figura 6: Camadas das Redes Neurais.

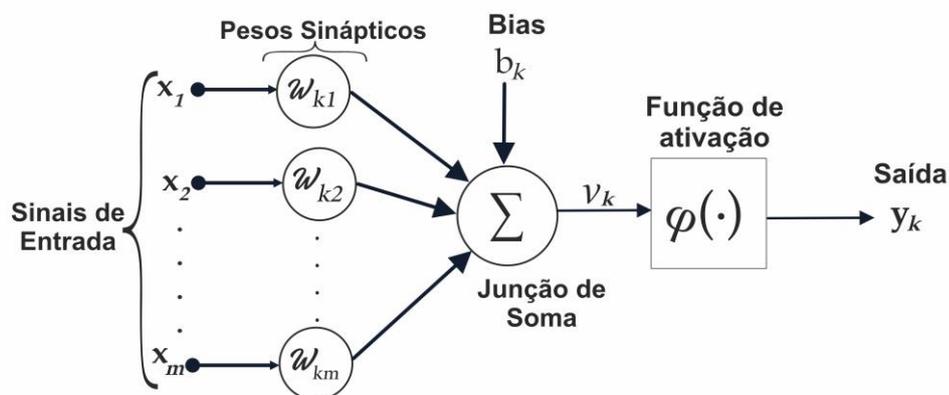


Fonte: Adaptado de Haykin (1998).

Na Figura 6 pode-se observar o funcionamento genérico de uma RNA. A habilidade de aprender em um determinado ambiente é a característica mais importante das redes neurais, melhorando assim seu desempenho. Isso ocorre mediante um processo iterativo de ajustes aplicado a seus pesos, chamado de treinamento, assim, o aprendizado ocorre a partir do momento que a rede encontra uma solução generalizada para uma classe de problemas.

A estrutura que ficou conhecida como a unidade básica de uma rede neural foi inicialmente projetada por McCulloch e Pitts (1943), onde os mesmos propuseram um modelo de neurônio artificial como uma unidade de processamento binária e mediante testes provaram que elas são capazes de executar várias operações lógicas. A arquitetura básica modelo de McCulloch-Pitts de um neurônio artificial é apresentada na Figura 7 (HAYKIN, 2009).

Figura 7: Modelo de um neurônio artificial.



Fonte: Adaptado de Haykin (2009).

A formação de neurônios artificiais possui três elementos básicos: as conexões de entradas, a junção de soma e uma função de ativação. Cada entrada possui ponderação por um peso sináptico, onde x_1, x_2, \dots, x_m são as entradas da rede; $w_{k1}, w_{k2}, \dots, w_{km}$ são os pesos sinápticos associados a cada entrada; a execução do neurônio se dá pelo somatório de todas as suas entradas multiplicadas por seus pesos sinápticos; b_k é o termo de *bias* (representada pela entrada fixa que aumenta ou diminui a entrada líquida da função de ativação), que tem a função de aumentar o grau de liberdade dos ajustes dos pesos; v_k é a combinação linear dos sinais de entrada; $\varphi(\cdot)$ é a função de ativação onde ocorre a transformação não linear; e y_k é a saída do neurônio (HAYKIN, 2009).

Para Haykin (2009), o aprendizado de uma rede neural é dependente de sua arquitetura (quantidade de neurônios e conexões) e das regras em que são ajustados os pesos da conexões entre neurônios. Em termos matemáticos a entrada total recebida um neurônio artificial pode ser representado como:

$$v_k = \sum_{j=0}^m w_{kj} x_j \quad (1)$$

na qual: v_k é a entrada total do neurônio; m representa o total de atributos de um objeto x e o número de terminais de entrada; w_{kj} , $1 \leq j \leq m$ representa o peso do vetor de pesos w , aplicado ao terminal de entrada j ; e x_j , $1 \leq j \leq m$ representa o atributo j do objeto x .

O propósito da função de ativação em uma RNA é restringir a amplitude da saída do neurônio de acordo com o problema a ser solucionado, ou seja, definir um limite para a saída por um intervalo de valor finito.

3.3.1 Treinamento de RNAs

Para Faceli *et al.*, (2011), a essência de um algoritmo de aprendizado é a regra utilizada para propor o seu aprendizado durante o treinamento, tendo em vista que é a partir da regra de atualização dos pesos que são determinados como os pesos das conexões entre os neurônios serão modificados. Nesse contexto, Buduma e Locascio (2017) apontam que essas regras de aprendizagem são chamadas otimizadores e

visam maximizar o desempenho de um determinado modelo de aprendizado, pois irão ajustar iterativamente seus parâmetros objetivando minimizar o erro. No entanto, Haykin (2009) destaca que a frequência de atualização dos pesos deve ser analisada, tendo em vista que influencia no desempenho obtido durante o treinamento da rede.

Segundo Buduma e Locascio (2017) para realizar a atualização dos pesos das conexões de uma rede neural existem duas abordagens diferentes:

- (a) Por padrão (*online*), em que os pesos são atualizados após a apresentação de cada padrão de treinamento.
- (b) Por ciclo (*batch*), onde os pesos são atualizados após um número n de padrões terem sido apresentados, sendo normalmente a mais precisa e estável.

É importante destacar que existem inúmeras maneiras de tornar esse processo ainda mais eficiente, que vão desde as observações do programador durante realização testes para esse fim, até a utilização de ferramentas automatizadas que visam determinar as melhores configurações de hiperparâmetros para o problema em questão. Os principais hiperparâmetros serão descritas na Subseção seguinte.

3.3.2 *Underfitting e Overfitting*

Dois dos problemas mais recorrentes em implementações de redes neurais estão relacionados ao quão bem um dado modelo se ajusta aos dados de treinamento, eles são conhecidos como *Underfitting* e *Overfitting*. No processo de aprendizado de um modelo, o conjunto de dados usados para o treinamento costumam ser iterados várias vezes, na tentativa de fazer com que o modelo consiga capturar melhor a tendência e a variância dos dados. Essa iteração é chamada de épocas (do inglês: epochs). Um dos problemas encontrado no aprendizado de máquina é saber exatamente por quantas épocas o modelo deve ser treinado para encontrar o resultado otimizado, pois caso seja treinado por poucas iterações pode ocorrer *underfitting* (sub-ajuste) e se treinado demais, acontece o *overfitting* (sobre-ajuste).

Underfitting ocorre quando o modelo não consegue capturar a tendência subjacente dos dados. A falta de complexidade da função que mapeia a entrada em uma saída prejudica a precisão do modelo.

Overfitting é um erro de modelagem que ocorre quando uma função (ou modelo) é treinada com um conjunto limitado de pontos de dados. Isto significa que caso haja *overfitting* no treinamento, o modelo construído será muito bom em classificar o *dataset* usado para o aprendizado, mas costuma obter um mau desempenho para dados ainda não utilizados, ou seja, não generaliza os padrões bem o suficiente.

Para evitar *underfitting* é necessário aumentar a complexidade/grau da função que mapeia a entrada na saída. Já para evitar o *overfitting*, é necessário utilizar algumas técnicas um pouco mais complexas, dentre as quais as que foram utilizadas neste trabalho, sendo a validação cruzada descrita da Subseção 4.5.3.

3.3.3 Configuração de Hiperparâmetros em Redes Neurais

Um dos problemas enfrentados ao utilizar uma rede neural para resolução de um determinado problema está relacionado a sua configuração no que diz respeito à definição de seus hiperparâmetros. Trata-se de uma tarefa complexa pois envolve a definição adequada de variáveis visando maximizar o seu desempenho. Para Claesen e Moor (2015), os hiperparâmetros são usados para configurar vários aspectos do algoritmo utilizado e influenciam de variadas formas na rede neural e em seu desempenho. Já Reimers e Gurevych (2017), enfatizam que a escolha de hiperparâmetros pode representar a diferença entre o desempenho pífio ou excelente.

Existem inúmeras pesquisas (por ex: Claesen e Moor (2015), Reimers e Gurevych (2017), dentre outros) que buscam encontrar formas para determinar uma configuração ideal de hiperparâmetros visando a resolução de problemas com redes neurais, no entanto, cada pesquisa tem suas peculiaridades onde as observações a serem trabalhadas são diferentes, bem como as STs tem suas especificidades. Desse modo, pode-se concluir que sempre que uma pesquisa nessa temática é publicada, os autores apontam apenas sugestões que estão fundamentadas em experiências vividas por especialistas no assunto e no bom senso, sendo que em várias situações podem não ser generalizadas a grande parte dos problemas.

Os principais hiperparâmetros de uma rede neural são detalhados na Tabela 3 a seguir.

Tabela 3: Principais hiperparâmetros de uma RNA e sua função.

Hiperparâmetros	Descrição
<i>Número de unidades de células recorrentes (Units)</i>	Esta variável define a profundidade da rede neural, tendo em vista que as RNPs podem trabalhar com um grande número de camadas. Quanto maior a profundidade maior o tempo de treinamento do modelo.
<i>Taxa de esquecimento (Dropout)</i>	Trata-se de uma técnica utilizada em RNAs modernas para prevenir <i>Overfitting</i> , neste caso, consiste em descartar uma certa porcentagem dos neurônios em cada etapa de treinamento. No momento de fazer previsões, todos os neurônios são mantidos e a rede neural atua como uma grande mistura de sub-redes menores.
<i>O tamanho do lote (BatchSize)</i>	É uma otimização que se refere a quantidade de exemplos de treinamento usados em uma iteração, ou seja, define o número de amostras que serão propagadas pela rede. A vantagem de usar um <i>BatchSize</i> menor que número das amostras é que ele requer menos memória, pois a rede será treinada com menos amostras. E a desvantagem é que não há garantia que o modelo vai convergir para o ótimo global.
<i>O número de épocas (Epochs)</i>	Define o número de vezes que todo o <i>dataset</i> de treino será apresentado para a rede durante o treinamento. É importante destacar que se o número de épocas for muito pequeno, a rede encontrará dificuldades em assimilar padrões que são pouco comuns nos dados, contudo, se esse valor for muito grande, pode ocorrer o <i>overfitting</i> .
<i>Algoritmos de otimização (Optimizer)</i>	A função do otimizador em uma rede neural é diminuir o erro entre os resultados obtidos em relação aos resultados desejados. Desse modo, quanto menor o erro, maior a acurácia dos resultados e conseqüentemente melhor o desempenho da rede. Reimers e Gurevych (2017), em extensivos testes realizados, concluíram que os otimizadores <i>Adam</i> , <i>Nadam</i> e <i>RMSprop</i> produziram resultados mais estáveis e melhores para as tarefas de rotulagem de sequência.
<i>Função de perda (loss function)</i>	Esta função calcula o erro em relação a distância entre o valor previsto e o valor real.
<i>Métricas de desempenho das RNAs (metrics)</i>	As métricas de desempenho calculam a diferença entre a predição esperada e o resultado calculado pelo modelo. Elas são utilizadas para treinamento e avaliação. As métricas utilizadas neste trabalho estão descritas na Subseção 4.7.

*Função de Ativação
(activation)*

É um elemento importante em uma RNA, pois é onde ocorre a decisão da ativação ou não de um neurônio, ou seja, se a informação recebida pelo neurônio é relevante ou se será ignorada. Na Subseção 3.3.3 serão descritas as funções de ativação comumente utilizadas em RNAs.

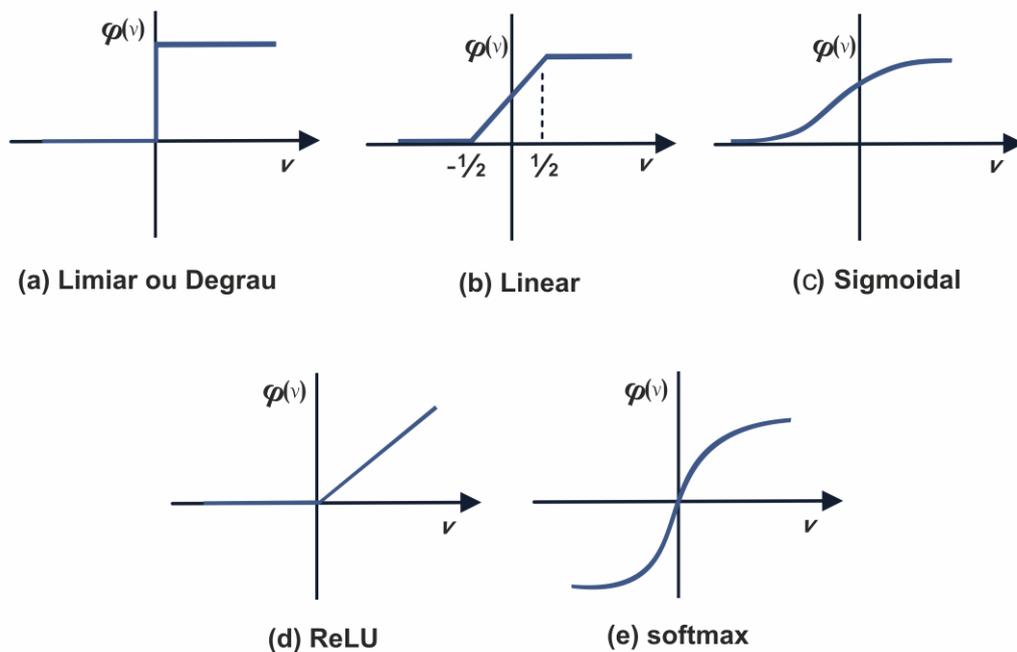
Fonte: Elaboração própria, 2022.

De modo geral, Goodfellow *et al.* (2016) destacam que os valores dos hiperparâmetros não são adaptados pelo próprio modelo de aprendizado, contudo é possível projetar um algoritmo alinhado a um determinado modelo de aprendizagem visando encontrar os melhores hiperparâmetros.

3.3.4 Funções de Ativação de uma RNA

Haykin (2009) aponta que existem diversas funções de ativação que podem ser usadas, com destaque para Limiar, Linear, Sigmoidal, ReLU (unidade linear Retificada, do inglês *Rectified Linear Units*) e a Softmax. As representações gráficas dessas funções de ativação são mostradas na Figura 8 a seguir.

Figura 8: Funções de Ativação de uma RNA.



Fonte: Adaptado de Haykin (2009).

Na Função Limiar ou degrau: A saída do neurônio assume valor “0” quando seu resultado for negativo, e valor “1” caso contrário, da seguinte forma:

$$\varphi(v) = \begin{cases} 1 & \text{se } v \geq 0 \\ 0 & \text{se } v < 0 \end{cases} \quad (2)$$

Na Função Linear: Esse tipo de função é visto como uma aproximação de um amplificador não linear, descrita como como:

$$\varphi(v) = \begin{cases} 1 & \text{se } v \geq \frac{1}{2} \\ v, & \text{se } -\frac{1}{2} < v < \frac{1}{2} \\ 0 & \text{se } v \leq -\frac{1}{2} \end{cases} \quad (3)$$

Função Sigmoidal: é a função de ativação mais utilizada em RNAs, normalmente definida como uma função monotonia crescente que apresenta propriedades assintóticas e de suavidade. Elas podem ser encontradas de várias formas, porém as mais comuns são: a função logística e a função tangente hiperbólica, representadas respectivamente por: $\varphi(v) = \frac{1}{1 + e^{-av}}$, onde a é o parâmetro de inclinação da função sendo que quanto maior seu valor, mais inclinada a curva se tornará; e: $\varphi(v) = \tanh(v)$, função que pode assumir valores negativos.

Função ReLU: é uma função de ativação não linear amplamente utilizada para extrair características, podendo ser utilizada tanto para pré-treinamento quanto para classificação, pois possui uma alta velocidade de convergência. Seu principal objetivo é reduzir o tempo de treinamento de RNPs, sendo aplicada aos neurônios das camadas ocultas. Se a entrada do neurônio for negativa, ele não é ativado o que torna o processamento mais eficiente. A equação da ReLU pode ser descrita como: $\varphi(v) = \max(0, v)$, onde, $\varphi(v)$ é a saída da função que retorna o valor máximo entre 0 e o valor de v , que é o parâmetro que está sendo analisado. A ReLU retorna o seu próprio valor para valores positivos e 0 para todos os valores negativos com tendência de “apagar” alguns neurônios durante um passo *forward*, o que aumenta a

velocidade do treinamento, mas por outro pode fazer com que esses neurônios “morram” e não aprendam nada se eles só receberem valores negativos.

Função Softmax: é utilizada para resolução de problemas de classificação e pode ser vista como uma generalização da função sigmoide para casos não-binários. Sua principal utilização é na camada de saída, quando o domínio do problema é multiclasse, onde cada neurônio de saída representa uma classe, e a soma da saída de todos os neurônios é 1, e o valor de cada neurônio de saída é a probabilidade da entrada pertencer a certa classe. Essa função pode ser descrita por:

$$\varphi(v) = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}} \quad (4)$$

na qual, z_i é o i -ésimo termo do vetor z e z_j é o j -ésimo termo do mesmo vetor, ou seja, é uma função que recebe um vetor com z números reais e o normaliza em uma distribuição probabilística para os z valores.

3.4 Redes Neurais Recorrentes

As RNRs possuem como característica as realimentações entre os seus neurônios, onde suas conexões não se limitam ao sentido de entrada para a saída como na *feedforward*. Deste modo, se houver uma ligação que caracterize como um retorno de uma informação de um neurônio ou entrada anterior, então esta rede neural pode ser considerada como recorrente. Essas redes são utilizadas principalmente em previsão de séries temporais, otimização de funções e identificação de sistemas, controle de processos; tendo em vista possibilitar soluções para problemas com dependência temporal entre as amostras (HAYKIN, 2009). Mais informações sobre RNRs e seu contexto histórico na IA são dadas na extensa revisão feita por Schmidhuber (2015).

Conforme Olah (2015) as RNR são redes neurais com *loops* que possuem estruturas de processamento representando uma ampla variedade de procedimentos dinâmicos, permitindo a criação de representações internas e dispositivos de memória que tornam-se capazes de processar e armazenar informações temporais e sinais sequenciais. Essas RNRs foram projetadas principalmente para reconhecer padrões em sequências de dados.

De forma bem detalhada e explicativa, Zaremba *et al.* (2014) descrevem que as RNRs são uma classe de redes neurais que recebem entrada de duas fontes: uma do presente, e outra de um ponto passado. Assim sendo, mediante as informações dessas duas fontes ocorre a decisão por meio de um circuito de feedback (saída de cada instante é uma entrada para o instante seguinte), para decidir o que deve ser feito a partir de uma nova entrada de dados.

De acordo com Haykin (2009), as arquiteturas de RNRs podem ser de quatro tipos: modelo recorrente de entrada e saída (*Input-Output Recurrent Model*), modelo de espaço de estados (*State-Space Model*), MLP recorrente (*Recurrent Multilayer Perceptrons - RMLP*) e rede de segunda ordem (*Second-Order Network*).

No modelo recorrente de entrada e saída, sua arquitetura genérica é formada por uma MLP possuindo uma única entrada que é aplicada a uma memória de linha de atraso tapada de q unidades. Esse modelo tem uma única saída que é realimentada à entrada por meio de outra memória de linha de atraso de derivação, também de q unidades. Além disso, o conteúdo das memórias de linha de atraso de toque são utilizados para alimentar a camada de entrada da MLP (HAYKIN, 2009).

Já no modelo de espaço de estados, os neurônios ocultos definem o estado da rede, onde a saída da camada oculta é realimentada para a camada de entrada (concatenação de nós de feedback e nós de origem) por meio de um banco de atrasos de tempo unitário, ficando conectada ao ambiente externo por meio dos nós de origem. Assim, a ordem do modelo é determinado pelo número de atrasos unitários usados para alimentar a saída da camada oculta, retornando à camada de entrada (HAYKIN, 2009).

O outro modelo trata-se da MLP recorrente, que possui uma ou mais camadas ocultas, onde cada camada possui retroalimentações em torno de si. É uma rede híbrida neural (*feedforward/feedback*) com a realimentação representada por conexões recorrentes e cruzamentos apropriados. Ela é usada na identificação da dinâmica de sistemas não lineares desconhecidos, sendo composta basicamente por uma camada de entrada, várias camadas ocultas e uma camada de saída (HAYKIN, 2009).

Por fim, o modelo de segunda ordem tem como característica a mudança de seus pesos dinâmicos de acordo com o estado interno, ou seja, a ativação dos nós de contexto. Os nós - que são a entrada para a rede de contexto - e a ativação deles atualiza os pesos da rede de funções a cada passo de tempo, através dos pesos da

rede de contexto. Vale ressaltar que o termo “ordem” é utilizado para fazer referência ao número de neurônios ocultos, cujas saídas são realimentadas para a camada de entrada por meio de um banco de atrasos de unidades de tempo. Ainda em outro contexto, o termo “ordem” é usado, geralmente para se referir à maneira pela qual o campo local induzido de um neurônio é definido (HAYKIN, 2009).

Para Haykin (2009), o principal problema das RNRs consiste no fato de que com o tempo o gradiente da função de custo decai exponencialmente e o modelo para de aprender, ou seja, ocorre a limitação da capacidade de “lembrar” de grandes quantidades de dados ao longo do processo. De acordo com Hochreiter e Schmidhuber (1997), esse problema é chamado de problema da dissipação do gradiente (*Vanishing Gradient*), que também ocorre em outras RNAs.

Assim sendo, para contornar problema, em 1997 Hochreiter e Schmidhuber propuseram uma arquitetura conhecida como LSTM. No mesmo ano Schuster e Paliwal (1997) apresentaram a RNN Bidirecional. Posteriormente Graves e Schmidhuber (Graves *et al.*, 2009; Graves e Schmidhuber, 2005, 2009) aplicaram os conceitos da RNN Bidirecional à LSTM, apresentando a abordagem LSTM Bidirecional. Nas subseções 3.3 e 3.4 as rede neural LSTM e BiLSTM serão descritas detalhadamente.

3.5 Redes Neurais LSTM

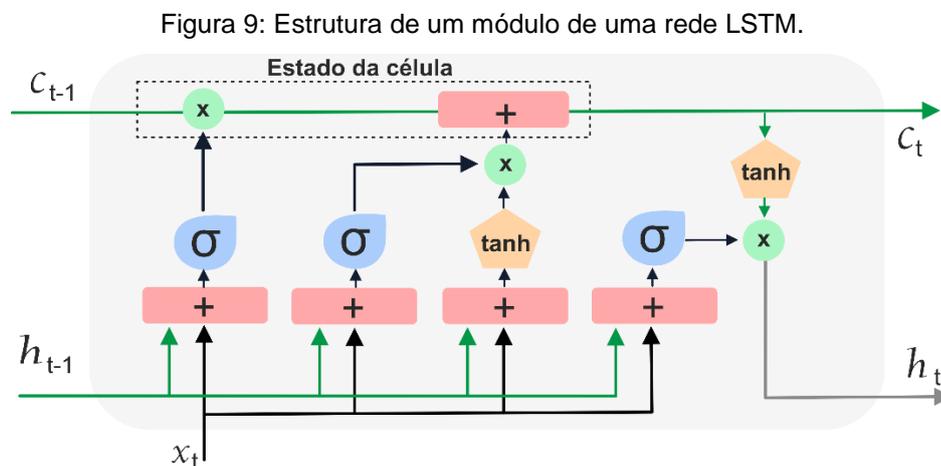
Conforme mencionado anteriormente, os conceitos de redes neurais LSTM foram inicialmente introduzidas por Hochreiter e Schmidhuber em 1997, visando propor uma solução para o problema do *vanishing gradient*, comum em RNRs. Além da proposta dessa estrutura de rede, foi apresentado também um recálculo para os gradientes obtidos no ajuste dos pesos por meio do *backpropagation*. De modo geral, a LSTM possui capacidade de reter maiores informações de entradas passadas (*long-term memory*) ao mesmo tempo que mantém a relevância de estados recentes.

Graves e Schmidhuber (2005) apontaram que a arquitetura LSTM foi motivada por uma análise do fluxo de erros em RNNs existentes, que constatou que longos atrasos de tempo eram inacessíveis às arquiteturas existentes, porque o erro retropropagado aumenta ou decai exponencialmente.

Uma unidade LSTM é composta de um neurônio e portões (*gates*) de entrada, esquecimento e saída. O neurônio é responsável pela memória da rede, representada

pela ativação baseada na soma ponderada e os portões são uma forma de, opcionalmente, permitir a passagem informação (HOCHREITER e SCHMIDHUBER, 1997; SCHMIDHUBER, 2014).

A estrutura de uma rede LSTM é semelhante a de uma RNR padrão, porém, os módulos possuem componentes internos diferentes. Em uma LSTM o estado da célula é a parte mais importante, tendo em vista transmitir informações ao longo da cadeia. A informação no estado da célula é descartada ou modificada por várias unidades que são chamadas de portões. A estrutura de um módulo de uma rede LSTM é mostrada na Figura 9 (HASAN *et al.*, 2019).



Fonte: Adaptado de Hasan *et al.* (2019).

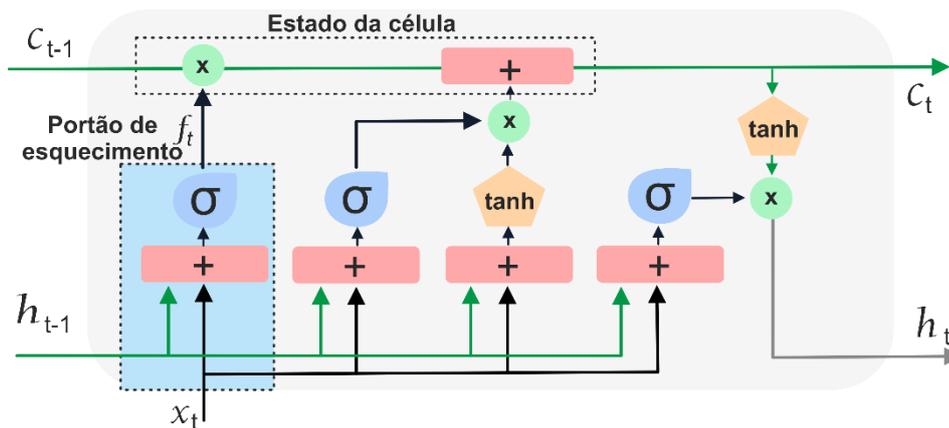
O funcionamento dos portões são descritos a seguir conforme Hasan *et al.* (2019), apresentados nas Figuras 10, 12 e 13, e nas Equações 4 a 9 (onde W_f , W_i , W_c e W_o , são parâmetros de pesos, b_f , b_i , b_c e b_o , são vetores de *bias*, \tilde{C}_t é a célula de memória candidata que determina o quanto levamos os novos dados em conta, e σ é a função *sigmoide*).

O portão de esquecimento f_t ilustrado na Figura 10 (destacado na cor azul dentro no módulo LSTM), consiste em uma camada *sigmoide* que leva o estado oculto anterior (h_{t-1}) e a entrada atual (x_t) para produzir uma saída entre 0 e 1. Essa camada decide quais informações devem ser mantidas ou descartadas pela rede. Um valor 0 significa esquecer as informações anteriores e 1 significa manter as informações anteriores. A saída do portão de esquecimento é escrita como:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (4)$$

Um exemplo de aplicação desse portão é no reconhecimento de escrita, na qual a rede deve ser capaz de sugerir palavras baseadas no assunto que está sendo escrito no momento e, também, reconhecer que o assunto mudou, esquecendo o primeiro assunto e com base no novo assunto sugerir novas palavras.

Figura 10: Módulo LSTM com destaque para o portão de esquecimento.



Fonte: Adaptado de Hasan *et al.* (2019).

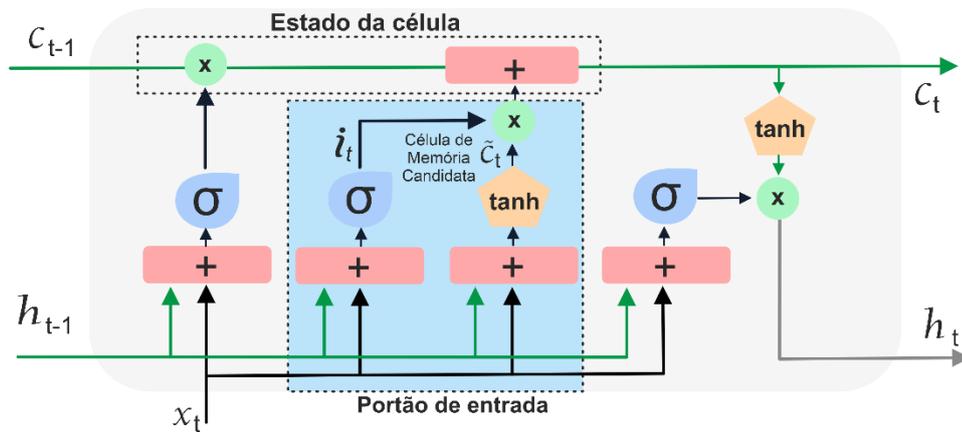
No portão de entrada i_t ilustrado na Figura 11 (destacado na cor azul dentro do módulo LSTM), são decididas quais novas informações serão armazenadas no estado da célula. Assim sendo, o portão de esquecimento utiliza uma função *sigmoide* e uma função *tanh* (tangente hiperbólica) para decidir quais informações serão adicionadas no estado da célula. Ambas as funções assumem (h_{t-1}) e (x_t) como valor de entrada. A saída *sigmoide* determina se a informação atual é importante ou não, enquanto a função *tanh* regula a rede tornando o valor entre -1 e +1, a partir da célula de memória candidata \tilde{C}_t . Finalmente, ambas as saídas são multiplicadas:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (5)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (6)$$

Na segunda parte dessa etapa os valores resultantes das duas camadas são combinados para atualizar o estado de célula.

Figura 11: Módulo LSTM com destaque para o portão de entrada.



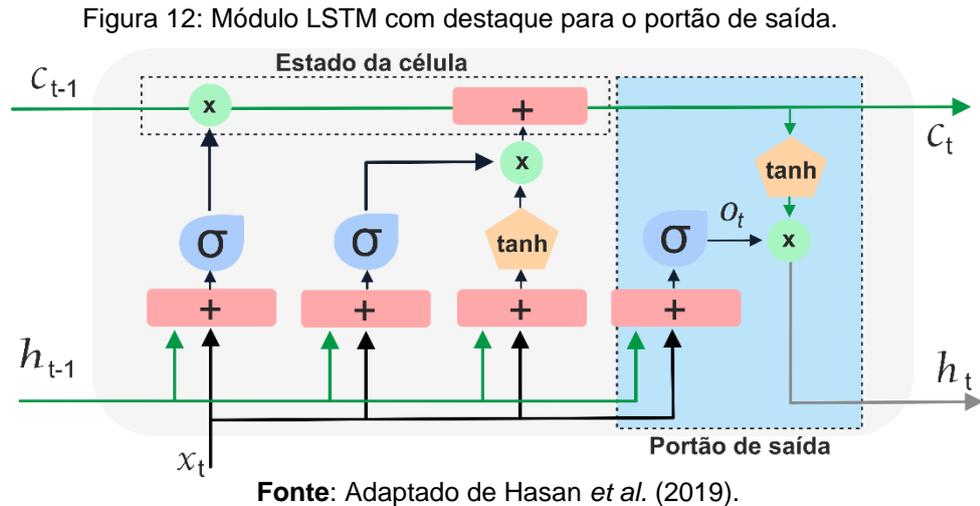
A partir da saída do portão de esquecimento e do portão de entrada, as informações no estado da célula são atualizadas realizando a multiplicação do estado da célula atual e da saída do portão de esquecimento. Se f_t for 0, a multiplicação também resultará em zero, o que significa abandono total do valor anterior. Caso contrário, se f_t for 1, ele é mantido. Posteriormente, a adição pontual atualiza o estado da célula como:

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t. \quad (7)$$

Por fim, o portão de saída O_t ilustrado na Figura 12 (destacado na cor azul dentro do módulo LSTM), determina a saída final e também atua como o próximo estado oculto, h_t . Neste portão, uma função *sigmoide* leva h_{t-1} e x_t como a entrada e o estado atual da célula C_t é passado através de uma função *tanh*. Em seguida, a saída *sigmoide* e a saída *tanh* são multiplicadas para determinar quais informações a camada oculta vai transportar, da seguinte forma:

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (8)$$

$$h_t = O_t \times \tanh(C_t) \quad (9)$$



A estrutura de portões utilizada pelas redes LSTM é o que permite a solução de diversos problemas relacionados a modelos sequenciais com dependências, como é o caso do problema de previsão de consumo de energia elétrica, abordado nesta pesquisa.

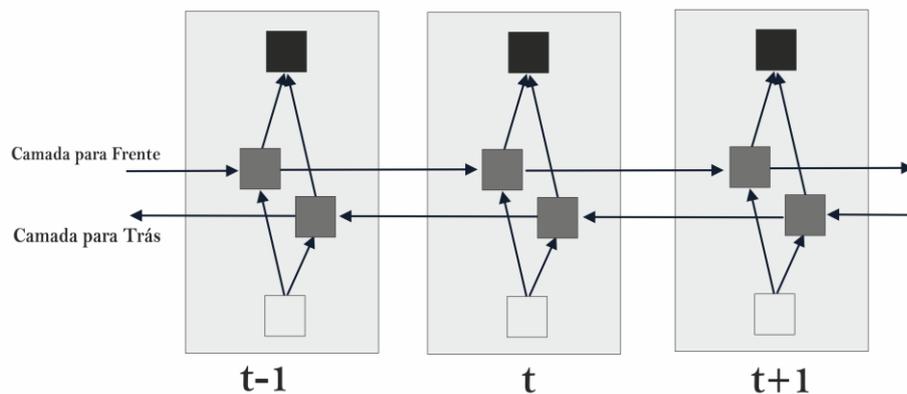
Visando otimizar o desempenho das RNRs, Schuster e Paliwal (1997) apresentaram uma abordagem para uso das RNR bidirecionais (BRNRs). A aplicação dessa técnica foi feita também com o objetivo melhorar o desempenho de qualquer RNR. Baseado nessa abordagem, Graves e Schmidhuber (Graves *et al.*, 2009; Graves e Schmidhuber, 2005, 2009), propuseram a LSTM Bidirecional. Na próxima Subseção são apresentados mais detalhes sobre as LSTMs bidirecionais.

3.6 Redes Neurais BiLSTM

Schuster e Paliwal (1997) apresentaram os primeiros conceitos de RNRs bidirecionais. Sua ideia básica é apresentar cada sequência de treinamento para frente e para trás a duas redes recorrentes separadas, ambas conectadas à mesma camada de saída. (Em alguns casos, uma terceira rede é usada no lugar da camada de saída, mas aqui usamos o modelo mais simples). Isso significa que, para cada ponto em uma determinada sequência, a BRNN tem informações completas e sequenciais sobre todos os pontos anteriores e posteriores à sequência (GRAVES e SCHMIDHUBER, 2005). Assim sendo, duas redes independentes podem processar os dados de entrada em sequências opostas, para dados cujo início e fim são conhecidos previamente, por exemplo, para o problema de estimativa do limite do

fonema (FUKADA *et al.*, 1999). Uma sequência é processada na direção usual de avanço (*forward state*), enquanto a sequência também é processada do final para o início (*backward state*), conforme mostrado na Figura 13.

Figura 13: Estrutura da célula de uma BRNR.



Fonte: Adaptado de Schuster e Paliwal (1997).

Dentro de uma estrutura BRNN, os neurônios de um RNN normal são divididos de forma bidirecional: um para estados atrasados ou direção de tempo negativa, e outro para estados avançados ou direção de tempo positiva. As entradas da direção inversa não estão conectadas aos resultados de ambos os estados. Assim, usando duas direções de tempo, podem ser usados dados de entrada, do passado e do futuro. O conceito BRNN foi combinado com a estrutura LSTM, e aplicações bem-sucedidas usando a Bidirecional LSTM (BiLSTM) foram desenvolvidas (por exemplo, Graves *et al.*, 2009; Graves e Schmidhuber, 2005, 2009).

Para Jurafsky e Martin (2000), o objetivo de uma BiLSTM consiste em obter mais conhecimento em relação a um determinado contexto ao capturá-lo em mais de uma perspectiva, para então concatenar ambas as saídas em uma representação contextual única em cada tempo t , conforme demonstrado na notação a seguir:

$$h_t^f = RNR_{\text{forward}}(x_1^t) \quad (10)$$

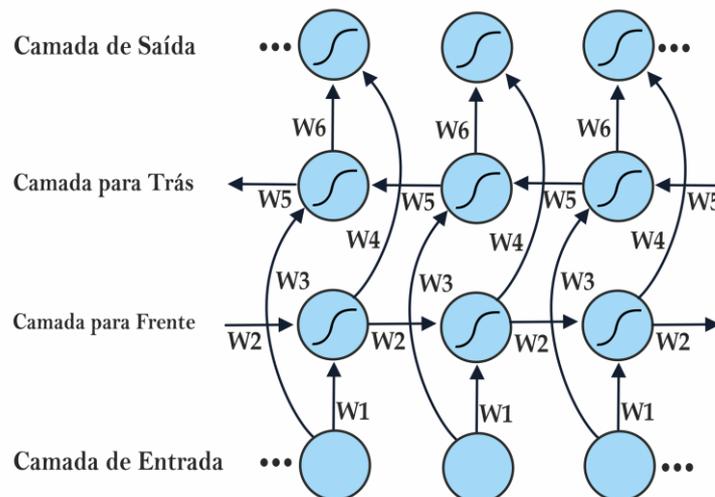
$$h_t^b = RNR_{\text{backward}}(x_t^n) \quad (11)$$

$$h_t = h_t^f \oplus h_t^b, \quad (12)$$

dado que h_t^f corresponde ao conhecimento de contexto obtido do começo até o passo t , e h_t^b o conhecimento é obtido da direita até o passo atual.

De acordo com Sharfuddin *et al.* (2018), na implementação de uma BiLSTM são utilizadas duas camadas de LSTM, sendo que uma das camadas é responsável pelos estados passados, enquanto a outra é responsável pelos estados futuros, conforme Figura 14.

Figura 14: Modelo da rede BiLSTM.



Fonte: Adaptado de Sharfuddin *et al.* (2018).

Conforme Zhao (2018), a estrutura da BiLSTM pode ser explicada da seguinte forma: a rede tem duas camadas ocultas, as setas horizontais representam o fluxo bidirecional no eixo temporal; as setas verticais representam o fluxo unidirecional da camada de entrada para a camada oculta e da camada oculta para a camada de saída (linhas retas). As linhas curvas são fluxos de unidades para frente e para trás, respectivamente.

De forma mais detalhada, Zhao (2018), explica o funcionamento da BiLSTM como: deixe $S = \{(y^i, x^i)\}_{i=1}^N$ representar o conjunto de N amostras. Para a amostra, a entrada x^i tem quatro recursos: trajetória em três dimensões e um relógio de tempo. A saída y^i depende de diferentes tarefas. Para a tarefa de classificação de acerto e erro, y^i tem um valor binário de acerto e erro. Para a tarefa de geração, y^i é a avaliação do próximo ponto x^{i+1} . Além disso, o autor aponta que uma única camada

BLSTM pode ser concatenada com uma sequência direta e uma sequência inversa, conforme demonstrado na notação a seguir:

$$\vec{h}_t = LSTM(x_t, \vec{h}_{t-1}) \quad (13)$$

$$\overleftarrow{h}_t = LSTM(x_t, \overleftarrow{h}_{t-1}) \quad (14)$$

$$y_t = g(W_{\vec{h}y} \vec{h}_t + W_{\overleftarrow{h}y} \overleftarrow{h}_t + b_y) \quad (15)$$

onde, $LSTM()$ é usada para representar todas as funções de uma LSTM padrão, $g()$ representa a função de ativação, W representa o peso e b representa o viés de uma determinada camada.

De acordo com Liang *et al.* (2019), durante a fase bidirecional da rede, as saídas produzidas são concatenadas e os seus valores normalizados por meio de uma função de ativação que produz a saída final da rede.

De modo geral, em relação a diferença entre as redes neurais LSTM e BiLSTM, Sharfuddin *et al.* (2018) destacam que as redes neurais LSTM unidirecionais, permitem o fluxo de entrada em apenas uma direção (para frente ou para trás); já na BiLSTM, a entrada flui em ambas as direções (para frente e para trás), sendo capaz de utilizar informações de ambos os lados, ou seja, ela adiciona uma nova camada LSTM que inverte a direção do fluxo de informações, onde a sequência de entrada flui para trás na camada LSTM adicional, e, em seguida, as saídas de ambas as camadas são combinadas de várias maneiras, como por exemplo, média, soma, multiplicação ou concatenação. A bidirecionalidade dos dados torna possível um melhor aprendizado das informações treinadas pela rede neural.

Existem alguns algoritmos de AM que são bastante usados para previsão de STs. No presente trabalho, para comparação com os modelos LSTM, foram usados os algoritmos XGBoost e RF, que serão descritos a seguir.

3.7 Algoritmo XGBoost

O algoritmo XGBoost é obtido a partir de uma biblioteca de código aberto que fornece uma implementação eficiente e escalável do framework *Gradient Boosting* (GB), sendo um método de *ensemble* de árvores de regressão, desenvolvido

inicialmente por Chen e He (2015). O XGBoost foi descrito por Chen e Guestrin (2016) e seu principal objetivo está relacionado a obtenção de uma execução rápida com bom desempenho do modelo, tentando diminuir as chances de *overfitting*.

De acordo com Friedman (2002), são os *ensembles* de árvores de regressão que apresentam hipóteses sobre os exemplos, acrescentando respostas de uma vasta quantidade de preditores simples. Isso ocorre em duas etapas: 1) uma árvore é criada adicionando uma ramificação de um atributo em sua estrutura para aperfeiçoar o preditor a cada fase, e após a sua definição estrutural calculam-se valores ótimos para as folhas; 2) a inserção de novas árvores ao *ensemble* é realizada com base no erro residual do modelo, moldando assim as imperfeições das fases anteriores para os novos preditores.

Para Chen e Guestrin (2016), a principal diferença entre o XGBoost e outras técnicas de GB, é que seus modelos usam um algoritmo sensível à dispersão na busca por ramificações, tornando a complexidade computacional do modelo linear com base no número de observações não ausentes, no mesmo sentido que adicionam otimizações do uso de recursos calculando paralelamente a aptidão dos atributos. De modo geral, o XGBoost aprimora a estrutura básica do GB mediante a otimização de sistemas e aprimoramentos algorítmicos, agregando recursos como a paralelização, computação distribuída, otimização de cache e processamento fora do núcleo.

Chen e Guestrin (2016), apontam que o modelo XGBoost adota uma estratégia aditiva, ou seja, para um determinado *dataset* com n amostras e m características $D = \{(X_i, y_i)\}$, a previsão é

$$\hat{y}_i = \sum_{k=1}^K f_k(X_i), f_k \in F, \quad (16)$$

onde F é o espaço da árvore de regressão

$$F = \{ f(X) = w_{q(X)} \}, \quad (17)$$

e K é o número de árvores. Cada f_k corresponde a uma estrutura de árvore independente q e pesos de folha w . A função objetivo é

$$L = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \quad (18)$$

onde l é a função de perda convexa diferenciável que mede a diferença entre a previsão \hat{y}_i e o alvo y_i , e Ω é o termo penaliza a complexidade do modelo, ou seja, as funções da árvore de regressão.

Chen e Guestrin (2016) também destacam que a utilização de matrizes esparsas e o aprimoramento de estruturas de dados possibilitou melhor processamento dos dados no treino e dos resultados da predição, além de reduzir consideravelmente o custo computacional.

Para se obter um modelo ótimo do XGboost, é imprescindível a configuração dos seus hiperparâmetros. Existem diversos deles, porém, serão destacados os mais importantes para o presente trabalho. Para controlar árvores de regressão no XGBoost, comumente são utilizados os hiperparâmetros conforme Tabela 4 a seguir.

Tabela 4: Principais hiperparâmetros do XGBoost e sua função.

Hiperparâmetros	Descrição
<i>n estimators</i>	É o número de árvores no modelo. Equivalente ao número de rodadas de reforço.
<i>max depth [padrão=6]</i>	Está relacionado a profundidade máxima das árvores e controla também seus níveis de subdivisões. Neste caso, o valor cada vez maior deste parâmetro torna o modelo mais complexo.
<i>learning_rate</i>	Diminui o tamanho do passo usado na atualização para evitar <i>overfitting</i> .
<i>Alpha [padrão=1]</i>	É responsável pela regularização L1 nos pesos das folhas. Pode ser usado em caso de dimensionalidade muito alta para que o algoritmo rode mais rápido quando implementado. Aumentar esse valor tornará o modelo mais conservador. Aplicado na regularização do XGBoost para reduzir o <i>overfitting</i> .
<i>Lambda [padrão=0]</i>	É responsável pela regularização L2 nos pesos das folhas, usado em caso de dimensionalidade muito alta para que o algoritmo seja executado mais rapidamente quando implementado.
<i>Subsample [padrão=1]</i>	É a proporção de subamostra das instâncias de treinamento
<i>early stopping rounds</i>	Realiza o controle de parada do aprendizado do algoritmo, após uma quantidade de interações (<i>rounds</i>) antes de alcançar o valor de <i>n_estimators</i> , que não contribuem para a redução do erro.

Fonte: Elaboração própria, 2022.

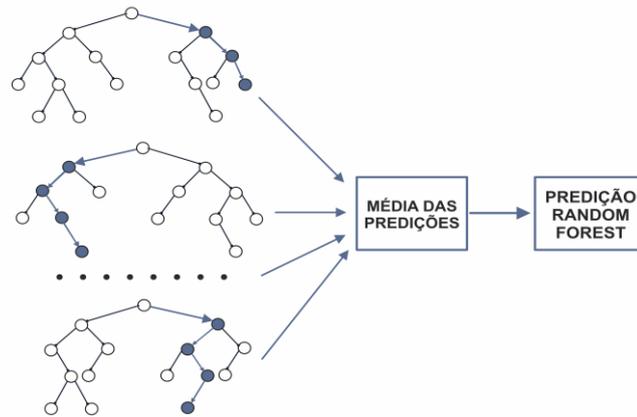
Chen e Guestrin (2016) afirmam que o fator que tornou o XGBoost muito utilizado nos últimos anos diz respeito a sua escalabilidade em todos os cenários, além da velocidade de sua execução, algumas vezes, mais de dez vezes mais rápido do que as soluções populares existentes em uma única máquina. Além disso, ele pode ser dimensionado para bilhões de exemplos em configurações distribuídas ou com memória limitada.

3.8 Algoritmo Random Forest

O algoritmo de Random Forest (RF) ou Floresta Aleatória, introduzido por Breiman (2001), é um termo geral usado para métodos de conjunto de dados utilizando classificadores do tipo árvore. Isso porque o algoritmo constrói uma grande quantidade de árvores de decisão de subconjunto de dados mediante um treinamento único definido, a partir de um meta-algoritmo para melhorar a classificação e a regressão de modelos conforme a estabilidade e precisão da classificação. Trata-se de um algoritmo de aprendizagem supervisionada que cria várias árvores de decisão e as combina para obter uma predição com maior acurácia e mais estabilidade. Para Biau (2012) este tipo de estrutura trouxe melhorias perceptíveis na precisão do aprendizado por classificação/regressão e se enquadra na esfera do aprendizado conjunto.

Conforme Iannace *et al.* (2019), cada árvore dentro de um RF é construída e treinada a partir de apenas um subconjunto aleatório dos dados no *dataset* (não o *dataset* completo), e neste caso, o melhor atributo não é selecionado para cada nó mas de um conjunto de atributos selecionados aleatoriamente. A aleatoriedade é um fator que passa a fazer parte da construção dos classificadores e visa aumentar sua diversidade e assim diminuir a correlação. A Figura 15 mostra um diagrama do algoritmo RF para regressão.

Figura 15: Algoritmo RF.



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

Existem vários hiperparâmetros que podem ser usados para configurar um algoritmo RF visando aumentar a precisão do modelo. Normalmente são utilizados os hiperparâmetros conforme Tabela 5 a seguir:

Tabela 5: Principais hiperparâmetros do RF e sua função.

Hiperparâmetros	Descrição
<i>n_estimators</i>	É o número de árvores no modelo. Usando <i>scikit-learn</i> , o número padrão para definição dos estimadores é 100.
<i>max_depth</i>	Está relacionado a profundidade, ou seja a altura máxima de crescimento das árvores dentro da floresta, podendo aumentar a precisão do modelo.
<i>min_samples_split</i>	Especifica o valor mínimo de amostras que um nó interno terá para se dividir em outros nós.
<i>min_samples_leaf</i>	Especifica o valor mínimo de amostras que um nó interno terá após ser dividido.
<i>max_features</i>	Busca encontrar a melhor divisão a partir dos subconjuntos aleatórios de recursos.
<i>max_samples</i>	Ajuda a escolher o valor máximo de amostras do <i>dataset</i> de treino para treinar individualmente cada árvore.

Fonte: Elaboração própria, 2022.

O desempenho de um algoritmo depende do ajuste de seus parâmetros e da seleção de variáveis. Nesse sentido, Tyralis e Papacharalampous (2017) apontam que existem vários estudos com procedimentos para estimar o conjunto ótimo de

parâmetros para o algoritmo RF. Esses procedimentos consideram o desempenho do algoritmo em função das variáveis selecionadas, do número de árvores, do número de possíveis direções de divisão em cada nó de cada árvore e o número de exemplos em cada célula.

3.9 IoT e Redes neurais profundas aplicadas a sistema de monitoramento de consumo de energia elétrica.

Conforme já mencionado anteriormente (Capítulo 1), existe uma crescente demanda em relação ao consumo de energia elétrica em todos os setores da sociedade, no entanto, é importante destacar que o aumento de geração, mesmo sendo imprescindível e estratégico, não pode ser avaliado como a única solução possível, pois fatores como o desperdício de energia elétrica também precisam ser abordados cuidadosamente.

Atualmente, existem várias formas de incentivo à redução de desperdício de energia, como os programas desenvolvidos pelos governos no mundo e projetos promovidos pelas concessionárias de distribuição de eletricidade. Outra forma desse controle que está sendo disseminado aos poucos e de forma eficaz, é o uso de medidores inteligentes (*Smart Meters*), que contribuem de forma significativa para conscientização e redução do desperdício. Um estudo realizado e publicado no Reino Unido, sobre o programa de implementação de medidores inteligentes (EISDB, 2019), apontou que cerca de 62% dos entrevistados afirmaram que os medidores inteligentes ajudaram a melhorar sua forma de consumir energia elétrica em suas residências; já 73% informaram que passaram a realizar suas atividades de forma mais eficiente e consciente.

De acordo com Atzori *et al.* (2010), as tecnologias para desenvolvimento de sistemas baseados em IoT estão agregando “inteligência” a uma enorme quantidade de equipamentos conectados à Internet, não mais se limitando especificamente a computadores e *smartphones*. No Plano Nacional de Internet das Coisas, Decreto nº 9.854, de 25 de Junho 2019 do Governo Federal (Brasil, 2019), com objetivo de estimular a implementação e desenvolvimento desta tecnologia no país, uma das metas é melhorar a qualidade de vida das pessoas promovendo ganhos de eficiência nos serviços em todos os sentidos.

Nesse contexto, torna-se importante o desenvolvimento de aplicações baseadas em IoT que permitam fornecer informações relevantes ao usuário sobre o consumo de energia em tempo real, tanto para um equipamento individual (ex: ar-condicionado, geladeira, máquina lava roupas, etc.), quanto para um prédio ou empresa com vários departamentos e equipamentos; bem como o uso de AP para realizar a predição do consumo para o mês seguinte (semana, ou dia), a fim de que o usuário possa tomar iniciativas visando controle sobre a utilização da energia elétrica com o objetivo de administrar o uso de maneira econômica.

Com base nisso, pesquisadores do Laboratório de Inteligência Computacional (LabIC) da Universidade Federal do Oeste do Pará (UFOPA) localizada na cidade de Santarém, Pará, Brasil, situada na região amazônica ocidental; realizaram pesquisas para desenvolvimento de um sistema (*framework*) que efetivasse a coleta de dados em equipamentos (ou prédio), com o objetivo de promover o monitoramento de energia elétrica e exibir os resultados em tempo real para o usuário, que passou a ser denominado EnergySaver. Nesse sentido, as pesquisas foram estruturadas da seguinte forma: 1) plataforma IoT: Varão *et al.* (2017), Varão *et al.* (2018), Teixeira *et al.* (2018); 2) Aquisição de dados: Amoedo *et al.* (2019); 3) Manual e Tutorial: Da Silva *et al.* (2021b); 4) Modelagem: Geller e Meneses (2021). Desse modo, julgou-se necessário o desenvolvimento de um módulo que utilizasse RNPs (neste caso, rede neurais BiLSTM) como parte integrante desse sistema, visando permitir ao usuário obter resultados da predição de consumo para o mês seguinte, com o objetivo de realizar possíveis tomadas de decisões quanto às ações para promoção da redução de consumo, por exemplo. O módulo descrito acima tornou-se objeto deste trabalho, bem como todas os conceitos a ele relacionados.

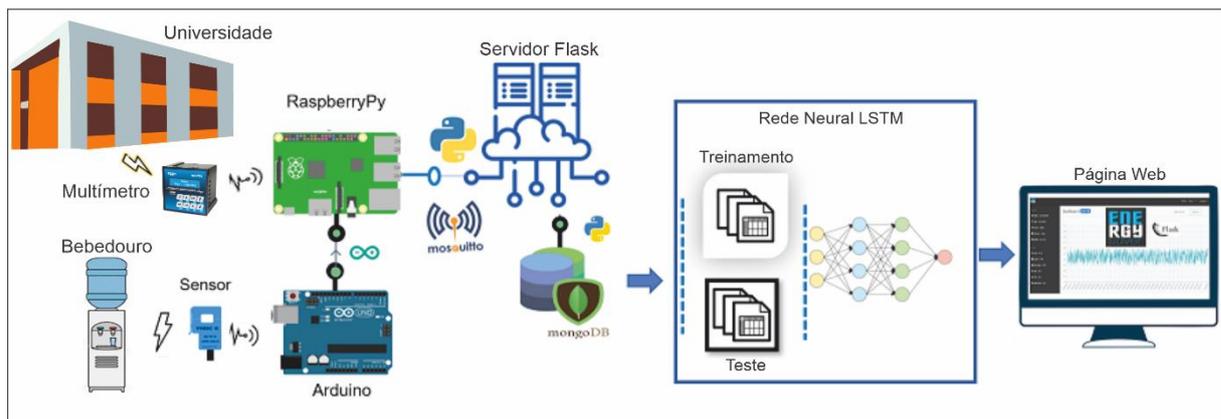
Na Seção 3.10 a seguir, tanto o sistema IoT EnergySaver quanto o desenvolvimento e implantação do módulo com redes neurais LSTM serão descritos detalhadamente.

3.10 O Sistema EnergySaver

Conforme mencionado anteriormente, o EnergySaver é um *framework* voltado para o monitoramento do consumo de energia, onde o sistema de captura de dados utiliza tecnologias *open source* aplicadas a IoT, sistemas embarcados e um módulo de predição com a rede neural LSTM.

Na Figura 16, o fluxo geral de operação do sistema é descrito da seguinte forma: inicialmente, os dados de consumo são capturados pelo multímetro (ou sensor de corrente elétrica e Arduino, caso seja equipamento individual), depois são enviados para o *Raspberry Pi3*, que por sua vez envia pelo *Mosquitto Broker* (MQTT) para o servidor *Flask*. No servidor, esses dados são armazenados no banco de dados *MongoDB* e exibidos em tempo real em uma página da *Web* existente no servidor local. Com os dados de consumo obtidos pelo servidor diariamente, todo primeiro dia de cada mês, a rede LSTM é treinada com dados dos meses anteriores e testada com os novos dados do mês atual. A partir disso, o sistema envia uma mensagem ao Administrador do Sistema com informações relacionadas ao consumo e à previsão para o mês seguinte.

Figura 16: Esquema conceitual do sistema EnergySaver.



Fonte: Adaptado de Geller e Meneses (2021).

Para modelar o EnergySaver foram utilizados diagramas da UML, considerado um padrão para modelagem de diferentes tipos de sistemas (GELLER e MENESES, 2021). A modelagem de sistemas IoT com UML ainda é alvo de pesquisas para a adaptação dessa linguagem visual às especificidades desses sistemas. Alguns trabalhos como Reggio (2018), Thramboulidis e Christoulakis (2016), Robles-Ramirez *et al.* (2017), consolidaram a UML como recurso para modelagem de sistemas IoT.

Um dos benefícios de utilizar uma linguagem já reconhecida como padrão é a facilidade de recursos disponíveis e a possibilidade de extensões para representar as características específicas de cada sistema. O desafio de modelar sistemas IoT com UML é representar um nível suficiente de detalhes para que os engenheiros os implementem enquanto abstraem as complexidades para que possam ser

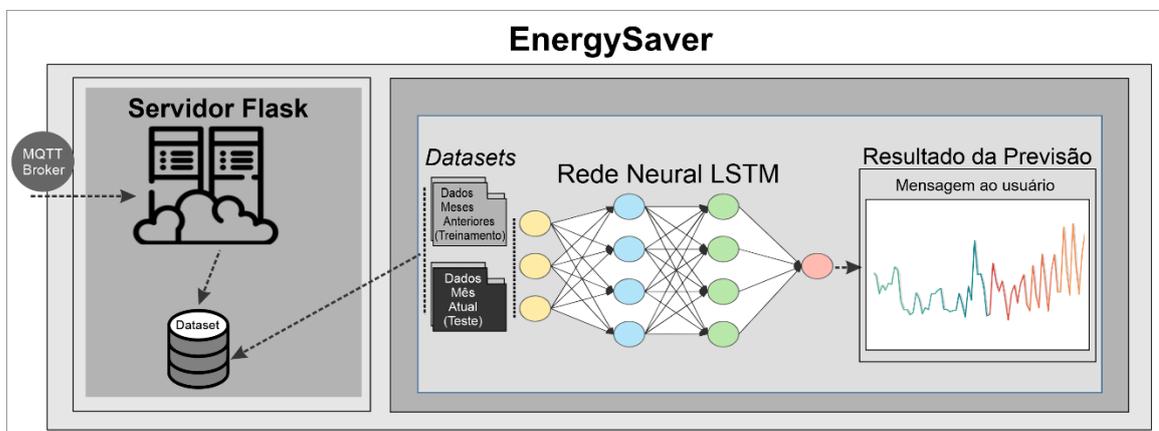
implantados com facilidade. Mais detalhes sobre a modelagem do EnergySaver podem ser encontrados em Geller e Meneses (2021). Em Da Silva *et al.* (2021a) são apresentadas as tecnologias utilizadas para o desenvolvimento do sistema, como Multímetro, Arduino com sensor de corrente elétrica, *Raspberry Pi*, *Mosquitto broker*, MongoDB e *micro-framework* Flask (mais informações no Apêndice A), além da criação de um módulo de predição para o sistema EnergySaver com RNPs.

3.10.1 Esquema módulo de predição proposto para o EnergySaver

Em Da Silva *et al.* (2021a) os autores apresentaram uma abordagem para criar um módulo de predição usando um modelo de rede LSTM, para o sistema IoT EnergySaver. Posteriormente, Da Silva *et al.* (2022) também realizaram a comparação estatística entre LSTM e os algoritmos XGBoost e RF para predição de consumo, onde conclui-se que a rede neural LSTM apresentou resultados estatisticamente superiores aos demais, mantendo assim essa RNP acoplada ao sistema.

O processo inicial de funcionamento do sistema IoT ocorre com a coleta de dados de consumo de energia elétrica em equipamento individual ou em um prédio, para criar um *dataset* a ser usado para treinamento, testes e predição com a rede LSTM. Com esses dados disponíveis no computador servidor e tratados de forma automatizada pelo módulo, a rede é treinada e testada para prever os próximos pontos (neste caso a predição foi definida para ser referente ao período de um mês). A Figura 17 mostra o fluxo geral do módulo LSTM.

Figura 17: Módulo LSTM proposto.



Fonte: Elaboração própria, 2022.

O módulo LSTM descrito na Figura 17 realiza tanto a limpeza de dados quanto a execução da rede LSTM para previsão. Este processo é realizado automaticamente no primeiro dia de cada mês. Os dados recebidos e armazenados no banco de dados devem passar por um processo de limpeza antes de serem utilizados pela rede para treinamento e teste.

Os resultados dos testes e validação deste módulo com redes neurais LSTM serão descritos detalhadamente no Capítulo 5 (Resultados), bem como a implementação do módulo no sistema IoT no Capítulo 6.

CAPÍTULO 4

4 METODOLOGIA PROPOSTA

Este capítulo caracteriza a pesquisa e os procedimentos metodológicos empregados para sua realização. O capítulo é dividido da seguinte forma: a Seção 4.1 traz a caracterização da metodologia de pesquisa; na Seção 4.2 é apresentada a abordagem metodológica adotada neste trabalho; a Seção 4.3 destaca o fluxo metodológico para a pesquisa; a Seção 4.4 descreve o estudo de caso e os *datasets* utilizados; na Seção 4.5 são detalhados os procedimentos para treinamento e testes; a Seção 4.6 descreve o pré-processamento dos dados; na Seção 4.7 são apresentadas as métricas de desempenho para avaliação dos resultados; na Seção 4.8 o detalhamento do projeto experimental computacional; e por fim, na Seção 4.9 são descritos os procedimentos para análise estatística.

4.1 Caracterização da Pesquisa

A presente pesquisa é caracterizada de acordo com as classificações de pesquisa descritas por Goretti e Biaggini (2014), que são: aplicada, quantitativa, prescritiva e bibliográfica. Do ponto de vista de sua natureza, a presente proposta de pesquisa é aplicada, pois visa gerar conhecimento para aplicação prática tendo como foco um problema específico. É quantitativa, pois pretende-se medir o índice de acerto do método proposto, comparando estatisticamente com outros métodos existentes na literatura. Sob o ponto de vista dos objetivos da pesquisa, é prescritiva, pois irá propor uma solução que fornece resposta direta ao problema em questão. A pesquisa é bibliográfica pois é elaborada a partir do estudo de abordagens semelhantes existentes na literatura para resolução do mesmo problema, tendo como base os mesmos procedimentos técnicos.

4.2 Abordagem Metodológica

Nesta pesquisa, optou-se pela utilização da abordagem metodológica DSR de Peffers *et al.* (2007), advinda da *Design Science* (DS) proposta na década de 60 por Fuller (1965) e Gregory (1966). A metodologia da DSR aborda princípios, práticas e

procedimentos necessários para realização de pesquisas científicas buscando atingir três objetivos, que são (PEFFERS *et al.*, 2007): 1) fornecer um processo metodológico para a realização de pesquisas em ciência; 2) basear-se em estudos da literatura científica e disciplinas de referência; e 3) fornecer aos pesquisadores um modelo específico para a estrutura e resultados obtidos pela pesquisa.

Conforme Peffers *et al.* (2007), o uso da metodologia DSR inclui seis etapas procedurais, que são sugeridas na ordem especificada na Figura 18, sendo elas: identificação e motivação do problema; definição dos objetivos da solução; projeto e desenvolvimento da solução; demonstração da solução, avaliação da solução e comunicação dos resultados.

Figura 18: Modelo de processo DSR.



Fonte: Adaptado de Peffers *et al.* (2007).

Cada etapa pode ser executada de acordo com a necessidade do projeto que está sendo desenvolvido. Assim, as etapas podem ser detalhas da seguinte forma (PEFFERS *et al.*, 2007): a) **Etapa 1** - Identificação do problema e sua motivação onde ocorre a definição do problema de pesquisa específico, apresentando-se uma justificativa para a sua investigação visando a construção de um artefato (DRESCH *et al.*, 2015); b) **Etapa 2** - Definição dos objetivos para a solução a ser desenvolvida, tendo-se como ponto de partida o conhecimento acerca do problema, bem como a noção do que é viável e factível; c) **Etapa 3** - Projetar e desenvolver o artefato, determinando-se a sua funcionalidade desejada; d) **Etapa 4** – Demonstração do uso do artefato resolvendo uma ou mais instâncias do problema por meio de um

experimento; e) **Etapa 5** – Avaliação do artefato quanto à solução do problema; f) **Etapa 6** – Comunicação e divulgação do problema e da relevância da propositura da solução.

De acordo Peffers *et al.* (2007), não é necessário que o pesquisador execute esta ordem sequencial de atividades, podendo começar em qualquer etapa. Assim sendo, se a ideia para a pesquisa resultar da observação do problema ou de pesquisas futuras sugeridas em um artigo de um trabalho anterior, o pesquisador pode avançar na sequência da metodologia proposta.

4.3 Fluxo Metodológico

Baseado na abordagem proposta por Peffers *et al.* (2007), a metodologia para elaborar esta pesquisa usou os princípios da DSR. No entanto, algumas etapas foram adaptadas e abstraídas para simplificar o diagrama que contém o fluxo metodológico, o qual é apresentado na Figura 19, seguido da definição de cada uma de suas principais etapas.

Figura 19: Fluxo metodológico DSR para a presente pesquisa.



Fonte: Elaboração própria, 2022.

A primeira etapa, nomeada **Identificação do Problema e Motivação**, resume-se a reconhecer a necessidade de aprimorar as abordagens que realizam a previsão em STs de consumo de energia elétrica, uma vez que diversas áreas podem se beneficiar de tal pesquisa e de seus resultados, tais como sistemas IoT que realizam coleta de dados em equipamento individuais ou edifícios. Nesta etapa, elementos iniciais como problematização, justificativa e objetivos são levantados.

Após a identificação do problema é iniciada a segunda etapa do fluxo metodológico, denominada **Definições Gerais**. Esta etapa engloba a delimitação do escopo do problema, através da busca por trabalhos relacionados e revisão da literatura já existente, com base na problemática a ser abordada pela pesquisa, bem como a identificação e proposição do artefato.

Na etapa de **Projeto e Demonstração da Solução**, o artefato de software é projetado e desenvolvido, a partir dos seguintes passos: a) projeto de desenvolvimento e validação do artefato proposto na etapa anterior. Foram utilizados quatro *datasets* (Seção 4.4) para avaliar e validar os resultados do modelo de rede LSTM assim que o mesmo seja desenvolvido; b) desenvolvimento do artefato: momento em que o artefato é de fato implementado, em que um modelo de RNP LSTM para previsão de STs de consumo de energia elétrica. Esta etapa está descrita no presente capítulo, e aborda os seguintes passos: 1) coleta e pré-processamento dos dados de consumo; 2) análise exploratória dos dados; 3) implementação de um modelo de rede neural LSTM; 4) treino e teste com validação cruzada para STs.

Após a realização do projeto e demonstração da solução, dá-se início à etapa de **Avaliação da Solução**, onde os resultados da rede LSTM com validação cruzada são mensurados estatisticamente (Seção 5.2). Além disso, é realizada a comparação estatística (Seção 5.3) entre resultados obtidos pela LSTM com os algoritmos de AM XGBoost e RF, utilizando dois *datasets* (UCI-Household e LABIC-Building) para treinamento e testes com VC-ST. Por fim, foi realizada a comparação estatística (Seção 5.5) entre resultados obtidos pela LSTM com a BiLSTM, utilizando quatro *datasets* (UCI-Household, LABIC-Building, Singapore e Tetouan-Zones) para treinamento e testes com VC-ST.

A última etapa que trata da **Disseminação** é importante para ressaltar a relevância da pesquisa, onde os resultados são compartilhados com a comunidade científica. A Seção 1.6 apresenta os resultados esperados nos âmbitos científicos, técnicos, acadêmicos e socioeconômicos. Já no capítulo 5, são apresentados os resultados alcançados.

Nas seções seguintes serão descritas as sequências metodológicas realizadas de forma preliminar, referente ao estudo de caso e *datasets*, treinamento e testes, pré-processamento dos dados, métricas de desempenho para avaliação, projeto experimental computacional e análise estatística.

4.4 Estudo de caso e *Datasets*

No presente trabalho, os experimentos foram conduzidos e avaliados em diferentes *datasets*, da seguinte forma: a) o consumo de uma residência individual; b) o consumo de um prédio universitário com diversos departamentos (salas de aula, auditórios, refeitório, laboratórios, entre outros); c) a demanda elétrica de Singapura; d) demanda elétrica de três zonas em uma cidade de Marrocos.

Os testes preliminares foram para encontrar o melhor modelo da rede neural LSTM (quanto ao números de camadas com VC-ST) utilizando inicialmente dois *datasets*. Os próximos experimentos com os mesmos *datasets* utilizando VC-ST, foram para realizar uma comparação estatística entre os resultados da LSTM, XGBoost e RF. Por fim, um terceiro experimento foi realizado para realizar a comparação estatística entre os resultados da LSTM e BiLSTM com os quatro *datasets*. Nas subseções seguintes, cada *dataset* será descrito detalhadamente.

4.4.1 *Dataset* UCI-Household

O *dataset* UCI-Household foi escolhido por ser amplamente utilizado na literatura para previsão de consumo de energia elétrica (Kim e Cho, 2018; Le *et al.*, 2019; Khan *et al.*, 2021), entre outros, para que sua reprodutibilidade e comparação com outros trabalhos possam ser verificadas.

Este *dataset* possui dados relativos ao consumo individual de eletricidade residencial de uma residência em Sceaux, França (*Individual household electric power consumption Data Set*), fornecidos pelo *UCI Machine Learning Repository*⁵, que é comumente usado por pesquisadores para a validação de suas arquiteturas de previsão de consumo de eletricidade. O período de coleta considerado é de dezembro de 2006 a novembro de 2010 (47 meses), apresentado em unidades de 1 minuto com um total de 2.075.259 pontos em quilowatts (kW).

Em nossos testes foi utilizada a ST univariada cujos dados referem-se à potência ativa global, sendo redimensionados para intervalos de 10 minutos (207.526 registros), onde foram utilizados os períodos de dezembro de 2006 a outubro de 2010

⁵ <https://archive.ics.uci.edu/ml/datasets/individual+household+electric+power+consumption>

(203.801 registros) para treinamento da rede e novembro de 2010 (3.725 registros) para validação.

4.4.2 *Dataset* LABIC-Building

O *dataset* LABIC-Building está relacionado a um dos prédios da universidade que o sistema IoT EnergySaver irá monitorar. Esta ST possui características como o consumo altamente influenciado pelos sistemas de ar condicionado, que possuem grande demanda na região amazônica, bem como a sazonalidade devido às atividades da Universidade.

Os dados do LABIC-Building foram obtidos na cidade de Santarém, Pará, Brasil, localizada na região amazônica ocidental. Os dados foram coletados por pesquisadores do Laboratório de Inteligência Computacional (LabIC) em um prédio da Universidade Federal do Oeste do Pará (UFOPA). O período de coleta foi de janeiro a agosto de 2019, por Amoêdo *et. al* (2020).

Os dados de consumo do período possuem 256.092 registros, com base na potência ativa global agregada em *Watts* (W), referente ao consumo total em cada período. Assim, após transformar os dados brutos em índices de forma fixa (por intervalos de 10 minutos), o *dataset* ficou com um total de 33.830 registros. Deste total, foram utilizados para validação cruzada 29.492 registros referentes aos meses de janeiro a julho de 2019, correspondendo a aproximadamente 87% do *dataset*, e para validação 4338 registros referentes ao mês de agosto de 2019, correspondendo a 13% do total.

4.4.3 *Dataset* Singapore

O *dataset* Singapore é uma ST de demanda elétrica em larga escala coletado em Singapura, fornecidos pelo site *Energy Market Company Pte Ltd* (EMC⁶). Os atributos presentes no *dataset* são: tipo de preço, data, período, preço (\$/MW) e demanda (MW).

⁶ <https://www.emcsg.com/marketdata/priceinformation>

Neste caso foi utilizada a ST univariada de demanda (em MW) com intervalos de 30 minutos. O período de coleta considerado é de 1º de janeiro de 2010 a 30 de março de 2011, com 21.701 registros.

4.4.4 *Dataset* Tetouan-Zones

O *dataset* Tetouan-Zones possui dados relativos ao consumo de três zonas de uma estação da rede de distribuição da cidade de Tetouan em Marrocos e já foi utilizado por várias abordagens na literatura para a validação de suas arquiteturas de previsão de consumo de eletricidade, como por exemplo Singh *et al.* (2018), Salam e El Hibaoui (2018) e Salam e El Hibaoui (2021).

O período de coleta considerado compreende a 01 de janeiro de 2017 a 30 de dezembro de 2017 (12 meses), apresentado em unidades de 10 minutos com um total de 52.416 pontos em quilowatts (kW).

Em nossos testes foi utilizada a ST univariada, cujos dados referem-se à demanda elétrica de três zonas de coleta de Tetouan, e foram utilizados os períodos de janeiro de 2017 a novembro de 2017 (48.096 registros) para treinamento da rede e dezembro de 2017 (4.320 registros) para validação.

4.5 Decomposição Automática de Séries Temporais

Os dados que constituem uma ST podem sofrer a influência de diversos fatores como as variações nas condições da natureza, fenômenos imprevisíveis, dentre outros. Esses fatores determinam os componentes das STs que precisam ser decompostos em tendência, sazonalidade e ruído. A decomposição de uma ST é um método que tem por objetivo dividir a série em componentes visando a facilitar a modelagem, além de funcionar como ferramenta de análise, podendo ser usada para informar modelos de previsão sobre o seu problema.

Existem vários métodos para decompor uma ST, como por exemplo a Decomposição Clássica, que consiste em um modelo univariado que utiliza formulações matemáticas simples para separar a série em outros componentes, visando utiliza-la para realizar previsões: (a) a tendência, que se refere à direção geral segundo a qual parece que o gráfico da série temporal se desenvolve em um longo intervalo de tempo; (b) o ciclo, que se refere às oscilações ao longo prazo ou aos

desvios em torno da reta de tendência; (c) a sazonalidade, que se refere a padrões idênticos ou quase, que uma série temporal parece obedecer; e (d) o termo aleatório, que aparece com flutuações de curto período, com deslocamento inexplicável.

A realização de uma decomposição clássica permite que a ST seja descrita a partir de dois modelos:

1. **Aditivo:** quando se supõe que os componentes da ST são somados para formar os dados, na forma: $y(t) = \text{Nível} + \text{Tendência} + \text{Sazonalidade} + \text{Ruído}$. Trata-se de um modelo linear onde as mudanças ao longo do tempo são consistentemente feitas na mesma quantidade. Para a sazonalidade a linearidade tem a mesma frequência (largura dos ciclos) e amplitude (altura dos ciclos).
2. **Multiplicativo:** que supõe que os componentes são multiplicados um pelo outro, na forma: $y(t) = \text{Nível} * \text{Tendência} * \text{Sazonalidade} * \text{Ruído}$. Esse modelo não é linear. As alterações aumentam ou diminuem com o tempo. Por exemplo, a sazonalidade não linear tem uma frequência e/ou amplitude crescente ou decrescente ao longo do tempo.

Neste trabalho, utilizou-se a decomposição clássica automática com modelo aditivo a partir da biblioteca *statsmodels*, um módulo Python que fornece ferramentas para análises estatísticas, que fornece uma implementação do método de decomposição a partir da função *seasonal_decompose()*. Essa função requer que seja especificado se o modelo é aditivo ou multiplicativo, e retorna um objeto de resultado. O objeto de resultado contém matrizes para acessar todos os componentes da decomposição. Essa função decompõe os dados em seus componentes que são: Tendência (trend), sazonalidade (seasonal) e resíduo (resid), além da ST padrão que poder ser exibida também.

Conforme descrito nas seções anteriores, para todos os *datasets* utilizou-se a ST univariada referente a potência ativa. Assim sendo, para a decomposição dos mesmos, foram analisados os resultados (descritos na Seção 5.1) referente ao modelo aditivo para tendência, sazonalidade e os resíduos.

4.6 Treinamento e Testes

4.6.1 Configurações do computador e dos algoritmos

Para a realização do treinamento e teste foi utilizado um computador com as seguintes configurações: processador Intel® Core™ i5 - 7400, 16GB de RAM e GPU Nvidia RTX 2080Ti com 11GB de VRAM.

Já os algoritmos utilizados nesta pesquisa (LSTM, BiLSTM, XGBoost e RF) foram implementados com Python usando a biblioteca Keras 2.6.0, Tensorflow 2.6.0 como *backend*, VC-ST com *TimeSeriesSplit* $k=10$ *fold*s. As principais configurações das redes neurais LSTM e BiLSTM utilizadas neste trabalho estão descritas na Tabela 6 a seguir.

Tabela 6: Configurações das redes LSTM e BiLSTM.

Parâmetro	Valor
<i>MinMaxScaler</i>	feature_range=(0,1)
<i>units</i>	100
<i>epochs</i>	100
<i>batch</i>	32
<i>dropout</i>	0.3
<i>optimizer</i>	<i>rmsprop</i>
Janela deslizante	90
<i>activation</i>	<i>linear</i>
<i>loss</i>	<i>mse</i>

Fonte: Elaboração própria, 2022.

Na tabela acima, o parâmetro “*MinMaxScaler*” foi utilizado para normalizar os valores de consumo de cada *dataset* para a entrada da rede, pois os mesmos possuem diferentes escalas (kW e MW). O otimizador RMSprop foi usado com base na abordagem de Reimers e Gurevych (2017) descrita na Subseção 3.3.3. O tamanho da janela deslizante com 90 observações foi definida empiricamente para treinamento, teste e validação, sendo aplicada a todas as arquiteturas, tendo em vista a observação dos resultados obtidos pela análise exploratória de dados (descrita na Seção 5.1), no que se refere às funções de *autocorrelação* (*autocorrelation function* - ACF) de potência ativa para cada *dataset*, onde pode-se concluir que uma janela deslizante

com 90 observações é adequada para treinamento e predição. Os demais parâmetros foram definidos a partir de testes empíricos.

As principais configurações do XGBoost e RF utilizadas neste trabalho estão descritas na Tabela 7 a seguir.

Tabela 7: Configurações do XGBoost e RF.

Parâmetro	Valor
<i>MinMaxScaler</i>	feature_range=(0,1)
<i>n_estimator</i> (XGBoost)	1000 (UCI-Household) 500 (LABIC-Building, Singapore e Tetouan-Zones)
<i>n_estimator</i> (RF)	100 para todos os <i>datasets</i>
janela deslizante	90

Fonte: Elaboração própria, 2022.

A maioria dos parâmetros de configuração do XGBoost e RF, foram definidos do mesmo modo que nas redes neurais LSTM e BiLSTM, exceto do *n_estimator* tendo em vista que foram realizados testes empíricos dos parâmetros para cada algoritmo e *dataset*, conforme descrito na tabela acima.

4.6.2 Janela Deslizante

Em Dietterich (2002), o autor fez uma revisão das principais técnicas aplicadas para resolver problemas de aprendizado sequencial supervisionado, e dentre elas destaca-se o método da janela deslizante (*sliding window*). Isso porque as STs são caracterizadas por serem formadas por observações sequenciais, e desse modo, é indispensável a utilização de uma técnica que mantenha a ordem de coleta dos dados ao mesmo tempo que transforme as entradas em um formato utilizável por algoritmos de aprendizagem supervisionada.

De acordo com Dietterich (2002), o método janela deslizante converte o problema de aprendizagem supervisionada sequencial em um problema de aprendizagem supervisionada clássica. O autor destaca que esse método é o mais adequado para ser usado em dados de STs, tendo em vista a possibilidade de ajustar o modelo a eventuais ciclos e tendências da série de dados.

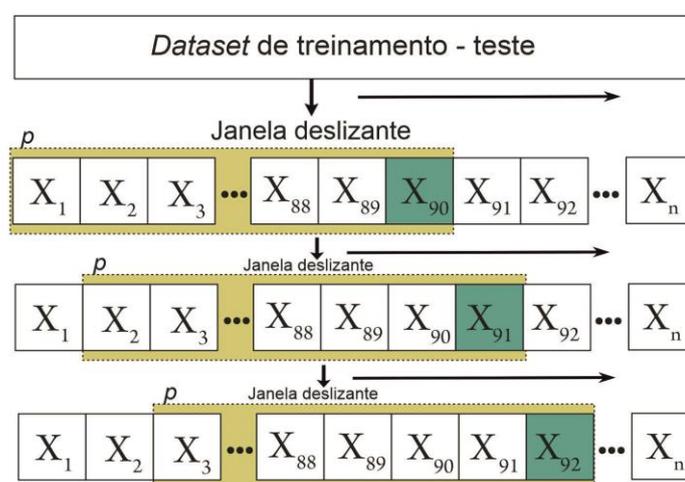
Esse método pode ser descrito da seguinte forma (DIETTERICH, 2002): seja um *dataset* sequencial $K = \{(x_t, y_t)\}_{t=1}^n$ com n exemplos, onde cada par (x_t, y_t) é formado por um vetor de atributos x_t e uma variável dependente y_t para um determinado tempo t . O classificador de janela deslizante é treinado convertendo cada exemplo de treinamento sequencial (x_t, y_t) em janelas. Assim sendo, a janela deslizante irá mapear o conjunto dos últimos w atributos $x_{t-w}, \dots, x_{t-1}, x_t$ até alcançar o instante t , para uma saída y_t . O w é o tamanho da janela e diz respeito ao número de instantes passados que se deseja utilizar.

A implementação desse método requer a definição de três hiperparâmetros, que são: a) tamanho da janela de treinamento- refere-se ao número de pontos de dados incluídos em uma passagem de treinamento; b) tamanho da janela de previsão- está relacionado ao número de pontos de dados a serem incluídos na previsão; c) passos de deslizamento- diz respeito ao número de pontos de dados saltados de um passe para outro.

De forma mais prática, essa técnica é utilizada inicialmente definindo uma janela de treinamento X_{sw} e um passo p . Em seguida, realiza-se o treinamento do modelo com os dados de X_1 a X_{sw} avançando uma quantidade p de registros, fazendo o treinamento com os dados de p a $X_{sw} + p$, prosseguindo até que todos os dados X_n existentes no *dataset* de treinamento sejam utilizados.

A Figura 20 apresenta um exemplo da uso da janela deslizante, onde a janela de treinamento $X_{sw} = 90$, o passo $p = 1$ e X_n representa a quantidade de dados existentes no *dataset*.

Figura 20: Exemplo treinamento-teste com janela deslizante.



Fonte: Elaboração própria, 2022.

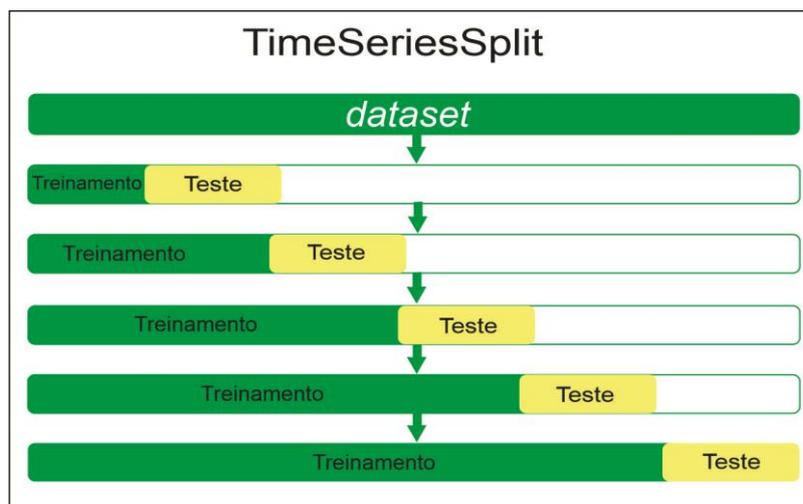
Para Dietterich (2002), o uso da janela deslizante faz com que os algoritmos de AM encontrem combinações lineares ou não-lineares entre atributos de tempos diferentes, contribuindo assim para a melhora do desempenho dos algoritmos ao desempenhar uma tarefa.

4.6.3 Validação cruzada de séries temporais

Uma forma de avaliar o comportamento de um modelo é usar a validação cruzada (KOHAVI, 1995; DANGETI, 2017). No entanto, em uma VC *KFold* convencional, os dados de treinamento e teste são selecionados aleatoriamente (por exemplo a função *KFold* da biblioteca Scikit-learn⁷). Tais VCs típicos, como os usados em problemas de classificação, não são aplicáveis no caso da ST de consumo (como é o caso deste trabalho), devido à dependência temporal dos dados. Assim, o *dataset* de treinamento deve conter apenas dados observados em período anterior aos dados existentes no *dataset* de teste (HYNDMAN e ATHANASOPOULOS, 2018).

No presente trabalho, utilizamos então função *TimeSeriesSplit* da biblioteca Scikit-learn, que diferente do *KFold* convencional, realiza a partição de uma série temporal em $n+1$ partes, onde n é o número de partições informado, buscando encontrar um número ótimo de elementos a serem usados para o treinamento do modelo, conforme mostrado na Figura 21.

Figura 21: Exemplo de validação cruzada com 5-folds de ST com *TimeSeriesSplit*.



Fonte: Adaptado do site scikit-learn.org⁸.

⁷ <https://scikit-learn.org>

⁸ Disponível em: https://scikit-learn.org/stable/modules/cross_validation.html. Acesso em: 10 fev. 2020.

No exemplo da Figura 21, em cada execução de treinamento/teste, os dados de treinamento são sempre anteriores aos dados de teste, de modo que a dependência do tempo é respeitada. O eixo horizontal representa o tamanho do *dataset* e o eixo vertical representa as iterações da VC. Em nosso trabalho a VC com *TimeSeriesSplit* foi realizado com 10 partições ($k=10$), pois Witten e Frank (2005) afirmam que testes extensivos realizados em diferentes *datasets* mostraram que 10 é um valor próximo ao número de partições em que as melhores estimativas podem ser obtidas. Para cada *fold TimeSeriesSplit* os melhores modelos LSTM e BiLSTM gerados foram salvos no formato “.h5”, bem como os melhores modelos do algoritmo XGBoost no formato “.model” e RF no formato “.joblib”.

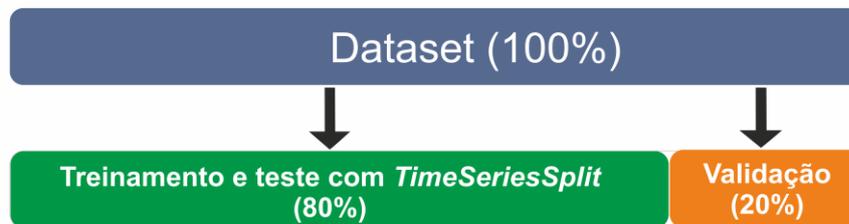
4.7 Pré-processamento dos dados

O pré-processamento dos dados existentes nos quatro *datasets* foi realizado usando a biblioteca Pandas⁹, eliminando os registros com dados nulos, valores ausentes (*missing data*) e caracteres com tipos de dados diferentes dos dados pré-estabelecidos para o campo. Dentre outros, os comandos mais comuns para essa verificação são: 1) *df.describe()* - retorna um resumo estatístico das variáveis numéricas; 2) *df.info()* - apresenta um resumo de valores não nulos encontrados; 3) *df.isnull().sum()* - retorna a soma dos valores nulos encontrados; 4) *df.dtypes* – descreve o tipo de cada *feature*. Outra modificação a ser destacada foi no *dataset* LABIC-building, em que os dados de 10 segundos foram modificados para intervalos de 10 minutos.

Em cada *dataset* os dados foram divididos em treinamento-teste e dados de validação, onde os primeiros 80% dos dados de cada *dataset* foram utilizados para treinamento e teste, e os 20% restantes para validação dos modelos treinados, conforme proposto na abordagem de Dangeti (2017). Um exemplo dessa divisão é apresentado na Figura 22.

⁹ <https://pandas.pydata.org/pandas-docs/stable/reference/frame.html>

Figura 22: Divisão dos dados de cada *dataset* em treinamento-teste e validação.



Fonte: Adaptado de Dangeti (2017).

É importante destacar que os dados de validação (20%) de cada *dataset*, não são utilizados na etapa de treinamento-teste dos modelos, evitando resultados tendenciosos (caso seu uso tenha sido realizado nessa etapa), tendo em vista que esses dados servem para verificar a imparcialidade dos modelos.

Neste trabalho, como objetivo foi realizar a predição dos dados de consumo para o último mês da coleta de cada *dataset*, essa divisão foi realizada por meses, ou seja, o *dataset* utilizado na fase de treinamento-testes estava sempre relacionado aos dados de consumo de todos os meses anteriores e o *dataset* de validação com dados do último mês de coleta.

Vale ressaltar que em nossos experimentos, utilizamos apenas os conceitos da abordagem de Dangeti (2017), no que diz respeito a separação entre dados de treinamento-teste e validação, tendo em vista que esses dados variam de acordo com o tamanho do *dataset*, ou seja, os dados de validação terão sempre uma quantidade de pontos similar para todos os *datasets* (pois trata-se de apenas um mês), já os dados de treinamento-teste são bem distintos por se tratar de períodos meses ou anos dependendo do *dataset*.

A normalização dos dados foi realizada usando o normalizador *MinMaxScaler*¹⁰ da *Scikit-learn*, que utiliza o mínimo e o máximo de cada atributo para normalizar seus dados, tendo em vista que conforme Bouktif *et al.* (2018), os modelos de aprendizado de máquina são sensíveis a escalas de entrada. Em nossa pesquisa, utilizamos normalizações na faixa de 0 e 1 para todos os algoritmos, conforme já detalhado na Subseção 4.5.1.

¹⁰ <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>

4.8 Métricas de desempenho para avaliação dos resultados

Em nossos testes foram utilizadas cinco métricas de desempenho para predição de STs, a saber, RMSE, NRMSE (*Normalized Root Mean Square Error*), MAE, MAPE e o coeficiente de determinação R^2 . Essas métricas são calculadas pelas Equações a seguir.

$$RMSE = \sqrt{\left[\frac{1}{n} \sum_{i=1}^n (|y_i - \hat{y}_i|)^2\right]} \quad (19)$$

$$NRMSE = \frac{RMSE}{\bar{y}} \quad (20)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (21)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| * 100 \quad (22)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (23)$$

Onde y_i , \hat{y}_i , e n , são respectivamente o valor medido, o valor previsto e o número total de amostras, e \bar{y} é o valor médio da saída observada sobre as amostras existentes. RMSE é o desvio padrão dos erros de previsão, que indica o quanto os pontos de dados estão espalhados da linha de regressão. NRMSE é uma função onde se calcula o erro quadrático médio normalizado em relação ao valor absoluto entre os valores previstos e observados. O MAE mede a magnitude média dos erros de previsão e ignora suas direções apresentando a média das diferenças absolutas entre os valores previstos e reais para todas as instâncias do *dataset* de teste, considerando que todas as diferenças individuais têm o mesmo peso. MAPE é uma métrica relativa, que expressa o valor médio do erro relativo como uma porcentagem dos dados reais. O coeficiente R^2 é usado para avaliar o desempenho de um modelo de regressão linear, resultando na quantidade de variação no atributo dependente de saída que é previsível a partir das variáveis independentes de entrada.

4.9 Projeto computacional

Os experimentos computacionais foram divididos em três grupos: a) experimento preliminar com camadas LSTM usando VC-ST, para verificar se há diferença estatisticamente significativa entre os modelos quanto ao número de camadas usando os *datasets* UCI-Household e LABIC-Building; b) experimento com VC-ST para comparar o desempenho da rede LSTM com os algoritmos XGBoost e RF, nos mesmos *datasets*; c) experimento com VC-ST para comparar o desempenho da rede LSTM com BiLSTM utilizando os *datasets* UCI-Household, LABIC-Building, Singapore e Tetouan-Zones.

Nos testes preliminares em (a), para cada *dataset* (UCI-Household e LABIC-Building), quatro arquiteturas LSTM (número de camadas) foram utilizadas com 2, 4, 6 e 8 camadas (descritos como L2, L4, L6 e L8). Para cada treinamento da VC-ST com 10 *folds*, um modelo foi gerado e salvo (10 no total), sendo aplicado ao conjunto de treinamento e teste, gerando 10 métricas MAE e RMSE cada. Por se tratar de testes preliminares, apenas duas métricas foram avaliadas, com base nos valores previstos normalizados.

Como não houve diferença estatisticamente significativa entre os modelos nos testes preliminares, o modelo L2 (ver resultados na Seção 5.2.1) foi escolhido para os experimentos de treinamento e testes posteriores, pois é o modelo mais rápido a ser treinado pela rede, com menor número de pesos, e também foi usado para testes, inclusive, com o *datasets* Singapore e Tetouan-Zones.

4.10 Análise estatística

4.10.1 Análise estatística para os experimentos preliminares

Visando determinar o melhor modelo e assumindo que os resultados não possuem distribuição normal, foram aplicados testes estatísticos não paramétricos de Kruskal-Wallis (abordagem de Dmitrienko *et al.*, 2007) para comparar três ou mais grupos de dados. Este teste é usado quando você deseja testar a hipótese de que várias amostras têm a mesma distribuição. A hipótese nula aponta que não há diferença estatisticamente significativa entre os grupos ($p > 0,05$). A hipótese alternativa indica que pelo menos um grupo é diferente ($p < 0,05$). Assim, o teste *post-*

hoc de Dunn (abordagem de Dmitrienko *et al.*, 2007) é necessário para comparação pareada entre os modelos quando há diferença estatisticamente significativa no teste de Kruskal-Wallis. Para os testes estatísticos foi utilizada a métrica RMSE.

Os testes de Kruskal-Wallis foram realizados na plataforma SAS¹¹ *OnDemand for Academics*, com o procedimento NPAR1WAY. Os testes *post-hoc* de Dunn foram realizados na mesma plataforma, com uma macro SAS (abordagem de Dmitrienko *et al.*, 2007). O limiar de 0,05 foi adotado para os testes estatísticos.

4.10.2 Análise estatística para os experimentos com redes neurais LSTM e BiLSTM

Visando determinar o melhor modelo e assumindo que os resultados não possuem distribuição normal, foram aplicados testes estatísticos não paramétricos de Friedman (1937) (baseado na abordagem de García *et al.*, 2010a), cujo objetivo é determinar se é possível concluir, a partir de uma amostra de resultados, se há diferença entre o desempenho dos métodos testados. Este teste estatístico realiza a comparação de resultados dentro cada conjunto por meio de um método de atribuição de rankings.

O primeiro passo para realizar o teste estatístico é converter os resultados originais em rankings. Tais rankings variam de 1 a k , sendo k o número de métodos testados, para cada problema, separadamente. Assim sendo, são classificados os k algoritmos de acordo com cada *dataset* N separadamente, gerando um *rank* conforme o desempenho desse algoritmo. Em caso de empate, um *rank* médio é atribuído (DEMŠAR, 2006). Tendo em vista que o teste de Friedman mostra os *rank*s médios de acordo com a classificação de cada algoritmo, é possível apenas identificar se existe ou não diferença estatística entre esses algoritmos.

O teste de Friedman é dado conforme a Equação abaixo (DEMŠAR, 2006):

$$\chi_F^2 = \frac{12N}{K(K+1)} \left[\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right] \quad (24)$$

Onde χ_F^2 representa a estatística de Friedman, de acordo com $k-1$ graus de liberdade, N é a quantidade de *datasets*, k é a quantidade de algoritmos e R_j é o *rank* médio desses algoritmos.

¹¹ <https://welcome.oda.sas.com/home>

Para se testar algo é necessário estabelecer uma hipótese nula e outra alternativa, sendo ambas antagônicas. Neste teste estatístico, sua hipótese nula indica que os algoritmos têm um comportamento semelhante, ou seja, possuem uma distribuição normal ($p > 0,05$). Porém, se sua hipótese nula for rejeitada, pelo menos um algoritmo difere dos demais ($p < 0,05$), exigindo um teste *post hoc*.

Assim sendo, o teste *post hoc* de Nemenyi (abordagem de Barrow *et al.*, 2013) foi usado para determinar quais grupos de algoritmos têm diferenças estatisticamente significativas. Nesse teste, se a diferença no *ranking* médio entre dois algoritmos for maior que uma diferença crítica, o teste indicará qual deles obteve melhor desempenho com base em seu ranking com nível de confiança α .

Os testes de Friedman foram realizados utilizando a linguagem R (*software R for Windows front-end*¹²), com o procedimento “friedman.test”. Os testes *post-hoc* de Nemenyi foram realizados no mesmo software, com a função `frdAllPairsNemenyiTest`. O limiar de 0,05 foi adotado para os testes estatísticos.

No capítulo 5 a seguir, serão apresentados os resultados gerais obtidos pela presente pesquisa.

¹² <https://www.r-project.org/>

CAPÍTULO 5

5 RESULTADOS

Este capítulo detalha os resultados obtidos a partir dos testes usando a metodologia proposta, sendo dividido da seguinte forma: a Seção 5.1 apresenta a análise exploratória dos dados para quatro *datasets*; na Seção 5.2 os resultados preliminares da rede neural LSTM (VC-ST) com os *datasets* UCI-Household e LABIC-Building são detalhados; a Seção 5.3 descreve os resultados dos experimentos com VC-ST entre a LSTM, XGBoost e RF para os *datasets* testados na Seção anterior; a Seção 5.4 aborda a implantação do módulo LSTM; a Seção 5.5 apresenta a comparação estatística entre as redes neurais LSTM e BiLSTM com dados de validação entre quatro *datasets*; e, por fim, a Seção 5.6 apresenta os principais resultados científicos, técnicos e acadêmicos.

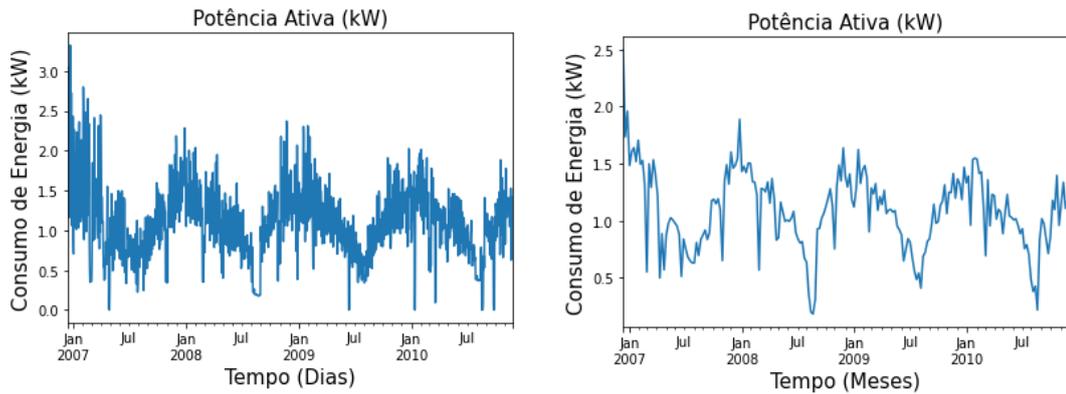
5.1 Análise Exploratória de Dados

Nas subseções seguintes, serão apresentados os resultados obtidos pela análise exploratória dos dados existentes nos *datasets* UCI-Household, LABIC-Building, Singapore e Tetouan-Zones, visando mostrar o comportamento das STs.

5.1.1 *Dataset* UCI-Household

Os dados existentes no *dataset* UCI-Household são referentes ao consumo individual de eletricidade residencial, coletados de dezembro de 2006 a novembro de 2010. A Figura 23 exibe a ST por dia e mês da potência ativa deste *dataset*.

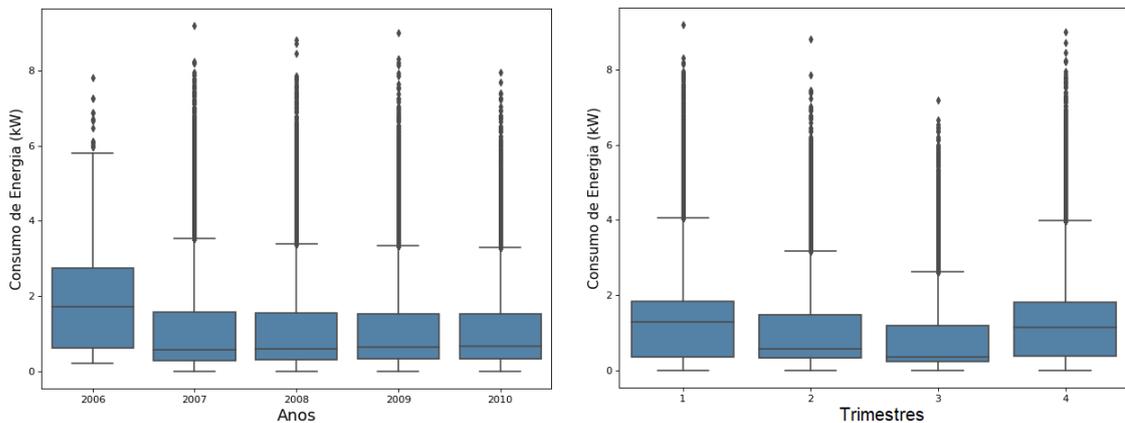
Figura 23: Amostras dos dados de consumo elétrico do *dataset* UCI-Household.



Fonte: Elaboração própria, 2022.

A Figura 24 apresenta os *boxplots* visando verificar a distribuição dos dados relativos ao consumo. Nele são apresentadas as estatísticas referentes à potência ativa, e revelam que o consumo de energia elétrica médio é quase constante ao longo dos anos, enquanto o gráfico trimestral mostra menor consumo no segundo e terceiro em relação aos demais, pois a demanda por potência na França aumenta no primeiro e quarto trimestres devido à carga de aquecimento quando é inverno.

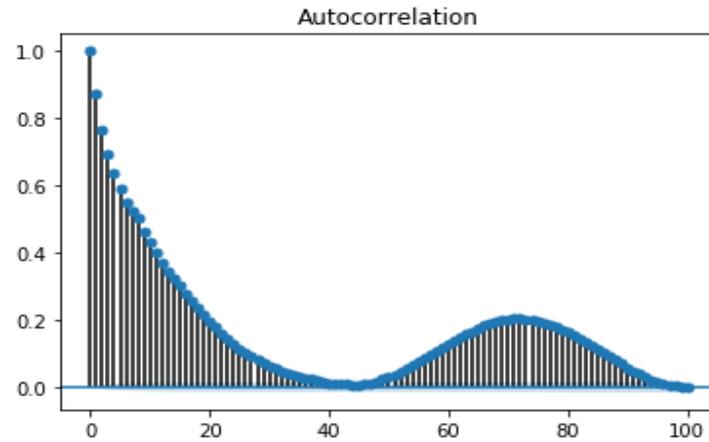
Figura 24: *Boxplots* de consumo de energia elétrica do *dataset* UCI-Household.



Fonte: Elaboração própria, 2022.

A Figura 25 exibe a ACF da potência ativa, onde a observação em um dado instante está relacionada a observações passadas. De acordo com a Figura 25, pode-se observar que uma janela deslizante com 90 observações é adequada para treinamento e previsão (ver também as Figuras 29, 33 e 37).

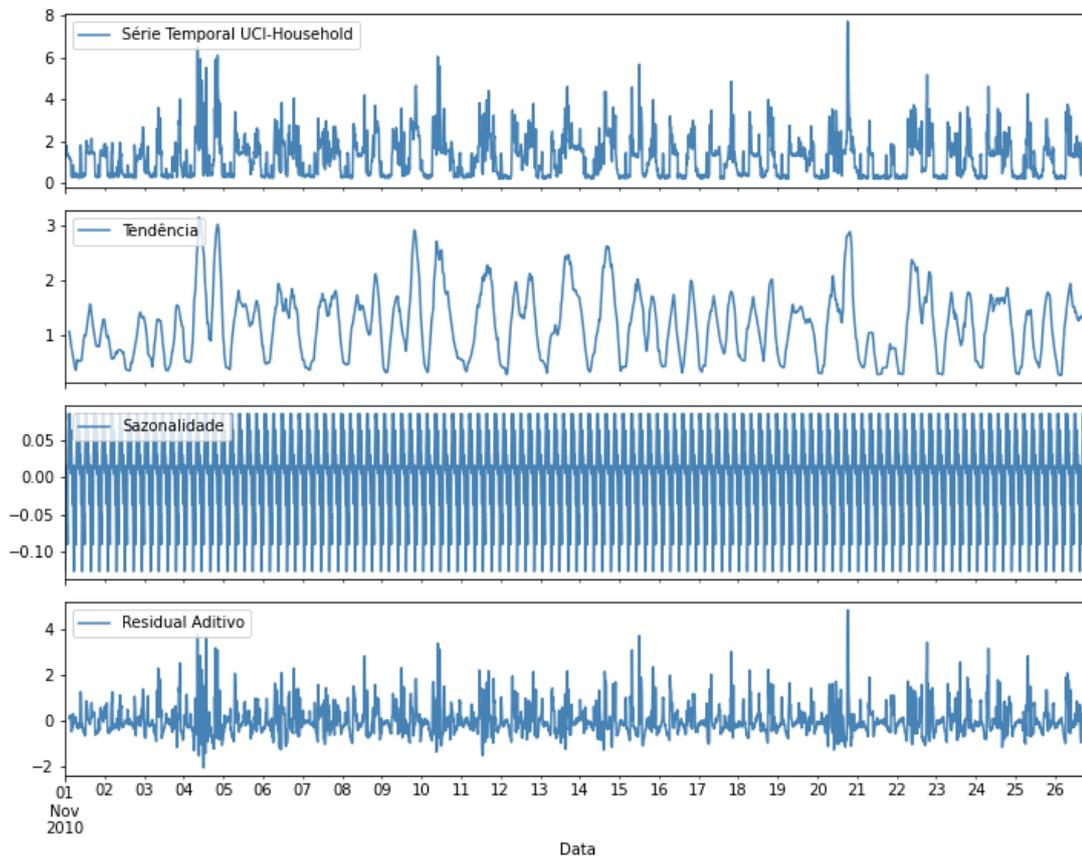
Figura 25: Função de autocorrelação no *dataset* UCI-Household.



Fonte: Elaboração própria, 2022.

A Figura 26 exibe os resultados da decomposição clássica automática da ST univariada da potência ativa para o *dataset* de validação referente ao mês de Novembro de 2010, utilizando o modelo aditivo para tendência, sazonalidade e resíduos.

Figura 26: Decomposição clássica da ST para o *dataset* UCI-Household.

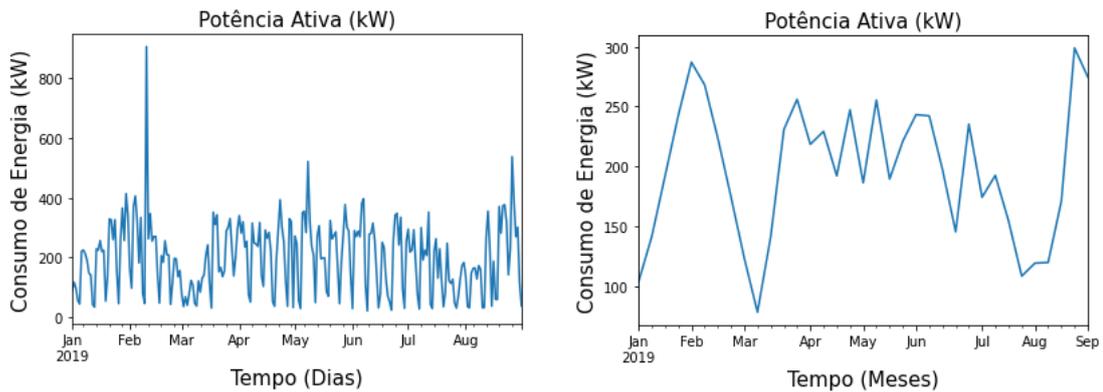


Fonte: Elaboração própria, 2022.

5.1.2 Dataset LABIC-Building

Os dados existentes no *dataset* LABIC-Building foram coletados em um prédio da UFOPA de janeiro a agosto de 2019. A Figura 27 exibe as amostras por dia e mês, sendo que os dados apresentados referem-se à potência ativa global.

Figura 27: Amostras dos dados de consumo elétrico do *dataset* LABIC-Building.

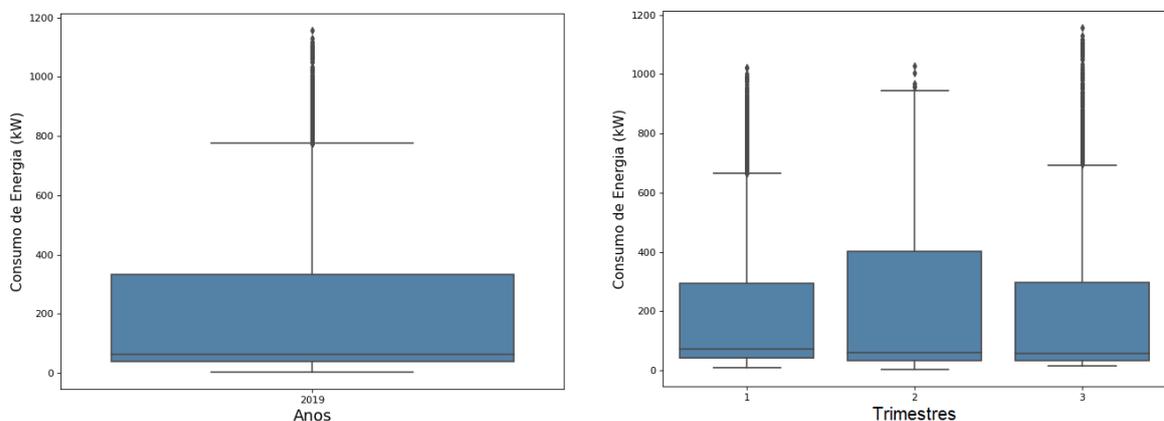


Fonte: Elaboração própria, 2022.

Na Figura 27, há algumas variações relacionadas aos dias de semana, que apresentam maior consumo, enquanto a demanda nos finais de semana é menor. Existem também alguns picos de consumo (*outliers*) que se desviam da média.

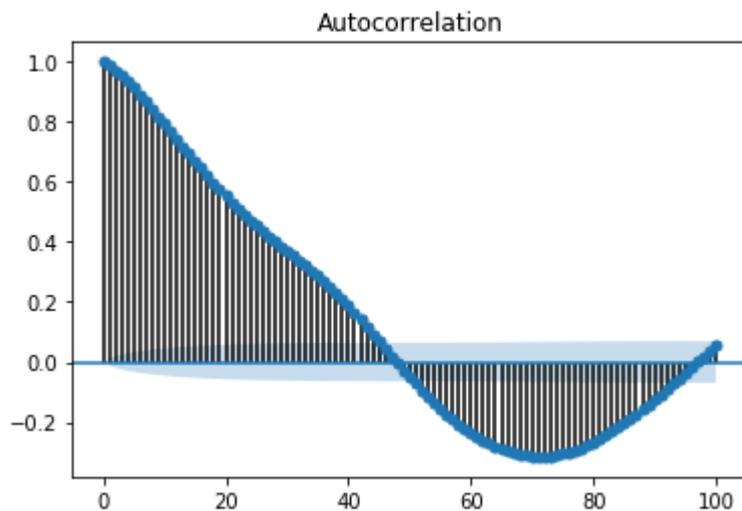
A Figura 28 apresenta os *boxplots* com o objetivo de verificar a distribuição dos dados relativos à potência ativa onde podem ser observados valores discrepantes elevados. Já a Figura 29 apresenta a ACF da potência ativa, corroborando para uma janela deslizante com 90 observações.

Figura 28: *Boxplots* de consumo de energia elétrica do *dataset* LABIC-Building.



Fonte: Elaboração própria, 2022.

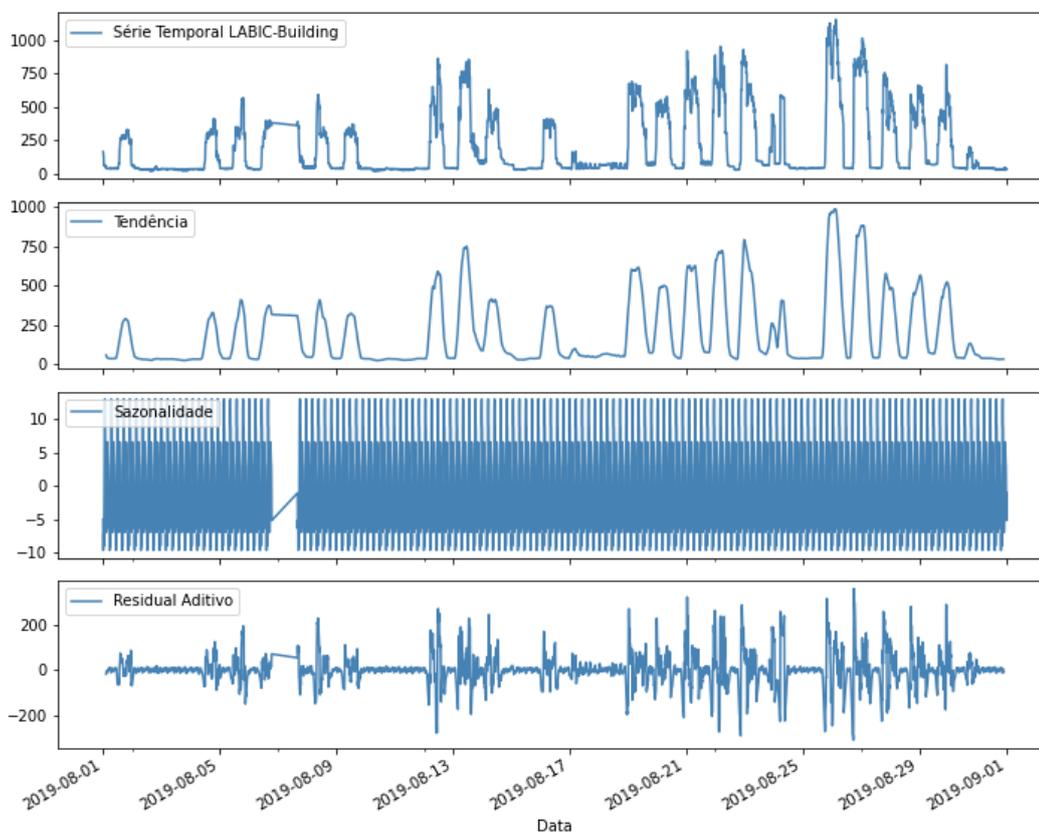
Figura 29: Função de autocorrelação no *dataset* LABIC-Building.



Fonte: Elaboração própria, 2022.

A Figura 30 exibe os resultados da decomposição clássica automática da ST univariada da potência ativa para o *dataset* de validação referente ao mês de Agosto de 2019, utilizando o modelo aditivo para tendência, sazonalidade e resíduos.

Figura 30: Decomposição clássica da ST para o *dataset* LABIC-Building.

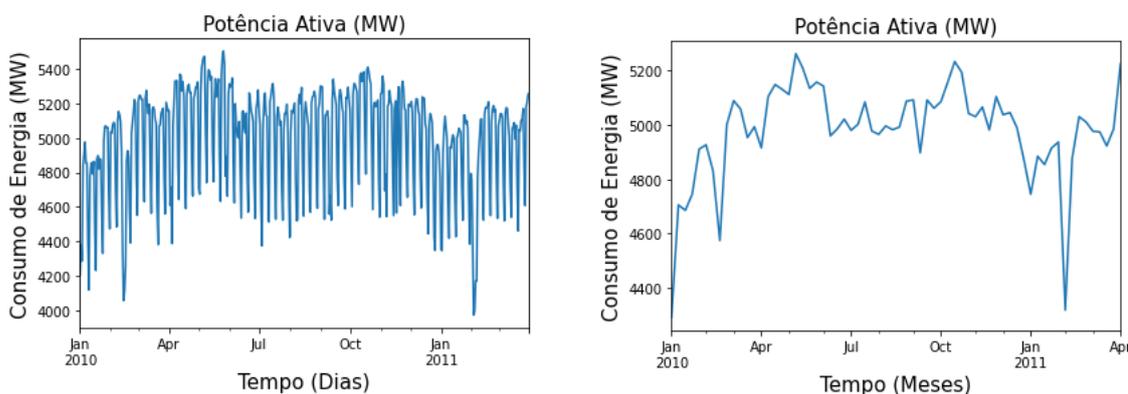


Fonte: Elaboração própria, 2022.

5.1.3 Dataset Singapore

Os dados existentes no *dataset* Singapore são referentes a demanda elétrica em larga escala coletados em Singapura, no período de janeiro de 2010 a março de 2011. A Figura 31 exhibe a ST por dia e mês da potência ativa deste *dataset*.

Figura 31: Amostras dos dados de consumo elétrico do *dataset* Singapore.

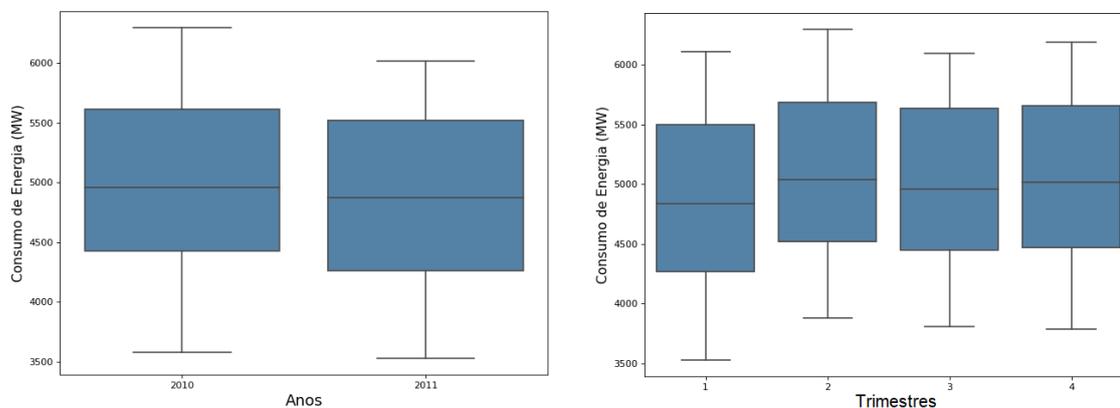


Fonte: Elaboração própria, 2022.

Os dados apresentados na Figura 31 também fornecem informações referentes aos dias e meses de consumo de energia elétrica, que indicam o comportamento da ST em todo o período.

A Figura 32 apresenta os *boxplots* com o objetivo de verificar a distribuição dos dados relativos à potência ativa.

Figura 32: *Boxplots* de consumo de energia elétrica do *dataset* Singapore.

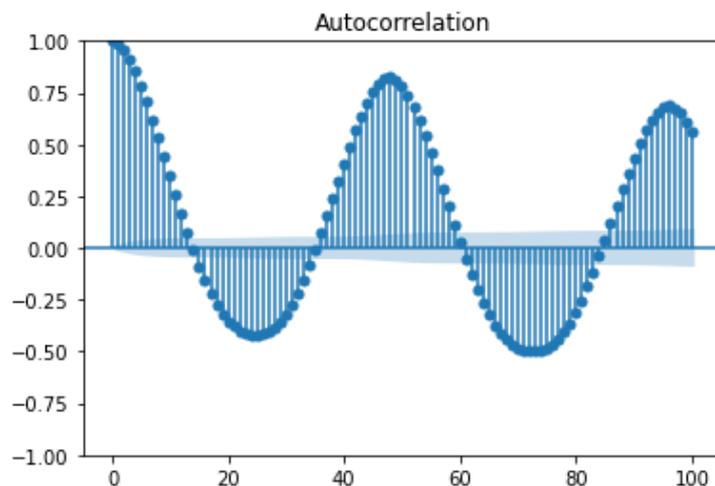


Fonte: Elaboração própria, 2022.

Na Figura 32 foram apresentadas as estatísticas referentes à potência ativa, e revelam que o consumo de energia elétrica médio é quase constante ao longo dos anos, enquanto o gráfico trimestral mostra menor consumo no primeiro e terceiro em relação aos demais.

A Figura 33 apresenta a ACF da potência ativa para o *dataset* Singapore.

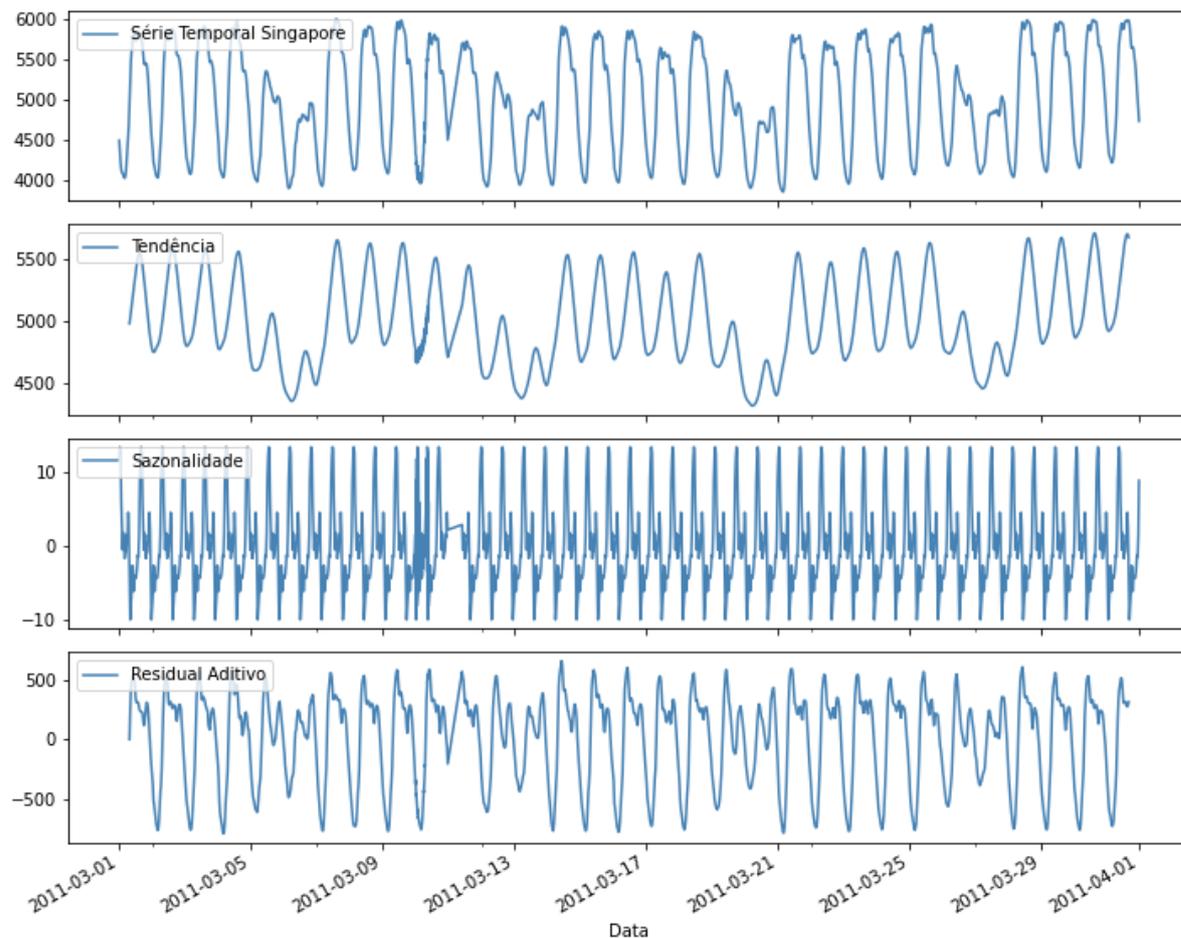
Figura 33: Função de autocorrelação no *dataset* Singapore.



Fonte: Elaboração própria, 2022.

A Figura 34 exibe os resultados da decomposição clássica automática da ST univariada da potência ativa para o *dataset* de validação referente ao mês de Março de 2011, utilizando o modelo aditivo para tendência, sazonalidade e resíduos.

Figura 34: Decomposição clássica da ST para o *dataset* Singapore.

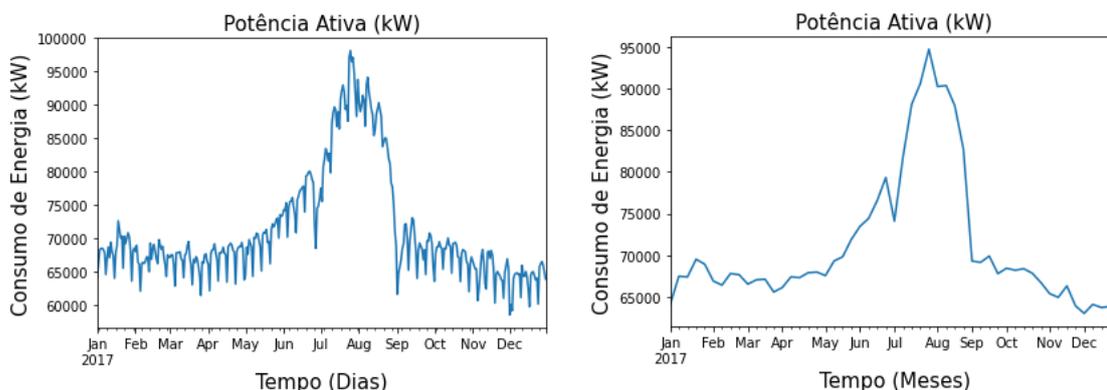


Fonte: Elaboração própria, 2022.

5.1.4 *Dataset* Tetouan-Zones

Os dados existentes neste *dataset* são relativos ao consumo de três zonas de uma estação da rede de distribuição da cidade de Tetouan em Marrocos, nos períodos de janeiro de 2017 a dezembro de 2017. A Figura 35 exhibe a ST por dia e mês da potência ativa (demanda) deste *dataset*.

Figura 35: Amostras dos dados de consumo elétrico do *dataset* Tetouan-Zones.

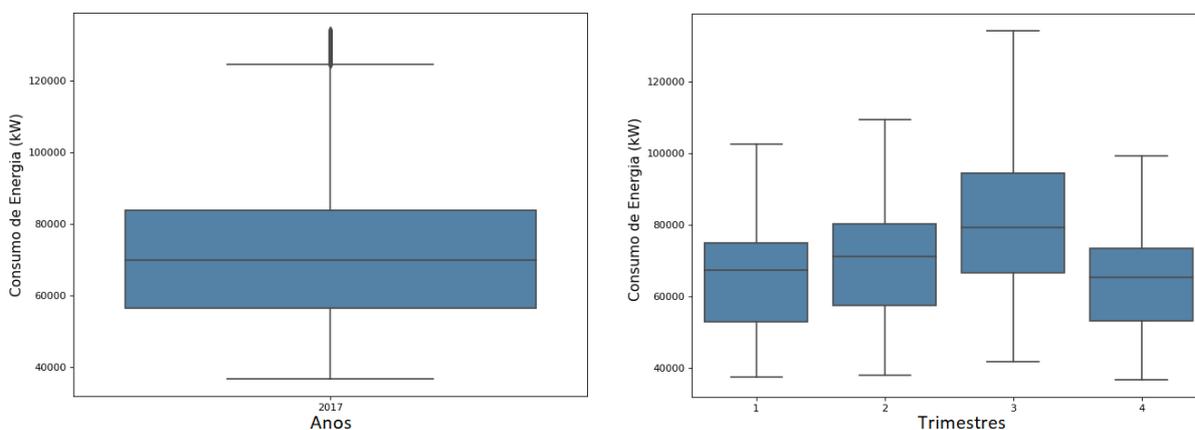


Fonte: Elaboração própria, 2022.

Os dados apresentados na Figura 35 mostram alguns picos no consumo de energia elétrica concentrados entre os meses de julho a setembro, que se dá devido ser o período de verão e de férias, onde ocorre um expressivo aumento do consumo de energia (o número de visitantes também aumenta nessa época do ano).

A Figura 36 apresenta os *boxplots* com o objetivo de verificar a distribuição dos dados relativos à potência ativa.

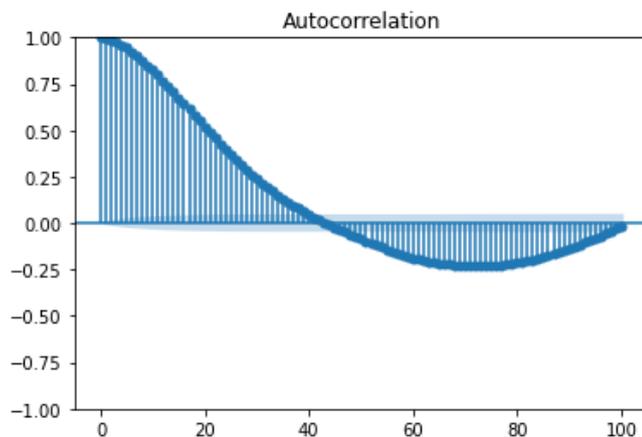
Figura 36: *Boxplots* de consumo de energia elétrica do *dataset* Tetouan-Zones.



Fonte: Elaboração própria, 2022.

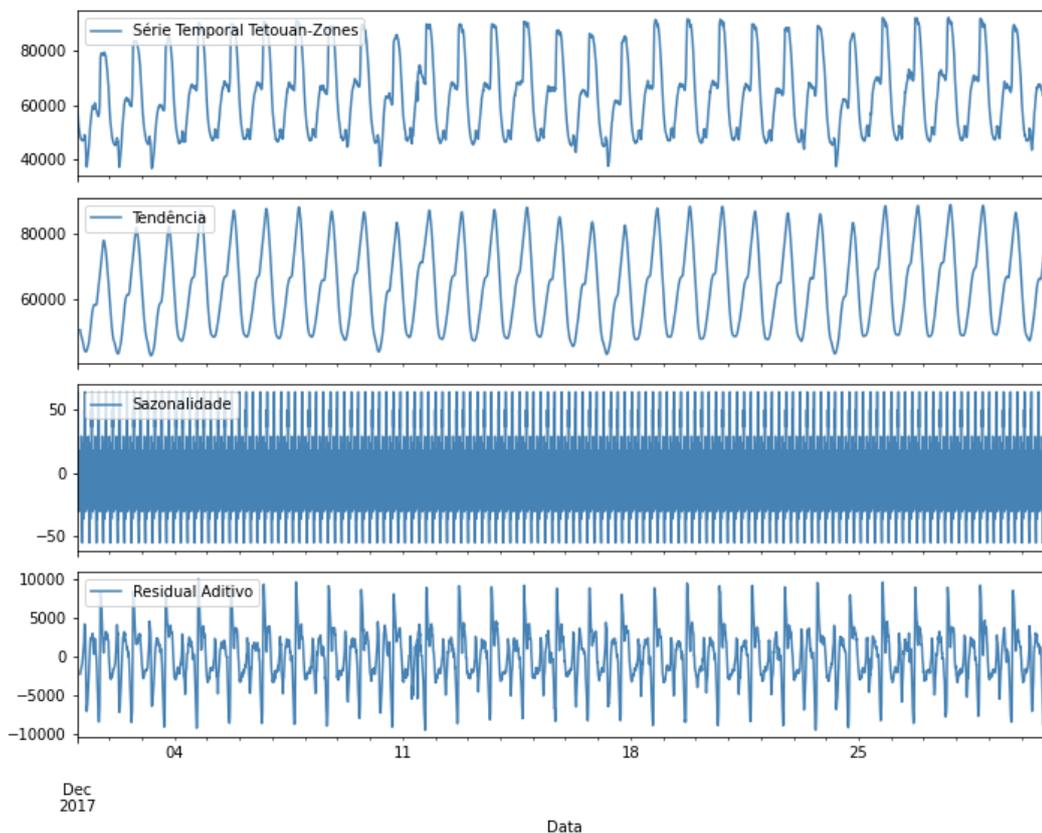
Na Figura 36 foram apresentadas as estatísticas referentes à potência ativa, revelando que o consumo de energia elétrica médio é quase constante ao longo dos anos, enquanto o gráfico trimestral mostra menor consumo no primeiro e quarto, já o segundo e terceiro apresentam uma crescente no consumo.

A Figura 37 apresenta a ACF da potência ativa, corroborando para uma janela deslizante com 90 observações.

Figura 37: Função de autocorrelação no *dataset* Tetouan-Zones.

Fonte: Elaboração própria, 2022.

A Figura 38 exibe os resultados da decomposição clássica automática da ST univariada da potência ativa para o *dataset* de validação referente ao mês de Dezembro de 2017, utilizando o modelo aditivo para tendência, sazonalidade e resíduos.

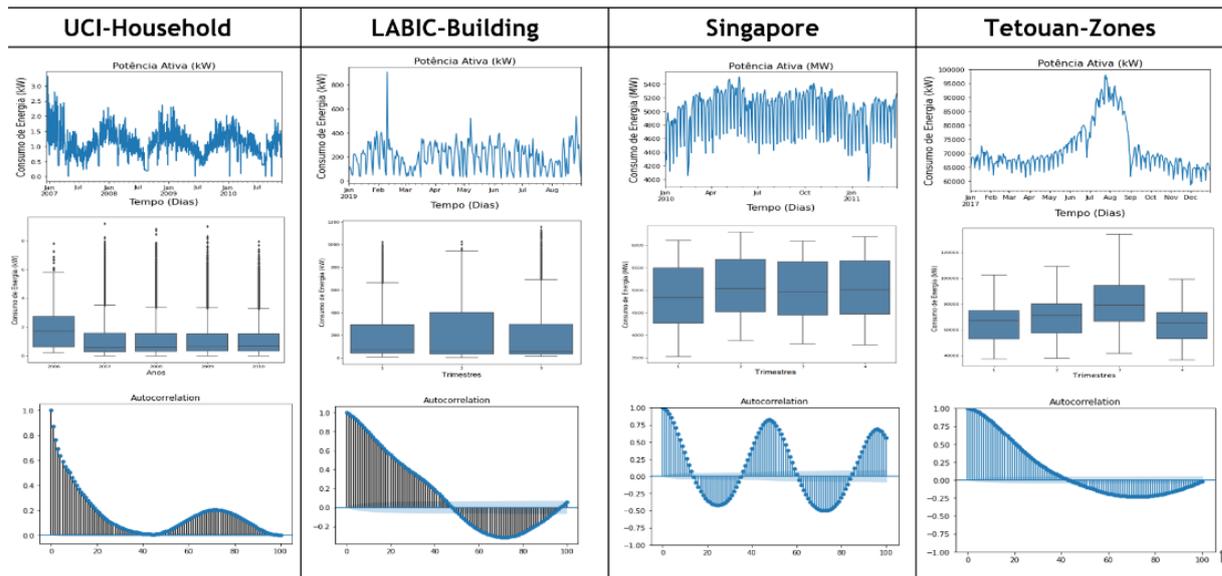
Figura 38: Decomposição clássica da ST para o *dataset* Tetouan-Zones.

Fonte: Elaboração própria, 2022.

5.1.5 Resumo da Análise Exploratória nos quatro *datasets*.

A Figura 39 destaca o resumo da análise das STs da potência ativa para todos os *datasets*, com objetivo de mostrar uma visão comparativa dos resultados.

Figura 39: Resumo da Análise Exploratória nos quatro *datasets*.



Fonte: Elaboração própria, 2022.

Na Figura 39 foram exibidas as STs por dia e mês da potência ativa de cada *dataset*, em seguida os Boxplots com objetivo de verificar a distribuição dos dados; e, por fim, a ACF da potência ativa, onde a observação em um dado instante está relacionada a observações passadas.

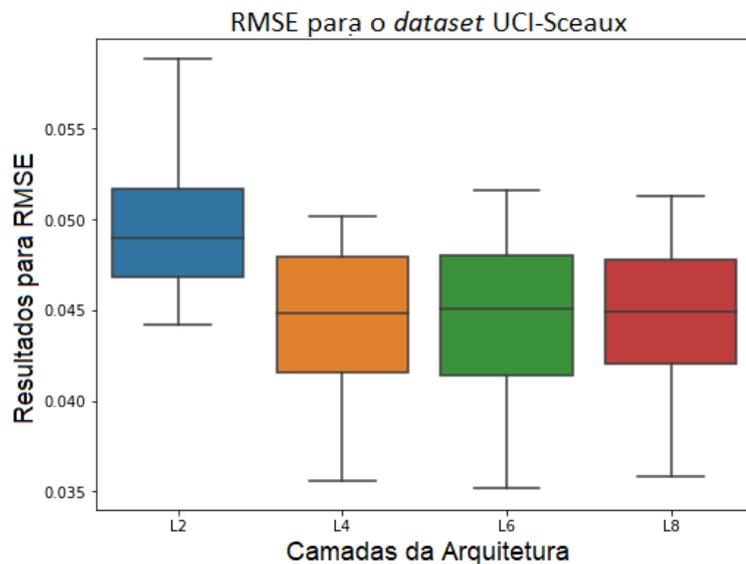
5.2 Resultados preliminares para dois *datasets*

Nas subseções seguintes, serão apresentados os resultados computacionais obtidos nos experimentos preliminares usando a rede neural LSTM (número de camadas) para os *datasets* UCI-Household e LABIC-Building (Subseção 5.2.1). Na Subseção 5.2.2 é apresentado o resultado da predição de consumo de energia elétrica com a rede LSTM para o *dataset* LABIC-Building, com dados de validação (conforme descrito na Subseção 4.6).

5.2.1 Análise do número de camadas da rede LSTM

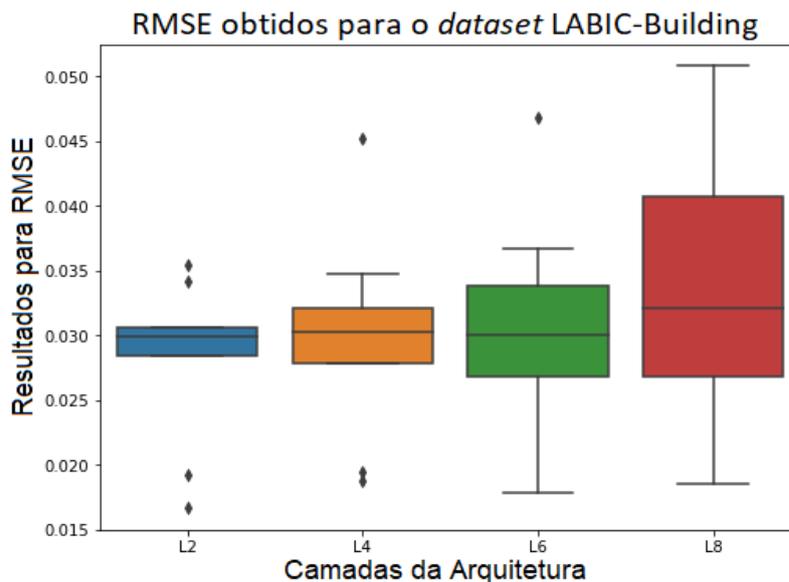
As Figuras 40 e 41 mostram os *boxplots* dos resultados obtidos da métrica RMSE em cada número de camadas LSTM (L2, L4, L6, L8) para os valores normalizados, nos *datasets* UCI-Household e LABIC-Building.

Figura 40: *Boxplots* dos resultados de RMSE obtidos para o *dataset* UCI-Household.



Fonte: Elaboração própria, 2022.

Figura 41: *Boxplots* dos resultados de RMSE obtidos para o *dataset* LABIC-Building.



Fonte: Elaboração própria, 2022.

Na Figura 40, para o *dataset* UCI-Household, aparentemente o modelo L2 teve o pior desempenho, porém a hipótese nula de desempenho semelhante não pôde ser rejeitada no teste de Kruskal-Wallis ($p = 0,0833$). Na Figura 41, para o *dataset* LABIC-Building, a arquitetura L8 apresenta amplitudes maiores. Neste caso, não houve diferença estatisticamente significativa entre o número de camadas dos modelos ($p = 0,7801$).

A Tabela 8 apresenta a média das métricas MAE e RMSE nos testes preliminares obtidos pelos 10 modelos, para cada *dataset*.

Tabela 8: Métricas médias obtidas pelos modelos de rede em cada *dataset*.

Dataset	Model	MAE	RMSE
UCI-Household	L2	0,0332	0,0498
	L4	0,0290	0,0442
	L6	0,0291	0,0444
	L8	0,0285	0,0444
LABIC-Building	L2	0,0202	0,0283
	L4	0,0217	0,0299
	L6	0,0220	0,0301
	L8	0,0258	0,0339

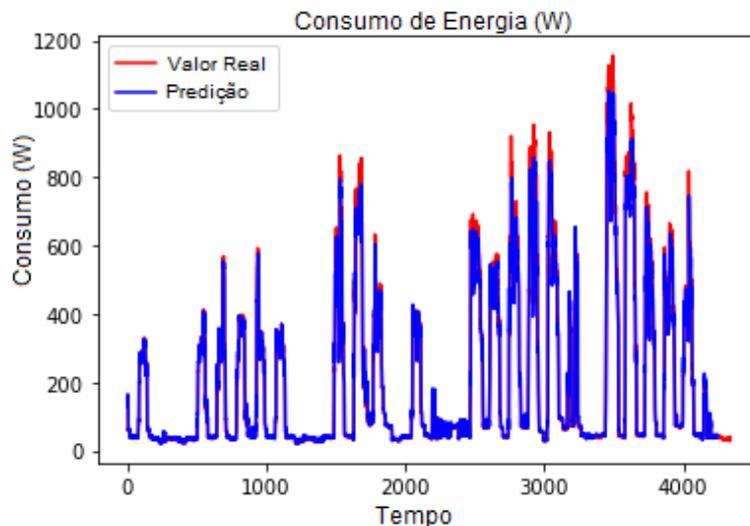
Fonte: Elaboração própria, 2022.

Na Tabela 8, para o *dataset* UCI-Household, o modelo de 4 camadas obteve menor RMSE médio (0,0442). Já no *dataset* LABIC-Building, o menor RMSE médio foi de 0,0283 correspondente ao modelo de 2 camadas.

5.2.2 Predição da rede LSTM em dados não usados para treinamento

Utilizando o *dataset* LABIC-Building, buscou-se prever o consumo de energia elétrica para o mês de agosto de 2019 (as últimas 4.338 observações da ST), utilizando o modelo L2 (com menor RMSE médio obtido na Seção 5.2.1). A Figura 342 exhibe os resultados da previsão com a rede LSTM utilizando as configurações descritas no Capítulo 4 Seção 4.5 e o modelo com 2 camadas (L2).

Figura 42: Resultado da execução da rede LSTM (observações do *dataset* LABIC-Building não usado para treinamento - agosto de 2019).



Fonte: Elaboração própria, 2022.

A Figura 42 mostra que o algoritmo LSTM foi capaz de aprender com os dados de treinamento e prever o mês de agosto de 2019 no *dataset* LABIC-Building. A previsão de agosto de 2019 obteve $RMSE = 0,0289$ e $MAE = 0,0205$. Com esses resultados apresentados é possível destacar que o modelo de rede proposto foi capaz de aprender com os dados de treinamento, e quando aplicado nos testes, as previsões ficaram próximas do valor real, o que demonstrou o sucesso dos testes preliminares.

5.3 Resultados entre LSTM, XGBoost e RF com dados de validação

Na Subseção 4.6 (pré-processamento dos dados) foi apresentado como seria realizada a divisão dos dados de cada *dataset* para treinamento-teste e validação. Assim sendo, os resultados apresentados nesta Seção estão relacionados aos dados de validação dos modelos treinados, bem como o tempo de execução dos algoritmos LSTM, XGBoost e RF, para os *datasets* UCI-Household e LABIC-Building.

5.3.1 Treinamento de Teste com o *dataset* UCI-Household

As Tabelas 9, 10 e 11, respectivamente, mostram as pontuações dos testes obtidos pelos algoritmos LSTM, XGBoost e RF, contendo cada uma das métricas RMSE, MAE, MAPE e R^2 aplicados ao *dataset* UCI-Household para VC-ST

executados 10 vezes. Na sequência, a Figura 43 exibe os *boxplots* para o RMSE dos modelos LSTM, XGBoost e RF.

Tabela 9: Resultados da rede LSTM para o *dataset* UCI-Household.

Fold (VC-ST)	Dataset UCI-Household com LSTM			
	RMSE	MAE	MAPE (%)	R ²
1	0,472	0,262	28,0	75,5
2	0,463	0,262	25,7	74,4
3	0,471	0,269	28,0	75,6
4	0,458	0,245	28,6	75,0
5	0,464	0,250	24,0	76,3
6	0,457	0,242	29,5	78,2
7	0,445	0,242	29,8	77,3
8	0,463	0,251	23,0	76,5
9	0,457	0,254	26,2	77,1
10	0,461	0,255	29,5	74,0
Média	0,461	0,253	27,2	76,0
Mediana	0,462	0,252	28,0	75,9
Desvio Padrão	0,008	0,009	2,4	1,3
Mínimo	0,445	0,242	23,0	74,0
Máximo	0,472	0,269	29,8	78,2

Fonte: Elaboração própria, 2022.

Tabela 10: Resultados do XGBoost para o *dataset* UCI-Household.

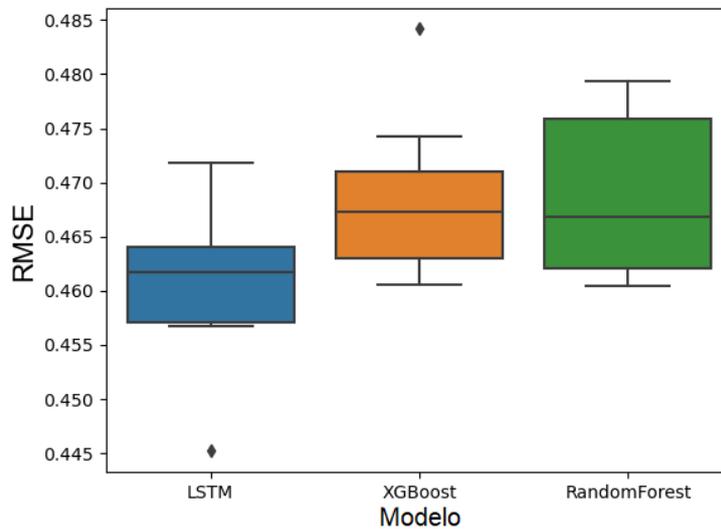
Fold (VC-ST)	Dataset UCI-Household com XGBoost*			
	RMSE	MAE	MAPE (%)	R ²
1	0,484	0,279	29,2	74,2
2	0,474	0,270	27,7	75,3
3	0,472	0,269	27,8	75,6
4	0,469	0,265	27,4	75,8
5	0,468	0,264	27,1	76,0
6	0,467	0,262	27,0	76,1
7	0,464	0,260	26,7	76,4
8	0,463	0,260	26,7	76,5
9	0,461	0,257	26,4	76,7
10	0,462	0,257	26,4	76,6
Média	0,468	0,264	27,2	75,9
Mediana	0,467	0,263	27,0	76,0
Desvio Padrão	0,007	0,007	0,8	0,7
Mínimo	0,461	0,257	26,4	74,2
Máximo	0,484	0,279	29,2	76,7

Fonte: Elaboração própria, 2022.

Tabela 11: Resultados do RF para o *dataset* UCI-Household.

Fold (VC-ST)	Dataset UCI-Household com RF			
	RMSE	MAE	MAPE (%)	R ²
1	0,479	0,264	26,0	74,8
2	0,477	0,262	25,8	75,0
3	0,477	0,262	25,6	75,0
4	0,473	0,259	25,6	75,4
5	0,469	0,257	25,1	75,8
6	0,464	0,255	25,3	76,3
7	0,462	0,256	25,7	76,6
8	0,461	0,258	25,9	76,6
9	0,463	0,256	25,5	76,4
10	0,461	0,256	25,6	76,7
Média	0,469	0,259	25,6	75,9
Mediana	0,467	0,258	25,6	76,1
Desvio Padrão	0,007	0,003	0,3	0,8
Mínimo	0,461	0,255	25,1	74,8
Máximo	0,479	0,264	26,0	76,7

Fonte: Elaboração própria, 2022.

Figura 43: *Boxplots* do RMSE obtido com os modelos LSTM, XGBoost e RF para o UCI-Household.

Fonte: Elaboração própria, 2022.

Apesar do menor RMSE médio e mediano da LSTM (respectivamente 0,461 e 0,462), de acordo com o teste de Kruskal-Wallis não há diferença estatisticamente significativa entre o RMSE dos algoritmos LSTM, XGBoost e RF para o *dataset* UCI-Household ($p = 0,0718$), mostrando uma tendência de resultados da rede LSTM mais baixos, mas sem a confirmação dos testes estatísticos.

5.3.2 Treinamento de Teste com o *dataset* LABIC-Building

As Tabelas 12, 13 e 14, respectivamente, mostram as pontuações dos testes obtidos pelos algoritmos LSTM, XGBoost e RF, contendo cada uma das métricas RMSE, MAE, MAPE e R^2 aplicados ao *dataset* LABIC-Building para VC-ST executados 10 vezes. Na sequência, a Figura 44 exibe os boxplots para o RMSE dos modelos LSTM, XGBoost e RF.

Tabela 12: Resultados da LSTM para o *dataset* LABIC-Building.

Fold (VC-ST)	Dataset LABIC-Building com LSTM			
	RMSE	MAE	MAPE (%)	R^2
1	35,517	17,122	10,5	97,9
2	36,167	21,485	22,6	97,8
3	37,761	18,137	10,1	97,6
4	35,388	17,387	12,6	97,9
5	35,348	17,956	12,8	97,9
6	37,509	21,297	18,4	97,6
7	35,843	17,498	11,4	97,8
8	35,883	18,915	14,8	97,8
9	36,544	19,617	14,9	97,8
10	35,058	17,778	12,7	97,9
Média	36,102	18,719	14,1	97,8
Mediana	35,863	18,047	12,7	97,8
Desvio Padrão	0,916	1,590	3,8	0,1
Mínimo	35,058	17,122	10,1	97,6
Máximo	37,761	21,485	22,6	97,9

Fonte: Elaboração própria, 2022.

Tabela 13: Resultados do XGBoost para o *dataset* LABIC-Building.

Fold (VC-ST)	Dataset LABIC-Building com XGBoost*			
	RMSE	MAE	MAPE (%)	R ²
1	61,420	27,576	13,7	93,7
2	45,504	21,206	13,3	96,5
3	39,425	19,724	12,9	97,4
4	39,666	19,142	11,6	97,4
5	38,766	18,571	11,1	97,5
6	38,468	18,580	11,1	97,5
7	38,034	18,334	10,9	97,6
8	37,979	18,025	10,5	97,6
9	38,823	18,442	10,3	97,5
10	38,931	18,401	10,4	97,5
Média	41,702	19,800	11,6	97,0
Mediana	38,877	18,576	11,1	97,5
Desvio Padrão	7,264	2,886	1,3	1,2
Minimo	37,979	18,025	10,3	93,7
Máximo	61,420	27,576	13,7	97,6

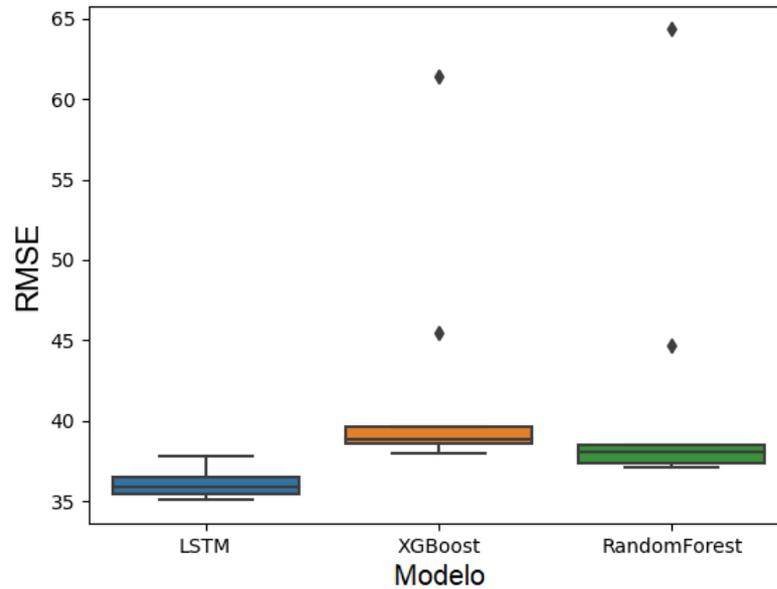
Fonte: Elaboração própria, 2022.

Tabela 14: Resultados do RF para o *dataset* LABIC-Building.

Fold (VC-ST)	Dataset LABIC-Building com RF			
	RMSE	MAE	MAPE (%)	R ²
1	64,347	29,863	16,0	93,0
2	44,696	20,867	11,4	96,6
3	37,944	18,799	10,8	97,6
4	37,379	18,193	10,3	97,7
5	37,061	18,142	10,6	97,7
6	37,437	18,561	11,4	97,6
7	37,215	18,200	11,0	97,7
8	38,132	18,358	10,8	97,6
9	38,516	18,364	10,6	97,5
10	38,427	18,366	10,6	97,5
Média	41,115	19,771	11,3	97,1
Mediana	38,038	18,365	10,8	97,6
Desvio Padrão	8,463	3,637	1,7	1,4
Minimo	37,061	18,142	10,3	93,0
Máximo	64,347	29,863	16,0	97,7

Fonte: Elaboração própria, 2022.

Figura 44: *Boxplots* do RMSE obtido com os modelos LSTM, XGBoost e RF para o LABIC-Building.



Fonte: Elaboração própria, 2022.

De acordo com os resultados do teste de Kruskal-Wallis para o *dataset* LABIC-Building, há diferença estatisticamente significativa entre as métricas de RMSE para os algoritmos LSTM, XGBoost e RF ($p < 0,0001$).

O teste de Dunn indicou que há diferença estatisticamente significativa entre a LSTM em relação aos algoritmos RF e XGBoost, conforme mostrado na Tabela 15. Deste modo, pode-se concluir que a LSTM superou estatisticamente os demais algoritmos para o caso testado (*dataset* LABIC-Building).

Tabela 15: Teste *post-hoc* de Dunn para o *dataset* LABIC-Building.

Número de comparação	Comparação por grupo	Diferença nas classificações médias	Corte em $\alpha=0,05$	Diferença de significância = **
1	LSTM-RF	10,9	9,42511	**
2	LSTM-XGB	16,7	9,42511	**
3	RF-XGB	5,8	9,42511	

A Tabela 16 mostra as pontuações médias dos testes consolidados para os modelos LSTM, XGBoost e RF, aplicados aos *datasets* UCI-Household e LABIC-Building, com base nos dados descritos nas Tabelas 9 a 14.

Tabela 16: Tabela consolidada com pontuações médias de teste dos algoritmos LSTM, XGBoost e RF para os *datasets* UCI-Household e LABIC-Building.

Models	Dataset UCI-Household (Média)				Dataset LABIC-Building (Média)			
	RMSE (W)	MAE (W)	MAPE (%)	R ²	RMSE (W)	MAE (W)	MAPE (%)	R ²
XGBoost*	0,468	0,264	27,2	75,9	41,702	19,800	11,6	97,0
RF	0,469	0,259	25,6	75,9	41,115	19,771	11,3	97,1
LSTM	0,461	0,253	27,2	76,0	36,102	18,719	14,1	97,8

Fonte: Elaboração própria, 2022.

Com base nos resultados obtidos pelos algoritmos com VC-ST bem como os testes estatísticos, evidenciou-se a superioridade apresentada pela rede Neural LSTM em relação aos algoritmos XGBoost e RF, sendo que a mesma foi escolhida para ser analisada a partir de novos experimentos com dois novos *datasets*.

A motivação para esses novos testes foi observar se o comportamento dos resultados obtidos pela LSTM nos *datasets* UCI-Household e LABIC-Building se mantém em novos *datasets*. Além disso, optou-se também por utilizar a rede neural BiLSTM, tendo em vista se tratar de uma variação da LSTM unidirecional utilizada com sucesso nos primeiros testes desta pesquisa.

Assim sendo, a próxima Subseção (5.4) descreve os resultados obtidos nesses testes onde as redes neurais LSTM e BiLSTM foram utilizadas com VC-ST para os *datasets* Singapore e Tetouan-Zones, que já foram apresentados nas subseções 5.1.3 (Singapore) e 5.3.4 (Tetouan-Zones).

Cabe ressaltar que para os novos testes (descritos na Subseção 5.4), foi adicionada a métrica NRMSE (descrita na Subseção 4.7) para todos os resultados obtidos pelos modelos, sendo usado também para exibição nos *boxplots* e na Tabela consolidada com resumo dos resultados.

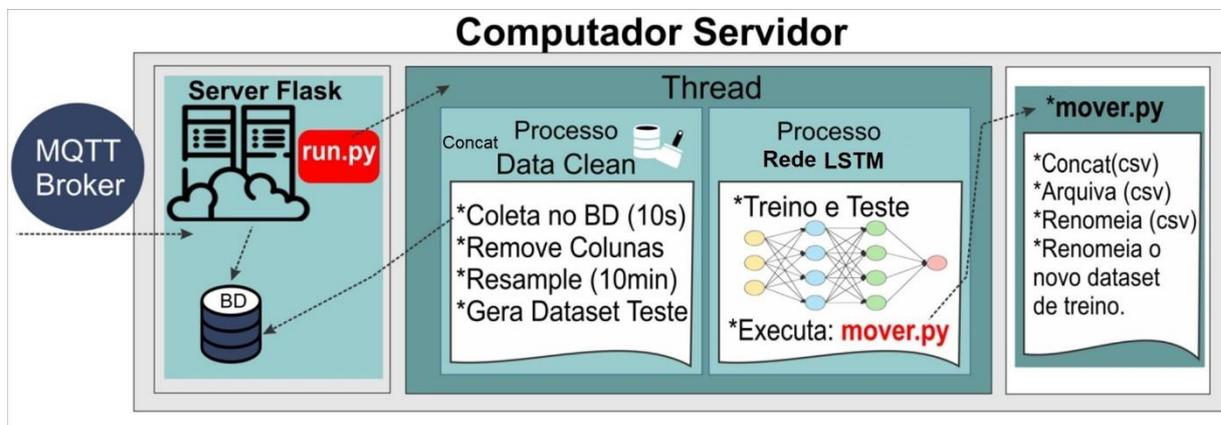
Tendo em vista a RNP LSTM apresentar resultados estatisticamente melhores que os algoritmos XGBoost e RF, a mesma foi utilizada para ser implantada como

módulo de predição de consumo de energia elétrica no sistema IoT EnergySaver, conforme descrito a seguir.

5.4 Implantação do módulo de predição com LSTM para o EnergySaver

Conforme descrito no Capítulo 3 (Subseção 3.10.1), o módulo com LSTM realiza tanto a limpeza de dados quanto a sua execução para previsão, automaticamente no primeiro dia de cada mês, conforme Figura 45.

Figura 45: Funcionamento do módulo com rede LSTM.



Fonte: Elaboração própria, 2022.

Na Figura 45, os dados recebidos e armazenados no banco de dados devem passar por um processo de limpeza antes de serem utilizados pela rede para treinamento e teste. Para isso, foi criada uma Thread¹³ que executa dois scripts, sendo um para realizar a limpeza dos dados e outro para executar a rede LSTM. Assim sendo, quando a Thread é executada, ocorre a limpeza dos dados de forma simultânea em que a LSTM é executada para realizar a previsão de consumo, bem como um processo de concatenação entre o arquivo de treinamento e teste, criando um novo arquivo de treinamento com os dados antigos e os dados utilizados para testar a rede do mês atual. No início do mês seguinte, todo o processo descrito nas etapas anteriores é executado novamente. De forma mais detalhada, o módulo LSTM executa automaticamente as etapas descritas a seguir.

- I. Inicialmente, é feita uma consulta para retornar os dados recebidos pelo *MongoDB* do primeiro ao último dia do mês, a fim de criar o *dataset* de teste (já que o *dataset* de treinamento da rede foi criado anteriormente).

¹³ Executa duas ou mais tarefas/processos computacionais de forma simultânea.

- II. Em seguida, remove os atributos desnecessários para a rede ('*_id*', '*user*', '*local*', '*device*', '*name_sensor*', '*type_sensor*', '*model_sensor*') e usa apenas os dados de data, hora e potência ativa.
- III. Como os dados são recebidos no *MongoDB* a cada 10 segundos (padrão do sistema) a rede é padronizada para usar índice base fixo (intervalos de 10 minutos). Assim, o módulo cria um novo *dataset* com data e hora a partir da potência ativa média contida em cada intervalo. Este novo *dataset* criado será usado para os testes com a rede LSTM.
- IV. Após isso, a rede LSTM é executada para realizar a previsão de consumo. Após a conclusão desse processo, uma mensagem é exibida ao usuário na página web com o resultado da previsão obtido pela LSTM.
- V. Por fim, é realizado um processo de concatenação entre o arquivo de treinamento e teste (dos meses anteriores e do mês atual que foi realizada a previsão) usados nos passos anteriores, criando um novo arquivo de treinamento para ser usado pela rede no início do mês seguinte, onde todo o processo descrito nas etapas anteriores é novamente executado de forma automática pelo módulo LSTM proposto.

É importante destacar que para possíveis erros de comunicação entre os dispositivos que enviam e recebem dados, o sistema EnergySaver tem o tratamento adequado onde envia mensagens na tela para o administrador/usuário, bem como para o e-mail do administrador, de forma automática, informando o problema ocorrido. Esses tratamentos de erro também são importantes para que a rede LSTM execute o processo de treinamento e teste com eficiência (DA SILVA *et al.*, (2021a).

Após a implantação do módulo LSTM para o sistema IoT EnergySaver, optou-se por realizar novos testes utilizando para isso a rede neural BiLSTM, tendo em vista se tratar de uma variação da LSTM unidirecional. Para isso, foram utilizados dois novos *datasets*, além dos dois que já haviam sido utilizados nos testes anteriores. Na Seção seguinte esses novos testes serão descritos detalhadamente.

5.5 Comparação estatística entre as redes neurais LSTM e BiLSTM com dados de validação

Nas subseções seguintes, serão apresentados os resultados obtidos pelas redes neurais LSTM e BiLSTM para os *datasets* UCI-Household, LABIC-Building, Singapore e Tetouan-Zones, com dados de validação, individualmente. Será também descrita a análise estatística para verificar se há diferença estatisticamente significativa entre as redes. Por fim, será apresentado o tempo de execução dos algoritmos para treinamento-teste em cada *dataset*.

5.5.1 Dados de validação com o *dataset* UCI-Household

As Tabelas 17 e 18, respectivamente, mostram os valores das métricas RMSE, NRMSE, MAE, MAPE e R^2 score, para os algoritmos LSTM e BiLSTM, aplicados ao *dataset* UCI-Household para VC-ST executados 10 vezes. Na sequência, a Figura 46 exibe os *boxplots* referentes a métrica NRMSE obtida para os modelos.

Tabela 17: Resultados da rede LSTM para o *dataset* UCI-Household.

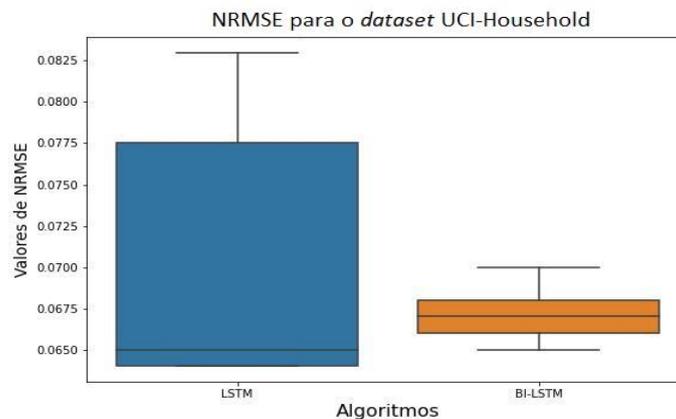
Fold (VC-ST)	Dataset UCI-Household com LSTM				
	RMSE	NRMSE	MAE	MAPE (%)	R^2
1	0,484	0,065	0,272	27,3	74,4
2	0,481	0,064	0,266	25,6	74,7
3	0,477	0,064	0,275	28,9	75,1
4	0,594	0,079	0,394	51,2	61,4
5	0,489	0,065	0,270	25,4	73,9
6	0,620	0,083	0,414	50,9	58,0
7	0,548	0,073	0,349	41,9	67,2
8	0,476	0,064	0,270	27,1	75,2
9	0,480	0,064	0,268	27,4	74,8
10	0,605	0,081	0,407	50,6	59,9
Média	0,525	0,070	0,318	35,6	69,5
Mediana	0,486	0,065	0,273	28,1	74,1
Desvio Padrão	0,060	0,008	0,065	11,6	7,1
Mínimo	0,476	0,064	0,266	25,4	58,0
Máximo	0,620	0,083	0,414	51,2	75,2

Fonte: Elaboração própria, 2022.

Tabela 18: Resultados da rede BiLSTM para o *dataset* UCI-Household.

Fold (VC-ST)	Dataset UCI-Household com Bi_LSTM				
	RMSE	NRMSE	MAE	MAPE (%)	R ²
1	0,497	0,066	0,291	30,5	72,9
2	0,510	0,068	0,286	27,3	71,5
3	0,506	0,068	0,290	27,6	72,0
4	0,506	0,068	0,290	27,6	72,0
5	0,499	0,067	0,307	36,2	72,7
6	0,488	0,065	0,286	31,2	74,0
7	0,491	0,066	0,266	24,0	73,6
8	0,499	0,067	0,287	24,1	72,8
9	0,485	0,065	0,265	23,6	74,3
10	0,522	0,070	0,391	59,7	70,2
Média	0,500	0,067	0,296	31,2	72,6
Mediana	0,499	0,067	0,288	27,6	72,8
Desvio Padrão	0,011	0,001	0,036	10,7	1,2
Mínimo	0,485	0,065	0,265	23,6	70,2
Máximo	0,522	0,070	0,391	59,7	74,3

Fonte: Elaboração própria, 2022.

Figura 46: *Boxplots* do NRMSE da LSTM e BiLSTM para o *dataset* UCI-Household.

Fonte: Elaboração própria, 2022.

5.5.2 Dados de validação com o *dataset* LABIC-Building

As Tabelas 19 e 20, respectivamente, mostram os valores das métricas RMSE, NRMSE, MAE, MAPE e R² score, para os algoritmos LSTM e BiLSTM, aplicados ao *dataset* LABIC-Building para VC-ST executados 10 vezes. Na sequência, a Figura 47 exibe os *boxplots* referentes a métrica NRMSE obtida para os modelos.

Tabela 19: Resultados da LSTM para o *dataset* LABIC-Building.

Fold (VC-ST)	Dataset LABIC-Building com LSTM				
	RMSE	NRMSE	MAE	MAPE (%)	R ²
1	35,883	0,032	18,079	12,8	97,8
2	36,193	0,032	22,095	24,4	97,8
3	37,972	0,033	18,648	11,3	97,6
4	35,352	0,031	17,933	14,3	97,9
5	35,065	0,031	17,912	13,1	97,9
6	37,042	0,033	22,176	22,1	97,7
7	35,660	0,031	18,113	13,6	97,9
8	36,060	0,032	19,389	15,9	97,8
9	36,158	0,032	20,287	17,3	97,8
10	34,809	0,031	17,981	13,5	98,0
Média	36,020	0,032	19,261	15,8	97,8
Mediana	35,971	0,032	18,380	13,9	97,8
Desvio Padrão	0,936	0,001	1,697	4,3	0,1
Minimo	34,809	0,031	17,912	11,3	97,6
Máximo	37,972	0,033	22,176	24,4	98,0

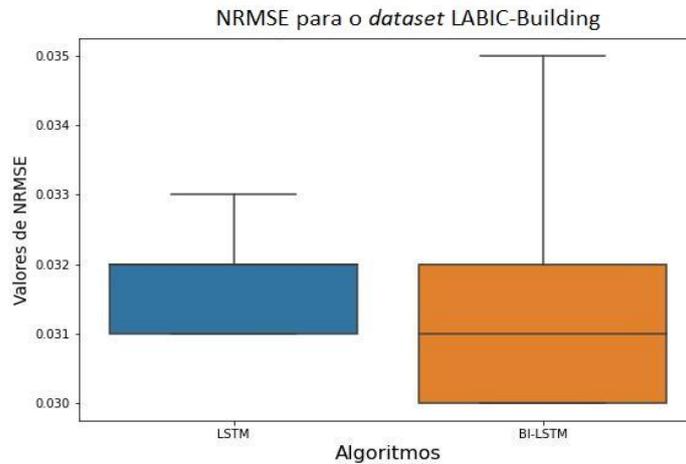
Fonte: Elaboração própria, 2022.

Tabela 20: Resultados da BiLSTM para o *dataset* LABIC-Building.

Fold (VC-ST)	Dataset LABIC-Building com BiLSTM				
	RMSE	NRMSE	MAE	MAPE (%)	R ²
1	36,587	0,032	22,003	22,6	97,8
2	39,469	0,035	27,645	36,5	97,4
3	36,895	0,032	18,588	12,4	97,7
4	37,574	0,033	25,741	32,1	97,6
5	34,093	0,030	17,225	11,9	98,0
6	34,116	0,030	19,217	18,5	98,0
7	33,521	0,030	16,427	11,0	98,1
8	34,342	0,030	17,134	11,2	98,0
9	34,413	0,030	20,092	21,1	98,0
10	35,989	0,032	21,705	22,0	97,8
Média	35,700	0,031	20,578	19,9	97,9
Mediana	35,201	0,031	19,654	19,8	97,9
Desvio Padrão	1,925	0,002	3,746	8,9	0,2
Minimo	33,521	0,030	16,427	11,0	97,4
Máximo	39,469	0,035	27,645	36,5	98,1

Fonte: Elaboração própria, 2022.

Figura 47: *Boxplots* do NRMSE obtido com LSTM e BiLSTM para o *dataset* LABIC-Building.



Fonte: Elaboração própria, 2022.

5.5.3 Dados de validação com o *dataset* Singapore

As Tabelas 21 e 22, respectivamente, mostram os valores das métricas RMSE, NRMSE, MAE, MAPE e R^2 score, para os algoritmos LSTM e BiLSTM, aplicados ao *dataset* Singapore para VC-ST executados 10 vezes. Na sequência, a Figura 48 exibe os *boxplots* referentes a métrica NRMSE obtida para os modelos.

Tabela 21: Resultados da LSTM para o *dataset* Singapore.

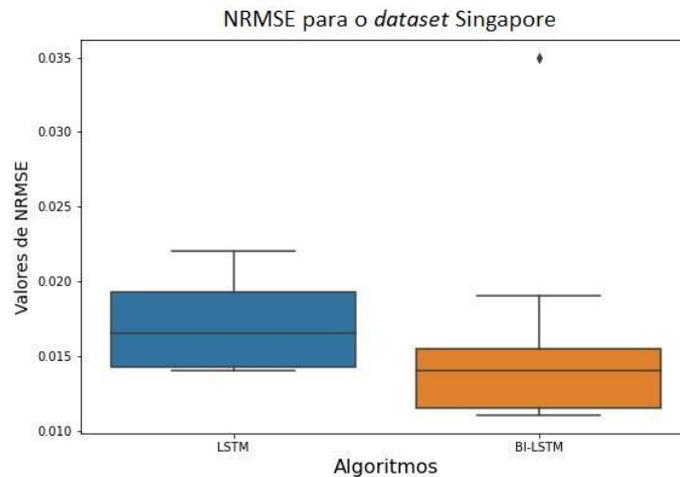
Fold (VC-ST)	Dataset Singapore com LSTM				
	RMSE	NRMSE	MAE	MAPE (%)	R^2
1	35,096	0,016	28,374	0,6	99,7
2	35,797	0,017	28,957	0,6	99,7
3	30,836	0,014	24,218	0,5	99,8
4	42,846	0,020	35,388	0,7	99,6
5	32,755	0,015	26,533	0,5	99,7
6	48,184	0,022	39,409	0,8	99,5
7	31,026	0,014	24,818	0,5	99,8
8	46,894	0,022	40,107	0,8	99,5
9	36,283	0,017	28,940	0,6	99,7
10	29,291	0,014	23,562	0,5	99,8
Média	36,901	0,017	30,031	0,6	99,7
Mediana	35,446	0,017	28,657	0,6	99,7
Desvio Padrão	6,781	0,003	6,131	0,1	0,1
Minimo	29,291	0,014	23,562	0,5	99,5
Máximo	48,184	0,022	40,107	0,8	99,8

Fonte: Elaboração própria, 2022.

Tabela 22: Resultados da BiLSTM para o *dataset* Singapore.

Fold (VC-ST)	Dataset Singapore com BiLSTM				
	RMSE	NRMSE	MAE	MAPE (%)	R ²
1	39,755	0,019	30,713	0,6	99,6
2	74,322	0,035	63,770	1,2	98,7
3	27,914	0,013	22,190	0,5	99,8
4	29,148	0,014	22,167	0,4	99,8
5	33,638	0,016	26,725	0,5	99,7
6	24,266	0,011	18,874	0,4	99,9
7	29,963	0,014	25,081	0,5	99,8
8	22,716	0,011	17,727	0,4	99,9
9	24,229	0,011	18,644	0,4	99,9
10	29,269	0,014	23,618	0,5	99,8
Média	33,522	0,016	26,951	0,5	99,7
Mediana	29,209	0,014	22,904	0,5	99,8
Desvio Padrão	15,179	0,007	13,537	0,3	0,4
Mínimo	22,716	0,011	17,727	0,4	98,7
Máximo	74,322	0,035	63,770	1,2	99,9

Fonte: Elaboração própria, 2022.

Figura 48: *Boxplots* do NRMSE obtido com os modelos LSTM e BiLSTM para o *dataset* Singapore.

Fonte: Elaboração própria, 2022.

5.5.4 Dados de validação com o *dataset* Tetouan-Zones

As Tabelas 23 e 24, respectivamente, mostram os valores das métricas RMSE, NRMSE, MAE, MAPE e R² score, para os algoritmos LSTM e BiLSTM, aplicados ao *dataset* Tetouan-Zones para VC-ST executados 10 vezes. Na sequência, a Figura 49 exhibe os *boxplots* referentes a métrica NRMSE obtida para os modelos.

Tabela 23: Resultados da LSTM para o *dataset* Tetouan-Zones.

Fold (VC-ST)	Dataset Tetouan-Zones com LSTM				
	RMSE	NRMSE	MAE	MAPE (%)	R ²
1	880,601	0,016	707,742	1,2	99,6
2	806,638	0,015	591,798	0,9	99,7
3	747,032	0,014	569,353	1,0	99,7
4	1264,597	0,023	1013,582	1,5	99,2
5	832,836	0,015	623,614	0,9	99,7
6	785,838	0,014	612,084	1,0	99,7
7	952,206	0,017	807,024	1,4	99,6
8	833,170	0,015	648,251	1,0	99,7
9	810,916	0,015	637,257	1,1	99,7
10	654,841	0,012	472,449	0,8	99,8
Média	856,867	0,016	668,315	1,1	99,6
Mediana	821,876	0,015	630,436	1,0	99,7
Desvio Padrão	163,161	0,003	149,273	0,2	0,2
Mínimo	654,841	0,012	472,449	0,8	99,2
Máximo	1.264,597	0,023	1.013,582	1,5	99,8

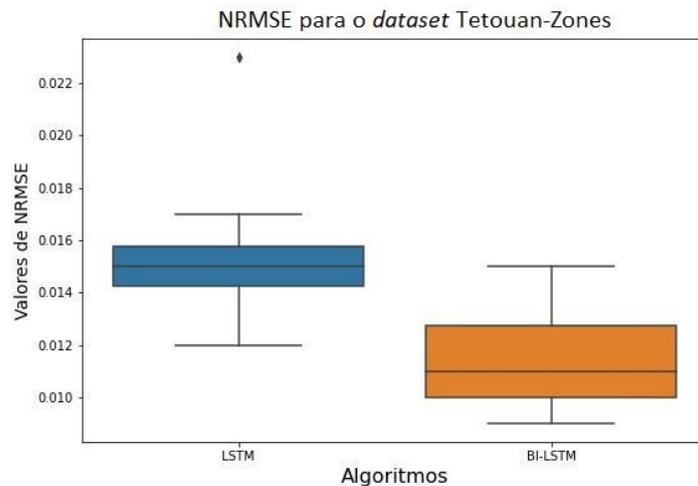
Fonte: Elaboração própria, 2022.

Tabela 24: Resultados da BiLSTM para o *dataset* Tetouan-Zones.

Fold (VC-ST)	Dataset Tetouan-Zones com BiLSTM				
	RMSE	NRMSE	MAE	MAPE (%)	R ²
1	618,613	0,011	457,710	0,8	99,8
2	576,300	0,010	429,827	0,7	99,8
3	834,071	0,015	641,204	1,2	99,7
4	850,158	0,015	653,708	1,0	99,6
5	730,202	0,013	579,820	1,0	99,7
6	648,534	0,012	514,702	0,8	99,8
7	573,357	0,010	407,655	0,6	99,8
8	523,353	0,009	386,385	0,6	99,9
9	581,298	0,011	432,836	0,7	99,8
10	526,451	0,010	380,853	0,6	99,9
Média	646,234	0,012	488,470	0,8	99,8
Mediana	599,956	0,011	445,273	0,7	99,8
Desvio Padrão	119,514	0,002	103,106	0,2	0,1
Mínimo	523,353	0,009	380,853	0,6	99,6
Máximo	850,158	0,015	653,708	1,2	99,9

Fonte: Elaboração própria, 2022.

Figura 49: *Boxplots* do NRMSE obtido com os modelos LSTM e BiLSTM para o *dataset* Tetouan-Zones.



Fonte: Elaboração própria, 2022.

5.5.5 Resumo dos Resultados validação em todos os *datasets*

A Tabela 25 mostra as pontuações médias de NRMSE e R^2 score dos testes consolidados para as redes LSTM e BiLSTM, aplicados aos *datasets* UCI-Household, LABIC-Building, Singapore e Tetouan-Zones. Na Subseção 5.5.6 os valores de NRMSE serão avaliados com os testes estatísticos de Friedman e os testes *post hoc* de Nemenyi.

Tabela 25: Tabela consolidada com pontuações médias das métricas RMSE e R^2 score obtidas no testes para todos algoritmos e *datasets*.

Modelos	UCI-Household		LABIC-Building		Singapore		Tetouan-Zones	
	NRMSE	R^2	NRMSE	R^2	NRMSE	R^2 (%)	NRMSE	R^2
LSTM	0,070	69,5	0,032	97,8	0,017	99,7	0,016	99,6
BiLSTM	0,067	72,6	0,031	97,9	0,016	99,7	0,012	99,8

Fonte: Elaboração própria, 2022.

Observando os resultados da Tabela 25, pode-se concluir que a rede BiLSTM obteve menores médias de NRMSE (destacado em negrito) para todos os *datasets* em comparação com a LSTM.

5.5.6 Análise Estatística dos Resultados obtidos pelas redes LSTM e BiLSTM.

Conforme mencionado na Subseção 4.7, para determinar a melhor arquitetura, foi realizado o teste estatístico não paramétrico de Friedman. Esse teste teve como base os resultados obtidos pela métrica NRMSE nos testes de validação pelas redes LSTM e BiLSTM para os *datasets* UCI-Household, LABIC-Building, Singapore e Tetouan-Zones.

De acordo com os resultados do teste de Friedman para todos os *datasets*, há diferença estatisticamente significativa entre as métricas de NRMSE entre os algoritmos LSTM e BiLSTM com p-valor = 0,0455.

Embora seja aplicado para a comparação de múltiplos algoritmos, o teste *post hoc* de Nemenyi (1963) quando aplicado a comparação entre LSTM e BiLSTM, também indicou que há diferença estatisticamente significativa com p-valor = 0,046, que é menor que o limiar 0,05.

De acordo com estes resultados pode-se concluir que a BiLSTM superou estatisticamente a LSTM em todos os *datasets* a partir da métrica NRMSE. Este resultado corrobora com os resultados apresentados pela abordagem de Graves e Schmidhuber (2005), onde concluíram que as redes bidirecionais superam as unidirecionais.

5.5.7 Tempo de execução dos algoritmos por *dataset*

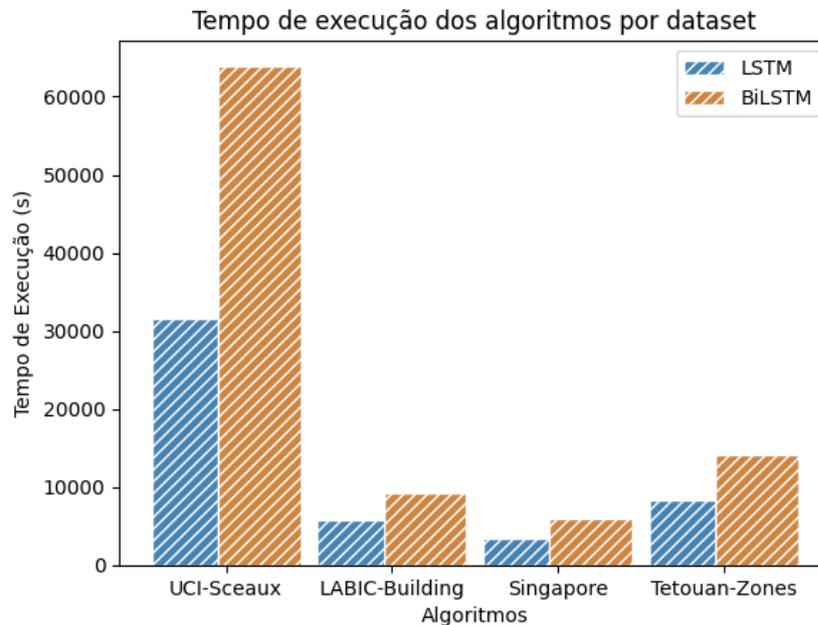
O tempo de execução apresentado nesta Subseção, está relacionado ao tempo gasto pelos algoritmos individualmente por *dataset*, para 10 execuções usando VC-ST para treinamento-teste e validação dos dados, conforme tabela 26 e Figura 50.

Tabela 26: Tempo de execução em horas da LSTM e BiLSTM para todos os *datasets*.

Modelos	UCI-Household	LABIC-Building	Singapore	Tetouan-Zones
	Tempo (h)	Tempo (h)	Tempo (h)	Tempo (h)
LSTM	8,75	1,58	0,97	2,28
BiLSTM	17,76	2,58	1,63	3,93

Fonte: Elaboração própria, 2022.

Figura 50: Tempo de execução em segundos da LSTM e BiLSTM para todos os *datasets*.



Fonte: Elaboração própria, 2022.

Na Tabela 26 e Figura 50, foram apresentados os tempos para treinamento-teste e validação usando VC-ST (10 execuções), da LSTM e BiLSTM para cada *dataset*. Percebe-se, de forma geral, que os maiores tempos de execução são da BiLSTM correspondendo a 63,8% do total gasto e a LSTM com 36,2%.

Em relação ao custo benefício entre o tempo de treinamento e ganho usando a BiLSTM em comparação a LSTM, pode-se destacar que nas Subseções 5.5.5 e 5.5.6, foram apresentados os resultados e testes estatísticos comprovando que a BiLSTM foi melhor estatisticamente, no entanto com o tempo de treinamento duas vezes maior, demandando mais recursos computacionais.

Esta diferença é bastante relevante para a escolha de modelos em cenários reais de aplicação, sendo que a decisão deve partir da avaliação de qual o panorama e a recorrência em que as previsões devem ser realizadas. Caso seja necessário um treinamento rápido do modelo, isso pode impactar nos objetivos da organização, especialmente se for utilizado um *dataset* bastante extenso para o seu treinamento. No entanto, em casos em que o treinamento do modelo possa ser realizado no período de tempo maior, não impactará no caso específico.

Todavia, conforme destacado na Seção 4.6, o modelo de treinamento da LSTM e BiLSTM pode ser salvo em formato “.h5” podendo ser usado para realizar a validação da rede a qualquer momento, demandando pouco tempo, tendo em vista o

modelo já estar salvo. Assim, um novo treinamento completo da rede pode ser realizado em período de tempo maior (bimestral, trimestral, semestral, etc.), sendo que para o seu uso emergencial, por exemplo, pode-se utilizar o modelo salvo anteriormente. Neste caso, o tempo gasto entre a LSTM e BiLSTM para previsão (validação) é similar, tendo em vista não precisar do treinamento adicional, utilizando para isso o modelo salvo.

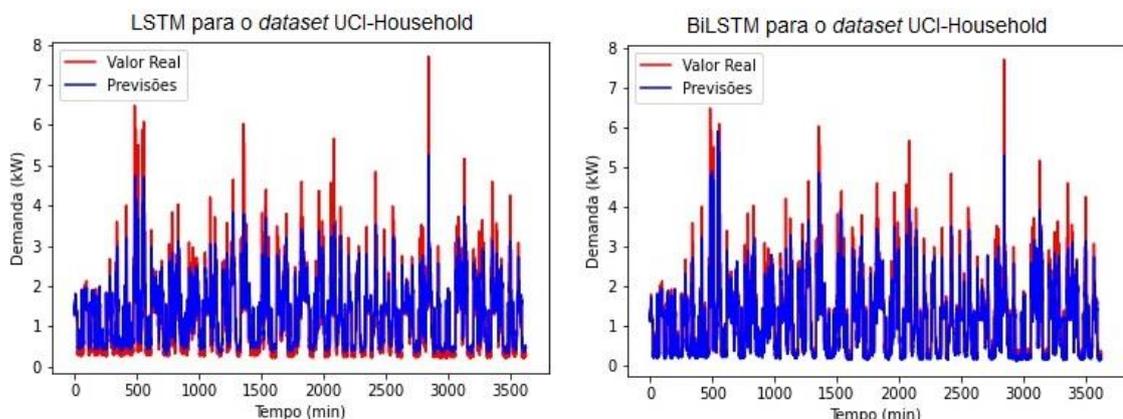
Outra possibilidade é utilizar *Transfer Learning* (OTOVIĆ, 2022; WEISS *et al.*, 2016) a partir de outras redes pré-treinadas para reduzir o tempo de treinamento, sendo necessário avaliar o desempenho. Tal implementação pode ser desenvolvida e avaliada em trabalhos futuros.

5.5.8 Predição com as redes LSTM e BiLSTM em dados não usados para treinamento para todos os *datasets*

Com as redes LSTM e BiLSTM treinadas anteriormente, buscou-se prever o consumo de energia elétrica nos *datasets* UCI-Household, LABIC-Building, Singapore e Tetouan-Zones, utilizando o último mês de coleta de cada um para predição, correspondendo aos dados de validação, conforme descrito na Seção 4.6 (pré-processamento dos dados).

A Figura 51 mostra predições típicas para o *dataset* UCI-Household realizadas pelas redes LSTM (NRMSE = 0,070) e BiLSTM (NRMSE = 0,068).

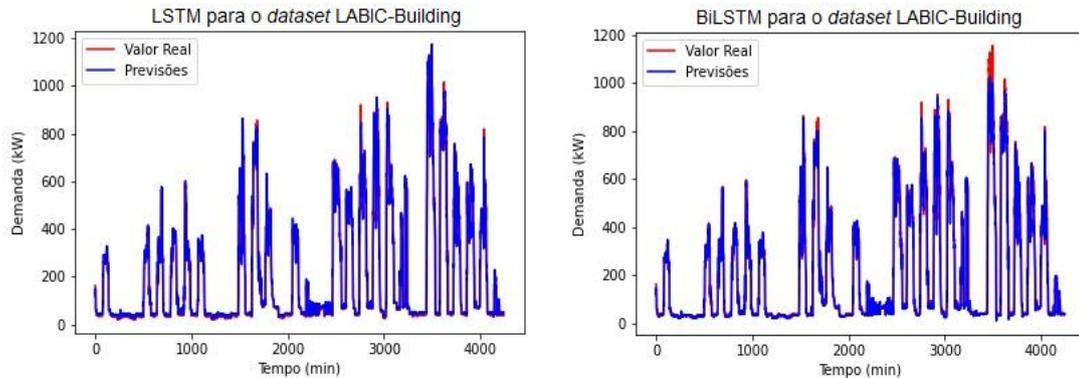
Figura 51: Resultado da predição com LSTM e BiLSTM para o *dataset* UCI-Household.



Fonte: Elaboração própria, 2022.

A Figura 52 mostra previsões típicas para o *dataset* LABIC-Building realizadas pelas redes LSTM (NRMSE = 0,032) e BiLSTM (NRMSE = 0,031).

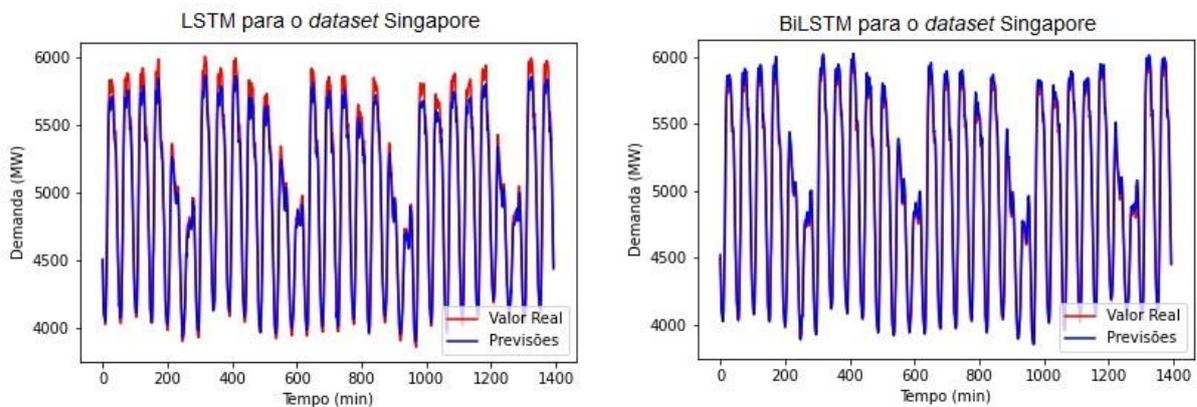
Figura 52: Resultado da previsão com LSTM e BiLSTM para o *dataset* LABIC-Building.



Fonte: Elaboração própria, 2022.

A Figura 53 mostra previsões típicas para o *dataset* Singapore realizadas pelas redes LSTM (NRMSE = 0,017) e BiLSTM (NRMSE = 0,016).

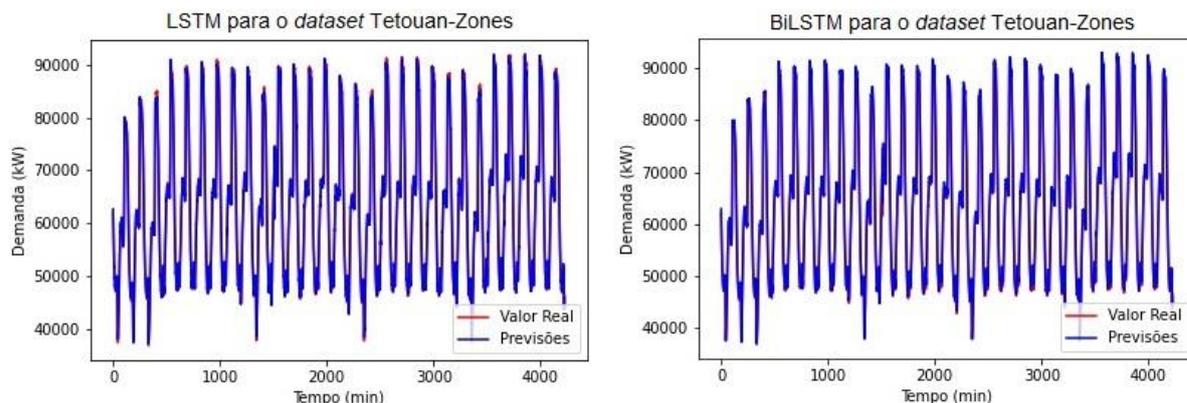
Figura 53: Resultado da previsão com LSTM e BiLSTM para o *dataset* Singapore.



Fonte: Elaboração própria, 2022.

A Figura 54 mostra previsões típicas para o *dataset* Tetouan-Zones realizadas pelas redes LSTM (NRMSE = 0,016) e BiLSTM (NRMSE = 0,012).

Figura 54: Resultado da predição com LSTM e BiLSTM para o *dataset* Tetouan-Zones.



Fonte: Elaboração própria, 2022.

Os resultados apresentados nas Figuras 51 a 54 confirmam que a rede BiLSTM foi estatisticamente superior à LSTM, sendo capaz de aprender com os dados de treinamento e testes, e quando aplicado na validação, as previsões ficaram próximas do valor real.

5.6 Resultados Científicos, Técnicos e Acadêmicos

Os principais resultados científicos, técnicos e acadêmicos, alcançados por este trabalho foram baseados nos resultados esperados descritos no Capítulo 1 Seção 1.6, proporcionados por meio dos experimentos realizados e descritos no decorrer dos Capítulos, divididos em resultados científicos, técnicos, acadêmicos e socioeconômicos, de acordo com o Quadro 2.

Quadro 2: Resultados alcançados

	<ul style="list-style-type: none"> Treinamento e Teste de STs de consumo de energia elétrica com a rede neural LSTM: Experimentos computacionais para prever o consumo de energia elétrica para o mês de agosto de 2019;
	<ul style="list-style-type: none"> Módulo de rede neural LSTM acoplado o sistema IoT EnergySaver.
	<ul style="list-style-type: none"> Publicação de Capítulo de Livro como coautor: SANTOS, C. A. M.; SILVA, D. G.; LOPES, J. B. S.; MENESES, A. A. M. Eficiência Energética: Estudos Preliminares para uma Gestão Otimizada de Energia na Amazônia. Produzindo Textos Técnicos em Inglês e Português no IFPA - Campus de Itaituba. 1ed.Curitiba-PR: EDITORA CRV, 2019, v. 1, p. 1-312.

Científicos	<ul style="list-style-type: none"> • Apresentação de trabalho em Congresso: DA SILVA, D. G.; GELLER M. T. B., MOURA M. S. S., MENESES, A. A. M.. "A Deep Learning Prediction Module for the IoT system EnergySaver for Monitoring and Estimating Power Consumption". In 16th Conference on Sustainable Development of Energy, Water and Environment Systems (SDEWES), Dubrovnik, Croácia, 2021a.
	<ul style="list-style-type: none"> • Artigo completo em revista indexada: DA SILVA, D. G.; GELLER M. T. B., MOURA M. S. S., MENESES, A. A. M., Performance Evaluation of LSTM Neural Networks for Consumption Prediction. E-Prime - Advances in Electrical Engineering, Electronics and Energy, 2022, 2, 100030. https://doi.org/10.1016/j.prime.2022.100030.
	<ul style="list-style-type: none"> • Publicação de trabalho como coautor: PEREIRA, Y. S., DA SILVA, D. G.; BARROSO, R. C., MENESES, A. A. M. Lung Computed Tomography Image Segmentation Using U-Net Convolutional Neural Network. LATIN-AMERICAN JOURNAL OF COMPUTING (LAJC), Vol X, Issue 1, January 2023.
	<ul style="list-style-type: none"> • Apresentação de trabalho em Eventos Científicos (co-autor): PEREIRA, Y. S., DA SILVA, D. G.; BARROSO, R. C., MENESES, A. A. M. Lung Computed Tomography Image Segmentation Using U-Net Convolutional Neural Network. XXV Encontro Nacional de Modelagem Computacional (XXV ENMC), XIII Encontro de Ciência e Tecnologia de Materiais (XIII ECTM), 9a Conferência Sul em Modelagem Computacional (9º MCSul) e o IX Seminário e Workshop em Engenharia Oceânica (IX SEMENGO). 2022.
Técnicos	<ul style="list-style-type: none"> • Publicações: DA SILVA, D. G.; TEIXEIRA, Y. B.; VARÃO, D. F. S.; SANTOS, C. A. M.; MOURA, M. S. S.; GELLER, M. T. B.; BENTES, J; MENESES, A. A. M. EnergySaver Software Manual. arXiv preprint arXiv:2107.06664, 2021. Disponível em: <https://arxiv.org/abs/2107.06664>. Acesso em: 15 jul. 2021b.
Acadêmicos	<ul style="list-style-type: none"> • Projetos de Iniciação Científica (Co-orientação): <ul style="list-style-type: none"> ▪ Projeto de pesquisa: “Desenvolvimento de um Módulo de Inteligência Artificial e Internet das Coisas com a Plataforma Nvidia Jetson Nano”. Discente: Rebeca Travassos Praia. ▪ Projeto de pesquisa: “Aplicação de Redes Neurais Convolucionais à Classificação de Imagens”. Discente: Yana Santos.

Fonte: Elaboração própria, 2023.

CAPÍTULO 6

6 DISCUSSÕES

Neste Capítulo vamos discutir a implantação de um módulo de predição de STs para um sistema IoT. Os testes realizados se subdividiram entre números de camadas para a rede LSTM, algoritmos baseados em AM em relação a LSTM e comparação entre a LSTM e BiLSTM; além destas discussões, exemplificamos propostas para uso eficiente de energia elétrica.

6.1 O módulo de Predição com a rede LSTM

O módulo de predição foi desenvolvido e implementado ao sistema EnergySaver (ver mais em Da Silva *et al.*, 2021a; Da Silva *et al.*, 2021b; e Apêndice A), assim sendo, mostrou-se a viabilidade técnica, tendo em vista a capacidade de transmissão de dados entre equipamentos distintos, bem como a exibição dos resultados tanto dos dados coletados quanto da predição, na tela em tempo real para o usuário.

No que se refere ao módulo baseado em AP para um sistema IoT, é importante destacar que algumas propostas já mencionadas anteriormente (Seção 1.1) apresentaram abordagens nesse sentido, como por exemplo Aswani *et al.* (2012), que apresentaram um modelo de sistema para gerenciamento de ar-condicionado, usando um esquema MPC baseado em AP, que ajusta a temperatura ambiente automaticamente a partir de dados coletados no local (sensores instalados na sala) e externamente por meio de um computador que obtém as previsões meteorológicas atualizadas em tempo real via internet, entretanto, em comparação com a nossa abordagem, o sistema proposto pelos autores não realiza a predição de consumo do equipamento.

Outra proposta relacionada ao presente trabalho foi apresentada por Ruano *et al.* (2018) que criaram um sistema de controle preditivo para equipamentos HVAC com objetivo de gerenciar a energia residencial usando uma plataforma IoT. O sistema fornece alertas para diagnósticos, permitindo operação autônoma perpétua em condições razoáveis e condições de iluminação comuns em edifícios e residências.

Todavia não realiza a predição nem utiliza modelos baseados em AP para esse fim, o que torna diferente da nossa abordagem.

Por fim, o exemplo que mais se assemelha ao nosso trabalho, trata-se da proposta de Lee *et al.* (2019) que apresenta uma abordagem para criação de um sistema de previsão de consumo de energia baseado em rede neural LSTM utilizando computação de borda. Foi implementado em um ambiente de escritório dentro do edifício. O sistema exibe os dados da predição de consumo para os usuários individualmente mediante o uso de *gateways*. A principal diferença para nosso trabalho está relacionada a exibição dos dados de predição da rede LSTM, já que os autores destacam que os dados são exibidos diariamente, e no caso da nossa proposta, a predição é realizada mensalmente. Além disso, as ferramentas e equipamentos utilizados para desenvolvimento do sistema de Lee *et al.* (2019) possuem diferenças significativas em relação às utilizados em nossa abordagem.

Assim sendo, destaca-se a originalidade deste trabalho, tendo em vista possuir características semelhantes a de outros autores na literatura atual, no entanto com diferenças significativas em relação ao sistema IoT e a proposta do módulo de predição com LSTM.

6.2 Comparação entre modelos LSTM

Tendo realizado os testes preliminares do sistema EnergySaver com o módulo de predição com sucesso, a próxima etapa foi realizar um estudo com a LSTM para verificar qual a melhor configuração em relação ao números de camadas para o tipo de séries temporais utilizadas neste trabalho.

No primeiro experimento (Seção 5.2 - Resultados preliminares para dois *datasets*), avaliamos modelos de rede LSTM com diferentes números de camadas para previsão de consumo em dois *datasets* (UCI-Household e LABIC-Building) usando o VC-ST. O número de camadas não causou diferença estatisticamente significativa entre os modelos (testes de Kruskal-Wallis com $p = 0,0833$), então selecionamos o modelo L2 para a realização de novos testes.

Outro ponto a ser destacado é a comparação com os resultados obtidos na validação apresentada no trabalho de Da Silva *et al.*(2021), que também utilizou o *dataset* LABIC-Building, porém sem o uso de VC-ST para encontrar a melhor arquitetura de rede. Nesse trabalho, os valores médios para dez execuções foram

RMSE = 0,0410 e MAE = 0,0263. No entanto, ao buscar os melhores modelos com VC-ST, os valores das métricas foram RMSE = 0,0289 e MAE = 0,0205 (Subseção 5.2.2), apresentando menores valores de erro de predição. Assim, a aplicação do VC-ST para a busca dos melhores modelos permitiu melhorar o desempenho geral da rede neural LSTM.

Comparações semelhantes foram utilizadas por vários autores como por exemplo Schirmer *et al.* (2019), que em sua abordagem avaliaram o desempenho de diferentes métodos de regressão para previsão de energia, entre eles a LSTM, onde testaram diferentes números de camadas. Já Mellouli *et al.* (2019), realizaram testes experimentais com modelos de AP para previsão de STs de temperatura interna e consumo de energia em uma câmara fria, e a partir de testes experimentais, concluíram que o melhor modelo de rede LSTM para o problema em questão, dentre outras configurações, seria com três camadas. Em Das *et al.* (2020), os autores realizaram uma análise comparativa entre RNAs, dentre elas a LSTM, e a partir de experimentos, conduziram os testes utilizando-a com três camadas.

Essas abordagens corroboram de certa forma com os resultados obtidos em nossos experimentos, tendo em vista que o modelo de rede LSTM com duas camadas apresentou resultados sem diferença estatisticamente significativa entre os modelos com maior números de camadas.

6.3 Comparação com outros algoritmos de AM

Em relação à comparação da LSTM com os algoritmos XGBoost e RF com VC-ST (Seção 5.3), para o *dataset* UCI-Household, mostrou-se uma leve tendência de melhores resultados da LSTM, embora não confirmado com testes estatísticos de Kruskal-Wallis, tendo em vista que obteve menor RMSE médio e mediano (respectivamente 0,461 e 0,462). Por outro lado, um melhor desempenho do modelo LSTM foi confirmado estatisticamente (Kruskal-Wallis e Dunn) para o *dataset* LABIC-Building.

Algumas abordagens realizaram testes semelhantes, como por exemplo Hadri *et al.* (2021), que avaliaram os algoritmos LSTM, XGBoost e SARIMA para previsão de consumo de eletricidade online, no entanto concluem que para esse problema, nenhum deles supera eficientemente as três estratégias de previsão (univariada, multivariada e multipasso). Já Shin e Woo (2022), avaliaram os algoritmos LSTM, RF

e XGBoost, para previsão de consumo de energia na Coreia, onde a LSTM apresentou o melhor desempenho de previsão em um *dataset* com dados referentes ao período coletado antes da pandemia da COVID-19, e o algoritmo RF apresentou melhores resultados em outro *dataset* com dados coletados pós pandemia. Em Runge e Saloux (2023), os autores realizaram uma comparação entre modelos de IA (dentre outros, RF, XGBoost e LSTM) para previsão de demanda futura de energia, apontando melhores resultados da LSTM e XGBoost.

Após os primeiros testes que apontaram que a rede LSTM foi estatisticamente superior aos outros algoritmos, ela foi escolhida para ser analisada a partir de novos experimentos com novos *datasets*, visando observar se o modelo iria manter os bons resultados.

6.4 Comparação entre LSTM e BiLSTM

Em novos testes, optou-se por utilizar a rede neural BiLSTM, tendo em vista se tratar de uma variação da LSTM unidirecional. Os testes foram realizados a partir de dois novos *datasets*, o Singapore e Tetouan-Zones. É importante destacar que a BiLSTM, como não havia sido utilizada nos testes anteriores com os *datasets* UCI-Household e LABIC-Building, foi testada com os quatro *datasets*.

Deste modo, com os resultados de ambas as redes utilizando VC-ST, foi realizada uma comparação estatística dos resultados obtidos por elas (Seção 5.5), onde o teste de Friedman apontou que para todos os *datasets*, havia diferença estatisticamente significativa entre elas, sendo necessário aplicar o teste *post hoc* de Nemenyi, que indicou haver diferença estatisticamente significativa entre as redes, concluindo que a BiLSTM apresentou resultados superiores à LSTM.

Esses melhores resultados da BiLSTM, corroboram com o resultado obtido por algumas abordagens como por exemplo Siami-Namini *et al.* (2019), que realizam um estudo para comparar o desempenho da LSTM e BiLSTM na previsão de STs, onde apontou que a BiLSTM obteve menores valores de RMSE. No entanto, diferente do nosso trabalho, os autores não realizam os testes estatísticos.

Outra abordagem que realizou comparação semelhante foi proposta por Fang e Yuan (2019) que avaliaram técnicas de melhoria de desempenho para modelos de AP na previsão de STs, usando LSTM e BiLSTM. Do mesmo modo, os resultados apontaram melhores resultados da BiLSTM em todos os testes para a métricas MAE,

MAPE e RMSE. À semelhança do autor citado anteriormente, não há menção a testes estatísticos para os resultados obtidos.

Outro exemplo a ser destacado foi a proposta de Rhif *et al.* (2020), que compararam o desempenho de modelos de AP para STs usando LSTM e BiLSTM. Os resultados dos testes apontaram que ambos os métodos apresentaram pequenas diferenças entre si para as métricas RMSE e R^2 , sendo que a BiLSTM foi melhor na maioria dos casos. Também não há menção sobre testes estatísticos.

Por fim, foram apresentadas as figuras de previsão da LSTM e BiLSTM para os quatro *datasets* supracitados (Subseção 5.5.8), que confirmaram os melhores resultados da BiLSTM em todos os *datasets*.

6.5 Propostas para o uso Eficiente de Energia Elétrica

Conforme mencionado na Seção 1.6 (resultados esperados), com este trabalho pretende-se também fornecer subsídios para promover o uso consciente de energia elétrica, a partir de compartilhamento de ações já utilizadas nos setores públicos e privados, disponíveis na literatura, existentes em nossa região.

É importante destacar que um sistema IoT, como o EnergySaver por exemplo, apenas possibilita ao usuário obter informações em tempo real o consumo elétrico do local (ou equipamento), além da previsão de consumo para o mês seguinte, mediante o módulo baseado em RNP LSTM. Já as ações a serem tomadas para diminuir o consumo, devem ser realizadas a partir de iniciativas dos interessados.

No contexto da Região Amazônica há muito o que se avançar nessa temática, porém, é importante destacar que existem algumas abordagens e programas institucionais que buscam alcançar esse objetivo. Esses programas podem servir como modelos a serem seguidos por outras cidades, universidades e setor privado. Dentre outros, pode-se destacar os projetos descritos a seguir.

- Foi lançada uma campanha educativa pela Universidade Federal do Pará (UFPA)¹⁴ em Outubro de 2019 denominada “É da Nossa Conta”, que visa lembrar a todos os seus frequentadores a responsabilidade que cada um tem em relação ao consumo de energia na Instituição. A campanha faz parte do Plano de Gestão de Logística Sustentável (PLS) da UFPA e busca reduzir os

¹⁴ <https://www.portal.ufpa.br>

gastos com energia elétrica por meio de várias medidas aplicadas, em vários setores da instituição. A informação, que está disponível no portal da universidade por meio da assessoria de Comunicação da UFPA, conta com a adesão dos servidores e da população em geral.

- Outra iniciativa a se destacar foi o Primeiro Hackathon Inova Tapajós, que ocorreu no Campus Tapajós da UFOPA, em novembro de 2019 e teve como propósito promover a cultura de inovação por meio do desenvolvimento de soluções para problemas reais relacionados à sustentabilidade. Por meio da metodologia VIA¹⁵ para inovação aberta, os desafios para os mais de 50 participantes foi baseado na transformação do uso de recursos energéticos e hídricos e a destinação de resíduos sólidos, do campus e da cidade. O evento objetivou a criação de soluções tecnológicas, visando a transferência dessas tecnologias para a sociedade, buscando a eficiência energética, melhoria e agilidade nos serviços conectando, assim, todo o ecossistema de inovação e empreendedorismo da cidade.
- Uma abordagem da literatura foi proposta por Souza (2015), onde o objetivo foi determinar os indicadores de eficiência energética como o consumo específico, fator de carga e custo médio da energia, bem como a determinação dos indicadores de qualidade da energia, mediante análise das grandezas elétricas aferidas nos principais sistemas consumidores de três indústrias com grande representação no cenário regional, no polo industrial de Manaus/AM. A metodologia incluiu levantamento de campo, aplicação de questionários de caracterização energética e medições elétricas. Os resultados apontaram que as indústrias apresentaram indicadores de eficiência, na sua maioria, melhores que a média nacional, contudo, foi constatado que grande parte dos sistemas consumidores das indústrias apresentaram distúrbios relacionados à qualidade da energia elétrica, como baixo fator de potência e conteúdo harmônico. Dessa forma, o autor concluiu que existe um potencial para eficiência energética nos sistemas industriais do referido polo industrial.

¹⁵ <http://via.ufsc.br/>

Diante disso, é possível observar que principalmente as universidades têm buscado cumprir seu papel social promovendo o desenvolvimento local e regional, por meio do tripé formado pelo ensino, pela pesquisa e pela extensão, tornando-se agências produtoras e geradoras de conhecimento, visando cooperar com as indústrias para favorecer a transferência dos conhecimentos e de tecnologia, promovendo com isso o bem estar da sociedade em geral.

Deste modo, buscando-se promover o consumo consciente de energia elétrica, a partir dessas ações, a sociedade em geral pode ser sensibilizada a adotar comportamentos individuais que resultem na redução do consumo e amenize os impactos no meio ambiente. As pequenas ações que o indivíduo pode fazer no seu cotidiano, se replicadas por várias pessoas, terão resultados significativos na redução do consumo de eletricidade.

Por fim, aliando essas ações a possibilidade de implantação de sistemas semelhantes ao EnergySaver, poderão surgir benefícios significativos para a sociedade em geral e principalmente para o meio ambiente.

CAPÍTULO 7

7 CONCLUSÃO E TRABALHOS FUTUROS

Obter eficiência energética é um problema central para a nossa sociedade. Desse modo, tanto no ponto de vista econômico quanto da sustentabilidade, são necessárias ações que possibilitem o desenvolvimento social, de modo que seja possível manter a comodidade social adquirida e, ao mesmo tempo, minimizar os danos aos recursos naturais, valendo-se de ações que promovam a eficiência energética, como por exemplo, a incorporação de novas tecnologias que realizem o monitoramento e predição de consumo, visando reduzir o consumo de energia a partir de ações concretas, conforme destacado por Gardner e Stern (2002) em sua abordagem.

Portanto, um estudo sobre as redes neurais para predição de consumo de energia elétrica na região amazônica tornou-se de extrema importância, principalmente por se tratar de uma área pouco explorada, bem como por ser direcionado a ambiente universitário com características peculiares devido ao elevado consumo de energia elétrica demandado por equipamentos de refrigeração devido as atividades do campus.

Há de se destacar também as peculiaridade climáticas e meteorológicas da região, especificamente na Cidade de Santarém-PA, que de acordo com IBGE (2019) o clima da região é quente e úmido, com temperatura média anual variando de 25 a 28 °C, com regime pluviométrico de aproximadamente 1.920 mm por ano. As temperaturas mais elevadas ocorrem nos meses de junho a novembro, enquanto o período de maior precipitação pluviométrica é de dezembro a maio.

Além disso, foram utilizados também outros *datasets* com dados reais comumente abordados na literatura para predição de STs de consumo de eletricidade. Tratam-se dos *datasets* UCI-Household, Singapore, Tetouan-Zones, além do LABIC-Building supracitado. Um ponto relevante que pode-se destacar é que os *datasets* possuem quatro padrões diferentes: residência, edificação pública, Cidade e País; ou seja, localidades, países e hábitos de consumo diferentes. De forma específica o *dataset* UCI-Household possui dados de consumo de uma residência em Paris na França (equipamentos domésticos); o *dataset* LABIC-Building possui dados de um Prédio universitário (salas de aula, auditórios, laboratórios, e etc.); o *dataset* Tetouan-

Zones que possui dados relativos ao consumo de três zonas de uma estação da rede de distribuição da cidade de Tetouan em Marrocos (residências, empresas, e etc.); e o *dataset* Singapore que possui dados de demanda elétrica em larga escala de Singapura (residências, prédios, cidades, etc..., a nível de País).

Assim sendo, neste trabalho, avaliou-se o desempenho estatístico de RNPs para implantação como módulo de predição ao sistema IoT EnergySaver. Para isso, inicialmente foram realizados estudos preliminares com a RNP LSTM visando analisar estatisticamente diferentes números de camadas (2, 4, 6, 8) utilizando VC-ST = 10 para as STs de cada *dataset*, além de outros parâmetros foram testados e definidos empiricamente. Para esses testes foram usados os *datasets* UCI-Household e LABIC-Building. Os resultados obtidos referente a média da métrica RMSE foram testados com Kruskal-Wallis, onde indicou que o desempenho foi semelhante para todas as camadas em todos os testes nos *datasets*. Assim sendo, escolheu-se a LSTM com duas camadas para realização de todos os testes posteriores.

Os testes seguintes tiveram como objetivo comparar estatisticamente os resultados da LSTM com os algoritmos de AM XBoost e RF, nos mesmos *datasets* utilizados nos testes preliminares, com a mesma métrica, utilizando VC-ST = 10 para as STs de cada *dataset*. Apesar de observar a tendência de menos valores de RMSE médio e mediano da LSTM em relação ao XBoost e RF, para o *dataset* UCI-Household, de acordo com o teste de Kruskal-Wallis não houve diferença estatisticamente significativa entre os modelos (Seção 5.3.1). Já para o *dataset* LABIC-Building, o resultado do teste de Kruskal-Wallis apontou que houve diferença estatisticamente significativa dos valores de RMSE médio e mediano entre a LSTM, XGBoost e RF ($p < 0,0001$), sendo necessário aplicar o teste *post-hoc* de Dunn que indicou que a LSTM superou estatisticamente os demais algoritmos (Seção 5.3.1).

Conforme mencionado anteriormente, um dos objetivos dos testes iniciais foi avaliar desempenho estatístico de RNPs para implantação como módulo de predição ao sistema IoT EnergySaver. Assim sendo, após os resultados obtidos que apontaram que a LSTM obteve melhores resultados, utilizou-se esse modelo para ser usado como módulo acoplado ao sistema.

No entanto, julgou-se necessária a realização novos testes entre a LSTM e outro modelo de AP, neste caso a rede neural BiLSTM, tendo em vista se tratar de uma variação da LSTM unidirecional, levando em consideração os resultados obtidos anteriormente mostraram melhores resultados da LSTM em comparação a algoritmos

de AM. Além disso, para os testes entre LSTM e BiLSTM, optou-se por utilizar dois novos *datasets* com padrões de consumo diferentes dos usados nos testes preliminares, neste caso os *datasets* Singapore e Tetouan-Zones.

Assim sendo, realizou-se a comparação estatística entre os resultados da LSTM e BiLSTM nos quatro *datasets*, a partir da média obtida pela métrica NRMSE com VC-ST = 10, para as STs de cada *dataset*. O teste de Friedman apontou que houve diferença estatisticamente significativa entre as métricas de NRMSE ($p=0,0455$) para todos os *datasets*, tornando-se necessária a utilização do teste *post hoc* de Nemenyi, que por sua vez indicou que a BiLSTM superou estatisticamente a LSTM em todos os *datasets*.

Com base nisso, levando-se em consideração os novos resultados obtidos que apontaram a BiLSTM sendo melhor estatisticamente que a LSTM, optou-se utilizar a BiLSTM no módulo de predição de STs para o sistema IoT EnergySaver.

De modo geral, este estudo foi importante por possibilitar uma análise sobre predição de STs de consumo de energia elétrica com AP, visando realizar a comparação estatística a partir de quatro demandas diferentes (residência, edificação pública, Cidade e País), representadas por quatro *datasets*. Além disso, buscou-se destacar que um dos *datasets* utilizados (LABIC-Building) é fruto de trabalhos de pesquisadores do LABIC-UFOPA (descrito na Seção 4.4.2), cuja ST possui informações relevantes por se tratar de dados de consumo de um prédio localizado cidade da região amazônica brasileira, com características peculiares, podendo servir como base para novos estudos em várias áreas do conhecimento. Portanto, pode-se evidenciar que seu uso nos testes detalhados anteriormente, torna-se um dos principais diferenciais neste trabalho.

Por fim, no âmbito deste trabalho, destaca-se também o uso dos métodos de predição baseados em RNP principalmente por lidarem com grandes quantidades de dados (*big data*), sendo o diferencial para outros modelos baseados em AM. No caso específico dos testes descritos anteriormente, os modelos de RNPs LSTM e BiLSTM apresentaram resultados estatisticamente superiores aos de AM, evidenciando a robustez dos modelos, principalmente a RNP BiLSTM. Também foram descritas algumas propostas para promoção do consumo eficiente de energia elétrica (Seção 6.5), que, aliando essas ações a possibilidade de implantação de sistemas semelhantes ao EnergySaver, poderão surgir benefícios significativos para a sociedade em geral e principalmente para o meio ambiente.

7.1 Trabalhos Futuros

Um dos principais valores de um trabalho de pesquisa como o desenvolvido nesta Tese é o aprofundamento sobre tema estudado, buscando agregar novos conhecimentos. Deste modo, para trabalhos futuros pretende-se:

- Testar e validar o modelo *Reservoir Computing* (BIANCHI *et al.*, 2020; MOON *et al.*, 2019). Trata-se de um tipo de RNR que permite uma modelagem que não exige a necessidade de treinamento dos pesos da camada de entrada e nem dos pesos internos da rede (*reservoir*), apenas os pesos da camada de saída (*readout*) são treinados
- Testar e validar o *Transfer Learning* (OTOVIĆ, 2022; WEISS *et al.*, 2016), que trata reutilização de um modelo pré-treinado para um novo problema, ou seja, explora o conhecimento adquirido em uma tarefa, por exemplo uma rede neural treinada em um determinado *dataset*, para melhorar a generalização em outro *dataset*, geralmente maior, visando resolver um novo problema.
- Utilizar novos *datasets* com STs de consumo de energia elétrica usados por pesquisadores atualmente para predição.

REFERÊNCIAS

- ABESCO. **Brasil desperdiça meia itaipu por ano**. Associação Brasileira das Empresas de Serviços de Conservação de Energia, 2018. URL <http://www.abesco.com.br/novidade/brasil-desperdica-meia-itaipu-por-ano/>. Acesso em: 15 Jul. 2020.
- AHMADI A., NABIPOUR M., MOHAMMADI-IVATLOO B., AMANI A. M., RHO S., and PIRAN M. J., “**Long-Term Wind Power Forecasting Using Tree-Based Learning Algorithms**”. IEEE Access, vol. 8, pp. 151511–151522, 2020, doi: 10.1109/ACCESS.2020.3017442.
- AI MAMUN A., SOHEL M., MOHAMMAD N., HAQUE SUNNY M. S., DIPTA D. R., and HOSSAIN E., “**A Comprehensive Review of the Load Forecasting Techniques Using Single and Hybrid Predictive Models**”. IEEE Access, vol. 8, pp. 134911–134939, 2020, doi: 10.1109/ACCESS.2020.3010702.
- AMOÊDO, L., VARÃO, D., JÚNIOR, J., MENESES, A. Aquisição de dados do consumo elétrico em uma edificação da UFOPA utilizando conceitos de IoT. 10.22533/at.ed.1572002034. 2020.
- ARJOMANDI-NEZHAD A., AHMADI A., TAHERI S., FOTUHI-FIRUZAbad M., MOEINI-AGHTAIE M., e LEHTONEN M., “**Pandemic-Aware Day-Ahead Demand Forecasting using Ensemble Learning**”. IEEE Access, vol. 10, pp. 7098–7106, 2022, doi: 10.1109/ACCESS.2022.3142351.
- ASWANI A., MASTER N., TANEJA J., CULLER D., e TOMLIN C. **Reducing Transient and Steady State Electricity Consumption in HVAC Using Learning-Based Model-Predictive Control**. Proceedings of the IEEE, vol. 100, no. 1, pp. 240–253, Jan. 2012, doi: 10.1109/JPROC.2011.2161242.
- ATZORI, L.; IERA, A.; MORABITO, G. **The internet of things: A survey**. Computer networks, v. 54, n. 15, p. 2787-2805, 2010.
- BARROW, D.; DRAYER, I.; ELLIOTT, P.; GAUT, G.; OSTING, B. **Ranking rankings: an empirical comparison of the predictive power of sports ranking methods**. Journal of Quantitative Analysis in Sports, June 2013, Vol.9(2), pp.187-202. DOI: 10.1515/jqas-2013-0013. 2013.
- BIANCHI, F. M. *et al.* **Reservoir computing approaches for representation and classification of multivariate time series**. IEEE transactions on neural networks and learning systems, v. 32, n. 5, p. 2169-2179, 2020.
- BIAU, G. **Analysis of a random forests model**. The Journal of Machine Learning Research, v. 13, n. 1, p. 1063-1095, 2012.

BOOTE, D. N.; BEILE, P. **“The quality of dissertation literature reviews: A missing link in research preparation”**. In: Annual meeting of the American Educational Research Association, San Diego, CA. 2004.

BOUKTIF, S.; FIAZ, A.; OUNI, A.; SERHANIM, M. A. **Optimal Deep Learning LSTM Model for Electric Load Forecasting using Feature Selection and Genetic Algorithm: Comparison with Machine Learning Approaches**. *Energies*. 1636; 2018.

BRASIL. Decreto nº 9.854, de 25 de junho de 2019. **Institui o Plano Nacional de Internet das Coisas e dispõe sobre a Câmara de Gestão e Acompanhamento de Desenvolvimento de Sistemas de Comunicação Máquina a Máquina e Internet das Coisas**. URL http://www.planalto.gov.br/ccivil_03/_ato2019-2022/2019/decreto/D9854.htm. Acesso em: 23 Set. 2019.

BREIMAN L. **“Random Forests”**. *Machine Learning*, vol. 45, pp. 5–32, 2001.

BUDUMA, N.; LOCASCIO, N. **Fundamentals of deep learning: designing next-generation machine intelligence algorithms**. O Reilly Media, Inc.", 2017.

CARVALHO, André, *et al.* **Inteligência Artificial: Uma abordagem de Aprendizado de Máquina**. Rio de Janeiro: Ltc, 2011.

CHAGAS, E. T. O. **Deep Learning e suas aplicações na atualidade**. *Revista Científica Multidisciplinar Núcleo do Conhecimento*. Ano 04, Ed. 05, Vol. 04, pp. 05-26 Maio de 2019. ISSN: 2448-0959.

CHEN T. e GUESTRIN C., **“XGBoost: A Scalable Tree Boosting System”**. in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

CHEN T. e HE T. **“XGBoost : eXtreme Gradient Boosting”**. R package version 0.71-2. pp. 1-4, 2015.

CHOLLET F., **Deep Learning with Python**. Shelter Island, NY, USA, Nov. 2018.

CHUNG, J., GULCEHRE, C., Cho, K., e BENGIO, Y. **Gated feedback recurrent neural networks**. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, volume 37 of ICML’15, pages 2067-2075, 2015.

CLAESEN, M.; MOOR, B. D. **Hyperparameter search in machine learning**. arXiv preprint arXiv:1502.02127, 2015.

CORTES, C.; VAPNIK, V. **Support vector machine**. *Machine learning*, v. 20, n. 3, p. 273-297, 1995.

DA SILVA D.G, GELLER M. T. B, MOURA M., e MENESES A. A. M., **“A Deep Learning Prediction Module for the IoT system EnergySaver for Monitoring and**

Estimating Power Consumption". In 16th Conference on Sustainable Development of Energy, Water and Environment Systems (SDEWES), Dubrovnik, Croácia, 2021a.

DA SILVA, D. G.; GELLER M. T. B., MOURA M. S. S., MENESES, A. A. M., **Performance Evaluation of LSTM Neural Networks for Consumption Prediction**. E-Prime - Advances in Electrical Engineering, Electronics and Energy, 2, 100030. <https://doi.org/10.1016/j.prime.2022.100030>. 2022.

DA SILVA, D. G.; TEIXEIRA, Y. B.; VARÃO, D. F. S.; SANTOS, C. A. M.; MOURA, M. S. S.; GELLER, M. T. B.; BENTES, J; MENESES, A. A. M. **EnergySaver Software Manual**. arXiv preprint arXiv:2107.06664, 2021. Disponível em: <<https://arxiv.org/abs/2107.06664>>. Acesso em: 15 jul. 2021b.

DANGETI, P. **Statistics for Machine Learning: Techniques for Exploring Supervised, Unsupervised, and Reinforcement Learning Models with Python and R**. Packt Publishing: Birmingham, UK, 2017.

DAS, A., ANNAQEEB, M. K., AZAR, E., NOVAKOVIC, V. & KJÆRGAARD, M. B. **Occupant-centric miscellaneous electric loads prediction in buildings using state-of-the-art deep learning methods**. Applied Energy 269, 115135. 2020.

DE SOUSA A. F. G., FURTADO H. C. M., MACÊDO W. N., and MENESES A. A. M. **"Analysis of Artificial Neural Network Point Forecasting Models and Prediction Intervals for Solar Irradiance Estimation"**. DINCON. American Journal of Engineering and Applied Sciences, vol. 13, no. 3, pp. 347–357, Mar. 2020, doi: 10.3844/ajeassp.2020.347.357.

DEMŠAR J., **"Statistical comparisons of classifiers over multiple data sets"**. Journal of Machine Learning Research, vol. 7, pp. 1–30, 2006.

DIETTERICH, T.; **Machine Learning for Sequential Data: A Review**. Structural, Syntactic, and Statistical Pattern Recognition, pp. 15-30, 2002.

DMITRIENKO A., CHUANG-STEIN C., and D'AGOSTINO R. **"Pharmaceutical Statistics Using SAS"**. vol. 53, no. 9. SAS Publishing, 2007.

DRESCH, A.; LACERDA, D. P.; ANTUNES Jr., J. A. V. **"Design science research: método de pesquisa para avanço da ciência e tecnologia"**. Bookman, Porto Alegre/RS, 2015.

EISDB. **Smart metering implementation programme - a report on progress of the realisation of smart meter consumer benefits**. Energy Industrial Strategy Department for Business, GOV.UK, 2019. URL https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/830668/smart-meters-benefits-realisation.pdf. Acesso em: 25 Ago. 2020.

EPE - EMPRESA DE PESQUISA ENERGÉTICA, **"Balanço Energético Nacional 2020: Ano base 2019"**. Rio de Janeiro : EPE, pp. 7–295, 2020.

FACELI, K.; LORENA, A. C.; GAMA, J.; CARVALHO, A. C. P. L. F. **Inteligência artificial: uma abordagem de aprendizado de máquina**. Rio de Janeiro: LTC. 378p. 2011.

FANG, X.; YUAN, Z. **Performance enhancing techniques for deep learning models in time series forecasting**. Engineering Applications of Artificial Intelligence, v. 85, p. 533-542, 2019.

FARSI B., AMAYRI M., BOUGUILA N., e EICKER U., “**On short-term load forecasting using machine learning techniques and a novel parallel deep LSTM-CNN approach**”. IEEE Access, vol. 9, pp. 31191–31212, 2021, doi: 10.1109/ACCESS.2021.3060290.

FAUSETT, L. V. **Fundamentals of Neural Networks - Architectures, Algorithms, and Applications**. Prentice Hall International Editions. USA, New Jersey, 1994.

FERNÁNDEZ-MARTÍNEZ, D.; JARAMILLO-MORÁN, M.A. **Multi-Step Hourly Power Consumption Forecasting in a Healthcare Building with Recurrent Neural Networks and Empirical Mode Decomposition**. Sensors 2022, 22, 3664.

FRACARI, F.; SANTOS, I.; SANCHEZ, G. **Smart Grid: uma nova forma de controle de Energia Elétrica**. Revista de Empreendedorismo, Inovação e Tecnologia, v. 2, n. 1, p. 15-22, 2015.

FRANK R. J., DAVEY N., e S. P. HUNT, “**Time series prediction and neural networks**”. Journal of Intelligent and Robotic Systems: Theory and Applications, vol. 31, no. 1–3, pp. 91–103, 2001, doi: 10.1023/A:1012074215150. 2001.

FREITAS, V., GUIMARÃES, A., FERREIRA, C., e ALVES, S. **Edifícios existentes - Medidas de melhoria de desempenho energético e da qualidade do ar interior**. Editora: Verlag Dashöfer. 1º Ed. ISBN: 978-972-8646-16-5. 2011.

FRIEDMAN, J. H. **Stochastic gradient boosting**. Computational Statistics & Data Analysis. Elsevier, v. 38, n. 4, p. 367–378, 2002.

FRIEDMAN, M. **The use of ranks to avoid the assumption of normality implicit in the analysis of variance**. Journal of the American Statistical Association, 32(200), 675-701. <http://dx.doi.org/10.1080/01621459.1937.10503522>. 1937.

FUKADA, T., SCHUSTER, M., AND SAGISAKA, Y. (1999). **Phoneme boundary estimation using bidirectional recurrent neural networks and its applications**. Systems and Computers in Japan, 30(4):20–30.

GARCÍA, S., DERRAC, J. e HERRERA F. **Un Tutorial Metodológico para hacer Comparaciones Estadísticas con Tests No Paramétricos en Propuestas de Minería de Datos**. Actas del V Simposio de Teoría y Aplicaciones de Minería de Datos., pp. 155-164. 2010a.

GARCÍA, S., FERNÁNDEZ A., LUENGO J., e HERRERA F., “**Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power**”. Information Sciences, vol. 180, no. 10, pp. 2044–2064, doi: 10.1016/j.ins.2009.12.010. 2010b.

GARDNER, G. T., e STERN, P. C. (2002). **Environmental problems and human behavior (2nd edition)**. Boston, MA: Pearson Custom Publishing. 2002.

GELLER M. T. B. e MENESES A. A. M., “**Modelling IoT Systems with UML: A Case Study for Monitoring and Predicting Power Consumption**”. American Journal of Engineering and Applied Sciences, vol. 14, no. 1, pp. 81–93, Jan. 2021, doi: 10.3844/ajeassp.2021.81.93.

GERRIG, R. J.; ZIMBARDO, P. G. **A psicologia e a vida**. 16. ed. Porto Alegre: Artmed, 2005.

GOODFELLOW, I.; BENGIO, Y., COURVILLE, A. **Deep Learning**. Cambridge: MIT Press, URL: <http://www.deeplearningbook.org>, 2016.

GORETTI C.; e BIAGGINI S. **Metodologia de pesquisa e construção do conhecimento.**” 4. ed. Florianópolis: UFSC, 2014. E-book.

GRAVES, A. AND SCHMIDHUBER, J. (2005). **Frame-wise phoneme classification with bidirectional LSTM and other neural network architectures**. Neural Networks, 18(5-6):602–610.

GRAVES, A. AND SCHMIDHUBER, J. (2009). **Offline handwriting recognition with multidimensional recurrent neural networks**. In Advances in Neural Information Processing Systems 21, pages 545–552. MIT Press, Cambridge, MA.

GRAVES, A., LIWICKI, M., FERNANDEZ, S., BERTOLAMI, R., BUNKE, H., AND SCHMIDHUBER, J. (2009). **A novel connectionist system for improved unconstrained handwriting recognition**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(5).

GROSS G. e GALIANA F. D., “**Short-Term Load Forecasting**”. Proceedings of the IEEE, vol. 75, no. 12, pp. 1558–1573, 1987, doi: 10.1109/PROC.1987.13927.

HADRI, S.; NAJIB, M.; BAKHOUYA, M.; FAKHRI, Y.; EL ARROUSSI, M. **Performance Evaluation of Forecasting Strategies for Electricity Consumption in Buildings**. Energies 2021, 14, 5831.

HASAN, M.N.; TOMA, R.N.; NAHID, A.-A.; ISLAM, M.M.M.; KIM, J.-M. **Electricity Theft Detection in Smart Grid Systems: A CNN-LSTM Based Approach**. Energies 2019, 12, 3310. <https://doi.org/10.3390/en12173310>.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. 2. ed. Stanford: Springer, 2009.

HAYKIN S. “**Neural networks: a comprehensive foundation**”. 2nd ed. Pearson, 1998.

HAYKIN, S. **Neural Networks and Learning Machines**. Third edition. Pearson Education India, 2009.

HIPPERT H. S., PEDREIRA C. E., e SOUZA R. C., “**Neural networks for short-term load forecasting: A review and evaluation**”. IEEE Transactions on Power Systems, vol. 16, no. 1, pp. 44–55, 2001, doi: 10.1109/59.910780.

HOCHREITER S. e SCHMIDHUBER J., “**Long Short-Term Memory**”. Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

HONG T., “**Energy Forecasting: Past , present and future**”. Foresight: The International Journal of Applied Forecasting, vol. 32, pp. 43–48, 2013.

HONG T., PINSON P., WANG Y., WERON R., YANG D., and ZAREIPOUR H., “**Energy Forecasting: A Review and Outlook**”. IEEE Open Access Journal of Power and Energy, vol. 7, no. October, pp. 376–388, 2020, doi: 10.1109/OAJPE.2020.3029979.

HOU T. *et al.*, “**A novel short-term residential electric load forecasting method based on adaptive load aggregation and deep learning algorithms**”. Energies, vol. 14, no. 22, 2021, doi: 10.3390/en14227820.

HYNDMAN R. e ATHANASOPOULOS G., “**Forecasting: principles and practice**”. 3rd ed. Melbourne, Australia: OTexts. [Online]. Available: OTexts.com/fpp3. 2018.

IANNACE, G.; CIABURRO, G.; TREMATERRA, A. **Wind Turbine Noise Prediction Using Random Forest Regression**. Machines, 7, 69. 2019. doi: 10.3390/machines7040069.

IBGE - Instituto Brasileiro de Geografia e Estatística. **Dados do Estado do Pará-PA em 2019**. Disponível: <http://www.ibge.gov.br/estadosat/>. Acesso em: 15 set. 2020.

IRREK, W.; THOMAS, S. **Defining Energy Efficiency**. Wuppertal Institut, Wuppertal, Alemanha, 2008.

ISO 50001 – **SISTEMA DE GESTÃO DE ENERGIA**. 2016. Disponível em: <<http://www.abesco.com.br/pt/iso-50001-gestao-de-energia/>>. Acesso em: 22 jun. 2022.

IVAKHNENKO, A. G., LAPA, V. G., e MCDONOUGH, R. N. (1967). **Cybernetics and forecasting techniques**. NY: American Elsevier Publishing Company, 1967.

JURAFSKY D., MARTIN J. H. **Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition**. Prentice Hall Prentice Hall. 2000.

KAUR, D. *et al.* **Smart grid energy management using RNN-LSTM: A deep learning-based approach**. In: 2019 IEEE global communications conference (GLOBECOM). IEEE, 2019, p.9 -13.

KHAN N., HAQ I. U., KHAN S. U., RHO S., LEE M. Y. e BAIK S. W., “**DB-Net: A novel dilated CNN based multi-step forecasting model for power consumption in integrated local energy systems**”. International Journal of Electrical Power & Energy Systems, vol. 133, p. 107023, Dec. 2021, doi: 10.1016/j.ijepes.2021.107023.

KIM T.-Y. e CHO S.-B., “**Predicting the Household Power Consumption Using CNN-LSTM Hybrid Networks**”. 2018, pp. 481–490. doi: 10.1007/978-3-030-03493-1_50.

KITCHENHAM, Barbara; CHARTERS, Stuart. “**Guidelines for performing Systematic Literature Reviews in Software Engineering**”. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report, 2007.

KOHAVI R. “**A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection**”. International Joint Conference of Artificial Intelligence, no. June, 1995.

LE T., VO M. T., VO B., HWANG E., RHO S. e BAIK S. W., “**Improving Electric Energy Consumption Prediction Using CNN and BiLSTM**”. Applied Sciences, vol. 9, no. 20, p. 4237, Oct. 2019, doi: 10.3390/app9204237.

LECUN, Y., BOSER, B., DENKER, J. S., HENDERSON, D., HOWARD, R. E., HUBBARD, W., E JACKEL, L. D. **Backpropagation applied to handwritten zip code recognition**. Neural computation, v. 1, n. 4, p. 541-551, 1989.

LECUN, Y. *et al.* **Convolutional networks and applications in vision**. In Proceedings of 2010 IEEE International Symposium on Circuits and Systems, pages 253-256, 2010.

LECUN Y., BENGIO Y. e HINTON G., “**Deep learning**”. Nature, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

LEE, S. H.; LEE, T.; KIM, S.; PARK, S. **Energy consumption prediction system based on deep learning with edge computing**. In: 2019 IEEE 2nd International Conference on Electronics Technology (ICET). IEEE, 2019. p. 473-477.

LEFRANÇOIS, G. R. **Teorias da Aprendizagem**. 5. ed. São Paulo: Cengage Learning, 2008.

LEITE, C. M.; LEITE, D. M. **Proteção Contra Descargas Atmosféricas**. São Paulo: Oficina da Mydia Editora, 2011.

LIANG, Y.; DENG, J.; CUI, B. **Bidirectional LSTM: An Innovative Approach for Phishing URL Identification**. In: SPRINGER, 2019. International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing. [S.l.], 2019. p. 326-337.

LINNAINMAA, S. **The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors** [Master's thesis], Univ. Helsinki. 1970.

MELLOULI, N., AKERMA, M., HOANG, M., LEDUCQ, D., DELAHAYE, A. **Deep Learning Models for Time Series Forecasting of Indoor Temperature and Energy Consumption in a Cold Room**. In: Nguyen, N., Chbeir, R., Exposito, E., Aniorté, P., Trawiński, B. (eds) Computational Collective Intelligence. ICCCI 2019. Lecture Notes in Computer Science(), vol 11684. Springer, Cham. https://doi.org/10.1007/978-3-030-28374-2_12. 2019.

MCCULLOCH, W. S.; PITTS, W. **A logical calculus of the ideas immanent in nervous activity**. The bulletin of mathematical biophysics, v. 5, p. 115-133, 1943.

MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. **Foundations of machine learning**. MIT press, 2018.

MOON, J. *et al.* **Temporal data classification and forecasting using a memristor-based reservoir computing system**. Nature Electronics, v. 2, n. 10, p. 480-487, 2019.

MORETTIN, P. A.; TOLOI, C. **Análise de séries temporais**. In: Análise de séries temporais. 2006. p. 538-538.

MOURA P., MORENO J. I., LÓPEZ G., and ALVAREZ-CAMPANA M., **“IoT Platform for Energy Sustainability in University Campuses”**. Sensors, vol. 21, no. 2, p. 357, Jan. 2021, doi: 10.3390/s21020357.

MYERS, D. G. **Psicologia**. 7. ed. Rio de Janeiro: LTC, 2006.

NELSON, M. M. e ILLINGWORTH, W. T. **A Practical Guide to Neural Nets**. Boston, MA: Addison-Wesley Longman Publishing Company, Inc. USA, 1990.

NEMENYI, P. B. **Distribution-free multiple comparisons**. Princeton University, 1963.

NOBRE, M. J. Aprendizagem e memória. In: BRANDÃO, M. L. **As bases biológicas do comportamento: introdução à neurociência**. São Paulo: Editora Pedagógica e Universitária, 2004. Cap. 6.

NVIDIA CORPORATION. **Deep Learning in a Nutshell: History and Training**. Dec 16, 2015. Disponível em: <<https://developer.nvidia.com/blog/deep-learning-nutshell-history-training>>. Acesso em: 20 Maio 2021.

OLAH, C. **Understanding LSTM Networks** [Blog]. 2015. Disponível online: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/> (acessado em 10 de Fevereiro de 2020).

OTOVIĆ, E. *et al.* **Intra-domain and cross-domain transfer learning for time series data—How transferable are the features?**. Knowledge-Based Systems, v. 239, p. 107976, 2022.

OZER I., EFE S. B., e OZBAY H. **“A combined deep learning application for short term load forecasting”**. Alexandria Engineering Journal, vol. 60, no. 4, pp. 3807–3818, 2021, doi: 10.1016/j.aej.2021.02.050.

PEFFERS K., *et al.* **“A design science research methodology for information systems research”**. Journal of Management Information Systems, [S. l.], v. 24, n. 3, p. 45–77, 2007. Disponível em: <https://doi.org/10.2753/MIS0742-1222240302>.

PEREIRA, Y. S., DA SILVA, D. G.; BARROSO, R. C., MENESES, A. A. M. **Lung Computed Tomography Image Segmentation Using U-Net Convolutional Neural Network**. XXV Encontro Nacional de Modelagem Computacional (XXV ENMC), XIII Encontro de Ciência e Tecnologia de Materiais (XIII ECTM), 9a Conferência Sul em Modelagem Computacional (9º MCSul) e o IX Seminário e Workshop em Engenharia Oceânica (IX SEMENGO). 2022.

RAFI S. H., AL-MASOOD N., DEEBA S. R., e HOSSAIN E., **“A short-term load forecasting method using integrated CNN and LSTM network”**. IEEE Access, vol. 9, pp. 32436–32448, 2021, doi: 10.1109/ACCESS.2021.3060654.

REGGIO G., **“A UML-based proposal for IoT system requirements specification”**. in Proceedings of the 10th International Workshop on Modelling in Software Engineering - MiSE '18, 2018, pp. 9–16. doi: 10.1145/3193954.3193956.

REIMERS, N.; GUREVYCH, I. **Optimal hyperparameters for deep lstm-networks for sequence labeling tasks**. arXiv preprint arXiv:1707.06799, 2017.

RHIF, M. *et al.* **Deep learning models performance for NDVI time series prediction: a case study on north west Tunisia**. In: 2020 Mediterranean and Middle-East Geoscience and Remote Sensing Symposium (M2GARSS). IEEE, 2020. p. 9-12.

ROBLES-RAMIREZ D. A., ESCAMILLA-AMBROSIO P. J. e TRYFONAS T. **“IoTsec: UML Extension for Internet of Things Systems Security Modelling”**. in 2017 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE), Nov. 2017, pp. 151–156. DOI: 10.1109/ICMEAE. 2017. 20.

RODRIGUES, A. M.; DE PIEDADE, A. C.; BRAGA, A. M. **Térmica de Edifícios**. Lisboa: Ed. Orion. ISBN: 9789728620134. 2009.

RUANO A., SILVA S., DUARTE H. e FERREIRA P. M., “**Wireless Sensors and IoT Platform for Intelligent HVAC Control**”. Applied Sciences, vol. 8, no. 3, p. 370, Mar. 2018, doi: 10.3390/app8030370.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. **Learning representations by back-propagating errors**. Nature, v. 323, n. 6088, p. 533-536, 1986.

RUNGE, J.; SALOUX, E. **A comparison of prediction and forecasting artificial intelligence models to estimate the future energy demand in a district heating system**. Energy, p. 126661, 2023.

RUSSEL, S.; NORVIG, P. **Inteligência Artificial**. Rio de Janeiro: Elsevier, 2013.

SALAM, A.; EL HIBAOU, A. **Comparison of machine learning algorithms for the power consumption prediction - Case study of tetouan city**. in: Proceedings of 2018 6th International Renewable and Sustainable Energy Conference, IRSEC 2018, 2018.

SALAM, A.; EL HIBAOU, A. **Energy consumption prediction model with deep inception residual network inspiration and LSTM**. Mathematics and Computers in Simulation, v. 190, p. 97-109, 2021.

SANTOS, C. A. M.; SILVA, D. G.; LOPES, J. B. S.; MENESES, A. A. M. **Eficiência Energética: Estudos Preliminares para uma Gestão Otimizada de Energia na Amazônia**. Produzindo Textos Técnicos em Inglês e Português no IFPA - Campus de Itaituba. 1ed. Curitiba-PR: EDITORA CRV, 2019, v. 1, p. 1-312.

SCHIRMER, P.A.; MPORAS, I.; POTAMITIS, I. **Evaluation of regression algorithms in residential energy consumption prediction**. In Proceedings of the 2019 3rd European Conference on Electrical Engineering and Computer Science (EECS), Athens, Greece, 28–30, December 2019; pp. 22–25.

SCHMIDHUBER, J. **Deep Learning in Neural Networks: An Overview**. Neural Networks. 61. 10.1016/j.neunet.2014.09.003. 2014.

SCHMIDHUBER, J. **Critique of Paper by “Deep Learning Conspiracy” (Nature 521 p 436)**. <http://people.idsia.ch/~juergen/deep-learning-conspiracy.html>. 2015.

SCHUSTER, M. e PALIWAL K. K. **Bidirectional recurrent neural networks**. in IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, Nov. 1997, doi: 10.1109/78.650093.

SERPANOS, D. e WOLF, M., **Internet-of-Things (IoT) Systems**. Cham: Springer International Publishing, 2018. doi: 10.1007/978-3-319-69715-4.

SHAQOUR A., T. Ono, HAGISHIMA A. e FARZANEH H., “**Electrical demand aggregation effects on the performance of deep learning-based short-term load forecasting of a residential building**”. *Energy and AI*, vol. 8, no. November 2021, p. 100141, 2022, doi: 10.1016/j.egyai.2022.100141.

SHARFUDDIN A.A., TIHAMI M.N., ISLAM M.S. **A deep recurrent neural network with BiLSTM model for sentiment classification**. International conference on bangla speech and language processing (ICBSLP), IEEE, pp. 1-4. 2018.

SIAMI-NAMINI, Sima; TAVAKOLI, Neda; NAMIN, Akbar Siami. **The performance of LSTM and BiLSTM in forecasting time series**. In: 2019 IEEE International Conference on Big Data (Big Data). IEEE, 2019. p. 3285-3292.

SHIN, S.-Y.; WOO, H.-G. **Energy Consumption Forecasting in Korea Using Machine Learning Algorithms**. *Energies* 2022, 15, 4880.

SINGH, A. P. *et al.* **Machine learning-based occupancy estimation using multivariate sensor nodes**. In: 2018 IEEE Globecom Workshops (GC Wkshps). IEEE, 2018. p. 1-6.

SOUZA, R. P. **Potencial para eficiência energética em segmentos do polo industrial de Manaus/AM**. Dissertação (Mestrado em Engenharia de Recursos da Amazônia (PPG-ENGRAM)) - Universidade Federal do Amazonas - UFAM, Manaus, AM, 178 f., 2015.

SRIVASTAVA N., HINTON G., KRIZHEVSKY A., SUTSKEVER I., SALAKHUTDINOV R. **Dropout: A simple way to prevent neural networks from overfitting**. *J. Mach. Learn. Res.*, 15 (56), pp. 1929-1958. 2014.

TEIXEIRA, Y., VARÃO, D., BENTES, J. e MENESES, A. **Desenvolvimento de uma plataforma de IoT para monitoramento de consumo de energia elétrica**. Simpósio de Computação do Oeste do Pará 1: 2,36–40. <http://www.ufopa.edu.br/portaldeperiodicos/index.php/anaisscoop>. 2017.

THRAMBOULIDIS, K. e CHRISTOULAKIS, F. “**UML4IoT—A UML-based approach to exploit IoT in cyber-physical manufacturing systems**”. *Computers in Industry*, vol. 82, pp. 259–272, 2016, doi: 10.1016/j.compind.2016.05.010.

TYRALIS, H. e PAPACHARALAMPOUS, G. **Variable selection in time series forecasting using random forests**. *Algorithms*, v. 10, n. 4, p. 114, 2017.

VARÃO, D. F. S.; TEIXEIRA, Y. B.; BENTES, J.; MENESES, A. A. M. **Desenvolvimento de uma Plataforma de IoT para Monitoramento de Consumo de Energia Elétrica**. In: I Simpósio de Computação do Oeste do Pará - SCOOP 2017, 2017, Santarém. Livro de Resumos - SCOOP 2017, 2017.

VARÃO, D. F. S.; TEIXEIRA, Y. B.; BENTES, J.; MENESES, A. A. M. **Desenvolvimento de um Sistema IoT para Monitoramento de Energia Elétrica**. In:

Simpósio Brasileiro de Engenharia Física, 2018, Foz do Iguaçu. Anais do Simpósio Brasileiro de Engenharia Física, 2018.

WEISS, K., KHOSHGOFTAAR, T.M. & WANG, D. **A survey of transfer learning.** *J Big Data* **3, 9 (2016)**. <https://doi.org/10.1186/s40537-016-0043-6>.

WITTEN I. e FRANK E., “**Data Mining: Practical Machine Learning Tools and Techniques**”. 2005.

WONG K. S. e KIM M. H., “**Privacy Protection for Data-Driven Smart Manufacturing Systems**,” *International Journal of Web Services Research*, vol. 14, no. 3, pp. 17–32, Jul. 2017, doi: 10.4018/IJWSR.2017070102.

ZAREMBA, W.; SUTSKEVER, I.; VINYALS, O. **Recurrent Neural Network Regularization**. arXiv e-prints, p. arXiv:1409.2329, set. 2014.

ZHANG, G.; PATUWO, B. E.; HU, M. Y. **Forecasting with artificial neural networks: The state of the art.** *International journal of forecasting*, v. 14, p. 35–62. DOI: 10.1016/S0169-2070(97)00044-7. 1998.

ZHAO H. X. e MAGOULÈS F., “**A review on the prediction of building energy consumption**”. *Renewable and Sustainable Energy Reviews*, vol. 16, no. 6, pp. 3586–3592, 2012, doi: 10.1016/j.rser.2012.02.049.

ZHAO, Yu *et al.* **Applying deep bidirectional LSTM and mixture density network for basketball trajectory prediction.** *Optik*, v. 158, p. 266-272, 2018.

APÊNDICE A – IMPLANTAÇÃO DE SISTEMAS COM IOT E AP

Viabilidade e Direcionamentos para Implantação de sistemas de monitoramento semelhantes usando IoT e AP

Conforme mencionado na Seção 3.9, existem iniciativas de órgãos governamentais que visam estimular a implementação e desenvolvimento de sistemas baseados em IoT, como por exemplo o Plano Nacional de Internet das Coisas do Governo Federal.

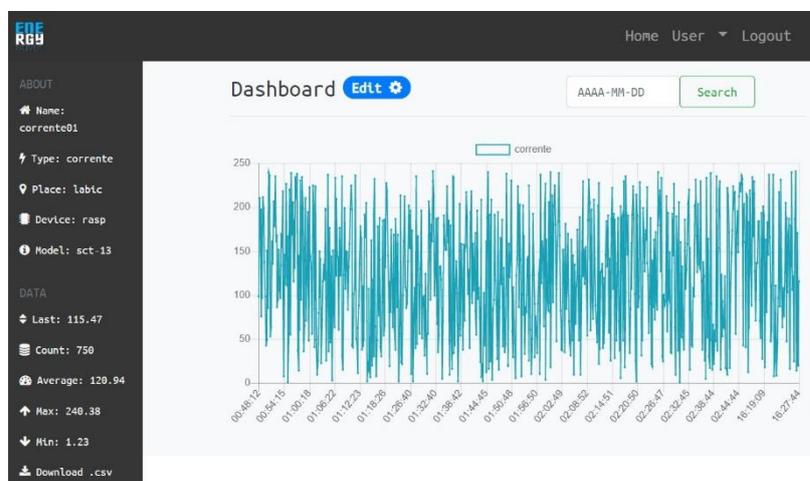
É importante destacar que esse tipo de sistema pode ser viável e acessível para qualquer pessoa desenvolver, tendo em vista a utilização de equipamentos de baixo custo disponíveis no mercado, e que podem ser adquiridos em sites de vendas online. Os principais componentes usados para o desenvolvimento do EnergySaver, por exemplo, foram (mais detalhes em Da Silva *et al.*, 2021).

- a) **Multimedidor**: instrumento digital microprocessado que foi desenvolvido para possibilitar o monitoramento da qualidade de energia trifásica. Neste sistema utilizou-se o multimedidor de grandezas elétricas MRI-TF92.
- b) **Arduino com sensor de corrente elétrica**: para aquisição de dados de consumo de equipamentos individualizados, utiliza-se este equipamento. O medidor de energia consiste basicamente em um sensor de corrente e um de tensão conectados a um circuito microprocessado, que no caso é o Arduino.
- c) **Raspberry Pi3**: considerado um miniPC que tem o tamanho de um cartão. A sua placa é um computador barato, portátil e versátil, usado principalmente em projetos de programação, robótica e em iniciativas em geral com software e hardware livre. No EnergySaver esse equipamento foi usado para receber os dados de consumo de energia elétrica coletados pelo multimedidor.
- d) **Mosquitto broker**: baseado no protocolo MQTT que fornece um método leve de executar mensagens usando um modelo de publicação / assinatura de um equipamento para outro usando redes sem fio.
- e) **MongoDB**: é um banco de dados orientado a documentos (*document database*) no formato JSON, usado para armazenar os dados de consumo coletados pelo equipamento e recebidos pelo servidor.
- f) **Flask micro-framework**: é um micro-framework multiplataforma que provê um modelo simples para o desenvolvimento web, sendo bastante utilizado

para criação de microsserviços, como APIs RESTful. O FrontEnd do EnergySaver onde se pode realizar a criação de usuários, cadastro de sensores e apresentar os dados de consumo recebidos em tempo real, foi criado a partir deste framework.

Todos esses equipamentos e softwares formam a base de funcionamento do sistema EnergySaver para coleta e exibição dos dados de consumo recebidos em tempo real, conforme Figura 50 a seguir:

Figura 55: Gráfico de dados de consumo recebidos pelo EnergySaver em tempo real.



Fonte: Elaboração própria, 2022.

Observando a Figura 50, é importante destacar que no gráfico são exibidas as informações de consumo e os horários que esses dados foram recebidos, a fim de possibilitar ao usuário uma visão geral de como o equipamento monitorado está sendo utilizado. Caso o usuário queira fazer o download dos dados obtidos até o momento, basta clicar na opção “Download .csv” e baixa-los. Essa opção é muito importante pois a grande vantagem de um arquivo CSV (*comma-separated-values*) é a possibilidade de sua importação e exportação, em uma linguagem que vários aplicativos podem ler, tornando-se fundamental para transferência de informações entre diferentes aplicações.

Detalhes sobre a implementação do sistema EnergySaver estão disponíveis em Da Silva *et al.* (2021b), onde os autores descrevem todos os passos para a instalação, configuração e utilização. Esse direcionamento é importante, pois

apresenta informações relevantes para o usuário, grupo de trabalho ou empresa, que tenha o objetivo de implantar um sistema parecido.

De acordo com Fracari *et al.* (2015), as Informações obtidas pelo monitoramento de consumo acrescentam informações importantes que possibilitam ao consumidor gerenciar suas demandas de forma que gere menor custo na conta de energia. Por fim, é importante destacar que o monitoramento de consumo de energia elétrica facilita a gestão, e permite que sejam realizadas tomadas de decisão principalmente no que se refere à eficiência energética e à redução de custos.