

UNIVERSIDADE FEDERAL DE SÃO CARLOS– UFSCAR  
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA– CCET  
DEPARTAMENTO DE COMPUTAÇÃO– DC  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO– PPGCC

**Diego Luiz Cavalca**

**Extração de Características por  
Transferência de Aprendizado Profundo  
no Âmbito do Monitoramento Não  
Invasivo de Cargas Residenciais**

São Carlos  
2024



**Diego Luiz Cavalca**

**Extração de Características por  
Transferência de Aprendizado Profundo  
no Âmbito do Monitoramento Não  
Invasivo de Cargas Residenciais**

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências Exatas e de Tecnologia da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Área de concentração: Inteligência Artificial

Orientador: Ricardo Augusto Souza Fernandes

São Carlos

2024





# UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia  
Programa de Pós-Graduação em Ciência da Computação

---

## Folha de Aprovação

---

Defesa de Tese de Doutorado do candidato Diego Luiz Cavalca, realizada em 08/01/2024.

### Comissão Julgadora:

Prof. Dr. Ricardo Augusto Souza Fernandes (UFSCar)

Prof. Dr. Arlindo Neto Montagnoli (UFSCar)

Prof. Dr. Fábio Anderson Silva Borges (UESPI)

Prof. Dr. Vinicius Ponte Machado (UFPI)

Prof. Dr. Daniel Barbosa (UFBA)

O Relatório de Defesa assinado pelos membros da Comissão Julgadora encontra-se arquivado junto ao Programa de Pós-Graduação em Ciência da Computação.

*Dedico esta trabalho à minha amada família,  
cujo apoio incansável foi a luz que guiou minha jornada acadêmica.*

---

# Agradecimentos

---

Agradeço à minha amada família – pai, mãe e irmã - cujo apoio incansável nos estudos, somado aos esforços admiráveis diante dos desafios da vida, moldou o caminho que percorri até aqui.

Agradeço à minha esposa, Ana Carla, pelo amor, companheirismo e paciência em todos os momentos desta longa caminhada, por sonhar este sonho comigo, sendo minha inspiração em todas as conquistas.

À minha querida filha Lavínia, raio de luz e fonte de alegria. Sua chegada ao final desta jornada transformou cada desafio em inspiração. Agradeço por ser a razão mais valiosa das minhas conquistas.

Agradeço também a Kira e Zoe, minhas fiéis companheiras, por trazerem alegria e companheirismo constante, tornando cada desafio mais leve.

Agradeço de forma especial ao meu orientador, Professor Ricardo, que acreditou e me apoiou em todos os momentos, tanto no aspecto acadêmico quanto pessoal. Sua orientação sábia e incentivo foram fundamentais para o meu crescimento.

Agradeço à Deus, que pavimenta cada passo em minha vida e oportuniza todas as conquistas.

Em tempo, agradeço à UFSCar, ao Programa de Pós-Graduação em Ciência da Computação - PPGCC/UFSCar e à CAPES, pelos recursos fundamentais que proporcionaram para apoiar meus estudos.



---

# Resumo

---

O desenvolvimento de técnicas que permitem a eficiente identificação de cargas, idealmente de forma não invasiva, é um fator chave para a implementação prática de sistemas de monitoramento de energia residencial. Recentemente, o uso de técnicas baseadas em aprendizado profundo têm ganhado atenção em diferentes domínios, como processamento de sinais e imagens, com destaque aos modelos baseados em redes neurais convolucionais. No entanto, o treinamento eficiente desses modelos é fortemente dependente da quantidade e do balanceamento dos dados, ou seja, características que normalmente não são encontradas em conjuntos de dados de monitoramento não invasivo de cargas. Para lidar com esses desafios, esta tese propõe uma abordagem baseada em três etapas, a saber: (i) transformação de séries temporais em imagens 2D; (ii) extração de características usando transferência de aprendizado profundo; e (iii) classificação/rotulação de cargas residenciais. Nesse sentido, foram consideradas cinco cargas presentes no conjunto de dados *Reference Energy Disaggregation Dataset*. Os resultados apontam que a abordagem proposta foi capaz de obter um *f1-score* médio de 84,2%. Além disso, ainda foi possível observar uma maior capacidade da abordagem proposta em inferir e generalizar suas respostas em comparação a outras abordagens avaliadas, apresentando resultados consistentes mesmo diante do desbalanceamento dos dados.

**Palavras-chave:** aprendizado profundo, gráfico de recorrência, identificação de cargas residenciais, monitoramento não invasivo de cargas.



---

# Abstract

---

The development of techniques that allow the efficient identification of loads, ideally in a nonintrusive way, is a key factor for the practical implementation of home energy management systems. Recently, the use of techniques based on deep learning has gained attention in different domains, such as signal and image processing, highlighting the models based on convolutional neural networks. However, the efficient training of these models is strongly dependent on the amount and balance of data, i.e., characteristics that are not normally found in nonintrusive load monitoring datasets. To deal with these challenges, this thesis proposes an approach based on three stages, which are: (i) transformation of time series into 2D images; (ii) feature extraction using deep transfer learning; and (iii) classification/labeling of loads. In this sense, it was considered five loads present in the Reference Energy Disaggregation Dataset. The results indicate that the proposed approach was able to obtain an average f1-score of 84.2%. Moreover, from the analysis of the results, it was also possible to observe a greater capacity of the proposed approach to infer and generalize its responses compared to other evaluated approaches, presenting a consistent result even in the face of unbalanced data.

**Keywords:** deep learning, recurrence plot, residential load identification, nonintrusive load monitoring.



---

# Lista de ilustrações

---

Figura 1 – Exemplos de assinaturas de potência dos quatro tipos de cargas: (a) lâmpada de 20W; (b) lâmpada controlada; (c) máquina de lavar roupas; (d) refrigerador. . . . .	30
Figura 2 – Etapas do processo de NILM. . . . .	32
Figura 3 – Crescimento de publicações sobre aprendizado profundo, considerando artigos indexados na base de dados <i>Scopus</i> . . . . .	33
Figura 4 – Diagrama de arquitetura de uma MLP com duas camadas ocultas. . . .	35
Figura 5 – Diagrama da arquitetura LeNet. . . . .	38
Figura 6 – Diagrama da arquitetura VGG16. . . . .	39
Figura 7 – Bloco Residual da arquitetura ResNet. . . . .	40
Figura 8 – Exemplo de implementação do bloco <i>Inception</i> . . . . .	41
Figura 9 – Arquitetura <i>Inception V1</i> . A caixa laranja é o "tronco" da rede, que possui algumas convoluções preliminares; as caixas roxas são classificadores auxiliares. As demais estruturas são os blocos <i>Inception</i> . . . . .	42
Figura 10 – <i>MobileNet</i> : mecanismo <i>depthwise separable convolutions</i> . . . . .	43
Figura 11 – Visão geral da arquitetura EfficientNet. . . . .	44
Figura 12 – Diferentes abordagens de <i>transfer learning</i> : (a) arquitetura do modelo original; (b) estratégia de ajuste fino; e (c) estratégia de extrator de características. . . . .	45
Figura 13 – Exemplo de séries temporais e seus respectivos RPs: (1) RP homogêneo; (2) RP gerado por sistemas periódicos; (3) RP de sistemas com variação no(s) parâmetro(s); e (4) RP de uma série temporal altamente não estacionária. . . . .	48
Figura 14 – Representação do modelo FHMM implementado para NILM. . . . .	54
Figura 15 – Arquitetura DAE projetada para NILM. . . . .	59
Figura 16 – Diagramas de funcionamento dos modelos seq2seq (à esquerda) e seq2point (à direita). . . . .	60

Figura 17 – Visão Geral da Metodologia Proposta. . . . .	68
Figura 18 – Diagrama da estrutura fundamental de uma árvore de decisão. . . . .	71
Figura 19 – Diagrama da análise comparativa conduzida neste estudo. . . . .	75
Figura 21 – Distribuição de probabilidade para Acurácia. . . . .	79
Figura 22 – Distribuição de probabilidade para <i>f1-score</i> . . . . .	79
Figura 23 – Desempenho dos Diferentes Extratores em Combinação com Classificadores Variados. . . . .	85
Figura 20 – Desempenho da abordagem DTLFE + MLP para cada carga, considerando diferentes janelas. . . . .	90

---

# Lista de tabelas

---

Tabela 1 – Conjuntos de dados disponíveis para pesquisas sobre NILM. . . . .	52
Tabela 2 – Tamanho da janela deslizante definida inicialmente para melhor caracterização do ciclo de operação das cargas ( <i>taxa amostral = 2 segundos</i> ). 67	
Tabela 3 – Comparação de estudos que utilizam o conjunto de dados REDD para o processo de identificação de cargas no contexto de NILM. . . . .	74
Tabela 4 – Desempenho geral dos classificadores avaliados. . . . .	75
Tabela 5 – Desempenho do DTLFE para a classificação individual de cargas. . . .	76
Tabela 6 – Desempenho do RR + DET para a classificação individual de cargas. .	76
Tabela 7 – Desempenho da abordagem <i>end-to-end</i> ( <i>features</i> = RP e classificador = CNN ) para a classificação individual de cargas. . . . .	77
Tabela 8 – Desempenho do framework DTLFE + MLP para cada carga. . . . .	77
Tabela 9 – Comparando o <i>f1-score</i> diante da janela original e otimizada para a abordagem DTLFE + MLP ( <i>taxa amostral = 2 segundos</i> ). . . . .	78
Tabela 10 – Resultados da Avaliação de Generalização em Diferentes Residências. .	81
Tabela 11 – Desempenho do DTLFE + MLP Treinado na Residência #3 à nível de Aparelho nas Demais Residências. . . . .	82
Tabela 12 – Arquiteturas pré-treinadas (ImageNet) Avaliadas no Processo de Extração de Características . . . . .	84
Tabela 13 – Impacto do Ajuste-fino das Arquiteturas Pré-treinadas no Processo de Classificação . . . . .	88



---

# Lista de siglas

---

**AFHMM** Additive Factorial Hidden Markov Model

**ALIP** Aided Linear Integer Programming

**BLUED** Building-Level fully labeled Electricity Disaggregation

**CNN** Convolutional Neural Network

**DAE** Denoise Autoencoder

**DET** Determinismo

**DL** Deep Learning

**DTLFE** Deep Transfer Learning Feature Extraction

**DNN** Deep Neural Network

**EUA** Estados Unidos da América

**FHMM** Factorial Hidden Markov Model

**FN** False Negative

**FP** False Positive

**FPGA** Field Programmable Gate Array

**GADF** Gramian Angular Difference Fields

**GPU** Graphic Processing Unit

**GRU** Gated Recurrent Unit

**HMM** Hidden Markov Model

**IP** Integer Programming

**LSTM** Long-Short Term Memory

**MCE** Modified Cross-Entropy

**MEF** Máquinas de Estado Finito

**MILP** Mixed-Integer Linear Programming

**ML** Machine Learning

**MLP** Multilayer Perceptron

**MSE** Mean Squared Error

**NILM** Non-intrusive Load Monitoring

**PCA** Principal Component Analysis

**PLAID** Plug Load Appliance Identification Dataset

**PNN** Probabilistic Neural Network

**RBF** Radial Basis Function

**REDD** Residential Energy Disaggregation Dataset

**RMSE** Root Mean Squared Error

**RNA** Rede Neural Artificial

**RNN** Recurrent Neural Network

**RP** Recurrence Plot

**RQA** Recurrent Quantification Analysis

**RR** Recurrence Rate

**SEP** Sistema Elétrico de Potência

**seq2point** Sequence to Point

**seq2seq** Sequence to Sequence

**SGD** Stochastic Gradient Descent

**SSHMM** Superstate Hidden Markov Model

**SVM** Support Vector Machine

**TN** True Negative

**TP** True Positive

**UK-DALE** United Kingdom Domestic Appliance Level Electricity

**WHITED** Worldwide Household and Industry Transient Energy Dataset

**WRG** Weighted Recurrent Graph

**XGBoost** Extreme Gradient Boosting

---

# Sumário

---

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>21</b>
1.1	Motivações e Justificativas . . . . .	24
1.2	Objetivos . . . . .	24
1.3	Lista de Publicações . . . . .	25
1.4	Organização do Trabalho . . . . .	26
<b>2</b>	<b>CONCEITOS FUNDAMENTAIS . . . . .</b>	<b>27</b>
2.1	Monitoramento Não Invasivo de Cargas . . . . .	27
2.1.1	Tipos de Cargas . . . . .	29
2.1.2	Etapas do Processo de NILM . . . . .	30
2.2	Aprendizado Profundo . . . . .	32
2.2.1	Rede Neural Artificial . . . . .	32
2.2.2	Rede Neural Convolutacional . . . . .	36
2.2.3	Principais Arquiteturas Convolucionais . . . . .	38
2.2.4	Transferência de Aprendizado . . . . .	44
2.3	Transformação de Séries Temporais em Imagens . . . . .	46
<b>3</b>	<b>REVISÃO BIBLIOGRÁFICA . . . . .</b>	<b>51</b>
3.1	Monitoramento não invasivo de cargas . . . . .	51
3.1.1	Conjuntos de Dados . . . . .	51
3.1.2	Abordagens Tradicionais . . . . .	53
3.1.3	Abordagens Baseadas em <i>Deep Learning</i> . . . . .	57
3.2	Lacunas de Pesquisa . . . . .	63
<b>4</b>	<b>METODOLOGIA PROPOSTA . . . . .</b>	<b>65</b>
4.1	Preparação dos Dados . . . . .	65
4.2	Ambiente Computacional . . . . .	67

<b>4.3</b>	<b>Framework Proposto</b> . . . . .	<b>67</b>
4.3.1	Módulo de Transformação da Série Temporal em Imagem . . . . .	68
4.3.2	Módulo de Extração de Características . . . . .	68
4.3.3	Módulo de Classificação . . . . .	69
<b>4.4</b>	<b>Métricas de Avaliação de Desempenho</b> . . . . .	<b>71</b>
<b>5</b>	<b>RESULTADOS</b> . . . . .	<b>73</b>
<b>5.1</b>	<b>Classificação de Cargas</b> . . . . .	<b>74</b>
5.1.1	Análise Geral de Desempenho . . . . .	75
5.1.2	Ajuste Fino das Janelas Deslizantes para o DTLFE . . . . .	77
<b>5.2</b>	<b>Análise de Generalização</b> . . . . .	<b>78</b>
5.2.1	Validação Cruzada . . . . .	78
5.2.2	Capacidade de Generalização Inter-Residências . . . . .	80
<b>5.3</b>	<b>Diferentes Arquiteturas Profundas para Extração de Características</b> . . . . .	<b>83</b>
5.3.1	Experimentação . . . . .	84
5.3.2	Análise Geral . . . . .	84
<b>5.4</b>	<b>Ajuste-fino da Arquitetura de Extração de Características</b> . . . . .	<b>86</b>
5.4.1	Impacto da Inicialização de Pesos . . . . .	88
5.4.2	Efeito da Estratégia de Amostragem de Dados para o Ajuste Fino . . . . .	89
5.4.3	Generalização e Complexidade do Modelo . . . . .	89
5.4.4	Considerações sobre Classificadores . . . . .	89
5.4.5	Observações Importantes . . . . .	89
<b>6</b>	<b>CONCLUSÕES E DIRETIVAS FUTURAS</b> . . . . .	<b>91</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>93</b>

---

# Capítulo 1

## Introdução

---

Em Sistemas Elétricos de Potência (SEP), diversas aplicações lidam com a análise de séries temporais, sendo possível destacar: (i) a detecção e classificação de distúrbios de qualidade da energia elétrica (LIU et al., 2018; THIRUMALA et al., 2019); (ii) a detecção e classificação de faltas (ABDALI; MAZLUMI; NOROOZIAN, 2019; ASGHARIGOVAR; POURGHASEM; SEYEDI, 2019); (iii) a detecção de condições de ilhamento para geradores distribuídos (NARAYANAN; SIDDIQUI; FOZDAR, 2017; KONG et al., 2018); e (iv) a identificação de cargas (HART, 1992; SRINIVASAN; NG; LIEW, 2006; BERGES et al., 2010; DONG et al., 2012; FERNANDES; SILVA; OLESKOVICZ, 2013; AIAD; LEE, 2016; SADEGHIANPOURHAMAMI et al., 2017; ZHAO et al., 2018; AFZALAN; JAZIZADEH; WANG, 2019; RASHID et al., 2019). Em geral, as tarefas de análise de séries temporais são divididas em categorias, sendo estas a aproximação de funções, regressão, classificação e agrupamento (HATAMI; GAVET; DEBAYLE, 2018).

No que tange especificamente a classificação de séries temporais, uma classificação univariada visa estabelecer a relação  $x^n \rightarrow y^n$ , de modo que uma amostra  $n$  (série temporal) de comprimento  $L : x^n = (x_1^n, x_2^n, \dots, x_L^n)$  está associada a uma classe  $y^n$ , tal que  $y^n \in \{1, 2, \dots, c\}$ . Neste contexto, conforme Fakhrazari e Vakilzadian (2017), os métodos existentes podem ser categorizados sob diferentes perspectivas:

- tipos de variáveis – métodos que atuam no domínio da frequência incluem análise espectral, enquanto métodos que atuam sob o domínio do tempo incluem autocorrelação, autorregressão e análise de correlação cruzada;
- processo de classificação – os métodos podem ser divididos em estratégias baseadas em instâncias (visam atribuir a  $x^n$  um rótulo a partir de uma medida de similaridade, como por exemplo a distância euclidiana) e estratégias baseadas em características

ou atributos (representam a série temporal em um novo espaço/contexto e extraem as características/atributos mais relevantes para definir fronteiras de decisão).

Todavia, a identificação de padrões não lineares é um grande desafio encontrado na área de séries temporais e está presente em sistemas dinâmicos e/ou processos estocásticos. Estes fenômenos são recorrentes, de modo que certas regiões do espaço de estados são frequentemente visitadas (GRABEN et al., 2016; PIOREK, 2019). Apesar da relevância de comportamentos recorrentes, a visualização desses comportamentos geralmente é complexa no domínio do tempo.

Buscando superar tais limitações, Eckmann, Kamphorst e Ruelle (1987) criaram uma técnica denominada de Gráfico de Recorrência (RP, do inglês *Recurrence Plot*). De maneira geral, o RP oferece uma representação visual única das recorrências em uma série temporal, revelando regularidades e comportamentos recorrentes que muitas vezes escapam à detecção no domínio do tempo. Ao transformar a série temporal em um espaço geométrico de recorrência, o RP destaca regiões do espaço de estados que são frequentemente revisitadas, permitindo uma compreensão mais profunda da dinâmica subjacente do sistema, emergindo como uma abordagem particularmente valiosa em contextos nos quais padrões complexos e não lineares estão intrínsecos no comportamento da série ao longo do tempo.

Diante do contexto descrito, fica evidente que a adoção de RPs extrapola o clássico processamento de sinais temporais discretos e abre caminho para o estudo de abordagens da área de processamento de imagens, fomentando um cenário próspero para a realização da tarefa de classificação de séries temporais por meio de sua combinação com técnicas de aprendizado profundo (DL, do inglês *Deep Learning*), conforme demonstrado em Yang et al. (2018). Tal estratégia corrobora com Fakhrazari e Vakilzadian (2017), os quais afirmam que, uma vez que o estudo de séries temporais é um campo conhecidamente complexo e permeia muitos domínios do conhecimento, a derivação de abordagens híbridas, as quais unem diferentes técnicas e metodologias, pode possibilitar o desenvolvimento de ferramentas de mineração de dados de séries temporais mais eficientes e precisas. Sob a perspectiva de sinal, é possível realizar regressões e prever comportamentos a partir da análise de fluxo de dados. Porém, no que tange a perspectiva de imagem, singularidades podem ser suavizadas e agrupamentos feitos, revelando padrões significativos e que *a priori* são ocultos. Dessa forma, abre-se caminho para a investigação de métodos avançados de processamento de imagens e classificação de séries temporais representadas por RPs.

Com o aprimoramento de hardware e software, os modelos de DL, mais especificamente os baseados em Redes Neurais Artificiais (RNA), alcançaram notável desempenho em problemas de reconhecimento de padrões. Dentre as diversas arquiteturas possíveis, as Redes Neurais Convolucionais (CNN, do inglês *Convolutional Neural Networks*) se tornaram uma das mais eficientes para tarefas de processamento de imagens e atingiram o estado da arte em problemas de reconhecimento de padrões.

Uma das primeiras aplicações de sucesso de CNNs na área de processamento de imagens foi desenvolvida por Lecun et al. (1998), onde os autores propuseram o modelo LeNet-5. No entanto, para a aplicação de CNN em problemas de classificação de séries temporais, basicamente são utilizadas duas estratégias distintas: (i) modificar a arquitetura tradicional da CNN e usar séries temporais 1D como entrada; ou (ii) transformar os sinais 1D em matrizes 2D e então aplicar a CNN, de maneira similar à aplicação tradicional deste modelo para a classificação de imagens.

A reconhecida habilidade das CNNs para capturar e hierarquizar informações complexas em dados visuais tem suscitado um crescente interesse em explorar esses modelos como ferramentas versáteis em diversos contextos. Para além de sua função convencional em tarefas de classificação, as CNNs têm sido eficazmente aplicadas na extração de características em variados domínios, como reconhecimento de objetos, processamento de imagens médicas e análise de texturas.

Um avanço notável nesse campo é a capacidade de reaproveitar diferentes arquiteturas de CNNs pré-treinadas em conjuntos massivos de imagens, frequentemente destinadas a tarefas de reconhecimento visual. Arquiteturas profundas treinadas em grandes conjuntos de dados públicos têm evidenciado uma extraordinária capacidade de generalização. Dentro desse contexto, a transferência de aprendizado, ou *transfer learning*, possibilita que as CNNs utilizem as representações previamente aprendidas em um domínio para aprimorar o desempenho em outro. Essa abordagem não apenas otimiza recursos computacionais e dados de treinamento, mas também impulsiona a eficácia do modelo em tarefas correlatas. Desse modo, a utilização das CNNs como extratores de características, especialmente quando alimentadas por arquiteturas pré-treinadas, emerge como uma estratégia promissora para enriquecer a representação de dados e, potencialmente, aprimorar o desempenho em uma diversidade de contextos aplicados.

Nesse sentido, a utilização de modelos pré-treinados para extrair automaticamente características dos RPs visa não apenas uma eficiente extração de características, mas também uma análise mais profunda das recorrências e padrões inerentes às séries temporais que se referem ao consumo de energia elétrica residencial. Dessa forma, essa conjunção entre modelos pré-treinados de DL e RP delineia um caminho inovador e promissor para aprimorar a identificação de cargas residenciais, enfatizando a sinergia entre diferentes abordagens no contexto do monitoramento não invasivo de cargas (NILM, do inglês *Nonintrusive Load Monitoring*).

Seguindo o contexto supracitado, diante aos notáveis resultados das CNNs no reconhecimento de padrões e a capacidade de RPs em caracterizarem a natureza periódica de séries temporais e suas recorrências, esta tese visa investigar a combinação destas duas abordagens para a resolução de problemas de classificação de séries temporais dentro de um *framework* único que envolve todo o *pipeline* de preparação de dados e classificação. Portanto, busca-se a implementação de uma metodologia que usufrua das vantagens de

RP e DL, especialmente através de arquiteturas pré-treinadas, na classificação de séries temporais. Entretanto, a metodologia proposta é avaliada no contexto NILM, sendo essa uma área em que, apesar de haver abordagens baseadas em DL, ainda não pode ser encontrado na literatura correlata uma investigação aprofundada acerca de uma metodologia híbrida contemplando a preparação de dados e avaliação de transferência de aprendizado como a proposta nesta tese.

## 1.1 Motivações e Justificativas

Conforme previamente apresentado, a classificação de séries temporais é um importante campo de pesquisa, uma vez que vários problemas relevantes podem ser compreendidos neste segmento, tais como:

- ❑ Estimativa de variação nos preços de ações em bolsas de valores;
- ❑ Previsão de vendas ou demandas de insumos;
- ❑ Avaliação de condições climáticas em determinados períodos de tempo;
- ❑ Diagnóstico de doenças por meio de exames de imagens ou outros tipos de sinais;
- ❑ Identificação de cargas por meio de NILM.

Um dos agentes motivadores dessa tese é a necessidade de investigação e desenvolvimento de métodos para a extração de conhecimento dos dados dependentes do tempo, os quais necessitam de tratamento especial e assertivo para a resolução eficiente de tarefas pertinentes à classificação de séries temporais. Além disso, problemas que envolvem séries temporais caracterizam desafios de grande relevância prática, uma vez que impactam diretamente o cotidiano da sociedade. Assim, nota-se a importância de realizar a análise de desempenho de RPs e DL quando aplicados nesta classe de problemas.

Essa tese também se motiva pela relevância do tópico diante da crescente demanda por energia elétrica e dos desafios intrínsecos à eficiência energética de consumidores residenciais.

## 1.2 Objetivos

Como objetivos desta tese, pode-se destacar a necessidade de averiguar o potencial do *framework* proposto ao monitoramento não invasivo de cargas residenciais e baseado em DL e RPs, considerando o conjunto de dados REDD (*Residential Energy Disaggregation Dataset*), uma vez que este é um dos mais utilizados como forma de se realizar um *benchmark* entre metodologias no contexto de NILM. Além disso, busca-se validar tal *framework* como uma alternativa eficiente de classificação de séries temporais em geral,

viabilizando futuras investigações e aplicações em outros domínios, como dados provenientes de mercados de energia (preço e demanda) e de medições de qualidade da energia elétrica (visando a classificação de distúrbios). Além disso, vale destacar que embora esta pesquisa tenha foco no problema de classificação, tal abordagem pode ser adaptada para outras tarefas, como agrupamento e regressão de séries temporais.

Em consonância com as justificativas e objetivos previamente delineados, esta pesquisa busca contribuir com o avanço da literatura:

1. na extração de atributos (também denominadas *features*, ou características) no decorrer desta tese, os quais são provenientes dos RPs, bem como os próprios gráficos, visto que esta representa uma contribuição que até o momento não foi avaliada de maneira abrangente no âmbito de NILM;
2. em validar a preparação de dados temporais para o problema abordado, analisando o impacto de ajustar individualmente por carga o tamanho de janela implementado no processo classificatório;
3. no uso de redes neurais profundas em conjunto com RPs para melhorar o desempenho na tarefa de classificação/rotulação das cargas, visto que a metodologia proposta ainda será comparada a outras já consolidadas por meio da literatura;
4. corroborando a transferência de aprendizado como um método robusto para a classificação de cargas em NILM diante de um cenário restrito de disponibilidade de dados; e
5. disponibilizar um *framework* Python, o PyNILM<sup>1</sup>, que visa estabelecer um ambiente estruturado e amigável para aplicações de aprendizado de máquina no contexto de NILM, facilitando a experimentação e a colaboração entre pesquisadores neste campo de pesquisa.

## 1.3 Lista de Publicações

Em tempo, é válido ressaltar que, durante o Doutorado, foram produzidos os seguintes artigos:

### Periódico:

- CAVALCA, D.L.; FERNANDES, R.A.S. Deep Transfer Learning-Based Feature Extraction: An Approach to Improve Nonintrusive Load Monitoring. **IEEE Access**, 2021.

### Publicação em Congresso Internacional:

<sup>1</sup> Disponível em <https://github.com/diegocavalca/pynilm>

- CAVALCA, D.L.; FERNANDES, R.A.S. Recurrence Plots and Convolutional Neural Networks Applied to Nonintrusive Load Monitoring. **IEEE PES General Meeting, 2020. Best Conference Papers.**
- CORRÊA, J.; CAVALCA, D.L.; FERNANDES, R.A.S. Gramian Angular Field and Recurrence Plots as Feature Engineering Techniques on Residential Appliances Labeling: A Comparative Analysis. **IEEE PES Innovative Smart Grid Technologies Latin America, 2023.**

#### Publicação em Congresso Nacional:

- FONSECA, A.L.A.; CAVALCA, D.L.; PEREIRA JUNIOR, B.R.; FERNANDES, R.A.S. Escalonamento otimizado para consumo de eletricidade residencial inteligente e sustentável no cenário de tarifa horária branca com imposições mínimas de conforto térmico. **XII Congresso Brasileiro de Planejamento Energético, 2020.**

## 1.4 Organização do Trabalho

Além do capítulo introdutório em questão, essa tese está estruturada nos seguintes capítulos:

- ❑ Capítulo 2 — disserta sobre os conceitos fundamentais compreendidos no escopo da pesquisa, tais como NILM, DL, *Transfer Learning* e RP;
- ❑ Capítulo 3 — apresenta a revisão bibliográfica desta pesquisa, contemplando desde o panorama e detalhamento histórico sobre o monitoramento não invasivo de cargas e seus métodos, até os recentes desenvolvimentos relacionados a redes neurais convolucionais e suas aplicações, combinadas ou não com RPs, no contexto de NILM;
- ❑ Capítulo 4 — versa sobre a metodologia proposta para o *framework*, visando a classificação de séries temporais por meio de RP e DL, em que são descritos os experimentos computacionais e os dados utilizados para a avaliação das hipóteses discutidas;
- ❑ Capítulo 5 — são apresentados e discutidos os resultados dos experimentos computacionais, comparando o algoritmo proposto com os demais considerados no contexto desta tese;
- ❑ Capítulo 6 — destina-se às conclusões e trabalhos futuros que podem ser advindos da tese.

---

## Capítulo 2

# Conceitos Fundamentais

---

Conforme mencionado, neste capítulo são apresentados os conceitos fundamentais que estão compreendidos nesta tese, onde é descrito o problema de pesquisa, ou seja, a identificação de cargas residenciais no contexto de NILM. Ademais, são apresentados fundamentos de redes neurais artificiais, elucidando o contexto histórico, atuais avanços em relação às arquiteturas e algoritmos de treinamento de modelos convolucionais e *transfer learning*, além do método de transformação de série temporal em imagem, especificamente o *Recurrence Plot*, visto que irão compor o *framework* proposto.

### 2.1 Monitoramento Não Invasivo de Cargas

Diante da crescente demanda energética, em detrimento dos avanços tecnológicos significativos, com cada vez mais equipamentos elétricos/eletrônicos presentes nas residências, o valor da energia elétrica — tanto do ponto de vista econômico quanto ambiental — continua a aumentar, cenário este que incrementa em grande escala a necessidade de compreender as características de consumo de energia elétrica do usuário final. Especificamente, para a definição de estratégias eficientes de gestão energética de ponta a ponta, é de grande valia saber como os equipamentos elétricos são usados e a energia consumida por eles. Isso pode ser benéfico para o consumidor, pois esses dados podem ser usados para responder algumas questões, como: “*Qual equipamento tem maior impacto na conta de energia mensal?*”; “*Existem alternativas mais eficientes para este equipamento?*”; “*Os hábitos de uso de eletricidade podem ser alterados para reduzir a conta mensal?*”. Ainda, pelo lado das concessionárias, estas poderão planejar de modo mais eficiente a operação do sistema de distribuição, visando evitar possíveis cenários de instabilidade, bem como oferecer preços flexíveis de acordo com a demanda para que seja possível otimizar os custos

gerais e, conseqüentemente, manter a satisfação do consumidor (VARDAKAS; ZORBA; VERIKOUKIS, 2015).

Todavia, o acesso aos dados de consumo instantâneo de energia dos equipamentos é um desafio complexo, uma vez que, em geral, estes não foram projetados para possuir tal característica. Portanto, essa informação não pode ser obtida sem a substituição destes equipamentos e/ou suas tomadas por versões ditas inteligentes, denotando um cenário invasivo e pouco escalável. O monitoramento não invasivo de cargas, NILM, proposto pela primeira vez por Hart (1992), é uma abordagem oposta que visa permitir aos usuários finais e concessionárias de energia um acesso simples, barato e menos obstrutivo a esses dados, factível para cenários industriais, comerciais e residenciais.

Formalmente, NILM, também chamado de desagregação de energia na literatura correlata, pode ser denotado de forma que, ao considerar um consumidor com  $i = 1, 2, \dots, n$  equipamentos conectados à rede elétrica, cada equipamento  $i$  possui um estado binário  $S_t^{(i)}$  (ligado/desligado) associado que indica seus respectivos estados de operação, além de um consumo médio de energia  $P_t^{(i)}$ . Assim, o consumo total de energia  $P(t)$  em cada instante de tempo  $t$  é dado por:

$$P_{agr}(t) = \sum_{i=1}^n (S_t^{(i)} \times P_t^{(i)}). \quad (1)$$

Vale destacar que a resolução de um problema de NILM é uma tarefa complexa, uma vez que o consumo energético agregado compreende diversas nuances operacionais, como o fato de que cargas maiores tendem a ocultar o consumo de energia das cargas menores. Ademais, quaisquer flutuações — ainda que mínimas — no consumo agregado de energia  $P_{agr}(t)$  tende a resultar em um vetor de estado  $S_t^*$  completamente distinto. Deste modo, para conseguir resolver eficientemente este problema, deve-se obter uma visão estatística significativa tanto dos equipamentos quanto do consumo agregado. Ao tentar obter tal entendimento, alguns desafios podem ser destacados no contexto residencial, tais como:

- **Variedade de equipamentos** — cada residência usa um conjunto diferente de equipamentos, os quais são geralmente de um fabricante e/ou modelo distinto. Por exemplo, algumas residências podem ter sistema de aquecimento elétrico, enquanto outras podem usar aquecimento a gás ou nenhum aquecimento; ainda pode-se mencionar o uso de televisores com tecnologias distintas, as quais diferem em consumo de energia, modelos e fabricantes;
- **Coleta de dados** — a fim de coletar dados suficientes de cenários do mundo real, ou seja, instalar uma grande quantidade de sensores em várias residências. Uma vez que este processo é caro e invasivo, os dados têm sido coletados principalmente por

grupos de pesquisa e as características de cada conjunto de dados variam, tornando difícil combinar dados de conjuntos diferentes;

- ❑ **Cálculos em tempo real** — para que uma solução de NILM se consolide como uma ferramenta viável, é importante que tenha um custo aceitável, o que leva à necessidade de sustentar sua operação em tempo real em plataformas de hardware simples e amplamente disponíveis;
- ❑ **Generalização** — soluções NILM treinadas ou projetadas usando quantidades finitas de dados, devem ser capazes de generalizar para outros cenários. Isso pode ser alcançado no escopo da solução implementada ou através de algum método de aprendizagem *online*. Como resultado da coleta de dados e a grande variedade de equipamentos, a generalização continua sendo um dos aspectos de mais difícil resolução em NILM.

Uma das maneiras de superar alguns desses desafios é por meio da compreensão dos diferentes tipos de equipamentos existentes, pois as diferenças em seus modos de operação afetam diretamente a modelagem deles.

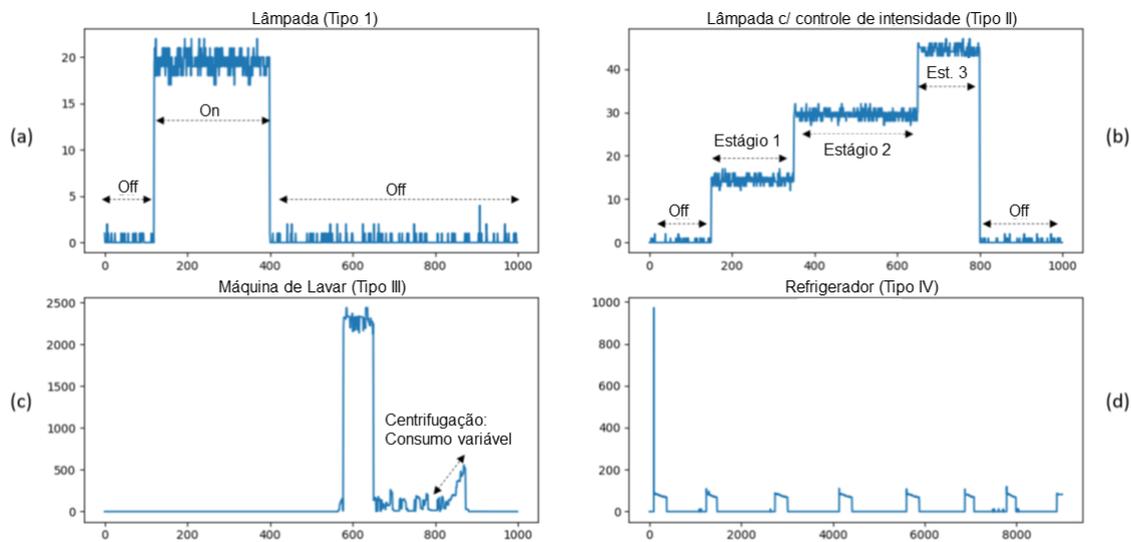
### 2.1.1 Tipos de Cargas

Um dos maiores desafios em NILM é a grande variedade de equipamentos elétricos/eletrônicos disponíveis no mercado. Uma das maneiras de mitigar essa dificuldade é por meio do agrupamento de equipamentos pelas características de sua assinatura de potência. Nesse sentido, Hart (1992) e Kim, Le e Kim (2017) observaram que os equipamentos podem ser divididos em quatro tipos, conforme os exemplos demonstrados na Figura 1, sendo eles:

- ❑ **TIPO I – Ligado/Desligado:** cargas que operam em estado binário (ON/OFF), de acordo com as necessidades do usuário (por exemplo, lâmpadas, torradeiras, etc.);
- ❑ **TIPO II – Máquinas de Estado Finito (MEF):** contempla cargas que têm um conjunto finito e discreto de estados operacionais. A transição entre esses estados pode ser controlada pelo usuário – como um secador de cabelo com várias configurações de aquecimento e ventilação – ou automática, como em máquina de lavar que possui ciclos de operação pré-definidos;
- ❑ **TIPO III – Variável continuamente:** essas cargas têm um componente cujo consumo de energia pode mudar em uma escala contínua, ao invés de pular entre estados discretos. Em alguns casos, essas cargas também são membros dos dois grupos anteriores. Por exemplo, uma lâmpada controlada por *dimmer*, terá uma potência variável, pois o usuário pode mudar sua intensidade, mas geralmente ainda será ligado e desligado pelo usuário. Em outros casos, a carga variável pode estar presente em um dos estados do Tipo II;

- **TIPO IV – Acionado continuamente/Cíclico:** por fim, cargas que têm uma natureza periódica e permanecem ativadas extensivamente ou mesmo permanentemente, alternando entre seus estados internos de operação. Por exemplo, uma geladeira geralmente estará sempre ligada e alternará entre o resfriamento e manutenção de temperatura. Aqui também pode haver alguma sobreposição com outros tipos de cargas.

Figura 1 – Exemplos de assinaturas de potência dos quatro tipos de cargas: (a) lâmpada de 20W; (b) lâmpada controlada; (c) máquina de lavar roupas; (d) refrigerador.



Fonte: Zoha et al. (2012).

### 2.1.2 Etapas do Processo de NILM

Em linhas gerais, conforme mostrado na Figura 2, um *framework* para NILM é estabelecido em quatro importantes etapas (Zoha et al., 2012), sendo elas:

- **Aquisição de dados** – parte do princípio que somente é possível realizar a desagregação através da assinatura das cargas que estejam compreendidas dentro do consumo agregado. A etapa inicial concentra todo o fluxo de coleta, organização e consolidação dos dados agregados de energia e de cada equipamento, tendo a preocupação de considerar todas as configurações necessárias, bem como a definição de taxa de amostragem adequada e sincronização temporal dos dados. Tais definições são fundamentais para se desagregar o consumo individual das cargas e viabilizar assim a identificação de seus padrões. Vale ressaltar que a desagregação das cargas é dependente de suas assinaturas individuais, as quais são posteriormente caracterizadas de acordo com seu estado operacional. No que tange o processo de coleta, a

taxa de amostragem determina o tipo de informação que pode ser extraída dos sinais elétricos e, neste sentido, conforme Ruano et al. (2019), existem duas estratégias de medição:

- *Baixa frequência* compreende sinais com taxa menor ou igual a 1 Hz, disponibilizada pela maioria dos medidores residenciais comercializados; e
- *Alta frequência* compreende sinais com taxas elevadas que permitem analisar eventos transitórios. No entanto, é realizada por medidores que possuem maior custo, o que inviabiliza sua adoção para uso geral;

□ **Detecção de Eventos** – a próxima etapa após a aquisição de dados é realizar o processamento dos dados brutos. Ainda nesta etapa, é feita a análise de eventos presentes na assinatura de potência, como a detecção de transição de estado do dispositivo (por exemplo, ligado para desligado), a partir das medições de energia. Ao final, munido das assinaturas de potência e a identificação de eventos, são implementadas técnicas de extração de informações que permitam identificar as cargas individualmente;

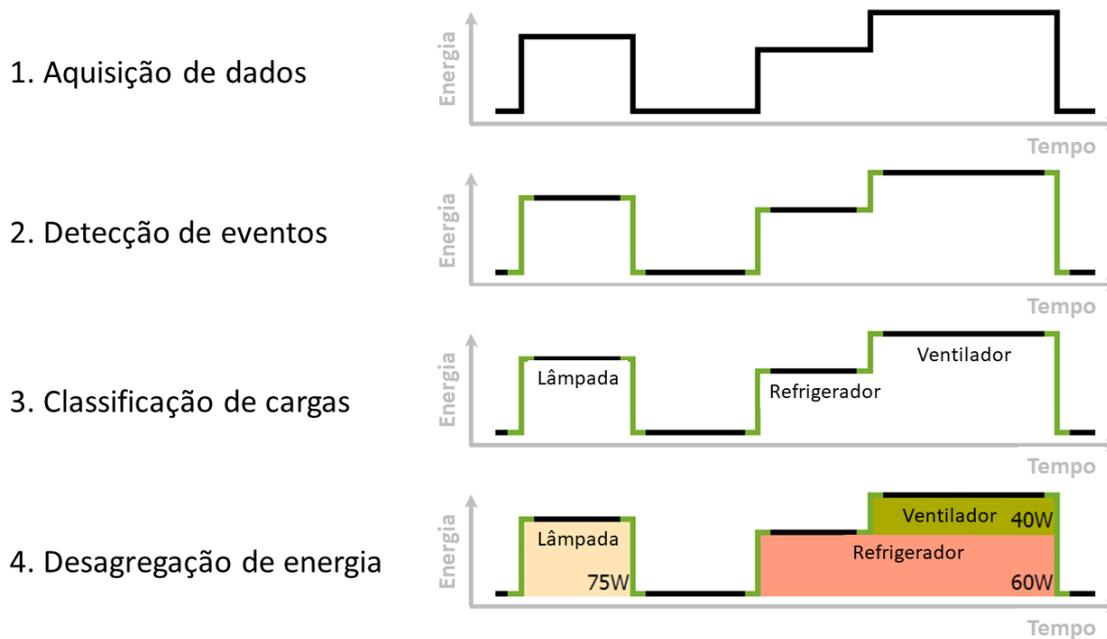
□ **Classificação de Cargas** – nesta etapa, algoritmos são preparados para realizar a classificação automática de estados específicos da carga a partir da medição agregada. Em grande parte da literatura, podem ser notadas abordagens de NILM que compreendem algoritmos baseados em técnicas de aprendizado de máquina supervisionado e, portanto, requerem dados rotulados para o treinamento do modelo de classificação;

□ **Desagregação de Consumo** – por fim, são executados algoritmos capazes de estimar o consumo individual de energia para cada uma das cargas identificadas a partir da medição agregada.

Cada uma destas etapas é de fundamental importância no processo de NILM e são alvos de contínua pesquisa, as quais originam novas abordagens e estudos (HASSAN; JAVED; ARSHAD, 2014). Portanto, as pesquisas sobre NILM têm como objetivo principal o desenvolvimento de métodos que possibilitem monitorar cargas e manter elevada acurácia. Entretanto, esse é um desafio não trivial, como já discutido, dada a inerente sobreposição de assinaturas de potência no consumo agregado, além das diferenças intrínsecas de uso e operação das cargas.

Neste sentido, esta tese visa apresentar uma abordagem alternativa para a classificação de cargas no contexto de NILM, propondo um *framework* baseado em três estágios, sendo: (i) transformação das séries temporais de consumo de energia em imagens bidimensionais; (ii) extração automática de características usando *deep transfer learning*; e (iii) classificação/rotulação de cargas.

Figura 2 – Etapas do processo de NILM.



Fonte: Adaptado de Zoha et al. (2012).

## 2.2 Aprendizado Profundo

Atualmente, não há uma definição única de aprendizado profundo. Em vez disso, muitas definições são propostas, mas todas compartilham ideias subjacentes semelhantes (ZHANG et al., 2018b). Em geral, todas as definições concordam que o aprendizado profundo envolve a concepção de um modelo de larga escala (em função do número de parâmetros) para a extração de uma estrutura hierárquica de atributos, diretamente dos dados. O campo de pesquisa em DL vem ganhando cada vez mais atenção nos últimos anos, conforme é possível observar na Figura 3, que consolida o número de publicações indexadas na base Scopus<sup>1</sup> sobre o tema a partir de 2010.

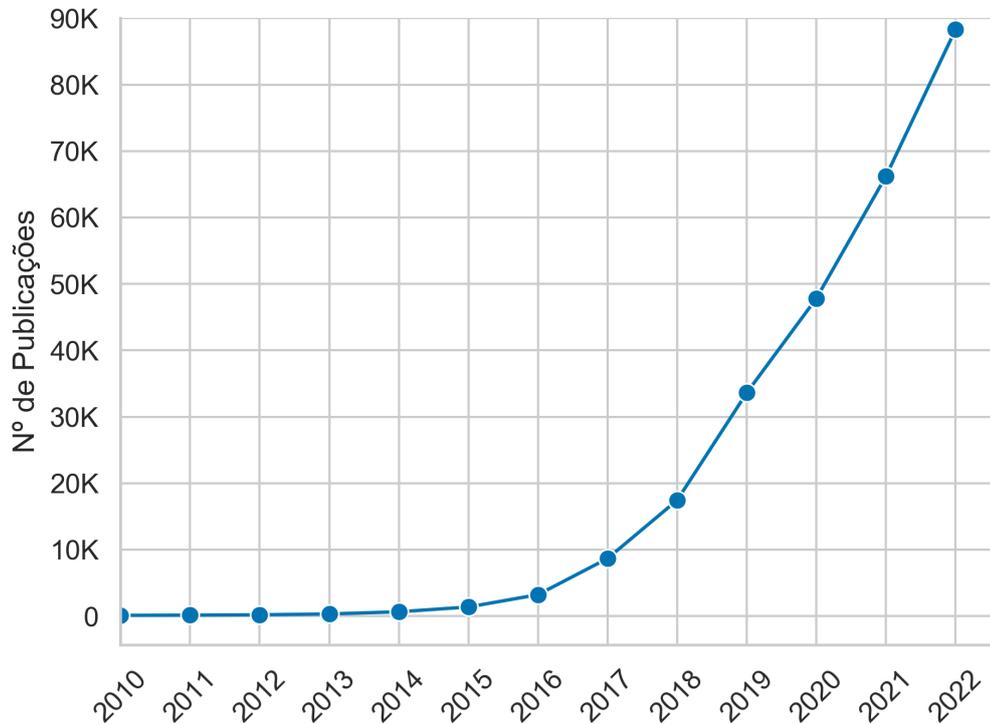
Devido à sua popularidade e ampla gama de definições, DL abrange uma grande variedade de algoritmos de aprendizado de máquina diferentes. No entanto, em sua maioria, o uso comum refere-se a algoritmos para projeto, treinamento e implantação de modelos conexionistas, mais precisamente baseados em redes neurais artificiais, conforme mostrado na sequência.

### 2.2.1 Rede Neural Artificial

Rede neural artificial se refere a uma classe de algoritmos de aprendizado de máquina, que contempla modelos baseados no aprendizado conexionista, cuja proposição é uma tentativa de representar matematicamente o método de processamento do cérebro

<sup>1</sup> Disponível em <http://scopus.com>.

Figura 3 – Crescimento de publicações sobre aprendizado profundo, considerando artigos indexados na base de dados *Scopus*.



Fonte: Autor.

humano. No cérebro, um neurônio é estimulado por sinais de entrada e, de acordo com algumas propriedades internas, passa o impulso elétrico para frente ou não. Inspirado neste processo, um neurônio artificial é modelado da mesma forma: recebe dados de entrada, realiza um simples cálculo sobre eles e, em seguida, ativa uma saída. Originalmente, a saída de neurônios artificiais era um valor binário, imitando os neurônios biológicos.

O conceito de neurônio artificial proposto por McCulloch e Pitts (1943), enquanto o primeiro algoritmo para treinar um neurônio artificial a partir de dados computacionais, o *perceptron*, foi introduzido por Rosenblatt (1958). Formalmente, o neurônio artificial foi modelado por meio da seguinte equação:

$$y = \frac{1 + \text{sin}al(w^T * x + b)}{2} = \begin{cases} 1, & w^T * x + b > 0 \\ 0, & \text{caso contrário} \end{cases} \quad (2)$$

onde  $x$  é a entrada,  $y$  é a saída,  $w$  é o peso do neurônio e  $b$  é o viés (ou limiar) do neurônio. Esta mesma equação, por conveniência de cálculo, pode ser denotada como  $\text{sin}al(w^T * x + b)$ , onde a saída do neurônio assume os valores  $\pm 1$ . No algoritmo *perceptron*, o ajuste dos pesos do neurônio, que denota a aprendizagem do modelo, é realizado pela equação:

$$w = w + \frac{(y_j - d_j) * x_j}{2}, \quad (3)$$

sendo  $d_j$  a saída correta esperada do neurônio para a  $j$ -ésima amostra do conjunto de dados, tal que  $j = \{1, 2, \dots, N\}$ . A regra de ajuste dos pesos é executada para cada uma das  $N$  amostras do conjunto de dados e, muitas vezes são necessárias várias execuções sobre todo o conjunto de dados. Assim, cada iteração desse processo é denominada por época.

Embora esses algoritmos sustentem as bases para os métodos de aprendizado profundo moderno, eles ainda eram muito simplistas e só podiam ser aplicados a uma classificação binária simples. Para lidar com tarefas mais complexas, neurônios artificiais foram então combinados para criar estrutura de rede, dando origem à supracitada RNA. Similar ao fluxo do cérebro humano, a saída de certos neurônios artificiais é usada como entrada para outros, permitindo cálculos mais complexos. Além disso, um dos principais avanços para esta classe de modelos foi o uso de outras funções de saída (também conhecidas como funções de ativação). Diferentes da função sinal disposta na equação 1, essas viabilizam um ajuste de pesos mais eficiente e, conseqüentemente, a geração de melhores modelos neurais. Algumas funções de ativação comuns são sigmóide, tangente hiperbólica e linear retificada, as podem ser respectivamente expressas por:

$$\sigma(x) = 1 - e^{-x}, \quad (4)$$

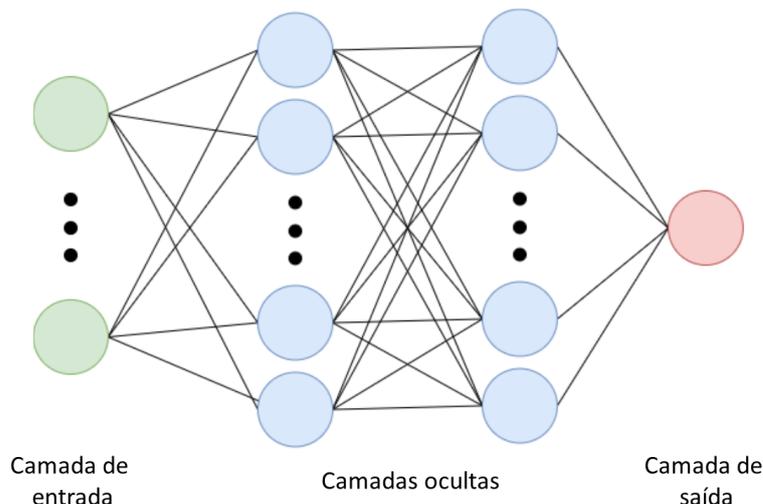
$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad (5)$$

$$ReLU(x) = \frac{x + |x|}{2}. \quad (6)$$

Uma arquitetura simples e comum para a estrutura de uma RNA é conhecida como *perceptron* de múltiplas camadas (MLP, do inglês *Multilayer Perceptron*), e sua estrutura básica inclui uma entrada (ou camada de entrada), seguida por um número discreto de camadas de neurônios intermediárias (chamadas de camadas ocultas) e, finalmente, uma camada de neurônios de saída. Em geral, uma MLP pode conter mais de uma camada oculta, com a saída de cada uma destas camadas servindo como entrada para a camada subsequente, conforme o exemplo demonstrado na Figura 4.

A complexidade da arquitetura composta em camadas na MLP requer um algoritmo de treinamento mais robusto em comparação com o neurônio artificial simples. Uma abordagem tradicional é usar o método *Newton-Raphson*, o qual permite atualizar os pesos da MLP usando pequenos incrementos na direção oposta do gradiente do erro em relação a cada peso. No entanto, para fazer isso, deve-se calcular o gradiente do erro, que geralmente é desconhecido e, em vez disso, é aproximado usando amostras do

Figura 4 – Diagrama de arquitetura de uma MLP com duas camadas ocultas.



Fonte: Adaptado de Mocanu, Nguyen e Gibescu (2017).

conjunto de dados. Na prática, geralmente isso é feito usando o método conhecido como descida estocástica do gradiente (SGD, do inglês *Stochastic Gradient Descent*), onde  $N$  subconjuntos das amostras são utilizados para calcular iterativamente o gradiente do erro dentro de uma época de treinamento.

Além disso, à medida que mais camadas são adicionadas à MLP, o cálculo da derivada em si se torna mais complexo devido à regra da cadeia. Para superar esse problema, é útil notar que, se o gradiente em relação a todos os pesos de camadas subsequentes é conhecido, a derivada atual é relativamente simples. Este princípio é usado para calcular os gradientes sequencialmente usando um algoritmo conhecido como *backpropagation* (LECUN, 1988). Finalmente, em muitos casos, é benéfico observar uma função do erro, geralmente conhecida como função de custo ou perda, em vez do erro propriamente dito.

Combinando todos os conceitos e métodos discutidos, pode-se obter um dos primeiros algoritmos importantes em aprendizado profundo: a descida estocástica do gradiente com *backpropagation* (BERNASCONI, 1990). Embora o SGD com *backpropagation* tenha sido concebido há muito tempo, ainda é um dos principais algoritmos utilizados para treinamentos de modelos de DL, bem como serve de base para abordagens mais modernas, como SGD com *momentum* (NESTEROV; NEMIROVSKII, 1994), *RMSProp* (TIELEMAN; HINTON, 2017) e ADAM (KINGMA; BA, 2015).

A capacidade da MLP encadear camadas ocultas permitiu avançar os resultados em várias áreas de pesquisa, principalmente em tarefas de visão computacional e processamento de linguagem natural. No entanto, a arquitetura canônica da MLP, conforme discutida, tem duas falhas principais: o número de parâmetros total que devem ser ajustados e o sobreajuste (do inglês, *overfitting*) do modelo aos dados. O número de parâmetros em uma MLP cresce na escala  $O(l_n W_d^2 D_e)$ , onde  $l_n$  é o número de neurônios de entrada,  $W_d$

é a quantidade de neurônios presentes nas camadas ocultas e  $D_e$  é o número de camadas ocultas. Obviamente, esta taxa de crescimento pode fazer o treinamento ser inviável para modelos de larga escala (profundos), pois requer a disponibilidade de recursos computacionais elevados. De fato, com o aumento de poder computacional nos últimos anos, principalmente com o advento da computação em nuvem, esse problema se tornou secundário ao *overfitting*.

O grande número de neurônios necessários para permitir que uma MLP faça um aprendizado significativo também leva a um modelo que é excessivamente adaptado ao conjunto de treinamento. Tal fenômeno ocorre porque cada neurônio é altamente dependente da ordem dos nós de entrada. Por exemplo, deslocar todos os pixels de uma imagem em uma distância de apenas um pixel em qualquer direção muda completamente o peso pelo qual cada pixel é multiplicado. Isso significa que, na prática, os neurônios da MLP tendem a ser dedicados a um subconjunto específico de amostras no conjunto de treinamento, fazendo com que estes se tornem muito sensíveis a qualquer mudança nessas amostras.

## 2.2.2 Rede Neural Convolutional

Em muitas tarefas nas quais redes neurais profundas (DNN, do inglês *Deep Neural Networks*) são utilizadas, como no campo de visão computacional, arquiteturas tradicionais são deixadas de lado em detrimento de abordagens que implementam o mecanismo de convolução matemática. Isto porque a convolução permite que modelos DNN sejam computacionalmente eficientes e invariantes a pequenas mudanças nos dados de entrada, como translação e escala em imagens. Analisando as deficiências previamente discutidas para modelos clássicos de rede neural baseada em aprendizado profundo, LeCun, Denker e Solla (1990) perceberam que é possível resolvê-las implementando a função de convolução em DNNs, alterando a estrutura fundamental de compartilhamento de peso entre os neurônios. Em sua proposta, os autores utilizam uma arquitetura na qual o funcionamento de um neurônio em sua entrada se baseia em uma função de convolução com um pequeno núcleo de pesos. Este novo mecanismo de compartilhamento de pesos resolve de modo eficiente o aprendizado do modelo profundo e o problema de *overfitting*. Esta descoberta foi fundamental para a difusão de modelos de aprendizado de máquina baseados em RNAs, viabilizando o avanço do estado da arte em uma ampla variedade de aplicações. Ainda, possibilitou a criação da que é, até hoje, a arquitetura de DNN mais utilizada na academia e indústria, denominada por rede neural convolutional (CNN). Em geral, a arquitetura de uma CNN é estruturada contemplando os seguintes componentes (CHOLLET, 2017):

- **Camadas Convolucionais** – composta por múltiplos *kernels* de convolução, os quais têm como função extrair características das imagens. É interessante notar que se torna possível obter características mais complexas (não lineares) acrescentando-se

o número destas camadas na rede, onde a saída de cada camada é utilizada como entrada nas camadas subsequentes. Estes *kernels* são ajustados automaticamente de modo supervisionado, por meio do algoritmo de *backpropagation*;

- **Camadas de Ativação** – responsáveis por realizar a operação de ativação não linear. Uma escolha tradicional para esta operação é a função ReLU (*Rectified Linear Units*), que permite computar a ativação elementar dos dados, em que  $f(x) = \max(0, x)$ . Esta escolha se dá, pois a função ReLU apresenta algumas vantagens sobre outras funções alternativas, tais como: (i) baixo custo computacional para realizar seu cálculo; (ii) reduz o problema de desaparecimento do gradiente, quando comparado com as funções tangente hiperbólica e sigmóide, devido às suas derivadas constantes (limiar rígido); e (iii) possui maior alcance de representação,  $[0, \infty]$ , em relação à função sigmóide,  $[0, 1]$ , por exemplo;
- **Camadas de Agregação (*Pooling*)** – nestas, os dados possuem sua escala reduzida, sem perda de representatividade. Em geral, este resultado é obtido calculando o máximo ou a média de uma pequena região das matrizes resultantes da convolução. Consequentemente, a agregação reduz a quantidade de parâmetros a serem aprendidos na CNN, contribuindo para o controle de *overfitting* e o melhor desempenho do modelo;
- **Camadas de *Dropout*** – estratégia para reduzir o *overfitting*, sendo que a principal ideia desta camada é eliminar aleatoriamente alguns neurônios e suas conexões. Empiricamente, esta estratégia se mostrou valiosa para o desenvolvimento de DL, uma vez que otimiza e valida a existência de versões menos robustas do modelo, o que contribui significativamente não só para a redução do *overfitting*, mas também para a melhoria de desempenho do modelo, pois reduz a dependência da rede por muitos parâmetros na camada em que a operação de *dropout* está sendo aplicada;
- **Camada Densa** – diferentemente da camada convolucional, na qual os pesos são conectados apenas em uma região dos dados, a camada densa está completamente conectada com a camada anterior.

Formalmente, na CNN o neurônio é modelado conforme a equação:

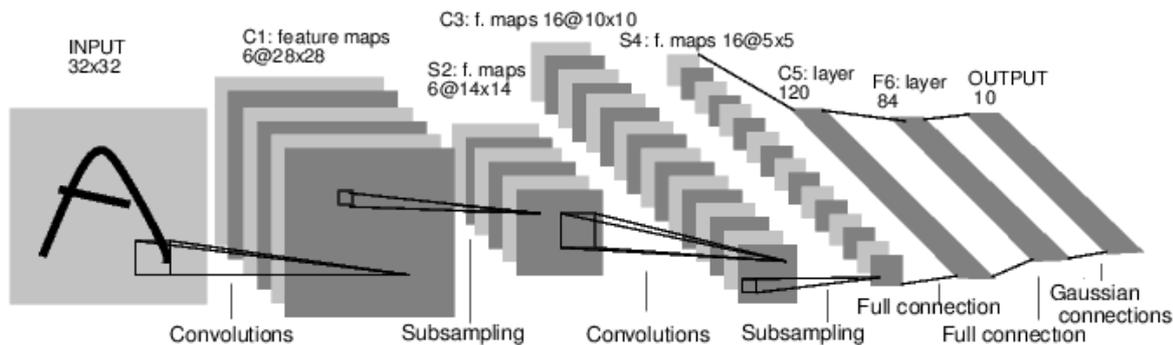
$$y = \alpha(w \otimes x + b), \quad (7)$$

sendo  $\otimes$  a operação de convolução,  $w$  o *kernel* de pesos do neurônio (também chamado de filtro) e  $\alpha$  é a função de ativação adotada. Vale ressaltar que, como o neurônio agora executa a convolução, sua saída  $y$  não é mais um valor escalar, mas um tensor, com o mesmo número de dimensões que a entrada  $x$ . Essa nova saída é geralmente conhecida como um recurso e, da mesma forma, a saída geral de uma camada convolucional é chamada de mapa de características (*feature map*).

Na prática, o processo convolucional realizado em uma CNN é geralmente uma combinação linear de convoluções em uma, duas ou três dimensões (para dados de séries temporais, imagens e vídeos, respectivamente). Isso significa que os pesos são compartilhados entre as dimensões espaço-temporais da entrada, mas não entre recursos diferentes, atribuindo a cada um seu próprio *kernel* convolucional. Além disso, cada estágio subsequente da rede incluirá uma redução na dimensão espaço-temporal do *feature map*. Esta redução de dimensionalidade, em geral, é feita através da média do valor máximo em uma determinada vizinhança (*pooling* máximo).

Uma das primeiras aplicações de sucesso de CNNs na área de processamento de imagens foi desenvolvida por Lecun et al. (1998). Neste trabalho, os autores propuseram o modelo LeNet-5, o qual foi utilizado para a tarefa de reconhecer dígitos manuscritos, obtendo uma taxa de erro de 0,8% no *dataset* MNIST, um marco para a área de pesquisa na época. A arquitetura usada no LeNet-5, expressa na Figura 5, ainda é a base para as aplicações de CNNs modernas, tendo como especificações uma camada de entrada com imagens de  $32 \times 32$  *pixels* e mais 7 camadas de pesos treináveis. As camadas convolucionais utilizam *kernels* com dimensões  $5 \times 5 \times 3$  e  $6 \times 6 \times 3$  para computar os mapas de características e *kernels* com dimensões  $2 \times 2 \times 3$  para realização de *pooling*. Por fim, além da camada de saída que possui conexão para 10 classes (os dígitos numéricos), a rede possui duas camadas totalmente conectadas de 120 e 84 neurônios.

Figura 5 – Diagrama da arquitetura LeNet.



Fonte: Lecun et al. (1998).

### 2.2.3 Principais Arquiteturas Convolucionais

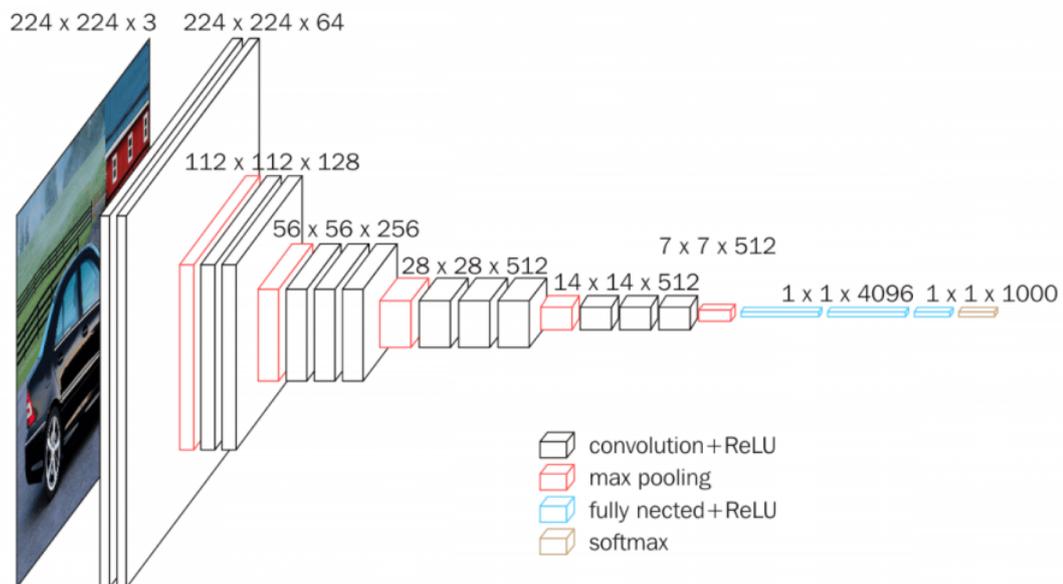
Novas arquiteturas de CNN vêm sendo investigadas e aplicadas com êxito em uma ampla variedade de problemas. Dentre vários trabalhos notáveis, destaca-se a proposição da arquitetura AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), a qual venceu o desafio ILSVRC 2012 (*ImageNet Large Scale Visual Recognition Competition*), inaugurando uma nova era para redes neurais profundas. Além desta, outras arquiteturas

relevantes de CNN são referências fundamentais na área de visão computacional, tais como VGG, ResNet, Inception, MobileNet e EfficientNet, as quais serão apresentadas a seguir.

### 2.2.3.1 VGG

Proposta por (SIMONYAN; ZISSERMAN, 2015), a arquitetura VGG (do inglês *Visual Geometry Group*) é um modelo de rede neural convolucional profunda amplamente utilizado em tarefas de visão computacional, especialmente em classificação de imagens. A característica-chave da arquitetura VGG é sua profundidade, com 16 ou 19 camadas, compostas principalmente por camadas convolucionais seguidas por camadas de max-pooling, conforme a Figura 6. As camadas convolucionais aplicam filtros à imagem de entrada, extraindo características em diferentes escalas espaciais. As camadas de max-pooling reduzem as dimensões espaciais dos mapas de características, mantendo as informações mais importantes.

Figura 6 – Diagrama da arquitetura VGG16.



Fonte: Simonyan e Zisserman (2015).

Um aspecto notável desta arquitetura é o uso de filtros pequenos, de dimensão  $3 \times 3$ , em toda a rede. Essa escolha de tamanho de filtro permite uma rede mais profunda, ao mesmo tempo que mantém o número de parâmetros computacionalmente eficiente. Ainda, segundo os autores, essa abordagem ajuda a capturar padrões mais complexos e melhorar a capacidade da rede de aprender representações hierárquicas.

A arquitetura VGG teve influência significativa no campo de aprendizado profundo, servindo como referência para modelos subsequentes. Ela alcançou desempenho de ponta

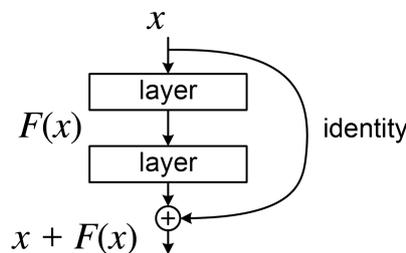
em várias tarefas de classificação de imagens, incluindo o ImageNet Large Scale Visual Recognition Challenge (ILSVRC) em 2014. A simplicidade e eficácia da arquitetura VGG a tornaram uma escolha popular para pesquisadores e profissionais em visão computacional. Também serviu como base para o desenvolvimento de arquiteturas mais avançadas, como ResNet e DenseNet, que se basearam nos princípios introduzidos pela VGG (ZAGORUYKO; KOMODAKIS, 2016).

### 2.2.3.2 ResNet

ResNet, também conhecida como *Residual Network*, é uma arquitetura de rede neural convolucional (CNN) que foi introduzida por He et al. (2016). A principal característica desta arquitetura é a utilização de conexões residuais, que permitem que a informação seja transmitida diretamente de uma camada para outra, contornando as camadas intermediárias.

Um componente fundamental na ResNet é o bloco residual (figura 7) que implementa duas camadas convolucionais  $3 \times 3$  e uma conexão residual (*identity*); este componente ajuda a mitigar o problema de desaparecimento do gradiente, que é comum em redes neurais profundas.

Figura 7 – Bloco Residual da arquitetura ResNet.



Fonte: He et al. (2016).

Devido à característica supracitada, a ResNet é conhecida por sua capacidade de permitir o treinamento de redes neurais muito profundas, com centenas de camadas, sem sofrer degradação no desempenho. Além disso, segundo os autores, a ResNet também utiliza blocos de convolução com atalhos, que ajudam a reduzir a complexidade computacional e melhorar a eficiência do treinamento.

No contexto de aprendizado profundo, a ResNet tem sido amplamente utilizada em várias aplicações, incluindo reconhecimento de objetos, classificação de imagens médicas, detecção de doenças, entre outros. Por exemplo, em um estudo sobre detecção de pneumonia em imagens médicas, o modelo ResNet-101 e ResNet-50 foram utilizados com sucesso Kareem, Liu e Sant (2022). Além disso, esta arquitetura também tem sido combinada com outras técnicas, como máquinas de vetores de suporte (SVM, do inglês

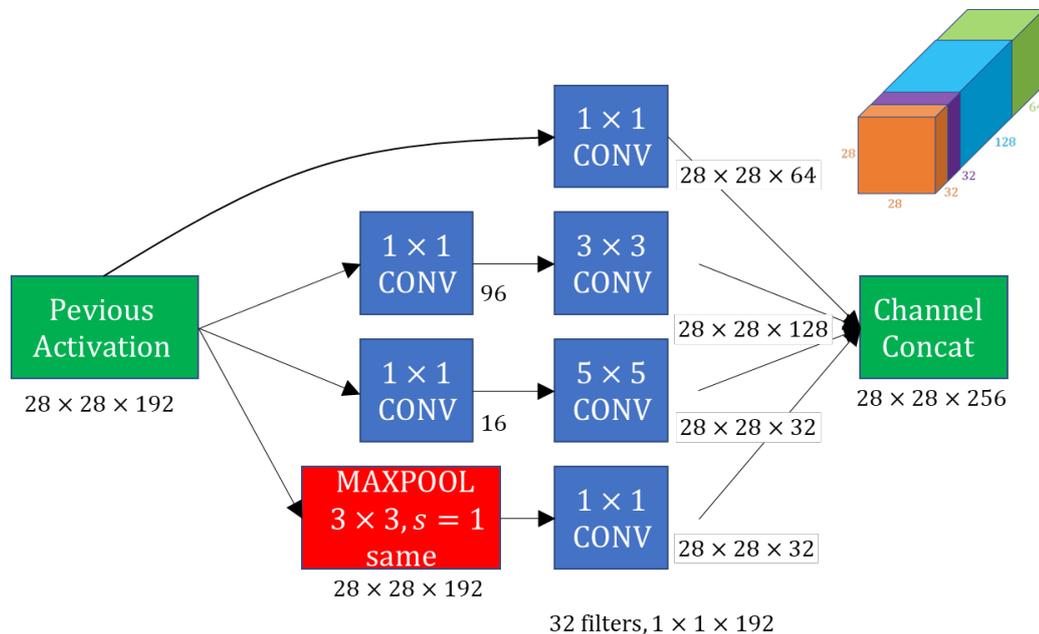
*Support Vector Machines*) e redes neurais recorrentes (RNN, do inglês *Recurrent Neural Networks*), para melhorar o desempenho em tarefas específicas Malik et al. (2022) Almutairi, Abubakar e Chiroma (2022).

Portanto, a arquitetura ResNet permite treinar redes neurais com um número elevado de camadas de forma eficiente, viabilizando a adoção de redes profundas sem degradação no desempenho, o que a torna uma escolha popular em várias aplicações de visão computacional.

### 2.2.3.3 Inception

A arquitetura *Inception*, também conhecida como *GoogLeNet*, foi proposta por Szegedy et al. (2015) e fez história ao vencer o desafio ImageNet 2014, representando um marco significativo no campo da visão computacional e do aprendizado profundo. Um dos aspectos cruciais abordados na concepção desta arquitetura foi a questão do tamanho dos núcleos de convolução utilizados nas redes neurais convolucionais. Conforme os autores, arquiteturas populares anteriores empregavam uma variedade de tamanhos de filtro, desde  $1 \times 1$  até  $11 \times 11$ .

Figura 8 – Exemplo de implementação do bloco *Inception*.



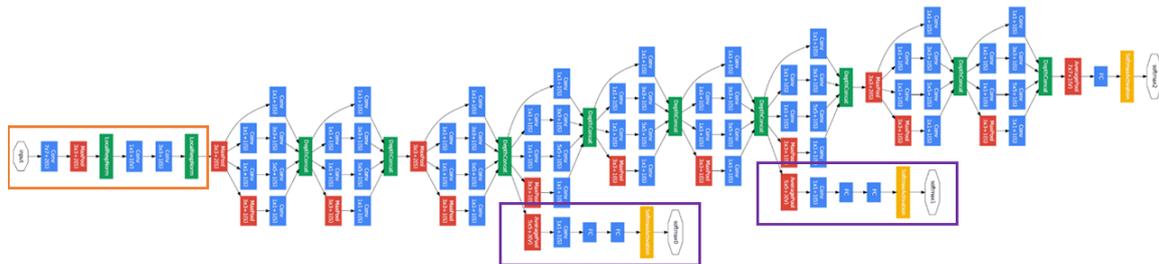
Fonte: Szegedy et al. (2015).

A abordagem inovadora da rede *Inception* foi a utilização de uma combinação de tamanhos de núcleos de convolução. O bloco convolucional fundamental na arquitetura *Inception* é conhecido como bloco *Inception*. Conforme a Figura 8, o bloco *Inception* é composto por quatro caminhos paralelos, com os três primeiros usando convoluções de diferentes tamanhos ( $1 \times 1$ ,  $3 \times 3$  e  $5 \times 5$ ) para extrair informações de diferentes escalas

espaciais. Os dois caminhos intermediários realizam convoluções  $1 \times 1$  para reduzir a complexidade do modelo. O quarto caminho emprega uma camada de *pooling* máximo (*max-pooling*) de  $3 \times 3$ , seguida por uma convolução para ajustar o número de canais. Os resultados de todos os caminhos são concatenados ao longo da dimensão do canal para formar a saída do bloco Inception.

A arquitetura *GoogLeNet* (*Inception V1*) possui um total de 27 camadas profundas (entre convolucionais e de *pooling*), utilizando um total de 9 blocos *Inception*, seguidos por uma camada de *pooling* médio global para reduzir a dimensionalidade, eliminando a necessidade de uma pilha de camadas totalmente conectadas na camada final Szegedy et al. (2015), conforme apresentado na Figura 9.

Figura 9 – Arquitetura *Inception V1*. A caixa laranja é o "tronco" da rede, que possui algumas convoluções preliminares; as caixas roxas são classificadores auxiliares. As demais estruturas são os blocos *Inception*.



Fonte: Szegedy et al. (2015).

A força da *Inception*, portanto, reside na combinação de filtros que exploram a imagem em uma variedade de tamanhos. Isso permite a eficiente detecção de detalhes em diferentes escalas, alocando diferentes quantidades de parâmetros para os filtros correspondentes.

Por fim, a arquitetura *Inception* provou ser altamente eficaz em várias competições de classificação de imagens, incluindo o desafio ILSVRC 2014, onde alcançou um desempenho notável e demonstrou uma taxa de erro significativamente menor em comparação com abordagens anteriores. Além disso, a *Inception* serviu como base para o desenvolvimento de modelos mais avançados, como a *Inception v3* e a *Inception-ResNet*, que melhoraram ainda mais o desempenho e a eficiência computacional (Szegedy et al. (2016)). Sua influência continua a ser notada na pesquisa em aprendizado profundo, consolidando seu status como uma das arquiteturas mais impactantes no campo da visão computacional.

#### 2.2.3.4 MobileNet

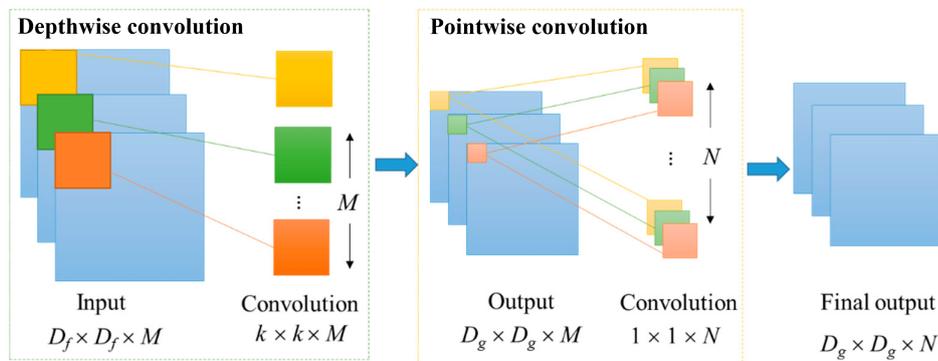
A *MobileNet*, proposta por (HOWARD et al., 2017), é uma arquitetura de CNN projetada com o objetivo de alcançar um equilíbrio entre precisão e eficiência computacional, permitindo que modelos de visão computacional sejam executados em plataformas com recursos limitados de hardware, especificamente *smartphones* e *tablets*.

Segundo os autores, uma das características fundamentais da arquitetura MobileNet é a utilização de convoluções separáveis em profundidade (*depthwise separable convolutions*). Essa técnica implementa a fatorização de convoluções que sugere, conforme a Figura 10, dividir o processo de convolução padrão em duas etapas (RODRIGUES et al., 2021), sendo:

- (a) uma convolução em profundidade (*depthwise convolution*) – responsável por aplicar convoluções individuais a cada canal do sinal de entrada;
- (b) uma convolução pontual (*pointwise convolution*) – adota uma convolução  $1 \times 1$  para combinar os resultados dessas convoluções separadas por canal.

No que tange os demais componentes da CNN, a MobileNet utiliza tanto a normalização em lote quanto as funções de ativação *ReLU* em ambas as camadas, contribuindo para sua eficiência e desempenho aprimorado. Os autores demonstraram que essa decomposição resulta em uma redução significativa na complexidade computacional, tornando o modelo mais leve e eficiente em termos de memória e processamento, viabilizando, assim, sua utilização em dispositivos móveis.

Figura 10 – *MobileNet*: mecanismo *depthwise separable convolutions*.



Fonte: Adaptado de (DANG; PANG; LEE, 2020).

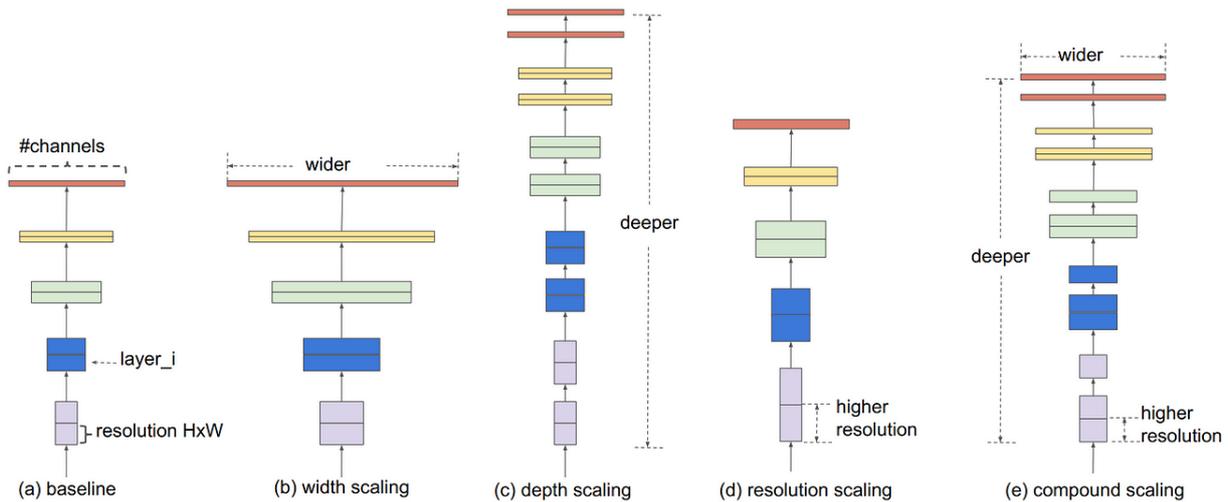
Desde sua proposição, a MobileNet demonstrou ser uma contribuição significativa para a área de visão computacional, possibilitando a execução eficiente de modelos de alta qualidade em dispositivos móveis. Sua eficácia foi comprovada em várias aplicações, incluindo detecção de objetos, segmentação semântica e reconhecimento de padrões (SANDLER et al., 2018).

### 2.2.3.5 EfficientNet

A EfficientNet é uma arquitetura de rede neural convolucional que foi desenvolvida pela equipe de pesquisa de Google (TAN; LE, 2019). O objetivo da EfficientNet é estabelecer

uma arquitetura eficiente em termos de recursos e desempenho, ao mesmo tempo em que oferece um bom desempenho em tarefas de aprendizado profundo.

Figura 11 – Visão geral da arquitetura EfficientNet.



Fonte: (TAN; LE, 2019).

Um conceito fundamental na arquitetura EfficientNet, conforme apresentado na Figura 11, é o escalonamento composto (do inglês, *compound scaling*), uma técnica que envolve o aumento simultâneo e equilibrado de três dimensões-chave de uma rede neural:

- ❑ **Profundidade** – referente ao número de camadas profundas, que permite aumentar a capacidade da rede de aprender representações complexas;
- ❑ **Largura** – diz respeito ao número de canais por camada, os quais visam aumentar a capacidade da rede de aprender representações detalhadas;
- ❑ **Resolução** – por fim, a resolução da imagem de entrada propriamente dita, a qual permite melhorar a capacidade da rede de aprender características de forma refinada, ou seja, representa um ajuste fino.

A arquitetura base da EfficientNet, denominada "B0", foi desenvolvida a partir de um processo de otimização Bayesiana, que teve como objetivo otimizar simultaneamente a acurácia e a quantidade de operações de ponto flutuante por segundo (FLOPS).

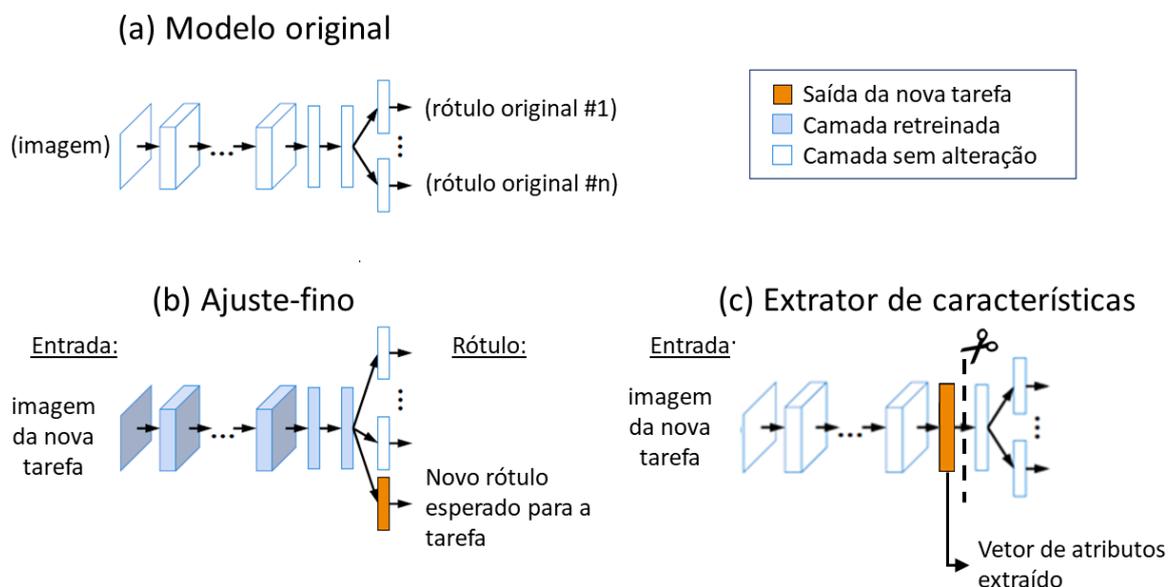
## 2.2.4 Transferência de Aprendizado

Embora originalmente treinados para o *benchmark* de classificação ImageNet (DENG et al., 2009), modelos como VGG e ResNet, têm sido amplamente usados como extratores de características para outras tarefas de classificação de imagens, bem como para

diferentes aplicações de visão computacional (por exemplo, detecção ou segmentação de objetos).

Conforme Hussain, Bird e Faria (2019), o processo de resolver uma tarefa  $A$  adaptando ou reutilizando um modelo que foi treinado para uma tarefa  $B$  é chamado de transferência de aprendizado (TL, do inglês *transfer learning*). Normalmente, o processo de *transfer learning* é aplicado de duas maneiras, a saber: através de ajuste fino ou usando o modelo original como extrator de características. Na primeira abordagem, é realizado um retreinamento controlado de um modelo original, onde o objetivo é modificar ligeiramente os pesos desse modelo para que ele possa resolver uma tarefa diferente. Dependendo da quantidade de dados disponíveis, pode-se escolher alterar os pesos de apenas algumas ou muitas camadas da rede. Na segunda abordagem, redes neurais profundas pré-treinadas também são usadas como extratores de características para codificar imagens em representações descritivas úteis, extraídas diretamente da saída de uma camada intermediária, resultando assim em um *embedding* (vetor de atributos não lineares) da amostra de entrada. Na Figura 12 é possível observar a comparação entre estes dois métodos.

Figura 12 – Diferentes abordagens de *transfer learning*: (a) arquitetura do modelo original; (b) estratégia de ajuste fino; e (c) estratégia de extrator de características.



Fonte: Adaptado de Li e Hoiem (2018).

Um dos desafios do processo de transferência de aprendizado é a falta de dados de qualidade e em grande volume, uma vez que para o *transfer learning* ser eficaz, pressupõe-se que o modelo da primeira tarefa foi treinado em um conjunto de dados grande e diversificado.

Neste contexto surge a ImageNet, uma base de dados de imagens em larga escala que tem sido amplamente utilizada para treinar e avaliar modelos de visão computacional. Ela

foi introduzida por (DENG et al., 2009) e consiste em cerca de 15 milhões de imagens, classificadas em 1.000 categorias. De acordo com os autores, a ImageNet foi construída com base na estrutura da WordNet, uma base de dados lexical que organiza palavras em sinônimos e relações semânticas. Cada categoria da ImageNet é representada por um conjunto de imagens relacionadas a um conceito específico. Essas imagens foram coletadas de várias fontes, como a web e contribuições de usuários.

Uma das principais aplicações da ImageNet é o desafio de reconhecimento visual em larga escala (ILSVRC, do inglês *ImageNet Large-scale Visual Recognition Challenge*), que ocorre anualmente. No ILSVRC, os participantes são desafiados a desenvolver modelos de aprendizado de máquina capazes de classificar imagens em milhares de categorias. Assim, a ImageNet tem sido fundamental para impulsionar o avanço da pesquisa em visão computacional, permitindo o desenvolvimento de modelos cada vez mais precisos.

Portanto, as estratégias de transferência de aprendizado tornam possível o uso de redes neurais profundas para aplicações para as quais grandes quantidades de dados não estão disponíveis para treinamento. Seguindo este princípio, um dos objetivos intrínsecos à pesquisa conduzida nesta tese é investigar o impacto da adoção de *transfer learning* para a classificação de cargas no contexto de NILM. Para tanto, é utilizada uma arquitetura VGG16 pré-treinada no conjunto de dados ImageNet (RUSSAKOVSKY et al., 2015) como extratora de características da imagem de entrada.

## 2.3 Transformação de Séries Temporais em Imagens

No contexto de NILM, os dados oriundos da medição de energia são dispostos na forma de série temporal, uma vez que o registro do consumo agregado possui uma ordem temporal natural.

Porém, um dos grandes desafios encontrados na manipulação de séries temporais é a identificação de padrões não lineares, os quais estão presentes em sistemas dinâmicos e/ou processos estocásticos. Estes fenômenos são recorrentes, de modo que certas regiões do espaço de estados são frequentemente visitadas (BERNDT; CLIFFORD, 1994; GRABEN et al., 2016; PIOREK, 2019). Apesar da relevância de comportamentos recorrentes em processos naturais, que vem sendo amplamente estudada a décadas, a visualização desses comportamentos geralmente é muito difícil no domínio do tempo (MONK; COMPTON, 1939).

Para superar os desafios na identificação de padrões não lineares em séries temporais, a transformação destes dados em imagens tem ganhado destaque, principalmente nos campos de aprendizado de máquina e processamento de imagens, devido à sua capacidade de representar informações temporais de forma mais visual e compacta.

Essa abordagem apresenta algumas vantagens, por exemplo, ao visualizar as séries temporais na forma de imagens, melhora-se a interpretabilidade, pois torna-se mais fácil

identificar padrões, tendências e anomalias. Isso é particularmente valioso em aplicações como a de NILM, onde a detecção de equipamentos elétricos individuais em séries temporais de consumo de energia é um desafio crucial. Além disso, as imagens geradas podem ser processadas por redes neurais convolucionais, que se destacaram em tarefas de visão computacional e processamento de imagens, uma vez que possuem grande capacidade de aprender padrões visuais que podem ser úteis para a classificação final.

A fim de estabelecer um mecanismo que supere as limitações inerentes à manipulação de dados temporais, Eckmann, Kamphorst e Ruelle (1987) criaram uma ferramenta chamada Gráfico de Recorrência. De acordo com os autores, um RP pode ser definido como: *"O Gráfico de Recorrência de uma série temporal de  $N$  pontos  $x(N)$  - sendo  $N$  o índice temporal - é uma matriz  $N \times N$  (matriz gráfica) preenchida por pontos brancos e pretos. O ponto preto, chamado de ponto recorrente, é colocado na matriz de recorrência com coordenadas  $(i, j)$  somente se a distância  $\rho(i, j)$  no instante  $n = 1$  e  $n = j$  (entre o estado atual do sistema e o estado a ser comparado) for menor que uma certa distância (raio)  $\epsilon$ , fixada no centro do estado corrente."*

Deste modo, tal solução permite a investigação da trajetória do espaço de fase  $m$ -dimensional por meio de uma representação 2D de suas recorrências. A partir da representação gráfica da série temporal obtida pelo RP, se torna possível saber em quais pontos essas trajetórias retornam a um estado previamente visitado. Em notação matemática, um RP pode ser expresso como:

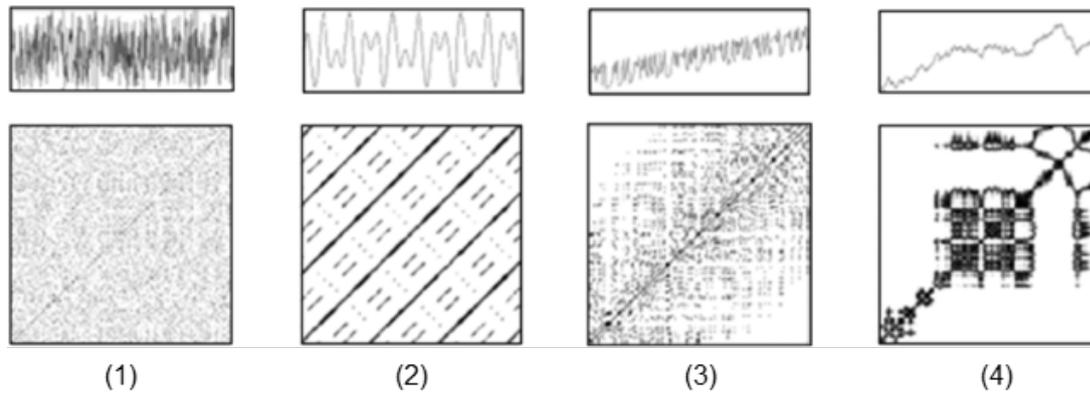
$$R_{i,j} = \theta(\epsilon - \|\vec{x}_i - \vec{x}_j\|), \vec{x} \in R^m, \quad i, j = 1, 2, \dots, N \quad (8)$$

onde  $N$  é o número de estados  $\vec{x}_i$  considerados;  $\epsilon$  é o raio da vizinhança (*threshold*) no ponto  $\vec{x}_i$ ;  $\|\cdot\|$  uma norma da vizinhança, comumente a norma euclidiana;  $\theta(\cdot)$  é a função *Heaviside*; e  $m$  é a dimensão de imersão.

O RP é interpretado de forma que se  $R_{i,j} = 1$ , o estado é dito recorrente e um ponto preto é marcado no RP. Caso  $R_{i,j} = 0$ , o estado é não recorrente e, conseqüentemente, define-se um ponto branco no gráfico. Portanto, a matriz  $R$  contempla características visuais como textura (pontos isolados, linhas diagonais, verticais e horizontais) e tipologia, caracterizando sistemas homogêneos, periódicos, variações lentas e variações abruptas (Figura 13).

Com base nessas informações, várias interpretações podem ser abstraídas, como, por exemplo, o desvanecimento para os cantos superior esquerdo e inferior direito significa que o processo contém uma tendência ou desvio; ou linhas/*clusters* verticais e horizontais mostram que alguns estados não mudam ou mudam lentamente por algum tempo e isso pode ser interpretado como estados laminares (MARWAN et al., 2007). No entanto, obviamente, existem padrões e informações em RP que nem sempre são fáceis de observar e interpretar visualmente.

Figura 13 – Exemplo de séries temporais e seus respectivos RPs: (1) RP homogêneo; (2) RP gerado por sistemas periódicos; (3) RP de sistemas com variação no(s) parâmetro(s); e (4) RP de uma série temporal altamente não estacionária.



Fonte: Adaptado de Marwan et al. (2007).

Por exemplo, em Silva, Souza e Batista (2013), os autores propuseram uma solução baseada em algoritmos de compressão de vídeo e imagens para classificar a similaridade entre diferentes séries temporais utilizando os RPs originados a partir destas, sob a premissa de que estes exprimem de modo gráfico o comportamento da série temporal. Os autores avaliam que estas informações podem ser úteis na classificação de séries temporais em uma ampla variedade de domínios de aplicação. Souza, Silva e Batista (2014), seguindo o mesmo contexto anteriormente descrito, conduzem um estudo que investiga a eficiência da combinação destes dois domínios para a classificação e identificação de padrões nos fluxos de dados. Neste estudo, os autores utilizam os RPs para classificar séries temporais coletadas em vários domínios, utilizando 38 *datasets* distintos. Ao final, após extraírem características textuais dos RPs, os autores alcançaram uma acurácia média de 78,71% na tarefa de classificação. Vale ressaltar que, apesar de proporem uma metodologia generalista, a abordagem discutida consegue superar técnicas tradicionais de classificação de séries temporais. Este resultado abre caminho para a investigação de métodos avançados de processamento de imagens para a classificação de séries temporais representadas por RPs.

Neste sentido, dentre as diversas arquiteturas de aprendizado profundo, conforme discutidas na seção anterior, as CNNs se tornaram uma das mais eficientes para tarefas de processamento de imagens e atingiram o estado da arte em problemas de reconhecimento de padrões devido à sua capacidade de realizar, em paralelo, a aprendizagem automática de características, de modo que seus desempenhos são mutuamente aprimorados utilizando um mecanismo computacional inspirado na organização e mecanismos do córtex visual dos mamíferos (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

Considerando que a aplicação de CNN em problemas de classificação de séries tem-

porais é dividida em duas estratégias (modificar a arquitetura tradicional da CNN e usar séries temporais 1D como entrada ou transformar os sinais 1D em matrizes 2D e então aplicar a CNN), Hatami, Gavet e Debayle (2018) conduziram um estudo que compara essas estratégias. Assim, os autores notaram que, apesar de grande parte da literatura focar em sinais 1D, a utilização de CNN aplicada a RPs se coloca como uma opção factível para a classificação de séries temporais, destacando que a representação gráfica da série temporal introduz diferentes tipos de características que não estão disponíveis para sinais unidimensionais. Os resultados demonstraram uma precisão competitiva desta abordagem, superando inclusive outros algoritmos definidos até então como estado da arte para a classificação de séries temporais em vários *benchmarks*. Todavia, os autores citam que apesar dos excelentes resultados obtidos, existe muito campo para a evolução do estudo, tais como a análise de diferentes arquiteturas de CNN e modelos alternativos de DL dedicados a problemas de visão computacional, adoção de outros métodos de otimização, avaliação de outras estratégias para a configuração otimizada de parâmetros dos modelos de DL, bem como a avaliação desta abordagem quando aplicada a outros conjuntos de dados.

Portanto, essa tese se fundamenta na necessidade de desenvolvimento e avaliação de métodos para a extração de conhecimento dos dados dependentes do tempo, os quais necessitam de tratamento especial e assertivo para a resolução eficiente de tarefas pertinentes à classificação de séries temporais. Neste sentido, entende-se que um *framework* híbrido que combine RPs e DL no contexto da identificação de cargas residenciais poderá contribuir de forma efetiva ao avanço do estado da arte.



---

## Capítulo 3

# Revisão Bibliográfica

---

Este capítulo visa apresentar o referencial teórico a cerca de NILM, abrangendo os conceitos históricos, conjuntos de dados disponibilizados pela comunidade científica, modelos tradicionais implementados, bem como as abordagens modernas baseadas em *deep learning* e uso de RPs para a classificação de cargas residenciais, estruturando a base fundamental para a pesquisa conduzida nesta tese.

### 3.1 Monitoramento não invasivo de cargas

Desde sua proposição, feita por Hart (1992), conforme já detalhado anteriormente, o campo de pesquisa de NILM tem atraído significativo interesse acadêmico e comercial. Atualmente, uma grande quantidade de trabalhos vem sendo publicados, bem como um crescente número de empresas estão encarando este tópico como estratégico para os seus negócios. Portanto, este capítulo irá apresentar uma visão geral das abordagens de NILM, compreendendo a revisão sobre os métodos e dados mais utilizados neste contexto.

#### 3.1.1 Conjuntos de Dados

Além do consumo agregado, os algoritmos projetados para a resolução de NILM dependem do consumo individual de cargas que compõem a residência analisada. Este consumo em nível de carga é essencial para realizar o treinamento de modelos baseados em aprendizado supervisionado, uma vez que fornece os rótulos esperados para realizar o processo de classificação de cargas e desagregação de consumo. Esta seção descreve quatro dos conjuntos de dados de NILM mais populares, enquanto a Tabela 1 fornece um

resumo de todos os conjuntos de dados disponíveis de modo aberto, conforme organizado por Himeur et al. (2020).

Tabela 1 – Conjuntos de dados disponíveis para pesquisas sobre NILM.

<i>Dataset</i>	Local	Duração	Nº de Casas	Aparelhos	Resolução Consumo Agregado	Resolução Consumo Aparelhos
REDD	EUA	119 dias	6	24	15 kHz	0.5 e 1 Hz
BLUED	EUA	8 dias	1	-	12 kHz	-
UK-DALE	Reino Unido	2,5 anos	5	5-54	16 kHz	6 seg.
Dataport	EUA	4 anos	722	1 a 25	1 min. 1 seg.	
BERDS	EUA	1 ano	1	4	20 seg.	20 seg.
Smart	EUA	3 meses	3	21-26	1 seg.	1 seg.
DRED	Holanda	6 meses	3	12	1 seg.	1 seg.
Tracebase	Alemanha	-	15	158	-	1 e 10 seg.
AMPDS	Canadá	1 ano	1	19	1 min.	1 min.
AMPds2	Canadá	2 anos	1	21	1 min.	1 min.
iAWE	Índia	73 dias	10	33	1 seg.	1 seg.
REFIT	Reino Unido	2 anos	20	11	8 seg.	8 seg.
GREEND	Áustria/Itália	1 ano	9	9	1 seg.	1 seg.
ECO	Suíça	8 meses	6	-	1 seg.	-
IHEPCDS	França	4 anos	1	3	1 min.	1 min.
OCTES	Vários	4-13 meses	33	-	7 seg.	-
HES	Reino Unido	1 mês - 1 ano	251	13-51	2 min.	2 min.
ACS-F1	Suíça	1 e 2 horas	-	100	10 seg.	10 seg.

Fonte: Adaptado de Himeur et al. (2020).

O *dataset* REDD (*Reference Energy Disaggregation Dataset*) foi disponibilizado publicamente para a comunidade científica em 2011 (KOLTER; JOHNSON, 2011). Este conjunto de dados contempla o registro do consumo de energia agregado, bem como em nível de carga, para seis residências no estado de Massachusetts, nos Estados Unidos (EUA), as quais foram monitoradas por aproximadamente 119 dias. Segundo os autores, contando com 1 *terabyte* de dados, este conjunto representava a maior coleção pública de dados de desagregação de energia até então.

Em 2012, o conjunto de dados BLUED (*Building-Level fully labeled Electricity Disaggregation*) foi publicado por Anderson et al. (2012), contemplando o registro de consumo energético de uma única família na Pensilvânia, EUA. Diferentemente da REDD, em vez de registrar o consumo das cargas por meio de medidores individuais, os autores adotaram o uso de *tags* de eventos para rotular as mudanças no estado das cargas ao longo do

tempo. Portanto, para cada carga, pode-se saber o início e o fim da atividade com base em rótulos de textos, mas nenhum consumo real de energia foi registrado.

O Imperial College disponibilizou o conjunto UK-DALE (*United Kingdom Domestic Appliance Level Electricity*) em 2015, o qual contempla registros acerca do consumo agregado para eletrodomésticos monitorados no período de 2,5 anos (KELLY; KNOTTENBELT, 2015). A coleta de dados abrangeu mais de 5 residências, cobrindo até 54 cargas monitoradas com resolução de 6 segundos. O registro de consumo agregado de cada residência é disponibilizado com resolução de alta (16 kHz) e baixa frequência (1 Hz). Ainda em 2015, Parson et al. (2015) publicaram o conjunto de dados Dataport/Peccan Street Inc. Cobrindo mais de 700 residências nos EUA, as quais tiveram registradas durante 4 anos suas informações de consumo agregado e consumo individual de cargas em uma resolução de 1 minuto, este é o maior *dataset* para NILM já disponibilizado até o momento.

### 3.1.2 Abordagens Tradicionais

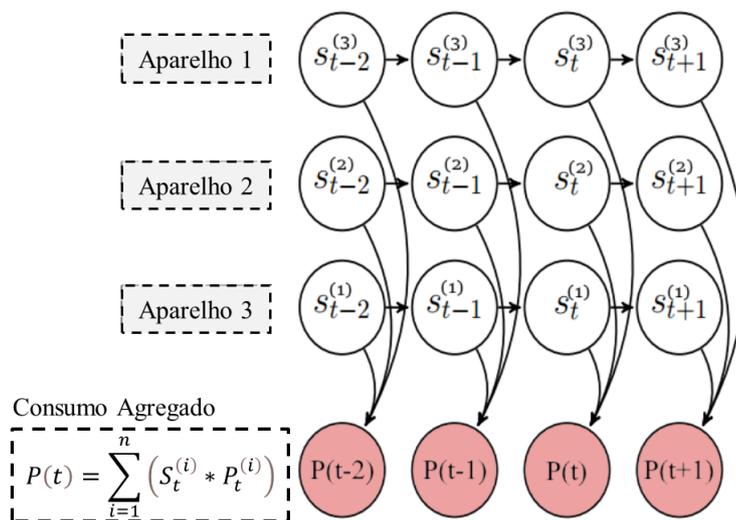
Uma estratégia amplamente difundida para resolver o problema de NILM é usar a programação inteira (IP, do inglês *Integer Programming*). Em Suzuki et al. (2008), os autores usam IP para desagregação de carga por meio da otimização do erro quadrático médio (MSE, do inglês *Mean Squared Error*) da corrente total, uma vez que é assumido que esta é composta por uma combinação linear de assinaturas de corrente pré-determinadas. Para a implementação deste método, os autores realizaram o registro individualizado de assinatura de corrente de cada estado operacional das cargas, levando em consideração a restrição de que todos os coeficientes da combinação linear devem ser inteiros não negativos. A fim de auxiliar a convergência, os autores também incluem algumas outras restrições nos valores dos coeficientes, como, por exemplo, impedindo uma carga de estar em dois estados simultaneamente. Embora neste trabalho os autores usem assinaturas de corrente, o IP pode ser facilmente adaptado a qualquer outra medida elétrica. Apesar de promissor, o IP em seu formato original requer a identificação de cada estado individual das cargas e utiliza qualquer relação entre estados, exceto na forma de restrições.

Bhotto, Makonin e Bajić (2017) implementam melhorias adicionais em uma solução denominada pelos autores de programação linear inteira auxiliada (ALIP, do inglês *Aided Linear Integer Programming*), a qual contempla restrições adicionais para evitar ambiguidades em soluções de IP, incluindo o aproveitamento de uma formulação de máquinas de estado das cargas. Também, tal método ainda introduz a filtragem da mediana para evitar soluções irreais da IP, como, por exemplo, comutação de certas cargas. Além destas melhorias, os autores refinam os resultados usando programação linear, sem restrições de inteiros, para contabilizar possíveis assinaturas em situações transitórias. Esta última modificação denota que o método ALIP proposto é na verdade uma forma de programação linear inteira-mista (MILP, do inglês *Mixed-Integer Linear Programming*).

Tomando como base o estudo anterior, Wittmann, Lopez e Rider (2018) sugerem resolver o problema utilizando uma janela inteira de amostras de uma vez. Justificam os autores que esta estratégia permite a inclusão de novas restrições relacionadas ao comportamento temporal das máquinas de estado das cargas. Além disso, os autores resolvem o problema de otimização para potência ativa e reativa simultaneamente, seguindo a mesma abordagem de Hart (1992).

Outra estratégia tradicional para resolução de NILM consiste na adoção de Modelos Ocultos de Markov (HMM, do inglês *Hidden Markov Model*) para inferir os estados das cargas a partir do consumo agregado em cada instante de tempo. Numa abordagem de NILM considerando HMM, os estados  $S_t^{(i)}$  de cada carga são tratados como uma sequência de variáveis ocultas definidas pela função de distribuição de probabilidade, enquanto o consumo agregado  $P(t)$  é uma variável observada correspondente à combinação desses estados ocultos. Esta abordagem foi inicialmente proposta em Parson et al. (2011), onde, considerando apenas 3 cargas, os autores projetaram manualmente as cadeias de *Markov* subjacentes com base em dados coletados e, em seguida, os ajustaram durante o processo de desagregação.

Figura 14 – Representação do modelo FHMM implementado para NILM.



Fonte: Adaptado de Zoha et al. (2012).

Uma das variantes de HMM implementadas para NILM é a *Factorial Hidden Markov Model* (FHMM), conforme ilustrada na Figura 14. Com base neste modelo, Zoha et al. (2013) implementam uma solução que obteve até 90% de acurácia para a desagregação de consumo para cargas do tipo II (ON/OFF). Kolter e Jaakkola (2012) desenvolveram uma versão alternativa desta abordagem, denominada pelos autores como AFHMM (*Additive Factorial Hidden Markov Model*), que estabeleceu até então uma nova linha de base para

o processo de desagregação, com acurácia média de 71% ao considerar 7 cargas de tipos distintos. No contexto de modelos *markovianos*, diversas variantes do FHMM foram investigadas, destacando-se os trabalhos de Kim et al. (2011), Zeifman e Roth (2011) e Parson et al. (2012). Em tempo, ressalta-se ainda o trabalho de Kong et al. (2016), onde uma adaptação de IP para NILM é apresentada, na qual a programação inteira é usada como uma medida para melhorar o desempenho de um modelo FHMM para desagregação de carga.

Porém, conforme Kim, Le e Kim (2017), abordagens baseadas em FHMM sofre de relevantes limitações, tais como:

1. O aumento nos estados ocultos do modelo e no número de aparelhos leva a um aumento exponencial na complexidade computacional que, por sua vez, degrada de modo significativo a acurácia nas inferências destes modelos;
2. Apesar da alta acurácia em cargas de Tipo I, os modelos HMM apresentam baixo desempenho quando avaliados em cargas de múltiplos estados (Tipo II);
3. Mesmo com cargas do Tipo I, estes modelos têm dificuldade em distinguir cargas com consumo energético semelhante.

A fim de mitigar o problema relacionado ao crescente número de estados, uma possível solução é usar a inerente dispersão das transições de estado das cargas, uma vez que, geralmente, não mais do que uma ou duas cargas mudam seu estado interno ao mesmo tempo. Isso significa que, embora o número de estados cresça exponencialmente, conforme mencionado, o número de probabilidades de transição diferentes de zero não. Essa abordagem foi discutida inicialmente por Makonin, Bajic e Popowich (2014) e, em seguida, expandida por Makonin et al. (2016), usando o que os autores chamaram de *Superstate Hidden Markov Model* (SSHMM). Explorando a esparsidade acima mencionada, os autores foram capazes de treinar um modelo para mais de 20 cargas, executando-o em tempo real com elevada acurácia (superior a 90%) para dados com baixa taxa amostral.

Abordagens baseadas em aprendizado de máquina (ML, do inglês *machine learning*) e reconhecimento de padrões têm sido investigadas no contexto de NILM. No que tange especificamente o problema de identificação de cargas – tema central do problema abordado nesta tese – tal objetivo pode ser encarado como uma tarefa de classificação, a qual pode ser resolvida através do treinamento supervisionado de modelos de ML com base em dados que contemplem, além do consumo agregado de energia da rede elétrica, os rótulos de operação das cargas em cada instante de tempo. Roos et al. (1994) propõem o treinamento de um conjunto de redes neurais artificiais utilizando dados compostos por atributos extraídos em regime permanente. Assim, as redes neurais são usadas em cascata para realizar a identificação das cargas. Abordagens similares também foram analisadas por Berges et al. (2009) e Kolter e Johnson (2011).

Porém, uma vez que as cargas têm características individuais, atributos como as potências ativa, reativa e aparente podem não ser suficientes para a identificação eficiente de todos os tipos. Srinivasan, Ng e Liew (2006) corroboram tal afirmação em seu trabalho, no qual apresentaram uma abordagem que utiliza informações de correntes harmônicas para realizar a identificação das cargas, avaliando diferentes algoritmos: redes neurais artificiais (com arquitetura de Função de Base Radial (RBF, do inglês *Radial Basis Function*) e *Perceptron* de Múltiplas camadas) e máquinas de vetores de suporte (SVM, do inglês *Support Vector Machines*) com *kernels* polinomiais, lineares e RBF. O processo de classificação das cargas obteve acurácias entre 80% e 90%. Além do resultado quantitativo obtido, os autores afirmaram que equipamentos de funcionamento semelhantes possuem assinaturas harmônicas similares, compreendendo sutis diferenças em correntes harmônicas de ordens elevadas. Em Fernandes, Silva e Oleskovicz (2013), os autores empregaram Redes Neurais Artificiais (RNA) do tipo MLP em dados obtidos por meio de ensaios laboratoriais, onde foi possível demonstrar o potencial das componentes harmônicas no processo de identificação das cargas, obtendo uma precisão média de 98% para seis diferentes aparelhos residenciais. Porém, os autores também exploraram as vantagens de algoritmos de seleção de atributos para determinação das componentes mais relevantes à identificação das cargas e suas combinações.

Diversas abordagens de processamento de sinais que consideram o domínio temporal e da frequência vêm sendo aplicadas para a resolução de NILM ao longo dos anos. Um exemplo é apresentado em Leeb, Shaw e Kirtley (1995), onde os autores implementam a análise espectral através da Transformada *Wavelet* para a identificação de 4 tipos de cargas. Ting et al. (2005) propõem o uso da trajetória V-I para a caracterização de cargas a partir do consumo agregado. Assim, as assinaturas baseadas na trajetória V-I são obtidas a partir da amostragem da forma de onda da tensão e da corrente em um ciclo (em regime permanente). Tal abordagem foi também explorada por Lam, Fung e Lee (2007) para realizar a taxonomia de cargas por meio de uma abordagem baseada em agrupamento hierárquico. Os autores notaram que os grupos de cargas da taxonomia baseada na trajetória V-I eram mais bem separados quando comparados às taxonomias baseadas em métricas de potência tradicionais.

Métodos baseados em formas de onda e processamento de sinais, incluindo Distorção Harmônica Total e Fator de Crista, foram usados em Yang, Chang e Lin (2007) e, segundo os autores, forneceram um ganho na capacidade de reconhecimento das cargas, atingindo acurácia de até 95.6% ao empregar um classificador baseado em rede neural. Em Lin, Tsai e Chen (2011), os autores também usaram o Fator de Crista para alcançar taxas de identificação superiores a 93% para três cargas diferentes. Gupta, Reynolds e Patel (2010) avaliaram o uso de interferências eletromagnéticas na faixa de MHz para o processo de identificação, obtendo acurácia de 94% e considerando até 20 cargas.

Diferentes técnicas baseadas em aprendizado não supervisionado e abordagens de oti-

mização foram investigadas para classificar eventos em NILM e atingiram resultados interessantes ao longo dos últimos anos. Machlev, Levron e Beck (2018), implementa uma versão modificada do método de Entropia Cruzada (MCE, do inglês *Modified Cross-Entropy*) obtendo um f1-score médio de 86% para a classificação de 13 diferentes aparelhos. Os autores também argumentam que o resultado foi obtido em uma abordagem que não requer treinamento prévio, nem utiliza hardware complexo. Moradzadeh et al. (2020) avaliaram uma abordagem de aprendizagem não supervisionada, aplicando o algoritmo *Principal Component Analysis* (PCA) para identificar os padrões de consumo dos dispositivos. Tomando como entrada a soma das curvas de carga individuais dos dispositivos, eles obtêm um f1-score médio de 94% para dados de três residências diferentes. Em Hua et al. (2021), os autores definem o problema de detecção de carga como um modelo de programação linear inteira mista e, considerando oito dispositivos diferentes, obtêm 92% de acurácia no processo de detecção de eventos.

Além disso, estudos têm sido conduzidos com o objetivo de reduzir a necessidade de dados ou recursos computacionais complexos para o treinamento de modelos de aprendizado de máquina aplicados em NILM. Singh e Majumdar (2019) discute em detalhes esses aspectos para o treinamento de modelos NILM supervisionados. Nesse sentido, usando um pequeno conjunto de dados, os autores modificam a abordagem de classificação baseada em representação esparsa para realizar a classificação de cargas e atingir f1-score médio acima de 65% para quatro cargas diferentes. A partir da compressão do sinal de consumo agregado usando o método *Signal2Vec*, Nalmpantis e Vrakas (2020) aplica um modelo de rede neural multicamadas para realizar a classificação de carga em uma estratégia multirrotulo e alcançar um f1-score de até 51,8% no cenário avaliado, contemplando treze cargas. Correa e Castro (2020) empregam uma arquitetura de rede neural artificial rasa para extrair atributos do sinal agregado de potência e, em sequência, aplica uma MLP para classificar o estado operacional de seis cargas, obtendo f1-score de 87,9%. Zhou et al. (2021), com o objetivo de mitigar problemas como o longo tempo de treinamento e incerteza de convergência de modelos baseados em redes neurais, os autores propõem uma abordagem de rede neural probabilística auto-organizada (PNN, do inglês *Probabilistic Neural Network*) para reduzir significativamente a complexidade da classificação da tarefa executada pelas cargas. A abordagem personalizada atinge f1-score médio de 56,75% avaliando 4 cargas diferentes.

### 3.1.3 Abordagens Baseadas em *Deep Learning*

Redes Neurais Profundas atingiram reconhecida capacidade de resolução de complexos problemas de classificação, principalmente no âmbito de visão computacional, domínio no qual vêm se consolidando como estado da arte em diversas tarefas, tornando-as assim uma escolha óbvia para a resolução de NILM. Dentre diversos motivos que justificam essa escolha, destacam-se:

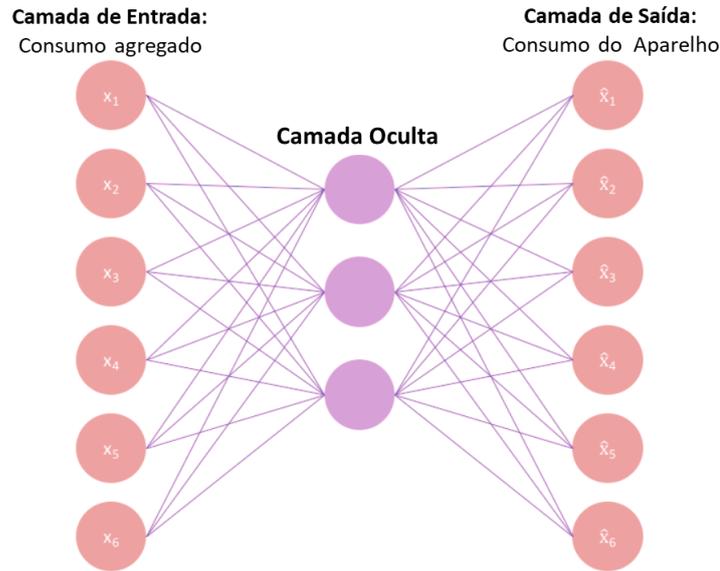
1. Modelos de DL são flexíveis, ao ponto de permitir que arquiteturas semelhantes sejam implementadas sem grandes modificações em problemas de domínios distintos. Assim, é possível reaproveitar o conhecimento estabelecido em estudos anteriores para potencializar a tarefa de interesse;
2. Permitem realizar a extração automática de atributos abstratos de alto nível, o que garante a detecção de cargas a partir de dados agregados com conhecimento prévio mínimo sobre as variáveis do sistema;
3. Apesar do alto custo computacional durante a fase inicial de treinamento, uma vez treinado, estes modelos realizam inferências com bom desempenho e esforço computacional de baixo custo.

De modo geral, todas as aplicações de DL no contexto de NILM se baseiam em variantes das arquiteturas convolucionais (CNN) e recorrentes (RNN, do inglês *Recurrent Neural Network*). Estas escolhas se dão pelo fato de que tais arquiteturas possuem capacidade de aprender atributos latentes de séries temporais, a qual é a estrutura de dados fundamental do problema de NILM.

Uma das primeiras tentativas de utilizar modelos de DL para a resolução de NILM foi conduzida por Kelly e Knottenbelt (2015), onde os autores adaptaram arquiteturas distintas para resolver o problema de desagregação: um modelo baseado em *Noise Autoencoder* (DAE) (VINCENT et al., 2008), um LSTM-bidirecional (SCHUSTER; PALIWAL, 1997) para desagregação direta e uma rede neural convencional para estimar o início, o fim e potência média de cada ativação das cargas. Os autores concluíram que as 3 abordagens avaliadas obtiveram um *f1-score* médio para 5 cargas melhor do que modelos alternativos de otimização combinatorial e HMM, além de mostrar relevante capacidade de generalização para residências não vistas durante o treinamento. Além dos resultados quantitativos, a solução apresentada apontou várias descobertas e, como qualquer trabalho pioneiro, limitações. No que tange a parte técnica, os autores implementaram arquiteturas canônicas e propuseram o treinamento de modelos individuais por carga, o que levou a um moroso processo de escolha de hiper parâmetros, além de demandar grande capacidade computacional, uma vez que foi necessário treinar uma arquitetura de mais de 150 milhões de parâmetros para cada uma das cargas. Todavia, este trabalho foi a pedra fundamental para o início de pesquisas que envolvem o uso de DL para a resolução de NILM.

Proposta inicialmente para aplicação no domínio de processamento de imagens, a arquitetura DAE foi projetada com o objetivo de reconstruir fotografias granuladas (VINCENT et al., 2008). De maneira simples, a DAE é uma rede neural que tenta reconstruir sua entrada por meio de um número esparsa de atributos na camada oculta, conforme a Figura 15.

Figura 15 – Arquitetura DAE projetada para NILM.

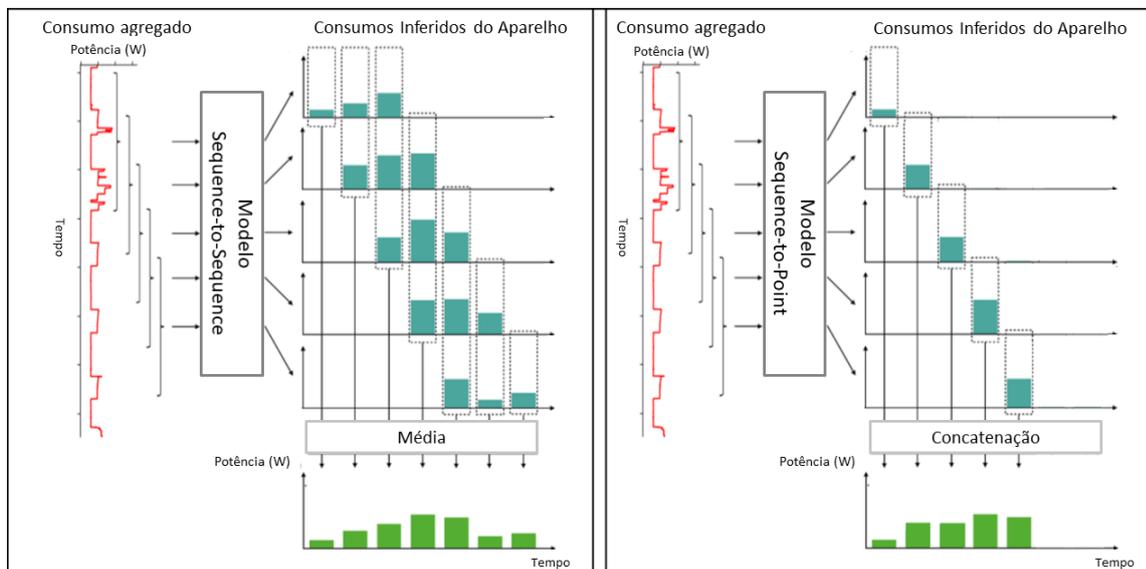


Fonte: Adaptado de Vincent et al. (2008).

Enquanto DAEs convencionais requerem manipulação artificial dos dados de entrada para treinar os modelos, em NILM o consumo agregado de energia elétrica é considerado como a entrada ruidosa, onde o objetivo é reconstruir a demanda de energia individual da carga de interesse em detrimento do ruído adicionado pelo consumo simultâneo de outras cargas. Conforme apresentado por Kelly e Knottenbelt (2015), a rede foi treinada a partir de uma sequência de tamanho fixo de consumo agregado (janela), e as camadas de convolução ocultas tentam reconstruir o consumo correspondente do aparelho de interesse. A princípio, o modelo aprende a melhor regressão não linear que permite mapear de uma janela deslizante de entrada do consumo agregado para uma janela de mesmo comprimento de consumo de energia individual da carga. Uma vez que isso é formulado como problema de regressão, a função de perda adotada para o treinamento do modelo é a Raiz do Erro Quadrático Médio (RMSE, do inglês *Root Mean Squared Error*). Durante a desagregação, as saídas da rede se sobrepõem em janelas do consumo da carga, como mostrado na Figura 16. Esta abordagem é conhecida como aprendizagem de sequência para sequência (seq2seq, do inglês *sequence-to-sequence*) e tem sido usada com sucesso nessa classe de problema (GRAIS; SEN; ERDOGAN, 2014). A principal desvantagem desta abordagem é que a produção final da meta tende a subestimar o consumo real devido à média entre as janelas sobrepostas produzidas.

Para mitigar as limitações do modelo *seq2seq*, Zhang et al. (2018a) propuseram uma abordagem alternativa chamada aprendizagem de sequência para ponto (seq2point, do inglês *sequence to point*), que tem sido amplamente usada desde então para modelar distribuições em tarefas relacionadas ao domínio de imagens e fala. O princípio por trás

Figura 16 – Diagramas de funcionamento dos modelos seq2seq (à esquerda) e seq2point (à direita).



Fonte: Adaptado de Kelly e Knottenbelt (2015).

do *seq2point* é que o modelo neural recebe uma janela de comprimento fixo do consumo agregado e produz em sua camada de saída, composta de um único neurônio, o ponto médio correspondente ao consumo individual da carga de interesse dentro da janela de entrada. Ao fazer isso, a rede aprende um mapeamento sequência para ponto e elimina qualquer sobreposição na saída, problema fundamental ao *seq2seq*. A Figura 16 mostra a comparação entre essas duas abordagens. Na imagem à direita é possível observar como o modelo *seq2point* fornece uma previsão de ponto único para cada janela de consumo da rede recebida como entrada e a saída final é uma concatenação de pontos gerada sobre a janela deslizante, a qual fornece a previsão para toda a duração do consumo agregado sem a necessidade de calcular a média ou realizar qualquer tipo de operação de agrupamento como na abordagem de aprendizagem sequência para sequência.

No que tange modelos recorrentes, a arquitetura LSTM (do inglês *Long-Short Term Memory*), assim como qualquer outra arquitetura de rede neural recorrente, é projetada para modelar as dependências temporais a partir do sinal analisado, o que a torna uma escolha óbvia de modelo de DL para NILM. Em Kim, Le e Kim (2017) ao invés de usarem simplesmente o sinal original, os autores propõem adotar uma assinatura de carga calculada manualmente, a qual é obtida considerando a suavização da potência agregada por meio de um filtro autorregressivo de pólo único e, posteriormente, incluindo a diferença de primeira ordem do sinal suavizado como entrada adicional para o modelo realizar a desagregação de aparelhos do tipo II.

Em Le, Kim e Kim (2016) os autores adotaram uma arquitetura de unidade recorrente fechada (GRU, do inglês *Gated Recurrent Unit*) ao invés de uma LSTM. Todavia,

a arquitetura implementada limitava a complexidade do modelo e, conseqüentemente, o desempenho. No entanto, a introdução de GRUs para NILM serviu de base para outros trabalhos. Em Rafiq et al. (2018), por exemplo, os autores incluíram uma regularização usando *dropout* (HINTON et al., 2012) para ambos LSTM e GRUs, e observou uma melhoria relevante principalmente na generalização do modelo, compreendendo uma acurácia média de 78%, e f1-score médio de 40%, em residências nas quais o modelo não havia sido treinado *a priori*.

Uma vertente que vem despertando a atenção da comunidade científica mais recentemente para a resolução de NILM é utilizar a transformação de sinais temporais em imagens 2D (BAPTISTA et al., 2018; BAETS et al., 2018; KYRKOU; NALMPANTIS; VRAKAS, 2019; SINGHAL; MAGGU; MAJUMDAR, 2019), uma vez que modelos baseados em arquitetura CNN são reconhecidamente estado da arte em diversas tarefas de visão computacional.

Em Baptista et al. (2018), os autores propuseram o uso de uma CNN embarcada em FPGA (*Field Programmable Gate Array*), responsável por rotular um total de 11 cargas contidas na base PLAID (*Plug Load Appliance Identification Dataset*). Neste sentido, o estágio de extração de características foi baseado na transformação dos sinais de tensão e corrente em trajetórias V-I, ou seja, conforme descrito anteriormente, imagens 2D que representam o comportamento das cargas. Ao analisar o desempenho da CNN, foi possível notar um f1-score de 78,16%.

Outra abordagem baseada nas trajetórias V-I foi proposta por Baets et al. (2018), em que os *pixels* da imagem recebem pesos. Além disso, tal abordagem fora validada sobre as bases PLAID e WHITED (*Worldwide Household and Industry Transient Energy Dataset*). Novamente, foi empregado um modelo preditivo baseado em CNN, o qual teve seu desempenho avaliado em termos de f1-score, obtendo respectivamente 77,60% e 75,46% para os conjuntos de dados analisados.

Cabe comentar que, apesar dos bons resultados obtidos em Baptista et al. (2018) e Baets et al. (2018), a obtenção das imagens 2D por meio de trajetórias V-I somente foi possível devido aos dados serem coletados com taxa amostral elevada (30 a 44 kHz), o que representa uma solução relativamente cara e normalmente inviável para a área de NILM. Além desse aspecto, as abordagens foram testadas sobre conjuntos de dados pouco utilizados na literatura.

Assim, em Kyrkou, Nalmpantis e Vrakas (2019), os autores propõem transformações de dados considerando os conjuntos UK-DALE e REDD. Nesta abordagem, os autores avaliam o comportamento de uma estratégia de *Transfer Learning* para cargas do tipo refrigerador. Neste sentido, os dados temporais (coletados a cada 1 Hz, ou seja, em baixa frequência) são transformados em imagens 2D por meio de GADF (*Gramian Angular Difference Fields*). Essas imagens são então fornecidas como entradas a uma CNN VGG16 (modelo pré-treinado) para alterar o espaço de características. Tal alteração foi

realizada por meio de dados da UK-DALE. No entanto, para validar o processo de classificação/rotulação de refrigeradores empregou-se os dados da REDD, em que um modelo preditivo baseado em RNA do tipo MLP obteve *f1-score* de aproximadamente 75%.

Uma abordagem baseada em *Deep Dictionary Learning* (TARIYAL et al., 2016) e *Deep Transform Learning* é proposta por Singhal, Maggu e Majumdar (2019), a qual é validada em relação aos conjuntos de dados REDD e Pecan Street. Apesar de ser uma abordagem inovadora, a mesma foi capaz de obter apenas 70,46% e 71,04% de *f1-score*, respectivamente. Vale comentar ainda que esses resultados foram obtidos ao considerar 5 cargas. Em relação à REDD (que é o conjunto de dados mais empregado como *benchmark*), os autores consideraram a rotulação de lava-louças, iluminação, lavadora/secadora de roupas e coifa/depurador de cozinha.

Embora a representação de imagem baseada em trajetória V-I tenha sido usada com sucesso para classificação de aparelhos em NILM, seu desempenho ainda deixa margem para melhorias pois essa representação bidimensional não é distinta o suficiente para reconhecer cargas que possuam consumo energético similar e/ou se enquadram na mesma categoria. Isso ocorre porque, nestes contextos, as trajetórias V-I acabam possuindo formas muito parecidas, independentemente da magnitude da corrente, o que degrada o desempenho dos modelos classificadores.

Como se sabe, uma eficiente representação de características causa um impacto positivo no processo de classificação de modelos de DL e, nessa tese, é proposta uma nova representação de características para classificação de cargas que se baseia em RPs, buscando analisar a dinâmica do sinal no espaço de fase para revelar os padrões repetidos e não lineares. Essa técnica tem sido usada extensivamente para representação de características em problemas de classificação de séries temporais em diferentes domínios (GARCIA-CEJA; UDDIN; TORRESEN, 2018; HATAMI; GAVET; DEBAYLE, 2018). Ao contrário da trajetória V-I, a representação do RP usa uma matriz de similaridade de distância para representar e visualizar padrões estruturais no sinal. Como consequência, a representação por RP também depende da magnitude dos sinais de corrente.

O uso de RP para a caracterização de cargas foi introduzido por Popescu et al. (2014) e vem recebendo mais atenção da comunidade de NILM desde então. Rajabi e Estebanari (2019), por exemplo, usaram RPs para realizar a desagregação de carga e empregaram uma CNN como modelo de classificação. Os autores verificaram que o uso de RP atinge um desempenho aproximadamente 20% superior para a métrica RMSE quando comparado a uma metodologia similar que utiliza o sinal original unidimensional como entrada para a CNN. Faustine e Pereira (2020) propuseram uma versão estendida do RP, chamada pelos autores de WRG (*Weighted Recurrent Graph*), a qual produz uma representação bidimensional que possui mais valores do que uma imagem binária produzida pelo RP canônico. Os autores argumentam que o *thresholding* aplicado pelo RP original causa perda de informação relevante na caracterização de cargas e, conseqüentemente,

degrada o processo de classificação. Usando uma CNN, os autores superam os resultados de modelos baseados em trajetória V-I para os conjuntos de dados COOLL, WHITED e PLAID, atingindo *f1-score* de 99,86%, 97,23% e 88,53%, respectivamente. Já Seon et al. (2021) propõem um método de imagem com lapso de tempo para classificar dispositivos. Utilizando um detector de razão de verossimilhança logarítmica com algoritmo máximo para eventos em tempo real, dados de consumo de energia são transformados em imagens usando GAF e RP, combinadas e inseridas em um modelo CNN-LSTM, com aumento de dados. Os autores destacam melhoria de até 30% na precisão da classificação, especialmente para aparelhos complexos como lava-louças e geladeira, em comparação com métodos convencionais de aprendizado profundo.

No que diz respeito à exploração do potencial de *transfer learning*, (D'INCECCO; SQUARTINI; ZHONG, 2020), (WANG et al., 2022) e (HOUIDI et al., 2021) fornecem contribuições significativas para sua aplicação no âmbito de NILM. O trabalho de D'Incecco, Squartini e Zhong (2020) destaca a viabilidade do uso de transferência de aprendizado, salientando que apenas as camadas totalmente conectadas requerem ajustes finos para otimização. Por sua vez, Wang et al. (2022) introduz a abordagem da assinatura de carga, representada pela trajetória V-I, promovendo a transferência de aprendizado. A eficácia do modelo pré-treinado é empiricamente demonstrada. Já o estudo de (HOUIDI et al., 2021) oferece novos resultados de *transfer learning*, ratificando a relevância e robustez das características selecionadas, as quais foram aprendidas em um conjunto de dados proposto e transferidas com sucesso para um conjunto de dados de maior escala. Em conjunto, essas pesquisas corroboram a importância e eficácia do *transfer learning* na tarefa crucial de classificação de cargas em NILM, proporcionando *insights* valiosos para aprimorar a precisão e a generalização dos modelos de desagregação de consumo de energia.

É relevante notar que, apesar do progresso significativo evidenciado pelos estudos anteriores, ainda existe uma lacuna perceptível na literatura em relação à quantidade limitada de pesquisas que exploram a adoção específica de *transfer learning* no monitoramento não invasivo de cargas. Assim, a escassez de trabalhos destaca a necessidade premente de investigações adicionais para preencher essa lacuna, a fim de aprofundar a compreensão e expandir o escopo de aplicação dessa abordagem inovadora no contexto da classificação de cargas em NILM.

## 3.2 Lacunas de Pesquisa

Decorrente da revisão da literatura descrita anteriormente, as seguintes lacunas de pesquisa promissoras puderam ser elencadas:

1. Em geral, NILM é abordado como um problema de regressão onde um modelo é treinado de modo supervisionado para prever o consumo individual da carga,

produzindo um número de valor real. No entanto, a maioria dos eletrodomésticos são conhecidos por terem estados de consumo discretos, onde a desagregação pode ser tratada como tarefa de classificação. Neste caso, o objetivo torna-se prever os estados da carga em vez de seus perfis de consumo;

2. A maioria dos estudos utiliza abordagens que consideram informações do sinal de corrente e tensão, componentes harmônicos ou trajetórias V-I para resolução de NILM. Entretanto, trabalhos recentes indicam que RPs podem ser uma forma de representação da informação eficiente para o consumo agregado e individual, uma vez que permite realizar uma caracterização mais precisa das cargas;
3. A transferência de aprendizado e a capacidade de generalização dos modelos, relevante para casos reais de aplicação de soluções de NILM — haja vista a diversidade presente dentro e entre residências, tanto no que diz respeito aos padrões de uso quanto à disponibilidade de cargas — é deixada em segundo plano pela maioria dos trabalhos, os quais focam essencialmente em otimizar uma métrica estatística de acordo com um conjunto específico de dados.

Portanto, esta pesquisa visa preencher estas lacunas de pesquisa através da metodologia conduzida no capítulo subsequente.

---

## Capítulo 4

# Metodologia Proposta

---

Nesse capítulo serão apresentados os aspectos detalhados da metodologia proposta para a resolução do problema de classificação de cargas, através da combinação de RP, DL e *Transfer Learning*, no contexto de NILM.

### 4.1 Preparação dos Dados

O conjunto de dados REDD, utilizado nesta tese, é dividido em baixas e altas frequências, contando com medições de seis residências. Todas as medições são organizadas em arquivos que contêm a estampa de tempo e energia consumida. Somente os dados de baixa frequência são utilizados, os quais foram coletados a uma taxa amostral de 1 Hz no painel principal de cada residência. Vale mencionar que, no contexto desta tese, foram considerados os dados da residência #3, uma vez que tais dados são os mais utilizados na literatura.

Seguindo a abordagem proposta em Kyrkou, Nalmpantis e Vrakas (2019), os dados foram ainda separados em conjuntos de treinamento (de 16 de abril de 2011 a 16 de maio de 2011) e validação (de 17 a 30 de maio de 2011). Na sequência, as séries temporais de treinamento e validação foram processadas usando janelas deslizantes e, portanto, a definição do tamanho da janela é um problema importante de ser resolvido no contexto de NILM, visto que janelas pequenas podem resultar na perda de informação útil para caracterizar determinadas cargas, enquanto janelas grandes podem incluir ruídos e/ou informações redundantes. Ainda, associado ao tamanho da janela deslizante, deve-se averiguar a melhor taxa amostral, ou seja, a combinação que resulte no melhor desempenho de modelos preditivos.

Por este motivo, ainda que de forma empírica, buscou-se obter a melhor relação entre o tamanho da janela deslizante e a taxa amostral adequada à classificação, visto que serão considerados modelos preditivos especializados na identificação de cada carga. Neste sentido, ressalta-se que, a princípio, foram consideradas as seguintes cargas: refrigerador, lava-louças, lavadora/secadora de roupas #1, lavadora/secadora de roupas #2 e micro-ondas. Essas cargas foram escolhidas em função da energia total consumida no período que compreende o conjunto de dados. Ou seja, foram escolhidas as cargas com a maior representatividade no consumo de energia referente à residência #3. A metodologia utilizada para obter o melhor *trade-off* entre o tamanho da janela e a taxa amostral para cada aparelho foi implementada com base nas seguintes etapas:

1. Define-se um conjunto  $J = [30, 60, 90, 180, 360, 540, 720, 900, 1080]$  de possíveis tamanhos de janelas e um conjunto  $T = [2, 3, 4, 5]$  de possíveis valores de taxa amostral;
2. Para cada combinação  $[J_i, T_i]$ :
  - a) Performa a reamostragem dos dados com base em  $T_i$ ;
  - b) Gera-se as janelas de comprimento  $J_i$ , e seus respectivos rótulos de ativação (ON/OFF), tanto na base de treinamento quanto na de validação;
  - c) Para cada aparelho;
    - i. Treina-se uma CNN (CAVALCA; FERNANDES, 2020);
    - ii. Avalia-se as métricas de interesse (Acurácia e F1-score);
    - iii. Arquiva o desempenho em uma matriz de resultados  $M$ ;
3. Extrai a melhor combinação  $[J_i, T_i]$  para cada aparelho.

Após a execução desta rotina, a melhor taxa amostral global obtida foi de 2 segundos, sendo esta a adotada. No que tange o tamanho das janelas, a Tabela 2 resume os melhores valores individuais obtidos. Desse modo, espera-se ser possível garantir um ponto de partida para uma melhor representação do ciclo de operação das cargas.

Após o janelamento das séries temporais, de ambos os conjuntos de dados, cada janela pôde ser rotulada para que *a posteriori* os modelos preditivos fossem treinados e validados. No caso das janelas provenientes da REDD, houve a necessidade de relacionar os dados das medições feitas no painel principal da residência com os dados adquiridos de medições individuais de cada carga. Essa relação foi estabelecida por meio das estampas de tempo, tornando-se possível gerar um vetor binário de cinco dimensões, onde cada posição representa determinada carga. Nota-se, portanto, que este vetor binário ainda possibilita que uma mesma janela de dados contenha simultaneamente a rotulação de mais do que uma carga.

Tabela 2 – Tamanho da janela deslizante definida inicialmente para melhor caracterização do ciclo de operação das cargas (*taxa amostral = 2 segundos*).

Carga	Tamanho da Janela
Refrigerador	1080
Lava-louças	90
Máquina de lavar/secar roupas #1	60
Máquina de lavar/secar roupas #2	60
Micro-ondas	30

Fonte: Autor.

## 4.2 Ambiente Computacional

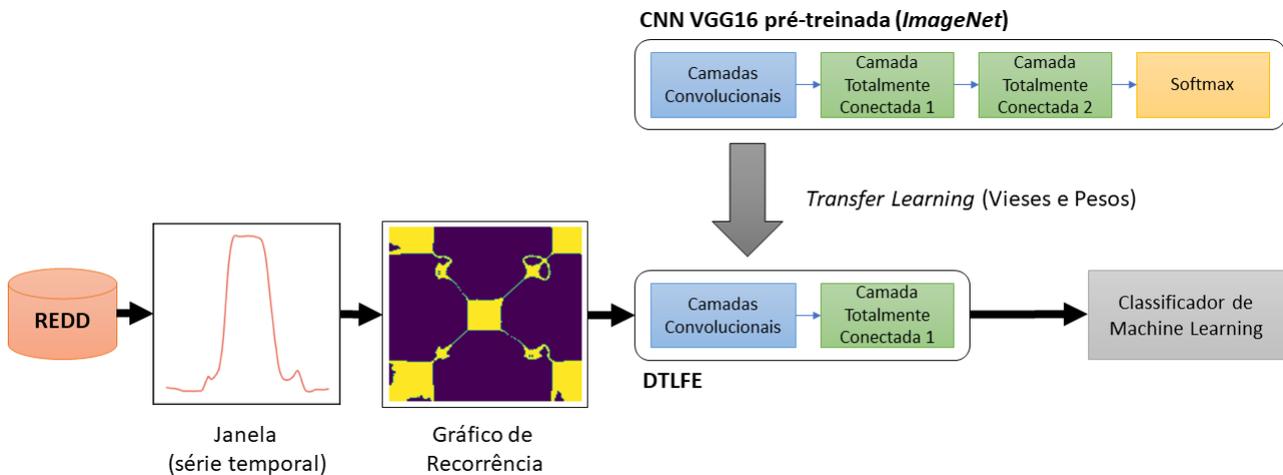
Visando garantir posterior disponibilização à comunidade científica, o framework tem sido implementado em linguagem Python 3, sendo que todo o processamento é realizado em um computador do tipo desktop com Intel Core i7-7700 CPU (3.60GHz), 16GB DDR4 RAM e uma placa gráfica (GPU, do inglês *Graphic Processing Unit*) NVIDIA Quadro M5000, a qual foi doada pela empresa NVIDIA® ao Laboratório de Inteligência Artificial Aplicada da UFSCar por meio de seu programa de apoio à pesquisa acadêmica na área de Inteligência Artificial. Cabe comentar que o uso da GPU aprimora consideravelmente o desempenho de sistemas baseados em redes neurais profundas, dada sua capacidade de processamento paralelo, factível a modelos de aprendizado de máquina conexionistas, classe na qual se enquadram as técnicas de DL.

Em tempo, dado o grau de inovação da abordagem discutida para a manipulação de séries temporais e o anseio pela reprodutibilidade que rege a pesquisa acadêmica, o framework denominado PyNILM será disponibilizado à comunidade científica. Portanto, tem-se como um dos produtos desta tese uma ferramenta de código aberto desenvolvida para fomentar outros projetos de pesquisas, potencializar novas descobertas e ampliar a visibilidade dessa pesquisa.

## 4.3 Framework Proposto

Uma visão geral da metodologia proposta nesta tese para a classificação de cargas no contexto de NILM é apresentada na Figura 17. De modo geral, a metodologia proposta se baseia em 3 estágios (módulos) fundamentais, sendo estes: i) transformação da série temporal em imagem; ii) extração de atributos (características); e iii) classificação/rotulação. Na sequência, são apresentados os detalhes de implementação de cada um destes módulos.

Figura 17 – Visão Geral da Metodologia Proposta.



Fonte: Autor.

### 4.3.1 Módulo de Transformação da Série Temporal em Imagem

Visto que este projeto se propõe a utilizar CNNs como modelos preditivos para NILM, o passo inicial se concentra na transformação das séries temporais (janelas) em imagens 2D. Neste sentido, foram gerados os RPs, os quais são representados por matrizes  $L \times L$ , dado que  $L$  é o tamanho da janela de dados.

### 4.3.2 Módulo de Extração de Características

Este módulo compreende o procedimento de extração automática de características, o qual se baseia no uso de uma CNN com arquitetura VGG16 (SIMONYAN; ZISSERMAN, 2015), a qual foi previamente treinada utilizando o conjunto de dados conhecido por *ImageNet* (DENG et al., 2009). Tal escolha é um ponto de partida comum a aplicações dentro da área de visão computacional. Na sequência, a última camada classificatória desta VGG16 é removida, de forma que a saída passa a ser um vetor de atributos (*features*) produzido pelas camadas anteriores em vez da rotulação dos dados. Portanto, esse procedimento que resulta no vetor de características é denominado na presente tese por DTLFE (*Deep Transfer Learning Feature Extraction*), o qual permite extrair automaticamente *features* dos RPs e usá-las como entradas de algoritmos clássicos de aprendizado de máquina, compreendidos no terceiro e último módulo do *framework*.

Como método comparativo para a etapa de extração de características, foram adotadas *features* calculadas com base na abordagem de análise de quantificação de recorrência (RQA, do inglês *Recurrence Quantification Analysis*), um método de análise de dados não linear que quantifica o número e a duração das recorrências de um sistema dinâmico apresentado por sua trajetória no espaço de estados (ZBILUT; WEBBER, 2006). Portanto,

a partir dos RPs, serão extraídos os indicadores de Taxa de Recorrência (RR, do inglês *Recurrence Rate*) e Determinismo (DET). O RR pode ser obtido conforme a equação 9:

$$RR = \frac{1}{N^2} \sum_{i,j=1}^N R_{i,j}, \quad (9)$$

em que  $R_{i,j}$  é a matriz resultante dos RPs e  $N$  é o número de estados. Ao obter RPs, o tempo de recorrência de uma série temporal pode ser determinado pelo comprimento da linha. Para medir essa característica, tem-se o *DET*, o qual pode ser obtido pela Equação 10:

$$DET = \frac{\sum_{l=l_{\min}}^N lP(l)}{\sum_{l=1}^N lP(l)}, \quad (10)$$

sendo  $P(l)$  o histograma dos comprimentos  $l$  das linhas diagonais dos RPs.

### 4.3.3 Módulo de Classificação

Por fim, o módulo de classificação recebe os vetores de *features* extraídos a partir dos RPs através do DTLFE e realiza a classificação binária dos estados das cargas. Nesse estudo são avaliadas distintas abordagens de ML, sendo elas: MLP, SVM e XGBoost. Todas foram escolhidas empiricamente e serão discutidas de forma breve na sequência.

#### 4.3.3.1 MLP

Conforme já discutido nesta tese, as RNAs do tipo MLP têm sido empregadas no contexto de NILM devido à sua capacidade de generalização e precisão na rotulação de cargas residenciais. O modelo de MLP utilizado nesse projeto consiste em 10 camadas ocultas e 1 camada de saída. Além disso, as funções de ativação das camadas ocultas e da camada de saída são, respectivamente a linear retificada (ReLU, do inglês *rectified linear unit*) e *softmax*. A principal razão para utilizar ReLU é que a mesma permite contornar o problema de *gradient vanish*, conforme demonstrado por Glorot, Bordes e Bengio (2011).

Vale mencionar que foi empregado o algoritmo de treinamento *backpropagation*, pois tem sido comumente aplicado em NILM (ROOS et al., 1994; MEDEIROS et al., 2019).

#### 4.3.3.2 SVM

O SVM se baseia no princípio de minimização do risco estrutural. Ademais, tal algoritmo estima uma função que reduz o erro de generalização, mostrando-se resistente ao problema de *overfitting*. Ainda é válido mencionar que, diferente da MLP, o algoritmo de SVM não é estocástico, ou seja, se não houver alterações no conjunto de dados, também não haverá modificações nos resultados.

Basicamente, o SVM busca criar um hiperplano que separe as classes do problema. Visto que cada amostra de dados em cada lado do hiperplano possui uma distância até

ele, a menor distância será a margem de separação. Assim, pode-se dizer que o hiperplano é ótimo quando a margem de separação é maximizada. Neste sentido, o estágio de treinamento de uma SVM consiste em determinar um hiperplano ótimo (DUDA; HART; STORK, 2012). Entretanto, esse processo de otimização é lento e demanda grande esforço computacional. Portanto, é comum o uso de uma flexibilização da margem de separação, a qual estabelece um nível de tolerância para aceitar amostras de dados que estejam fora do limite do hiperplano (BISHOP, 2006).

Nos casos em que os dados não são linearmente separáveis (o que ocorre em NILM), o teorema de Cover é usado, que sugere aumentar a dimensionalidade dos dados por meio do uso de *kernels*, permitindo conseguir separá-los linearmente (BISHOP, 2006). Por este motivo, neste projeto foram utilizados algoritmos de SVM com *kernel* de base radial.

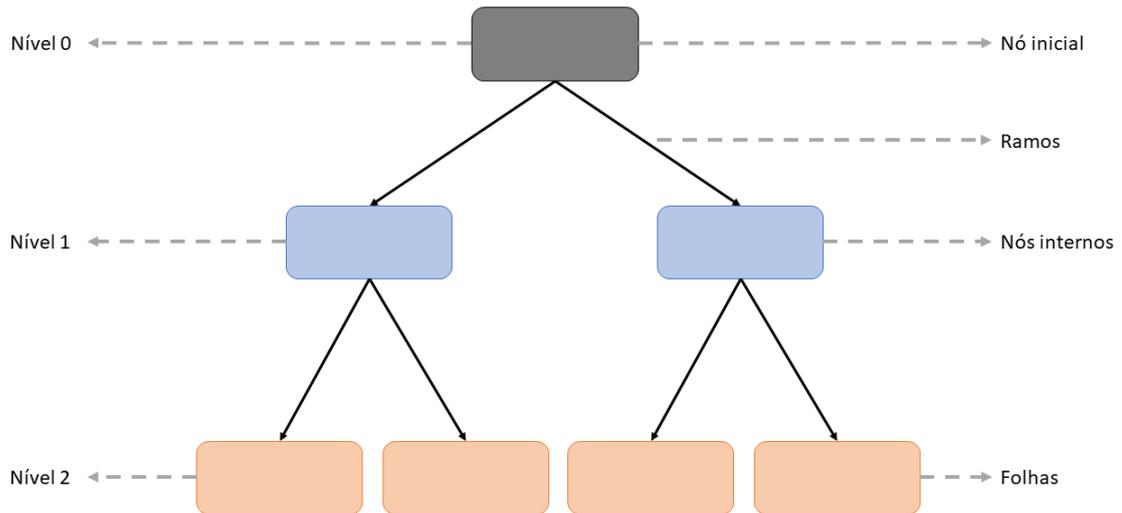
### 4.3.3.3 XGBoost

Proposto por Chen e Guestrin (2016), o *Extreme Gradient Boosting* (XGBoost) é uma biblioteca de aprendizado de máquina que implementa de modo eficiente a construção de modelos baseados em árvores de decisão utilizando a estratégia de *gradient boosting trees*, onde é realizado o agrupamento de modelos de árvore de decisão simples de modo que o modelo final seja mais geral e robusto.

As árvores de decisão são modelos amplamente utilizados em tarefas de classificação e regressão, nos quais o aprendizado destes se concentra na segmentação iterativa dos dados em regiões (conjuntos de variáveis explicativas) cada vez menores e mais específicas até atingirem um tamanho simplificado o bastante para ser rotulada, geralmente utilizando tendências centrais (geralmente média ou moda) da variável resposta, de modo que cada região construída diminua uma determinada medida de erro (por exemplo, a raiz do erro quadrático médio) (QUINLAN, 1986). Conforme é possível observar na Figura 18, uma árvore é composta por um nó inicial (raiz), nós internos, ramos e folhas. De modo geral, cada nó avalia um atributo e cada percurso na árvore (da raiz até a folha) é uma regra de classificação.

Apesar de fornecer um modelo com alto grau de interpretabilidade, árvores de decisão apresentam baixa capacidade preditiva e não são utilizadas em conjuntos de dados complexos. A fim de potencializar o aprendizado desta classe de modelos, o algoritmo de Floresta Aleatória (do inglês, *Random Forest*) permite construir um modelo robusto a partir da combinação de múltiplos modelos mais fracos. Para isto, este algoritmo inicialmente organiza o conjunto de dados original em diferentes subconjuntos (tanto a nível de amostras quanto de atributos) e realiza o treinamento de árvores individuais dentro de cada um, combinando-os ao final, de modo que a média das predições individuais sejam consideradas como a estimativa do modelo final. De modo geral, este algoritmo toma como base a estratégia de *bootstrapping aggregation*, ou *bagging*, a fim de reduzir a variância das estimativas, uma das principais fraquezas da árvore de decisão.

Figura 18 – Diagrama da estrutura fundamental de uma árvore de decisão.



Fonte: Autor.

Além do *bagging*, uma alternativa é o algoritmo *boosting* que estabelece a construção de múltiplas árvores de modo sequencial, sendo que cada árvore utiliza informações extraídas da árvore anterior (sua progenitora), ou seja, ao invés de construir uma única grande árvore a partir do conjunto de dados original, são construídas  $B$  árvores com poucos nós intermediários, de forma que a árvore  $\hat{f}^i(x)$  é ajustada utilizando resíduos da árvore  $\hat{f}^{i-1}(x)$ , tal que, a cada iteração  $i \in \{1, \dots, B\}$ , a função estimada do modelo é dada por  $\hat{f}(x) = \hat{f}(x) + \lambda * \hat{f}^i(x)$ , sendo inicialmente  $\hat{f}(x) = 0$  e  $\lambda$  o parâmetro encolhimento (onde  $\lambda > 0$ ). Já o *gradient boosting* é uma variação do algoritmo *boosting*, o qual adota a minimização de uma função de custo  $L(y, f(x))$  em detrimento dos resíduos (FRIEDMAN, 2001). Assim, em cada etapa, são calculados os gradientes da função custo  $g_j$  em relação a cada árvore  $\hat{f}^i(x)$ , tal que:

$$g_i = \left[ \frac{\delta L(y, f(x_i))}{\delta f(x_i)} \right]_{f=f^{i-1}}, \quad i = 1, \dots, n. \quad (11)$$

Por fim, para atingir eficiência computacional e alto desempenho, os autores do XG-Boost implementaram diversas estratégias, dentre as quais destaque-se a penalização das árvores, visando controlar a velocidade de aprendizado, a aleatorização de parâmetros para diminuir a correlação entre as árvores que compõem o modelo e o encolhimento dos nós folha para controlar o *overfitting* do modelo final.

## 4.4 Métricas de Avaliação de Desempenho

A métrica mais utilizada pela literatura para mensurar o desempenho dos modelos preditivos na área de NILM é a Acurácia (*acc*, do inglês *accuracy*), a qual é dada por:

$$acc = \frac{TP + TN}{TP + FP + TN + FN} \times 100, \quad (12)$$

sendo TP e FP os verdadeiros e falsos positivos, enquanto TN e FN são os verdadeiros e falsos negativos, respectivamente. Entretanto, em problemas de classificação binária, geralmente, há a necessidade de se evitar o viés que pode ser gerado por conta da classe majoritária. Para tanto, faz-se necessário calcular o *f1-score* que representa uma relação entre *Precision* e *Recall*, conforme mostrado na sequência:

$$Precision = \frac{TP}{TP + FP}, \quad (13)$$

$$Recall = \frac{TP}{TP + FN}, \quad (14)$$

$$f1score = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \quad (15)$$

Portanto, uma vez que a distribuição das classes do problema de classificação de cargas pode conter forte desbalanceamento, apesar de reportar a acurácia durante a apresentação dos resultados, a métrica *f1-score* será aquela de maior relevância e utilizada como padrão para a comparação dos métodos.

---

## Capítulo 5

# Resultados

---

Com o objetivo de oferecer uma visão abrangente sobre a classificação de cargas no contexto do NILM, esta seção inicia com uma análise dos trabalhos discutidos anteriormente na seção Revisão Bibliográfica. Esta análise se concentra especificamente nos estudos que fazem uso do conjunto de dados REDD, o mesmo adotado nesta tese. Assim, um resumo comparativo acerca do desempenho destes trabalhos está estruturado na Tabela 3. Em tempo, é oportuno mencionar que a maioria destes trabalhos omite detalhes importantes de suas respectivas metodologias, principalmente em questões relacionadas à implementação e experimentação — como período amostral utilizado para treino/validação, pré-processamento realizado no sinal, etc..

De forma breve, destaca-se (MORADZADEH et al., 2020), que utilizam a série do consumo agregado e o método de redução de dimensionalidade *PCA*, alcançando um notável *f1-score* de 94,7%. (HUA et al., 2021) adotam a série do consumo agregado e o método *CUSUM*, atingindo um *f1-score* de 93,25%. Já (CORREA; CASTRO, 2020) utilizam detecção de eventos e *NN Embedding* com um MLP, obtendo um *f1-score* de 85,15%. Por fim, é interessante destacar o trabalho de (KYRKOU; NALMPANTIS; VRAKAS, 2019), que convertendo a série de consumo agregado em imagens (GADF), e utilizando CNN e MLP em combinação para o processo classificatório, alcançam aproximadamente 75% de *f1-score*.

Essa variedade metodológica evidencia a complexidade da tarefa de classificação de cargas em NILM e sugere a importância de considerar diferentes estratégias dependendo das características específicas do conjunto de dados e do contexto de aplicação.

Na sequência, adentramos a discussão aprofundada dos resultados obtidos nesta pesquisa, apresentando detalhes sobre o desempenho geral do framework proposto, bem como investigações sobre sua capacidade de generalização e discussões sobre diferentes perspec-

Tabela 3 – Comparação de estudos que utilizam o conjunto de dados REDD para o processo de identificação de cargas no contexto de NILM.

Referência	Atributos	Classificador	Taxa Amostral	Cargas	Janela	<i>f1-score</i> (%)
(MORADZADEH et al., 2020)	Série do Consumo Agregado	PCA	3s	33	48h	94.7
(HUA et al., 2021)	Série do Consumo Agregado	CUSUM	3s	8	-	93.25
(MACHLEV; LEVRON; BECK, 2018)	Série do Consumo Agregado	MCE	4s	13	-	86.0
(CORREA; CASTRO, 2020)	Detecção de Eventos, <i>NN Embedding</i>	MLP	3s	6	-	85.15
(KYRKOU; NALMPANTIS; VRAKAS, 2019)	GADF	CNN, MLP	6s	1	6.4min	75
(SINGHAL; MAGGU; MAJUMDAR, 2019)	Série do Consumo Agregado	DL	1min	4	60min	70.2
(SINGH; MAJUMDAR, 2019)	Série do Consumo Agregado	ML-SCR	10min	4	-	65.4
(ZHOU et al., 2021)	Série do Consumo Agregado	PNN	1s	4	3s - 100s	56.75
(NALMPANTIS; VRAKAS, 2020)	Signal2Vec	MLP	6s	13	10min - 24h	48.2

Fonte: Autor.

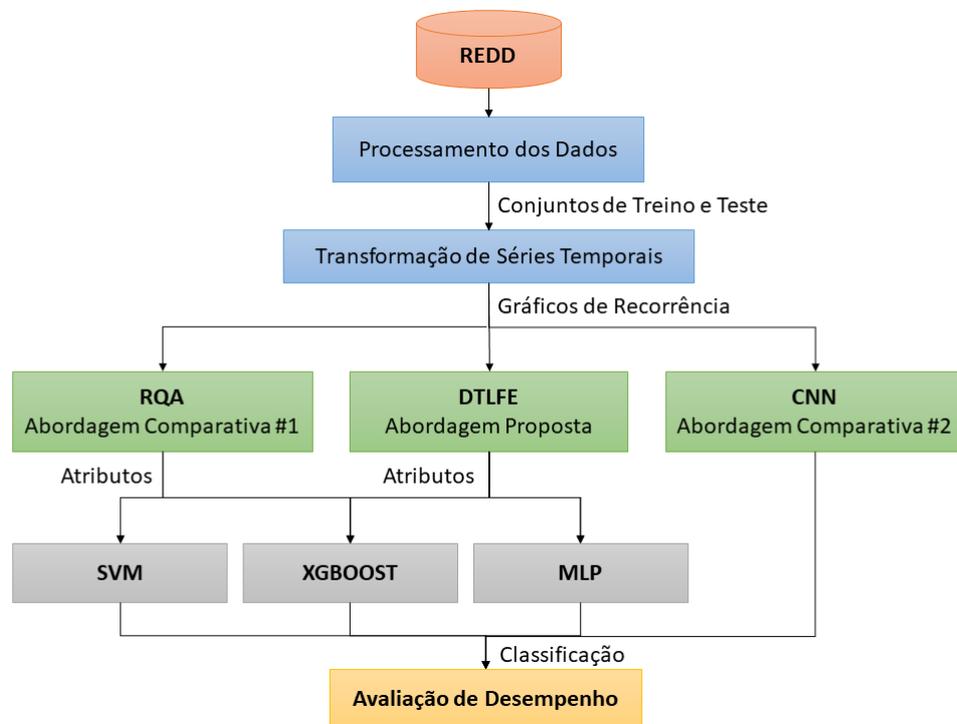
tivas de otimização de seus componentes.

## 5.1 Classificação de Cargas

No contexto de classificação das cargas selecionadas da residência #3 da base REDD, todos os experimentos foram realizados a partir da divisão do conjunto de dados em lotes de treinamento e teste, conforme detalhado na seção 4.1. Em tempo, vale ressaltar que uma abordagem *end-to-end* foi considerada como hipótese adicional para classificação de cargas, em que uma CNN — cuja arquitetura foi definida empiricamente — realiza a classificação direta a partir de um RP dado como entrada. Na sequência, serão apresentados os resultados e análises para: i) o desempenho geral, comparando as duas outras abordagens destacadas na Figura 19; ii) a generalização da abordagem proposta; e iii)

ajuste fino dos tamanhos de janelas por carga e seus impactos nas métricas de avaliação de desempenho.

Figura 19 – Diagrama da análise comparativa conduzida neste estudo.



Fonte: Autor.

### 5.1.1 Análise Geral de Desempenho

Com base nos pressupostos supracitados, o desempenho geral dos classificadores pode ser observado por meio da Tabela 4.

Tabela 4 – Desempenho geral dos classificadores avaliados.

<i>Features</i>	Classificador	Acurácia (%)	<i>f1-score</i> (%)
DTLFE	MLP	97,5	75,7
DTLFE	XGBoost	97,9	74,3
DTLFE	SVM	97,7	71,9
RP	CNN	95,7	68,8
RQA	SVM	95,7	53,4
RQA	XGBoost	95,1	52,3
RQA	MLP	95,7	48,9

Fonte: Autor.

Ao ordenar os resultados por *f1-score*, conforme mostrado na Tabela 4, nota-se que a extração automática de *features* em conjunto com a MLP possui um desempenho superior no cenário avaliado, compreendendo uma melhora que varia entre 7% e 26,8% em relação às outras abordagens avaliadas. Entretanto, ainda se faz importante averiguar o comportamento de cada abordagem de extração de *features* em torno da classificação individual das cargas residenciais. Tais resultados podem ser verificados por meio das Tabelas 5, 6 e 7.

Tabela 5 – Desempenho do DTLFE para a classificação individual de cargas.

Carga	Classificador	Acurácia (%)	<i>f1-score</i> (%)
Lava-louças	MLP	96,8	55,9
Refrigerador	MLP	98,2	63,8
Micro-ondas	MLP	96,3	81,4
Lavadora/Secadora de roupas #1	XGBoost	99,8	98,5
Lavadora/Secadora de roupas #2	SVM	98,8	94,0
<b>Média</b>	-	<b>98,0</b>	<b>78,7</b>

Fonte: Autor.

Tabela 6 – Desempenho do RR + DET para a classificação individual de cargas.

Carga	Classificador	Acurácia (%)	<i>f1-score</i> (%)
Lava-louças	MLP	96,8	49,2
Refrigerador	SVM	98,2	71,7
Micro-ondas	XGBoost	91,7	54,3
Lavadora/Secadora de roupas #1	XGBoost	96,0	50,5
Lavadora/Secadora de roupas #2	XGBoost	93,1	58,2
<b>Média</b>	-	<b>95,1</b>	<b>56,7</b>

Fonte: Autor.

Ao analisar as Tabelas 5 e 6, é possível notar que o uso de modelos preditivos individuais, ou seja, que empregam o uso de classificadores especializados na classificação de cada carga, aprimora os resultados, observando-se que a média de *f1-score* para o uso de DTLFE foi de 78,7%, ou seja, 3% maior do que aquele resultado apresentado na Tabela 4. Incremento ainda maior pôde ser verificado para o uso de RR e DET como *features* de entrada dos modelos preditivos, em que o ganho foi de ao menos 3,3%.

Estes resultados demonstram que o *framework* proposto é eficiente até mesmo diante do desbalanceamento inerente aos dados de operação das cargas (rótulos ON e OFF, que

Tabela 7 – Desempenho da abordagem *end-to-end* (*features* = RP e classificador = CNN ) para a classificação individual de cargas.

Carga	Acurácia (%)	<i>f1-score</i> (%)
Lava-louças	95,3	53,6
Refrigerador	98,5	74,6
Micro-ondas	92,9	63,3
Lavadora/Secadora de roupas #1	95,9	72,4
Lavadora/Secadora de roupas #2	95,6	56,7
<b>Média</b>	<b>95,6</b>	<b>68,7</b>

Fonte: Autor.

indicam a utilização ou não das cargas dentro de cada janela deslizante) da base REDD. A Tabela 8 consolida o desempenho do classificador MLP, utilizando as *features* DTLFE (DTLFE + MLP) para a classificação de cada carga e o número de amostras disponíveis para cada classe.

Tabela 8 – Desempenho do framework DTLFE + MLP para cada carga.

Carga	Amostras "ON"	Amostras "OFF"	Acurácia (%)	<i>f1-score</i> (%)
Lava-louças	16	623	96,8	55,8
Refrigerador	425	2	98,1	63,8
Micro-ondas	28	483	96,3	81,4
Lavadora/Secadora de roupas #1	187	7.353	99,2	94,6
Lavadora/Secadora de roupas #2	30	819	96,9	83,0

Fonte: Autor.

### 5.1.2 Ajuste Fino das Janelas Deslizantes para o DTLFE

Como dito anteriormente, um fator que afeta diretamente os resultados de classificação de cargas no contexto de NILM é o tamanho da janela, pois esta deve ser cuidadosamente ajustada para que não seja tão pequena a ponto de perder ciclos de operação de carga, e não tão larga para ter grande sobreposição com outras cargas. Com base neste preceito, até aqui, foi definida a melhor janela para cada aparelho com base na avaliação de um espaço discreto de possibilidades, conforme apresentado na seção 4.1. Porém, como explicado, esta definição tomou como base a adoção de uma CNN arbitrária, a qual foi treinada e avaliada diretamente a partir dos RPs, não fazendo uso dos benefícios encontrados na abordagem DTLFE. Partindo deste princípio, a hipótese é que, uma vez que

a combinação de *features* DTLFE e classificador MLP (DTLFE + MLP) apresentou os melhores resultados, a seleção otimizada das janelas deve levar em conta esta abordagem, seguindo a mesma estratégia já implementada anteriormente. Tal hipótese foi avaliada e seus resultados podem ser observados na Figura 20.

Tabela 9 – Comparando o *f1-score* diante da janela original e otimizada para a abordagem DTLFE + MLP (*taxa amostral = 2 segundos*).

Carga	Janela Original	Janela Otimizada
Lava-louças	90 (63,8%)	2040 (63%)
Refrigerador	1080 (55,8%)	720 (87%)
Micro-ondas	30 (83%)	900 (81%)
Lavadora/Secadora de roupas #1	60 (81,4%)	90 (96%)
Lavadora/Secadora de roupas #2	60 (94,6%)	2040 (90%)
<b>Média</b>	<b>75,7%</b>	<b>83,4%</b>

Fonte: Autor.

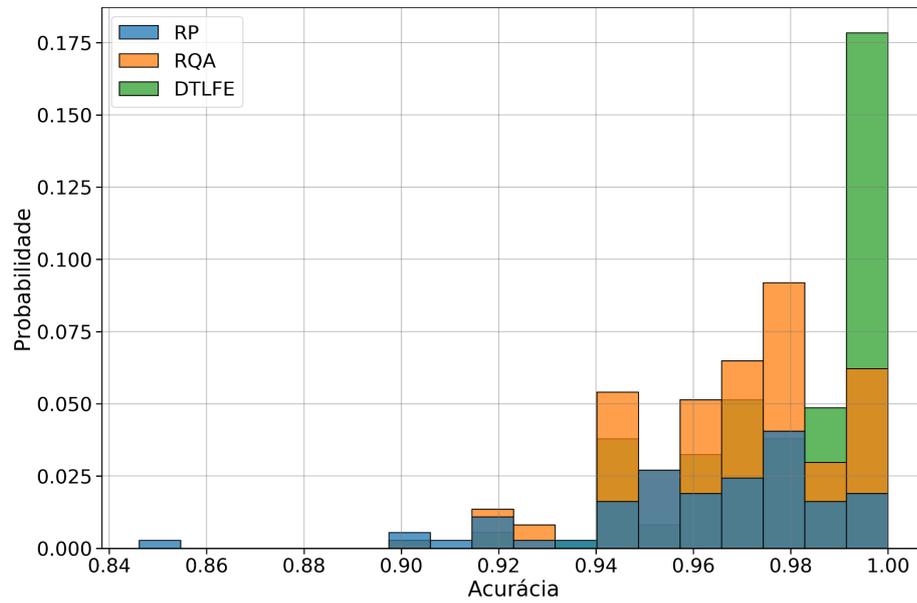
Conforme a Tabela 9, é possível observar que o ajuste-fino do tamanho da janela de acordo com a abordagem implementada tem relevante impacto no *f1-score*, uma vez que o desempenho do classificador DTLFE+MLP varia em uma faixa entre 1,4% e 23,2% para as cargas consideradas quando se compara as janelas definidas *a priori*, com base no classificador arbitrário, com as janelas otimizadas especificamente para esta abordagem. Além disso, de forma geral, houve aumento de 7,7% na utilização de janelas diferentes para cada uma das cargas.

## 5.2 Análise de Generalização

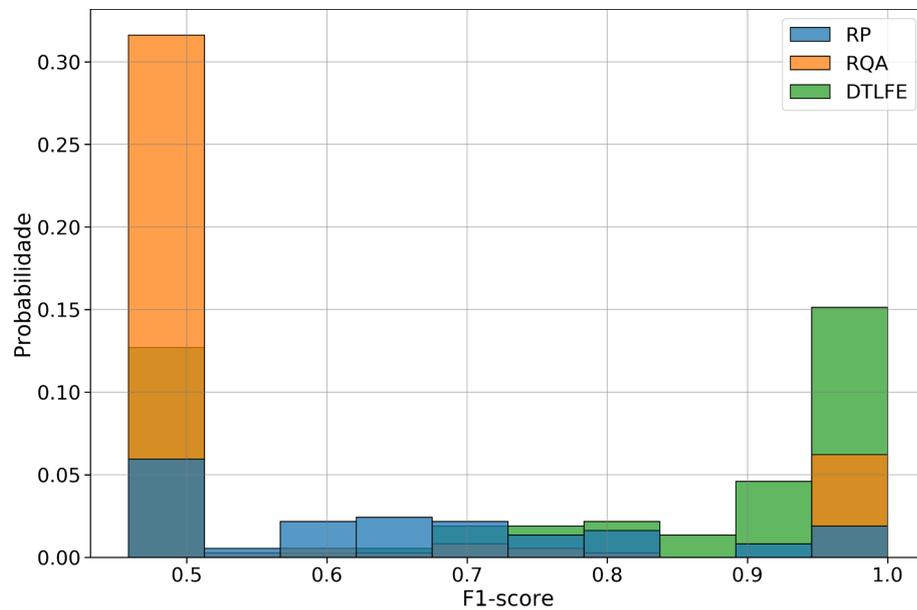
### 5.2.1 Validação Cruzada

Conforme já mencionado, o conjunto de teste foi definido a partir da delimitação de um período (de 17 a 30 de maio de 2011). Porém, para avaliar a generalização da aprendizagem do modelo MLP (empregada na melhor abordagem encontrada até então), este classificador foi submetido a um conjunto de testes gerado sem considerar o aspecto temporal, ou seja, em amostras arbitrárias. Para esta avaliação, foi adotada uma estratégia de validação cruzada de 10 vezes a partir do conjunto completo de dados, combinando os conjuntos de treinamento e teste definidos *a priori*. Assim, cada uma das abordagens de extração de características foi analisada. Cabe ressaltar que, no caso de usar RPs, um classificador baseado em CNN foi considerado em vez do MLP. As distribuições de probabilidade obtidas para as métricas Acurácia e *f1-score* são apresentadas respectivamente nas Figuras 21 e 22.

Figura 21 – Distribuição de probabilidade para Acurácia.



Fonte: Autor.

Figura 22 – Distribuição de probabilidade para  $f1$ -score.

Fonte: Autor.

De maneira geral, é possível observar que o DTLFE apresenta uma boa distribuição de probabilidade, indicando que este possui a melhor capacidade de extração de atributos que potencializam aprendizado e generalização entre as abordagens discutidas neste estudo. Com isso, é interessante notar que a abordagem usando características RQA fornece os resultados mais instáveis em termos de  $f1$ -score. Dessa forma, tais características

não fornecem informações suficientes para o treinamento da MLP e, conseqüentemente, dificultam sua generalização.

## 5.2.2 Capacidade de Generalização Inter-Residências

Dado seu destacado desempenho na classificação de cargas da residência #3, a avaliação da generalização da abordagem DTLFE + MLP em contextos residenciais não previamente vistos é crucial para determinar a aplicabilidade prática do sistema de monitoramento não invasivo de cargas. Os resultados obtidos ao aplicar a abordagem em diferentes residências e aparelhos podem fornecer *insights* sobre sua capacidade de adaptação a cenários diversos.

### 5.2.2.1 Análise Geral

Para este fim, foi conduzido um experimento que envolveu a combinação dos dados de quatro residências específicas (1, 2, 3 e 5), todas compartilhando os aparelhos de interesse: lava-louças, micro-ondas, refrigerador e lavadora/secadora de roupas. Essa escolha deliberada busca avaliar a capacidade do modelo em reconhecer padrões comuns a esses aparelhos em residências distintas.

Na amostragem de treino, foram consideradas todas as combinações possíveis dentro do conjunto de residências 1, 2, 3 e 5. Isso significa que o conjunto de treinamento pôde conter desde dados de uma única residência até a inclusão de três residências simultaneamente. Essa abordagem visa explorar a diversidade presente nos padrões de consumo dessas residências, proporcionando ao modelo uma exposição a diferentes contextos residenciais durante o treinamento. Para o conjunto de teste, uma das residências foi mantida independente das utilizadas no treinamento, garantindo uma avaliação justa da generalização do modelo em cenários não previamente vistos.

Os resultados obtidos ao avaliar a generalização da abordagem DTLFE + MLP para diferentes conjuntos de treinamento e teste, disponíveis na Tabela 10 destacam aspectos de adaptabilidade do modelo em contextos residenciais variados, tais como:

- ❑ **Alta Generalização da Residência #2** – o modelo apresentou consistentemente altos valores de *f1-score* em diversas configurações de conjuntos de treinamento. Isso sugere que o modelo treinado com dados dessa residência tem uma capacidade maior de generalização, mantendo um desempenho robusto mesmo quando testado em diferentes cenários residenciais. Esse resultado indica que os padrões de consumo específicos capturados durante o treinamento na residência #2 são transferíveis e reconhecíveis em outras residências;
- ❑ **Desafios na Generalização da Residência #1** – o desempenho do modelo variou significativamente de acordo com os conjuntos de treinamento. O melhor desempe-

Tabela 10 – Resultados da Avaliação de Generalização em Diferentes Residências.

Residência de Teste	Residência(s) de Treino	f1-score médio (%)	Média
1	[2]	61,2	52,4%
	[2, 5]	59,2	
	[2, 3, 5]	58	
	[5]	48,8	
	[2, 3]	46,8	
	[3]	46,8	
	[3, 5]	46,2	
2	[1, 3, 5]	86,8	83,1%
	[1, 5]	86,8	
	[1]	86,8	
	[5]	86,8	
	[3, 5]	86	
	[3]	77,2	
	[1, 3]	71,8	
3	[1, 2]	61,6	55,4%
	[1, 5]	59,4	
	[1]	59,2	
	[1, 2, 5]	57,6	
	[5]	52,6	
	[2, 5]	49,2	
	[2]	48,8	
5	[1]	65,4	59,6%
	[1, 3]	64,8	
	[3]	61,6	
	[1, 2, 3]	59,8	
	[2]	59,8	
	[2, 3]	57,2	
	[1, 2]	48,6	
		<b>Média Geral</b>	<b>62,6%</b>

Fonte: Autor.

nho foi observado ao treinar o modelo exclusivamente com dados da residência #2, indicando que, em alguns casos, utilizar informações de uma única residência para treinamento pode resultar em uma melhor generalização para a residência #1. No entanto, conforme mais residências foram incluídas no treinamento, o desempenho tendeu a diminuir. Isso sugere que a diversidade de padrões entre as residências 1 e 2 pode apresentar desafios para a generalização do modelo;

- **Impacto da Residência #3 no Treinamento** – ao treinar o modelo com tal residência, notou-se um impacto variável nos resultados. Em alguns casos, a inclusão da residência #3 no treinamento resultou em melhor desempenho, enquanto em outros casos, a exclusão levou a resultados superiores. Isso destaca a complexidade da influência específica de cada residência no desempenho do modelo, indicando que a interação entre diferentes conjuntos de dados de treinamento pode ser não linear

e dependente do contexto.

Essas observações ressaltam a importância de considerar cuidadosamente a composição dos conjuntos de treinamento ao avaliar a generalização do modelo em contextos residenciais diversos. Além disso, a consistência notável na generalização para a residência #2 sugere que o modelo pode beneficiar-se significativamente de dados específicos de treinamento provenientes dessa residência.

### 5.2.2.2 Análise por Aparelho

Prosseguindo com a investigação sobre a generalização do DTLFE + MLP em diferentes contextos residenciais, nessa análise o foco se volta a uma abordagem inter-aparelhos. Agora, é explorada uma perspectiva adicional, examinando o desempenho do modelo treinado na residência central desta pesquisa (Residência #3) ao inferir sobre o funcionamento de aparelhos específicos em outras residências (especificamente, #1, #2 e #5). Essa abordagem permite uma avaliação mais detalhada das capacidades de generalização do modelo, revelando sua consistência na identificação de padrões de consumo associados a aparelhos específicos em diferentes contextos residenciais. Cabe mencionar que a análise inter-aparelhos contribui para a ampliação da compreensão sobre a robustez do modelo diante de variações nos perfis de consumo de energia, fornecendo informações relevantes para futuros aprimoramentos e refinamentos na proposta apresentada. Os resultados podem ser observados na Tabela 11.

Tabela 11 – Desempenho do DTLFE + MLP Treinado na Residência #3 à nível de Aparelho nas Demais Residências.

Residência	Aparelho	f1-score (%)	Média
1	Lava-louças	43	46,8%
	Refrigerador	50	
	Micro-ondas	47	
	Lavadora/Secadora de roupas	47	
2	Lava-louças	100	77,2%
	Refrigerador	9	
	Micro-ondas	100	
	Lavadora/Secadora de roupas	100	
5	Lava-louças	100	61,6%
	Refrigerador	64	
	Micro-ondas	47	
	Lavadora/Secadora de roupas #1	48	
	Lavadora/Secadora de roupas #2	49	

Fonte: Autor.

Ao analisar os resultados, fica evidente o notável desempenho obtido na residência #2, onde o DTLFE + MLP obteve *f1-score* médio de 77,2%. Essa superioridade pode ser atribuída a diversos fatores que merecem consideração.

Primeiramente, é relevante destacar possíveis semelhanças ou características compartilhadas entre as residências 2 e 3, onde o modelo foi inicialmente treinado. Ou seja, similaridades significativas nos padrões de consumo de energia, disposição de aparelhos, ou até mesmo na distribuição temporal de eventos, podem fazer o modelo beneficiar-se da transferência de conhecimento, contribuindo para uma adaptação mais eficaz.

Além disso, a composição específica dos aparelhos na residência #2 pode influenciar positivamente o desempenho do modelo. Por exemplo, se os aparelhos presentes nessa residência forem representativos dos tipos de carga para os quais o modelo foi treinado, isso pode resultar em uma identificação mais precisa e, conseqüentemente, em métricas de desempenho mais elevadas.

Outro ponto relevante a considerar é a consistência nos padrões de consumo de energia ao longo do tempo na residência #2. Se essa consistência for mais evidente em comparação com outras residências, o modelo pode ter uma tarefa facilitada na identificação de padrões, contribuindo para seu desempenho superior.

Ademais, questões relacionadas à qualidade e quantidade dos dados também desempenham um papel crucial. Neste caso, possuindo a residência #2 um conjunto de dados mais abrangente, equilibrado e representativo, isso pode resultar em um treinamento mais eficaz do modelo, o que pode ajudar explicar a melhor capacidade de generalização para este contexto específico.

A interação complexa desses fatores pode ser a chave para compreender o desempenho superior do modelo na residência #2 frente às demais. Investigar mais detalhadamente esses elementos pode proporcionar o aprimoramento da aplicabilidade prática do sistema em ambientes residenciais diversos.

Cabe destacar que a limitação imposta pela anonimização dos dados no conjunto REDD impede uma investigação mais aprofundada. Todavia, para enriquecer a compreensão sobre os fatores que contribuem para o desempenho diferenciado do modelo em diferentes contextos residenciais, seria vantajoso conduzir uma análise similar em conjuntos de dados públicos que disponibilizem informações mais detalhadas sobre o perfil de consumo das residências. Esta abordagem, embora esteja além do escopo desta tese, representa um caminho promissor para pesquisas futuras, possibilitando uma validação mais robusta das hipóteses levantadas.

## 5.3 Diferentes Arquiteturas Profundas para Extração de Características

Uma vez que o framework DTLFE apresentou resultados significativos na classificação de cargas discutida até aqui, outra análise relevante é a investigação do impacto da escolha de diferentes arquiteturas profundas para realizar a extração de características dos RPs. Portanto, foram selecionadas arquiteturas previamente treinadas no conjunto

de dados ImageNet, as quais foram avaliadas em termos de desempenho geral ( $f1$ -score), comparando os resultados obtidos com a abordagem anteriormente citada (VGG16). A seguir, são apresentados os experimentos e as análises correspondentes.

### 5.3.1 Experimentação

Os experimentos foram conduzidos considerando as diferentes arquiteturas profundas disponibilizadas na biblioteca Tensorflow, listadas na Tabela 12. Conforme os dados e contextos da abordagem apresentados na seção 4.1, para cada arquitetura, a última camada classificatória foi removida a fim de obter um vetor de características, que é usado para a classificação de cargas utilizando os 3 classificadores já apresentados (SVM, MLP e XGBoost).

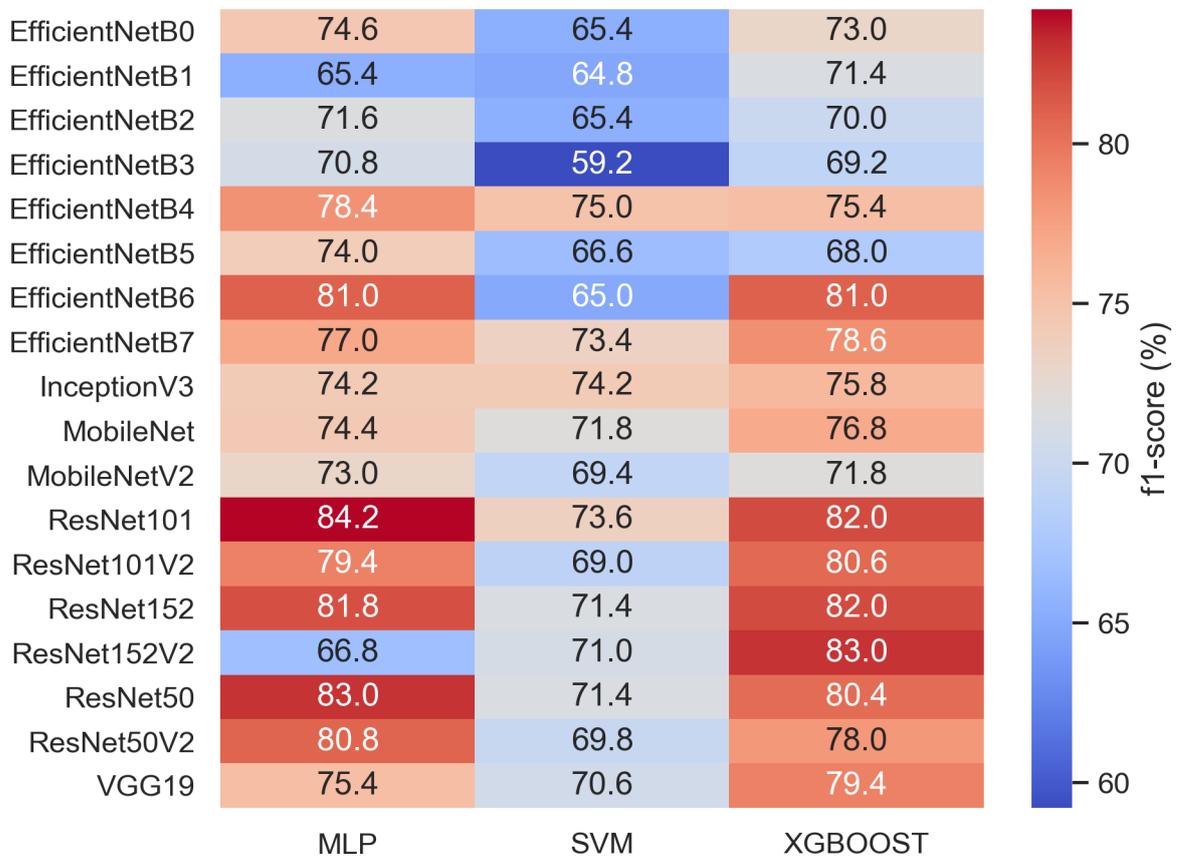
Tabela 12 – Arquiteturas pré-treinadas (ImageNet) Avaliadas no Processo de Extração de Características

Nome do modelo	Total de neurônios	Profundidade
EfficientNetB0	5.3M	132
EfficientNetB1	7.9M	186
EfficientNetB2	9.2M	186
EfficientNetB3	12.3M	210
EfficientNetB4	19.5M	258
EfficientNetB5	30.6M	312
EfficientNetB6	43.3M	360
EfficientNetB7	66.7M	438
InceptionV3	23.9M	189
MobileNet	4.3M	55
MobileNetV2	3.5M	105
ResNet50	25.6M	107
ResNet50V2	25.6M	103
ResNet101	44.7M	209
ResNet101V2	44.7M	205
ResNet152	60.4M	311
ResNet152V2	60.4M	307
VGG19	143.7M	19

### 5.3.2 Análise Geral

Conforme apresentado na Figura 23, destaca-se, inicialmente, que a arquitetura ResNet101, em conjunção com um classificador MLP, evidencia o desempenho mais notável, alcançando um  $f1$ -score de 84,2%. Esta constatação supera resultados anteriormente obtidos, notadamente a combinação da VGG16 com MLP, que registrou um  $f1$ -score de 83,4%.

Figura 23 – Desempenho dos Diferentes Extratores em Combinação com Classificadores Variados.



Fonte: Autor.

Em sequência, a ResNet50, em parceria com um classificador MLP, e a ResNet152V2, utilizando o XGBoost como classificador, destacam-se com  $f1$ -scores de 83%. Este cenário sugere a eficácia dessas configurações específicas na tarefa em questão.

Observa-se, ademais, a variação no desempenho associada aos diferentes classificadores. Tanto os classificadores MLP quanto XGBoost emergem como principais, contudo, sua eficácia revela-se sensível à arquitetura específica utilizada. Este fenômeno sugere a necessidade de uma seleção cuidadosa do classificador em consonância com a arquitetura adotada.

No que concerne às arquiteturas, EfficientNetB6 e ResNet101 demonstram, em geral, desempenho robusto, superando a marca de 80% de  $f1$ -score. Em contrapartida, arquiteturas como MobileNet e InceptionV3 exibem desempenho ligeiramente inferior, com  $f1$ -scores situados na faixa de 70 a 76%.

A utilização do classificador SVM revela um desempenho variável, com eficácia destacada em algumas arquiteturas e desafios em outras. Notavelmente, a combinação EfficientNetB3 com SVM resultou em um  $f1$ -score de 59,2%, indicando uma interação menos eficiente entre esta arquitetura específica e o classificador SVM.

As arquiteturas da família EfficientNet (B0 a B7) exibem desempenhos distintos, ressaltando a influência da arquitetura específica na tarefa em questão. Variações notáveis no desempenho são observadas, indicando a necessidade de uma análise mais aprofundada das características individuais dessas arquiteturas.

Em análise mais abrangente, verifica-se que, para algumas arquiteturas ResNet (ResNet50, ResNet101, ResNet152), a variação na profundidade da rede não resulta em diferenças substanciais no desempenho. Este resultado sugere que, em determinados contextos, a complexidade adicional associada a uma maior profundidade não se traduz necessariamente em ganhos significativos de desempenho.

É válido destacar o desempenho da MobileNet em conjunto com o classificador XGBoost, alcançando um *f1-score* de 76,8%. Este resultado reforça sua relevância em contextos práticos de baixo recurso computacional, evidenciando que sua eficiência a torna uma opção estratégica para dispositivos embarcados e cenários com recursos limitados, garantindo desempenho competitivo e inferência rápida.

Concluindo, a presente análise é relevante para a escolha de arquiteturas e classificadores no processo de extração de características, destacando nuances específicas que merecem consideração cuidadosa em aplicações práticas.

## 5.4 Ajuste-fino da Arquitetura de Extração de Características

Uma hipótese investigada nesta tese foi a de realizar o ajuste fino (*fine-tuning*) das arquiteturas VGG16 e ResNet50, utilizando dados específicos de consumo de energia da base UK-DALE e das residências #1, #2, e #5 da base REDD, a fim de otimizar o processo de extração de características e, conseqüentemente, a classificação de cargas.

Em termos gerais, o ajuste fino consiste em adaptar os parâmetros de uma rede neural pré-treinada para um conjunto de dados específico, possibilitando que a rede aprenda características mais relevantes para a tarefa em questão. No contexto da classificação de cargas em NILM, essa estratégia pode ser particularmente benéfica, pois permite que a arquitetura pré-treinada capture padrões específicos relacionados ao consumo de energia nessas residências.

Ao utilizar dados das base UK-DALE e REDD para o ajuste fino, presume-se que se torna possível capturar nuances e características distintas nesses contextos, otimizando assim o desempenho da rede para a tarefa de classificação de cargas. Assim, a hipótese de ajuste fino destas arquiteturas é fundamentada em princípios sólidos e pode ser um caminho viável para potencializar a capacidade das redes neurais em capturar características relevantes para a classificação de cargas no contexto de NILM, levando a resultados mais robustos e adaptados às particularidades do conjunto de dados de interesse.

Com base nisto, foram estabelecidas duas abordagens distintas para o ajuste fino:

1. **Dados de todos os aparelhos** – retrainar a rede neural com todos os aparelhos presentes nas bases de dados oferece a vantagem de capturar a gama completa de padrões e características associadas a diferentes tipos de cargas. Isso pode levar a um modelo mais generalizado, capaz de lidar com uma variedade mais ampla de situações. No entanto, é crucial garantir que a quantidade e diversidade de dados sejam suficientes para evitar *overfitting* e promover a generalização;
2. **Dados dos cinco aparelhos que mais consomem energia** – focar no ajuste fino apenas dos cinco aparelhos que mais consomem energia em cada residência é uma estratégia que leva em consideração a relevância prática dos aparelhos de maior impacto no consumo total. Essa abordagem pode resultar em um modelo mais eficiente e especializado para identificar as cargas que mais contribuem para o consumo global de energia. No entanto, é fundamental garantir que os cinco aparelhos escolhidos sejam representativos e abranjam uma variedade de tipos de cargas.

Essa abordagem dupla oferece uma oportunidade valiosa para entender como a seleção dos aparelhos impacta o desempenho do modelo e se concentrar nos casos mais relevantes para a aplicação prática de NILM.

Por fim, uma outra investigação oportuna no processo de ajuste fino está relacionada à inicialização dos pesos dos modelos, sendo esta uma estratégia relevante que pode impactar significativamente o desempenho do modelo. No contexto desta tese, tanto ResNet50 quanto VGG16 foram avaliadas considerando duas estratégias distintas para este fim, sendo: 1) inicialização aleatória; e 2) utilizando pesos do treinamento prévio na base *ImageNet*.

Ao começar com pesos aleatórios, o modelo tem a oportunidade de aprender padrões específicos do conjunto de dados de ajuste fino. Essa abordagem pode ser particularmente eficaz se os padrões associados à classificação de cargas nas residências estudadas diferirem substancialmente daqueles presentes nas imagens de treinamento do *ImageNet*. No entanto, essa estratégia pode exigir mais tempo e dados para convergir para um desempenho otimizado. Na inicialização com pesos pré-treinados na *ImageNet*, o conhecimento prévio adquirido pela arquitetura em uma tarefa mais ampla de classificação de imagens pode ser benéfico, especialmente se as características aprendidas na *ImageNet* forem transferíveis para a tarefa específica de classificação de cargas. Essa abordagem geralmente requer menos dados para convergir, pois a rede já possui uma compreensão inicial das características visuais relevantes.

Diante do cenário anteriormente discutido, foram executados diversos experimentos e seus resultados estão consolidados na Tabela 13.

Tabela 13 – Impacto do Ajuste-fino das Arquiteturas Pré-treinadas no Processo de Classificação

Dados	Ajuste-fino	Inicialização	Arquitetura	Classificador	f1-score (%)
todos aparelhos	Aleatória	ResNet50		XGBoost	79,4
				SVM	69,2
				MLP	68,2
		VGG16		XGBoost	72,6
				MLP	67,2
				SVM	58,2
	ImageNet	ResNet50		XGBoost	81,8
				MLP	77,8
				SVM	71
		VGG16		XGBoost	77,8
				MLP	69,6
				SVM	66
top-5	Aleatória	ResNet50		XGBoost	72,8
				MLP	67,6
				SVM	62,4
		VGG16		XGBOOST	70,6
				MLP	58,4
				SVM	58,2
	ImageNet	ResNet50		XGBOOST	83,6
				MLP	82,8
				SVM	69,8
		VGG16		XGBOOST	79,4
				MLP	69,4
				SVM	67,8

### 5.4.1 Impacto da Inicialização de Pesos

Os resultados evidenciam que a escolha da inicialização de pesos exerce uma influência significativa no desempenho do modelo. Notavelmente, para a arquitetura ResNet50 iniciada com pesos da *ImageNet* e considerando apenas os cinco aparelhos de maior consumo (top-5), obteve-se o melhor desempenho, alcançando um *f1-score* de 83,6%. Isso sugere que a inicialização com pesos pré-treinados em uma tarefa ampla, como a classificação de imagens da *ImageNet*, pode fornecer uma base mais robusta para o ajuste fino em NILM, especialmente quando se concentra nos aparelhos mais relevantes.

Por outro lado, iniciar com pesos aleatórios resultou em desempenho inferior para várias configurações, indicando que a transferência de conhecimento prévio é benéfica para a tarefa específica de classificação de cargas.

## 5.4.2 Efeito da Estratégia de Amostragem de Dados para o Ajuste Fino

Ao considerar as duas estratégias de ajuste fino (todos aparelhos vs. top-5), observa-se que, em muitos casos, a abordagem de top-5 superou a estratégia que considera todos os aparelhos. Isso sugere que focar nos aparelhos mais impactantes pode ser uma estratégia eficaz para melhorar o desempenho do modelo, possivelmente reduzindo a complexidade do problema e concentrando os esforços de aprendizado nas cargas mais relevantes.

## 5.4.3 Generalização e Complexidade do Modelo

A análise dos resultados também levanta questões sobre a generalização do modelo. Configurações que consideraram top-5 aparelhos apresentaram frequentemente melhor desempenho, indicando que a simplificação do problema pode contribuir para uma melhor generalização. Isso sugere que, ao ajustar modelos para NILM, é crucial equilibrar a complexidade do modelo com a relevância prática das cargas de interesse.

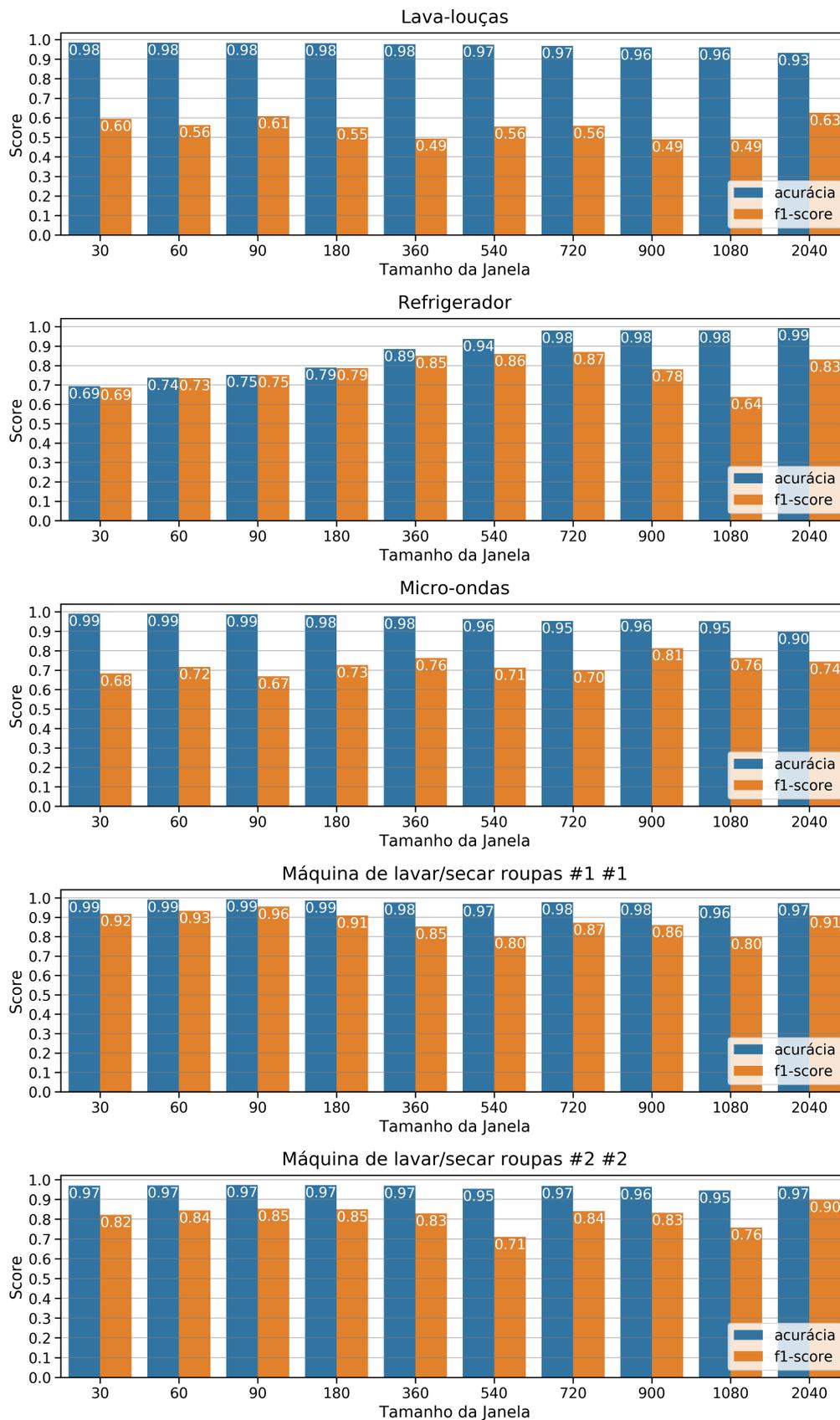
## 5.4.4 Considerações sobre Classificadores

A escolha do classificador também desempenha um papel crítico. Configurações que utilizaram XGBoost frequentemente alcançaram desempenho superior, sugerindo que, para este conjunto de dados e tarefa específica, a abordagem de boosting pode ser mais eficaz do que MLP ou SVM.

## 5.4.5 Observações Importantes

Com base nos resultados, fica evidente que a escolha da arquitetura, a seleção minuciosa de cargas para o ajuste fino e o modo de inicialização dos pesos são fatores cruciais que impactam o desempenho do modelo. Dessa forma, ao considerar cuidadosamente esses detalhes, é possível não apenas otimizar o desempenho local, mas também fortalecer a capacidade do modelo de extrair características relevantes e generalizar efetivamente para diferentes conjuntos de dados e condições operacionais. Essa abordagem refinada é essencial para impulsionar o avanço contínuo dos modelos de classificação de cargas no contexto de NILM.

Figura 20 – Desempenho da abordagem DTLFE + MLP para cada carga, considerando diferentes janelas.



Fonte: Autor.

---

## Capítulo 6

# Conclusões e Diretivas Futuras

---

O uso da transformação das séries temporais em imagens 2D a partir de RP se mostrou uma alternativa robusta para a rotulação de cargas residenciais contidas no conjunto de dados REDD. Ademais, notou-se uma maior eficácia ao realizar a extração automática de *features* por meio de um modelo CNN pré-treinado (base ImageNet) e, na sequência, utilizar esse vetor de *features* como entradas de modelos preditivos baseados em redes neurais artificiais do tipo MLP. Este resultado indica que a adoção de *transfer learning*, ainda que utilizando um modelo pré-treinado em uma base de imagens de domínio completamente destoante ao contexto de NILM, otimiza o processo classificatório final, sugerindo que o reaproveitamento do conhecimento aprendido anteriormente (generalização) proporciona a extração de *features* mais significativas.

Adicionalmente, a determinação do tamanho de janela e da taxa de amostragem adequada para a classificação de cada carga em função também da abordagem de classificação escolhida é um fator que contribui para a melhoria dos resultados, visto que haverá menor redundância e perda de informações. É válido destacar que algumas cargas ainda apresentam baixa taxa de classificação, principalmente em termos *de f1-score*. Uma hipótese para tal resultado é que isso ocorra devido à menor frequência de utilização dessas cargas, fazendo com que haja maior desbalanceamento entre as classes de dados. Logo, o desbalanceamento dos dados ainda é um desafio a ser superado nas pesquisas realizadas em NILM.

Na análise da generalização do modelo para diferentes contextos residenciais, observou-se um desempenho diferenciado em residências específicas, destacando a Residência #2 como um caso notável. A consistência nos padrões de consumo ao longo do tempo e a similaridade nas características dos aparelhos nessa residência podem ter contribuído significativamente para esse resultado expressivo. No entanto, ressalta-se a necessidade de

estudos mais aprofundados, dificultados pela limitação de acesso aos detalhes específicos do conjunto de dados REDD.

A análise de diversas arquiteturas de extratores de características destacou a eficácia da ResNet101. No entanto, observou-se variações no desempenho conforme o classificador utilizado, enfatizando a importância da escolha personalizada da combinação extrator-classificador. Além disso, a arquitetura MobileNet mostrou desempenho promissor, especialmente em cenários de recursos computacionais restritos, tornando-a uma alternativa viável para aplicações práticas. Sua eficiência computacional abre oportunidades para implementações em dispositivos com limitações de hardware, ampliando o alcance do modelo em ambientes residenciais e comerciais com recursos limitados.

A investigação sobre o ajuste fino do modelo extrator demonstrou a sensibilidade do desempenho a diferentes configurações. Fica evidente que a escolha criteriosa dos dados de ajuste fino, inicialização de pesos e seleção de cargas impacta diretamente a capacidade do modelo de generalizar para novos contextos. Esse entendimento refinado sobre o ajuste fino destaca a importância da preparação cuidadosa dos dados para otimizar o processo de ajuste fino e, conseqüentemente, o desempenho do modelo.

Em síntese, esta tese não apenas avança o entendimento no campo da classificação de cargas residenciais, mas também delinea direções promissoras para pesquisas futuras. Aprofundar a análise em conjuntos de dados mais detalhados, estender a avaliação de generalização e explorar estratégias avançadas para ajuste fino são passos essenciais. A abordagem equilibrada entre inovação técnica e considerações práticas destaca o potencial deste trabalho para orientar desenvolvimentos futuros no âmbito de NILM.

---

## Referências

---

ABDALI, A.; MAZLUMI, K.; NOROOZIAN, R. High-speed fault detection and location in dc microgrids systems using multi-criterion system and neural network. **Applied Soft Computing Journal**, Elsevier Ltd, v. 79, p. 341–353, 6 2019. ISSN 15684946.

AFZALAN, M.; JAZIZADEH, F.; WANG, J. Self-configuring event detection in electricity monitoring for human-building interaction. **Energy and Buildings**, Elsevier Ltd, v. 187, p. 95–109, 3 2019. ISSN 03787788.

AIAD, M.; LEE, P. H. Unsupervised approach for load disaggregation with devices interactions. **Energy and Buildings**, Elsevier Ltd, v. 116, p. 96–103, 3 2016. ISSN 03787788.

ALMUTAIRI, M.; ABUBAKAR, S.; CHIROMA, H. Detecting elderly behaviors based on deep learning for healthcare: recent advances, methods, real-world applications and challenges. **Ieee Access**, v. 10, p. 69802–69821, 2022.

ANDERSON, K. et al. Blued: A fully labeled public dataset for event-based non-intrusive load monitoring research. In: ACM. **Proceedings of the 2nd KDD workshop on data mining applications in sustainability (SustKDD)**. [S.l.], 2012. v. 7, p. 1–5.

ASGHARIGOVAR, S.; POURGHASEM, P.; SEYEDI, H. High impedance fault protection scheme for smart grids based on wpt and elm considering evolving and cross-country faults. **International Journal of Electrical Power and Energy Systems**, Elsevier Ltd, v. 107, p. 412–421, 5 2019. ISSN 01420615.

BAETS, L. D. et al. Appliance classification using vi trajectories and convolutional neural networks. **Energy and Buildings**, v. 158, p. 32–36, 2018. ISSN 0378-7788. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0378778817312690>>.

BAPTISTA, D. et al. Implementation strategy of convolution neural networks on field programmable gate arrays for appliance classification using the voltage and current (v-i) trajectory. **Energies**, v. 11, p. 1–18, 2018. ISSN 19961073.

BERGES, M. et al. Learning systems for electric consumption of buildings. **Proceedings of the 2009 ASCE International Workshop on Computing in Civil Engineering**, v. 346, p. 1–10, 2009.

BERGES, M. E. et al. Enhancing electricity audits in residential buildings with nonintrusive load monitoring. **Journal of Industrial Ecology**, John

- Wiley Sons, Ltd, v. 14, p. 844–858, 10 2010. ISSN 10881980. Disponível em: <<http://doi.wiley.com/10.1111/j.1530-9290.2010.00280.x>>.
- BERNASCONI, J. Learning in neural networks. In: **Dynamics and Stochastic Processes Theory and Applications**. [S.l.]: Springer, 1990. p. 42–54.
- BERNDT, D.; CLIFFORD, J. Using dynamic time warping to find patterns in time series. **KDD-94 Workshop on Knowledge Discovery in Databases**, p. 359–370, 1994.
- BHOTTO, M. Z. A.; MAKONIN, S.; BAJIĆ, I. V. Load disaggregation based on aided linear integer programming. **IEEE Transactions on Circuits and Systems II: Express Briefs**, Institute of Electrical and Electronics Engineers Inc., v. 64, p. 792–796, 7 2017. ISSN 15497747.
- BISHOP, C. M. **Pattern Recognition and Machine Learning (Information Science and Statistics)**. [S.l.]: Springer-Verlag, 2006. ISBN 0387310738.
- CAVALCA, D. L.; FERNANDES, R. A. S. Recurrence plots and convolutional neural networks applied to nonintrusive load monitoring. In: **2020 IEEE Power Energy Society General Meeting (PESGM)**. [S.l.: s.n.], 2020. p. 1–5.
- CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: . Association for Computing Machinery, 2016. v. 13-17-August-2016, p. 785–794. ISBN 9781450342322. Disponível em: <<http://dx.doi.org/10.1145/2939672.2939785>>.
- CHOLLET, F. **Deep Learning with Python**. 1st. ed. [S.l.]: Manning Publications Co., 2017. ISBN 1617294438.
- CORREA, S. d. J. d. C.; CASTRO, A. R. Automated non-intrusive load monitoring system using stacked neural networks and numerical integration. **IEEE Access**, IEEE, v. 8, p. 210566–210581, 2020.
- DANG, L.; PANG, P.; LEE, J. Depth-wise separable convolution neural network with residual connection for hyperspectral image classification. **Remote Sensing**, MDPI AG, v. 12, n. 20, p. 3408, Oct 2020. ISSN 2072-4292. Disponível em: <<http://dx.doi.org/10.3390/rs12203408>>.
- DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: **2009 IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2009. p. 248–255.
- D’INCECCO, M.; SQUARTINI, S.; ZHONG, M. Transfer learning for non-intrusive load monitoring. **Ieee Transactions on Smart Grid**, v. 11, p. 1419–1429, 2020.
- DONG, M. et al. An event window based load monitoring technique for smart meters. **IEEE Transactions on Smart Grid**, v. 3, p. 787–796, 2012. ISSN 19493053.
- DUDA, R. O.; HART, P. E.; STORK, D. G. **Pattern classification**. [S.l.]: John Wiley Sons, Ltd, 2012.
- ECKMANN, J. P.; KAMPHORST, O. O.; RUELLE, D. Recurrence plots of dynamical systems. **Europysics Letters**, v. 4, p. 973–977, 1987. ISSN 12864854.

- FAKHRAZARI, A.; VAKILZADIAN, H. A survey on time series data mining. **IEEE International Conference on Electro Information Technology**, p. 476–481, 2017. ISSN 21540373.
- FAUSTINE, A.; PEREIRA, L. Improved appliance classification in non-intrusive load monitoring using weighted recurrence graph and convolutional neural networks. **Energies**, v. 13, 2020. ISSN 19961073.
- FERNANDES, R. A. S.; SILVA, I. N. da; OLESKOVICZ, M. Load profile identification interface for consumer online monitoring purposes in smart grids. **IEEE Transactions on Industrial Informatics**, v. 9, p. 1507–1517, 2013. ISSN 1941-0050.
- FRIEDMAN, J. H. Greedy function approximation: A gradient boosting machine. **Annals of Statistics**, Institute of Mathematical Statistics, v. 29, p. 1189–1232, 10 2001. ISSN 00905364.
- GARCIA-CEJA, E.; UDDIN, M. Z.; TORRESEN, J. Classification of recurrence plots' distance matrices with a convolutional neural network for activity recognition. **Procedia Computer Science**, Elsevier B.V., v. 130, p. 157–163, 2018. ISSN 18770509. Disponível em: <<https://doi.org/10.1016/j.procs.2018.04.025>>.
- GLOROT, X.; BORDES, A.; BENGIO, Y. Deep sparse rectifier neural networks. In: . JMLR Workshop and Conference Proceedings, 2011. p. 315–323. ISSN 1938-7228. Disponível em: <<http://proceedings.mlr.press/v15/glorot11a.html>>.
- GRABEN, P. B. et al. Optimal estimation of recurrence structures from time series. **Epl**, v. 114, 2016. ISSN 12864854.
- GRAIS, E. M.; SEN, M. U.; ERDOGAN, H. Deep neural networks for single channel source separation. In: IEEE. **2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2014. p. 3734–3738.
- GUPTA, S.; REYNOLDS, M. S.; PATEL, S. N. Electrisense: Single-point sensing using emi for electrical event detection and classification in the home. In: . Association for Computing Machinery, 2010. p. 139–148. ISBN 9781605588438. Disponível em: <<https://doi.org/10.1145/1864349.1864375>>.
- HART, G. W. Nonintrusive appliance load monitoring. **Proceedings of the IEEE**, v. 80, p. 1870–1891, 1992. ISSN 15582256.
- HASSAN, T.; JAVED, F.; ARSHAD, N. An empirical investigation of v-i trajectory based load signatures for non-intrusive load monitoring. **IEEE Transactions on Smart Grid**, v. 5, p. 870–878, 2014. ISSN 19493053.
- HATAMI, N.; GAVET, Y.; DEBAYLE, J. Bag of recurrence patterns representation for time-series classification. **Pattern Analysis and Applications**, Springer London, p. 1–11, 2018. ISSN 14337541.
- HE, K. et al. Deep residual learning for image recognition. **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, 2016.
- HIMEUR, Y. et al. Building power consumption datasets: Survey, taxonomy and future directions. **Energy and Buildings**, The Authors, v. 227, p. 110404, 2020. ISSN 03787788. Disponível em: <<https://doi.org/10.1016/j.enbuild.2020.110404>>.

- HINTON, G. E. et al. Improving neural networks by preventing co-adaptation of feature detectors. 7 2012. Disponível em: <<https://arxiv.org/abs/1207.0580>>.
- HOUIDI, S. et al. Comparative evaluation of non-intrusive load monitoring methods using relevant features and transfer learning. **Energies**, v. 14, p. 2726, 2021.
- HOWARD, A. G. et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. **arXiv preprint arXiv:1704.04861**, 2017.
- HUA, D. et al. Simultaneous disaggregation of multiple appliances based on non-intrusive load monitoring. **Electric Power Systems Research**, Elsevier, v. 193, p. 106887, 2021.
- HUSSAIN, M.; BIRD, J. J.; FARIA, D. R. A study on cnn transfer learning for image classification. In: . Springer Verlag, 2019. v. 840, p. 191–202. ISBN 9783319979816. ISSN 21945357. Disponível em: <[https://link.springer.com/chapter/10.1007/978-3-319-97982-3\\_16](https://link.springer.com/chapter/10.1007/978-3-319-97982-3_16)>.
- KAREEM, A.; LIU, H.; SANT, P. Review on pneumonia image detection: a machine learning approach. **Human-Centric Intelligent Systems**, v. 2, p. 31–43, 2022.
- KELLY, J.; KNOTTENBELT, W. Neural nilm: Deep neural networks applied to energy disaggregation. In: . Association for Computing Machinery, Inc, 2015. p. 55–64. ISBN 9781450339810. Disponível em: <<https://dl.acm.org/doi/10.1145/2821650.2821672>>.
- KIM, H. et al. Unsupervised disaggregation of low frequency power measurements. In: . Society for Industrial and Applied Mathematics Publications, 2011. p. 747–758. ISBN 9780898719925. Disponível em: <<https://epubs.siam.org/page/terms>>.
- KIM, J.; LE, T. T. H.; KIM, H. Nonintrusive load monitoring based on advanced deep learning and novel signature. **Computational Intelligence and Neuroscience**, v. 2017, 2017. ISSN 16875273.
- KINGMA, D. P.; BA, J. L. Adam: A method for stochastic optimization. In: . International Conference on Learning Representations, ICLR, 2015. Disponível em: <<https://arxiv.org/abs/1412.6980v9>>.
- KOLTER, J. Z.; JAAKKOLA, T. **Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation**. 2012.
- KOLTER, J. Z.; JOHNSON, M. J. Redd: A public data set for energy disaggregation research. p. 1–6, 2011. ISSN 9781450308403.
- KONG, W. et al. Improving nonintrusive load monitoring efficiency via a hybrid programming method. **IEEE Transactions on Industrial Informatics**, IEEE Computer Society, v. 12, p. 2148–2157, 12 2016. ISSN 15513203.
- KONG, X. et al. Deep learning hybrid method for islanding detection in distributed generation. **Applied Energy**, Elsevier Ltd, v. 210, p. 776–785, 1 2018. ISSN 03062619.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: . [S.l.]: Curran Associates Inc., 2012. p. 1097–1105.
- KYRKOU, L.; NALMPANTIS, C.; VRAKAS, D. Imaging time-series for nilm. In: MACINTYRE, J. et al. (Ed.). [S.l.]: Springer, Cham, 2019. v. 100, p. 188–196. ISBN 978-3-030-20256-9.

- LAM, H. Y.; FUNG, G. S.; LEE, W. K. A novel method to construct taxonomy electrical appliances based on load signatures. **IEEE Transactions on Consumer Electronics**, v. 53, p. 653–660, 5 2007. ISSN 00983063.
- LE, T.-T.-H.; KIM, J.; KIM, H. Classification performance using gated recurrent unit recurrent neural network on energy disaggregation. In: IEEE. **2016 international conference on machine learning and cybernetics (ICMLC)**. [S.l.], 2016. v. 1, p. 105–110.
- LECUN, Y. A theoretical framework for back-propagation. In: TOURETZKY, D.; HINTON, G.; SEJNOWSKI, T. (Ed.). [S.l.]: Morgan Kaufmann, 1988. p. 21–28.
- LECUN, Y.; DENKER, J. S.; SOLLA, S. A. Optimal brain damage. In: **Advances in neural information processing systems**. [S.l.: s.n.], 1990. p. 598–605.
- LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, p. 2278–2324, 1998.
- LEEB, S. B.; SHAW, S. R.; KIRTLEY, J. L. Transient event detection in spectral envelope estimates for nonintrusive load monitoring. **IEEE Transactions on Power Delivery**, v. 10, p. 1200–1210, 1995. ISSN 19374208.
- LI, Z.; HOIEM, D. Learning without forgetting. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE Computer Society, v. 40, p. 2935–2947, 12 2018. ISSN 19393539.
- LIN, Y. H.; TSAI, M. S.; CHEN, C. S. Applications of fuzzy classification with fuzzy c-means clustering and optimization strategies for load identification in nilm systems. In: . [S.l.: s.n.], 2011. p. 859–866. ISBN 9781424473175. ISSN 10987584.
- LIU, H. et al. Complex power quality disturbances classification via curvelet transform and deep learning. **Electric Power Systems Research**, Elsevier Ltd, v. 163, p. 1–9, 10 2018. ISSN 03787796.
- MACHLEV, R.; LEVRON, Y.; BECK, Y. Modified cross-entropy method for classification of events in nilm systems. **IEEE Transactions on Smart Grid**, IEEE, v. 10, n. 5, p. 4962–4973, 2018.
- MAKONIN, S.; BAJIC, I.; POPOWICH, F. Efficient sparse matrix processing for nonintrusive load monitoring (nilm). **2nd International Workshop on Non-Intrusive Load Monitoring**, 2014. Disponível em: <[http://nilm.ca/papers/NILM\\_2014.pdf](http://nilm.ca/papers/NILM_2014.pdf)>.
- MAKONIN, S. et al. Exploiting hmm sparsity to perform online real-time nonintrusive load monitoring. **IEEE Transactions on Smart Grid**, Institute of Electrical and Electronics Engineers Inc., v. 7, p. 2575–2585, 11 2016. ISSN 19493053.
- MALIK, M. et al. A performance comparison of classification algorithms for rose plants. **Computational Intelligence and Neuroscience**, v. 2022, p. 1–16, 2022.
- MARWAN, N. et al. Recurrence plots for the analysis of complex systems. **Physics Reports**, v. 438, p. 237–329, 2007. ISSN 03701573.

- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The Bulletin of Mathematical Biophysics**, Kluwer Academic Publishers, v. 5, p. 115–133, 12 1943. ISSN 00074985. Disponível em: <<https://link.springer.com/article/10.1007/BF02478259>>.
- MEDEIROS, A. P. et al. Event classification in non-intrusive load monitoring using convolutional neural network. **2019 IEEE PES Conference on Innovative Smart Grid Technologies, ISGT Latin America 2019**, IEEE, p. 1–6, 2019.
- MOCANU, E.; NGUYEN, P. H.; GIBESCU, M. **Deep Learning for Power System Data Analysis**. [S.l.]: Elsevier Inc., 2017. 125-158 p. ISBN 9780128119693.
- MONK, A. T.; COMPTON, A. H. Recurrence phenomena in cosmic ray intensity. **Reviews of Modern Physics**, v. 11, p. 173–179, 1939.
- MORADZADEH, A. et al. Improving residential load disaggregation for sustainable development of energy via principal component analysis. **Sustainability**, Multidisciplinary Digital Publishing Institute, v. 12, n. 8, p. 3158, 2020.
- NALMPANTIS, C.; VRAKAS, D. On time series representations for multi-label nilm. **Neural Computing and Applications**, Springer, v. 32, n. 23, p. 17275–17290, 2020.
- NARAYANAN, K.; SIDDIQUI, S. A.; FOZDAR, M. Hybrid islanding detection method and priority-based load shedding for distribution networks in the presence of dg units. **IET Generation, Transmission and Distribution**, Institution of Engineering and Technology, v. 11, p. 586–595, 2 2017. ISSN 17518687. Disponível em: <<https://onlinelibrary.wiley.com/doi/10.1049/iet-gtd.2016.0437>>.
- NESTEROV, Y.; NEMIROVSKII, A. **Interior-Point Polynomial Algorithms in Convex Programming**. Society for Industrial and Applied Mathematics, 1994. Disponível em: <<https://epubs.siam.org/page/terms>>.
- PARSON, O. et al. Dataport and nilmtk: A building data set designed for non-intrusive load monitoring. In: IEEE. **2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)**. [S.l.], 2015. p. 210–214.
- \_\_\_\_\_. Using hidden markov models for iterative non-intrusive appliance monitoring. 2011.
- \_\_\_\_\_. Non-intrusive load monitoring using prior models of general appliance types. In: **Twenty-Sixth AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2012.
- PIOREK, M. **Analysis of Chaotic Behavior in Non-linear Dynamical Systems**. [S.l.: s.n.], 2019. v. 160. Remember to read Chapters 3 and 4. ISBN 978-3-319-94886-7.
- POPESCU, F. et al. Recurrence plot analysis for characterization of appliance load signature. **IEEE International Conference on Communications**, p. 4–7, 2014. ISSN 15503607.
- QUINLAN, J. R. **Induction of Decision Trees**. 1986. 81-106 p.
- RAFIQ, H. et al. Regularized lstm based deep learning model: First step towards real-time non-intrusive load monitoring. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2018. p. 234–239. ISBN 9781538664100.

- RAJABI, R.; ESTEBSARI, A. Deep learning based forecasting of individual residential loads using recurrence plots. **2019 IEEE Milan PowerTech, PowerTech 2019**, 2019.
- RASHID, H. et al. Can non-intrusive load monitoring be used for identifying an appliance's anomalous behaviour? **Applied Energy**, Elsevier Ltd, v. 238, p. 796–805, 3 2019. ISSN 03062619.
- RODRIGUES, I. et al. Classifying covid-19 positive x-ray using deep learning models. **Ieee Latin America Transactions**, v. 19, p. 884–892, 2021.
- ROOS, J. G. et al. Using neural networks for non-intrusive monitoring of industrial electrical loads. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 1994. p. 1115–1118. ISBN 0780318803.
- ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**, v. 65, p. 386–408, 11 1958. ISSN 0033295X. Disponível em: </record/1959-09865-001>.
- RUANO, A. et al. Nilm techniques for intelligent home energy management and ambient assisted living: A review. **Energies**, v. 12, p. 1–29, 2019. ISSN 19961073.
- RUSSAKOVSKY, O. et al. Imagenet large scale visual recognition challenge. **International Journal of Computer Vision**, Springer New York LLC, v. 115, p. 211–252, 12 2015. ISSN 15731405. Disponível em: <https://link.springer.com/article/10.1007/s11263-015-0816-y>.
- SADEGHIANPOURHAMAMI, N. et al. Comprehensive feature selection for appliance classification in nilm. **Energy and Buildings**, Elsevier B.V., v. 151, p. 98–106, 2017. ISSN 03787788. Disponível em: <http://dx.doi.org/10.1016/j.enbuild.2017.06.042>.
- SANDLER, M. et al. Mobilenetv2: Inverted residuals and linear bottlenecks. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2018. p. 4510–4520.
- SCHUSTER, M.; PALIWAL, K. K. Bidirectional recurrent neural networks. **IEEE Transactions on Signal Processing**, v. 45, p. 2673–2681, 1997.
- SEON, J. et al. Time-lapse image method for classifying appliances in nonintrusive load monitoring. **Energies**, v. 14, p. 7630, 2021.
- SILVA, D. F.; SOUZA, V. M. D.; BATISTA, G. E. Time series classification using compression distance of recurrence plots. **Proceedings - IEEE International Conference on Data Mining, ICDM**, p. 687–696, 2013. ISSN 15504786.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. In: . International Conference on Learning Representations, ICLR, 2015. Disponível em: <http://www.robots.ox.ac.uk/>.
- SINGH, S.; MAJUMDAR, A. Non-intrusive load monitoring via multi-label sparse representation-based classification. **IEEE Transactions on Smart Grid**, IEEE, v. 11, n. 2, p. 1799–1801, 2019.

- SINGHAL, V.; MAGGU, J.; MAJUMDAR, A. Simultaneous detection of multiple appliances from smart-meter measurements via multi-label consistent deep dictionary learning and deep transform learning. **IEEE Transactions on Smart Grid**, v. 10, p. 2969–2978, 2019.
- SOUZA, V. M.; SILVA, D. F.; BATISTA, G. E. Extracting texture features for time series classification. **Proceedings - International Conference on Pattern Recognition**, p. 1425–1430, 2014. ISSN 10514651.
- SRINIVASAN, D.; NG, W. S.; LIEW, A. C. Neural-network-based signature recognition for harmonic source identification. **IEEE Transactions on Power Delivery**, v. 21, p. 398–405, 2006. ISSN 08858977.
- SUZUKI, K. et al. Nonintrusive appliance load monitoring based on integer programming. In: . [S.l.: s.n.], 2008. p. 2742–2747. ISBN 9784907764296.
- SZEGEDY, C. et al. Going deeper with convolutions. In: . [S.l.]: IEEE Computer Society, 2015. v. 07-12-June-2015, p. 1–9. ISBN 9781467369640. ISSN 10636919.
- \_\_\_\_\_. rethinking the inception architecture for computer vision. 2016.
- TAN, M.; LE, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In: **International conference on machine learning**. [S.l.: s.n.], 2019. p. 6105–6114.
- TARIYAL, S. et al. Deep dictionary learning. **IEEE Access**, IEEE, v. 4, p. 10096–10109, 2016. ISSN 21693536.
- THIRUMALA, K. et al. A classification method for multiple power quality disturbances using ewt based adaptive filtering and multiclass svm. **Neurocomputing**, Elsevier B.V., v. 334, p. 265–274, 3 2019. ISSN 18728286.
- TIELEMAN, T.; HINTON, G. Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. **Technical Report**, 2017.
- TING, K. H. et al. A taxonomy of load signatures for single-phase electric appliances. **IEEE Power Electronics Specialist Conference**, v. 1, p. 24 – 29, 2005. Disponível em: <<http://www.lucente.us/past/pubs/2005-IEEE-PESC-preprint.pdf>>.
- VARDAKAS, J. S.; ZORBA, N.; VERIKOUKIS, C. V. A survey on demand response programs in smart grids: Pricing methods and optimization algorithms. **IEEE Communications Surveys Tutorials**, v. 17, p. 152–178, 2015. ISSN 1553-877X.
- VINCENT, P. et al. Extracting and composing robust features with denoising autoencoders. In: . Association for Computing Machinery, 2008. p. 1096–1103. ISBN 9781605582054. Disponível em: <<https://doi.org/10.1145/1390156.1390294>>.
- WANG, L. et al. Pre-trained models for non-intrusive appliance load monitoring. **Ieee Transactions on Green Communications and Networking**, v. 6, p. 56–68, 2022.
- WITTMANN, F. M.; LOPEZ, J. C.; RIDER, M. J. Nonintrusive load monitoring algorithm using mixed-integer linear programming. **IEEE Transactions on Consumer Electronics**, Institute of Electrical and Electronics Engineers Inc., v. 64, p. 180–187, 5 2018. ISSN 00983063.

- YANG, H.-T.; CHANG, H.-H.; LIN, C.-L. Design a neural network for features selection in non-intrusive monitoring of industrial electrical loads. In: IEEE. **2007 11th International Conference on Computer Supported Cooperative Work in Design**. [S.l.], 2007. p. 1022–1027.
- YANG, Y.-X. et al. A recurrence quantification analysis-based channel-frequency convolutional neural network for emotion recognition from eeg. **Chaos: An Interdisciplinary Journal of Nonlinear Science**, AIP Publishing LLC, v. 28, p. 085724, 8 2018. ISSN 1054-1500.
- ZAGORUYKO, S.; KOMODAKIS, N. Wide residual networks. **Proceedings of the British Machine Vision Conference 2016**, 2016.
- ZBILUT, J. P.; WEBBER, C. L. **Recurrence Quantification Analysis**. John Wiley Sons, Inc., 2006. Disponível em: <<https://onlinelibrary.wiley.com/doi/full/10.1002/9780471740360.ebs1355><https://onlinelibrary.wiley.com/doi/abs/10.1002/9780471740360.ebs1355><https://onlinelibrary.wiley.com/doi/10.1002/9780471740360.ebs1355>>.
- ZEIFMAN, M.; ROTH, K. Nonintrusive appliance load monitoring: Review and outlook. **IEEE Transactions on Consumer Electronics**, v. 57, p. 76–84, 2011. ISSN 00983063.
- ZHANG, C. et al. Sequence-to-point learning with neural networks for non-intrusive load monitoring. In: **Proceedings of the AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2018. v. 32, n. 1.
- ZHANG, W. J. et al. On definition of deep learning. In: . [S.l.]: IEEE Computer Society, 2018. v. 2018-June, p. 232–236. ISBN 9781532377914. ISSN 21544832.
- ZHAO, B. et al. Improving event-based non-intrusive load monitoring using graph signal processing. **IEEE Access**, Institute of Electrical and Electronics Engineers Inc., v. 6, p. 53944–53959, 2018. ISSN 21693536.
- ZHOU, Z. et al. Self-organizing probability neural network-based intelligent non-intrusive load monitoring with applications to low-cost residential measuring devices. **Transactions of the Institute of Measurement and Control**, SAGE Publications Sage UK: London, England, v. 43, n. 3, p. 635–645, 2021.
- ZOHA, A. et al. Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 12, n. 12, p. 16838–16866, 2012.
- \_\_\_\_\_. Low-power appliance monitoring using factorial hidden markov models. In: IEEE. **2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing**. [S.l.], 2013. p. 527–532.