

Universidade Estadual Paulista – UNESP
Instituto de Ciência e Tecnologia – Campus de Sorocaba

Marcos Vicente Cruz

**MÉTODO NÃO INTRUSIVO PARA
DESAGREGAÇÃO DE CARGAS BASEADAS
EM INVERSORES:**
o caso dos condicionadores de ar

Sorocaba

2025

Marcos Vicente Cruz

**MÉTODO NÃO INTRUSIVO PARA DESAGREGAÇÃO
DE CARGAS BASEADAS EM INVERSORES:**
o caso dos condicionadores de ar

Dissertação apresentada à Universidade Estadual Paulista (UNESP), Instituto de Ciência e Tecnologia, Sorocaba, para obtenção do título de Mestre em Engenharia Elétrica.

Área de Concentração: Automação

Orientador: Prof. Dr. Fernando Pinhabel Marafão

Coorientador: Prof. Dr. Felipe Leite Paes

Sorocaba

2025

C957m Cruz, Marcos Vicente
Método não intrusivo para desagregação de cargas
baseadas em inversores : o caso dos condicionadores de ar /
Marcos Vicente Cruz. -- Sorocaba, 2025
133 p. : il., tabs.

Dissertação (mestrado) - Universidade Estadual Paulista
(UNESP), Instituto de Ciência e Tecnologia, Sorocaba
Orientador: Fernando Pinhabel Marafão
Coorientador: Felipe Leite Paes

1. Inteligência artificial. 2. Redes inteligentes de
energia. 3. Energia elétrica - Conservação. I. Título.

CERTIFICADO DE APROVAÇÃO

TÍTULO DA DISSERTAÇÃO: Método não intrusivo para desagregação de cargas baseadas em inversores: o caso dos condicionadores de ar

AUTOR: MARCOS VICENTE CRUZ

ORIENTADOR: FERNANDO PINHABEL MARAFÃO

COORIENTADOR: FELIPE LEITE PAES

Aprovado como parte das exigências para obtenção do Título de Mestre em Engenharia Elétrica, área: Automação pela Comissão Examinadora:

Prof. Dr. FERNANDO PINHABEL MARAFÃO (Participação Virtual)
Diretoria / UNESP / Câmpus de Sorocaba - ICTS

Prof. Dr. WESLEY ANGELINO DE SOUZA (Participação Virtual)
Departamento Acadêmico de Elétrica (DAELE) / Universidade Tecnológica Federal do Paraná - UTFPR /
Campus de Cornélio Procopio

Prof. Dr. LUIS ARMANDO DE ORO ARENAS (Participação Virtual)
Departamento de Engenharia Ambiental / UNESP / Câmpus de Sorocaba - ICTS

Sorocaba, 03 de novembro de 2025.

Agradecimentos

Gostaria de começar reconhecendo que, desde o início desta jornada, é impossível alcançar qualquer objetivo sozinho; sempre precisamos das outras pessoas ao nosso redor para crescer, aprender e seguir em frente.

Agradeço primeiramente ao meu orientador, Fernando Pinhabel Marafão, e ao meu co-orientador Felipe Leite Paes e ao Luiz Bernardino, pelo apoio, dedicação, paciência e valiosos ensinamentos ao longo de todo o processo.

Dedico uma menção especial à memória da minha mãe, cujo amor e exemplo foram fundamentais para que eu chegasse até aqui, mesmo que não possa estar presente para compartilhar este momento.

Agradeço profundamente ao meu pai, pelo apoio constante, dedicação e força que me acompanham dia após dia.

Agradeço à minha família pelo amor, compreensão e apoio em todos os momentos.

Aos meus colegas da UNESP de Sorocaba, pelo incentivo, troca de ideias e momentos de descontração que tornaram essa jornada mais leve.

Por fim, agradeço a todos que, de alguma forma, ajudaram e acreditaram no desenvolvimento deste trabalho.

Resumo

Este trabalho apresenta técnicas de desagregação para monitoramento não intrusivo de cargas (sigla em inglês NILM) para dispositivos baseados em inversores. Embora cargas baseadas em inversores sejam mais eficientes em termos de consumo energético, elas adicionam uma camada extra de complexidade para algoritmos baseados em NILM, devido ao seu consumo de energia variável e complexo. Existem poucos estudos sobre esses tipos de cargas e as metodologias existentes demandam alto custo computacional. Este trabalho inclui um conjunto de dados de cargas com inversores e convencionais, uma proposta de algoritmo de detecção de eventos que é capaz de detectar cargas com variações de curta e longa duração; que levam de segundos a minutos a acontecer, com a possibilidade de desagregar eventos de longa duração simultâneos. A proposta faz o uso da Teoria da Potência Conservativa (sigla em inglês CPT) e da análise harmônica da corrente para extrair características elétricas para a identificação de cargas, além do uso de algoritmos de Inteligência Artificial que necessitam de relativo baixo custo computacional para classificação, tais como o KNN, o MLP e o SVM. É apresentada uma versão em C++ do algoritmo com informações de desempenho para sistemas embarcados e computador doméstico. Os resultados indicam que é viável utilizar os algoritmos e teorias citados em sistemas computacionais modernos com precisão semelhante a algoritmos mais modernos, mas que usam mais poder computacional.

Palavras-chave: Carga baseada em inversor, gerenciamento energético, identificação de carga, NILM, CPT, desagregação de cargas.

Abstract

This work presents disaggregation techniques for non-intrusive load monitoring (NILM) for inverter-based devices. Although inverter-based loads are more energy efficient, they add an extra layer of complexity to NILM-based algorithms due to their variable and complex power consumption. There are few studies on these types of loads, and existing methodologies require high computational costs. This work includes a dataset of inverter and conventional loads, as well as a proposal for an event detection algorithm capable of detecting loads with short and long duration variations, which occur over seconds to minutes, with the possibility of disaggregating simultaneous long-term events. The proposal makes use of the Conservative Power Theory (CPT) and current harmonic analysis to extract electrical characteristics for the identification of loads, in addition to the use of Artificial Intelligence algorithms that require relatively low computational cost for classification, such as KNN, MLP, and SVM. A C++ version of the algorithm is presented with performance information for embedded systems and home PCs. The results indicate that it is feasible to use the aforementioned algorithms and theories in modern computing systems with similar precision to more modern algorithms, but which use more computational power.

Keywords: Inverter-based load, energy management, load identification, NILM, CPT, load disaggregation.

Lista de ilustrações

Figura 1 – Diagrama simplificado de um sistema ILM.	22
Figura 2 – Diagrama simplificado de um sistema NILM.	23
Figura 3 – Processo típico de sistema NILM.	27
Figura 4 – Diagrama esquemático de três tipos de eventos.	30
Figura 5 – Diagrama de pré-processamento de eventos.	31
Figura 6 – Detecção de eventos por CUSUM.	31
Figura 7 – Exemplo do conceito de NILM com cargas desagregadas.	37
Figura 8 – Potência ativa de um ar-condicionado com inversor.	38
Figura 9 – Ampliação da potência ativa de um ar-condicionado com inversor.	39
Figura 10 – Dispositivo usado na aquisição do conjunto de dados.	44
Figura 24 – Potência ativa (CPT) de um ar-condicionado.	60
Figura 25 – Resultados a aplicação da função janela $W(t)$	63
Figura 26 – Potência da carga processada para encontrar os picos de variação	64
Figura 27 – Exemplo de primeiro início processado de uma carga.	65
Figura 28 – Corte de potência da carga.	66
Figura 29 – Fluxograma simplificado para encontrar o limiar mínimo de uma carga.	67
Figura 30 – Comparativo entre uma carga única e cargas agregadas.	68
Figura 31 – Fluxograma simplificado do detector de eventos de curta duração.	69
Figura 32 – Padrão de consumo de uma carga.	70
Figura 33 – Diagrama simplificado de evento.	71
Figura 34 – Diagrama simplificado de evento com parada lenta.	71
Figura 35 – Diagrama simplificado de evento com reversão na parada.	72
Figura 36 – Diagrama simplificado de evento com parada rápida.	72
Figura 37 – Comportamento do Detector de Eventos em cenários diferentes.	73
Figura 38 – Comportamento do Detector de Eventos em cenários diferentes.	74
Figura 39 – Comportamento ideal do Detector de Eventos.	74
Figura 40 – Falso evento de curta duração.	75
Figura 41 – Diagrama simplificado de evento de longa duração.	76
Figura 42 – Corrente das cargas e respectivas DFT.	78
Figura 42 – Corrente das cargas e respectivas DFT.	79
Figura 43 – Consumo de ar-condicionado.	82
Figura 44 – Diagramas de Voronoi comparando o seletor de dados ($dP \times dV$).	84
Figura 45 – Diagramas de Voronoi comparando o seletor de dados ($dW \times dP$).	84
Figura 46 – Diagramas de Voronoi comparando o seletor de dados ($dV \times dW$).	85
Figura 47 – Diagramas de Voronoi comparando o seletor de dados ($dI \times dP$).	85
Figura 48 – Diferença de voltagem sintética e original.	87

Figura 49 – Consumo de cargas com eventos assinalados. Asteriscos vermelhos: eventos de curta duração. Asteriscos verdes: eventos de longa duração.	95
Figura 49 – Consumo de cargas com eventos assinalados. Asteriscos vermelhos: eventos de curta duração. Asteriscos verdes: eventos de longa duração.	96
Figura 50 – Padrão de consumo do Cenário 1.	99
Figura 51 – Padrão de consumo do Cenário 2.	99
Figura 52 – Padrão de consumo do Cenário 3.	100
Figura 53 – Padrão de consumo do Cenário 6.	102
Figura 54 – Diagrama de Voronoi da carga com múltiplos <i>setpoints</i>	103
Figura 55 – Carga real e sintética ascendente para desagregação.	104
Figura 56 – Carga real e sintética ascendente para desagregação.	105
Figura 57 – Diagramas de voronoi do cenário 1.	123
Figura 58 – Diagramas de voronoi do cenário 2.	124
Figura 59 – Diagramas de voronoi do cenário 3.	125
Figura 60 – Localização de eventos do cenário 1.	127
Figura 61 – Localização de eventos do cenário 2.	128
Figura 62 – Localização de eventos do cenário 3.	129
Figura 63 – Diagrama com o formato do pacote de dados.	132
Figura 64 – Código do Seletor de Dados.	133

Lista de tabelas

Tabela 1 – Categorização de algumas técnicas de Monitoramento de Carga Não Intrusiva (NILM).	33
Tabela 2 – Resultados dos métodos por tempo de execução e exatidão.	41
Tabela 3 – Especificações técnicas do medidor AIW-PEX.	44
Tabela 4 – Precisão das grandezas mensuráveis.	45
Tabela 5 – F -scores médios e sua relação à porcentagem F_1 .	64
Tabela 6 – F -scores médios e sua relação à porcentagem F_2 .	68
Tabela 7 – Comparação entre diferentes valores de F_3 .	83
Tabela 8 – F -scores para vários cenários.	100
Tabela 9 – F -scores para vários cenários.	101
Tabela 10 – Análise PCA para o Cenário 1.	101
Tabela 11 – Análise PCA para o Cenário 2.	101
Tabela 12 – Análise PCA para o Cenário 3.	101
Tabela 13 – F -scores para múltiplos <i>setpoints</i> .	102
Tabela 14 – Análise PCA para o Cenário 6.	102
Tabela 15 – Análise PCA para uma rampa (Cenário 4).	104
Tabela 16 – Análise PCA para duas rampas (Cenário 5).	105
Tabela 17 – Tempo de desagregação da Plataforma 1.	106
Tabela 18 – Tempo de desagregação da Plataforma 2.	106
Tabela 19 – Tempo de desagregação da Plataforma 3.	106
Tabela 20 – Tempo total de calculo dos atributos para a Plataforma 1.	106
Tabela 21 – Tempo total de calculo dos atributos para a Plataforma 2.	107
Tabela 22 – Tempo total de calculo dos atributos para a Plataforma 3.	107
Tabela 23 – Tempo de treinamento da Plataforma 1.	107
Tabela 24 – Tempo de treinamento da Plataforma 2.	107
Tabela 25 – Tempo de treinamento da Plataforma 3.	107
Tabela 26 – Tempo total de calculo do limiar mínimo.	108
Tabela 27 – Tempo de desagregação da Plataforma 1.	108
Tabela 28 – Tempo de desagregação da Plataforma 2.	108
Tabela 29 – Tempo de desagregação da Plataforma 3.	108
Tabela 30 – Tempo total de calculo dos atributos para a Plataforma 1.	108
Tabela 31 – Tempo total de calculo dos atributos para a Plataforma 2.	108
Tabela 32 – Tempo total de calculo dos atributos para a Plataforma 3.	109
Tabela 33 – Tempo de treinamento da Plataforma 1.	109
Tabela 34 – Tempo de treinamento da Plataforma 2.	109
Tabela 35 – Tempo de treinamento da Plataforma 3.	109

Tabela 36 – Tempo total de calculo do limiar mínimo. 109

Lista de abreviaturas e siglas

AC	Corrente Alternada
API	Interface de Programação de Aplicativos
BTU	Unidade Térmica Britânica
CA	Corrente Alternada
CC	Corrente Contínua
CNN	Rede Neural Convolucional
CPT	Teoria da Potência Conservativa
CUSUM	Gráfico de Controle Cumulativo da Soma
DFT	Transformada Discreta de Fourier
FFT	Transformada Rápida de Fourier
FSM	Máquina de Estado Finito
GLR	Razão de Verossimilhança Generalizada
GOF	Bondade do Ajuste
GTFD	Distribuições Generalizadas de Tempo-Frequência na Classe de Cohen
HHT	Transformada de Huang-Hilbert
IEEE	Instituto de Engenheiros Elétricos e Eletrônicos
ILM	Monitoramento Intrusivo de Carga
IoT	Internet das Coisas
KNN	k-Vizinhos mas Próximos
LDA	Análise de Discriminante Linear
LSTM	Memória de Longo e Curto Prazo
MAF	Filtro de Média Móvel
MLP	Perceptron Multi-camadas

NILM	Monitoramento de Carga não Intrusivo
ONS	Operador Nacional do Sistema Elétrico
PCA	Análise de Componentes Principais
PLL	Malha de Captura de Fase
PRODIST	Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional
RNN	Rede Neural Recorrente
RSS	Raiz da Soma dos Quadrados
SILM	Monitoramento Semi-intrusivo de Carga
STFT	Transformada de Fourier de Curto Prazo
SVM	Máquina de Vetores de Suporte
THD	Distorção Harmônica Total
VFD	Unidade de Frequência Variável
VSD	Motor Acionado por Inversor
VTCD	Variação de Tensão de Curta Duração

Lista de símbolos

dI	Diferença de harmônicas de corrente
dP	Diferença de potência ativa
dW	Diferença de energia reativa
dV	Diferença de harmônicas da potência residual (nula)
G_e	Condutância equivalente da carga
I^u	Corrente de desbalanço eficaz
I_a^u	Corrente ativa de desbalanço eficaz
I_r^u	Corrente reativa de desbalanço eficaz
I_v	Corrente residual coletiva eficaz
$i(t)$	Corrente em função do tempo
i_a	Corrente ativa
I_a	Corrente ativa eficaz
i_a^b	Corrente ativa balanceada
i_r	Corrente reativa
I_r	Corrente reativa eficaz
i_r^b	Corrente reativa balanceada
$p(t)$	Potência ativa instantânea
P_{max}	Potência máxima
P	Potência ativa
W	Energia reativa
$w(t)$	Energia reativa instantânea
$W(t)$	Função janela
Q	Potência reativa

S	Comprimento de <i>buffer</i> circular
t	Período no tempo
t_n	Períodos inicial no tempo
T	Período
U	Tensão eficaz
$v(t)$	Tensão em função do tempo
V	Potência residual (nula)
$x_{h m}$	Sinal de referência unitário de fase
$x_{h\perp m}$	Sinal de referência unitário de quadratura
ΔP	Variação da potência ativa
ΔQ	Variação da potência reativa
θ_{1m}	Ângulo de sincronização da frequência fundamental
θ_{hm}	Ângulo de sincronização harmônica
λ	Fator de potência
λ_D	Fator de linearidade
λ_Q	Fator de reatividade
λ_U	Fator de assimetria

Sumário

1	INTRODUÇÃO	21
1.1	O sistema ILM	21
1.2	O sistema NILM	22
1.3	Cargas baseadas em inversores	23
1.3.1	A questão do ar-condicionado com inversor	23
1.4	Objetivos	24
1.5	Organização do trabalho	25
2	REVISÃO BIBLIOGRÁFICA	27
2.1	A metodologia NILM	27
2.1.1	Aquisição de Dados	27
2.1.2	Pré Processamento	28
2.1.3	Detecção de Eventos	28
2.1.3.1	Métodos heurísticos especialistas	28
2.1.3.2	Modelos probabilísticos	29
2.1.3.3	Modelos de correspondência	29
2.1.3.4	Métodos de detecção para cargas baseadas em inversores	30
2.1.3.5	Extração de Características	32
2.1.3.6	Identificação da Carga	33
2.1.3.7	Desagregação da Potência da Carga	36
2.2	Deficiências na desagregação para cargas com inversor	37
2.3	A Teoria da Potência Conservativa	40
2.4	Desagregação de cargas usando a CPT	41
3	METODOLOGIA	43
3.1	Conjunto de dados de análise	43
3.1.1	Equipamento usado na aquisição de dados	43
3.1.2	Formato dos dados do conjunto de dados	45
3.1.3	Descrição de conjunto de dados agregados	45
3.1.4	Descrição das cargas coletadas	46
3.2	Estratégia de Extração de características	59
3.3	Detector de Eventos	62
3.3.1	Limiar mínimo	63
3.3.2	Detector de Eventos Curta Duração	66
3.3.3	Deficiências no detector de eventos de curta duração	73
3.3.4	Detector de Eventos de Longa Duração	74

3.3.5	Deficiências do detector de Eventos de Longa Duração	76
3.3.6	Procedimento de geração de evento para desligamento abrupto	76
3.4	Extração das grandezas elétricas	77
3.5	Análise harmônica da corrente	77
3.6	Análise harmônica da potência	80
3.7	Desagregação de evento de ar-condicionado com diferentes setpoints	81
3.8	Extração de dados de aprendizado	82
3.9	Seleção de Dados	83
3.10	Modelamento de cargas para teste de validação	85
3.10.1	Dificuldades na simulação de harmônicas de tensão	86
3.10.2	Dificuldades na simulação de queda de tensão	86
3.10.3	Dificuldades na simulação da frequência de rede	87
3.11	Metodologia de desagregação	87
3.12	Parâmetros de Inteligência artificial usados e diferenças entre plata- formas	89
3.13	Implementação em C++ do algoritmo	90
3.13.1	Plataforma 1	91
3.13.2	Plataforma 2	91
3.13.3	Plataforma 3	92
3.13.4	Bibliotecas utilizadas	92
3.14	Outros usos da lista de eventos	94
4	RESULTADOS	95
4.1	Resultados do Detector de Eventos	95
4.2	Resultados da desagregação	98
4.2.1	Resultados de Ar-condicionado com múltiplos <i>setpoints</i>	101
4.2.2	Resultados de Ar-condicionado com Inversor e Cargas Sintéticas	103
4.3	Resultados de tempo de execução em C++	105
4.3.1	Resultados para 1024 amostras por ciclo	105
4.3.2	Resultados para 128 amostras por ciclo	107
5	CONCLUSÕES E TRABALHOS FUTUROS	111
5.1	Trabalhos futuros	111
5.2	Publicações Realizadas	112
	REFERÊNCIAS	113
	APÊNDICE A – DIAGRAMAS DE VORONOI	123
	APÊNDICE B – GRÁFICOS DOS EVENTOS	127

APÊNDICE C – DESCRIÇÃO DO FORMATO DO CONJUNTO DE DADOS	131
APÊNDICE D – CÓDIGO-FONTE DO SELETOR DE DADOS . .	133

1 Introdução

Com o uso crescente de aparelhos elétricos e a demanda por sistemas elétricos de resfriamento e aquecimento, a demanda por eletricidade vem crescendo constantemente no último século [1]. Essa tendência contribui para maiores emissões de gases de efeito estufa, devido à dependência excessiva de eletricidade gerada por combustíveis fósseis [2], aumentando o preço da eletricidade para os consumidores. Para resolver esses problemas, sistemas de monitoramento foram desenvolvidos para auxiliar os consumidores a estarem mais cientes de como a energia é gasta e como reduzir seu uso. O objetivo principal desses sistemas é fomentar a conscientização energética (*energy-awareness*) aos usuários e o controle de gastos [3]. Se a conta de eletricidade é mostrada juntamente com o consumo individual detalhado de cada carga, o consumidor reduz até 12% desta despesa [4].

Neste contexto, três métodos foram criados para analisar de forma inteligente o consumo de cargas elétricas: um analisa o consumo de cada aparelho individualmente, chamado monitoramento de carga intrusiva (ILM), o segundo identifica cada carga analisando o consumo agregado de energia, chamado monitoramento de carga não intrusiva (NILM), e um terceiro, o SILM que se propõe a utilizar partes de ambos sistemas nesta tarefa. O método NILM tem a vantagem de minimizar o número de sensores necessários para instalação em comparação ao ILM, que requer a instalação de um sensor para cada carga. O sistema SILM propõe uma estratégia mista em que cargas agrupadas são monitoradas [5].

1.1 O sistema ILM

O Monitoramento Intrusivo de Carga (ILM) envolve a medição do consumo de eletricidade de um ou alguns aparelhos usando medidores de baixo custo. O termo “intrusivo” significa que os medidores estão localizados próximo ao aparelho monitorado [6]. Os dados desses medidores de baixo custo são enviados a um dispositivo concentrador, no qual são apresentados gráficos de consumo por cada medidor. Soluções como esta são encontradas facilmente no mercado. Este tipo de sistema inteligente necessita da instalação de diversos dispositivos, como a tomada inteligente e a central que concentra os dados, tornando a instalação e manutenção mais complexa. Existem alguns tipos de tomadas inteligentes que também podem acionar e desligar o equipamento automaticamente, segundo a configuração feita pelo usuário [7]. A Figura 1 mostra de maneira simplificada um sistema ILM.

Os sistemas ILM não são abordados nesse trabalho por causa da desvantagem da necessidade de instalação de vários dispositivos, tornando o sistema NILM mais prático de ser empregado em instalações elétricas e, conseqüentemente, mais interessante para

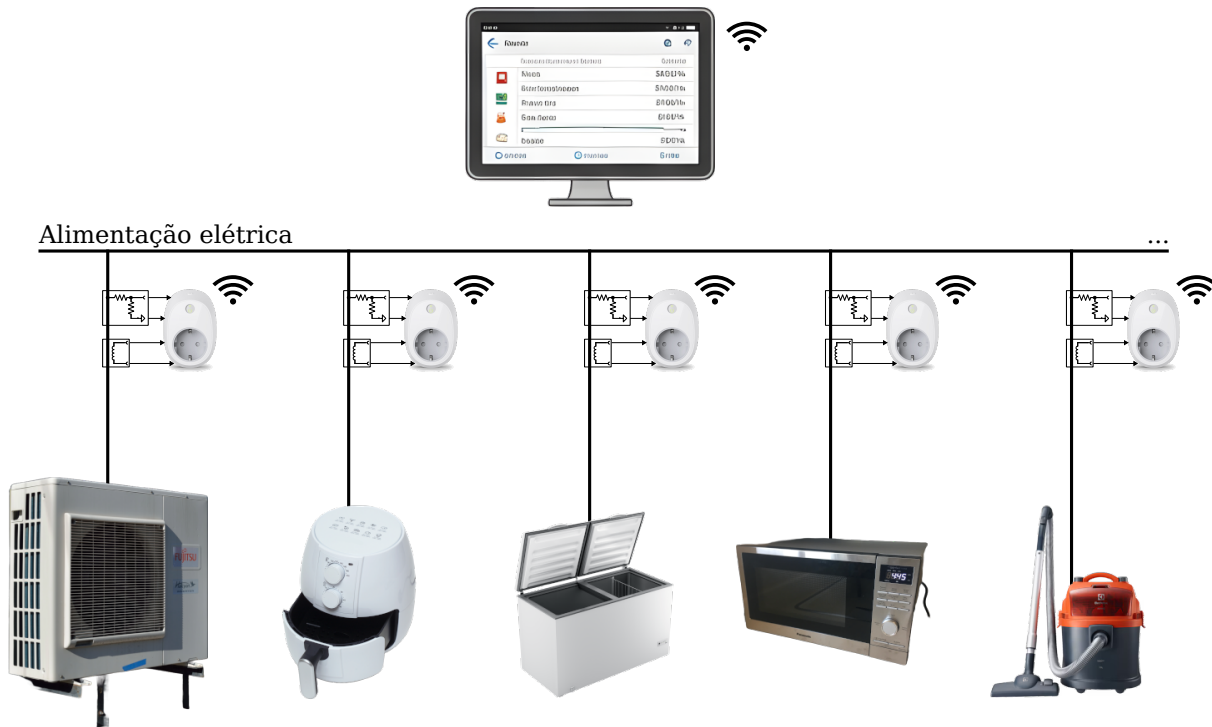


Figura 1 – Diagrama simplificado de um sistema ILM.

Fonte: Autoria Própria

pesquisar sobre.

1.2 O sistema NILM

NILM é um processo para analisar mudanças na tensão e corrente fornecidos a uma residência, comércio ou indústria, com o objetivo de identificar quais aparelhos são usados, bem como o seu consumo individual de energia. Assim, NILM, da sigla em inglês para *Non-Intrusive Load Monitoring*, é o nome dado a sistemas de monitoramento de cargas elétricas de forma não intrusiva.

Em sistemas NILM é necessário somente um equipamento de monitoramento, enquanto nos sistemas ILM, são necessários vários equipamentos. Ele é considerado não intrusivo por eliminar a necessidade de instalação de vários dispositivos medidores [7], tornando-o mais vantajoso no momento da implantação em comparação a sistemas ILM.

Os sistemas de NILM foram inicialmente propostos por Hart em 1992 [8], e visam extrair informação estratificada de cada dispositivo elétrico em uma residência pela observação de grandezas elétricas agregadas da entrada de energia [9]. O consumo de cada dispositivo elétrico conectado ao ponto comum e seus períodos ligados e desligados são então extraídos [9][10].

O método NILM foi pouco explorado pela comunidade científica por décadas até que melhores algoritmos de aprendizado de máquina e sistemas computacionais de maior

poder de processamento surgiram. A Figura 2 mostra de forma simplificada o sistema NILM.

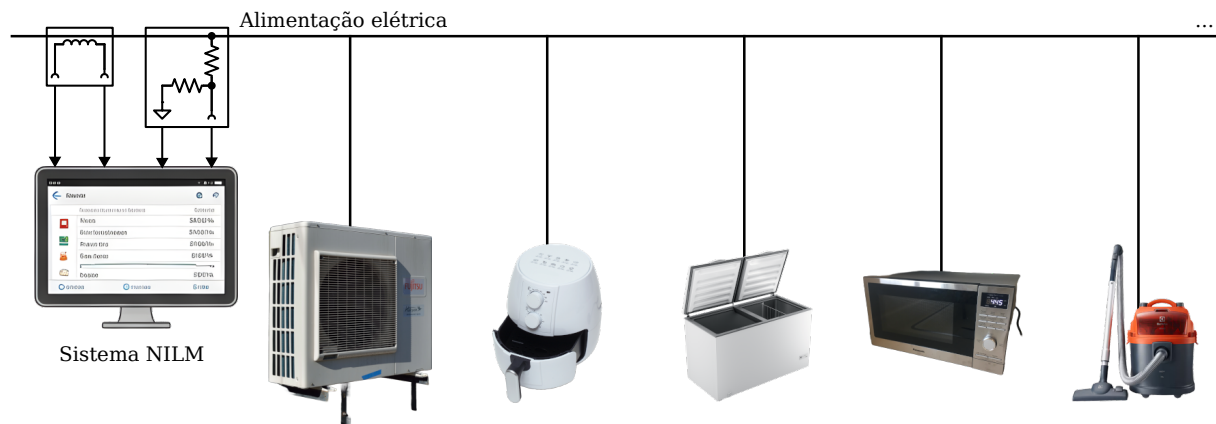


Figura 2 – Diagrama simplificado de um sistema NILM.

Fonte: Autoria Própria

1.3 Cargas baseadas em inversores

Inversores, também chamados de *AC Drives*, ou VFD, são dispositivos eletrônicos que realizam a conversão CC-CA por meio de dispositivos eletrônicos de potência. Eles também são responsáveis por controlar a velocidade e o torque de motores elétricos [11, 12]. Cargas baseadas em inversores são cargas que usam inversores para a conversão CC-CA. Alguns exemplos comuns de cargas baseadas em inversores são motores acionados por inversor (VSD), ar-condicionados, máquinas de lavar, sistemas fotovoltaicos e veículos elétricos.

O advento de cargas baseadas em inversores traz uma camada extra de complexidade aos sistemas NILM. Conforme abordaremos nesse trabalho, os estudos não cobrem todos os problemas criados por eles. Estudos abordando eventos de ativação simultânea ou de início lento são escassos ou não existem. Alguns estudos precisam de todo o histórico de dados para funcionar, limitando sua utilização, pois mais espaço de armazenamento é necessário. Estudos focando em aumentar o desempenho dos algoritmos também são escassos.

1.3.1 A questão do ar-condicionado com inversor

Os equipamentos de ar-condicionado com bomba de calor têm muitas vantagens, como alta eficiência e sua adequação tanto para aquecimento quanto para resfriamento. No mercado global, os sistemas de ar-condicionado que têm um compressor acionado por um motor de indução de velocidade constante, conectado diretamente à fonte de alimentação comercial, são amplamente utilizados. Este sistema emprega controle *on-off* para controlar

a temperatura ambiente e causa uma grande variação na temperatura ambiente e também uma perda de energia devido aos frequentes ciclos *on-off*, quando a temperatura ambiente está próxima da temperatura predefinida. Por outro lado, o controle de um ar-condicionado com inversor contribui para o conforto devido à temperatura ambiente estável, e o consumo de energia do sistema de ar-condicionado pode ser reduzido para menos da metade do que um convencional [13]. Considerando as vantagens relativas à eficiência energética e ao conforto térmico, os aparelhos de ar-condicionado equipados com inversor vêm se tornando progressivamente mais comuns, o que torna esse dispositivo um objeto de estudo de maior relevância.

O processo de inicialização de um ar-condicionado com inversor é diferente em comparação com um convencional. Os equipamentos de ar-condicionado com inversor aumentam gradualmente a frequência do motor do compressor durante o processo de início para mitigar o pico de corrente de partida e melhorar a eficiência operacional. Seu consumo de energia não aumenta monotonamente e pode experimentar aumentos e reduções de consumo, o que torna sua forma de onda de início significativamente diferente, em comparação com outros aparelhos comuns [14].

1.4 Objetivos

Essa dissertação de mestrado visa apresentar uma nova metodologia de monitoramento não intrusivo para cargas que têm como componentes internos motores acionados por inversores. O cenário do estudo é um ambiente residencial com a existência de cargas monofásicas, bifásicas e trifásicas.

Os algoritmos escolhidos são aqueles em que se possa obter uma boa exatidão na desagregação da carga e que ao mesmo tempo sejam de relativamente baixo custo computacional. Com isso, a implementação de hardware pode ser feita usando componentes de baixo consumo energético, e potencialmente, menor custo. Esses sistemas NILM de baixo consumo podem, então, ser oferecidos em massa a consumidores domésticos.

Este trabalho estende os estudos feitos por Souza [7] e Paes [15], possibilitando a identificação e desagregação de cargas com padrão de consumo mais complexo, tais como, cargas baseadas em inversores.

As principais contribuições deste trabalho são:

- Criação de um novo algoritmo para detectar os tipos de eventos que ocorrem em cargas baseadas em inversores, de transição lenta e rápida, usando uma janela fixa de dados. Este algoritmo deverá também detectar eventos de transição rápida que acontecem durante um evento de transição lenta e eventos de transição lenta simultâneos;

- Desenvolvimento do algoritmo de identificação e desagregação de cargas, utilizando métodos de inteligência artificial para a identificação, para ser usado com os tipos de carga utilizados nessa dissertação;
- Uso de parcelas da Teoria da Potência Conservativa e da análise harmônica da corrente para extração de atributos de classificação das cargas;
- Proposição de um valor ótimo de limiar mínimo a considerar como carga ligada, determinado por um algoritmo específico;
- Obtenção de um conjunto de dados com as cargas baseadas em inversores e outras cargas relevantes, para a metodologia proposta ser testada e validada.

1.5 Organização do trabalho

Esta dissertação de mestrado está dividida em capítulos, resumidos brevemente a seguir:

- Capítulo 2: Apresenta uma revisão bibliográfica das técnicas de NILM existentes e suas deficiências para analisar cargas baseadas em inversores.
- Capítulo 3: Descreve a obtenção do conjunto de dados para teste e validação dos novos métodos, uma descrição dos métodos de extração de características e dos algoritmos de detecção de evento para cargas com inversores, a apresentação da metodologia de desagregação das cargas e a apresentação de uma implementação desenvolvida em C++ dos métodos apresentados.
- Capítulo 4: São mostrados os resultados de precisão obtidos dos algoritmos para a desagregação de ar-condicionado com inversor embutido, juntamente com outras cargas relevantes e as métricas de desempenho da implementação em C++.
- Capítulo 5: São apresentadas as conclusões desta dissertação, os trabalhos publicados e sugestões para trabalhos futuros.

2 Revisão Bibliográfica

Esta revisão bibliográfica irá abordar os conceitos básicos de um sistema NILM, a aquisição de dados, o pré-processamento, a detecção de eventos de acionamento e desligamento, a extração de características e a desagregação e identificação da carga. Depois serão abordadas as limitações dos algoritmos de NILM atuais para cargas baseadas em inversores. Por fim, será abordada também a Teoria da Potência Conservativa (CPT) que será usada para a extração de características elétricas.

2.1 A metodologia NILM

O sistema NILM analisa as mudanças na tensão e corrente fornecidos a uma residência, estabelecimento ou indústria, com objetivo de identificar quais equipamentos são usados, bem como o seu consumo individual de energia. Somente um dispositivo é necessário para efetuar tal medição. O processo típico de um sistema NILM pode ser descrito como na Figura 3.

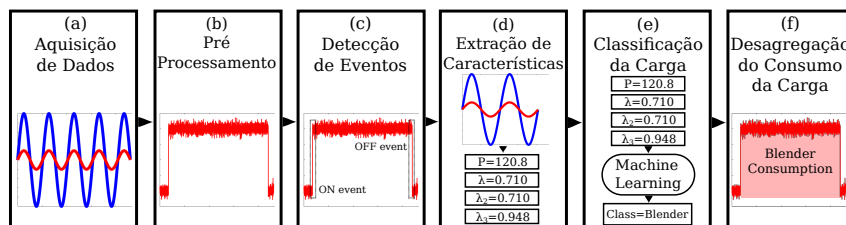


Figura 3 – Processo típico de sistema NILM.

Fonte: Garcia et. al.[16]

2.1.1 Aquisição de Dados

Neste estágio são adquiridos os dados de consumo agregado das cargas no ponto de acoplamento comum. Os sinais elétricos podem ser obtidos com o uso de equipamentos adequados.

A frequência de amostragem dos dados varia conforme o algoritmo de extração de características usado. Zeifman e Roth [17] classificam os métodos dependendo desta frequência. Os métodos macroscópicos são aqueles em que as características são extraídas na frequência de amostragem de 1 kHz ou menos. Os métodos microscópicos são aqueles em que as características são extraídas na frequência de amostragem de kHz ou MHz [18, 19, 17].

A aquisição de dados pode ser também independente da frequência de amostragem, utilizando análise de eventos, índices de probabilidade entre outros; Rottondi et al. [20]; Huchtkoetter e Reinhardt [21]; Bouhouras et al. [22]; Haq and Jacobsen [23] [16].

2.1.2 Pré Processamento

De acordo com Aguiar [9], os sinais elétricos para tarefas de treinamento do sistema NILM podem ser obtidos a partir de conjuntos de dados públicos e por hardware de aquisição dedicado. Os dados para efetuar a desagregação devem ser obtidos pelo hardware de aquisição do sistema de NILM. Geralmente, as informações dos conjuntos de dados e dos sistemas de aquisição não estão formatados adequadamente para a tarefa NILM. Assim o estágio de pré-processamento consiste na padronização desses sinais de entrada para que a tarefa NILM específica possa ser executada adequadamente. Algumas subtarefas da fase de pré-processamento são:

- Aquisição de sinais em intervalos de tempo adequados;
- Subamostragem;
- Normalização;
- Eliminação de ruído;
- Divisão de subconjuntos de treinamento, teste e validação.

A escolha das subtarefas de pré-processamento depende da natureza dos dados e pode variar dependendo da tarefa NILM específica.

2.1.3 Detecção de Eventos

A detecção de evento é feita encontrando alterações no consumo energético que caracterizam um acionamento ou desligamento de alguma carga elétrica.

Existem vários métodos de detecção, como mostrado a seguir, com seus respectivos autores [24, 25].

2.1.3.1 Métodos heurísticos especialistas

Estes métodos requerem um conhecimento prévio sobre a carga a ser detectada. Por exemplo, as principais cargas são detectadas com base em regras, como a duração do estado transitório [24].

Hart [8] utiliza a potência normalizada alimentando um detector de borda, no qual encontra os momentos no tempo e durações das mudanças. De acordo com Hart “A

detecção de borda de potência desempenha um papel essencial em uma implementação baseada em eventos do NILM [16].

Souza et al. [26] propuseram o uso de uma máquina de estado (FSM) que monitora a corrente agregada e considera um evento de ligamento ou desligamento quando houver um aumento maior que um limiar determinado.

Yi et al. [27] propuseram um método baseado em janelas compostas para ambientes industriais trifásicos. Eles levam em consideração a deformação que transformadores de correntes e tensão causam no sinal adquirido e no tempo maior de *startup* de grandes motores usados em indústrias.

Garcia et al. [16] utilizaram o conceito de janelamento para a segregação do sinal de potência através de um conjunto móvel de amostras representando estados estacionários e transitórios, então a parcela residual de corrente calculada pela Teoria da Potência Conservativa (CPT) é utilizada para identificar os estados transitórios.

2.1.3.2 Modelos probabilísticos

Luo et. al. [28] apresentou um método de detecção baseado na Razão de Verossimilhança Generalizada (GLR). O algoritmo de detecção GLR calcula a decisão estatística a partir do logaritmo natural de uma razão de distribuições de probabilidade antes e depois de uma mudança média da potência. Berges et al. [29] também propôs um método baseado em GLR e melhorou o método de Luo reduzindo o número de parâmetros que precisam ser definidos e implementando um sistema de votação sobre a saída de maximização da detecção estatística. Yang et al. [30] propôs um método de detecção baseado na Bondade do Ajuste (GOF). O método GOF visa estabelecer que, para alguma distribuição de probabilidade disponível, um conjunto de dados poderia ser derivado dela. O teste GOF utiliza o teste qui-quadrado ou teste χ^2 [9].

2.1.3.3 Modelos de correspondência

Leeb et al. [31] propôs detectar eventos usando modelos de sinais de potência. Shaw et al. [32] usou um modelo com Distância euclidiana. Basu et al. [33] aplicou Distorção Dinâmica do Tempo para detectar eventos e Liu et al. [34] melhorou esta proposta por aplicar a Variante da Distorção Dinâmica do Tempo para a detecção de Eventos. Os modelos de correspondência podem ser expandidos para tornar a classificação de cargas, mas a dependência da distância Euclidiana pode ser um problema com esses métodos. Por exemplo, como observado por Liu [35], dois sinais similares fora de fase podem ter uma grande distância Euclidiana, complicando a detecção.

2.1.3.4 Métodos de detecção para cargas baseadas em inversores

O algoritmo para a detecção de uma carga baseada em inversor difere daquele de uma carga não inversora devido ao seu processo de inicialização mais complexo. Na Figura 4, os exemplos das formas de onda de potência do processo de inicialização de um ar-condicionado com inversor, aquecedor de água elétrico e refrigerador são mostrados em (a), (b) e (c), respectivamente.

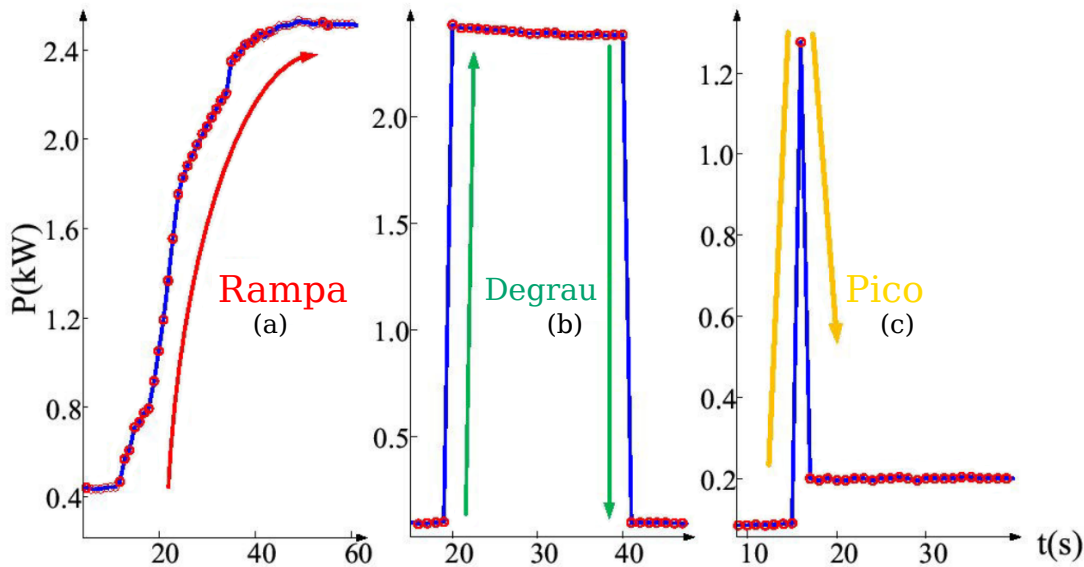


Figura 4 – Diagrama esquemático de três tipos de eventos.

Fonte: Adaptado de Wei et. al.[14]

Detectar eventos de degrau e pico é relativamente simples, já que as mudanças são repentinas; conforme os métodos apresentados na seção 2.1.3. Ainda assim, uma abordagem diferente é necessária para detectar eventos do tipo rampa, já que eles podem levar segundos para terminar, frequentemente coincidindo com outros eventos.

Wei et al. [14] propõem uma detecção de eventos baseada no algoritmo de linearização por partes de curvas [36]. Primeiro, o método de detecção de bordas é usado para detectar e marcar os eventos do tipo degrau e do tipo pico, e os resultados são registrados em uma lista de eventos em ordem sequencial. Cada registro inclui o momento inicial ts e final td do evento, e o incremento de potência ativa ΔP , incremento de potência reativa ΔQ e taxa de mudança de potência K . Os dados adquiridos são então separados em múltiplos segmentos usando as bordas detectadas anteriormente como separadores. Então, o processamento de linearização de segmentação de curva, que é parte do algoritmo de linearização por partes, é executado, e mais subsegmentos são gerados. Os valores ts , td , ΔP e K são calculados a partir dos segmentos e, se ΔP e K forem maiores que limites específicos, o respectivo evento é marcado como tipo rampa e registrado na lista de eventos.

Um pré-processamento é realizado caso outros eventos não-rampa aconteçam durante o evento de rampa, como visto na Figura 5. O ΔP do evento mais novo é subtraído

do evento de rampa mais antigo para tornar o evento de rampa inteiro novamente.

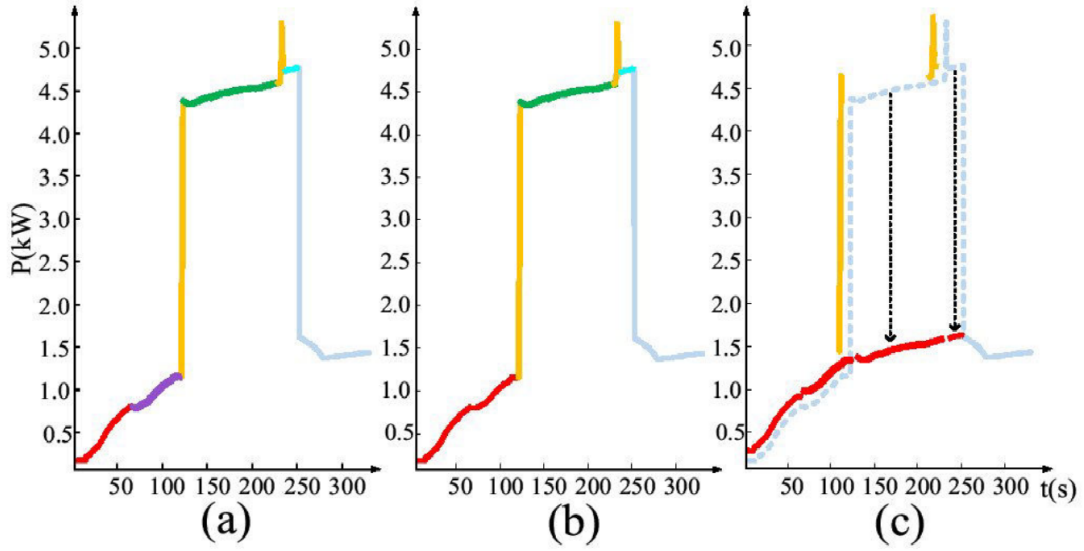


Figura 5 – Diagrama de pré-processamento de eventos.

Fonte: Adaptado de Wei et. al.[14]

Hua et al. [37] propõem usar o algoritmo CUSUM para detecção de eventos. A Figura 6 é usada para ilustrar o algoritmo de detecção de eventos CUSUM. Após o tempo de início (tempo t_0) do evento ser detectado, o processo de detecção do momento de regime permanente (tempo t_4) do evento é inserido, conforme mostrado na Figura 6. A série temporal de potência mudará até que o evento entre em regime permanente.

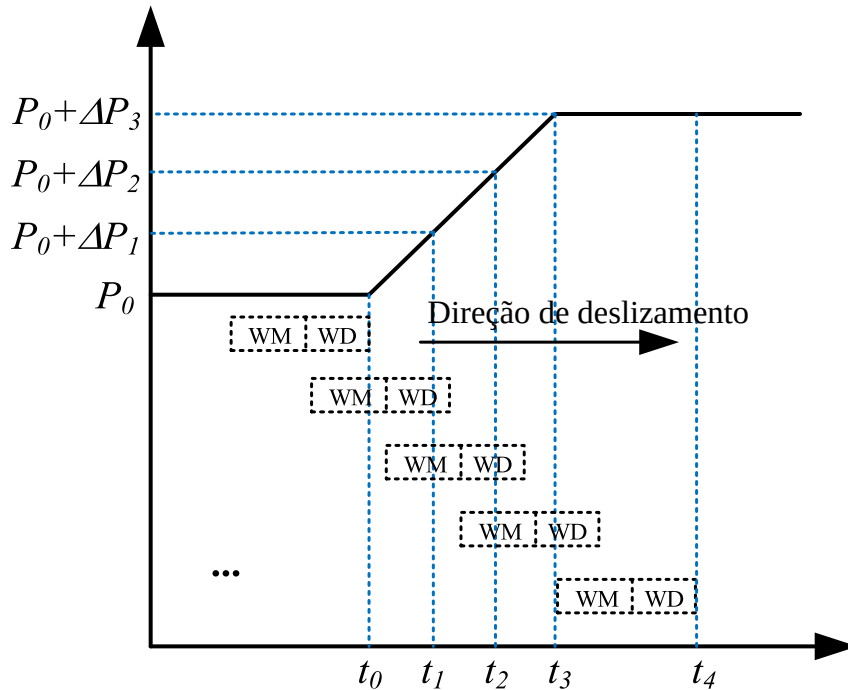


Figura 6 – Detecção de eventos por CUSUM.

Fonte: Adaptado de Hua et. al.[37]

No CUSUM, duas janelas deslizantes consecutivas são definidas, sendo WM uma janela de cálculo média, N_m o número de pontos de amostragem, assumindo que o comprimento é m . WD é a janela de detecção de eventos, e o número de pontos de amostragem é N_m , assumindo que o comprimento é n . Para uma série temporal de potência $P = p(k), k = 1, 2, \dots$, pode-se calcular os valores médios M_m, M_d da janela de cálculo média e da janela de detecção de eventos, conforme mostrado em (2.1) e (2.2).

$$M_m = \frac{\sum_{k=m}^{k+m-1} p(k)}{m}. \quad (2.1)$$

$$M_d = \frac{\sum_{k=m}^{k+m+n-1} p(k)}{m}. \quad (2.2)$$

Então, pode-se definir a quantidade cumulativa g_k^+ de eventos de ativação e a quantidade cumulativa g_k^- de eventos de desativação, conforme mostrado em (2.3) e (2.4).

$$g_k^+ = \max(0, g_{k-1}^+ + M_d - (M_m + \beta)) \quad g_0^+ = 0. \quad (2.3)$$

$$g_k^- = \max(0, g_{k-1}^- + M_d - (M_m - \beta)) \quad g_0^- = 0. \quad (2.4)$$

Em que β representa a quantidade de flutuação da potência em regime permanente. O limite H é usado para determinar se um evento ocorreu. Quando $g_k^+ > H$, indica-se que há um evento de ativação acontecendo, e $g_k^- > H$ indica que há um evento de desativação acontecendo. Quando $0 < g_k^+ < H$, significa que pode haver um evento neste momento. No entanto, como $g_k^+ < H$, não é certo que um evento tenha ocorrido. Então, a variável de atraso de tempo d é introduzida. Seja $d = d + 1$ e calculado g_{k+d}^+ até $g_{k+d}^+ > H$. Neste momento, o evento específico é determinado.

2.1.3.5 Extração de Características

O número de características a ser utilizado e o método de utilização depende do método de extração. Conforme inicialmente descrito por Zeifman e Roth. [17], em geral, NILM macroscópicos monitoram mudanças na assinatura de potência, enquanto NILM microscópicos usam detalhes de forma de onda para extrair características para a desagregação de carga [16]. A Tabela 1 apresenta uma lista de estudos NILM e as características extraídas do ponto de vista dos atributos.

Sadeghianpourhamami et al. [82] categorizam os extratores de características em: definições físicas convencionais, análise em tempo-frequência, trajetórias-VI, extratores baseados em redes neurais profundas e, mais recentemente, a comunidade científica tem apresentado muitos extratores baseados em redes neurais convolucionais profundas [9].

Amostragem	Técnica de amostragem de sinal	Atributos	Referências
Baixa-frequência (Macroscópico)	Assinatura de de energia ativa/reactiva e variações	P, Q	[8, 38, 39, 40, 41, 42]
	Assinatura de potência ativa e variações	P	[43, 44, 45, 46]
	Assinatura de potência e indicadores de potência	$P, I_{RMS}, U_{RMS}, I_{MAX}, U_{MAX}, PF(\lambda)$	[47, 48]
	Assinatura de potência e indicadores de qualidade	$P, PF(\lambda)$	[49]
	Probabilidade de on/off de cargas	$P_{load}^{ON}, P_{load}^{OFF}$	[50]
Alta-frequência (Microscópico)	Pulsos de potência discretos temporais	$P_{pulse}(t)$	[51]
	Decomposição harmônica	$P, Q, H_{1,3,5,...N}$	[52, 53, 54, 55]
	Assinatura de potência e decomposição harmônica	P, THD	[56, 57]
	Assinatura de potência e Teoria de Potência	$P, Q, SouA, H, I_f, CPT_{parcelas}$	[18, 58, 59, 60, 61, 62, 15]
	Wavelets	W_{coef}, W_e	[19, 63, 64, 65, 66, 67, 68, 69]
	Transformada <i>Scattering</i>	$i(t)$	[70, 71, 72]
	Transformada Huang-Hilbert	$h_j(w)$	[73]
	Trajétoria de voltagem e corrente	VI_{traj}	[74, 75, 76, 77]
	Imagens SDP	$P(t)$	[78]
	Imagens de Distorção Harmônica	$i(t)$	[79]
	Ruído harmônico em transiente de potência	$H_f, 36 \leq f \leq 500Hz$	[80, 81]
	Combinação de características independentes	$VI_{traj}, i(t), p(t)$ e outros	[57]

Tabela 1 – Categorização de algumas técnicas de Monitoramento de Carga Não Intrusiva (NILM).

Fonte: Souza et. al.[26]

Nota: P : potência ativa; Q : potência reativa; I_{RMS} : corrente efetiva; U_{RMS} : voltagem efetiva; U_{MAX} : voltagem máxima; I_{MAX} : corrente máxima; $PF(\lambda)$: fator de potência; P_{load}^{ON} : probabilidade de carga ligada; P_{load}^{OFF} : probabilidade de carga desligada; $P_{pulse}(t)$: tempo de pulso de potência; $H_{1,3,5,...N}$: n ésima componente harmônica ($n = 1, 3, \dots, N$); THD : distorção harmônica total de tensão ou corrente; S ou A : potência aparente; H : potência harmônica; I_f : corrente inativa; $CPT_{parcelas}$: parcelas de potência da CPT; W_{coef} : coeficientes wavelet; W_e : coeficientes equivalentes wavelet; VI_{traj} : trajetória de corrente x tensão; H_f : componentes de potência não fundamental; $i(t)$: corrente instantânea; $p(t)$: potência instantânea.

Para Yu et al. [83, 15], é possível adotar três diferentes estratégias para extração de características da carga, sendo uma baseada na análise no domínio do tempo, outra baseada na análise no domínio da frequência e para complementar, a terceira baseada na assinatura de potência (ciclo ON-OFF).

2.1.3.6 Identificação da Carga

A etapa de identificação da carga determina quais cargas estão presentes no sinal agregado, tomando as características como entradas. Geralmente, modelos de classificação com aprendizado de máquina são utilizados para esta etapa. Alguns métodos clássicos usados para identificação de carga são k-Vizinhos mais próximos, árvore de decisão, máquina de vetores de suporte, análise discriminante linear, redes neurais artificiais, mecanismos de atenção e métodos combinados, brevemente detalhados a seguir:

- k-Vizinhos mais Próximos (KNN): este método classifica o exemplo de teste comparando-o com o conjunto de dados de treinamento. A comparação é baseada na distância euclidiana, conforme descrevem Cherkassky e Mulier [84]. Os mesmos destacam que este método depende totalmente das distâncias entre as amostras de teste e o conjunto de treinamento e, por isso, pode ser considerado uma das abordagens de classificação mais simples. No entanto, Lazzaretti et al. [85] apontam que ele requer o

armazenamento de todos os exemplos de treinamento, o que pode ser uma limitação para sistemas embarcados com requisitos de memória mais restritivos [9].

- **Árvore de Decisão (DT):** este método emprega várias divisões binárias concatenadas organizadas em uma estrutura de árvore. Cherkassky e Mulier [84] destacam que cada divisão (nó da árvore) refere-se a um recurso específico e ao valor do parâmetro correspondente para comparação. Na fase de teste, o exemplo é avaliado em cada nó da árvore, e a classe correspondente é a classe majoritária no nó folha. O processo de treinamento e as divisões relacionadas baseiam-se no Ganho de Informação [9].
- **Máquina de Vetores de Suporte (SVM):** este classificador foi proposto por Vapnik [86] para maximizar a margem de separação entre pares de classes, baseado em um modelo linear (hiperplano). Uma das vantagens significativas da formulação SVM é que ela pode ser formulada como uma otimização convexa. Além disso, o problema pode ser definido em termos do produto escalar entre os recursos. Cherkassky e Mulier [84] destacam que, usando o truque do núcleo [87] para tornar os dados linearmente separáveis, o produto escalar pode ser substituído por um núcleo de produto escalar no espaço de recursos usando o núcleo, permitindo a separação não linear entre classes [9].
- **Análise Discriminante Linear:** o LDA é um classificador linear que emprega hiperplanos para diferenciar dados de duas classes diferentes. Ele assume uma distribuição normal com uma matriz de covariância igual e anteriores iguais para ambas as classes. Com isso, o hiperplano separador é definido, reduzindo a dimensionalidade de forma que maximize a separação entre classes e minimize a variância intraclasse. Assim, sua complexidade e, conseqüentemente, seu *overfitting* são reduzidos [84, 9].
- **Redes Neurais Artificiais (RNA):** são uma família de classificadores cuja estrutura e funções são baseadas em redes neurais biológicas [88].

O MLP [88] é uma classe de rede neural *feedforward* compostas por múltiplas camadas de neurônios. Diferentemente do perceptron, esta pode ter uma ou mais camadas ocultas entre a camada de entrada e a de saída. Essa arquitetura permite que o MLP aprenda relações não lineares complexas nos dados, algo que um perceptron não é capaz de executar.

As Redes Neurais Convolucionais (CNN) [89] são um tipo especializado de rede neural para processar com uma topologia conhecida, semelhante a uma grade. O nome “rede neural convolucional” vem do fato que esta rede empregar a operação de convolução.

Redes Neurais Recorrentes (RNN) [89] são classificadores baseados em redes neurais projetados para processar dados sequenciais, onde a ordem da informação é importante. Variações mais recentes como a Memória de curto longo prazo (LSTM)

são usadas para mitigar o problema de gradientes evanescentes que ocorrem em certos algoritmos de RNN [90].

- Mecanismos de Atenção: [91] é uma técnica de aprendizado de máquina que permite aos modelos ponderar dinamicamente a importância de diferentes partes dos dados de entrada. Um mecanismo moderno de atenção como o introduzido por Bahdanau et al. [92] é essencialmente uma média ponderada. O algoritmo *Transformer* [91] dispensa a recorrência e convoluções dos algoritmos de RNN e CNN para se basear somente em mecanismos de atenção com vantagens na paralelização e no tempo de treinamento.
- Métodos Combinados: este método combina diferentes modelos de aprendizado fracos, calculando a média da saída dos classificadores individuais, melhorando a precisão final, conforme explicam Cherkassky e Mulier [84]. É importante notar que os modelos fracos geralmente apresentam uma resposta individual fraca em termos de precisão. No entanto, combinar os vários classificadores tende a melhorar a precisão geral. Um exemplo é o método Floresta Aleatória que combina diversas árvores de decisão para compor uma classificação final. O conjunto utiliza normalmente os seguintes modelos fracos: AdaBoostM1, AdaBoostM2, Bag, GentleBoost, LogitBoost, LPBoost, LSBoost, RobustBoost, RUSBoost, Subspace e TotalBoost [84, 9].

Além dessas estratégias, outras possibilidades são descritas a seguir:

Vavouris et al. [93] utiliza o algoritmo seq2subseq [94] com *Generative Adversarial Network* (GAN) condicional como implementado em [95], usando a mesma abordagem na divisão de dados entre treinamento (60%), validação (10%) e teste (30%), bem como nas etapas de pós-processamento para desagregar cargas em um sistema trifásico.

Gowrienantan et al. [96], propõe utilizar uma rede neural profunda convolucional (CNN baseada na transformada wavelet [97], sequência a sequência de 2 estágios [98]) treinada para cada uma das cargas alvo. Um dispositivo de medição grava cada fase do sistema trifásico e o envia para um servidor na nuvem, onde está a CNN. Todo o processamento é feito na nuvem e o gasto de cada carga é enviado a um aplicativo de celular para o usuário.

Lin et. al. [99] propõe um algoritmo de desagregação baseado em redes neurais de atenção e convolução dilatada empilhada, utilizando o conjunto de dados público UK-DALE [100]. O algoritmo utiliza os dados macroscópicos deste conjunto de dados e os autores afirmam que uma hora de dados de eletricidade podem ser processados em menos de um segundo, mas a plataforma em que os testes foram executados não foi especificada.

Azad et. al. [101] propõe o algoritmo de desagregação usando um modelo de aprendizado profundo com um mecanismo de atenção baseado em *Transformer* e Agrupamento

Temporal. Foi utilizado o Google Colab com as seguintes especificações: 12 GB de RAM, uma GPU T4 com 16 GB de memória e um disco rígido de 78 GB.

Rafiq et. al. [102] utiliza um algoritmo de desagregação baseado em uma rede neural recorrente profunda usando um espaço de entrada multi-recurso. Um método de informação mútua é usado para selecionar os parâmetros elétricos de regime permanente que tinham maior influência no consumo de energia de cada aparelho alvo. Em seguida, os parâmetros de regime permanente selecionados, baseados no espaço de entrada multifunções (MFS), são utilizados para treinar o modelo de memória de longo-curto prazo (LSTM) bidirecional de quatro camadas para cada dispositivo alvo. Por fim, uma técnica de pós-processamento é utilizada na etapa de desagregação para eliminar sequências preditas irrelevantes, aprimorando a precisão da classificação e da estimativa do algoritmo.

Saleh et. al. [73] propõe o uso da Transformada Huang-Hilbert (HHT) para a extração de características no espectro da frequência. A transformada Huang-Hilbert tem uma abordagem adaptativa orientada a dados que não sofre as deficiências das funções de janela da Transformada de Fourier de Curto Prazo (STFT) ou dos vários núcleos das Distribuições Generalizadas de Tempo-Frequência (GTFD) em termos de tratamento de aspectos não lineares de um sinal, e tem uma vantagem sobre a transformada Wavelet, pois não requer a escolha de uma função mãe. Uma limitação da HHT é seu alto uso computacional.

Os avanços mais recentes em sistemas NILM estão em resolver os problemas da baixa disponibilidade de dados de aprendizado rotulados. Para isso, algumas estratégias vêm sendo desenvolvidas, como o aprendizado por transferência em que o sistema NILM se utiliza de dados de aprendizado que são originários de outro sistema NILM [103]. A aprendizagem por transferência federada efetua o aprendizado por transferência sem utilizar dados que expõem a privacidade das residências [104]. A adaptação de domínio estuda as técnicas necessárias à transferência dos dados de aprendizado de uma residência a outra [105]. E existe finalmente o meta aprendizado, que busca resolver o problema de *overfitting* que pode ocorrer quando há uma quantidade pequena de amostras de dados de transferência [106].

2.1.3.7 Desagregação da Potência da Carga

Segundo Hart [8], o consumo total de potência representa a soma das cargas individuais durante sua operação. Considerando que $P_i(t)$ é um vetor que contém toda a informação de potência que todas as cargas do sistema consomem e $e(t)$ um termo de erro, então a potência agregada no tempo t é generalizada como apresentado na equação 2.5.

$$P(t) = \sum_{i=1}^n ((a_i(t) \times P_i(t)) + e(t)).$$

$$a_i(t) = \begin{cases} 1 & \text{se a carga } i \text{ está ligada em } t. \\ 0 & \text{se a carga } i \text{ está desligada no tempo } t. \end{cases} \quad (2.5)$$

Considerando que os passos anteriores definem o momento (*Detecção de Borda de Potência*) e qual carga (*Classificação da Carga*) gerou o evento, o valor médio da potência ativa e o tipo de carga pode determinar o consumo aproximado. Para fazer isso, é necessário formar grupos de eventos que correspondem ao período da operação da carga.

Considerando cada pico como um novo evento, o sistema de NILM usa um banco de dados para gravar amostras de potência ativa durante o período que a carga está ligada. Isto significa que cada carga produzirá um lote de amostras (Souza et al. [26]) que representa sua história de operação.

Para obter um valor mais aproximado do consumo de energia da carga, o algoritmo de desagregação proposto por Hart [8], extrai o valor médio de cada segmento associado com cada classe de carga para avaliar a média de potência ativa durante aquele período. Segmentos que contém mais que uma carga ativada usa o valor histórico de cada uma, como mostrado na Figura 7.

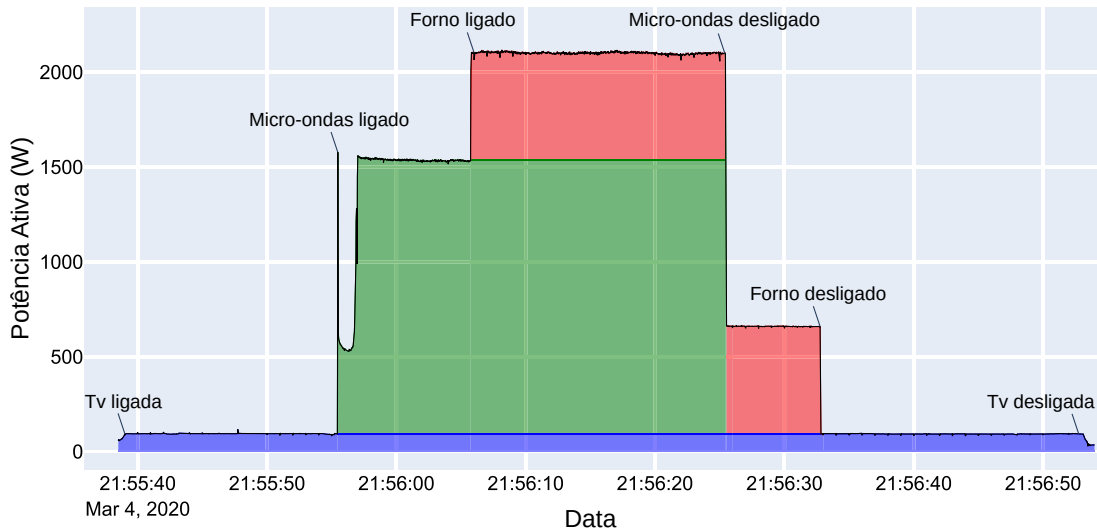


Figura 7 – Exemplo do conceito de NILM com cargas desagregadas.

Fonte: Adaptado de Garcia et. al.[16]

2.2 Deficiências na desagregação para cargas com inversor

Na seção 2.1.3.4 foram revisadas duas metodologias para detecção de cargas baseadas em inversores, propostas por Wei et al. e Hua et al. Nesta seção serão mostrados

alguns cenários em que essas propostas podem ser melhoradas. Para isso foi feito um estudo de caso utilizando um ar-condicionado modelo Samsung Digital Inverter monofásico com capacidade de resfriamento de 12000 BTU/h.

Na Figura 8 é mostrado o consumo de potência ativa do início desta carga. É possível notar alguns picos de potência no início da medição, degraus de aumento de consumo conforme o aparelho passa por seu processo de início e aumentos graduais de potência intercalados por degraus de redução. A região (a) na Figura mostra um aumento repentino de potência que provavelmente representa o acionamento de algum subsistema do aparelho, e a região (b) mostra variações rápidas, provavelmente causadas pelo funcionamento do inversor do aparelho. A região (b) visualmente variou mais que a região (a). A Figura 9 mostra uma versão ampliada da Figura 8, na qual uma grande variação da potência ativa pode ser notada, maior do que a variação média potência.

Um tratamento extra é necessário ao alimentar o detector de eventos para que este não confunda as variações rápidas descritas anteriormente como sendo vários eventos de ativação e desativação, que podem confundir o algoritmo de aprendizado de máquina na etapa de identificação, possivelmente atribuindo os eventos espúrios a outra carga. Mesmo que ocorra a correta identificação pela atribuição dos eventos espúrios com o funcionamento normal do aparelho, a etapa de identificação estará sendo solicitada desnecessariamente já que não houve nenhum acionamento ou desligamento real.

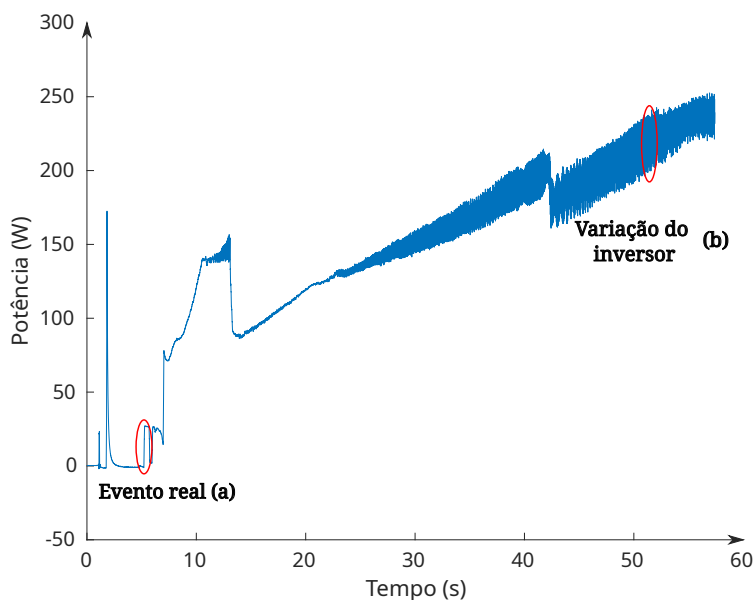


Figura 8 – Potência ativa de um ar-condicionado com inversor.

Fonte: Autoria própria

As variações rápidas de potência vistas nas Figuras 8 e 9 podem ser mitigadas reduzindo a frequência de amostragem de medição. Os autores dos estudos citados utilizam o tempo de amostragem de um segundo. A desvantagem desta abordagem é que menos

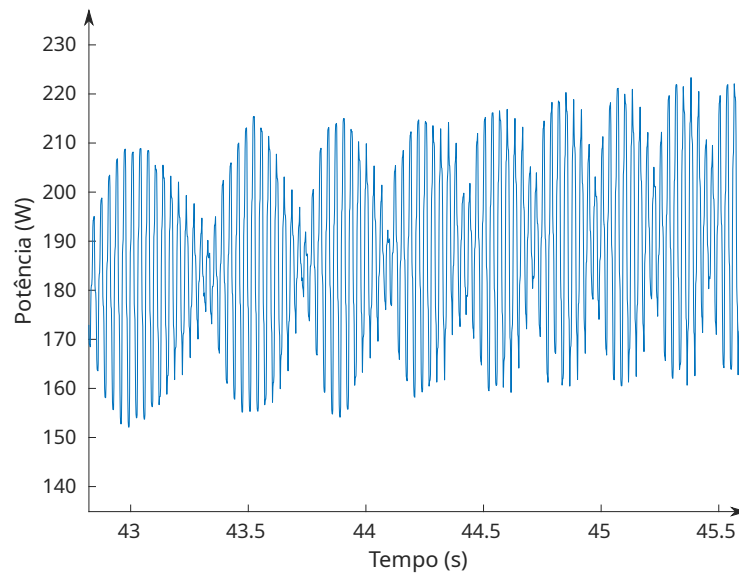


Figura 9 – Ampliação da potência ativa de um ar-condicionado com inversor.

Fonte: Autoria própria

informação pode ser extraída ao analisar a assinatura de potência das cargas, principalmente se for utilizada a Teoria da Potência Conservativa (CPT), apresentada na seção 2.3, para extração de características elétricas, que necessita de ciclos completos da forma de onda de tensão e corrente alternadas.

O tempo de processamento necessário para executar todo o processo de desagregação é outra questão que pode ser melhorada. Nos artigos de Wei et al. e Hua et al. não foram identificadas as informações de tempo de execução da tarefa, inviabilizando uma avaliação mais detalhada, mas Hua et al. mencionam em seu artigo que a sua metodologia “pode alcançar o monitoramento próximo ao tempo real [37]”.

No entanto, há um problema com a detecção de eventos simultâneos do tipo rampa. Os algoritmos propostos por Wei et al. e Hua et al. não incluem dois ou mais eventos do tipo rampa ocorrendo aproximadamente ao mesmo tempo. Eventos subsequentes e simultâneos do tipo rampa podem ser detectados analisando o aumento da rampa causado por eles, uma análise mais aprofundada é necessária após a detecção do aumento para determinar a carga.

Há também uma questão sobre a quantidade de dados que precisa ser armazenada para executar a detecção de eventos usando o método proposto por Wei et al. Este método específico de detecção de eventos precisa de uma quantidade arbitrária de memória para realizar a tarefa. Ele deve registrar um número arbitrário de eventos do tipo degrau e do tipo pico antes de executar a linearização da segmentação da curva para gerar os segmentos necessários para encontrar os eventos do tipo rampa. É necessário manter uma quantidade não especificada do histórico de potência ativa e reativa para executar a tarefa. Isso torna

difícil para um engenheiro de *hardware* inferir a quantidade de memória necessária para implementar este método de detecção de eventos. É especialmente problemático se um sistema de aquisição de alta taxa de amostragem for empregado.

Embora não exista menção direta a isso, os algoritmos apresentados guardam uma grande quantidade não especificada do histórico de consumo agregado das cargas para analisar, o que tem um impacto na escolha do equipamento de análise, especialmente se o tempo de captura for microscópico.

Os métodos apresentados pelos autores não são projetados para funcionar em equipamentos de ar-condicionado com inversores trifásicos. Esses ar-condicionados são mais comuns em edifícios comerciais, limitando sua utilização em tais instalações.

Os algoritmos de detecção de eventos para cargas com inversor apresentados na seção 2.1.3.4 são inadequados sem tratamento extra para cargas com alta variação e sistemas com altas taxas de amostragem (métodos microscópicos), porque eles podem confundir o algoritmo de detecção de eventos, criando eventos falsos.

2.3 A Teoria da Potência Conservativa

A CPT – Conservative Power Theory – foi proposta por Tenti et al. [107, 108] e define grandezas elétricas chamadas de conservativas, as quais são calculadas a partir do produto escalar entre tensões e correntes (ou suas derivadas e integrais imparciais temporais), de um circuito com “m” fases, no qual aquelas tensões e correntes, assim como suas derivadas e integrais imparciais, satisfazem as leis de tensões e correntes de Kirchhoff e, conseqüentemente, o teorema de Tellegen (Teorema da Conservação de Energia).

São definidas duas grandezas conservativas principais: a potência ativa instantânea $p(t)$ e a energia reativa instantânea $w(t)$. Além disso, a CPT, com base nas definições de potência ativa e energia reativa, define uma decomposição da corrente em parcelas ortogonais que visam representar os fenômenos físicos (características do circuito) provocados pelas cargas do circuito (elementos lineares e não lineares). Estas parcelas são descritas a seguir: Corrente ativa i_a , aquela responsável por transferir a mínima potência ativa por fase no circuito; Corrente reativa i_r , aquela responsável pelo fluxo mínimo de energia reativa por fase no circuito; Corrente residual i_v , aquela que não transfere potência ativa nem energia reativa (não linearidades da carga); Corrente ativa balanceada i_a^b , aquela responsável pela transferência mínima da potência ativa coletiva da fonte para a carga, além de ser proporcional e estar em fase com a tensão; Corrente reativa balanceada i_r^b , aquela responsável pelo fluxo mínimo de energia reativa coletiva no circuito, e ela pode ser distorcida e está defasada em relação à tensão; e Corrente de desbalance $i^u = i_a^u + i_r^u$.

Para cada parcela de corrente está associado um termo de potência, obtido pelo

produto entre o valor eficaz da tensão e o valor eficaz da corrente correspondente: Potência ativa $P = UI_a^b$, aquela tida como o consumo médio de potência do circuito, representando a realização de trabalho útil na carga; Potência reativa $Q = UI_r^b$, que está relacionada à defasagem entre tensões e correntes, a qual pode ser provocada pela presença de elementos armazenadores de energia ou por cargas eletrônicas; Potência de desbalanço $N = U\sqrt{I_a^{u^2} + I_r^{u^2}}$, que representa o desbalanço da carga; Potência de distorção $D = UI_v$, a parcela de potência consumida devido a presença de não-linearidades no circuito, não realiza trabalho útil nem caracteriza o armazenamento de energia, e a Potência aparente $A = UI$ que é a potência total do circuito.

Finalmente são definidos alguns fatores, apresentados a seguir: Fator de potência λ que relaciona a corrente ativa balanceada com todas as outras parcelas de corrente, através da corrente coletiva total, e resulta unitário somente no caso de circuito puramente resistivo balanceado; Fator de não linearidade λ_D , que representa a presença de potência de distorção, ou seja, correntes que não contribuem com a potência ativa nem a reativa, mas causam perdas na distribuição; o Fator de reatividade λ_Q , revela a presença de energia reativa em indutores, capacitores ou por efeito de cargas não lineares; e o Fator de assimetria λ_U , que representa a presença de potência de desbalanço na carga.

2.4 Desagregação de cargas usando a CPT

Souza [7] demonstrou em seus estudos que, utilizando a Teoria da Potência Conservativa (CPT) [107] para a extração de características elétricas e utilizando o algoritmo de classificação KNN para a identificação da carga, foi possível obter uma boa exatidão na desagregação aliado a um relativo baixo custo computacional. A Tabela 2 contém as informações de exatidão mencionadas anteriormente.

Tabela 2 – Resultados dos métodos por tempo de execução e exatidão.

Método	Exatidão (s)	Tempo médio de execução do treinamento (s)	Tempo médio de execução da classificação (s)	Tempo médio de execução da classificação de uma amostra (s)
OPF	0.9982 ± 0.0014	9.7074 ± 1.0983	4.8087 ± 0.2339	0.0048 ± 0.0146
OPF (PCA)	0.9971 ± 0.0005	9.5534 ± 1.2298	3.8482 ± 0.3801	0.0033 ± 0.0101
MLP	0.8875 ± 0.1192	8.2209 ± 1.3394	0.0883 ± 0.0127	0.0377 ± 0.0114
MLP (PCA)	0.9122 ± 0.0738	7.9821 ± 1.2287	0.1179 ± 0.0101	0.0389 ± 0.0093
SVM	0.9098 ± 0.0382	0.0578 ± 0.0092	0.1175 ± 0.0285	0.0459 ± 0.0193
SVM (PCA)	0.9152 ± 0.0302	0.0681 ± 0.0083	0.0994 ± 0.0184	0.0567 ± 0.0115
KNN	0.9987 ± 0.0011	Não existente	0.0399 ± 0.0093	0.0073 ± 0.0018
KNN (PCA)	0.9988 ± 0.0009	Não existente	0.0384 ± 0.0083	0.0101 ± 0.0077
DT	0.9161 ± 0.0228	0.0145 ± 0.0098	0.0297 ± 0.0018	0.0160 ± 0.0088
DT (PCA)	0.9227 ± 0.0294	0.0271 ± 0.0081	0.0108 ± 0.0022	0.0073 ± 0.0010

Nota: OPF: Optimum Path Forrest; PCA: Principal Component Analysis; MLP: Multi Layer Perceptron; SVM: Support Vector Machine; KNN: K-Nearest Neighbors; DT: Decicion Tree.

Fonte: Souza et. al.[7]

O algoritmo de KNN foi o mais eficiente para o conjunto de dados que o autor testou, por não ter a etapa de treinamento e por usar um tempo computacional baixo na etapa de classificação.

Neste capítulo foram revisadas a metodologia NILM, seus algoritmos mais comuns, as metodologias existentes de desagregação de cargas baseadas em inversor e suas deficiências. No próximo capítulo será apresentada uma nova metodologia para desagregar cargas baseadas em inversor, juntamente com o conjunto de dados usado e uma implementação em C++ do algoritmo.

3 Metodologia

Este capítulo apresenta a metodologia proposta para a desagregação de cargas com motores acionados por inversores. Nas próximas seções serão apresentados os conjuntos de dados utilizados para a análise e uma nova estratégia para a extração de características para as cargas com inversor que necessitam de um tratamento extra devido à sua maior variabilidade. Serão propostos novos algoritmos para a detecção de eventos de acionamento e desligamento de cargas, considerando que cargas com inversores frequentemente aumentam e diminuem o seu padrão de consumo aleatoriamente e em diferentes velocidades. Serão apresentadas discussões sobre os resultados dos algoritmos apresentados e ferramentas implementadas no Simulink/MATLAB (versão 2024a) que serão utilizadas para o teste e validação da metodologia proposta.

3.1 Conjunto de dados de análise

O conjunto de dados foi criado medindo formas de onda de tensão e corrente individuais trifásicas de vários dispositivos, como *air fryers*, fornos de micro-ondas, fornos elétricos, ar-condicionado, aspiradores de pó, *freezer*, TV, secadoras e máquinas de lavar. Para alguns dispositivos mencionados anteriormente, além dos dados individuais foram coletados dados agregados desses dispositivos a outros presentes no mesmo ponto de acoplamento (nesse caso, o ponto de acoplamento comum, o PAC), para testes do sistema de NILM, no caso de haver cargas desconhecidas conectadas na rede. Embora as cargas medidas tenham sido monofásicas e bifásicas, as instalações nas quais as cargas foram monitoradas eram trifásicas.

3.1.1 Equipamento usado na aquisição de dados

O analisador de qualidade de energia AIW-PEX, fabricado pela Sigmasys Engenharia, foi utilizado para efetuar a aquisição do conjunto de dados. Na Figura 10 é apresentado o medidor juntamente com seus acessórios. Este medidor foi homologado pela ONS para utilização em campanhas de medição de harmônicos.

O AIW-PEX atende a norma internacional IEC 61000-4-30 [109], sendo classificado como um medidor Classe A. A medição dos indicadores de Qualidade da Energia Elétrica atende as recomendações das normas IEC 61.000-4-7 [110], IEC 61.000-4-15 [111] e IEC 61.000-4-30, assim como do módulo 8 do PRODIST [112]. Na Tabela 3 são apresentadas as principais características técnicas do medidor.



Figura 10 – Dispositivo usado na aquisição do conjunto de dados.

Fonte: <<https://pqsys.sigmasys.com.br>>

Tabela 3 – Especificações técnicas do medidor AIW-PEX.

Característica	Especificação
Alimentação	80 – 600 VAC
Bateria	12 VDC
Medição de tensão	20 – 600 VAC (3F + N + GND)
Medição de corrente	0,1 – 100 A (Split core) 1 – 5000 A (Cintas flexíveis)
Armazenamento interno	16 GB
Memória externa (opcional)	Até 2TB (Cartão SD)
Frequência	50 – 60 Hz
Conectividade	Ethernet, Wi-Fi (Cliente e AP), Dados Móveis (4G, GPRS)
Geolocalização	GPS

Fonte: <<https://pqsys.sigmasys.com.br>>

O medidor AIW-PEX tem 8 canais de medição: 4 canais para tensão mais o terra e 4 canais para corrente. Cada canal de tensão e corrente tem 16bits de resolução no A/D e amostragem de 1024 amostras por ciclo, resultando em 61440Hz de taxa real de amostragem se configurado para funcionar em redes de 60Hz ou 51200Hz se configurado para redes de 50Hz. Todos os canais são amostrados simultaneamente. Os erros máximos garantidos pelo fabricante, para cada parâmetro mensurável, estão apresentados na Tabela 4.

Tabela 4 – Precisão das grandezas mensuráveis.

Característica	Especificação
Tensão	$\pm 0,1 \%$
Corrente	$\pm 0,1 \%$ do fundo de escala
Frequência	$\pm 0,01 \text{ Hz}$
Distorções harmônicas	$\pm 5,0 \%$
Flutuação de tensão	$\pm 5,0 \%$
Desequilíbrio de tensão	$\pm 0,15 \%$
Eventos	Janela 1/2 Ciclo

Fonte: <<https://pqsys.sigmasys.com.br>>

O processador principal do analisador tem as seguintes especificações:

- Processador Allwinner A40i Quad-Core ARM Cortex-A7 @ 1.2 GHz;
- Acelerador NEON com ponto flutuante integrado ao processador;
- Memória DRAM de 2GB DDR3L, Memória Flash eMMC de 16GB.

Para o propósito desse trabalho, foi utilizado o módulo de oscilografia com um cartão SD de 64GB para a captura das formas de onda, sendo possível gravar até 12 horas ininterruptas. Dados de tensões, correntes e harmônicas foram capturados em pacotes agregados de 10 minutos (conforme a norma IEC 61.000-4-30) e 1 minuto para facilitar a localização da atividade das cargas. O medidor fornece as magnitudes das componentes harmônicas individuais até a 50^a ordem.

3.1.2 Formato dos dados do conjunto de dados

Foi criado um formato de arquivo específico para a gravação do conjunto de dados. Este formato utiliza a DFT (sigla em inglês para Transformada Discreta de Fourier) e o algoritmo zlib [113, 114, 115] para compactar as quatro formas de onda de tensão e corrente que o equipamento consegue medir, mais a hora da aquisição. O software de aquisição e compactação foi desenvolvido em C++ e está embutido no medidor AIW-PEX. O software de descompactação foi desenvolvido em *Python*. Os detalhes sobre o formato constam no apêndice C.

3.1.3 Descrição de conjunto de dados agregados

Além dos dados de forma de onda, foram adquiridos dados agregados em 1 minuto e 10 minutos com a data da aquisição, valores de tensão, corrente, potências ativas, reativas e componentes harmônicas até a 50^a. Os dados agregados de 10 minutos atendem às normas IEC 61.000-4-7, IEC 61.000-4-15 e IEC 61.000-4-30. O formato dos arquivos gravados é o .csv.

Estes arquivos foram incluídos no conjunto de dados para facilitar a localização do ligamento e desligamento das cargas nos arquivos de forma de onda descritos no apêndice C. O nome do arquivo contém as informações do tempo de agregação e data aproximada da primeira aquisição. Tomando, por exemplo, o nome `measurements-1min_2024-04-25_23_27_04.csv`, o nome do arquivo é `measurements`, o tempo de agregamento é de um minuto e a data da primeira aquisição é 25/04/2024 às 23:27:04.

Para algumas aquisições, também foi criado um arquivo com os eventos de VTCD que ocorreram. Estes contêm as informações da queda ou aumento da tensão. A necessidade da inclusão desses dados foi para que, durante o aprendizado e desagregação, seja possível distinguir um comportamento anômalo da carga. Estes arquivos estão em formato csv e seus nomes contêm a data aproximada do início do período de aquisição. Estes arquivos começam pelo nome `measurements_events`. Por exemplo, o nome `measurements_events_2024-04-25_23_29_56.csv` se refere aos eventos ocorridos a partir de 25/04/2024 às 23:29:56.

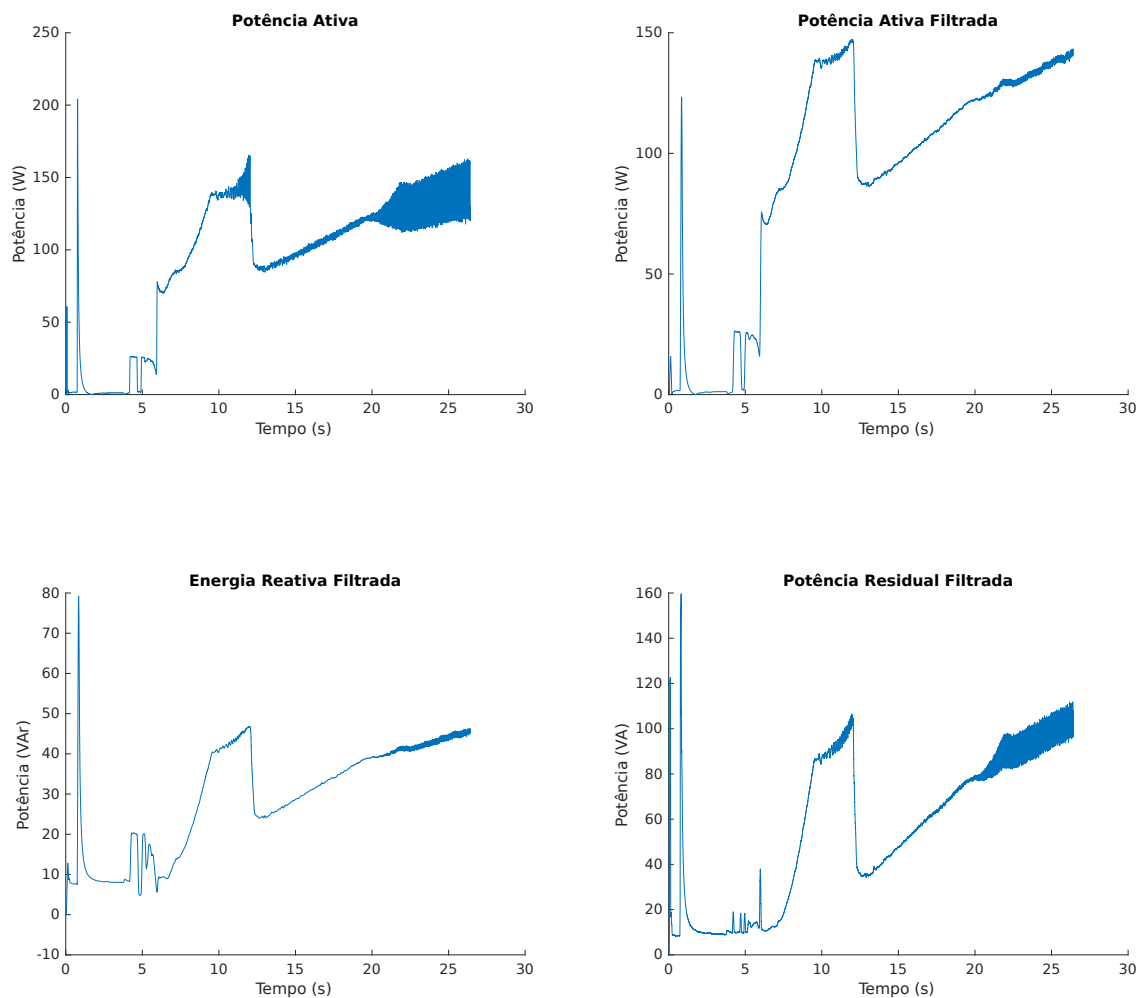
Finalmente, foram inclusos arquivos de texto, no formato `.txt`, com a descrição dos momentos aproximados nos quais as cargas foram acionadas, alguma configuração aplicada na carga ou alguma ação que pudesse alterar o seu padrão de consumo. Por exemplo, os momentos da alteração do *setpoint* de um ar-condicionado ou o fechamento da entrada de ar de um aspirador de pó.

3.1.4 Descrição das cargas coletadas

Nesta seção são apresentadas as cargas adquiridas que foram usadas neste trabalho e também outras cargas que não foram utilizadas. Há dados agregados de 10 minutos, 1 minuto, informações de VTCD e as formas de onda de tensão e corrente no tempo. Os dados agregados de 10 minutos e 1 minuto são usados para localizar os respectivos dados de forma de onda, já que seus respectivos arquivos são menores. Para algumas cargas, existem os dados de formas de onda de tensão e corrente individualizados e também dados com outras cargas agregadas, separados em diretórios 'individual' e 'aggregate', respectivamente. O arquivo 'log.txt', que pode estar presente em alguns diretórios, contém as combinações de cargas agregadas para estas medições. Os gráficos de potência são do início da carga e representam, respectivamente: a potência ativa da CPT, a potência ativa da CPT filtrada, a energia reativa da CPT filtrada e a potência residual (V) da CPT filtrada. As razões da escolha da CPT e estratégias de filtro serão explicadas na seção 3.2.

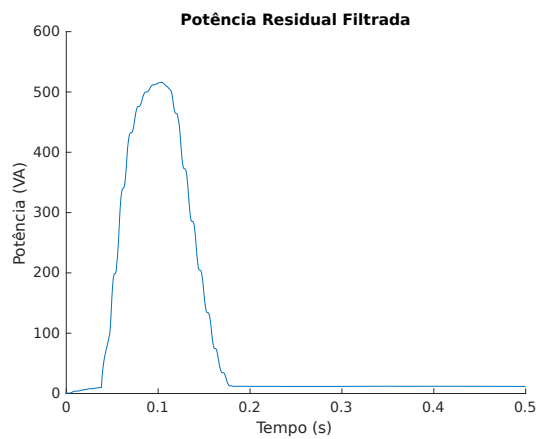
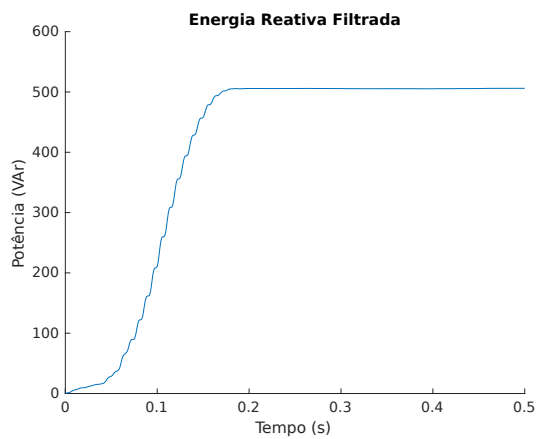
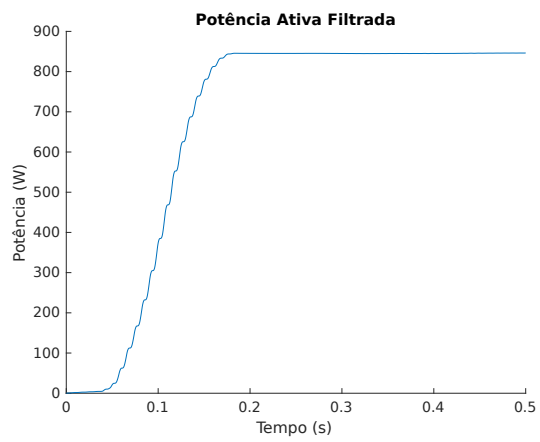
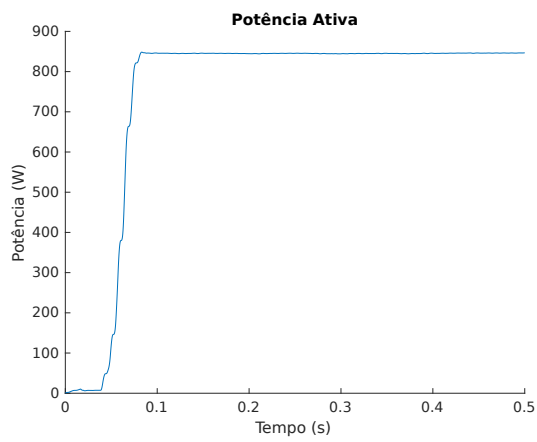
Estes dados são utilizados nas próximas etapas da metodologia para o teste e validação de todos os algoritmos desenvolvidos.

- Ar-condicionado:
 - Marca: Samsung Digital Inverter 12000Btu/h
 - Período da captura: 25-04-2024, das 11:44 até 12:43
 - Diretório: 'wave/House/individual/repack'
 - Diretório: 'wave/Air Conditioning 31_08_2024/individual/repack'
 - Diretório: 'wave/Air Conditioning (House) 03_09_2024/individual/repack'
 - Gráficos de potência:



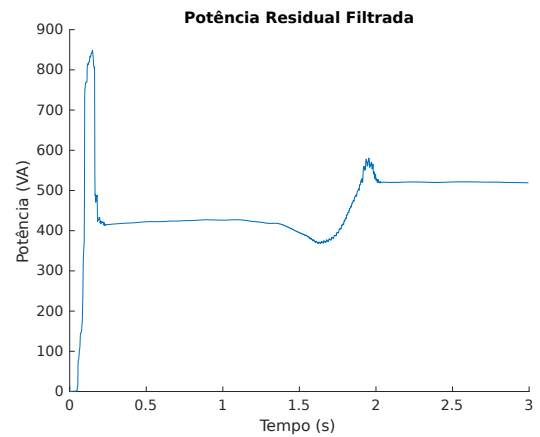
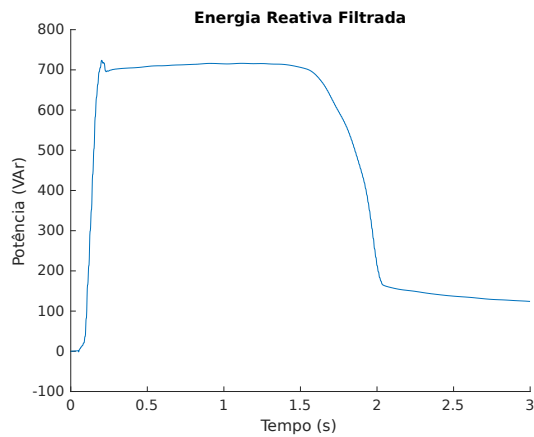
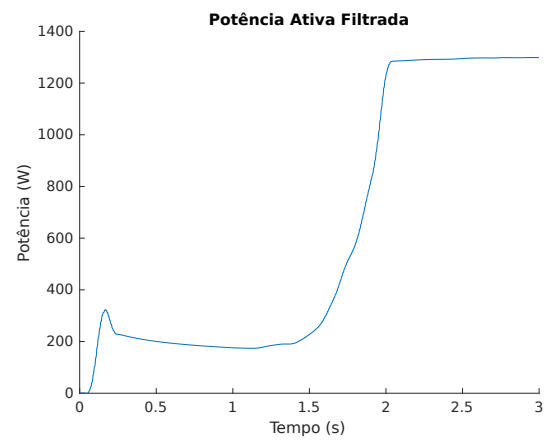
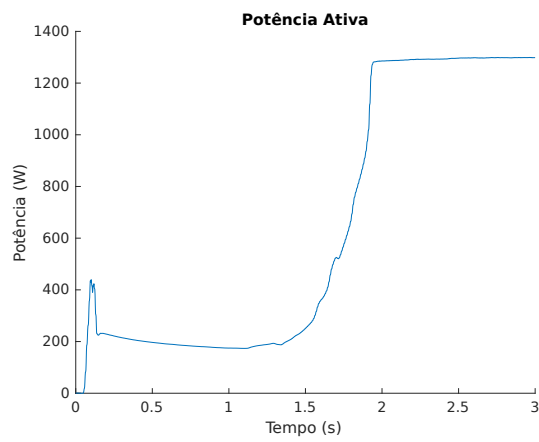
Fonte: Autoria própria.

- *Air fryer*:
 - Marca: Mondial Grand Family 5,5L
 - Período da captura: 25-04-2024, das 09:45:00 até 9:52
 - Diretório: 'wave/House/individual/repack'
 - Gráficos de potência:



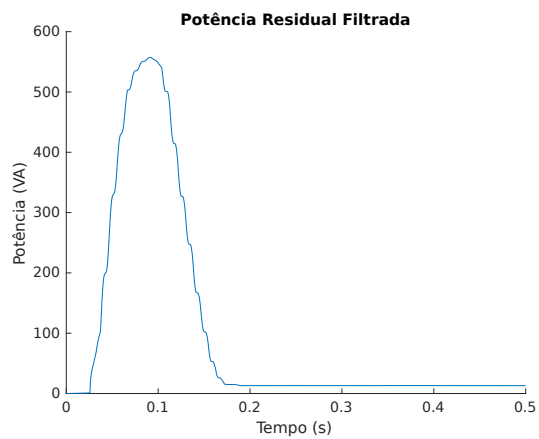
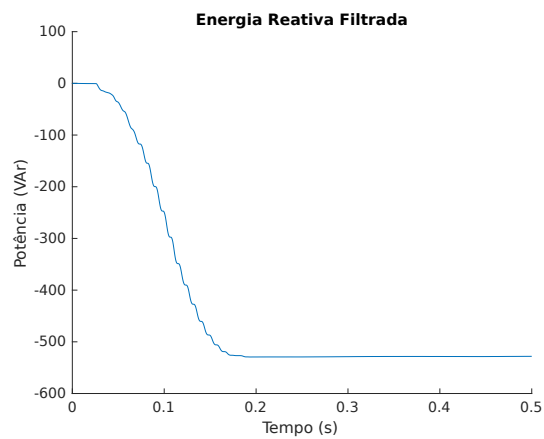
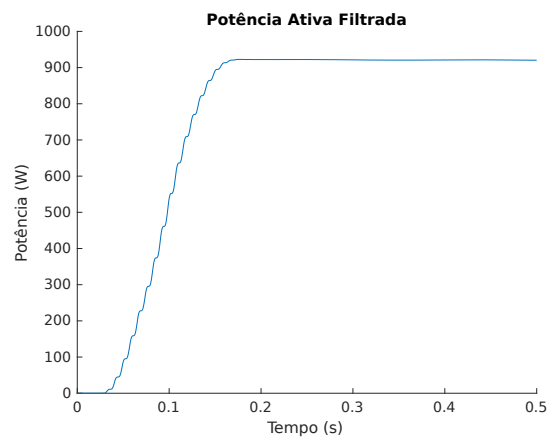
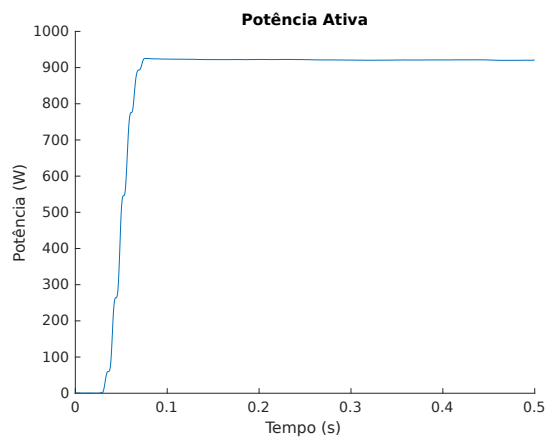
Fonte: Autoria própria.

- Micro-ondas:
 - Marca: Panasonic NN-GT6965
 - Período da captura: 25-04-2024, das 9:51 até 10:16
 - Diretório: 'wave/House/individual/repack'
 - Gráficos de potência:



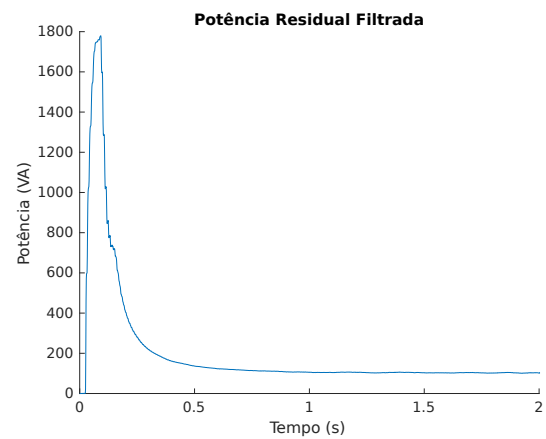
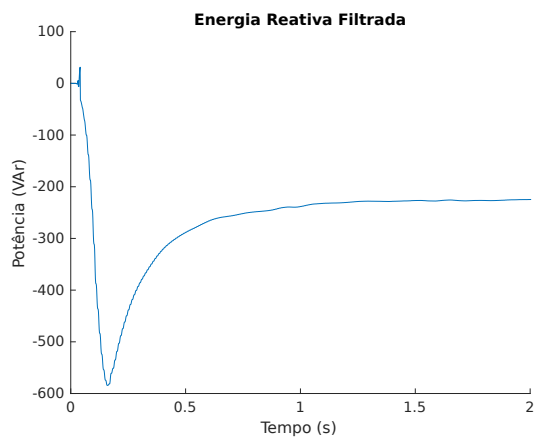
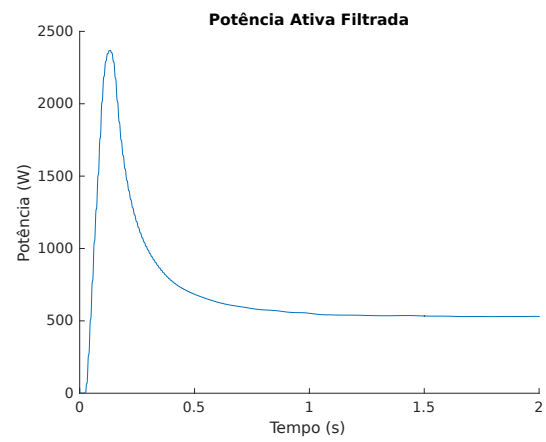
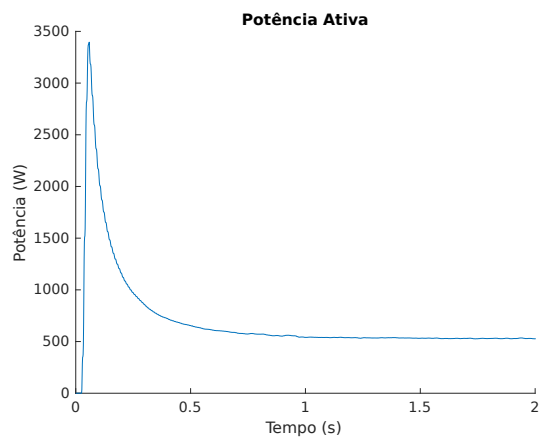
Fonte: Autoria própria.

- Forno Elétrico:
 - Marca: Electrolux 2000W
 - Período da captura: 25-04-2024, das 10:40 até 11:31
 - Diretório: 'wave/House/individual/repack'
 - Gráficos de potência:



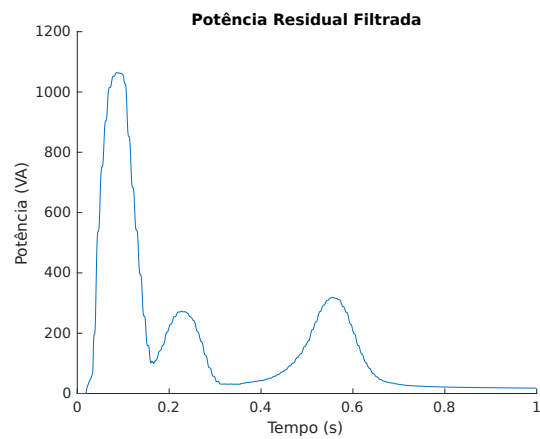
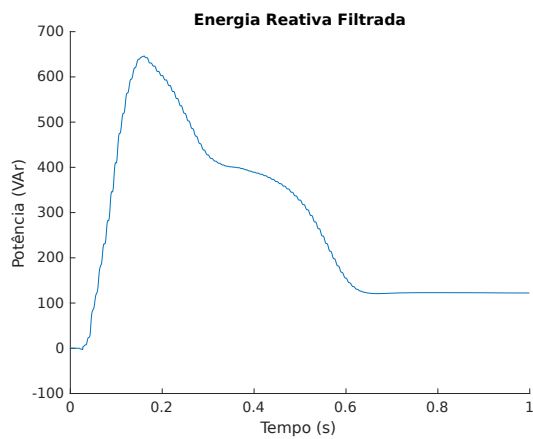
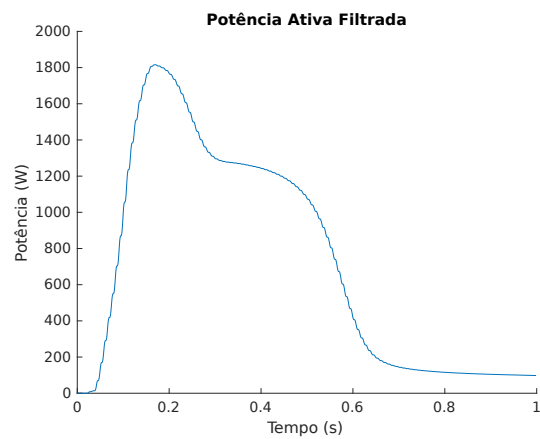
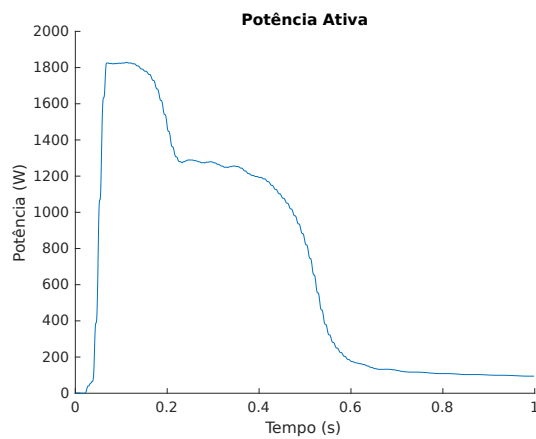
Fonte: Autoria própria.

- Aspirador de pó:
 - Marca: Electrolux 1800W
 - Período da captura: 25-04-2024, das 12:39 até 12:45
 - Diretório: 'wave/House/individual/repack'
 - Gráficos de potência:



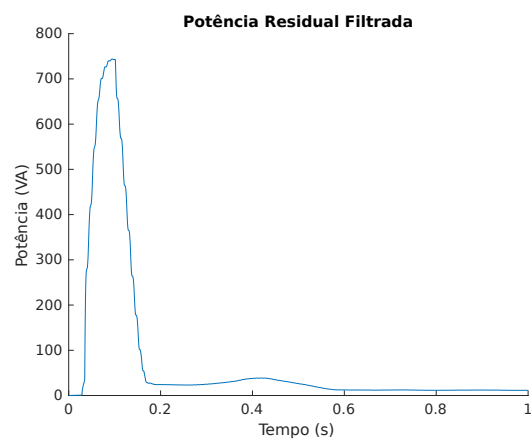
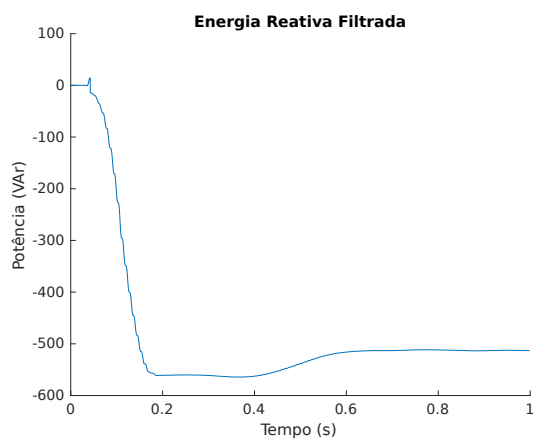
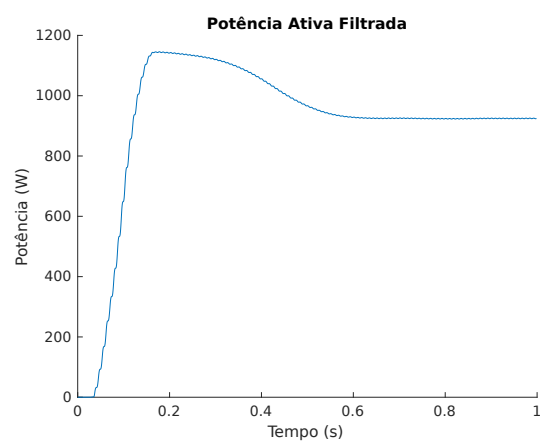
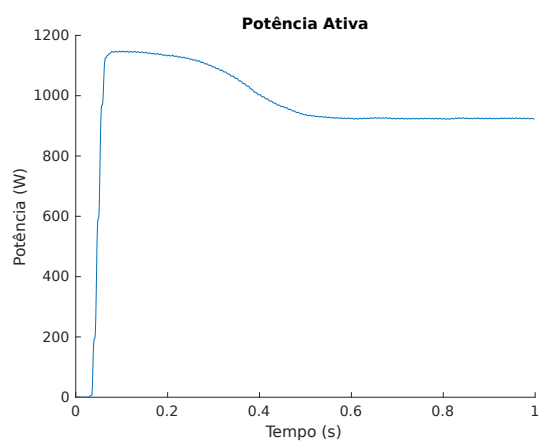
Fonte: Autoria própria.

- *Freezer*:
 - Marca: Midea RCFB21
 - Período da captura: 25-04-2024, das 12:52 até 13:39
 - Diretório: 'wave/House/individual/repack'
 - Gráficos de potência:



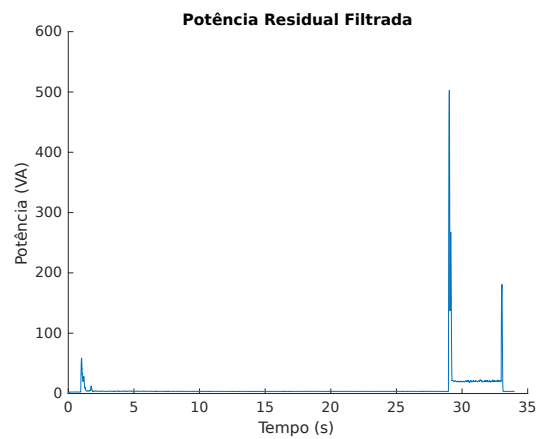
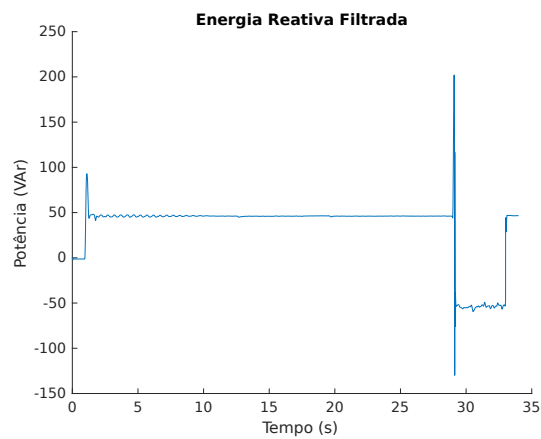
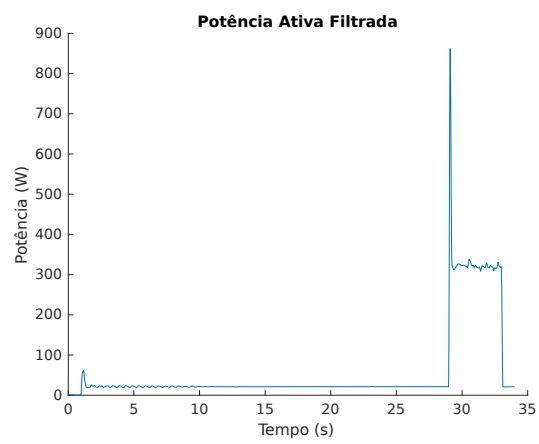
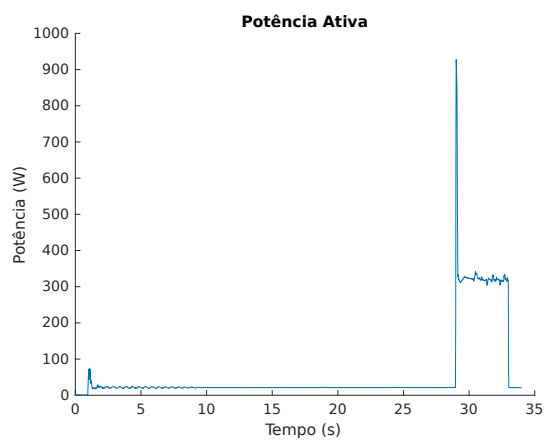
Fonte: Autoria própria.

- Secadora:
 - Marca: Brastemp Ative 2000W
 - Período da captura: 25-04-2024, das 14:02 até 14:06
 - Diretório: 'wave/House/individual/repack'
 - Gráficos de potência:



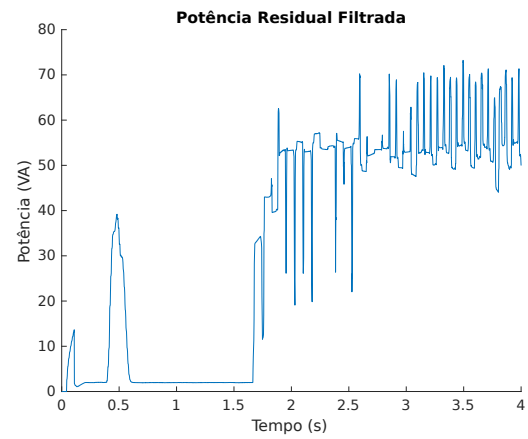
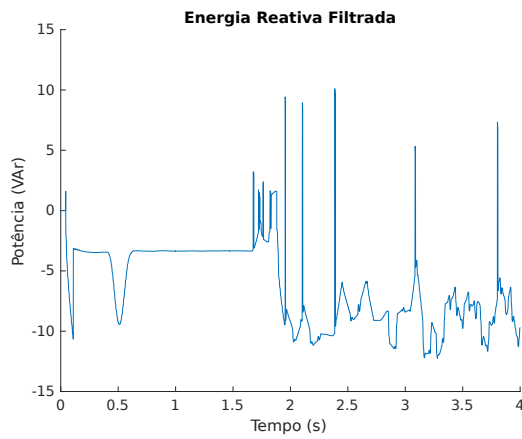
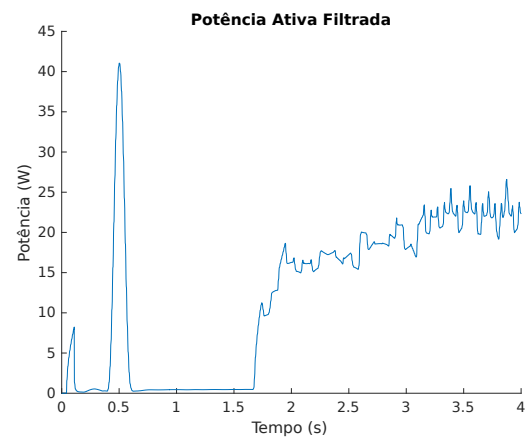
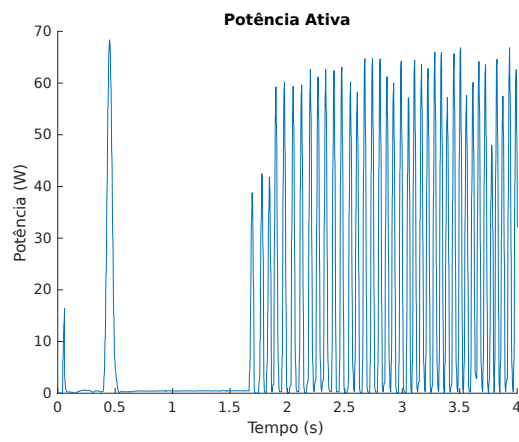
Fonte: Autoria própria.

- Máquina de lavar:
 - Marca: Panasonic NAF16085
 - Período da captura: 25-04-2024, das 14:14 até 14:16
 - Diretório: 'wave/House/individual/repack'
 - Gráficos de potência:



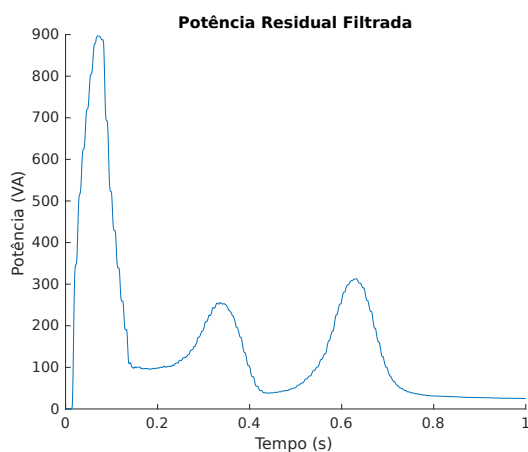
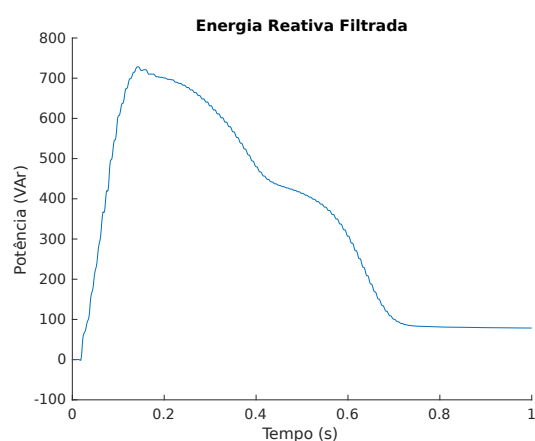
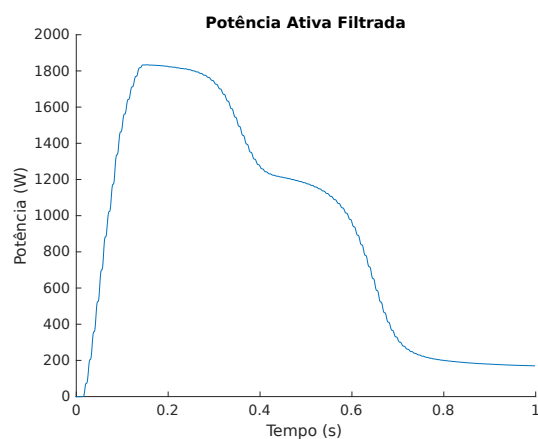
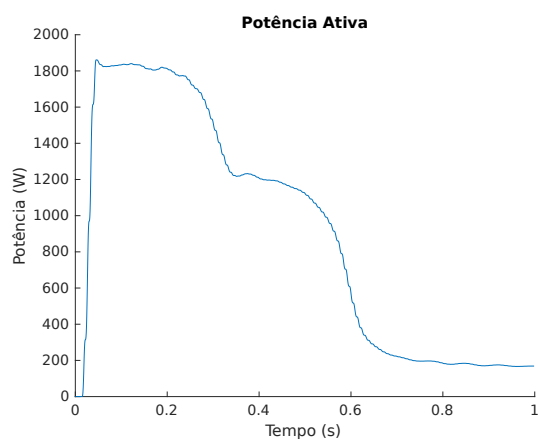
Fonte: Autoria própria.

- TV:
 - Marca: Samsung 65”
 - Período da captura: 25-04-2024, das 13:41 até 13:49
 - Diretório: 'wave/House/individual/repack'
 - Gráficos de potência:



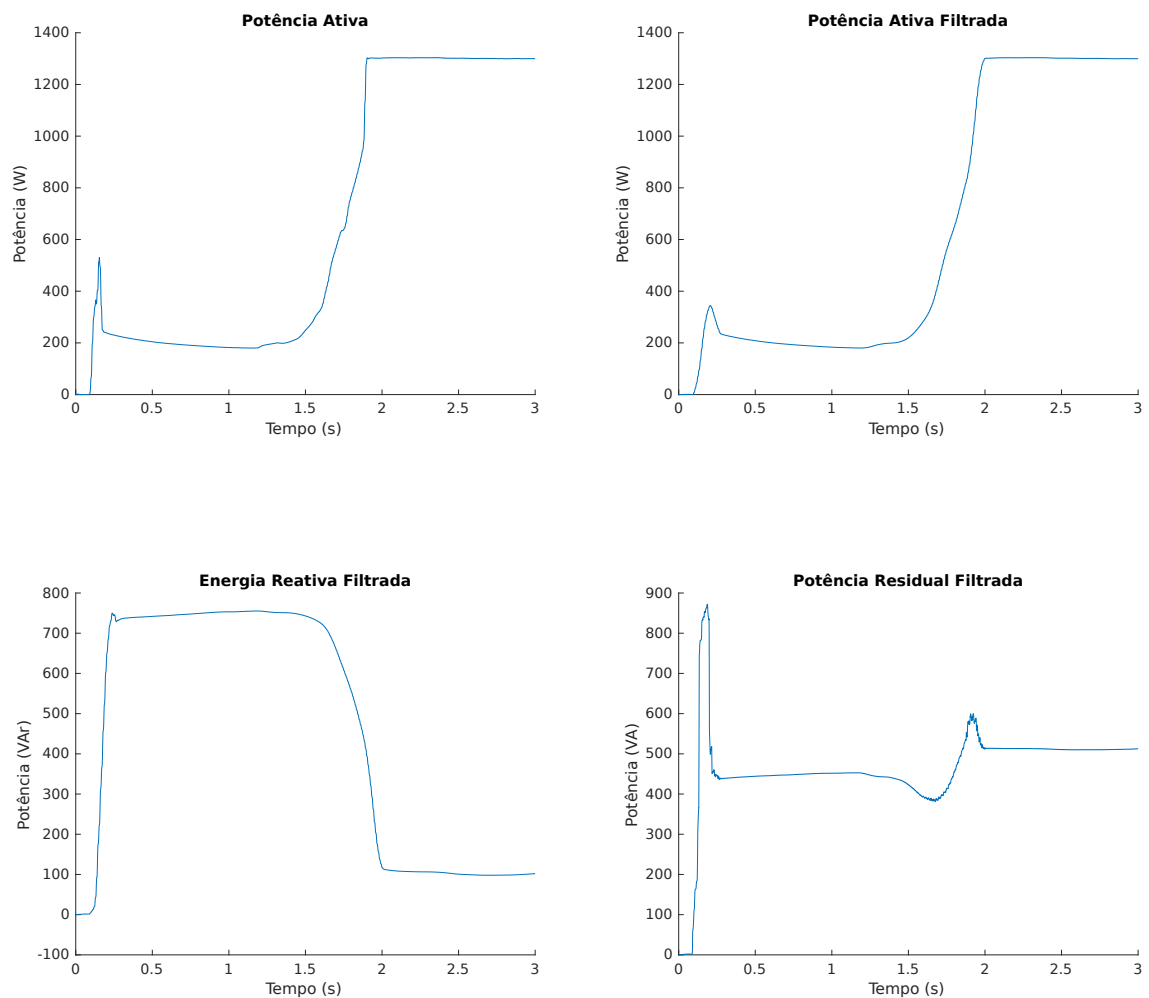
Fonte: Autoria própria.

- Refrigerador:
 - Marca: Electrolux DW42X Frost-Free
 - Período da captura: 29-11-2023, das 04:00 até 16:00
 - Período da captura: 13-12-2023, das 16:00 até 04:00
 - Diretório: 'wave/Refrigerator-day/repack'
 - Diretório: 'wave/Refrigerator-night/repack'
 - Gráficos de potência:



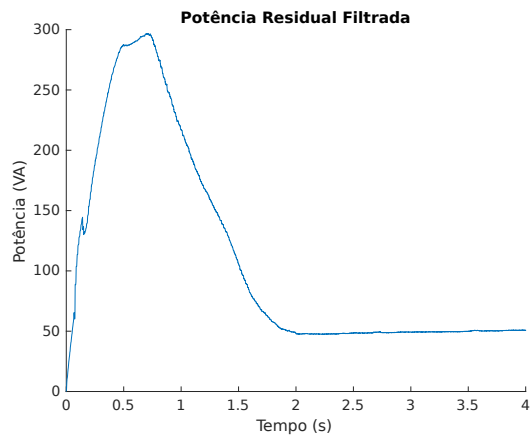
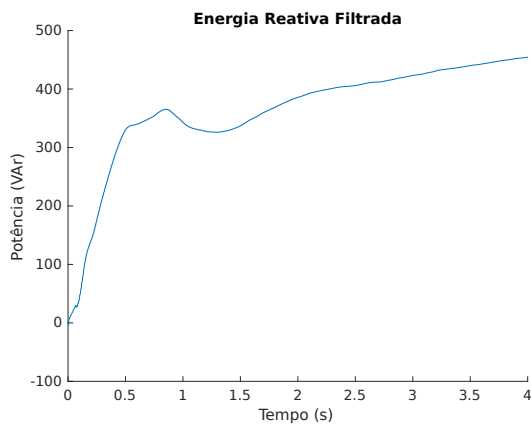
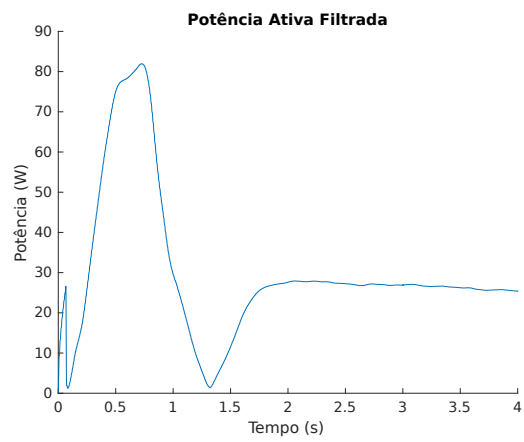
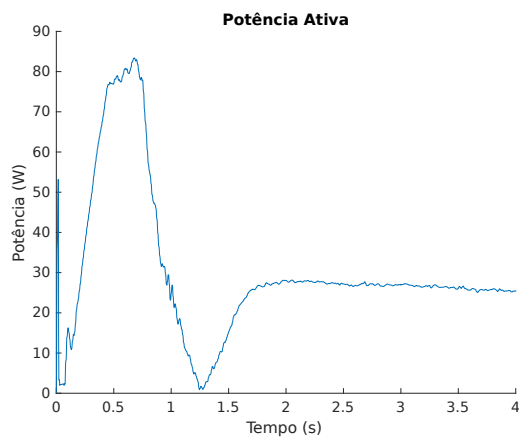
Fonte: Autoria própria.

- Micro-ondas:
 - Marca: Electrolux Continental MC34B/127V
 - Período da captura: 14-12-2023, das 04:00 até 16:00
 - Período da captura: 18-12-2023, das 16:00 até 04:00
 - Diretório: 'wave/Microwave-day/repack'
 - Diretório: 'wave/Microwave-night/repack'
 - Gráficos de potência:



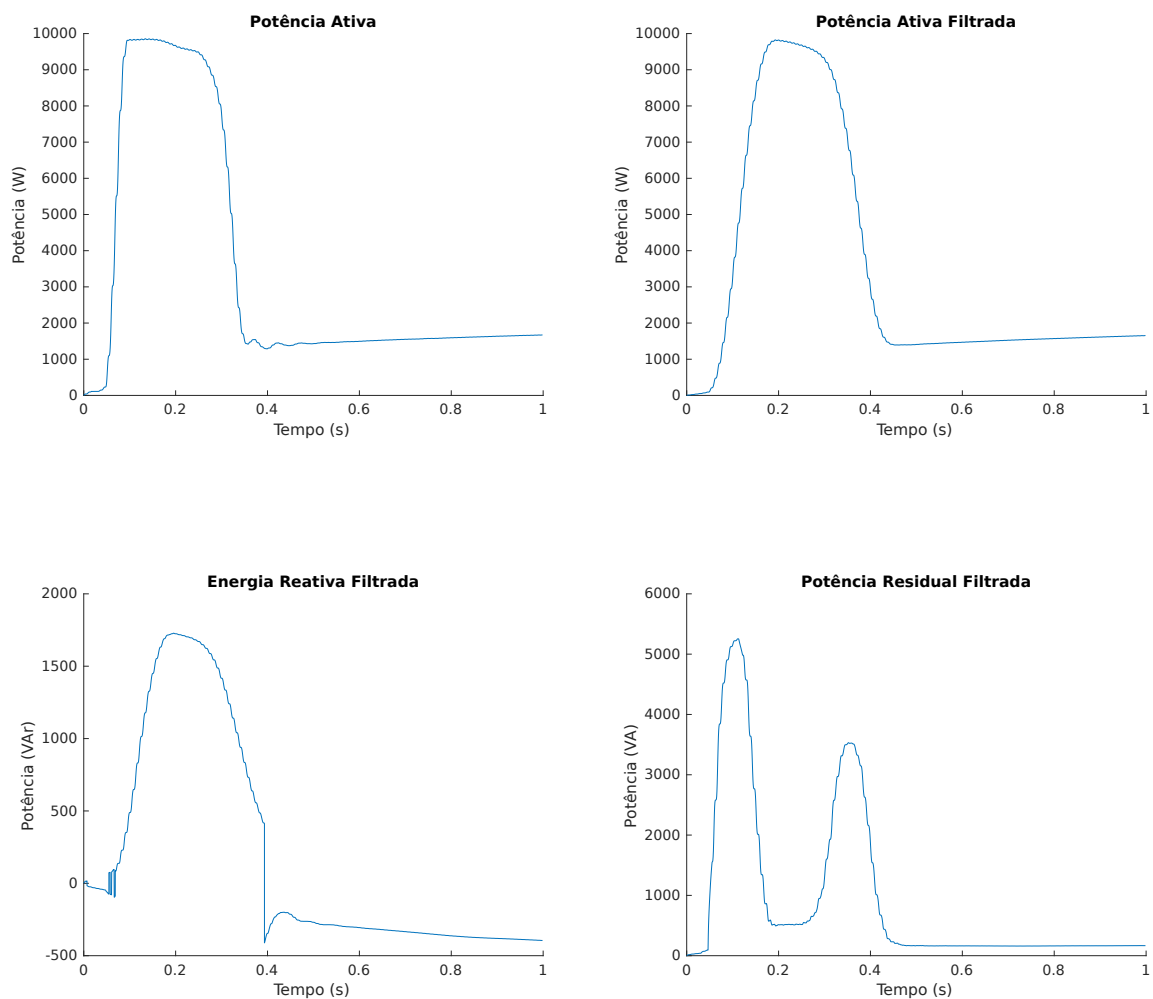
Fonte: Autoria própria.

- Ar-condicionado:
 - Marca: Elgin split teto inverter modelo PVFI60B2DB 60000 Btu/h
 - Período da captura: 20-03-2024, das 09:33 até 17:42
 - Diretório: 'wave/Air Conditioning 20_03_2024/repack'
 - Gráficos de potência:



Fonte: Autoria própria.

- Ar-condicionado:
 - Marca: Springer Carrier Midea 38CCF60M5 42MPCA60M5 220 58000 Btu/h
 - Período da captura: 15-03-2024, 10:04 até 16-03-2024, 02:27
 - Diretório: 'wave/Air Conditioning 15_03_2024/repack'
 - Gráficos de potência:



Fonte: Autoria própria.

3.2 Estratégia de Extração de características

Para este trabalho, será adotada a estratégia de extração de características baseada na análise da assinatura de potência, utilizando a Potência Ativa da Teoria da Potência Conservativa (CPT), para a detecção de eventos ON-OFF.

Durante a coleta de dados, foram encontradas unidades de ar-condicionado com inversor que apresentaram padrões de consumo, como na Figura 24a, já mostrado na seção anterior.

A dificuldade com esse padrão de consumo é a criação excessiva de eventos que podem sobrecarregar o sistema NILM, identificando o mesmo dispositivo sucessivas vezes. O método proposto será de utilizar um cálculo da CPT, com o período T dos produtos internos e valores médios como sendo quatro vezes o período da frequência fundamental da rede elétrica, cujo resultado é evidenciado na Figura 24b. A CPT nesse caso funciona como se tivesse um filtro MAF embutido, eliminando a utilização de filtros dedicados.

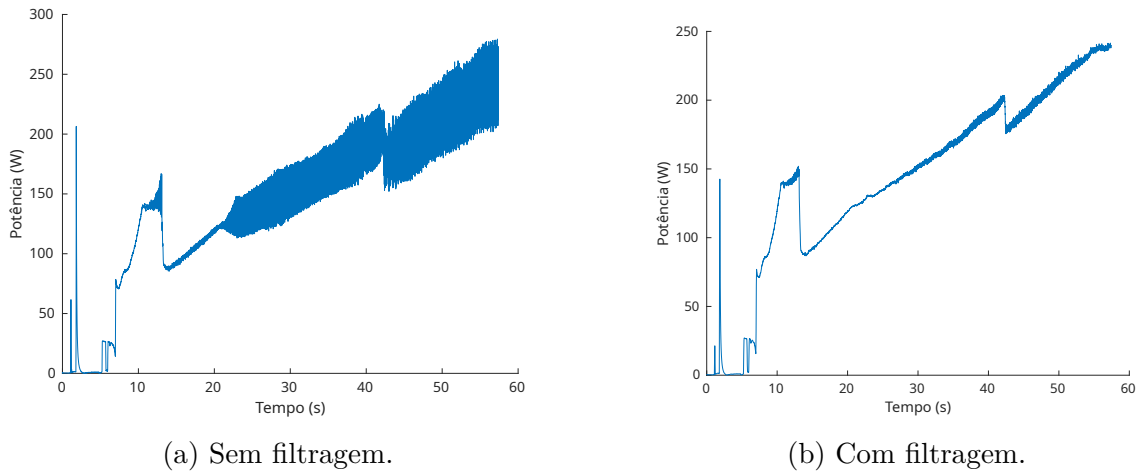


Figura 24 – Potência ativa (CPT) de um ar-condicionado.

Fonte: Autoria própria.

Os valores de potência ativa P e W_{sgn} , baseado na energia reativa W da CPT, são calculados, respectivamente, a partir do valor eficaz da corrente ativa (I_a) e corrente reativa (I_r), expressas como:

$$P = UI_a. \quad (3.1)$$

O valor U é a tensão eficaz, dado a seguir:

$$U = \sqrt{\frac{1}{T} \int_0^T [v(t)]^2 dt}. \quad (3.2)$$

Sendo que T é o período no tempo ciclo da frequência fundamental multiplicado por quatro, e $v(t)$ é a tensão medida em função do tempo. O valor da energia reativa W da CPT é calculado por:

$$W = \langle \hat{v}(t), i(t) \rangle. \quad (3.3)$$

O valor $i(t)$ é a corrente medida em função do tempo. A operação $\langle x, y \rangle$, é chamada de produto interno de duas grandezas periódicas, $x(t)$ e $y(t)$, sendo definida como:

$$\langle x, y \rangle = \frac{1}{T} \int_0^T x(t)y(t)dt. \quad (3.4)$$

Caso o produto interno de $x(t)$ e $y(t)$ resulte em zero, tais grandezas serão ditas ortogonais, isto é:

$$\langle x, y \rangle = 0. \quad (3.5)$$

A operação $\hat{x}$ é definida como integral imparcial, tal como a seguir:

$$\hat{x} = x_f - \bar{x}_f. \quad (3.6)$$

Por sua vez, x_f é definida como a integral no tempo da variável $x(t)$:

$$x_f = \int_0^t x(\tau)d\tau. \quad (3.7)$$

E o valor médio $\bar{x}_f$ é definido como:

$$\bar{x}_f = \langle x \rangle = \frac{1}{T} \cdot \int_0^T x(t)dt \quad (3.8)$$

O valor da energia reativa W_{sgn} é dado por:

$$W_{sgn} = UI_r \cdot sgn(W). \quad (3.9)$$

A função $sgn(x(t))$ é o sinal da função no tempo, dado por:

$$sgn(x(t)) = \begin{cases} +1, x(t) \geq 0. \\ -1, x(t) < 0. \end{cases} \quad (3.10)$$

Os valores eficazes das correntes ativas e reativas são, respectivamente, dadas por:

$$I_a = \sqrt{\frac{1}{T} \int_0^T [i_a]^2 dt}. \quad (3.11)$$

$$I_r = \sqrt{\frac{1}{T} \int_0^T [i_r]^2 dt}. \quad (3.12)$$

As variáveis no tempo da corrente ativa e reativa, i_a e i_r são, respectivamente, definidas conforme a seguir:

$$i_a = \frac{\langle v, i \rangle}{\|v\|^2} = \frac{P}{V^2} = G_e v. \quad (3.13)$$

$$i_r = \frac{\langle \hat{v}, i \rangle}{\|\hat{v}\|} \hat{v} = \frac{W_r}{\hat{V}^2} \hat{v}. \quad (3.14)$$

Sendo que a norma Euclidiana é definida como:

$$\|x\| = \sqrt{\langle x, x \rangle} = \sqrt{\frac{1}{T} \int_0^T x^2(t) dt} = X. \quad (3.15)$$

Sendo X o valor eficaz da variável x . O parâmetro G_e é referido como uma condutância equivalente da carga. A corrente ativa corresponde à parcela que, efetivamente, transfere potência para a carga e possui a mesma forma de onda da tensão.

O cálculo das potências ativa P e energia reativa W_{sgn} foram pensados para auxiliar a filtragem do excesso de variações da carga.

3.3 Detector de Eventos

Esta etapa do algoritmo encontra os tempos de início e fim de uma variação no padrão de consumo, que pode significar uma carga não identificada sendo acionada ou desligada, ou dispositivo previamente identificado que, por qualquer razão, alterou significativamente o seu padrão de consumo.

Sistemas NILM tipicamente definem montantes mínimos de variação de consumo para minimizar chamadas às etapas de desagregação e identificação de cargas. Se o consumo se altera além de um certo limiar, pode ser a indicação de algum acionamento ou desligamento que deverá ser analisado posteriormente. O algoritmo de detecção de eventos proposto neste trabalho define um montante mínimo de variação de consumo a partir do uso de dados de aprendizado das cargas que serão desagregadas. Estes dados de aprendizado podem ser usados posteriormente na etapa de identificação para treinar o algoritmo de aprendizado de máquina que é usado nessa etapa, se for necessário. Oscilações de consumo que não atingem este montante mínimo não são consideradas. O algoritmo para encontrar este montante mínimo é descrito na seção 3.3.1.

O algoritmo de detecção de eventos se divide em duas partes que não são usadas simultaneamente, dependendo de quão rápidas são as mudanças na potência. A primeira irá procurar por eventos de curta duração, proposto na seção 3.3.2, e a segunda parte irá procurar por eventos de longa duração, proposto na seção 3.3.4. Tornou-se necessária esta divisão devido ao montante de dados necessário para a correta detecção de cargas que variam seu padrão de consumo muito lentamente, mas há a desvantagem da baixa precisão da duração do evento para eventos de longa duração.

3.3.1 Limiar mínimo

Um valor limiar mínimo de potência ΔP é proposto para que, quando uma determinada carga seja acionada ou desligada, o seu momento de início e fim de consumo seja determinado e também o tempo que essa carga tenha levado a chegar ao seu consumo ou gerações nominais.

Primeiramente, serão capturados individualmente o consumo no tempo das cargas que serão desagregadas para serem utilizadas como dado de aprendizado e calculadas as suas respectivas potências ativas instantâneas, baseando-se na CPT. Deverá ser aplicada uma função janela $W(t)$ de comprimento N , correspondente ao comprimento dos dados capturados da carga para cada uma delas, conforme equação 3.16.

$$W(t) = \begin{cases} 0, t < 0. \\ \frac{t \times f}{n}, t < (n \times f). \\ 1, (n \times f) \leq t < (N - (n \times f)). \\ \frac{(N-t) \times f}{n}, t \geq N. \\ 0, t > N. \end{cases} \quad (3.16)$$

A Figura 25 contém exemplo de uma função janela e do resultado da sua aplicação em uma carga sintética.

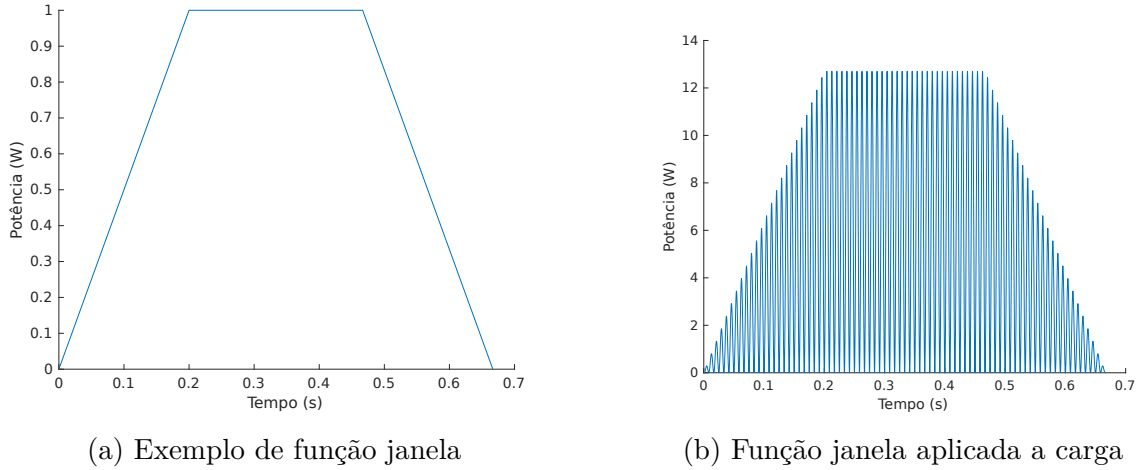


Figura 25 – Resultados a aplicação da função janela $W(t)$

Fonte: Autoria própria.

Esse janelamento remove falsos positivos caso a potência dos dados de aprendizado não inicie com zero. A constante n é a quantidade de ciclos da frequência fundamental da rede que esse algoritmo irá ignorar no início e fim das amostras de aprendizado. Depois da aplicação da função janela, será aplicada uma operação de diferenças finitas para encontrar a derivada aproximada dos dados, e um fator de corte δ , para redefinir para

zero a potência no tempo $p(t)$, se $p'(t) > \delta$ ou se $p'(t) < -\delta$, sendo que $p'(t)$ é a derivada aproximada de $p(t)$. Um segundo fator de corte determinado pela porcentagem F_1 em relação à potência máxima $p(t)$ é aplicado. Esse fator de corte é usado para remover certas variações de potência, de modo que, quando a diferença finita for aplicada a $p(t)$, variações insignificantes não sejam indevidamente amplificadas. O filtro MAF embutido no cálculo da CPT ajuda a remover estas variações.

Foram testados valores de F_1 começando em 0% e aumentando até o momento em que houve uma queda acentuada de precisão. Escolhendo valores iguais ou maiores que 10%, causaram uma perda de precisão conforme a Tabela 5, que mostra o F -score ($\beta = 1$) médio para vários cenários da desagregação e utilizando o classificador KNN:

F_1	0%	1%	5%	10%	20%	30%	40%	50%
F -score médio	91,00%	91,00%	91,00%	69,34%	69,34%	69,34%	34,96%	34,96%

Tabela 5 – F -scores médios e sua relação à porcentagem F_1 .

Fonte: Autoria própria.

Não é recomendado manter o valor F_1 em zero apesar de não haver alteração na precisão devido a não ser possível determinar se $p'(t)$ calculado nesse passo não venha a amplificar pequenas variações em cargas que não foram testadas nesse trabalho. O resultado desse processamento é mostrado na Figura 26, no qual se pode notar os vários picos gerados pelo consumo variável da carga.

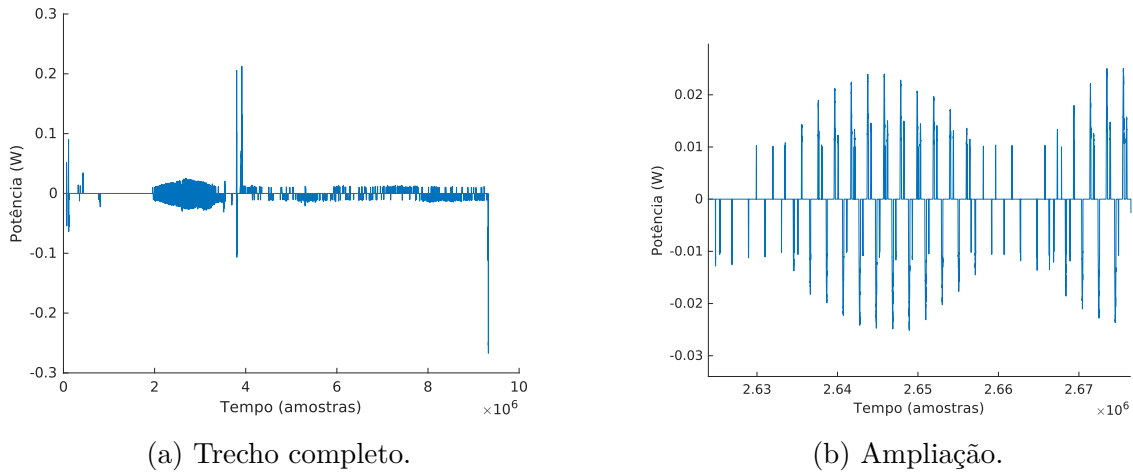


Figura 26 – Potência da carga processada para encontrar os picos de variação

Fonte: Autoria própria.

O próximo passo é encontrar os pontos no tempo em que os dados processados diferem de zero, ou seja, a posição em que as variações de potência são mais significativas. Conforme mostrado na Figura 26, o algoritmo produz vários picos, positivos e negativos, que representariam as variações de consumo. O momento de início (t_s) é determinado como o primeiro momento no tempo em que $p'(t)$ não é zero. O tempo final (t_e) é determinado

como o primeiro momento em que houve a passagem de n ciclos da frequência fundamental, depois de t_s , em que a $p'(t)$ se mantém em zero. A Figura 27 representa um exemplo do início processado de uma carga.

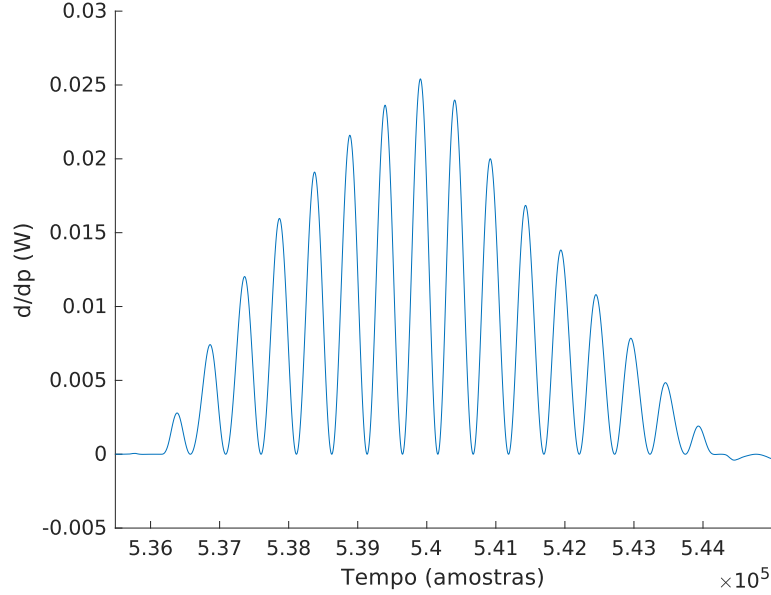


Figura 27 – Exemplo de primeiro início processado de uma carga.

Fonte: Autoria própria.

Com os tempos t_s e t_e , é extraída uma fatia de dados de potência da carga (Figura 28) em que é calculado o módulo, é encontrado o valor máximo de potência real P_{max} da fatia e o momento no tempo t_{max} em que o valor máximo ocorreu. A potência máxima P_{max} é gravada na matriz M. O processo de detecção de picos é então repetido até que sejam encontrados todos os valores máximos da carga e armazenados na matriz M. A partir da matriz M, é obtido o valor mínimo m. O fluxograma está na Figura 29.

O processo é repetido para todas as cargas e o valor mínimo dos resultados obtidos é utilizado como limiar de início de potência ΔP . Caso alguma outra carga precise ser adicionada depois que ΔP tenha sido determinado, o processo para encontrar o limiar mínimo deve ser executado somente para a nova carga e o novo ΔP será o valor mínimo entre o ΔP anterior e o resultado do algoritmo aplicado para a nova carga.

Devido às diferenças de magnitude de potência entre as diferentes cargas que serão desagregadas, um segundo ajuste de ΔP pode ser necessário. Este ajuste depende da diferença de consumo entre as cargas que serão desagregadas, sendo que certas combinações de cargas não serão desagregadas corretamente usando os algoritmos de inteligência artificial para classificação propostos. Esse ajuste será explicado na seção 3.9.

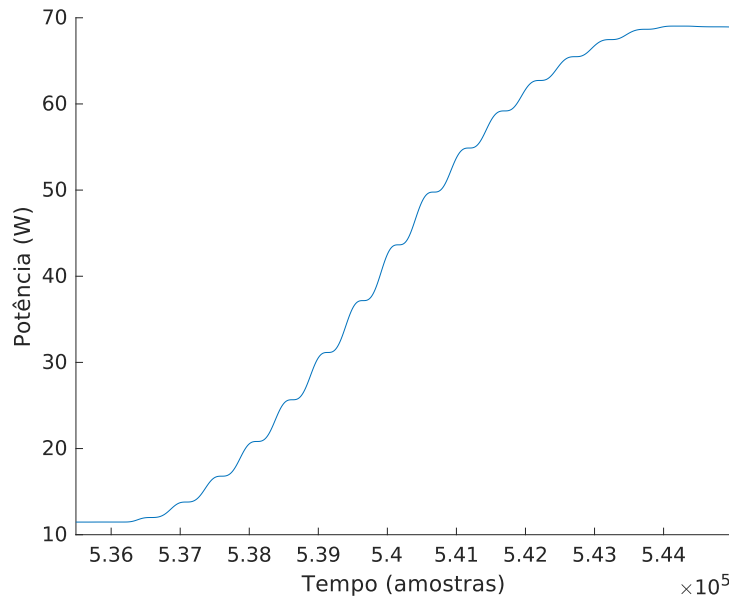


Figura 28 – Corte de potência da carga.

Fonte: Autoria própria.

3.3.2 Detector de Eventos Curta Duração

Esta etapa do algoritmo de detecção de eventos compara a potência ativa da CPT do presente com dados sucessivamente mais antigos para poder ser encontrada uma rampa de aumento ou diminuição de consumo. Os decrementos acontecem em ciclos da frequência fundamental da rede. O algoritmo varre os dados no passado na procura desta rampa até que sejam atingidos um dos seguintes momentos:

1. S segundos no passado,
2. Momento imediatamente após a ultima detecção de um evento ou,
3. Início da captura de dados.

A constante S é usada para calcular o comprimento de um *buffer* circular interno que irá guardar dados para determinar o tempo de ligamento ou desligamento de uma carga. Esse comprimento depende do equilíbrio entre a precisão dos tempos de evento adquiridos e os limites de *hardware* da implementação. Quando no *buffer* interno for encontrada uma rampa ascendente ou descendente de potência cujo módulo da variação for maior que ΔP , o algoritmo vai iniciar um *buffer* secundário. Este segundo *buffer* é preenchido com as variações de potência de meio ciclo entre o início e o fim do trecho até agora adquirido, onde houve a variação de ΔP . É encontrado o maior valor (ou menor valor se a rampa for descendente) dessa variação.

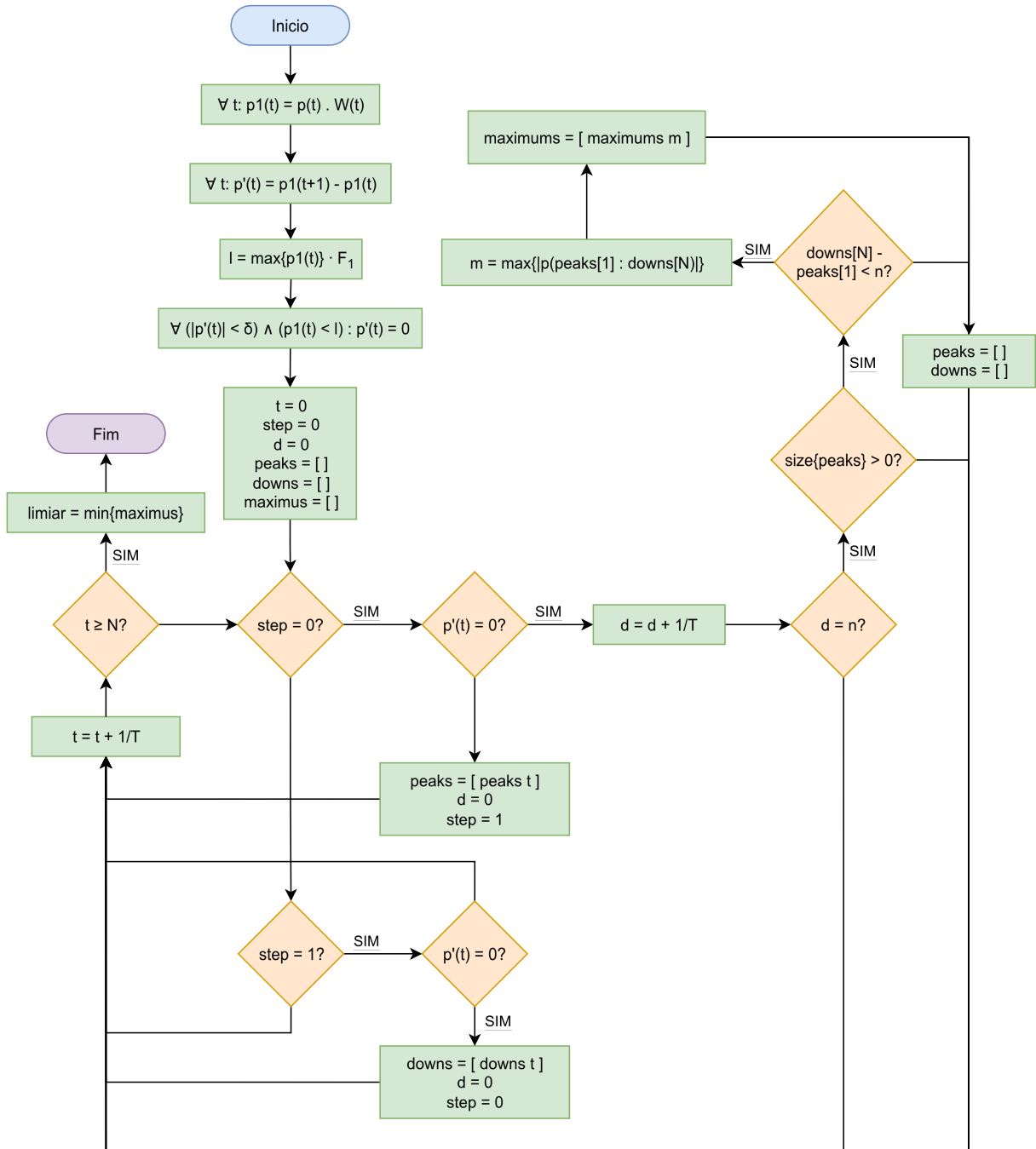


Figura 29 – Fluxograma simplificado para encontrar o limiar mínimo de uma carga.

Fonte: Autoria própria.

Nota: N : Comprimento da amostra; $p(t)$: Informações de potência para aprendizado; $W(t)$: Função janela; $p1(t)$: Potência após janelamento; δ : Fator de corte da potência; T : Taxa de amostragem; F_1 : Fator de corte em relação à potência máxima.

O algoritmo vai continuar a adquirir dados e preencher a tabela secundária com novas variações a cada meio ciclo até que a variação de meio ciclo de potência reduza a F_2 por cento da variação máxima encontrada, ou que esta variação caia para zero ou inverta seu sinal. Essa comparação entre variações é necessária porque, quando existem cargas variando o seu consumo lentamente juntamente com cargas que variam o consumo

rapidamente, a classificação de cargas de variação mais rápida se torna uma tarefa mais complexa, já que não há um início ou fim bem definido do evento de curta duração, conforme mostrado na Figura 30:

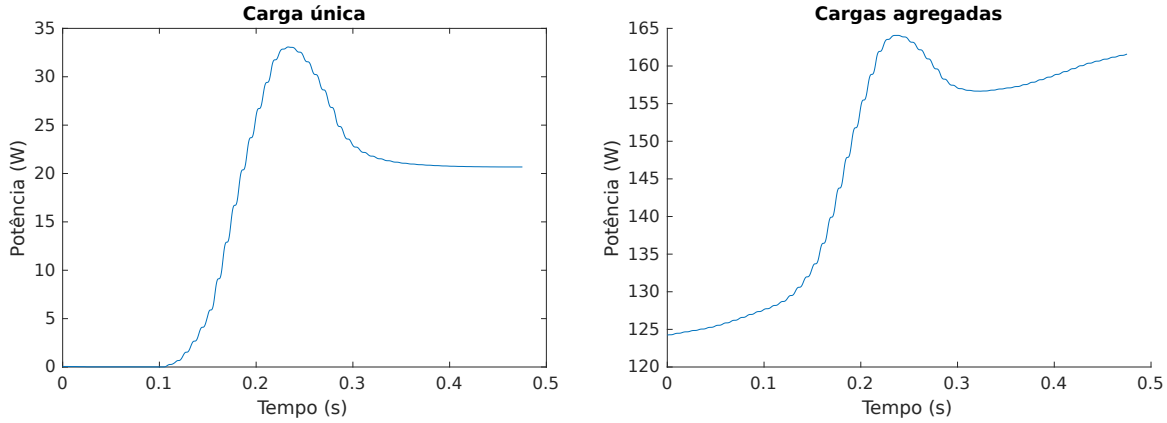


Figura 30 – Comparativo entre uma carga única e cargas agregadas.

Fonte: Autoria própria.

Para a escolha do valor de F_2 , foi usado o F -score ($\beta = 1$) médio de vários cenários desagregação usando o classificador KNN, em que um valor entre 1% e 5% maximizou a precisão, conforme a Tabela 6.

F_2	0%	1%	5%	10%	20%	30%	40%	50%
F -score médio	67,98%	91,00%	91,00%	69,01%	80,81%	66,15%	78,24%	47,78%

Tabela 6 – F -scores médios e sua relação à porcentagem F_2 .

Fonte: Autoria própria.

A partir da verificação da variação, o algoritmo irá procurar pelo valor máximo ou mínimo da potência, dependendo, respectivamente, se a rampa é ascendente ou descendente, utilizando os dados já capturados. O valor máximo ou mínimo será marcado como o fim do evento. A interrupção da varredura que acontece após a detecção do último evento é necessária para o algoritmo não confundir eventos. O fluxograma do algoritmo proposto é mostrado na Figura 31.

A escolha dos saltos em passos de meio ciclo é devido a um padrão encontrado pela observação do aumento ou diminuição da variação de potência. Conforme observado na Figura 32a, a variação de potência muda rapidamente a cada amostra de potência ativa calculada, mas mostra uma variação mais lenta quando o passo de observação é de meio ciclo da rede ou quando o passo de observação é múltiplo de meio ciclo. Os pontos marcados pelos pontos vermelhos na Figura mostram as posições de um passo de observação de meio ciclo.

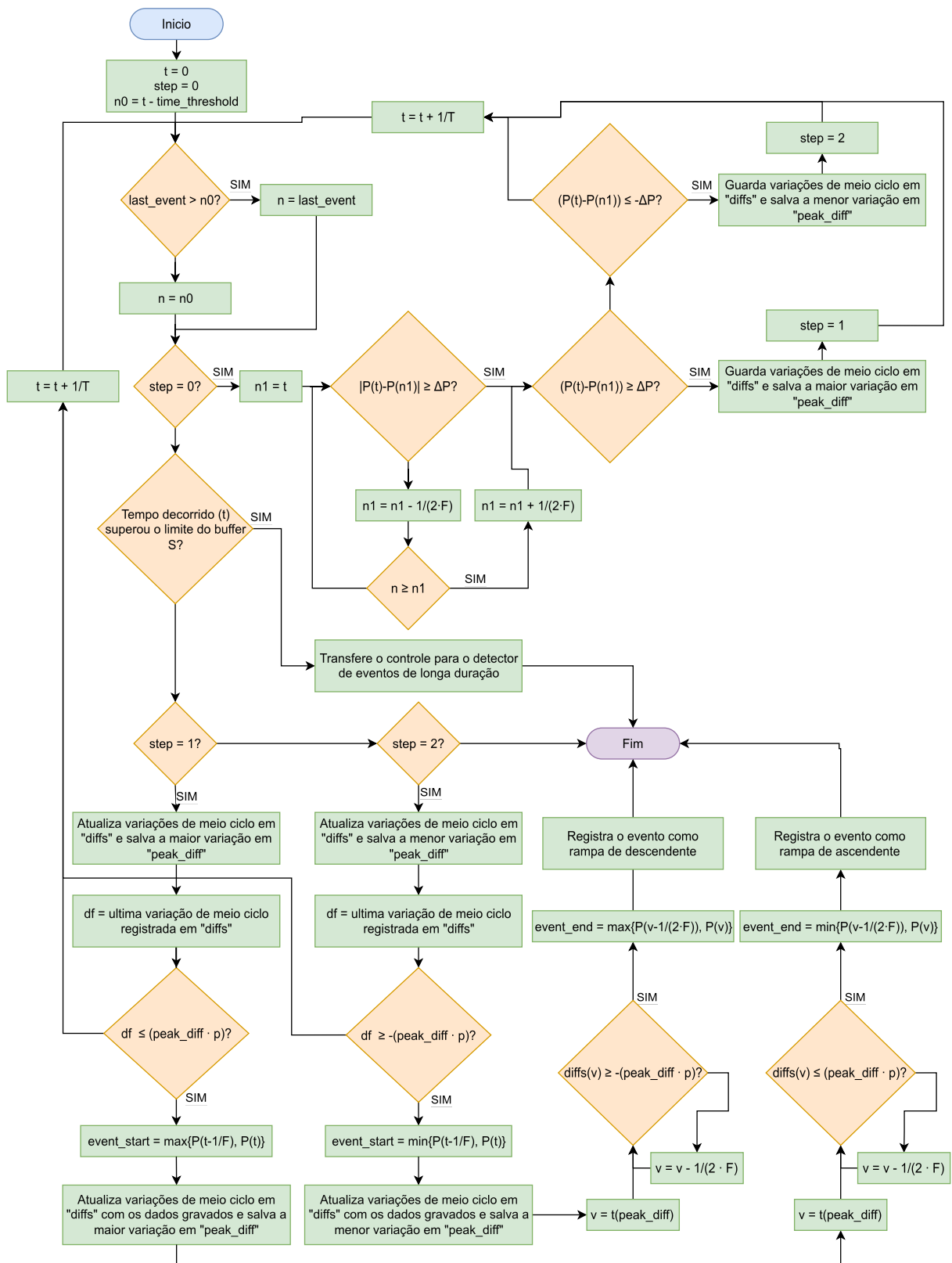


Figura 31 – Fluxograma simplificado do detector de eventos de curta duração.

Fonte: Autoria própria.

Nota: S: Comprimento do *buffer* interno; t: momento presente; last_event: Momento do ultimo evento; p: Declínio percentual na variação percentual para considerar o fim do evento (F_2); $P(t)$: Potência ativa em função do tempo; ΔP : Limiar Mínimo; F : Frequência da rede; T: Taxa de amostragem.

A Figura 32b mostra a mesma carga, no mesmo momento, mostrando a potência instantânea em que os pontos vermelhos correspondem à Figura 32a. É possível notar uma leve variação de potência em cada ponto que indica um aumento também gradual da potência média consumida pela carga, e uma variação abrupta a cada meio ciclo. Nesta segunda figura, não há nenhum tipo de filtragem.

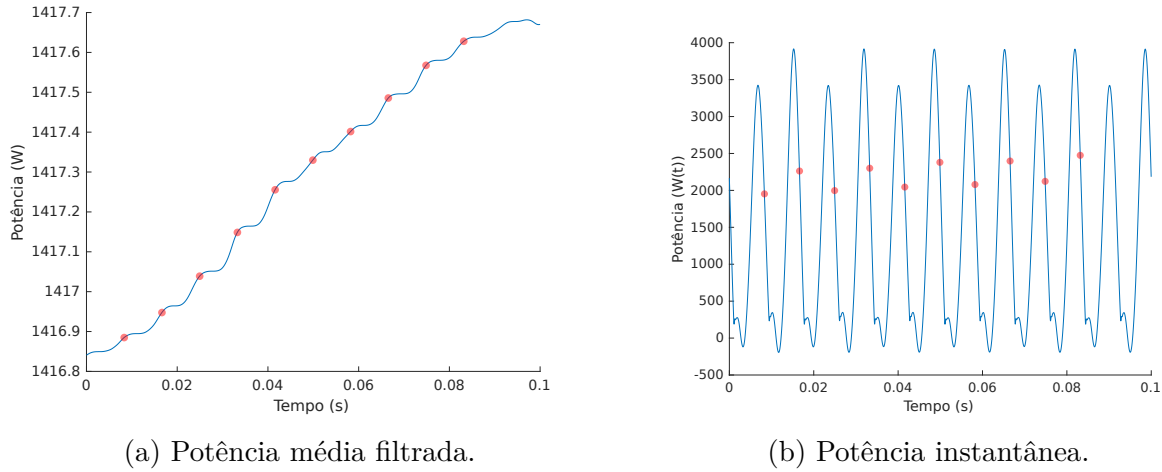


Figura 32 – Padrão de consumo de uma carga.

Fonte: Autoria própria.

O algoritmo passará então a procurar o início do evento para que o seu tempo de transição possa ser determinado. Isso é feito fazendo o caminho inverso à detecção de fim de evento: procurar para trás o momento em que há o ΔP , preencher o *buffer* secundário no sentido contrário ao mesmo tempo em que o máximo ou mínimo é encontrado, procurar a redução de variação de F_2 por cento, queda para zero ou inversão de sinal e procurar o máximo ou mínimo nos dados já adquiridos.

A Figura 33 mostra o caminho que o algoritmo percorre ao analisar os dados de potência. O detector de eventos de curta duração irá considerar os pontos 3 e 4 como uma rampa ascendente. O ponto 4 é interpretado como um ponto de máximo e os pontos 3 e 2 são interpretados como uma rampa inversa. As setas negras indicam o sentido da varredura do algoritmo de detecção de eventos de curta duração. Como há um incremento de potência entre os pontos 3 e 4, a região entre esses pontos será considerada um evento de aumento de consumo.

As Figuras 34, 35 e 36 mostram os potenciais resultados da detecção de eventos de curta duração. Os pontos P_2 e P_3 determinam a região de variação máxima, os pontos P_4 e P_5 determinam o fim do evento, onde as variações diminuem ou invertem, e os pontos P_0 e P_1 determinam o início do evento. É importante salientar que os inícios e términos dos eventos não serão necessariamente iguais e simétricos como representados nesses gráficos; o início pode acontecer como o evento de um dos gráficos mostrados e o final como outro.

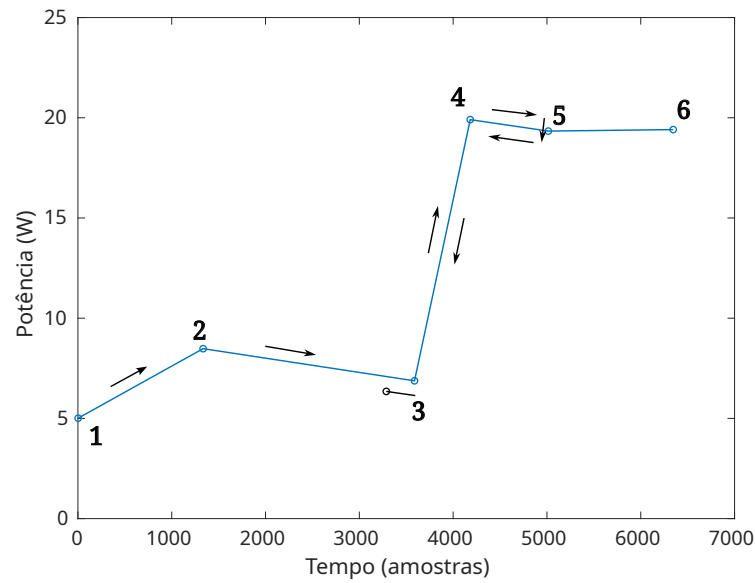


Figura 33 – Diagrama simplificado de evento.

Fonte: Autoria própria.

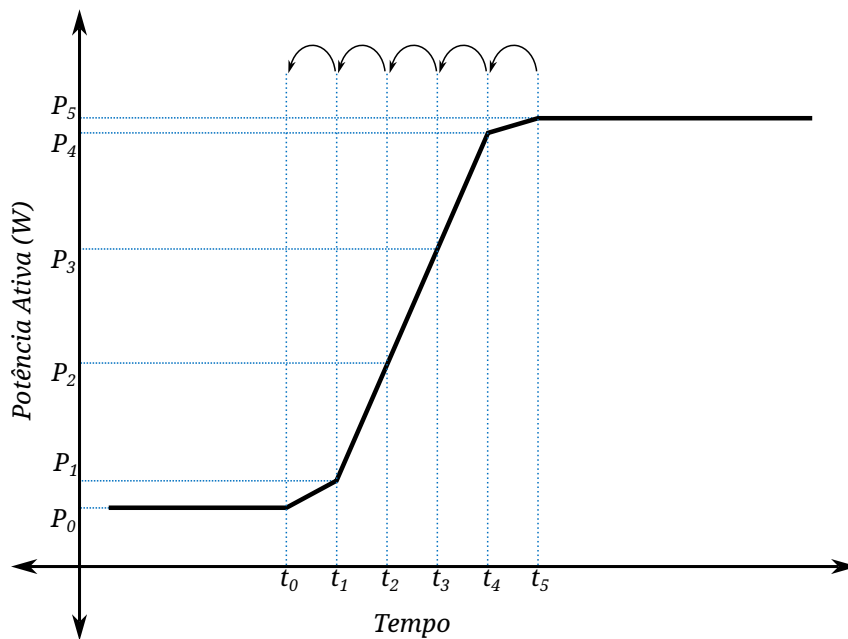


Figura 34 – Diagrama simplificado de evento com parada lenta.

Fonte: Autoria própria.

O evento da Figura 34 representa casos em que a potência agregada não termina rapidamente, o que pode acontecer devido a diversos fatores, mas uma possibilidade importante seria a presença de uma carga variando da potência lentamente enquanto outra apresenta uma variação repentina.

A Figura 35 representa casos em que a potência agregada chega a um valor máximo e então diminui. Uma possibilidade seria o acionamento de uma carga capacitiva, em que

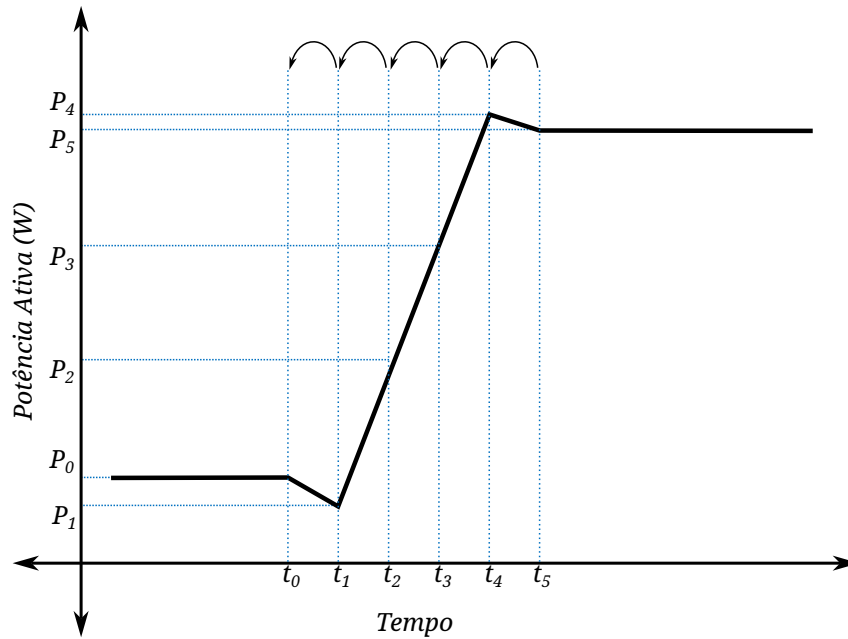


Figura 35 – Diagrama simplificado de evento com reversão na parada.

Fonte: Autoria própria.

há um aumento repentino de consumo enquanto o capacitor é carregado, ou uma carga indutiva, como um motor elétrico.

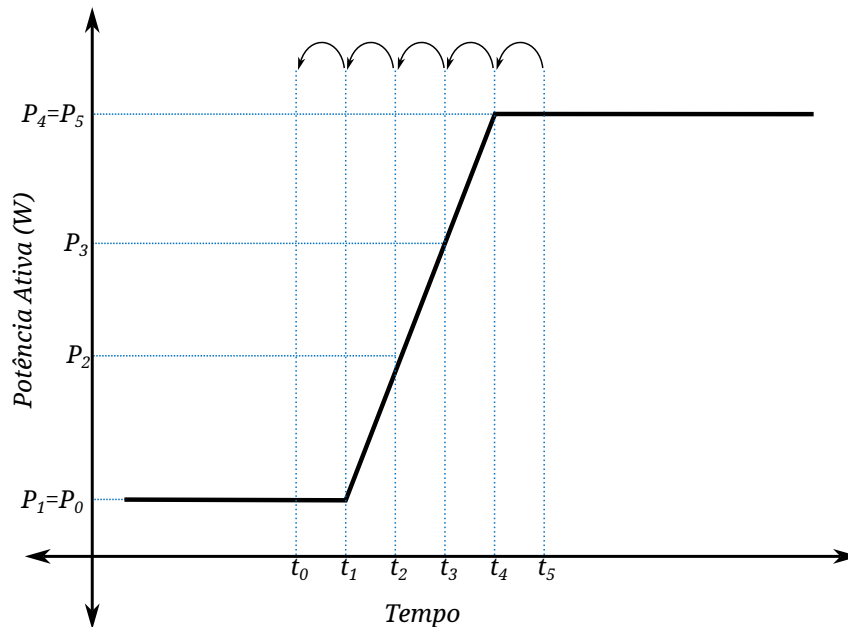


Figura 36 – Diagrama simplificado de evento com parada rápida.

Fonte: Autoria própria.

A Figura 36 representa casos em que a potência agregada chega a um valor máximo e então para de aumentar. Uma possibilidade seria o acionamento de uma carga resistiva em que existe o degrau de alteração de consumo quando esta é acionada ou desligada.

As variações são determinadas através da inclinação, considerando a variação da

potência aproximadamente linear, como visto em (3.17) e (3.18).

$$\frac{(P_1 - P_0) \div (t_1 - t_0)}{(P_3 - P_2) \div (t_3 - t_2)} \leq F_1. \quad (3.17)$$

$$\frac{(P_5 - P_4) \div (t_5 - t_4)}{(P_3 - P_2) \div (t_3 - t_2)} \leq F_1. \quad (3.18)$$

3.3.3 Deficiências no detector de eventos de curta duração

O detector de eventos de curta duração apresenta algumas deficiências ao procurar pelo início ou fim do evento em alguns casos específicos, principalmente quando eventos de longa duração e curta duração acontecem ao mesmo tempo. Na Figura 37, são mostrados esses casos.

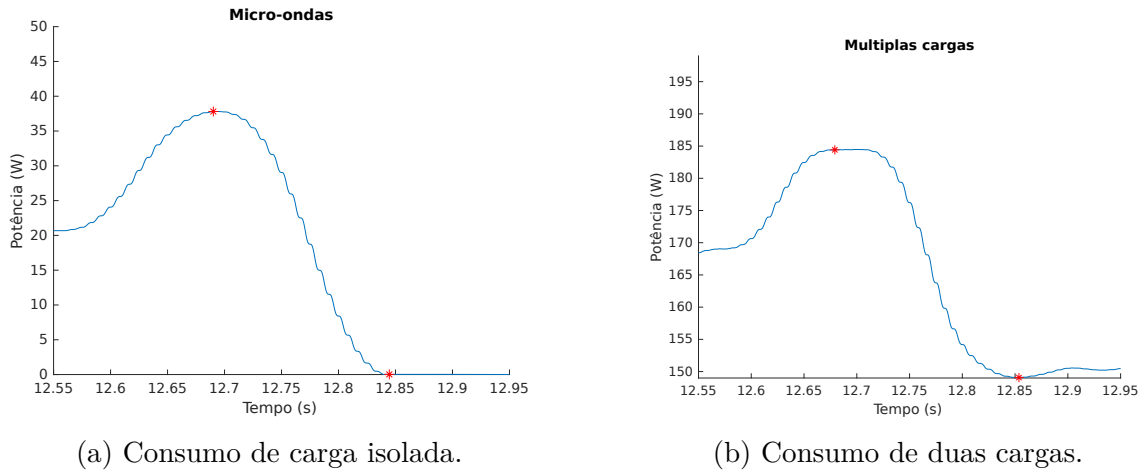


Figura 37 – Comportamento do Detector de Eventos em cenários diferentes.

Fonte: Autoria própria.

Nesta figura, é mostrado um evento em uma carga única à esquerda e o mesmo evento com outra carga acontecendo ao mesmo tempo, à direita. Os asteriscos são os lugares em que o detector de evento marcou o início e o fim do evento. O evento é marcado em lugares diferentes devido à influência da segunda carga. O comprimento do evento é alterado significativamente, o que causa um impacto negativo no momento da desagregação. Esse tipo de alteração é mais significativa em variações pequenas relativas a outras cargas presentes no sistema de desagregação. Para coletar os dados para a criação da Figura 37, foi necessária a desativação da seleção de dados explicada na seção 3.9. Esse problema sempre acontece quando os limites de um evento são encontrados, mas a sua influência é significativamente menor quando a carga de maior consumo é a geradora de evento, conforme a Figura 38.

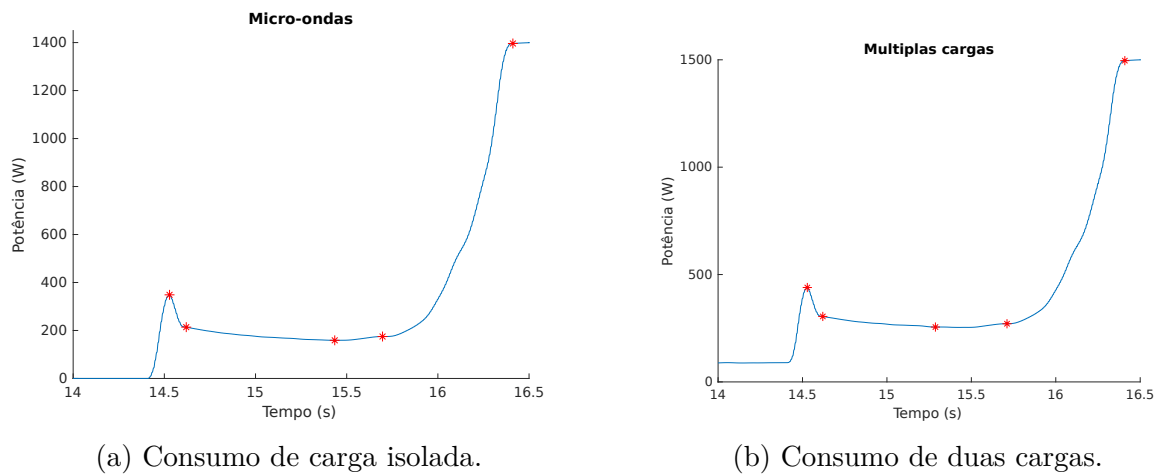


Figura 38 – Comportamento do Detector de Eventos em cenários diferentes.

Fonte: Autoria própria.

Uma solução para isto seria modificar o detector de eventos de curta duração para que ele procure por um ponto mais próximo de onde a variação comece, conforme a Figura 39.

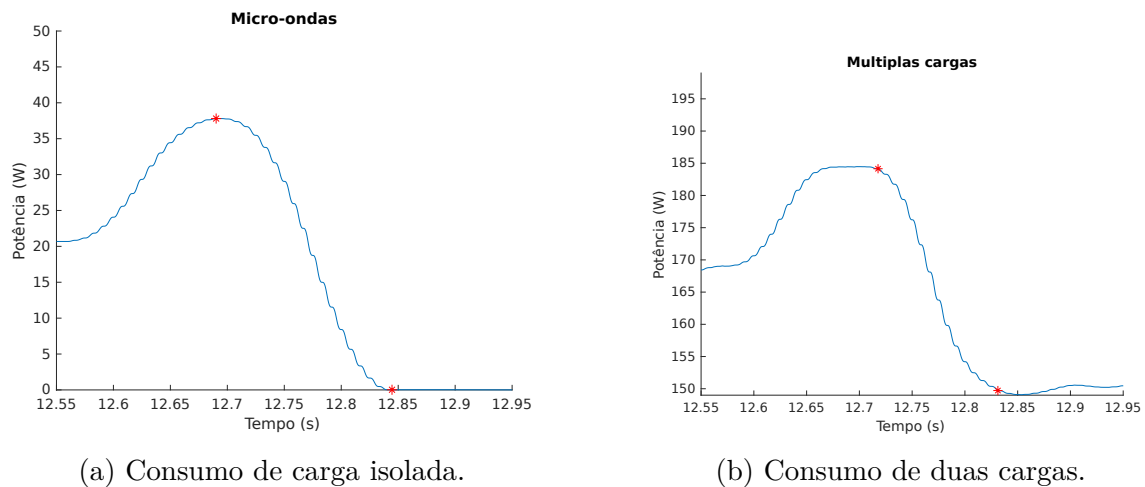


Figura 39 – Comportamento ideal do Detector de Eventos.

Fonte: Autoria própria.

Uma segunda questão é a execução de múltiplos eventos de curta duração simultâneos. O algoritmo não prevê possibilidade do acontecimento de múltiplos eventos desse tipo simultâneos, e por conseguinte, essa combinação de eventos não serão reconhecidos como múltiplos eventos, mas somente como apenas um, e devido à característica dos algoritmos de inteligência artificial de classificação utilizados, um evento incorreto será reportado.

3.3.4 Detector de Eventos de Longa Duração

Esta etapa procura mudanças lentas no consumo que podem levar de vários segundos a minutos. Alterações lentas de potência são ignoradas pelo detector de eventos de curta

duração porque não acontecem na janela de dados do seu *buffer* circular. Um evento de longa duração é encontrado comparando a diferença entre a potência ativa da CPT do último evento e a potência ativa do presente. Se o módulo da potência for maior que ΔP , um evento de longa duração pode estar presente. Para confirmar a ocorrência do evento, uma varredura por eventos de curta duração é então realizada para o tamanho máximo S do *buffer* circular. É feita uma verificação da velocidade da variação de potência na varredura de evento de curta duração. Somente a parte da detecção do início do evento de curta duração é efetuada porque o fim do evento já é conhecido. Se a velocidade da alteração de potência encontrada ao calcular o evento de curta duração sugerir que o início do evento aconteceu próximo ao seu final, o detector de eventos de longa duração irá considerar o evento atual como sendo de curta duração e transferir o controle para o detector de eventos de curta duração para que ele seja corretamente tratado.

O algoritmo considera o evento como sendo de longa duração se a velocidade de variação de potência ativa pelo comprimento do evento de curta duração calculado for menor que um quinto da velocidade de variação de ΔP pelo comprimento do *buffer* de tamanho S . A Figura 40 mostra um caso quando isso acontece. O asterisco vermelho mostra a posição da detecção falsa de evento de curta duração e o asterisco verde mostra o evento de curta duração real.

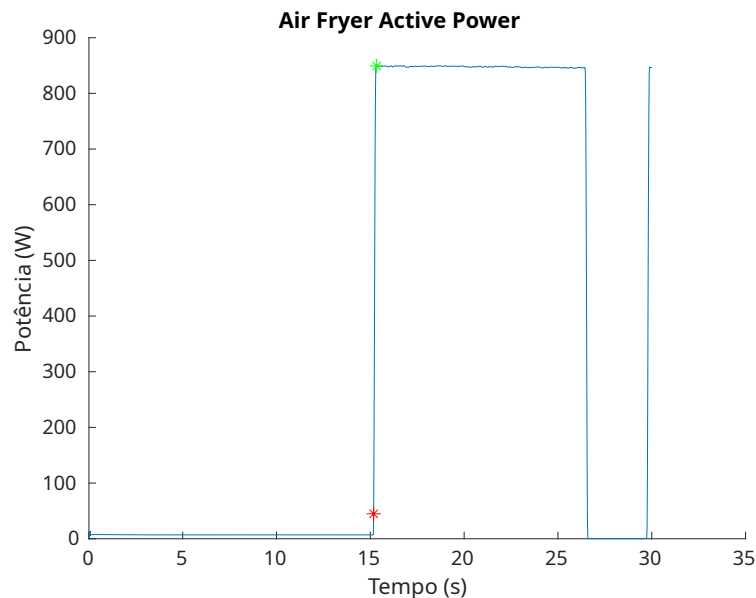


Figura 40 – Falso evento de curta duração.

Fonte: Autoria própria.

Isso acontece quando o conjunto de cargas agregadas não varia de potência pelo tamanho máximo do *buffer* de tamanho S e logo a seguir, um evento de curta duração acontece. Neste caso, um evento de curta duração deveria ser registrado ao invés de um evento de longa duração. Caso a varredura indique que realmente aconteceu um evento de longa duração, ele será registrado como tal.

O comprimento de evento que esta etapa obtém tem uma exatidão inferior se comparado ao detector de eventos de curta duração porque não há dados para encontrar o ponto exato de onde começou o evento. Na Figura 41, temos um exemplo hipotético de um evento lento. Os pontos (a) e (d) são respectivamente os momentos no tempo de início e fim, mas o evento real inicia em (b). Os pontos (c) e (d) representam as grandezas elétricas guardadas no *buffer* circular de comprimento S . O detector de eventos de longa duração é impedido de funcionar até que se passem S segundos desde o último evento registrado. Isso implica que a duração mínima desse tipo de evento será de S segundos.

Para o caso de uma certa carga tiver uma variação de consumo que seja $m * \Delta P$ Watts e uma variação de tempo $n * S$ segundos, vários eventos de longa duração encadeados podem ocorrer.

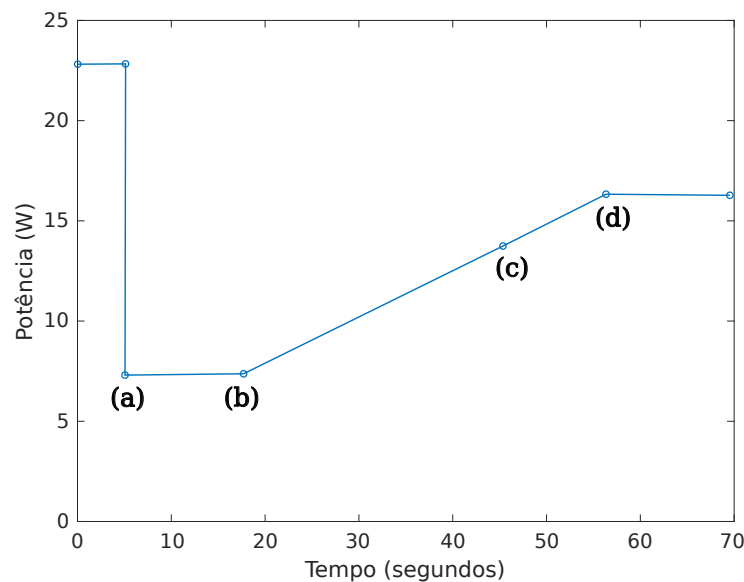


Figura 41 – Diagrama simplificado de evento de longa duração.

Fonte: Autoria própria.

3.3.5 Deficiências do detector de Eventos de Longa Duração

Para um cenário em que duas cargas estejam simultaneamente variando suas potências de modo que a variação total seja menor que o módulo de ΔP , ou seja, quando a variação de uma carga seja de aumento de consumo e que a segunda, seja de diminuição de consumo, o detector de eventos de curta duração não será capaz de detectar os respectivos eventos.

3.3.6 Procedimento de geração de evento para desligamento abrupto

Para o caso de cargas em que não houve a aquisição dos dados de classificação no momento de seu desligamento, ou para cargas que possuem um sistema de espera em

que esta mantém um pequeno consumo, é adicionado ao final da coleta de eventos um evento extra de curta duração para que o algoritmo possa entender que esta carga desligou quando esta estiver sendo desagregada. Para isto, serão utilizados os valores elétricos da última aquisição e estes serão subtraídos pelos seus respectivos valores zerados, simulando, respectivamente, um início e fim de evento. Este procedimento só será executado se, ao final da aquisição de dados de classificação, o módulo da última potência ativa for superior ao ΔP calculado. O comprimento do evento será definido em uma amostra para que o algoritmo entenda que esse é um evento de curta duração.

Para cargas cujo final da aquisição para classificação do módulo da potência ativa seja menor que o ΔP , esse procedimento é desnecessário. Cargas ligadas simultaneamente que individualmente não consomem mais que o ΔP podem causar um evento espúrio quando são desligadas simultaneamente.

3.4 Extração das grandezas elétricas

As grandezas elétricas usadas para desagregar e identificar uma carga serão extraídas após a detecção de um evento. Para o caso de variação de consumo que tenha sido identificada como um evento de curta duração, os momentos no tempo de início e fim deste evento serão usados para extrair grandezas. Usando a Figura 33 como exemplo, no ponto 4 serão extraídas as características de fim de evento e no ponto 3 as de início de evento. No caso do evento ter sido processado pelo detector de eventos de longa duração, as características do início serão extraídas utilizando os dados de consumo do último evento ocorrido e as finais com as características obtidas no fim do evento de curta duração.

Com os momentos de tempo de início e fim de evento obtidos, as grandezas elétricas da variação de potência do evento são determinadas aproveitando as características conservativas das grandezas elétricas obtidas pela CPT.

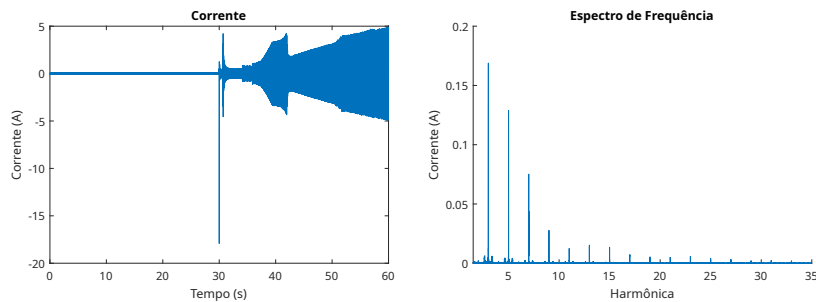
3.5 Análise harmônica da corrente

A análise harmônica será usada para extrair dados que auxiliam na identificação de carga. Alonso [116] desenvolveu uma estratégia para extrair apenas alguns componentes de fase e quadratura dos harmônicos de corrente. Para isso, ele utiliza a tensão medida para alimentar um algoritmo PLL, extraindo o ângulo de sincronização fundamental θ_{1m} . Isso é usado para calcular os ângulos de sincronização harmônica $\theta_{hm} = h \cdot \theta_{1m}$, que fornecem as referências para os componentes harmônicos. Esses ângulos são usados com a DFT da corrente capturada $i(t)$ para extrair os sinais de referência harmônicos de fase e quadratura $x_{h||m}$ e $x_{h\perp m}$, respectivamente. Esses valores têm seus ângulos de deslocamento normalizados, de modo que operações matemáticas, como a subtração da corrente em

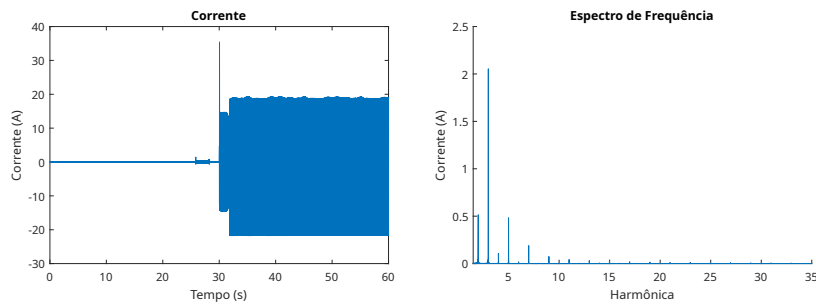
diferentes momentos no tempo, mostrem como os harmônicos mudaram ao longo do tempo.

A vantagem de usar a análise harmônica como um atributo da fase de identificação de carga é que o ângulo de deslocamento de um harmônico pode aumentar ou diminuir sua magnitude quando duas partes da forma de onda agregada são subtraídas após um aumento ou diminuição repentina de potência, mostrando que algumas cargas estão absorvendo ou gerando esse harmônico.

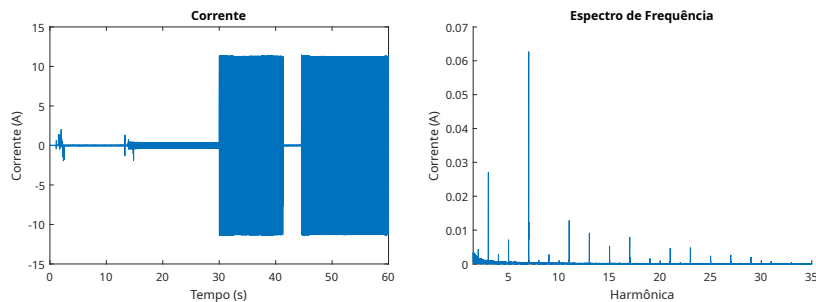
Para o algoritmo de desagregação usado serão usadas as componentes harmônicas de corrente até a 15ª, a frequência fundamental será descartada porque ela é correlata à potência ativa devido à Lei de Ohm e harmônicas além da 15ª também serão descartadas porque elas normalmente carregam pouca informação, como mostrado na Figura 42, que mostra um padrão de decréscimo contínuo a medida que a frequência aumenta.



(a) Ar-condicionado.



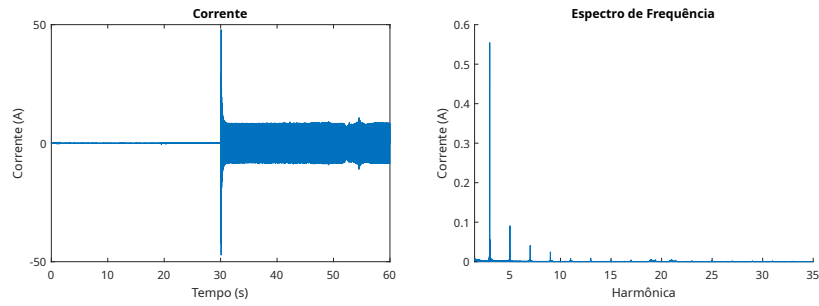
(b) Forno de micro-ondas.



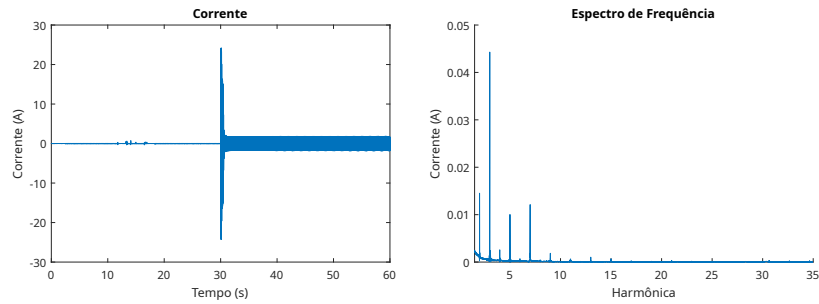
(c) *Air fryer*.

Figura 42 – Corrente das cargas e respectivas DFT.

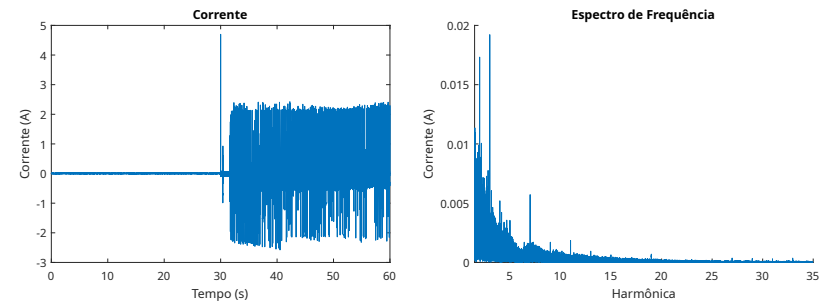
Fonte: Autoria própria.



(d) Aspirador de pó.



(e) Freezer.



(f) TV.

Figura 42 – Corrente das cargas e respectivas DFT.

Fonte: Autoria própria.

No lado esquerdo de cada um dos gráficos das cargas está a corrente medida e no lado direito está o módulo das componentes harmônicas de corrente geradas pela carga. O espectro de frequência foi calculado usando a DFT, baseando-se no recorte de tempo correspondente à sua respectiva carga. A frequência fundamental foi removida dos gráficos à direita para deixar evidenciadas as harmônicas subsequentes. Nestas figuras, empregou-se a DFT de forma não convencional para destacar as componentes harmônicas. Devido ao decréscimo contínuo de energia das componentes harmônicas, aquelas mais baixas, como a terceira harmônica, tendem a concentrar mais energia. A exceção é o *air fryer* em que a sétima harmônica concentra mais energia. Feitos testes de precisão, foi possível utilizar até a terceira harmônica no processo de classificação sem que houvesse alguma diminuição no *F-score* ($\beta = 1$), então a questão de usar a 15ª harmônica como corte se justifica somente

para o caso da desagregação de outra carga não testada neste trabalho. Para um cenário de cinco cargas no qual o F -score de classificação seja 91,53% utilizando até a 15ª harmônica, utilizar até a 3ª harmônica reduziu o F -score para 82,48%.

As harmônicas extraídas da aquisição não são usadas diretamente, é usado o RSS complexo da diferença entre o início e o fim do evento, conforme a Fórmula 3.19:

$$dI = \sqrt{\sum_{h=2}^n |x'_h|^2} = \sqrt{\sum_{h=2}^n (x_{h||m}^{\prime 2} + x_{h\perp m}^{\prime 2})}. \quad (3.19)$$

Sendo que:

$$x'_{h||m} = x_{h||m_{fim}} - x_{h||m_{inicio}}. \quad (3.20)$$

$$x'_{h\perp m} = x_{h\perp m_{fim}} - x_{h\perp m_{inicio}}. \quad (3.21)$$

Os sinais de referência $x_{h||m_{inicio}}$ e $x_{h||m_{fim}}$ são, respectivamente, os sinais de referência unitários de fase $x_{h||m}$ de cada harmônica para o início e fim do evento, e $x_{h\perp m_{inicio}}$ e $x_{h\perp m_{fim}}$ são os sinais de referência unitários de quadratura $x_{h\perp m}$ para o início e fim do evento, todos específicos para a harmônica h , o valor de n é o limite de harmônicas usadas nesse trabalho que vai até a 15ª.

3.6 Análise harmônica da potência

A análise harmônica da potência é usada como atributo alternativo à análise harmônica da corrente, explicada na seção 3.5. O método de extração deste atributo é similar à versão da corrente; a tensão adquirida é usada para alimentar um algoritmo de PLL para extrair o ângulo θ_{1m} e os ângulos θ_{hm} da sincronização harmônica. Isso é usado para calcular os ângulos de sincronização harmônica $\theta_{hm} = h \cdot \theta_{1m}$, que fornecem as referências para os componentes harmônicos. Esses ângulos são usados com a DFT da potência capturada $p(t)$ para extrair os sinais de referência harmônicos de fase e quadratura $x_{h||m}$ e $x_{h\perp m}$, respectivamente. Uma diferença em comparação à análise da harmônica de corrente é que, devido à frequência de variação de $p(t)$ ser o dobro da frequência da rede, o cálculo do RSS complexo calculado, similarmente à análise de corrente, deve ser somado de um, conforme a equação 3.22:

$$dV = \sqrt{\sum_{h=3}^{n+1} |x'_h|^2} = \sqrt{\sum_{h=3}^{n+1} (x_{h||m}^{\prime 2} + x_{h\perp m}^{\prime 2})}. \quad (3.22)$$

Sendo que:

$$x_{h||m'} = x_{h||m_{fim}} - x_{h||m_{inicio}}. \quad (3.23)$$

$$x_{h\perp m'} = x_{h\perp m_{fim}} - x_{h\perp m_{inicio}}. \quad (3.24)$$

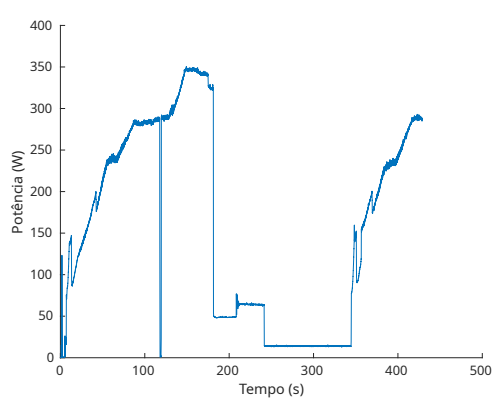
Os sinais de referência $x_{h||m_{inicio}}$ e $x_{h||m_{fim}}$ são, respectivamente, os sinais de referência unitários de fase $x_{h||m}$ de cada harmônica para o início e fim do evento, e $x_{h\perp m_{inicio}}$ e $x_{h\perp m_{fim}}$ são os sinais de referência unitários de quadratura $x_{h\perp m}$ para o início e fim do evento, todos específicos para a harmônica h , o valor de n é o limite de harmônicas usadas nesse trabalho que vai até a 15^a.

3.7 Desagregação de evento de ar-condicionado com diferentes setpoints

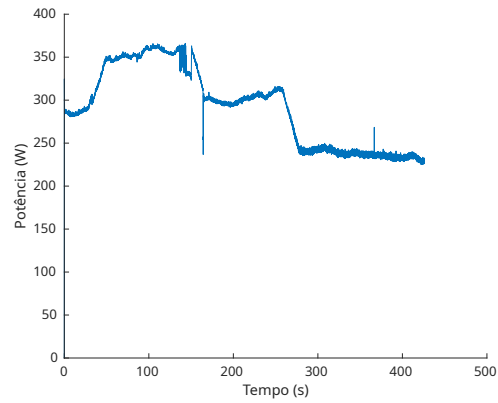
Aparelhos de ar-condicionado podem ter a sua temperatura alterada arbitrariamente pelo usuário, então torna-se necessário que o sistema NILM seja capaz de efetuar a correta desagregação em situações como esta. Foram capturados dados de consumo do ar-condicionado nos modos de resfriamento e aquecimento e em temperaturas diversas. A mudança de setpoint foi feita conforme a lista a seguir:

- 20°C - frio,
- 16°C - frio,
- 22°C - frio,
- 24°C - calor,
- 28°C - calor.

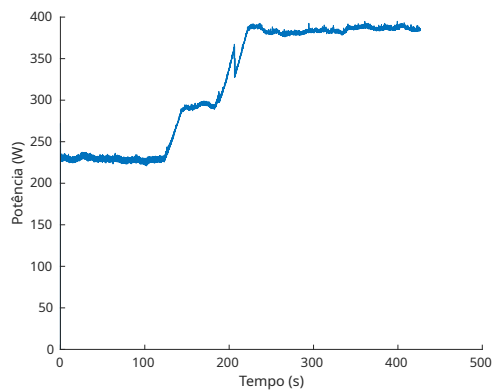
As alterações no setpoint causaram de fato uma variação de potência no ar condicionado, como mostra a Figura 43. Pode ser notado na Figura 43a, que o início do ar-condicionado é feito duas vezes, com o segundo início sendo aproximadamente depois do segundo 200. As variações de potência então seguem um padrão complexo, com aumentos e reduções de potência. A conclusão sobre estas figuras é que tanto o algoritmo detector de evento quanto o algoritmo de desagregação devem funcionar apesar do comportamento aleatório característico para esse tipo de carga.



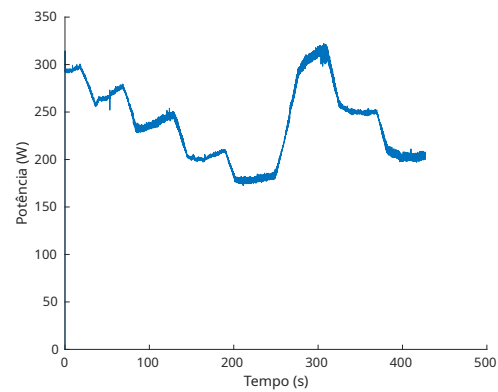
(a) Sequencia 1.



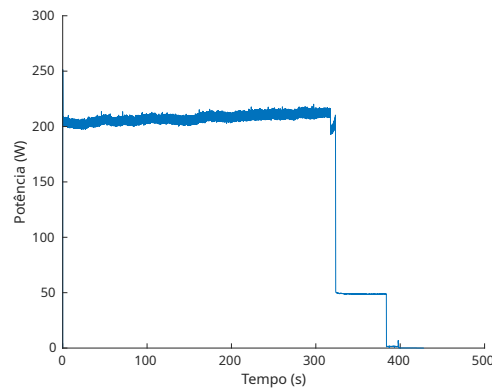
(b) Sequencia 2.



(c) Sequencia 3.



(d) Sequencia 4.



(e) Sequencia 5.

Figura 43 – Consumo de ar-condicionado.

Fonte: Autoria própria.

3.8 Extração de dados de aprendizado

Antes da extração dos dados de aprendizado, o Limiar Mínimo ΔP deve ser determinado segundo a seção 3.3.1 para as cargas que serão desagregadas. A extração de dados para aprendizado utiliza os mesmos algoritmos de detecção de eventos e extração de características explicadas nas seções anteriores. Os eventos de curta e longa duração são gravados a partir dos dados de aprendizado em uma lista de eventos juntamente com

seus respectivos rótulos e salvos para uso posterior. É adicionado um evento extra de desligamento para uma corrente fictícia de zero se a potência ativa da CPT for maior que o ΔP já calculado. Isso é necessário para que, durante a fase de modelamento e simulação, apresentada na seção 3.10, o algoritmo consiga detectar corretamente o desligamento abrupto da carga.

3.9 Seleção de Dados

Observou-se que cargas de alto consumo tendem a ter mais padrões de variação do que aquelas de baixo consumo, dificultando sua separação usando os algoritmos de aprendizado de máquina propostos. A solução para isso é a implementação de seleção de dados, na qual dados cujo ΔP cai abaixo de um certo limiar são excluídos dos dados de treinamento. O limite é calculado tomando o ΔP máximo de todas as cargas e calculando uma porcentagem F_3 para as cargas avaliadas neste trabalho, e excluindo os dados de treinamento cujo ΔP está abaixo desse resultado. O valor de 3,5% foi escolhido, maximizando o resultado do F -score ($\beta = 1$) médio de vários cenários de desagregação de carga, e utilizando o algoritmo de classificação KNN, cujos resultados são evidenciados na Tabela 7:

F_3	0%	1%	1,5%	2%	2,5%	3%	3,5%	4%	4,5%	5%
F -score médio	29%	34%	74%	71%	70%	91%	91%	91%	90%	90%

Tabela 7 – Comparação entre diferentes valores de F_3 .

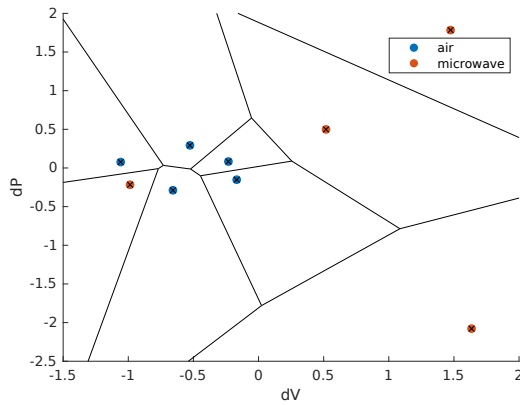
Fonte: Autoria própria.

O Limiar Mínimo calculado na Seção 3.3.1 deve ser recalculado se algum dado for cortado pelo ΔP mínimo encontrado nos dados de treinamento recém-selecionados. Finalmente, o novo valor é o ΔP mínimo encontrado nos dados de carga restantes. A desvantagem dessa seleção é que algumas combinações de cargas não serão desagregadas de forma confiável se a maioria ou todos os seus dados de treinamento forem excluídos. As etapas de coleta de dados de treinamento executando o detector de eventos e obtendo uma lista de eventos das cargas devem ser executadas novamente para o novo Limiar Mínimo. O código do MATLAB da Figura 64, no apêndice D mostra como a correção é feita.

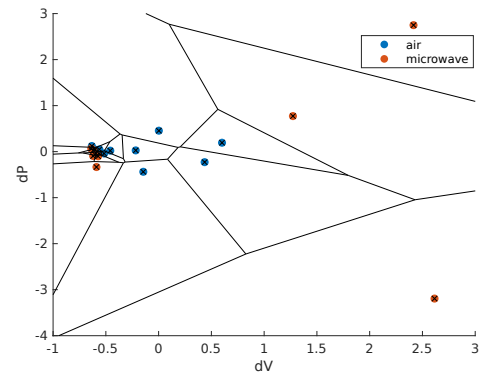
Os valores de retorno *threshold_power* e *loads* são, respectivamente, o Limiar Mínimo e os dados que são usados com os algoritmos de inteligência artificial. O valor de retorno *cut* são os dados que foram cortados.

Os diagramas de Voronoi das Figuras 44, 45, 46 e 47 mostram as fronteiras de decisão com o seletor de dados desligado à esquerda e com o seletor de dados ligados à direita. São comparadas a diferença de potência ativa (dP), a diferença de energia reativa (dW), a diferença das harmônicas da potência residual (dV) e a diferença de harmônicas

de corrente (dI). Para todos os casos, as áreas cujas fronteiras de decisão são próximas foram removidas, facilitando o emprego de algoritmos de aprendizado de máquina que usam menos poder computacional, como os utilizados nesse trabalho. Foram usadas duas cargas para a comparação: um forno de micro-ondas e um ar-condicionado com inversor.



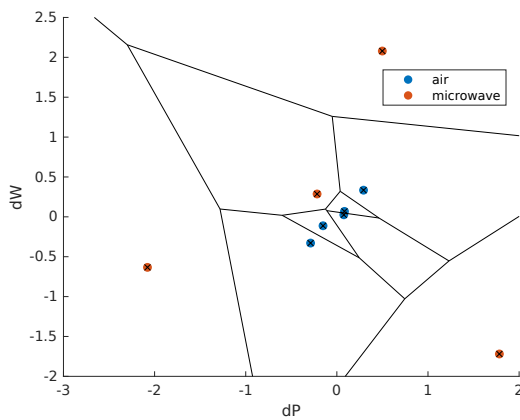
(a) Seletor de dados ligado.



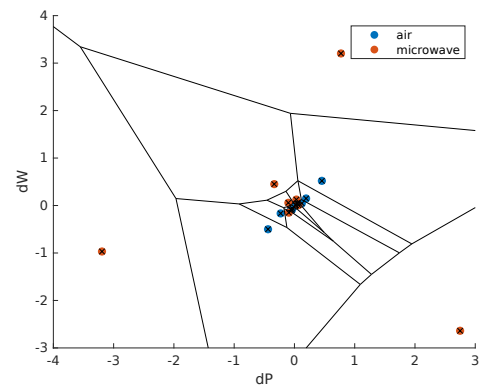
(b) Seletor de dados desligado.

Figura 44 – Diagramas de Voronoi comparando o seletor de dados (dP x dV).

Fonte: Autoria própria.



(a) Seletor de dados ligado.



(b) Seletor de dados desligado.

Figura 45 – Diagramas de Voronoi comparando o seletor de dados (dW x dP).

Fonte: Autoria própria.

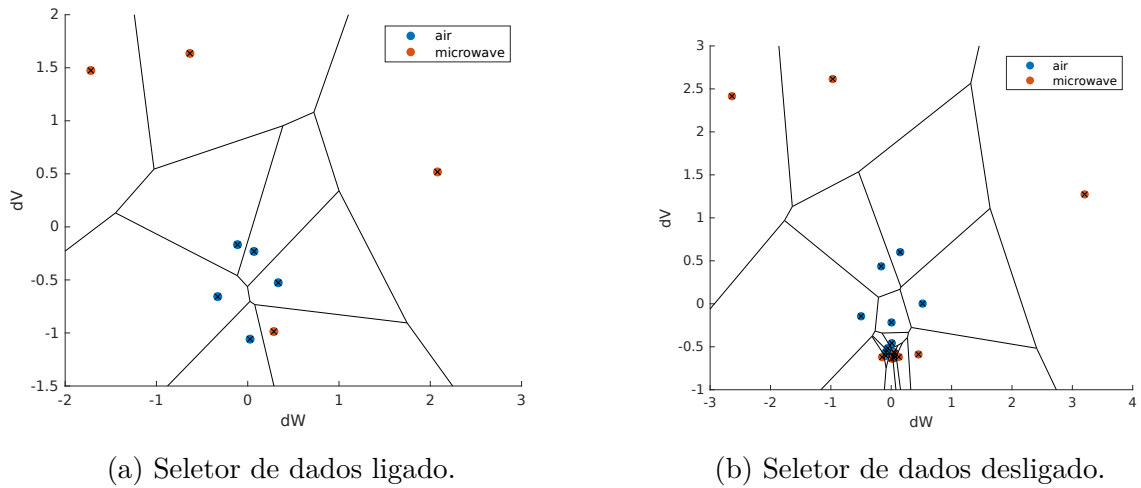


Figura 46 – Diagramas de Voronoi comparando o seletor de dados (dV x dW).

Fonte: Autoria própria.

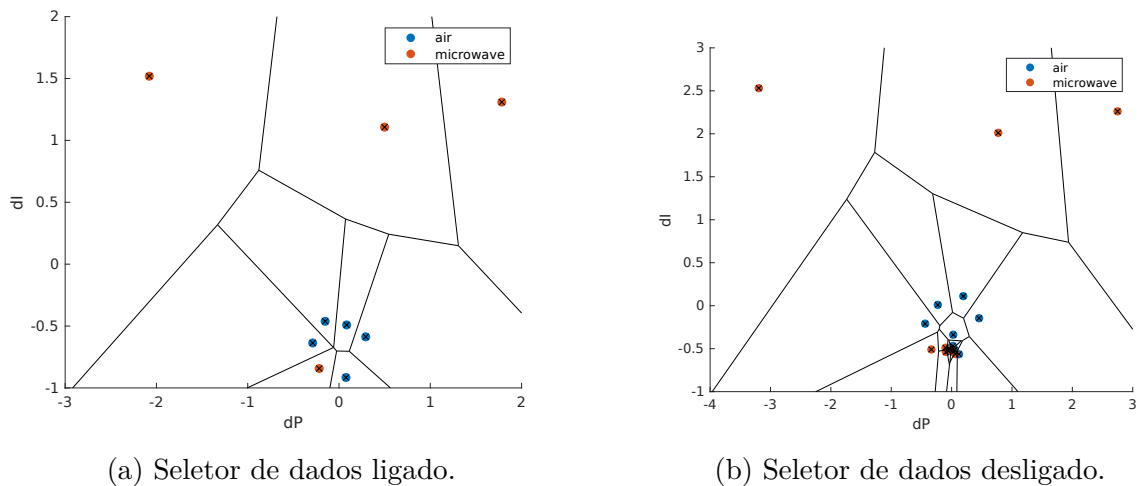


Figura 47 – Diagramas de Voronoi comparando o seletor de dados (dI x dP).

Fonte: Autoria própria.

3.10 Modelamento de cargas para teste de validação

Foram criados modelos de simulação no MATLAB/Simulink para a criação de padrões de consumo agregados. Eles utilizam registros da forma de onda de corrente no tempo para serem usados em um gerador de corrente dentro do Simulink. Há dentro do modelo um detector de passagem por zero da frequência fundamental da rede, alimentado pela tensão simulada, que serve para sincronizar o recorte da corrente, para que a característica de potência da carga seja reproduzida com as mesmas características da medição original. O modelo também tem uma entrada para configurar o momento no tempo que a carga deve iniciar. O desligamento do modelo da carga acontecerá quando acabarem os dados de corrente recortados.

Os dados usados nos recortes para criar os modelos serão extraídos do conjunto

de dados obtidos para essa dissertação. O recorte de corrente escolhido será cortado utilizando um detector de passagem por zero da frequência fundamental da rede na tensão medida, para que a potência reproduzida posteriormente na simulação tenha as mesmas características de potência.

Devido às pequenas diferenças entre a frequência da rede no momento da aquisição e a frequência da forma de onda de tensão sintetizada pelo Simulink, foi necessário a adição de um passo de reamostragem do sinal de tensão e corrente adquirido. Os cruzamentos por zero da tensão adquirida são encontrados e são calculados os cruzamentos por zero da onda sintética. Esses conjuntos de passagens por zero são utilizados para alimentar a função *interp1* do MATLAB, que adiciona ou remove amostras tanto da voltagem quanto da corrente adquirida para que seus cruzamentos por zero coincidam.

Sem este ajuste, as formas de onda de corrente adquiridas criam um deslocamento entre a tensão sintética que pode ser facilmente notado quando a potência ativa e a energia reativa da CPT são calculadas. Após o processamento de segundos ou até minutos de dados, este deslocamento fica evidente, surgindo como uma anomalia na potência ativa e energia reativa calculada pelo Simulink.

3.10.1 Dificuldades na simulação de harmônicas de tensão

Os dados de padrão de consumo sintetizados pelo Simulink são incapazes de simular as harmônicas de tensão corretamente. O circuito da simulação consiste em várias cargas adicionadas em paralelo, e como nesta configuração a tensão aplicada nelas é a mesma, não é possível simular corretamente o circuito resultante, já que a tensão adquirida é diferente para cada carga e para haver uma simulação precisa, a tensão original medida por cada carga deveria ser aplicada.

A Figura 48 mostra a tensão sintética e a tensão medida de uma carga. É possível notar que a diferença é pequena, mas há um impacto no cálculo das parcelas de potência da CPT e dos outros atributos elétricos usados.

3.10.2 Dificuldades na simulação de queda de tensão

Os dados de padrão de consumo sintetizados pelo Simulink são incapazes de simular corretamente a queda de tensão para cargas medidas em pontos diferentes da residência. O circuito da simulação consiste em várias cargas adicionadas em paralelo, mas no circuito original da residência, estas cargas estão distribuídas em vários sub-circuitos espalhados com diferentes impedâncias de linha, que por sua vez, causam diferentes quedas de tensão, alterando de maneira diferente os padrões de consumo em comparação com a queda de tensão original que cada carga experimenta. Devido a essa deficiência, não houve uma tentativa de simular uma impedância de linha para os casos de desagregação apresentados.

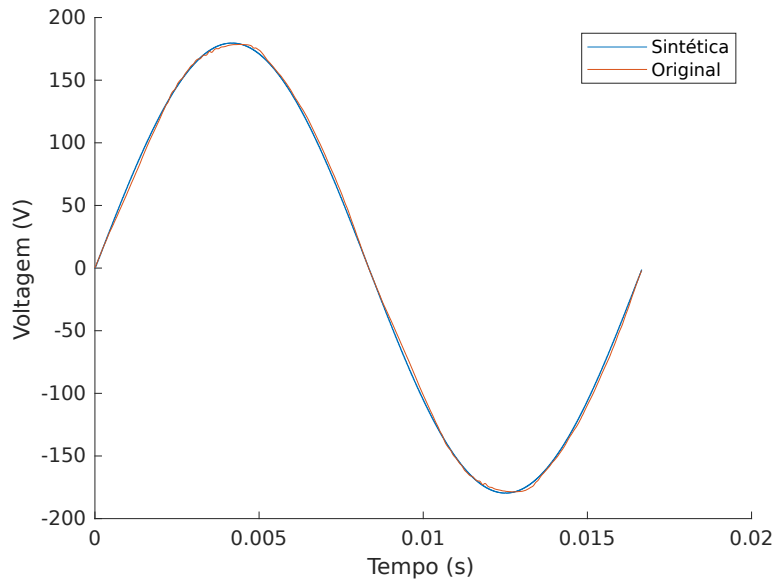


Figura 48 – Diferença de tensão sintética e original.

Fonte: Autoria própria.

3.10.3 Dificuldades na simulação da frequência de rede

Existe uma deficiência na sintetização dos dados de corrente no Simulink. Devido as pequenas diferenças e irregularidades de frequência da rede nos dados adquiridos em relação à forma de onda de tensão geradas pelo Simulink, foi necessário aplicar um algoritmo de interpolação para que as passagens por zero da tensão e corrente adquiridas coincidam com as tensões sintéticas geradas pelo Simulink, de modo que as potências ativas e reativas calculadas durante o processo estejam preservadas.

3.11 Metodologia de desagregação

Com os pontos de início e fim do evento adquiridos, a potência ativa P e a energia reativa W_{sgn} são calculadas a partir da teoria de potência CPT, e até o 15º harmônico de um ciclo da forma de onda de corrente são calculados e processados segundo a análise de corrente harmônica mencionada na seção 3.5. Todos eles são calculados a partir dos pontos inicial e final do evento. Nenhum deslocamento é necessário para as potências ativa e reativa, uma vez que a teoria de potência CPT garante que uma simples subtração é suficiente para obter um aumento ou diminuição de suas respectivas potências. As características utilizadas para reconhecer a carga são as diferenças das respectivas grandezas calculadas, a saber, a diferença de potência ativa dP calculada a partir de P , a diferença de energia reativa dW calculada a partir de W_{sgn} e o RSS da diferença nos harmônicos dos vetores de corrente dI calculada a partir da análise harmônica.

Existem duas abordagens diferentes de desagregação, dependendo da velocidade

do evento e das outras cargas em paralelo. Quanto mais rápido for o evento, menor a influência das outras cargas sobre ele; portanto, para eventos de curta duração, as outras cargas podem ser ignoradas com pouco impacto na desagregação, e para eventos de longa duração, as outras cargas são levadas em consideração devido ao impacto crescente em sua desagregação.

Para determinar a melhor abordagem, o algoritmo de desagregação categoriza os eventos em dois tipos: “LENTO” e “RÁPIDO”. O tipo de evento “RÁPIDO” é tratado individualmente e suas características são usadas para classificar a carga. O tipo de evento “LENTO”, que pode ser combinado com as cargas de outros eventos “LENTO”, é usado para determinar o padrão de consumo das cargas. Observou-se que a partida de uma carga geralmente é um evento “RÁPIDO”, portanto, quando um evento “LENTO” é detectado, presume-se que este evento pertence a uma carga já ligada, a menos que não haja cargas detectadas no momento; neste caso, este evento “LENTO” específico é tratado como um evento “RÁPIDO”. A detecção de um evento “LENTO” com cargas ligadas faz com que o algoritmo atualize o consumo de energia das cargas.

Os dados de aprendizado usados para classificar o tipo de evento “RÁPIDO” são uma lista de eventos para cada carga, juntamente com seus respectivos dP , dW e dI , calculados previamente executando o detector de eventos em cada carga e extraíndo as diferenças de P , W_{sgn} e harmônicas. O sinal de dP , chamado neste trabalho de *type*, é usado como um atributo extra, pois ajuda a separar cargas com baixa variação. A desagregação de um evento “RÁPIDO” emprega uma abordagem mais direta, na qual os atributos dP , dW , dI e *type* são alimentados diretamente ao algoritmo de aprendizado de máquina.

Para o evento do tipo “LENTO”, os dados de aprendizagem utilizados para a classificação são criados especificamente para as cargas ligadas, prevendo-se a ocorrência de eventos “LENTO” simultâneos. Esses dados de aprendizagem mudam dependendo de quais cargas são ligadas em um determinado momento. Eles são produzidos calculando-se a combinação simples dos dados de aprendizagem das cargas individuais marcadas como um evento “LENTO”. A combinação de cargas é calculada para cada grupo de cargas ligadas possíveis. Por exemplo, se houver três cargas ativas, as combinações dos eventos de uma, duas ou três cargas são calculadas. Para o resultado esperado da classificação, um código especial é criado para identificar as cargas individuais e seu respectivo consumo, que será posteriormente atribuído às cargas corretas. Como já são conhecidas as cargas ativas devido aos eventos anteriores, apenas o consumo de energia precisa ser determinado.

Para desagregar o evento do tipo “LENTO”, são utilizados os atributos sP , sW . Eles são calculados dividindo dP e dW de cada carga envolvida pelos comprimentos de seus respectivos eventos e, em seguida, somando-os para formar a combinação simples das cargas envolvidas. Os algoritmos de classificação testados para eventos “RÁPIDO” incluem KNN, MLP e SVM, selecionados por seu baixo custo computacional, tempos de

treinamento rápidos e precisão relativamente alta para a tarefa, conforme demonstrado por Souza [7]. Para eventos do tipo “LENTO”, o KNN foi escolhido devido a não necessitar de treinamento, tornando-o ideal para geração de atributos em tempo real.

É necessário desenvolver um algoritmo para determinar qual evento é considerado “RÁPIDO” ou “LENTO”. Os eventos de longa duração são sempre considerados “LENTO” e os eventos de curta duração devem satisfazer certas condições. Um evento de curta duração é considerado “LENTO” se o módulo da velocidade de variação de dP pelo comprimento do evento for menor que um quinto da velocidade de variação de ΔP pelo comprimento do *buffer* de tamanho S . Caso essa condição não seja satisfeita, ele é considerado “RÁPIDO”. A necessidade dessa classificação se dá pelo fato de certos eventos variarem lentamente, apesar de durarem pouco tempo.

3.12 Parâmetros de Inteligência artificial usados e diferenças entre plataformas

Foram usados os seguintes parâmetros para configurar os algoritmos de inteligência artificial:

- KNN:

No classificador KNN, a configuração de vizinho mais próximo k foi deixada em 1 para todas as plataformas.

- MLP:

Para o classificador MLP as configurações diferem entre plataformas devido à disponibilidade e desempenho. Os parâmetros em comum são:

- Camadas ocultas: 10,
- Neurônios das camadas ocultas: 40,
- Critério de parada: 10^{-9} ,
- Camada de saída: Softmax.

Para o MATLAB, os seguintes parâmetros são usados:

- Função de ativação das camadas ocultas: Tangente hiperbólica sigmoide (*tansig*),
- Função de treinamento: Gradiente descendente com *momentum* e taxa adaptativa (*traingdx*),
- Taxa de aprendizado: 0.01

Para a implementação em C++ para 1024 amostras por ciclo, certos parâmetros são trocados, conforme mostrado a seguir:

- Função de ativação das camadas ocultas: Unidade Linear Retificada (ReLU),
- Função de treinamento: Gradiente descendente com política de atualização Adam,
- Taxa de aprendizado: 0.01

Para a implementação em C++ para 128 amostras por ciclo, certos parâmetros são trocados, conforme mostrado a seguir:

- Função de ativação das camadas ocultas: Unidade Linear Retificada (ReLU),
- Função de treinamento: Gradiente descendente,
- Taxa de aprendizado: 0.02.

O uso da função de ativação tansig e da função de treinamento traingdx na implementação do MATLAB tornaria desnecessária a alteração de parâmetros para as implementações em C++, mas haveria a penalidade de desempenho da execução de uma tangente hiperbólica caso a função transig fosse usada, e a biblioteca MLPack não oferece uma função de treinamento equivalente a traingdx.

- SVM:

Já o classificador SVM foi configurado com os seguintes parâmetros:

- Função *kernel*: Função de base radial (rbf),
- Valor γ : 10,
- Valor C: 10.

3.13 Implementação em C++ do algoritmo

Após o desenvolvimento do algoritmo no MATLAB, foi criada a versão em C++ para que um teste de desempenho em hardware real pudesse ser criado. O teste foi feito em três plataformas distintas: X86_64, ARM® Cortex-A7 e ARM® Cortex-A72. As próximas subseções contêm as características de cada plataforma. Esta implementação carrega na memória todos os dados de cenários antes de processá-los e gerar a lista de atributos usados na desagregação. Isso é útil somente para facilitar a medição de desempenho dos algoritmos. Em um ambiente real, somente uma carga deve ser analisada por vez, já que haverá somente um dispositivo.

3.13.1 Plataforma 1

Este sistema é um PC Desktop que também foi usado para o desenvolvimento inicial do algoritmo no MATLAB. O sistema tem as seguintes características de hardware:

- Processador: Intel Core i9-9900K 1 processador físico; 8 núcleos; 16 *threads* @ 3,6GHz;
- Memória: 32GB DDR4 SDRAM;
- Armazenamento: 1,8TB SATA;
- GPU: nVidia GeForce RTX 4060 Ti 8GB de GDDR6;
- Sistema Operacional POP! OS 22.04, baseado em Linux.

3.13.2 Plataforma 2

Esta é a placa dedicada usada pelo Medidor de Qualidade de Energia AIW-PEM e AIW-PEX, usados para a aquisição de dados. Eles têm as seguintes características de hardware:

- SoC: A40i;
- Processador: Quad-Core ARM Cortex-A7 @ 1.2 GHz;
- Memória: 2GB DDR3 SDRAM;
- Armazenamento: 16GB eMMC;
- GPU: ARM® Mali™400 MP2;
- 1x Porta Ethernet 100MBps;
- 2x Portas USB 2.0;
- 1x Porta USB 2.0 Tipo-C OTG;
- 1x Porta HDMI;
- 1x Porta MIPI DSI;
- 18x GPIOs com SPIs, I^2C s, USARTs e S/PDIF;
- Modem LTE (4G) embutido;
- Dimensões 120x100 mm;
- Sistema Operacional Ubuntu 18.04, baseado em Linux.

3.13.3 Plataforma 3

Esta placa é a Tinkerboard 2s da RockChip. Foi usada para o teste de desempenho por ser um processador ARM de 64bits (big.LITTLE). Seguem as especificações de hardware:

- SoC: RK3399;
- Processador: Dual-core Arm® Cortex®-A72 @ 2.0 GHz;
- Processador: Quad-core Arm® Cortex®-A53 @ 1.5 GHz;
- Memória: 2GB DDR4 LPDDR4;
- Armazenamento de Massa: 16GB eMMC;
- GPU: Arm® Mali™-T860 MP4 GPU @ 800 MHz;
- 1x Porta Ethernet 100MBps;
- 3x Portas USB 3.2;
- 1x Porta USB 3.2 Tipo-C OTG;
- 1x Porta HDMI;
- 1x Porta MIPI CSI-2 (2 pistas);
- 40x GPIOs com SPIs, I^2C s, USARTs, PWM e S/PDIF;
- Dimensões 85x56 mm;
- Sistema Operacional Debian 11, baseado em Linux.

3.13.4 Bibliotecas utilizadas

Nesta seção estão descritas as bibliotecas usadas na implementação em C++ do algoritmo. O dialeto do C++ usado para compilar o código é o C++20. Ele depende de bibliotecas externas, e algumas dessas dependências devem ser instaladas usando o gerenciador de pacotes de um sistema baseado em GNU/Linux. Não foram feitos testes para compilar esse código utilizando o WSL para Windows ou compilação em sistemas Macintosh. No POP! OS 22.04, baseado na distribuição Debian, todas as dependências relevantes podem ser instaladas conforme a seguir:

```
# sudo apt-get install libarmadillo-dev libensmallen-dev libcereal-dev  
libmatio-dev ocaml ocamlbuild libnum-ocaml-dev texinfo build-essential cmake
```

Os nomes das bibliotecas podem variar dependendo da distribuição. Certas distribuições podem não conter as referidas bibliotecas ou estas podem ser antigas. Para estes casos, estas bibliotecas devem ser compiladas a partir de seu código-fonte. Outras dependências, como o `cpt`, o `fftw`, o `mlpack` e o `libsvm`, são instaladas como submódulos no repositório git da implementação e serão automaticamente baixadas da internet pelo *script* de compilação. Essa implementação foi compilada nas plataformas descritas na seção 3.13. Para efetuar a compilação do código, executar o comando *build.sh* conforme a seguir:

```
# ./build.sh
```

O executável está localizado em `./build/disaggregator`. A descrição das bibliotecas vem a seguir:

- `gcc >= 11.4.0`;

O compilador C++.

- `cmake >= 3.14`;

Utilizado para automatizar a compilação do código.

- `armadillo >= 10.8`;

- `ensmallen >= 2.10.0`;

- `mlpack >= 4.6.2`;

O `mlpack` é uma biblioteca de algoritmos de Machine Learning e Inteligência Artificial, escrita em C++, construída sobre a biblioteca `armadillo` e a biblioteca de otimização numérica `ensmallen`.

- `libsvm >= 3.36`;

Esta biblioteca é exclusiva para algoritmos de Inteligência Artificial do tipo Support Vector Machine (SVM).

- `cereal >= 1.1.2`;

Utilizada para gerar os arquivos utilizados nos algoritmos de Inteligencia Artificial.

- `matio >= 1.5.21`;

Executa a leitura dos arquivos de forma de onda de tensão e corrente para teste do algoritmo.

- `ocaml >= 4.13.1`;

- `ocamlbuild >= 0.14.1`;

- `libnum-ocaml-dev >= 1.4`;

- texinfo ≥ 6.8 ;
- fftw $\geq 3.3.10$;

Biblioteca que calcula a DFT de um sinal com tamanhos arbitrários. As bibliotecas ocaml, ocamlbuild, libnum-ocaml-dev e texinfo são bibliotecas auxiliares usadas na compilação do fftw. Esta biblioteca tem uma licença dupla GPL e proprietária que limita seu uso na indústria.

- cpt;

Calcula a CPT a partir dos dados de tensão e corrente.

- matplotlib++;

Usada para gerar gráficos com propósitos de depuração, com uma API similar ao MATLAB.

3.14 Outros usos da lista de eventos

A lista de eventos pode ser usada para alimentar algoritmos diferentes de inteligência artificial, especificamente algoritmos que buscam sequências como o LSTM, Transformer, e outros. Estes podem ser úteis ao observar o comportamento do início das cargas testadas nesse trabalho: todas as cargas testadas têm uma sequência de início bem definida. Além disso, cargas de baixo consumo que se tornaram difíceis de serem detectadas ao estarem combinadas com cargas de alto consumo podem, potencialmente, ser corretamente classificadas usando esses tipos de algoritmos.

O detector de eventos também pode ser usado por outros algoritmos de NILM, principalmente aqueles que não são baseados em eventos. Para estes casos, o detector de eventos funcionaria como uma espécie de filtro para reduzir o processamento no momento da classificação da carga.

4 Resultados

Neste capítulo são apresentados os resultados deste trabalho. Nas próximas seções serão apresentados o resultado do algoritmo de detecção de eventos, a precisão da desagregação de equipamentos de ar-condicionado baseados em inversor combinadas com outras cargas, inclusive com a alteração do *setpoint* de temperatura do equipamento e do tempo de execução da implementação em C++ para taxas de amostragem de 1024 e 128 amostras por ciclo.

4.1 Resultados do Detector de Eventos

Nesta seção são apresentados os resultados do algoritmo de detecção de eventos. Os asteriscos vermelhos dos gráficos são as posições do final dos eventos de curta duração e os verdes são as posições do final dos eventos de longa duração.

Na Figura 49 é possível notar uma sequência ascendente de asteriscos verdes e vermelhos. Isso é causado pelo algoritmo de detecção de eventos quando a carga com inversor aumenta gradativamente o consumo de energia. Isso será usado pelo algoritmo de desagregação para estimar o consumo desta carga e para ajudar na desagregação de outro evento que venha acontecer durante este.

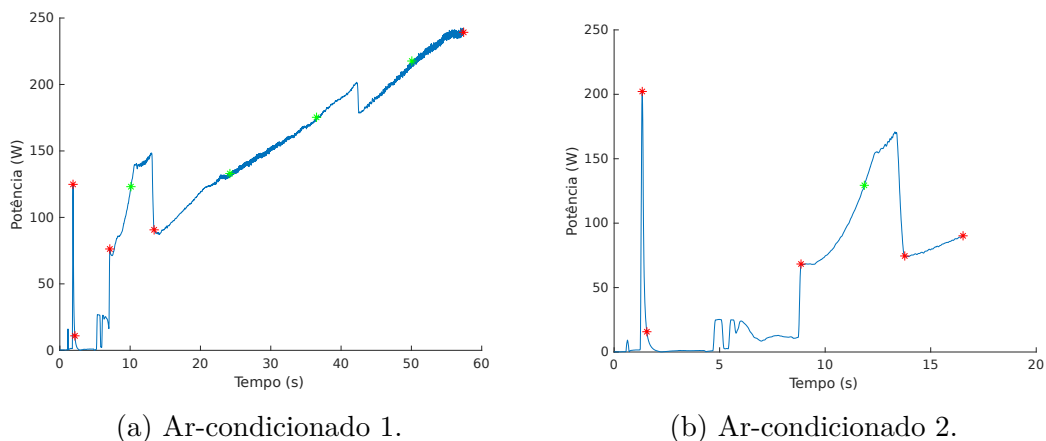
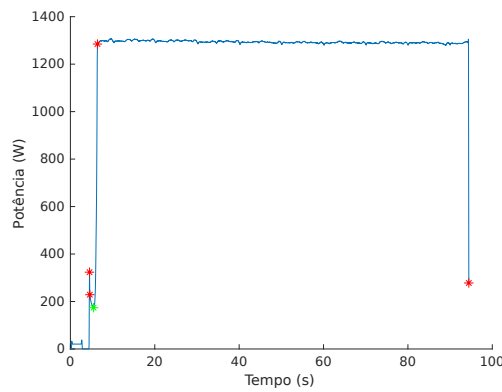
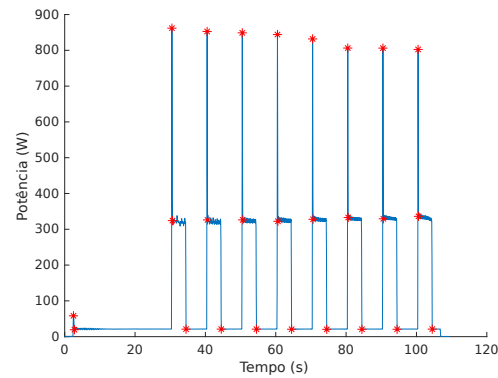


Figura 49 – Consumo de cargas com eventos assinalados. Asteriscos vermelhos: eventos de curta duração. Asteriscos verdes: eventos de longa duração.

Fonte: Autoria própria.



(c) Micro-ondas.



(d) Máquina de Lavar.

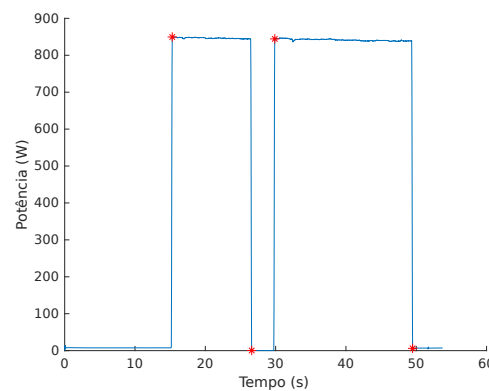
(e) *Air Fryer*.

Figura 49 – Consumo de cargas com eventos assinalados. Asteriscos vermelhos: eventos de curta duração. Asteriscos verdes: eventos de longa duração.

Fonte: Autoria própria.

A seguir estão listados as os pontos gerados pelo detector de eventos para cada carga mostrada na Figura 49:

- Ar-condicionado 1:

type	speed	time	length	dP	dW	dV	dI
UP	RÁPIDO	1.8786	0.098486	123.4896	-70.3159	76.7581	0.65768
DOWN	RÁPIDO	2.1453	0.26668	-113.9536	62.4665	57.5955	0.52005
UP	RÁPIDO	7.1111	0.13223	59.922	-2.4425	28.4242	0.19322
UP	LENTO	10.1111	3	46.8662	-23.4455	112.6287	0.79514
DOWN	RÁPIDO	13.4127	0.3513	-57.7993	21.2102	132.5519	0.94739
UP	LENTO	24.1884	10.7757	42.3371	-16.7631	107.6643	0.74523
UP	LENTO	36.5117	12.3232	42.3377	-12.9666	47.4156	0.30457
UP	LENTO	50.0638	13.552	42.3411	-71.705	82.5833	1.1057
DOWN	RÁPIDO	57.4333	1.6276e-05	-239.2416	137.7015	201.4823	1.5555

- Máquina de Lavar:

type	speed	time	length	dP	dW	dV	dI
UP	RÁPIDO	2.4923	0.16667	57.5244	94.2401	7.0332	0.089711
DOWN	RÁPIDO	2.7769	0.25	-41.6416	-41.9017	4.8775	0.05181
UP	RÁPIDO	30.4823	0.12983	840.6531	152.5046	101.3624	0.95341
DOWN	RÁPIDO	30.6299	0.14762	-537.7445	-251.9611	79.9003	0.60517
DOWN	RÁPIDO	34.5135	0.15	-298.8505	99.8693	33.3234	0.24177
UP	RÁPIDO	40.489	0.12337	831.5691	133.5332	71.8426	0.60498
DOWN	RÁPIDO	40.6162	0.12725	-526.5294	-232.2678	76.7566	0.6147
DOWN	LENTO	44.4056	3.7893	-42.3415	92.5937	32.0171	0.22249
DOWN	RÁPIDO	44.5904	0.19266	-276.3517	2.9075	1.0327	0.0087739
UP	RÁPIDO	50.487	0.12822	827.6598	128.7038	32.822	0.32623
DOWN	RÁPIDO	50.6313	0.14429	-522.594	-227.8993	27.4393	0.37348
DOWN	LENTO	54.4066	3.7753	-42.378	92.8305	33.8623	0.24702
DOWN	RÁPIDO	54.5319	0.13333	-277.2739	2.2756	1.3868	0.010954
UP	RÁPIDO	60.488	0.12747	822.806	124.6468	43.3311	0.39737
DOWN	RÁPIDO	60.6187	0.13078	-521.6783	-223.4658	44.9501	0.43145
DOWN	LENTO	64.4047	3.7859	-42.4043	91.8535	32.6701	0.24146
DOWN	RÁPIDO	64.5252	0.12852	-274.4236	3.599	1.1081	0.0090998
UP	RÁPIDO	70.4785	0.11842	810.9907	139.1496	91.9114	0.7637
DOWN	RÁPIDO	70.6052	0.12673	-503.9227	-237.5646	68.1836	0.58815
DOWN	RÁPIDO	74.504	0.15262	-303.717	100.25	33.4105	0.24596
UP	RÁPIDO	80.473	0.1217	785.1749	119.7716	80.5868	0.59502
DOWN	RÁPIDO	80.5993	0.1263	-473.4813	-220.3521	62.5938	0.47619
DOWN	LENTO	84.3907	3.7914	-42.345	9.1359	32.0582	0.23167
DOWN	RÁPIDO	84.4983	0.1118	-280.9015	0.74335	0.94627	0.0073163
UP	RÁPIDO	90.4698	0.12155	785.0762	106.3754	75.771	0.55756
DOWN	RÁPIDO	90.5952	0.12544	-476.8654	-205.7121	56.1573	0.4284
DOWN	RÁPIDO	94.4928	0.13464	-309.0674	102.1037	102.9455	1.0354
UP	RÁPIDO	100.4674	0.12186	781.1389	107.8314	63.9336	0.48267
DOWN	RÁPIDO	100.5841	0.11764	-466.7938	-208.7816	53.1393	0.38511
DOWN	LENTO	104.3826	3.7985	-42.3842	99.245	32.0684	0.22771
DOWN	RÁPIDO	104.4899	0.10833	-276.3471	1.98	0.84203	0.0078625

- Ar-condicionado 2:

type	speed	time	length	dP	dW	dV	dI
UP	RÁPIDO	1.3333	0.10492	200.5889	115.4536	93.0779	0.83798
DOWN	RÁPIDO	1.552	0.21867	-186.4652	-104.6246	74.233	0.68276
UP	RÁPIDO	8.8559	0.13597	56.9262	14.1831	3.0205	0.014223
UP	LENTO	11.8559	3	61.0803	27.5842	148.4458	1.037
DOWN	RÁPIDO	13.771	0.39167	-95.8612	-34.152	158.2399	1.1045
DOWN	RÁPIDO	16.5333	1.6276e-05	-90.1776	-31.9506	72.2098	0.50886

- Micro-ondas:

type	speed	time	length	dP	dW	dV	dI
UP	RÁPIDO	4.5299	0.1228	323.4942	632.0352	253.2914	4.5949
DOWN	RÁPIDO	4.6049	0.075016	-94.8672	61.3981	10.8192	0.15643
DOWN	LENTO	5.4917	0.8951	-55.5043	20.3997	14.583	0.15227
UP	RÁPIDO	6.4083	0.63866	1090.3009	-548.927	428.3071	5.7098
DOWN	RÁPIDO	94.4333	1.6276e-05	-278.0668	-27.2202	1.0905	0.0078973

- *Air Fryer*:

type	speed	time	length	dP	dW	dV	dI
UP	RÁPIDO	15.2968	0.14316	841.8863	482.8277	67.9511	0.34
DOWN	LENTO	26.4713	11.1745	-42.3372	-24.0835	129.0241	0.8734
DOWN	RÁPIDO	26.5964	0.13333	-829.8157	-499.4575	321.2341	3.0948
UP	RÁPIDO	29.8801	0.1493	844.0866	513.4735	42.6955	0.17174
DOWN	LENTO	49.3638	19.4837	-42.3605	-22.4668	43.374	0.16873
DOWN	RÁPIDO	49.4794	0.12391	-819.626	-474.7621	366.5575	3.5376

O valor de ΔP calculado por esses resultados é de 42,3370.

4.2 Resultados da desagregação

Foram testados três cenários com combinação de cargas: duas cargas, três cargas e cinco cargas. As informações entre parênteses são os tempos de início de cada carga. Os asteriscos vermelhos representam os eventos de curta duração e os verdes, os eventos de longa duração nos gráficos de potência dos cenários. As configurações da carga foram as seguintes:

- Cenário 1:
 - Ar-condicionado (no início);
 - Forno de micro-ondas (no segundo 10).

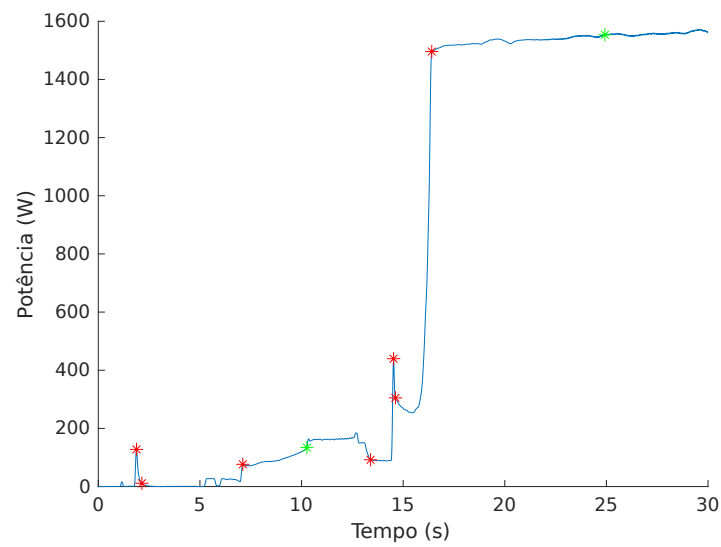


Figura 50 – Padrão de consumo do Cenário 1.

Fonte: Autoria própria.

- Cenário 2:
 - Ar-condicionado (no início);
 - *Air fryer* (no segundo 5).
 - Forno de micro-ondas (no segundo 10);

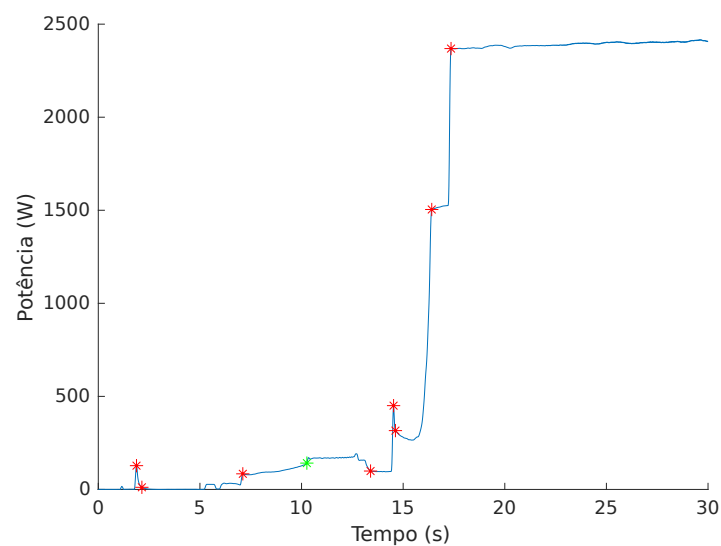


Figura 51 – Padrão de consumo do Cenário 2.

Fonte: Autoria própria.

- Cenário 3:
 - Ar-condicionado (no início);

- *Air fryer* (no segundo 5);
- Forno de micro-ondas (no segundo 10);
- Aspirador de pó (no segundo 20).
- *Freezer* (no segundo 26);

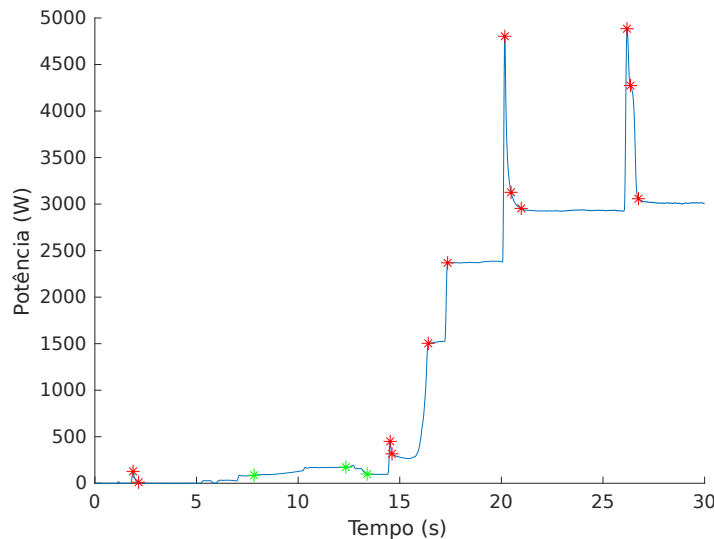


Figura 52 – Padrão de consumo do Cenário 3.

Fonte: Autoria própria.

Os resultados com os F -scores ($\beta = 1$) são mostrados na Tabela 8. Os atributos de inteligência artificial utilizados neste teste foram a diferença da potência ativa (dP), a diferença da energia reativa (dW), a diferença das harmônicas de corrente (dI) e se o evento é de aumento ou diminuição de consumo (*type*).

Cenário	KNN	MLP	SVM
Cenário 1	88,31%	88,31%	96,15%
Cenário 2	91,53%	91,53%	100%
Cenário 3	93,14%	93,14%	87,00%

Tabela 8 – F -scores para vários cenários.

Os resultados para os atributos diferença de potência ativa (dP), diferença de energia reativa (dW), a diferença das harmônicas da potência residual (dV) e se o evento é de aumento ou diminuição de consumo (*type*) são mostrados na Tabela 9.

No Cenário 3, o número de eventos de longa duração foi maior que a quantidade esperada, levando em consideração os eventos individuais.

Foi efetuada a análise PCA dos atributos usados no algoritmo de inteligência artificial para cada um dos cenários para ser conhecida a variabilidade dos atributos escolhidos, que são listados nas Tabelas 10, 11 e 12:

Cenário	KNN	MLP	SVM
Cenário 1	88.31%	86,31%	100%
Cenário 2	91.53%	100%	100%
Cenário 3	93.14%	86.48%	57,67%

Tabela 9 – *F-scores* para vários cenários.

Atributo	Porcentagem	Atributo	Porcentagem
dP	41,59%	dP	41,56%
dW	33,54%	dW	28,74%
dV	18,16%	dI	22,58%
<i>type</i>	6,72%	<i>type</i>	7,11%

Tabela 10 – Análise PCA para o Cenário 1.

Atributo	Porcentagem	Atributo	Porcentagem
dP	54,02%	dP	54,06%
dW	26,59%	dW	25,74%
dV	13,25%	dI	14,24%
<i>type</i>	6,04%	<i>type</i>	5,96%

Tabela 11 – Análise PCA para o Cenário 2.

Atributo	Porcentagem	Atributo	Porcentagem
dP	49,93%	dP	50,45%
dW	30,35%	dW	28,06%
dV	15,25%	dI	16,99%

Tabela 12 – Análise PCA para o Cenário 3.

A diferença entre as tabelas à esquerda e à direita é em relação ao atributo dV e dI. As tabelas à esquerda contêm o atributo dV e as tabelas à direita contém o atributo dI. As porcentagens indicam a variabilidade que cada atributo tem no momento da classificação. No Cenário 3, o atributo *type* foi suprimido porque a variabilidade sem este foi maior que 95% sem que ele esteja presente, indicando a sua menor importância quando mais cargas estão presentes.

4.2.1 Resultados de Ar-condicionado com múltiplos *setpoints*

O ar-condicionado com inversor foi testado com múltiplos *setpoints* sendo trocados em intervalos de segundos. Para este caso, a taxa de amostragem foi reduzida para o padrão de aquisição de 1024 amostras por ciclo para 64 amostras por ciclos, devido à quantidade de dados serem analisados. Devido à falta de dados com múltiplos *setpoints*, os dados usados para treinamento e teste são os mesmos, mas recortados e rearranjados de tal forma a não acontecerem ao mesmo tempo. Foi adicionada mais uma carga, o forno de micro-ondas, para fins de teste de desagregação. Os atributos usados foram a potência ativa (dP), energia reativa (dW), as harmônicas de corrente (dI) e o tipo de rampa (*type*).

A Tabela 13 contém os resultados. A variabilidade extraída pela análise PCA está na Tabela 14.

	KNN	MLP	SVM
Cenário 6	92,83%	92,83%	84,38%

Tabela 13 – *F-scores* para múltiplos *setpoints*.

Atributo	Porcentagem
dP	37,46%
dW	26,95%
dV	22,58%
<i>type</i>	13,01%

Tabela 14 – Análise PCA para o Cenário 6.

A Figura 53 contém o gráfico de potência ativa. Os asteriscos vermelhos são os eventos de curta duração e os verdes, de longa duração. O ar-condicionado inicia no momento zero e o forno de micro-ondas inicia no segundo 10.

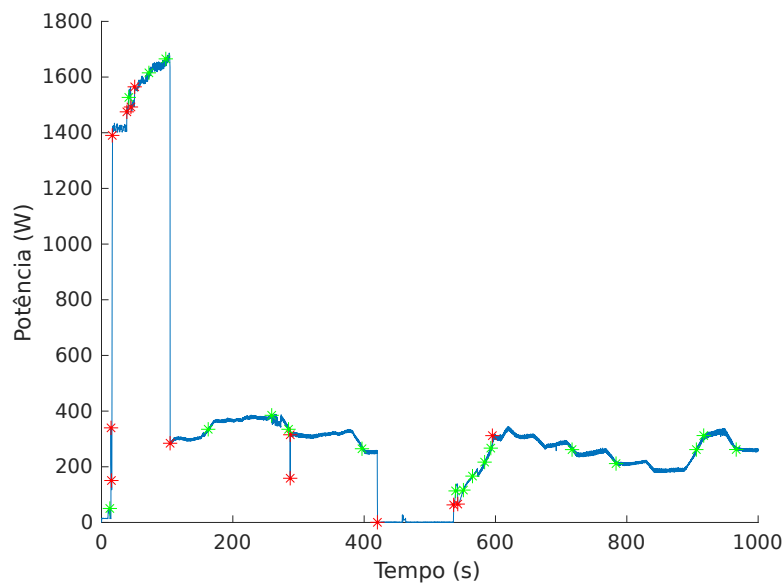


Figura 53 – Padrão de consumo do Cenário 6.

Fonte: Autoria própria.

A Figura 54 mostra as fronteiras de decisão para este caso, especificamente.

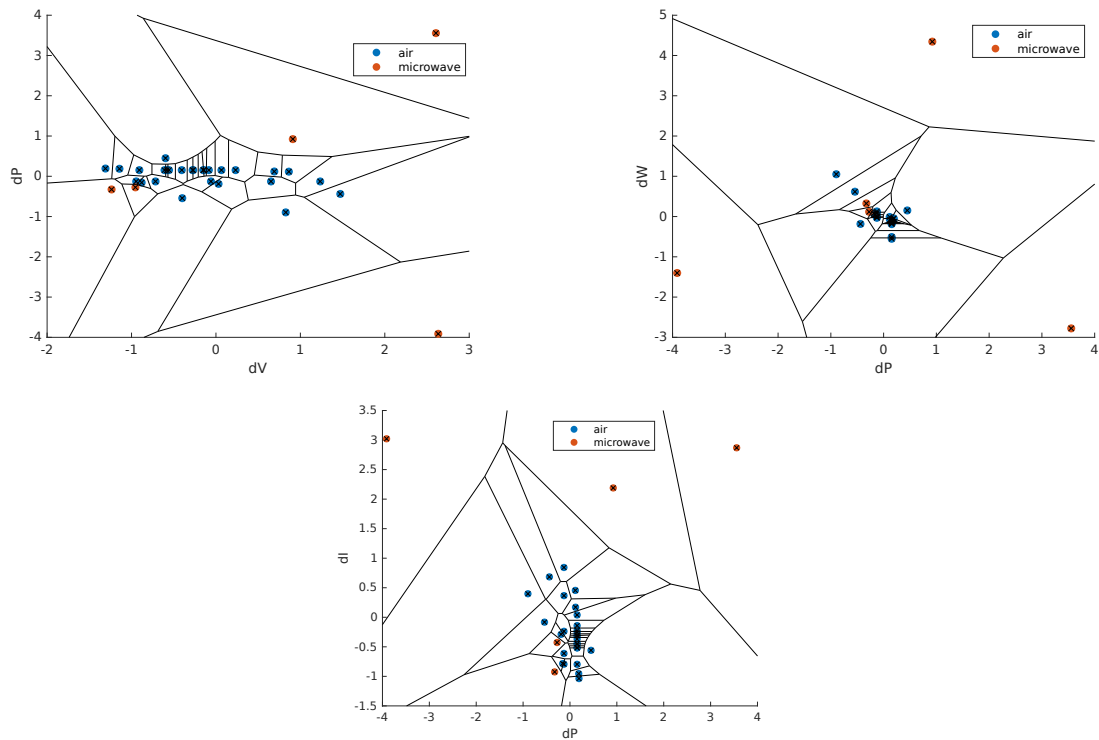


Figura 54 – Diagrama de Voronoi da carga com múltiplos *setpoints*.

Fonte: Autoria própria.

4.2.2 Resultados de Ar-condicionado com Inversor e Cargas Sintéticas

Foram adicionadas as cargas com inversor e algumas cargas sintéticas para testar a capacidade do algoritmo de desagregar cargas com eventos de longa duração simultâneas. Para isso, foram criados dois tipos de carga sintética: uma com uma rampa ascendente e outra com uma rampa ascendente e descendente. Para a detecção de duas rampas ascendentes, o algoritmo necessita calcular o aumento da velocidade da variação de potência dos eventos gerados e verificar se esse aumento existe nos dados agregados. E para o caso de desagregar cargas com rampas ascendentes e descendentes simultâneas, o algoritmo precisa calcular a queda ou a inversão da variação para a correta detecção das cargas agregadas.

A Figura 55 mostra o teste do ar-condicionado com a carga sintética ascendente e o seu resultado agregado. Os pontos verdes significam o início e os vermelhos o fim do evento. Os pontos marrons são os pontos em que houve um fim de um evento e o início de outro evento. É importante notar que nem sempre os números de eventos detectados nas cargas agregadas coincidem com os eventos nas cargas individuais, por isso a necessidade de calcular a velocidade da variação para utilizá-la; assim no caso do surgimento de eventos de longa duração extras, esses serão desagregados corretamente.

A Figura 56 mostra o teste do ar-condicionado com a carga sintética ascendente

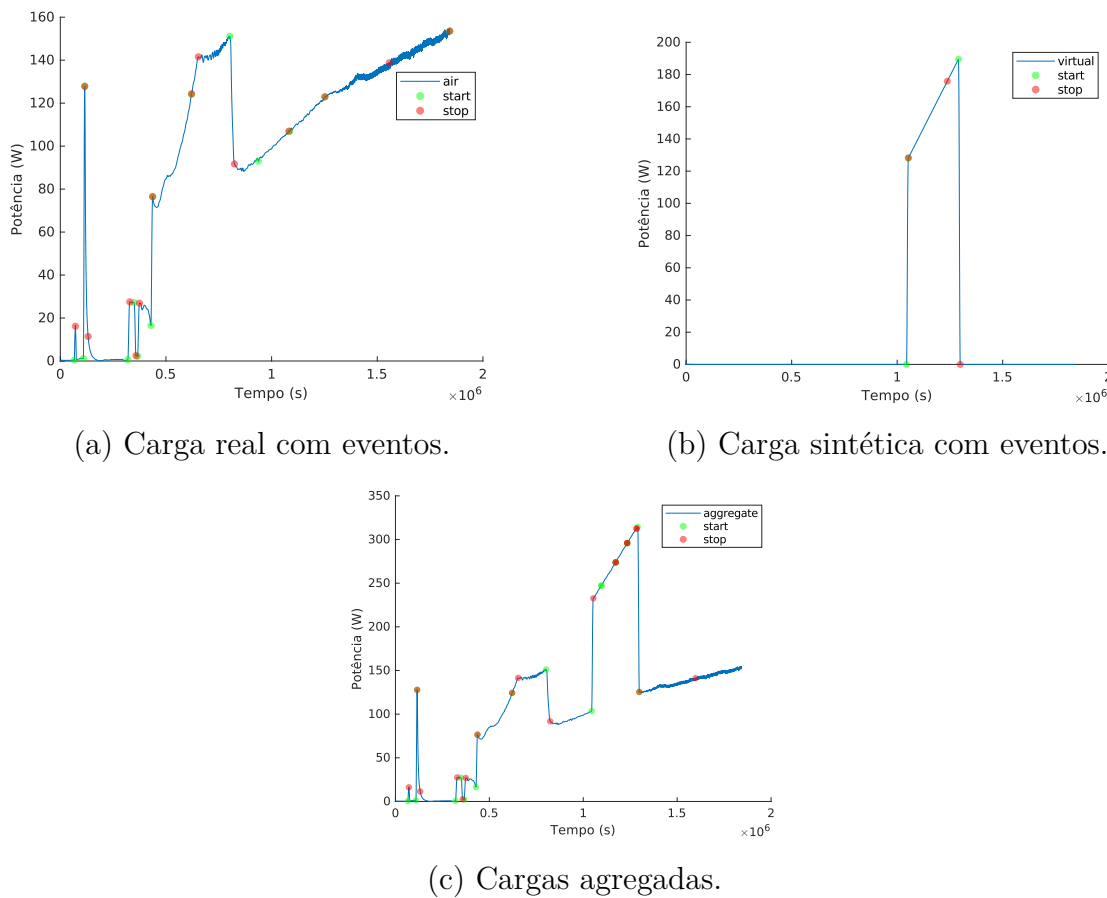


Figura 55 – Carga real e sintética ascendente para desagregação.

Fonte: Autoria própria.

e descendente e o seu resultado agregado. Aqui também é possível notar o aumento da quantidade de eventos detectados.

Para os dois cenários mostrados, a taxa de acerto foi de 100%.

Foi efetuada a análise PCA dos atributos usados nos dois cenários, nas Tabelas 15 e 16.

Atributo	Porcentagem	Atributo	Porcentagem
dP	67,69%	dP	69,92%
dW	19,75%	dW	19,02%
dV	8,42%	dI	7,97%

Tabela 15 – Análise PCA para uma rampa (Cenário 4).

Devido à diferença na quantidade de eventos de longa duração gerados, a quantidade total de eventos desagregados foi maior que os eventos individuais das cargas. Para esses casos, os eventos de longa duração de mesmo tipo que ocorreram sequencialmente foram combinados para o cálculo do *F-score*.

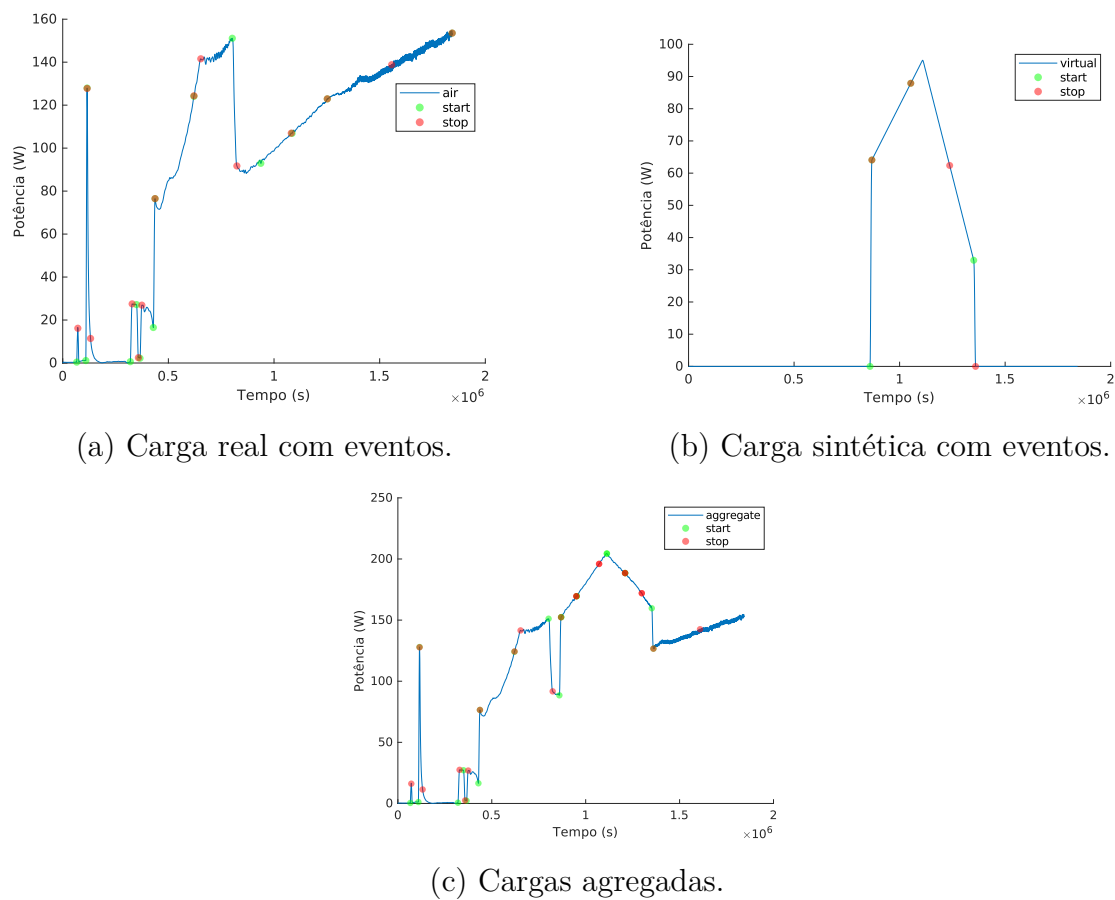


Figura 56 – Carga real e sintética ascendente para desagregação.

Fonte: Autoria própria.

Atributo	Porcentagem	Atributo	Porcentagem
dP	68,91%	dP	69,23%
dW	20,70%	dW	20,38%
dV	8,90%	dI	8,87%

Tabela 16 – Análise PCA para duas rampas (Cenário 5).

4.3 Resultados de tempo de execução em C++

Nesta seção são apresentados os resultados de tempo de execução do treinamento, do cálculo do limiar mínimo e do tempo de treinamento dos algoritmos de Inteligência Artificial. O tempo de execução foi medido usando duas taxas de amostragens distintas: 1024 amostras por ciclo (61440Hz) e 128 amostras por ciclo (7680Hz).

4.3.1 Resultados para 1024 amostras por ciclo

Nas tabelas 17, 18 e 19 temos o tempo de execução, em segundos, de 30 segundos de dados dos vários algoritmos de Inteligência Artificial juntamente com os cenários para as três plataformas. Neste teste, é executado o detector de eventos e o algoritmo de

inteligência artificial classificador mostrados nas tabelas para eventos de curta duração, e também a extração, montagem, treinamento e posterior desagregação dos eventos de longa duração. A partir dos tempos das tabelas apresentadas, é possível concluir que a Plataforma 2 é inviável para ser utilizada com essa taxa de amostragem, já que a velocidade de execução da desagregação é maior que os 30 segundos de dados. As Plataformas 3 e 1 têm desempenho suficiente para a tarefa.

Cenário	Plataforma 1		
	KNN	MLP	SVM
1	$0,40 \pm 0,02s$	$0,37 \pm 0,003s$	$0,36 \pm 0,002s$
2	$0,39 \pm 0,005s$	$0,40 \pm 0,003s$	$0,38 \pm 0,005s$
3	$0,32 \pm 0,003s$	$0,32 \pm 0,007s$	$0,32 \pm 0,003s$

Tabela 17 – Tempo de desagregação da Plataforma 1.

Cenário	Plataforma 2		
	KNN	MLP	SVM
1	$39,8 \pm 0,2s$	$40,0 \pm 0,2s$	$39,8 \pm 0,1s$
2	$41,4 \pm 0,1s$	$41,6 \pm 0,2s$	$41,2 \pm 0,1$
3	$33,1 \pm 0,1s$	$33,1 \pm 0,2s$	$33,2 \pm 0,2$

Tabela 18 – Tempo de desagregação da Plataforma 2.

Cenário	Plataforma 3		
	KNN	MLP	SVM
1	$12,5 \pm 0,1s$	$13,2 \pm 0,1s$	$12,6 \pm 0,1s$
2	$13,3 \pm 0,1s$	$13,4 \pm 0,2s$	$13,0 \pm 0,2s$
3	$10,7 \pm 0,1s$	$11,3 \pm 0,3s$	$10,9 \pm 0,2s$

Tabela 19 – Tempo de desagregação da Plataforma 3.

As tabelas 20, 21 e 22 apresentam os tempos totais que cada uma das plataformas levam para extrair os dados usados na desagregação, incluindo o calculo do limiar mínimo ΔP , a extração de parcelas da CPT, as harmônicas de corrente, o algoritmo de PLL e o treinamento dos algoritmos de classificação.

Cenário	Plataforma 1		
	KNN	MLP	SVM
1	$1,86 \pm 0,04s$	$2,18 \pm 0,01s$	$1,72 \pm 0,01s$
2	$2,79 \pm 0,05s$	$3,33 \pm 0,03s$	$2,94 \pm 0,01s$
3	$4,94 \pm 0,06s$	$5,26 \pm 0,05s$	$4,89 \pm 0,01s$

Tabela 20 – Tempo total de calculo dos atributos para a Plataforma 1.

As Tabelas 23, 24 e 25 mostram somente o tempo de treinamento, em segundos, dos algoritmos de inteligência artificial para a desagregação de eventos de curta duração para cada um dos cenários. Os tempos de treinamento utilizaram os núcleos da CPU

Cenário	Plataforma 2		
	KNN	MLP	SVM
1	188±0,8s	212±0,4s	187±0,3s
2	301±0,5s	324±0,5s	301±0,6
3	538±0,7s	559±0,7s	538±0,7

Tabela 21 – Tempo total de calculo dos atributos para a Plataforma 2.

Cenário	Plataforma 3		
	KNN	MLP	SVM
1	56,5±0,02s	60.9±0,82s	57,8±0,1s
2	93,3±0,03s	94.5±0,11s	91.8±0,1s
3	165±0,1s	167±0,11s	166±0,1s

Tabela 22 – Tempo total de calculo dos atributos para a Plataforma 3.

disponíveis para o caso do algoritmo MLP, e para os algoritmos de KNN e SVM, somente um núcleo é utilizado.

Cenário	Plataforma 1		
	KNN	MLP	SVM
2	2,76±1,0 μ s	0,44±0,012s	36,9±3,7 μ s
3	5,13±3,1 μ s	0,45±0,029s	59,7±5,1 μ s
5	4,48±2,2 μ s	0,42±0,012s	113±4,4 μ s

Tabela 23 – Tempo de treinamento da Plataforma 1.

Cenário	Plataforma 2		
	KNN	MLP	SVM
2	39,7±8,4 μ s	23,4±0,2s	281,7±26 μ s
3	37,8±6,6 μ s	22,5±0,2s	524,3±23 μ s
5	35,1±5,2 μ s	21,7±0,1s	1225±26 μ s

Tabela 24 – Tempo de treinamento da Plataforma 2.

Cenário	Plataforma 3		
	KNN	MLP	SVM
2	9,77±3,6 μ s	2,31±0,02s	98,8±5,6 μ s
3	11,7±3,4 μ s	2,44±0,02s	152±5,6 μ s
5	9,49±2,3 μ s	2,31±0,02s	285±5,1 μ s

Tabela 25 – Tempo de treinamento da Plataforma 3.

A tabela 26 apresenta os tempos de execução do calculo do limiar mínimo.

4.3.2 Resultados para 128 amostras por ciclo

Nesta seção estão as tabelas com informações de desempenho para a implementação em C++ dos algoritmos ajustados para 128 amostras por ciclo (7680Hz). O funcionamento

Cenário	Plataforma 1	Plataforma 2	Plataforma 3
1	88,2±1.6ms	2,84±0.02s	419±6,5ms
2	123±2,3ms	4,21±0.01s	638±10ms
3	238±1,4ms	6.96±0.03s	1,13±0.02s

Tabela 26 – Tempo total de calculo do limiar mínimo.

é idêntico à implementação com 1024 amostras, mas esta requer menos processamento, mantendo a acurácia. Os resultados estão nas Tabelas 27 a 36.

Cenário	Plataforma 1		
	KNN	MLP	SVM
1	46,7±0,5ms	46,4±0,3ms	44,6±0,2ms
2	46,3±0,4ms	47,4±0,2ms	46,0±0,3ms
3	40,0±0,1ms	39,6±0,4ms	40,2±0,2ms

Tabela 27 – Tempo de desagregação da Plataforma 1.

Cenário	Plataforma 2		
	KNN	MLP	SVM
1	3.75±0.03s	3.76±0.06s	3.68±0.02s
2	3.97±0.05s	3.92±0.04s	3.90±0.03s
3	3.03±0.02s	3.08±0.06s	3.06±0.04s

Tabela 28 – Tempo de desagregação da Plataforma 2.

Cenário	Plataforma 3		
	KNN	MLP	SVM
1	0,56±0,01s	0,49±0,01s	0,6±0,02s
2	0,56±0,02s	0,54±0,02s	0,54±0,016s
3	0,47±0,06s	0,42±0,02s	0,41±0,013s

Tabela 29 – Tempo de desagregação da Plataforma 3.

Cenário	Plataforma 1		
	KNN	MLP	SVM
1	0,23±0,01s	0,70±0,03s	0,22±0,00s
2	0,34±0,01s	0,82±0,02s	0,34±0,00s
3	0,62±0,00s	1,02±0,01s	0,62±0,00s

Tabela 30 – Tempo total de calculo dos atributos para a Plataforma 1.

Cenário	Plataforma 2		
	KNN	MLP	SVM
1	18.5±0.1s	41.8±0.5s	18.3±0.1s
2	29.8±0.7s	52.0±0.2s	29.5±0.2s
3	53.0±0.2s	74.4±0.4s	52.9±0.4s

Tabela 31 – Tempo total de calculo dos atributos para a Plataforma 2.

Cenário	Plataforma 3		
	KNN	MLP	SVM
1	$2,98 \pm 0,03s$	$5,09 \pm 0,03s$	$3,35 \pm 0,04s$
2	$4,35 \pm 0,01s$	$6,49 \pm 0,05s$	$4,68 \pm 0,01s$
3	$9,98 \pm 0,06s$	$11,9 \pm 0,08s$	$8,19 \pm 0,04s$

Tabela 32 – Tempo total de calculo dos atributos para a Plataforma 3.

Cenário	Plataforma 1		
	KNN	MLP	SVM
2	$3,51 \pm 1,7\mu s$	$478 \pm 30ms$	$26,1 \pm 0,8\mu s$
3	$2,65 \pm 0,6\mu s$	$471 \pm 20ms$	$47,7 \pm 1,0\mu s$
5	$2,83 \pm 0,6\mu s$	$414 \pm 7ms$	$107 \pm 6,8\mu s$

Tabela 33 – Tempo de treinamento da Plataforma 1.

Cenário	Plataforma 2		
	KNN	MLP	SVM
2	$31,0 \pm 5,2\mu s$	$23,1 \pm 0,2s$	$253 \pm 21\mu s$
3	$5,88 \pm 0,54ms$	$21,3 \pm 0,1s$	$157 \pm 0,0ms$
5	$33,3 \pm 4,6ms$	$21,4 \pm 0,1s$	$1,23 \pm 0,07ms$

Tabela 34 – Tempo de treinamento da Plataforma 2.

Cenário	Plataforma 3		
	KNN	MLP	SVM
2	$11,26 \pm 4,5\mu s$	$2,33 \pm 0,02s$	$93,3 \pm 5,4\mu s$
3	$11,1 \pm 2,9\mu s$	$2,32 \pm 0,02s$	$142 \pm 3,7\mu s$
5	$10,1 \pm 2,9\mu s$	$2,23 \pm 0,08s$	$283 \pm 5,57\mu s$

Tabela 35 – Tempo de treinamento da Plataforma 3.

Cenário	Plataforma 1	Plataforma 2	Plataforma 3
1	$10,6 \pm 0,2ms$	$290 \pm 2ms$	$50,5 \pm 0,6ms$
2	$15,0 \pm 0,3ms$	$451 \pm 12ms$	$73,0 \pm 1,3ms$
3	$29,4 \pm 0,2ms$	$712 \pm 22ms$	$132 \pm 1ms$

Tabela 36 – Tempo total de calculo do limiar mínimo.

5 Conclusões e Trabalhos Futuros

Neste trabalho foi apresentada uma proposta de metodologia para a desagregação de cargas com inversor embutido. Para isso, foram criados algoritmos capazes de operar com cargas com um padrão de consumo mais complexo que em sua operação normal, existem aumentos e reduções aleatórias de consumo. Foi apresentado um novo algoritmo para a detecção de eventos, capaz de analisar e indicar o ligamento e desligamento de cargas ou unidades de geração com início repentino ou suave. O algoritmo conseguiu desagregar várias cargas de variação lenta.

Um conjunto de dados foi criado especificamente para este trabalho, com ar-condicionado com inversor de frequência embutido e outras cargas para testar o novo algoritmo. Foi apresentado o formato de dados baseado em DFT e zLib para compactar os dados de aprendizado e teste de maneira eficiente.

O algoritmo apresentado mostra que algoritmos de Inteligência Artificial mais antigos como o KNN, o MLP, e o SVM podem ser utilizados para a desagregação de cargas de padrão de consumo complexo como aparelhos de ar-condicionado com inversor desde que os padrões de consumo menos importantes sejam ignorados ou mesmo as cargas de baixo consumo como lâmpadas únicas de LED ou dispositivos em modo de espera. Cargas como uma lâmpada única de LED têm pouca relevância no quesito de controle de gastos quando comparadas a um ar-condicionado com inversor ou a um forno de micro-ondas, que podem ter padrões de consumo que vão de dezenas a centenas de vezes maiores que a lâmpada.

Uma implementação em C++ deste algoritmo foi desenvolvida e esta mostrou que ele pode ser utilizado em sistemas IoT presentes no mercado, respeitando os limites de taxa de amostragem, não sendo necessária a utilização de um hardware específico ou alguma solução de implementação mais complexa como o processamento na nuvem em que algoritmos mais modernos, mas dispendiosos em termos de consumo energético seriam usados, muitas vezes invalidando as potenciais vantagens econômicas pelo próprio consumo do sistema de desagregação.

5.1 Trabalhos futuros

Outros algoritmos de Inteligência Artificial podem ser testados, como as CNNs, RNNs e algoritmos baseados em atenção, como o Transformer. A utilização de algoritmos mais modernos pode possibilitar que o detector de eventos possa analisar mais detalhes sobre as cargas ou analisar cargas de menor consumo juntamente com cargas de alto

consumo.

Como foi mostrado na seção 3.3.3, o detector de eventos pode ter sua precisão na detecção de eventos melhorada, possibilitando uma maior precisão na desagregação de cargas.

Este algoritmo pode ser otimizado para ser utilizado em conjunto com chips aceleradores de Aprendizado de Máquina, como a plataforma nVidia Jetson ou o K210.

Pode ser criado um algoritmo para otimizar os valores de F_1 , F_2 e F_3 que durante esse trabalho foram otimizados individualmente, mas não foi pesquisada uma possível relação entre eles e não foram consideradas outras cargas que poderiam gerar outros valores ótimos.

Uma técnica de aprendizado por transferência pode ser desenvolvida para este sistema para permitir que este possa ser empregado em várias residências sem necessitar de treinamento para as cargas anteriormente treinadas.

5.2 Publicações Realizadas

- IX Workshop Integração de Saberes Ambientais: Cidades Inteligentes e Sustentáveis;
Data do Evento: 2 – 4 de Outubro de 2024
Título do artigo: Desafios na desagregação de cargas baseadas em inversores
- SEPOC 2024;
Data do Evento: 20 – 23 de Outubro de 2024
Título do artigo: Challenges of inverter-based load disaggregation
DOI: 10.1109/SEPOC63090.2024.10747446
- III SECITEM - Simpósio de Educação, Ci via Eventos S. J. da Boa Vista;
Data do Evento: 6 – 8 de Novembro de 2025
Título do artigo: Metodologia para desagregação de cargas baseadas em inversores
- SEPOC 2025;
Data do Evento: 9 – 12 de Novembro de 2025
Título do artigo: Load disaggregation method for inverter-based equipment

Referências

- 1 INTERNATIONAL Energy Outlook 2023 - U.S. Energy Information Administration (EIA). 2023. <https://www.eia.gov/outlooks/ieo/index.php>. Citado na página 21.
- 2 NET Zero by 2050 – Analysis. 2021. <https://www.iea.org/reports/net-zero-by-2050>. Citado na página 21.
- 3 GARCIA, F. D. et al. NILM-based approach for energy efficiency assessment of household appliances. *Energy Informatics*, v. 3, n. 1, p. 10, out. 2020. ISSN 2520-8942. Citado na página 21.
- 4 GOPINATH, R. et al. Energy management using non-intrusive load monitoring techniques – State-of-the-art and future research directions. *Sustainable Cities and Society*, v. 62, p. 102411, nov. 2020. ISSN 22106707. Citado na página 21.
- 5 DASH, S.; SODHI, R.; SODHI, B. A Semi-Intrusive Load Monitoring Approach for Demand Response Applications. In: *2019 8th International Conference on Power Systems (ICPS)*. Jaipur, India: IEEE, 2019. p. 1–6. Citado na página 21.
- 6 RIDI, A.; GISLER, C.; HENNEBERT, J. A Survey on Intrusive Load Monitoring for Appliance Recognition. In: *2014 22nd International Conference on Pattern Recognition*. Stockholm, Sweden: IEEE, 2014. p. 3702–3707. ISSN 1051-4651. Citado na página 21.
- 7 SOUZA, W. A. D. *Estudos de Técnicas de Análise e Tecnologias Para o Desenvolvimento de Medidores Inteligentes de Energia Residenciais*. Tese (Doutorado) — UNESP, 2016. Citado 5 vezes nas páginas 21, 22, 24, 41 e 89.
- 8 HART, G. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, v. 80, n. 12, p. 1870–1891, dez. 1992. ISSN 1558-2256. Citado 5 vezes nas páginas 22, 28, 33, 36 e 37.
- 9 AGUIAR, E. L. D. *Contributions To NILM Based On Deep Convolutional Networks*. Tese (Doutorado) — UTFPR, 2023. Citado 6 vezes nas páginas 22, 28, 29, 32, 34 e 35.
- 10 SCHIRMER, P. A.; MPORAS, I. Non-Intrusive Load Monitoring: A Review. *IEEE Transactions on Smart Grid*, v. 14, n. 1, p. 769–784, jan. 2023. ISSN 1949-3061. Citado na página 22.
- 11 WHAT Does an Inverter Do ? 2024. https://www.fujielectric.com/about/column/detail/inverter_01.html. Citado na página 23.
- 12 NASCIMENTO, C. L. et al. Inversores De Frequência E A Sua Revolução No Processo De Acionamento De Motores Elétricos. In: *ResearchGate*. Uberlândia, Brasil: ResearchGate, 2022. p. 1–6. ISSN 2596-2221. Citado na página 23.
- 13 OHYAMA, K.; KONDO, T. Energy-Saving Technologies for Inverter Air Conditioners. *IEEE Transactions on Electrical and Electronic Engineering*, v. 3, n. 2, p. 183–189, 2008. ISSN 1931-4981. Citado na página 24.

- 14 WEI, Z. et al. Non-intrusive Inverter Air Conditioner Identification Using General Knowledge. In: *2021 IEEE 5th Conference on Energy Internet and Energy System Integration (EI2)*. Taiyuan, China: IEEE, 2021. p. 3302–3306. Citado 3 vezes nas páginas 24, 30 e 31.
- 15 PAES, F. L. *Metodologia para desagregação de cargas em instalações residenciais através de medidores cognitivos de energia*. Tese (Dissertação de Mestrado) — Universidade Estadual Paulista (Unesp), nov. 2018. Citado 2 vezes nas páginas 24 e 33.
- 16 GARCIA, F. D. et al. NILM-based approach for energy efficiency assessment of household appliances. *Energy Informatics*, v. 3, n. 1, p. 10, out. 2020. ISSN 2520-8942. Citado 5 vezes nas páginas 27, 28, 29, 32 e 37.
- 17 ZEIFMAN, M.; ROTH, K. Nonintrusive appliance load monitoring: Review and outlook. *IEEE Transactions on Consumer Electronics*, v. 57, n. 1, p. 76–84, fev. 2011. ISSN 1558-4127. Citado 2 vezes nas páginas 27 e 32.
- 18 SOUZA, W. A. et al. A NILM Dataset for Cognitive Meters Based on Conservative Power Theory and Pattern Recognition Techniques. *Journal of Control, Automation and Electrical Systems*, v. 29, n. 6, p. 742–755, dez. 2018. ISSN 2195-3899. Citado 2 vezes nas páginas 27 e 33.
- 19 GILLIS, J. M.; ALSHAREEF, S. M.; MORSI, W. G. Nonintrusive Load Monitoring Using Wavelet Design and Machine Learning. *IEEE Transactions on Smart Grid*, v. 7, n. 1, p. 320–328, jan. 2016. ISSN 1949-3061. Citado 2 vezes nas páginas 27 e 33.
- 20 ROTTONDI, C. et al. An optimisation-based energy disaggregation algorithm for low frequency smart meter data. *Energy Informatics*, v. 2, n. 1, p. 13, set. 2019. ISSN 2520-8942. Citado na página 28.
- 21 HUCHTKOETTER, J.; REINHARDT, A. A study on the impact of data sampling rates on load signature event detection. *Energy Informatics*, v. 2, n. 1, p. 24, set. 2019. ISSN 2520-8942. Citado na página 28.
- 22 BOUHOURLAS, A. S. et al. Load Signature Formulation for Non-Intrusive Load Monitoring Based on Current Measurements. *Energies*, Multidisciplinary Digital Publishing Institute, v. 10, n. 4, p. 538, abr. 2017. ISSN 1996-1073. Citado na página 28.
- 23 HAQ, A. U.; JACOBSEN, H.-A. Prospects of Appliance-Level Load Monitoring in Off-the-Shelf Energy Monitors: A Technical Review. *Energies*, Multidisciplinary Digital Publishing Institute, v. 11, n. 1, p. 189, jan. 2018. ISSN 1996-1073. Citado na página 28.
- 24 ALCALÁ, J. et al. Event-Based Energy Disaggregation Algorithm for Activity Monitoring From a Single-Point Sensor. *IEEE Transactions on Instrumentation and Measurement*, v. 66, n. 10, p. 2615–2626, out. 2017. ISSN 1557-9662. Citado na página 28.
- 25 ANDERSON, K. D. et al. Event detection for Non Intrusive load monitoring. In: *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*. Montreal, QC, Canada: IEEE, 2012. p. 3312–3317. ISSN 1553-572X. Citado na página 28.
- 26 SOUZA, W. A. D. et al. Load Disaggregation Using Microscopic Power Features and Pattern Recognition. *Energies*, Multidisciplinary Digital Publishing Institute, v. 12, n. 14, p. 2641, jan. 2019. ISSN 1996-1073. Citado 3 vezes nas páginas 29, 33 e 37.

- 27 YI, S. et al. A New Event-detection Method Based on Composite Windows in NILM for Industrial Settings. In: *2019 IEEE Sustainable Power and Energy Conference (iSPEC)*. Beijing, China: IEEE, 2019. p. 2768–2771. Citado na página 29.
- 28 LUO, D. et al. Monitoring HVAC equipment electrical loads from a centralized location - Methods and field test results. *ASHRAE Transactions*, v. 108, p. 841–857, jan. 2002. Citado na página 29.
- 29 BERGES, M. et al. User-Centered Nonintrusive Electricity Load Monitoring for Residential Buildings. *User-Centered Nonintrusive Electricity Load Monitoring for Residential Buildings*, nov. 2011. Citado na página 29.
- 30 YANG, C.; SOH, C.; YAP, V. Comparative Study of Event Detection Methods for Non-intrusive Appliance Load Monitoring. *Energy Procedia*, v. 61, p. 1840–1843, dez. 2014. Citado na página 29.
- 31 LEEB, S.; SHAW, S.; KIRTLEY, J. Transient event detection in spectral envelope estimates for nonintrusive load monitoring. *IEEE Transactions on Power Delivery*, v. 10, n. 3, p. 1200–1210, jul. 1995. ISSN 1937-4208. Citado na página 29.
- 32 SHAW, S. R. et al. Nonintrusive Load Monitoring and Diagnostics in Power Systems. *IEEE Transactions on Instrumentation and Measurement*, v. 57, n. 7, p. 1445–1454, jul. 2008. ISSN 1557-9662. Citado na página 29.
- 33 BASU, K. et al. Time series distance-based methods for non-intrusive load monitoring in residential buildings. *Energy and Buildings*, v. 96, p. 109–117, jun. 2015. ISSN 0378-7788. Citado na página 29.
- 34 LIU, B.; LUAN, W.; YU, Y. Dynamic time warping based non-intrusive load transient identification. *Applied Energy*, v. 195, p. 634–645, jun. 2017. ISSN 0306-2619. Citado na página 29.
- 35 LIU, H. *Non-Intrusive Load Monitoring: Theory, Technologies and Applications*. Singapore: Springer, 2020. ISBN 978-981-15-1860-7. Citado na página 29.
- 36 BERLIN, E.; LAERHOVEN, K. V. An on-line piecewise linear approximation technique for wireless sensor networks. In: *IEEE Local Computer Network Conference*. Denver, CO, USA: IEEE, 2010. p. 905–912. ISSN 0742-1303. Citado na página 30.
- 37 HUA, D. et al. Simultaneous disaggregation of multiple appliances based on non-intrusive load monitoring. *Electric Power Systems Research*, v. 193, p. 106887, abr. 2021. ISSN 03787796. Citado 2 vezes nas páginas 31 e 39.
- 38 DRENKER, S.; KADER, A. Nonintrusive monitoring of electric loads. *IEEE Computer Applications in Power*, v. 12, n. 4, p. 47–51, out. 1999. ISSN 1558-4151. Citado na página 33.
- 39 COLE, A.; ALBICKI, A. Data extraction for effective non-intrusive identification of residential power loads. In: *IMTC/98 Conference Proceedings. IEEE Instrumentation and Measurement Technology Conference. Where Instrumentation Is Going (Cat. No.98CH36222)*. St. Paul, MN, USA: IEEE, 1998. v. 2, p. 812–815 vol.2. ISSN 1091-5281. Citado na página 33.

- 40 NORFORD, L. K.; LEEB, S. B. Non-intrusive electrical load monitoring in commercial buildings based on steady-state and transient load-detection algorithms. *Energy and Buildings*, v. 24, n. 1, p. 51–64, jan. 1996. ISSN 0378-7788. Citado na página 33.
- 41 BIANSOONGNERN, S.; PLUNGKLANG, B. Non-Intrusive Appliances Load Monitoring (NILM) for Energy Conservation in Household with Low Sampling Rate. *Procedia Computer Science*, v. 86, p. 172–175, jan. 2016. ISSN 1877-0509. Citado na página 33.
- 42 XIAO, P.; CHENG, S. *Neural Network for NILM Based on Operational State Change Classification*. Shanghai, China: arXiv, 2019. Comment: 5 pages, 4 figures. Citado na página 33.
- 43 POWERS, J.; MARGOSSIAN, B.; SMITH, B. Using a rule-based algorithm to disaggregate end-use load profiles from premise-level data. *IEEE Computer Applications in Power*, v. 4, n. 2, p. 42–47, abr. 1991. ISSN 1558-4151. Citado na página 33.
- 44 FARINACCIO, L.; ZMEUREANU, R. Using a pattern recognition approach to disaggregate the total electricity consumption in a house into the major end-uses. *Energy and Buildings*, v. 30, n. 3, p. 245–259, ago. 1999. ISSN 0378-7788. Citado na página 33.
- 45 MARCEAU, M. L.; ZMEUREANU, R. *Nonintrusive Load Disaggregation Computer Program to Estimate the Energy Consumption of Major End Uses in Residential Buildings*. Tese (Doutorado) — Concordia University, maio 1999. Citado na página 33.
- 46 BARANSKI, M.; VOSS, J. Genetic algorithm for pattern detection in NIALM systems. In: *Genetic Algorithm for Pattern Detection in NIALM Systems*. The Hague, Netherlands: IEEE, 2004. Citado na página 33.
- 47 RUZZELLI, A. G. et al. Real-Time Recognition and Profiling of Appliances through a Single Electricity Sensor. In: *Real-Time Recognition and Profiling of Appliances through a Single Electricity Sensor*. Boston, MA, USA: IEEE, 2010. p. 21–25. Citado na página 33.
- 48 KELLY, J.; KNOTTENBELT, W. Neural NILM: Deep Neural Networks Applied to Energy Disaggregation. In: *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*. New York, NY, USA: Association for Computing Machinery, 2015. (BuildSys '15), p. 55–64. ISBN 978-1-4503-3981-0. Citado na página 33.
- 49 FIGUEIREDO, M.; ALMEIDA, A. D.; RIBEIRO, B. An Experimental Study on Electrical Signature Identification of Non-Intrusive Load Monitoring (NILM) Systems. In: . Coimbra, Portugal: ResearchGate, 2011. p. 31–40. ISBN 978-3-642-20266-7. Citado na página 33.
- 50 KIM, H. et al. Unsupervised Disaggregation of Low Frequency Power Measurements. *Proceedings of the 2011 SIAM International Conference on Data Mining (SDM)*, p. 747–758, abr. 2011. Citado na página 33.
- 51 KOUTITAS, G. C.; TASSIULAS, L. Low Cost Disaggregation of Smart Meter Sensor Data. *IEEE Sensors Journal*, v. 16, n. 6, p. 1665–1673, mar. 2016. ISSN 1558-1748. Citado na página 33.

- 52 SULTANEM, F. Using appliance signatures for monitoring residential loads at meter panel level. *IEEE Transactions on Power Delivery*, v. 6, n. 4, p. 1380–1385, out. 1991. ISSN 1937-4208. Citado na página 33.
- 53 SRINIVASAN, D.; NG, W.; LIEW, A. Neural-network-based signature recognition for harmonic source identification. *IEEE Transactions on Power Delivery*, v. 21, n. 1, p. 398–405, jan. 2006. ISSN 1937-4208. Citado na página 33.
- 54 LAUGHMAN, C. et al. Power signature analysis. *IEEE Power and Energy Magazine*, v. 1, n. 2, p. 56–63, mar. 2003. ISSN 1558-4216. Citado na página 33.
- 55 BOUHOURLAS, A. S. et al. A NILM algorithm with enhanced disaggregation scheme under harmonic current vectors. *Energy and Buildings*, v. 183, p. 392–407, jan. 2019. ISSN 0378-7788. Citado na página 33.
- 56 DONG, M. et al. Non-Intrusive Signature Extraction for Major Residential Loads. *IEEE Transactions on Smart Grid*, v. 4, n. 3, p. 1421–1430, set. 2013. ISSN 1949-3061. Citado na página 33.
- 57 LIN, S. et al. A nonintrusive load identification method for residential applications based on quadratic programming. *Electric Power Systems Research*, v. 133, p. 241–248, abr. 2016. ISSN 0378-7796. Citado na página 33.
- 58 TESHOME, D. F.; HUANG, T. D.; LIAN, K.-L. Distinctive Load Feature Extraction Based on Fryze’s Time-Domain Power Theory. *IEEE Power and Energy Technology Systems Journal*, v. 3, n. 2, p. 60–70, jun. 2016. ISSN 2332-7707. Citado na página 33.
- 59 NGUYEN, M. et al. A novel feature extraction and classification algorithm based on power components using single-point monitoring for NILM. In: *2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*. Halifax, NS, Canada: IEEE, 2015. p. 37–40. ISSN 0840-7789. Citado na página 33.
- 60 HUANG, T. D.; WANG, W.-S.; LIAN, K.-L. A New Power Signature for Nonintrusive Appliance Load Monitoring. *IEEE Transactions on Smart Grid*, v. 6, n. 4, p. 1994–1995, jul. 2015. ISSN 1949-3061. Citado na página 33.
- 61 GARCIA, F. D.; Angelino de Souza, W.; MARAFÃO, F. P. Embedded NILM as Home Energy Management System: A Heterogeneous Computing Approach. *IEEE Latin America Transactions*, v. 18, n. 02, p. 360–367, fev. 2020. ISSN 1548-0992. Citado na página 33.
- 62 SOUZA, W. A. et al. Selection of features from power theories to compose NILM datasets. *Advanced Engineering Informatics*, v. 52, p. 101556, abr. 2022. ISSN 1474-0346. Citado na página 33.
- 63 CHAN, W.; SO, A.; LAI, L. Harmonics load signature recognition by wavelets transforms. In: *DRPT2000. International Conference on Electric Utility Deregulation and Restructuring and Power Technologies. Proceedings (Cat. No.00EX382)*. London, UK: IEEE, 2000. p. 666–671. Citado na página 33.
- 64 SU, Y.-C.; LIAN, K.-L.; CHANG, H.-H. Feature Selection of Non-intrusive Load Monitoring System Using STFT and Wavelet Transform. In: *2011 IEEE 8th International Conference on E-Business Engineering*. Beijing, China: IEEE, 2011. p. 293–298. Citado na página 33.

- 65 DUARTE, C. et al. Non-intrusive load monitoring based on switching voltage transients and wavelet transforms. In: *2012 Future of Instrumentation International Workshop (FIIW) Proceedings*. Gatlinburg, TN, USA: IEEE, 2012. p. 1–4. Citado na página 33.
- 66 GRAY, M.; MORSI, W. G. Application of wavelet-based classification in non-intrusive load monitoring. In: *2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*. Halifax, NS, Canada: IEEE, 2015. p. 41–45. ISSN 0840-7789. Citado na página 33.
- 67 TABATABAEI, S. M.; DICK, S.; XU, W. Toward Non-Intrusive Load Monitoring via Multi-Label Classification. *IEEE Transactions on Smart Grid*, v. 8, n. 1, p. 26–40, jan. 2017. ISSN 1949-3061. Citado na página 33.
- 68 CHANG, H.-H. Non-Intrusive Demand Monitoring and Load Identification for Energy Management Systems Based on Transient Feature Analyses. *Energies*, Multidisciplinary Digital Publishing Institute, v. 5, n. 11, p. 4569–4589, nov. 2012. ISSN 1996-1073. Citado na página 33.
- 69 CHANG, H.-H. et al. Power-Spectrum-Based Wavelet Transform for Nonintrusive Demand Monitoring and Load Identification. *IEEE Transactions on Industry Applications*, v. 50, n. 3, p. 2081–2089, maio 2014. ISSN 1939-9367. Citado na página 33.
- 70 MALLAT, S. Group Invariant Scattering. *Communications on Pure and Applied Mathematics*, v. 65, n. 10, p. 1331–1398, 2012. ISSN 1097-0312. Citado na página 33.
- 71 AGUIAR, E. L. D. et al. Scattering Transform for Classification in Non-Intrusive Load Monitoring. *Energies*, Multidisciplinary Digital Publishing Institute, v. 14, n. 20, p. 6796, jan. 2021. ISSN 1996-1073. Citado na página 33.
- 72 BRUNA, J.; MALLAT, S. Invariant Scattering Convolution Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 35, n. 8, p. 1872–1886, ago. 2013. ISSN 1939-3539. Citado na página 33.
- 73 SALEH, A.; BENYOUCEF, D.; ABDESLAM, D. O. Exploring The Practicality Of Huang-Hilbert Transform For NILM Feature Extraction. In: *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*. Delft, Netherlands: IEEE, 2020. ISBN 978-1-7281-5635-4. Citado 2 vezes nas páginas 33 e 36.
- 74 DU, L. et al. Electric Load Classification by Binary Voltage–Current Trajectory Mapping. *IEEE Transactions on Smart Grid*, v. 7, n. 1, p. 358–365, jan. 2016. ISSN 1949-3061. Citado na página 33.
- 75 HASSAN, T.; JAVED, F.; ARSHAD, N. An Empirical Investigation of V-I Trajectory Based Load Signatures for Non-Intrusive Load Monitoring. v. 5, n. 2, p. 870–878, mar. 2014. ISSN 1949-3061. Citado na página 33.
- 76 GAO, J. et al. A feasibility study of automated plug-load identification from high-frequency measurements. In: *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. Orlando, FL, USA: IEEE, 2015. p. 220–224. Citado na página 33.
- 77 BAETS, L. D. et al. Appliance classification using VI trajectories and convolutional neural networks. *Energy and Buildings*, v. 158, p. 32–36, jan. 2018. ISSN 0378-7788. Citado na página 33.

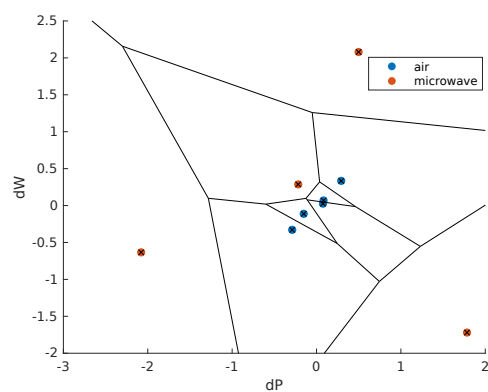
- 78 LI, L. et al. Research on Electricity Consumption Characteristic Analysis and Load Identification Method Based on Symmetrized Dot Pattern Information Fusion. In: *2024 3rd International Conference on Energy and Electrical Power Systems (ICEEPS)*. Guangzhou, China: IEEE, 2024. p. 690–694. Citado na página 33.
- 79 PAPAGEORGIOU, P. et al. A Time-Driven Deep Learning NILM Framework Based on Novel Current Harmonic Distortion Images. *Sustainability*, Multidisciplinary Digital Publishing Institute, v. 15, n. 17, p. 12957, jan. 2023. ISSN 2071-1050. Citado na página 33.
- 80 PATEL, S. et al. At the flick of a switch: Detecting and classifying unique electrical events on the residential power line. p. 271–288, set. 2007. ISSN 978-3-540-74852-6. Citado na página 33.
- 81 GUPTA, S.; REYNOLDS, M. S.; PATEL, S. N. ElectriSense: Single-point sensing using EMI for electrical event detection and classification in the home. In: *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*. New York, NY, USA: Association for Computing Machinery, 2010. (UbiComp '10), p. 139–148. ISBN 978-1-60558-843-8. Citado na página 33.
- 82 SADEGHIANPOURHAMAMI, N. et al. Comprehensive feature selection for appliance classification in NILM. *Energy and Buildings*, v. 151, p. 98–106, set. 2017. ISSN 0378-7788. Citado na página 32.
- 83 YU, L. et al. Nonintrusive appliance load monitoring for smart homes: Recent advances and future issues. *IEEE Instrumentation & Measurement Magazine*, v. 19, n. 3, p. 56–62, jun. 2016. ISSN 1941-0123. Citado na página 33.
- 84 CHERKASSKY, V.; MULIER, F. M. *Learning from Data: Concepts, Theory, and Methods / IEEE eBooks / IEEE Xplore*. [S.l.]: IEEE, 2007. ISBN 0-471-15493-8. Citado 3 vezes nas páginas 33, 34 e 35.
- 85 LAZZARETTI, A. E. et al. A Multi-Agent NILM Architecture for Event Detection and Load Classification. *Energies*, Multidisciplinary Digital Publishing Institute, v. 13, n. 17, p. 4396, jan. 2020. ISSN 1996-1073. Citado na página 33.
- 86 VAPNIK, V. N. *Statistical Learning Theory*. [S.l.]: Wiley, 1998. ISBN 978-0-471-03003-4. Citado na página 34.
- 87 BURGESS, C. J. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, v. 2, n. 2, p. 121–167, jun. 1998. ISSN 1573-756X. Citado na página 34.
- 88 HAYKIN, S. S. *Neural Networks and Learning Machines*. Hamilton, Ontario, Canada: Prentice Hall, 2009. ISBN 978-0-13-147139-9. Citado na página 34.
- 89 GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: The MIT Press, 2016. ISBN 978-0-262-03561-3. Citado na página 34.
- 90 SAHU, R.; SRIVASTAVA, S.; KUMAR, R. Modelling of a Non-Linear Dynamic System Using Long Short-Term Memory. In: *2023 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*. Greater Noida, India: IEEE, 2023. p. 1016–1021. Citado na página 35.

- 91 VASWANI, A. et al. *Attention Is All You Need*. Long Beach, CA, USA: arXiv, 2023. Comment: 15 pages, 5 figures. Citado na página 35.
- 92 BAHDANAU, D.; CHO, K.; BENGIO, Y. *Neural Machine Translation by Jointly Learning to Align and Translate*. San Diego, California, USA: arXiv, 2016. Comment: Accepted at ICLR 2015 as oral presentation. Citado na página 35.
- 93 VAVOURIS, A. et al. Benefits of three-phase metering for load disaggregation. In: *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. New York, NY, USA: Association for Computing Machinery, 2022. (BuildSys '22), p. 393–397. ISBN 978-1-4503-9890-9. Citado na página 35.
- 94 DLZRM. *Sequence-to-Subsequence Learning With Conditional GAN for Power Disaggregation*. 2023. Citado na página 35.
- 95 VAVOURIS, A. et al. Low-Frequency Non-Intrusive Load Monitoring of Electric Vehicles in Houses with Solar Generation: Generalisability and Transferability. *Energies*, Multidisciplinary Digital Publishing Institute, v. 15, n. 6, p. 2200, jan. 2022. ISSN 1996-1073. Citado na página 35.
- 96 GOWRIENANTHAN, B. et al. Deep Learning Based Non-Intrusive Load Monitoring for a Three-Phase System. *IEEE Access*, v. 11, p. 49337–49349, 2023. ISSN 2169-3536. Citado na página 35.
- 97 OORD, A. V. D. et al. *WaveNet: A Generative Model for Raw Audio*. London, UK: arXiv, 2016. Citado na página 35.
- 98 ZHANG, C. et al. Sequence-to-Point Learning With Neural Networks for Non-Intrusive Load Monitoring. *Proceedings of the AAAI Conference on Artificial Intelligence*, v. 32, n. 1, abr. 2018. ISSN 2374-3468. Citado na página 35.
- 99 LIN, S. et al. Nonintrusive Load Disaggregation Based on Attention Neural Networks. *International Transactions on Electrical Energy Systems*, v. 2025, n. 1, p. 3405849, 2025. ISSN 2050-7038. Citado na página 35.
- 100 KELLY, J.; KNOTTENBELT, W. The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. *Scientific Data*, Nature Publishing Group, v. 2, n. 1, p. 150007, mar. 2015. ISSN 2052-4463. Citado na página 35.
- 101 AZAD, M. I.; RAJABI, R.; ESTEBSARI, A. Nonintrusive Load Monitoring (NILM) Using a Deep Learning Model with a Transformer-Based Attention Mechanism and Temporal Pooling. *Electronics*, Multidisciplinary Digital Publishing Institute, v. 13, n. 2, p. 407, jan. 2024. ISSN 2079-9292. Citado na página 35.
- 102 RAFIQ, H. et al. A Deep Recurrent Neural Network for Non-Intrusive Load Monitoring Based on Multi-Feature Input Space and Post-Processing. *Energies*, Multidisciplinary Digital Publishing Institute, v. 13, n. 9, p. 2195, jan. 2020. ISSN 1996-1073. Citado na página 36.
- 103 LI, D. et al. Transfer learning for multi-objective non-intrusive load monitoring in smart building. *Applied Energy*, v. 329, p. 120223, jan. 2023. ISSN 0306-2619. Citado na página 36.

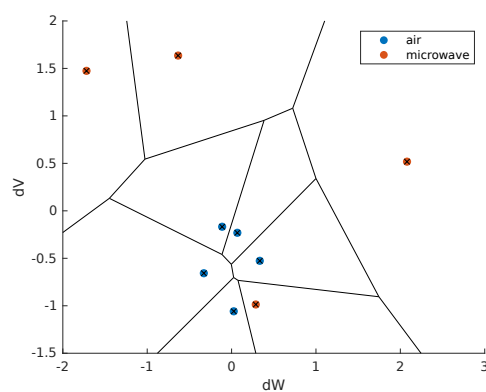
- 104 WANG, H.; SI, C.; ZHAO, J. *A Federated Learning Framework for Non-Intrusive Load Monitoring*. Shenzhen, China: arXiv, 2021. Citado na página 36.
- 105 MUAZ, M.; ZINNIKUS, I.; SHAHID, M. NILM Domain Adaptation: When Does It Work? In: *2024 10th International Conference on Smart Computing and Communication (ICSCC)*. Saarbrücken, Germany: IEEE, 2024. p. 524–528. Citado na página 36.
- 106 DING, D. et al. Load Recognition With Few-Shot Transfer Learning Based on Meta-Learning and Relational Network in Non-Intrusive Load Monitoring. *IEEE Transactions on Smart Grid*, v. 15, n. 5, p. 4861–4876, set. 2024. ISSN 1949-3061. Citado na página 36.
- 107 TENTI, P.; MATTAVELLI, P.; PAREDES, H. K. M. Conservative Power Theory, sequence components and accountability in smart grids. In: *2010 International School on Nonsinusoidal Currents and Compensation*. Łagów, Poland: IEEE, 2010. p. 37–45. ISSN 2375-1428. Citado 2 vezes nas páginas 40 e 41.
- 108 PAREDES, H. K. M. *Teoria de Potência Conservativa: Uma Nova Abordagem para o Controle Cooperativo de Condicionadores de Energia e Considerações Sobre Atribuição de Responsabilidades*. Tese (Doutorado) — UNESP, 2011. Citado na página 40.
- 109 IEC 61000-4-30:2015+AMD1:2021 CSV. 2021. <https://webstore.iec.ch/en/publication/68642>. Citado na página 43.
- 110 IEC 61000-4-7:2002. 2002. <https://webstore.iec.ch/en/publication/4226>. Citado na página 43.
- 111 IEC 61000-4-15:2010. 2010. <https://webstore.iec.ch/en/publication/4173>. Citado na página 43.
- 112 Ministério de Minas e Energia, P. e. . h. A. *Regras e Procedimentos de Distribuição (Prodist)*. 2022. <https://www.gov.br/aneel/pt-br/centrais-de-conteudos/procedimentos-regulatorios/prodist>. Citado na página 43.
- 113 ZIV, J.; LEMPEL, A. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, v. 23, n. 3, p. 337–343, maio 1977. ISSN 1557-9654. Citado na página 45.
- 114 HUFFMAN, D. A. A Method for the Construction of Minimum-Redundancy Codes. *Proceedings of the IRE*, v. 40, n. 9, p. 1098–1101, set. 1952. ISSN 2162-6634. Citado na página 45.
- 115 ZLIB Home Site. 2025. <https://zlib.net/>. Citado na página 45.
- 116 ALONSO, A. M. d. S. *Multi-Purpose Coordinated Control of Distributed Energy Resources in Transactive AC Microgrids*. Tese (Doutorado) — Universidade Estadual Paulista (Unesp), set. 2021. Citado na página 77.

APÊNDICE A – Diagramas de Voronoi

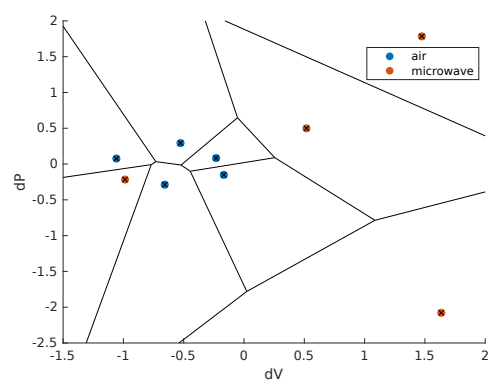
Nesta seção estão apresentados os diagramas de Voronoi para os cenários de desagregação.



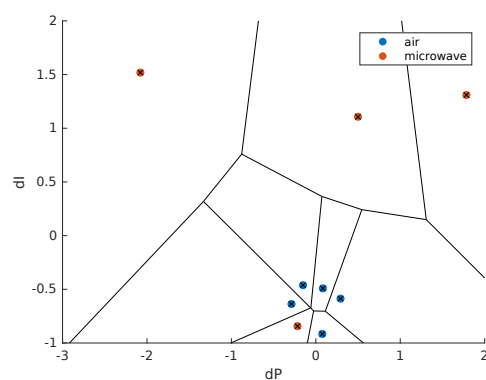
Potência ativa x energia reativa.



Energia reativa x potência nula.



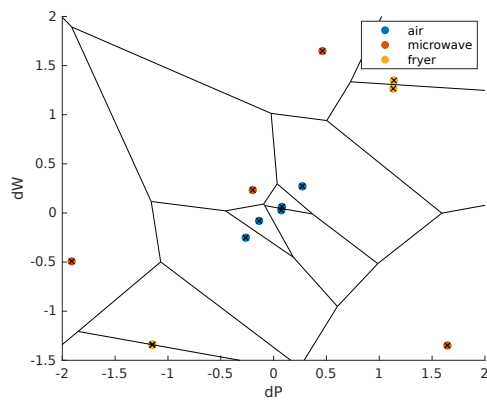
Potência nula x potência ativa.



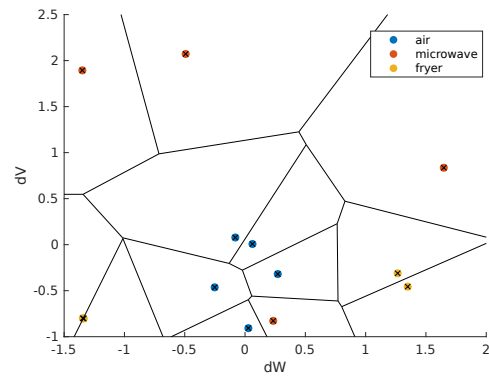
Potência ativa x harmônicas de corrente.

Figura 57 – Diagramas de voronoi do cenário 1.

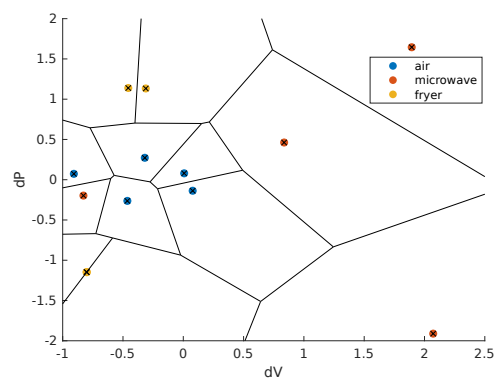
Fonte: Autoria própria.



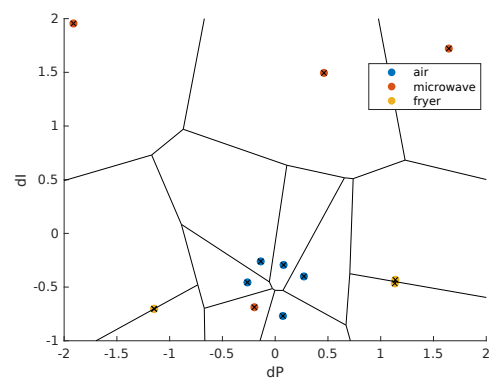
Potência ativa x energia reativa.



Energia reativa x potência nula.



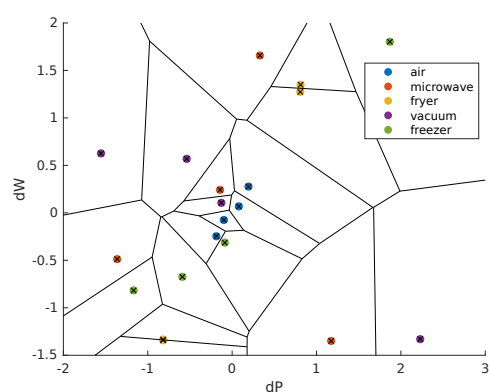
Potência nula x potência ativa.



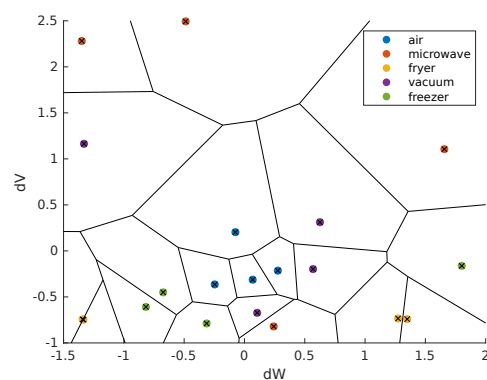
Potência ativa x harmônicas de corrente.

Figura 58 – Diagramas de voronoi do cenário 2.

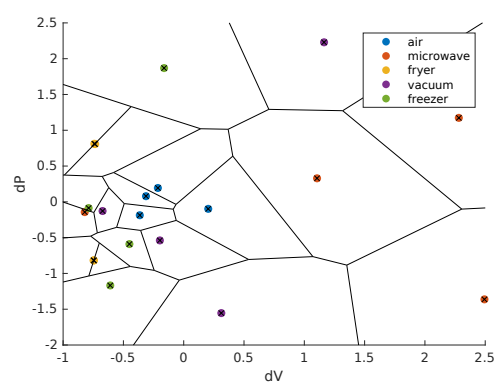
Fonte: Autoria própria.



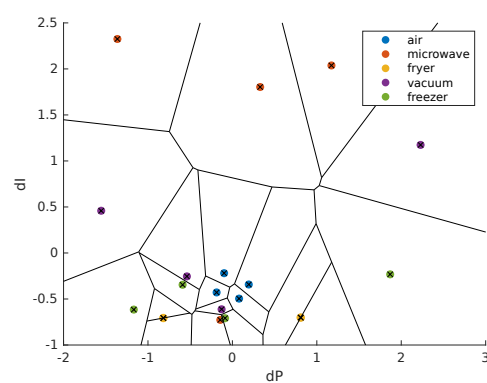
Potência ativa x energia reativa.



Energia reativa x potência nula.



Potência nula x potência ativa.



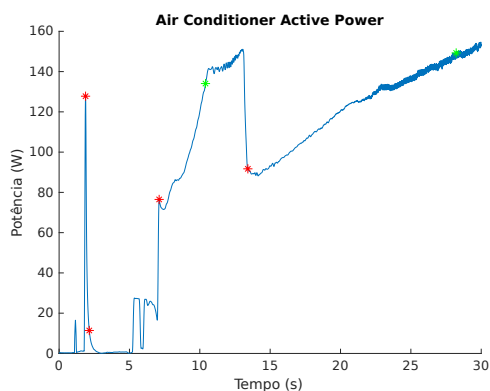
Potência ativa x harmônicas de corrente.

Figura 59 – Diagramas de voronoi do cenário 3.

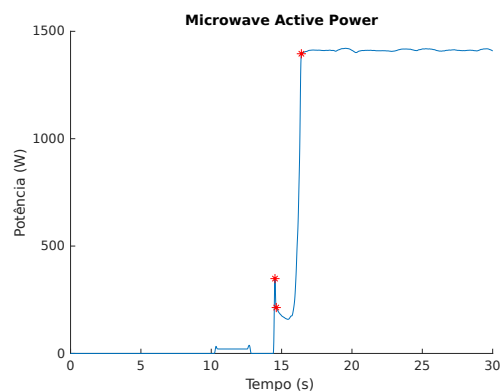
Fonte: Autoria própria.

APÊNDICE B – Gráficos dos eventos

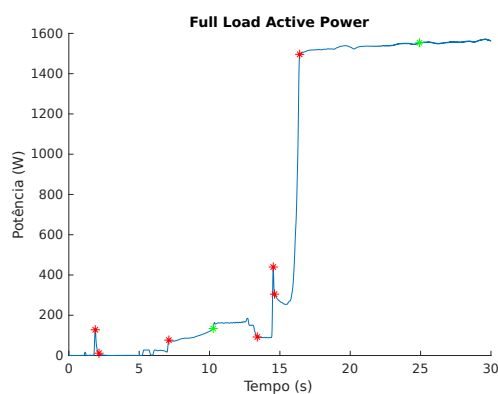
Nesta seção estão apresentados os gráficos da potência ativa com o final de eventos marcados por um asterisco. Os asteriscos vermelhos representam os eventos de curta duração e os asteriscos verdes representam os eventos de longa duração.



Potência ativa do ar-condicionado.



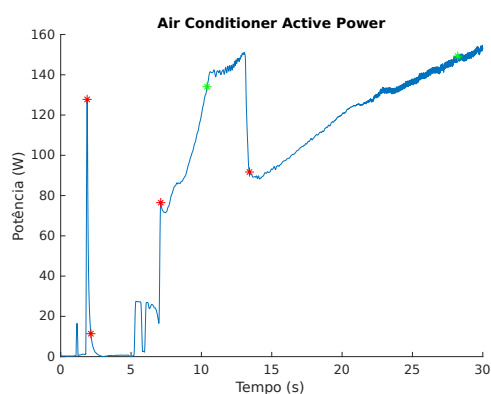
Potência ativa do forno de micro-ondas.



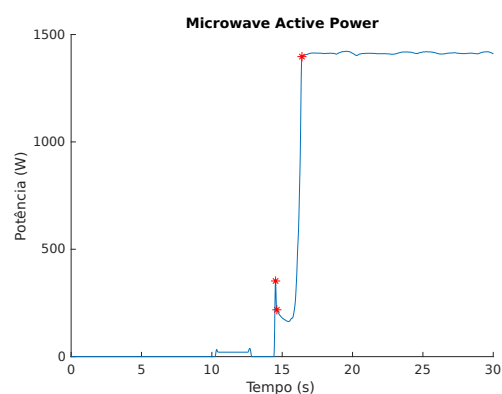
Potência ativa das cargas agregadas.

Figura 60 – Localização de eventos do cenário 1.

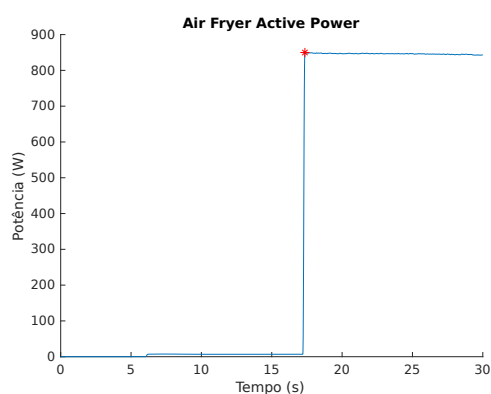
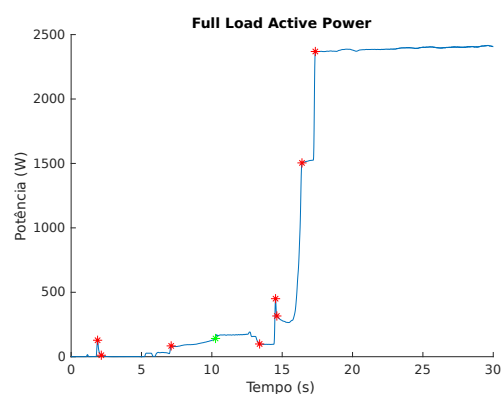
Fonte: Autoria própria.



Potência ativa do ar-condicionado.



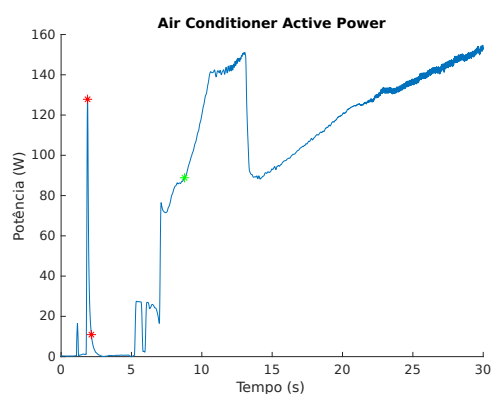
Potência ativa do forno de micro-ondas.

Potência ativa do *air fryer*.

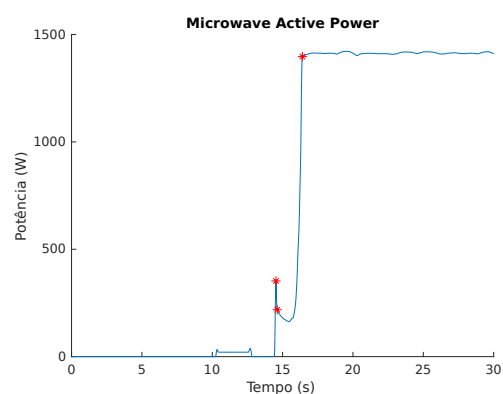
Potência ativa das cargas agregadas.

Figura 61 – Localização de eventos do cenário 2.

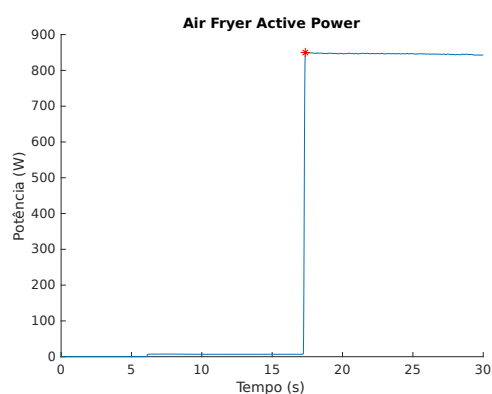
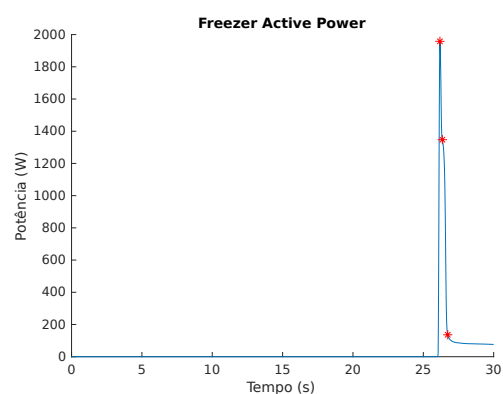
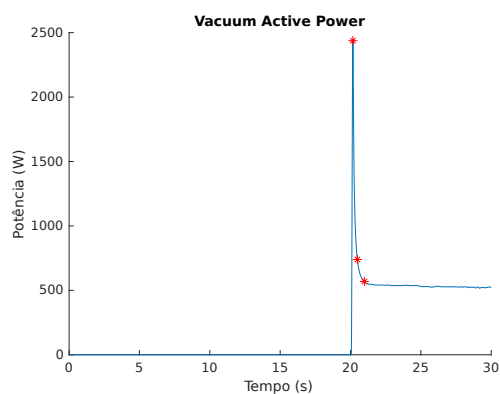
Fonte: Autoria própria.



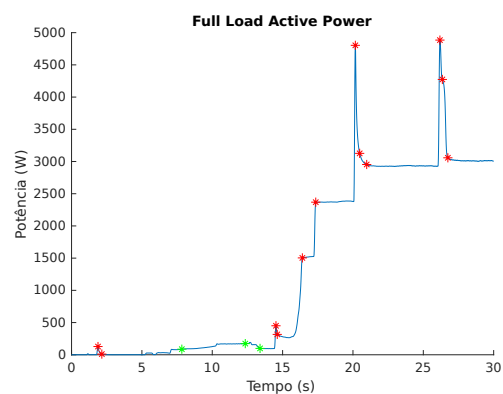
Potência ativa do ar-condicionado.



Potência ativa do forno de micro-ondas.

Potência ativa do *air fryer*.Potência ativa do *freezer*.

Potência ativa do aspirador de pó.



Potência ativa das cargas agregadas.

Figura 62 – Localização de eventos do cenário 3.

Fonte: Autoria própria.

APÊNDICE C – Descrição do formato do conjunto de dados

Os dados de forma de onda do conjunto de dados são organizados em pacotes de 4096 amostras de tensão e corrente, juntamente com o horário da gravação. Também foi aplicado um processo de calibração, conforme descrito a seguir.

Após os procedimentos de conformidade de sinal, uma DFT é aplicada a cada 1024 amostras colhidas, o que corresponde a um ciclo da rede elétrica. Em seguida, é encontrado M , o módulo máximo do resultado da DFT dividido por 2^{b-1} , sendo que b é a quantidade de *bits* de resolução do A/D, no caso, 16. Os resultados da DFT cujos valores em módulo são menores que M serão redefinidos para zero.

A justificativa por trás dessa aparente remoção de informações está no fato de que essas raia de frequências não correspondem ao sinal real, mas apenas a erros gerados pelo algoritmo de cálculo da DFT, especialmente se o formato numérico do algoritmo for ponto flutuante. Ainda assim, esse algoritmo adiciona uma zona segura para remoção, pois ao não considerar o fundo de escala da grandeza como limite de corte, valores pequenos de tensão ou corrente ficarão preservados. Ao igualar os dados que não carregam informação a zero, fica facilitada a compactação quando é aplicado o algoritmo de zlib. O algoritmo irá se deparar com vários segmentos com valores iguais a zero devido às características usuais dos sinais.

Após o processamento descrito, o pacote de dados com 4096 amostras será montado. A hora da aquisição inserida no início tem o seguinte formato: os primeiros 4 bytes representam a hora no formato *UNIX epoch*, os próximos 4 bytes representam os nanossegundos no formato *32-bit signed integer*. Depois são copiados os dados de amostras. Devido à propriedade de simetria conjugada de um sinal real transformado pela DFT, serão copiados somente os primeiros 513 valores para o pacote de dados, para cada canal e para os 4 ciclos. O formato numérico usado pelas grandezas no pacote é o IEEE 754. A Figura 63a mostra a estrutura do pacote antes da aplicação do algoritmo de zlib. Os eixos x e y representam o *offset* em bytes. A Figura 63b representa a sequência completa de como um pacote é montado. O tamanho do pacote é de 16424 bytes.

No final da montagem de um pacote, o algoritmo de zlib é aplicado e o resultado é salvo na unidade de armazenamento do medidor. Conforme mais dados são colhidos, outros pacotes são gerados e os dados são adicionados ao fim do arquivo. O medidor gera um tamanho máximo de arquivo de 100MB, caso mais dados estejam programados para serem colhidos, outro arquivo será criado.

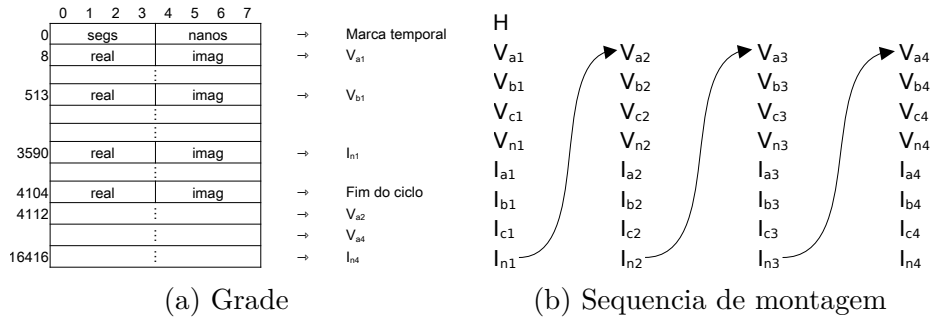


Figura 63 – Diagrama com o formato do pacote de dados.

Fonte: Autoria própria.

O nome do arquivo com os dados contém informações necessárias para serem usadas na descompactação. Este nome é gerado automaticamente pelo medidor, por exemplo: `wave-data-2024-04-25 14-44-02.387246897_60Hz_4096spc.dat`, nos quais é possível identificar os seguintes componentes:

- **wave-data:** Esta é a parte principal do nome, indicando o tipo de dado contido no arquivo, notadamente como um arquivo de dados de formas de onda;
- **2024-04-25 14-44-02.387246897:** Esta seção representa a data de quando os dados foram coletados.
 - **2024-04-25:** Indica o ano (2024), o mês (04 - abril) e o dia (25);
 - **14-44-02:** Especifica a hora (14), os minutos (44) e os segundos (02);
 - **.387246897:** Esta parte indica as frações de segundo.
- **60Hz:** Esta parte informa sobre a frequência da rede que o sinal foi medido.
- **4096spc:** Esta seção indica o tamanho do pacote de dados.

Os itens sobre a frequência da rede e tamanho de pacotes de dados são importantes no momento da descompactação, caso o arquivo seja renomeado, estas seções devem ser preservadas.

APÊNDICE D – Código-fonte do seletor de dados

```
% Function: calc_ml_params
%
% Description: This script calculates the minimum threshold, extracts the
% data used by the ML algorithms and do their data culling.
%
% Parameters:
% powers (required): Power data for each load
% names (required): Name of each load
% power_threshold (required): Power variation threshold
% cycles_threshold (required): Cycles variation threshold
% sampling_rate (required): The sampling rate
% frequency (required): Mains frequency
% percentage (required): Decline in percentage variation to consider the end of the event
% min_ia (required): Percentual culling limit of ML data compared to max
% max_past_data (required): Maximum buffer size (S)
% max_harmonics (required): Maximum harmonic to be considered
% angle_factor (required): Maximum angular coefficient quotient

function [threshold_power, loads, cut] = calc_ml_params(powers, names, power_threshold, cycles_threshold,
sampling_rate, frequency, percentage, min_ia, max_past_data, max_harmonics, angle_factor)
    cycle_length = round(sampling_rate / frequency);
    % Find the minimum threshold
    threshold_power = find_minimum_threshold(powers, power_threshold, cycles_threshold, sampling_rate, frequency,
percentage);

    % Create the ML data and the cumulative table
    X = table;
    loads = cell(1, width(names));
    for i = 1:width(names)
        loads{i} = event_detector(powers{i}, 'P', threshold_power, cycle_length, max_past_data, max_harmonics,
percentage, angle_factor);
        l = loads{i};
        l.name = repmat(string(names{i}), height(l), 1);
        X = [X; l];
    end

    % Cut off the data that is less than the threshold
    cut = X(abs(X.dP) < (max(abs(X.dP)) * min_ia), :);
    % Recalculate the minimum threshold if we had to cut off some data
    if height(cut) > 0
        % Culls the data and find the new threshold_power
        X = X(abs(X.dP) >= (max(abs(X.dP)) * min_ia), :);
        [~, idx] = sort(X.time);
        X = X(idx, :);
        threshold_power = min(abs(X.dP));
        % Recreate the ML data
        loads = cell(1, width(names));
        for i = 1:width(names)
            loads{i} = event_detector(powers{i}, 'P', threshold_power, cycle_length, max_past_data, max_harmonics,
percentage, angle_factor);
        end
    end
end
end
```

Figura 64 – Código do Seletor de Dados.

Fonte: Autoria própria.