UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E INFORMÁTICA INDUSTRIAL

EVERTON LUIZ DE AGUIAR

CONTRIBUIÇÕES PARA NILM BASEADAS EM REDES CONVOLUCIONAIS PROFUNDAS

TESE

CURITIBA
2023

EVERTON LUIZ DE AGUIAR

CONTRIBUIÇÕES PARA NILM BASEADAS EM REDES CONVOLUCIONAIS PROFUNDAS

Contributions to NILM Based on Deep Convolutional Networks

Tese apresentada como requisito para obtenção do título de Doutor em Engenharia Elétrica e Informática Industrial do Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. André Eugenio Lazzaretti
Coorientador: Prof. Dr. Daniel Rodrigues Pipa

CURITIBA

2023

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Curitiba

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

EVERTON LUIZ DE AGUIAR

CONTRIBUIÇÕES PARA NILM BASEADAS EM REDES CONVOLUCIONAIS PROFUNDAS

Trabalho de pesquisa de doutorado apresentado como requisito para obtenção do título de Doutor Em Ciências da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Engenharia De Computação.

Data de aprovação: 13 de Setembro de 2023

Dr. Andre Eugenio Lazzaretti, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Daniel Rodrigues Pipa, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Fernando Pinhabel Marafao, Doutorado - Universidade Estadual Paulista - Unesp

Dra. Lucia Valeria Ramos De Arruda, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Vitor Hugo Ferreira, Doutorado - Universidade Federal Fluminense (Uff)

Wesley Angelino De Souza, - Universidade Tecnológica Federal do Paraná

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 13/09/2023.

I dedicate this work to my family. In particular,
I dedicate it to my beloved grandparents who
passed away during the period of my doctorate.

ACKNOWLEDGEMENTS

I want to thank my family for their unconditional support throughout my journey, culminating in this thesis. To my dear mother, Loreni, my blessed father, José Valmor, and my esteemed brother Gabriel. This achievement is also yours. To my girlfriend Priscila, for always being by my side and making me a better person every minute.

I thank my advisors, Dr. Lazzaretti and Dr. Pipa. Thank you for all the encouragement you always gave me. Thank you for being patient, understanding my limitations, and helping me discover my potential. I consider you two true friends, and your working methods will always be a reference.

Professor André, thank you also for the lessons in life and coexistence you have provided me during these last four years. You are an enlightened person.

To my dear friends Bruna Mulinari and Lucas Nolasco, thank you for your immeasurable contribution to my work, all the knowledge sharing, and the friendship we built. To my long-time friend Flavio Grando, for the entire partnership, for introducing me to the research group, and for lending me his apartment. To my friend and co-worker Ricardo Bernardi, thank you for all your assistance in the final stage of my work and the great conversations during the ride.

To my friends in life, who are also my family, I leave a huge hug of gratitude. To avoid making this section full of tears, I will not name each one by name.

To God, this entity that I don't know how to define, I don't know how to explain, but that I love and fear.

Taking a new step, uttering a new word, is what
people fear most (Fyodor Dostoevsky).

RESUMO

AGUIAR, Everton Luiz de. **Contribuições para NILM baseadas em redes convolucionais profundas.**. 2023. 152 f. Tese (Doutorado em Engenharia Elétrica e Informática Industrial) – Universidade Tecnológica Federal do Paraná. Curitiba, 2023.

O monitoramento não intrusivo de cargas elétricas (NILM) consiste em determinar o perfil de consumo de energia elétrica de cada carga de uma unidade consumidora sem a necessidade de utilizar sensores independentes para cada uma dessas cargas. As aplicações de NILM ajudam o consumidor a economizar energia e auxiliam no planejamento da rede de distribuição. Desde 2015 foram propostos diversos métodos de extração de características para NILM baseados em Aprendizado Profundo, os quais alcançaram resultados do estado-da-arte para classificação e desagregação de cargas. Apesar disso, os resultados obtidos pelos métodos baseados em Deep Learning são dependentes da quantidade de dados de treinamento, e as arquiteturas das redes profundas são determinadas empiricamente. Tendo isso em vista, esse documento propõe frameworks para extrair e selecionar features utilizando a Transformada Scatering para classificar sinais NILM, tanto cargas individuais quanto cargas agregadas. Nós também introduzimos uma nova arquitetura multi-label e multi-tarefas de rede convolucional para NILM, cujos filtros não são aprendidos e sim determinados analiticamente por meio de wavelets. Os resultados de classificação do método proposto superam os métodos do estado-da-arte para vários cenários de teste, destacando-se os cenários com dataset reduzido. Nós também propomos uma abordagem expandida para NILM, na qual incluímos dados de geração distribuída (DG) fotovoltaica (PV) agregados com as cargas elétricas. Criamos um novo dataset público contendo DG-PV e dados NILM, e realizamos um estudo de ablação para avaliar tanto a influência da presença de microinversores PV na classificação das cargas agregadas quanto a influência da presença de cargas agregadas na identificação de microinversores PV na rede de distribuição.

Palavras-chave: NILM. Redes Convolucionais Profundas. Transformada Scattering. Atrás Do Medidor. Geração Distribuída.

ABSTRACT

AGUIAR, Everton Luiz de. **Contributions to NILM Based on Deep Convolutional Networks.**. 2023. 152 p. Thesis (PhD in Engenharia Elétrica e Informática Industrial) – Universidade Tecnológica Federal do Paraná. Curitiba, 2023.

Non-intrusive monitoring of electrical loads (NILM) determines the profile of electrical energy consumption of each load of a consumer unit without the need to use independent sensors for each of these loads. NILM applications help the consumer save energy and assist in planning the distribution network. Since 2015, several feature extraction methods for NILM based on Deep Learning have been proposed, which have achieved state-of-the-art results for classification and disaggregation. Despite this, the results obtained by methods based on deep learning depend on the amount of training data, and the architectures of deep networks are empirically determined. With that in mind, this document proposes frameworks to extract and select features using the Scattering Transform to classify NILM signals, both individual loads and aggregated loads. We also introduce a new multi-label and multi-task convolutional network architecture for NILM, whose filters are not learned but analytically determined through wavelets. The classification results of the proposed method surpass the state-of-the-art techniques for several test scenarios, highlighting the scenarios with a reduced dataset. We also propose an expanded approach to NILM, including distributed generation (DG) photovoltaic (PV) data aggregated with electrical loads. We create a new public dataset containing DG-PV and NILM data, and perform an ablation study to evaluate both the influence of the presence of PV microinverters on the classification of aggregated loads and the influence of the existence of aggregated loads on the identification of PV microinverters in the distribution power grid.

Keywords: NILM. Deep Convolutional Networks. Scattering Transform. Behind The Meter. Distributed Generation.

LIST OF ALGORITHMS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

INITIALISM

ADC	Analog to Digital Converter
AS	Aggregated Sampling Frequency
ASP	Acquisition and Signal Processing
AW	Acquisition Window
ASW	Acquisition Switching Window
B.	Number of buildings where the data was
BN	Batch Normalization
BTM	Behind The Meter
CPT	Conservative Power Theory
CPD	Conventional Physical Definitions
CNN	Convolutional Neural Network
CWT	Continuous Wavelet Transform
DA	Data Augmentation
DBSCAN	Density-Based Spatial Clustering
DCN	Deep Convolutional Networks
DG	Distributed Generation
DPS	Voltage Surge Protection, from Portuguese "Dispositivo de Proteção contra Surto"
DWT	Discrete Wavelet Transform
EMD	Empirical Mode Decomposition
EMI	Electromagnetic Interference
ENS	Ensemble Method
EV	Electric Vehicle
FFT	Fast Fourier Transform
FL-BNN	Federated Learning Bayesian Neural Network
FN	False Negative
FP	False Positive
GOF	Goodness Of Fit
GAN	Generative Adversarial Network
GLR	Generalized Likelihood Ratio
HHT	Hilbert-Huang Transform
HC	Hill Climbing
IDE	Inverter Detection Experiment
LDA	Linear Discriminant Analysis
LDE	Loads Detection Experiment

LSTM	Long Short-Term Memory
LIDE	Loads and Inverter Detection Experiment
MAE	Mean Absolute Error
MCB	Main Control Board
MSL	Multi-Sampled Loads
MR-TDLNN	Multi-channel Recurrent Tapped Delay Line Convolutional Neural Network
MTL	Multi-Task Learning
NTP	Network Time Protocol
PCC	Point of Common Connection
PQ	Active-Reactive Powers
PMU	Phasor Measurement Unit
PS	Power Signature
PSM	Power Switching Module
PV	Photovoltaic
RAM	Random-Access Memory
RG	Research Gap
RQ	Research Question
RTC	Real-Time Control
SFC	Sampling Frequency Class
SVM	Support Vector Machine
SB	Sensor Board
SSWT	Synchrosqueezed Wavelet Transform
ST	Scattering Transform
STFT	Short-time Fourier Transform
TFA	Time-Frequency Analysis
THDi	Total Harmonic Distortion of Current
TN	True Negative
TP	True Positive
UTFPR	Federal University of Technology, from Portuguese “Universidade Tecnológica Federal do Paraná”
V-I	Voltage-Current
WGAN	Wasserstein Generative Adversarial Network
WRG	Weighted Recurrent Graph

ACRONYMS

ANEEL	National Electric Energy Agency, from Portuguese “Agência Nacional de Energia Elétrica"
COOLL	Controlled On/Off Loads Library dataset
DeepDFML	Deep CNN-Based architecture for Detection, Feature extraction and Multi-Label classification in NILM signals
DeSpaWN	Denoising Sparse Wavelet Network
FLOP	Floating Point Operation
GFLOP	Giga Floating Point Operation
LIT	Laboratory for Innovation and Technology in Embedded Systems

NILM	Non-Intrusive Load Monitoring
PLAID	Plug Load Appliance Edentification Dataset
REDD	Reference Energy Disaggregation Data Set
t-SNE	t-Distributed Stochastic Neighbor Embedding
TDI	Aggregate Triggering With a Finite Set of Different Intervals
TFI	Aggregate Triggering With Fixed Intervals
TIL	Sequential Triggering of Individual Loads
TRI	Aggregate Triggering With a Random Set of Different Intervals
UK-DALE	United Kingdom Domestic Appliance-Level Electricity

CONTENTS

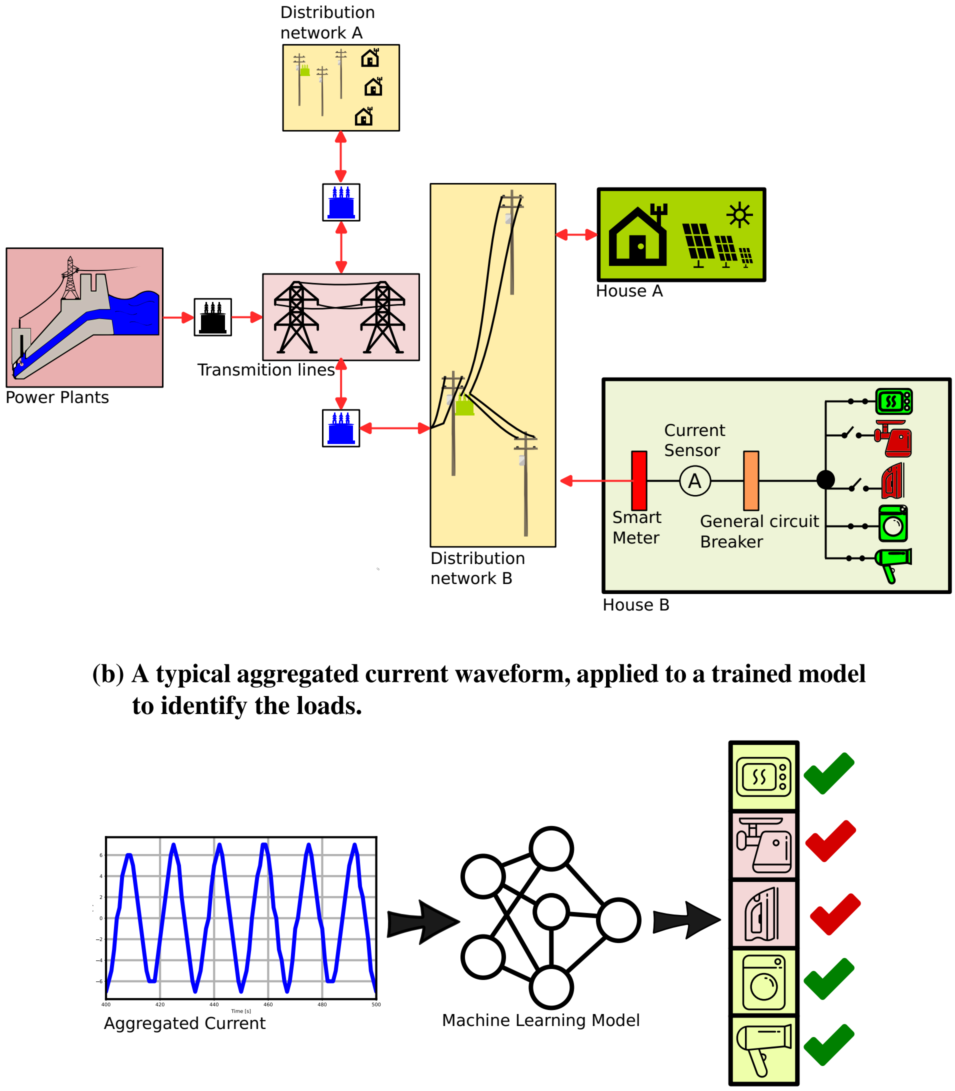
1 INTRODUCTION

Non-intrusive load monitoring (NILM), initially presented by Hart (1992), consists in extracting stratified information for each appliance in a house from the observation of electrical quantities (voltage, current, or power) aggregated to the energy delivery point. Figure 1a shows a typical connection diagram of two houses (houses A and B) in the distribution network B. Power transformers connect the distribution networks (A and B) to the transmission lines. The high-power power plants are connected to the transmission lines employing step-up power transformers. House B has a Smart-Meter, an electronic measuring device that monitors the total consumption of the residence, in addition to the value of voltage and electric current at the house's entrance. Also, let us consider in Figure 1a that, at a given instant of time, the residents of House B have the microwave oven, the washing machine, and the hair dryer turned on at the same time (representation in green), and that the electric iron and shower are turned off. NILM consists of, for example, determining which loads are connected at that instant when only the aggregate electric current is known (Current Sensor, in Figure 1). Fig 1b illustrates how this process is done: a waveform of the aggregate electric current (sum of all charges at a given instant of time) is applied to a pre-trained machine learning model to identify which individual appliances generated the analyzed aggregate signal.

Two important tasks regarding NILM are *disaggregation* and *classification*. The disaggregation task separates each load curve (current or power, typically) from an aggregated single signal (in most cases located at the switchboard). The classification task corresponds to identifying which appliances are turned on/off and when these events happened from an aggregated single electrical signal.

There are event-based and non-event-based methods applied to NILM. Events are state changes for each appliance present in the aggregate current or power signal (LU; LI, 2020). The event-based methods that implement NILM comprise four basic steps (SADEGHIAN-POURHAMAMI *et al.*, 2017): Preprocessing, Event Detection, Feature Extraction, and Load Identification. Normalization, subsampling, denoising, and division of training, testing, and validation sets are normally performed in the preprocessing step. The event detection step is responsible for determining the connection instants of each electrical load that integrates the aggregate signal, and it allows signal disaggregation. Disaggregated electrical signals enter the feature extraction step. Feature extraction methods determine the power signature of each

Figure 1 – Typical non-intrusive load monitoring approach inserted with distribution, transmission, and generation stages.



(a) House B, with an aggregated current measure, connected to the distribution and transmission networks through power transformers.

(b) A typical aggregated current waveform, applied to a trained model to identify the loads.

Source: Own authorship.

individual electrical signal. These power signatures allow a classification model to be trained in the final load identification step. Finally, the load identification step determines which loads are present in the input aggregate signal. We dedicate Chapter 2 to discuss in detail each of these steps.

One of the most relevant stages of feature extraction is defining the power signature (PS). PS is a particular representation that characterizes the behavior of each appliance based on the features. The PS depends on the chosen feature extractor. The first works that discussed power signatures considered only changes (events) that occurred after a time interval in the aggregate power curve that was in a steady state. These approaches were limited to linear loads and restricted to loads with an operating regime with varying power (electric irons, heaters, etc.). It was challenging with these early methods to determine the PS of switched or non-linear electronic circuits (LAUGHMAN *et al.*, 2003).

In addition to the early methods that have already been mentioned, one can obtain the PS through the features extracted from the electrical time signal (voltage, current, or energy). Based on the idea proposed in Sadeghianpourhamami *et al.* (2017), one can categorize the feature extractors into Conventional Physical Definitions (CPD), Time-Frequency Analysis (TFA), Voltage-Current (VI) Trajectories, or, more recently, Deep Convolutional Networks based methods.

The CPD-based feature extractors are simpler to implement but have poor discriminability and less capability of extracting transient information (HART, 1992). TFA feature extractors are more discriminant than CPD since they generally deal with high-frequency data but are time-shifting covariant, having a dependency on event detection (BURRUS *et al.*, 1998). However, they are normally dependent on event detection techniques (SADEGHIANPOURHAMAMI *et al.*, 2017). Finally, Deep Convolutional Networks (DCN) feature extractors have been getting attention in the last years because the extracted features are highly discriminative, and they are not engineered (no need for pre-knowledge about the appliances) (KELLY; KNOTTENBELT, 2015a).

One can find many architectures based on DCN in recent literature (as of 2015) to address the NILM problem. In general, these architectures are classified according to the data sampling frequency. We define sampling frequencies below 50Hz as *Low Frequencies* and above 50Hz as *High Frequencies*, as defined in Basu *et al.* (2016). Several works use low frequency datasets to train CNN architectures for disaggregation or NILM classification (CHEN *et al.*, 2018; CHEN *et al.*, 2020; MORADZADEH *et al.*, 2021; CHEN *et al.*, 2020; MASSIDDA *et al.*, 2020; KASELIMI *et al.*, 2019). One can find high frequency strategies using deep CNN in (HOUIDI *et al.*, 2020; WU; WANG, 2019; BAETS *et al.*, 2018; FAUSTINE; PEREIRA, 2020a; FAUSTINE; PEREIRA, 2020b; HIMEUR *et al.*, 2021; YANG *et al.*, 2020; MORÁN *et al.*, 2020; LAZZARETTI *et al.*, 2020; MUKAROH *et al.*, 2020). According to Shannon's theorem, the maximum representable frequency component of a continuous signal in its discrete version, for perfect reconstruction, is half the sampling frequency. This assumption helps high-frequency feature extraction methods to define more discriminative load signatures, which makes the choice of this type of method more appropriate for NILM.

Some of the main limitations with the architectures based on DCN applied to NILM are: (i) The proposed architectures generally do not integrate *detection, disaggregation, and classification*, i.e., the network performs only one or two of these tasks. The exception is the research presented in Nolasco *et al.* (2022); (ii) As they use *deep learning* techniques, these solutions demand a large amount of data for training; (iii) DCN approaches to NILM can be considered "black boxes" in some respects. The network architecture (e.g., the number of layers, size of convolutional filters, and type of regularization) are parameters determined empirically; (iv) The computational effort grows with the increase in the number of coefficients (weights) of the learned convolutional filters.

Mallat (2012) proposed the Scattering Transform (ST) and defined a time-frequency representation of a one-dimensional signal. ST has an architecture analogous to a Convolutional Neural Network (CNN). However, in the ST, the filter coefficients are not trained but are analytically determined using wavelets. In addition, the structure of the ST (number of layers and convolutional filters) can also be determined analytically. As a consequence, ST needs less training data than CNN for classification tasks (AGUIAR *et al.*, 2021b; BRUNA; MALLAT, 2012).

Typical works related to NILM restrict the analysis to residential electrical loads. Few recent works, like Jaramillo *et al.* (2020), Jaramillo *et al.* (2021b), Jaramillo *et al.* (2021c), Jaramillo *et al.* (2022), Jaramillo *et al.* (2023), include PV in the context of NILM. Parallel to this, due to the increasing penetration of photovoltaic (PV) distributed generation (DG) in homes, the Behind the Meter (BTM) estimation is a known problem in Power Systems (ERDENER *et al.*, 2022). BTM estimation consists of disaggregating from the *net load* the energy generated by a PV unit that is not seen by the electricity utility. To the best of our knowledge, no published works address BTM estimation in NILM context using DCN.

The non-observability of micro and mini DG systems by the electricity concessionaire raises some prospects for technological solutions and practical applications that could result from the union between BTM and NILM: (i) Systems embedded connected to prosumers, sharing stratified data with the energy concessionaire via an integrated management system; (ii) Prepaid electricity consumption pricing systems, based on the information from the integrated management system, given by smart meters with load disaggregation and generation in real-time.

From another point of view, future applications of NILM and BTM in industrial energy management systems for classification, disaggregation, and detection of electrical loads and DG could be very promising, given the impact of the industrial sector on energy consumption. Such applications are limited, however, by the unavailability of data or the difficulty of data acquisition in the industrial environment.

The characteristics of the ST, the promising results of the DCN applied to NILM, and the research gap related to combining the concepts of NILM with BTM estimation encouraged us to develop this thesis. This thesis proposes a series of contributions to NILM using DCN. We begin by presenting the general concepts of NILM, and in chapter 3, we propose new feature extractors for individual loads using the ST. After that, in chapter 4, we extend the approach to feature extraction and NILM classification of aggregate loads. In chapter 5, we propose a new architecture with ST to replace CNN in a multi-task application. Finally, in chapter 6, we propose a new dataset for NILM including PV data, in addition to a complete ablation study regarding merging NILM with BTM estimation.

1.1 GENERAL OBJECTIVE

The main objective of this work is to propose a new approache to feature extraction and a new network architecture applied to non-intrusive load monitoring in the presence of distributed generation.

1.2 SPECIFIC OBJECTIVES

Also, the the following specific objectives can be highlighted:

1. To propose a new framework for feature extraction (power signature) of NILM signals applying the Scattering Transform;
2. To study and evaluate theoretical aspects of Scattering Transform, verifying its advantages and possible limitations;
3. To evaluate publicly available datasets for comparison of results;
4. To propose, evaluate and compare the results of different feature selection techniques based on Scattering Transform for NILM;
5. To evaluate the separability of the classes obtained with the features extracted from Scattering Transform;

6. To disaggregate and classify NILM aggregated signals with the Scattering Transform;
7. To compare results obtained from proposed frameworks to the state-of-the-art approaches results;
8. To improve state-of-the-art classification scores;
9. To propose a novel architecture that replaces the traditional Convolutional Neural Networks by Scattering Transform to reach disaggregation and multi-label classification at once;
10. To design and implement a new experimental setup to collect data from appliances and distributed generation;
11. To construct a novel high-frequency dataset mixing electrical loads with distributed generation;
12. To evaluate the proposed dataset with state-of-the-art classification methods;
13. To propose an ablation study to evaluate the influence of distributed generation in the aggregated loads classification and the influence of the electrical loads in the distributed generation classification;
14. To elaborate scientific papers and submit them to relevant Journals.

1.3 STRUCTURE OF THIS DOCUMENT

We organize this document as follows:

- Chapter 2 presents the theoretic foundations regarding NILM. Section 2.1 discusses each step of the typical framework applied to NILM. Section 2.2 presents the Scattering Transform, the main mathematical tool we use to improve the DCN-based feature extractors. Finally, we reserve Section 2.3 exclusively to discuss the related works regarding the feature extractors based on DCN;
- Chapter 3 introduces a framework to extract features of individual electrical loads with Scattering Transform. We discuss each step of the proposal in Section 3.1. We explain six proposed feature selection techniques in Subsection 3.1.4. We present the experimental setup in Section 3.2, the comparisons with state-of-the-art approaches in Section 3.3, and the chapter conclusions in Section 3.4;

- Chapter 4 expands the analysis of Chapter 3 to aggregated loads, and evaluates classification performance of Scattering Transform with two publicity aggregated datasets. Section 4.1 presents the proposed classification strategy, and each step of this proposal. Section 4.2 presents the experimental setup and the classification results obtained from the proposed framework. Section 4.3 presents the discussions about the obtained results, and compares them with related works in a more detailed approach than Chapter 3. The conclusions of Chapter 4 are highlighted in Section 4.4;

- Chapter 5 presents a novel architecture for NILM multi-label classification and disaggregation called ST-NILM, based on Scattering Transform. We detail each stage of the proposed ST-NILM architecture in Section 5.1. We present and discuss the results of ST-NILM in Section 5.3, and make the chapter conclusions in Section 5.4;

- Chapter 6 presents an interconnection between NILM and BTM estimation. Section 6.2 presents a novel high-frequency public dataset with aggregated appliances and PV distributed generation data, and Section 6.3 presents a complete ablation study on the influence of PV generation on the loads classification and vice-versa;

- Chapter 7 points to the overall conclusions, future works, and the open challanges for the sequence of this work.

Figure 2 shows a visual map of the structure of this document, highlighting the most relevant Sections and Subsections of the index.

1.4 PUBLICATIONS

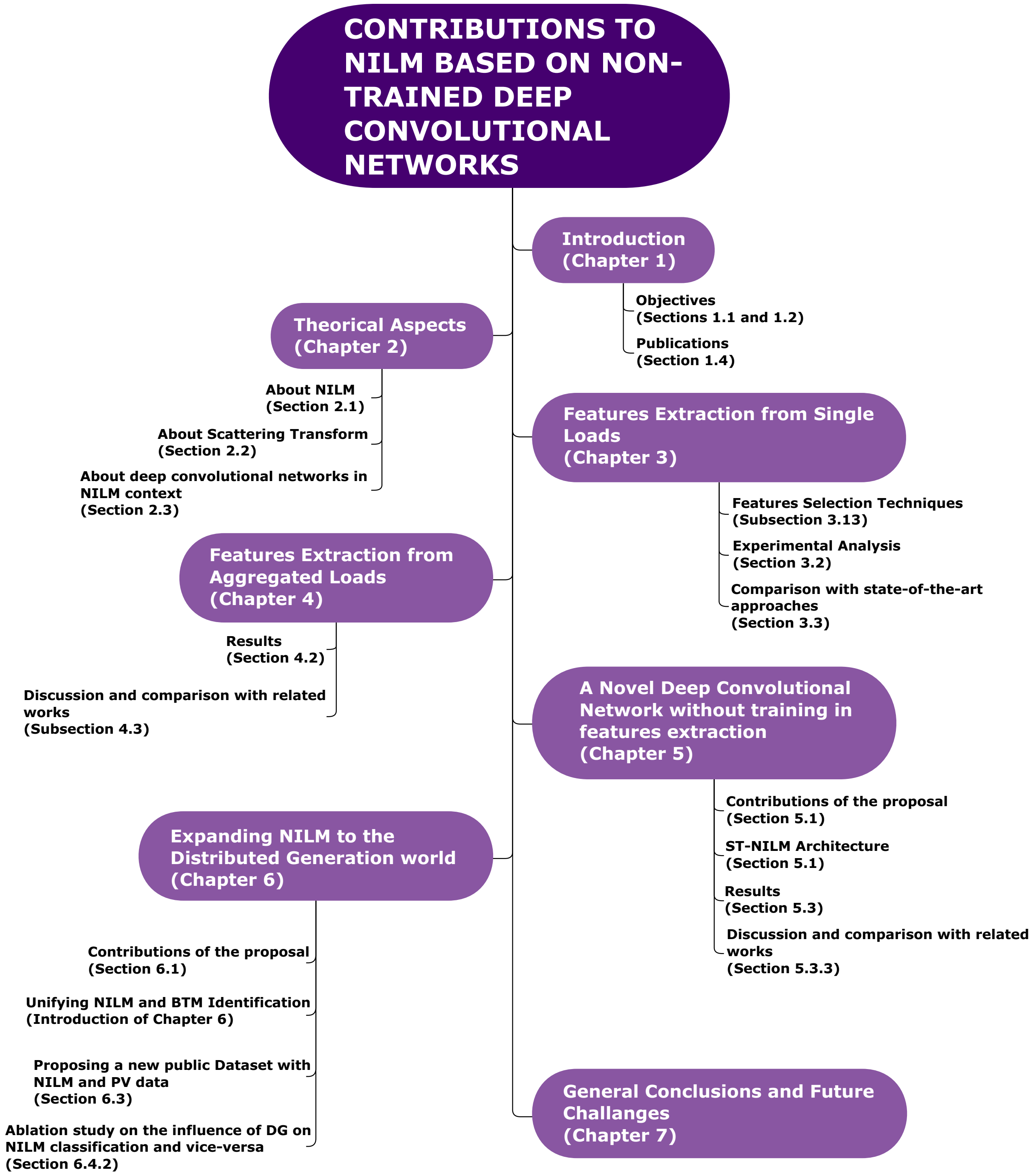
Table 1 shows the list of publications produced in the course of the author’s doctoral research. Table 1 also shows the chapter of this document related to each of these publications.

1.5 REPOSITORIES PRODUCED AS A RESULT OF THIS WORK

We have produced two public repositories, on GitHub, as a result of this work:

- *ST-NILM Repository:* Available at https://github.com/LucasNolasco/ST-NILM, and presented in detail in Chapter 5;

Figure 2 – Map of this document, with the most relevant Sections and Subsections


CONTRIBUTIONS TO NILM BASED ON NON-TRAINED DEEP CONVOLUTIONAL NETWORKS
Introduction (Chapter 1)
Objectives (Sections 1.1 and 1.2)
Publications (Section 1.4)
Theorical Aspects (Chapter 2)
About NILM (Section 2.1)
About Scattering Transform (Section 2.2)
About deep convolutional networks in NILM context (Section 2.3)
Features Extraction from Single Loads (Chapter 3)
Features Selection Techniques (Subsection 3.13)
Experimental Analysis (Section 3.2)
Comparison with state-of-the-art approaches (Section 3.3)
Features Extraction from Aggregated Loads (Chapter 4)
Results (Section 4.2)
Discussion and comparison with related works (Subsection 4.3)
A Novel Deep Convolutional Network without training in features extraction (Chapter 5)
Contributions of the proposal (Section 5.1)
ST-NILM Architecture (Section 5.1)
Results (Section 5.3)
Discussion and comparison with related works (Section 5.3.3)
Expanding NILM to the Distributed Generation world (Chapter 6)
Contributions of the proposal (Section 6.1)
Unifying NILM and BTM Identification (Introduction of Chapter 6)
Proposing a new public Dataset with NILM and PV data (Section 6.3)
Ablation study on the influence of DG on NILM classification and vice-versa (Section 6.4.2)
General Conclusions and Future Challanges (Chapter 7)

Source: Own authorship

- *DG-NILM-V1 Repository:* Available at https://github.com/evertoneie/DG-NILM, and presented in detail in Chapter 6.

Table 1 – Publications related to this qualification document

	Title	**Type**	**Status**	**Publisher**	**Related Chapter**
1	Scattering Transform for Classification in Non-Intrusive Load Monitoring	Journal paper	Published	Energies MDPI	Chapter 4
2	Performance of Scattering Transform Feature Extraction for Electrical Load Classification	Conference paper	Published	XV Congresso Brasileiro de Inteligência Computacional	Chapter 3
3	Features Extraction and Selection with Scattering Transform for Load Classification	Journal paper	Published	Learning & NonLinear Models	Chapter 3
4	ST-NILM: A Wavelet Scattering-Based Architecture for Feature Extraction and Multi-Label Classification in NILM Signals	Journal paper	Under Review	IEEE Sensors Journal	Chapter 5
5	A Novel High-frequency Dataset and Load Identification Approach with Photovoltaic Distributed Generation	Journal paper	Under Review	IEEE Transactions on Smart Grids	Chapter 6

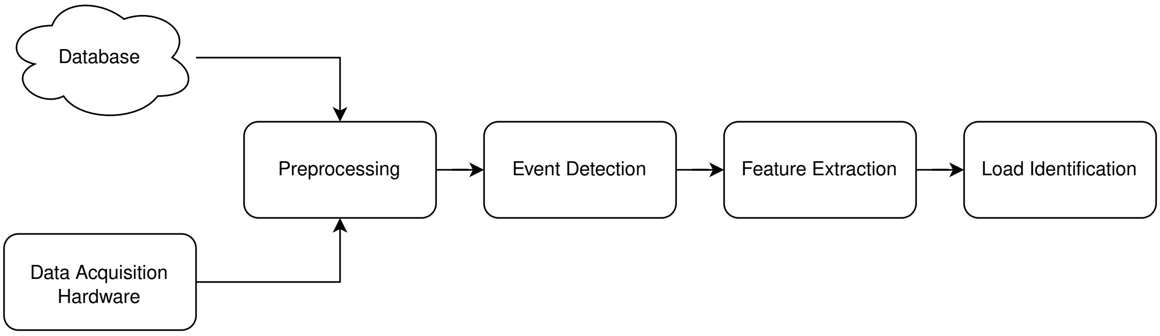
2 THEORETICAL ASPECTS AND RELATED WORKS

This chapter is dedicated to discussing the most important theoretical aspects used throughout this document and presenting recent related works, emphasizing the potential contributions of this research in the literature. In Section 2.1, we discuss the theoretical aspects regarding all the stages of NILM and some important general related works. Section 2.2 presents the Scattering Transform mathematical formulation, properties, and an analogy with Convolutional Neural Networks (CNN). We present in Section 2.3 a review of state-of-the-art feature-extraction methods used in NILM, considering that the state of the art in terms of feature extraction are the methods based on deep learning. Section 2.4 presents the definitions of Distributed Generation and Behind the Meter Estimation, relating these definitions to NILM. We cover the leading frameworks to classify and detect distributed generation in the NILM context in the Subsection 2.4.1 and the main datasets with NILM and distributed generation in Subsection 2.4.2. Section 2.5 presents the potential contributions of this research in the literature.

2.1 THEORETIC ASPECTS ABOUT NILM

NILM methods consist of discovering segmented information for each electrical appliance connected to a common point without measuring the individual electrical quantities of each appliance. Disaggregation and classification are performed from an aggregate electrical signal, which corresponds to the sum of the electrical currents of each electrical load connected to the common bus. Figure 3 shows a typical structure of the NILM methods proposed in the state-of-the-art.

Figure 3 – Typical NILM Fremework.



Source: Own authorship.

We present, in the following Subsections, the stages of the general diagram in figure 3.

2.1.1 Preprocessing

The electrical signals for NILM tasks can be obtained from public datasets and dedicated acquisition hardware. Generally, these data from the datasets and the acquisition system are not properly formatted for the NILM task. The preprocessing stage in Figure 3 then consists of standardizing these input signals so that the particular NILM task can be executed appropriately. Some sub-tasks of the preprocessing stage are:

- Clipping of signals at adequate time intervals;
- Subsampling;
- Normalization;
- Denoising;
- Training, testing and validation subsets splitting (for supervised learning).

The choice of preprocessing subtasks depends on the nature of the data and may vary depending on the particular NILM task. We highlight below, in summary, the preprocessing strategies applied in each part of this work:

- In the Chapter 3, we apply subsampling, normalization, and clipping to prepare the COOLL dataset (PICON *et al.*, 2016) for the classification task (AGUIAR *et al.*, 2021a);
- In Chapter 4, the preprocessing consists of cutting two distinct regions of the original aggregated signal. One closed to the event (transient region) and another far from the event (steady-state region) for the *turn-on* and *turn-off* instants in the LIT-SYN (RENAUX *et al.*, 2020) dataset. For this case, we also use subsampling and normalization.
- In Chapter 5, we cut the aggregated input data and put these cut signals as inputs of the shared network;
- In Chapter 6, we generate segments from the original acquisition windows, considering several consecutive samples without load events after a specific event occurrence.

2.1.2 Event Detection

The event detection step identifies the time instants in which electrical loads are connected to the electrical grid. In cases of supervised learning, there is availability of a set of training examples in which there is previous information on the time instants in which the loads are connected or disconnected. Besides that, the event identification method must be able to generalize detection to new unknown examples (test set).

The main types of event detection methods for NILM, based on the classification presented in Liu (2019), are:

- *Expert Heuristics Models:* Hart (1992) presented a method based on standard deviation. Alcala *et al.* (2017) introduced an envelope-based extraction method. Expert heuristics models presented similar performance to the probabilistic models, without any training process (LIU, 2019). However, these methods were limited to low-frequency datasets;

- *Probabilistic Models:* Luo *et al.* (2002) presented an event detection method based on Generalized Likelihood Ratio (GLR). Berges *et al.* (2011) also proposed a GLR-based event detection method, and improved the model proposed in (LUO *et al.*, 2002). Yang *et al.* (2014) proposed an event detection method based on Goodness of Fit (GOF). Experimental results indicated that the false-positive performance of GOF based event detection model was better than that of the expert heuristics methods, but the variations in the temporal power signal that happen in the real world affect the results of the probabilistic models (LIU, 2019);

- *Template Matching Models:* LEEB *et al.* (1995) proposed to detect the events using template power signals. Shaw *et al.* (2008) used a template with Euclidian Distance. Basu *et al.* (2015) applied Dynamic Time-Warping to detect the events, and Liu *et al.* (2017) improved this proposal by applying Variant of Dynamic Time-Warping for events detection. Template matching event detection methods can be expanded to become appliance classification methods, but the distance information dependence can be a problem with these methods. For example, two similar signals out of phase can present a large Euclidean distance, impairing detection (LIU, 2019);

- *Hybid Models:* Zheng *et al.* (2018) proposed a Density-Based Spatial Clustering of applications with Noise (DBSCAN) event detection strategy. Afzalan *et al.* (2019) proposed and automatized DBSCAN method. Baets *et al.* (2018) improved the GOF method proposed by Yang *et al.* (2014). Zhao *et al.* (2016) proposed an adaptive event detection strategy based on Graph Signal Processing (GPS). Wild *et al.* (2015) proposed an event detection method based on active search. Hybrid detection models presented improvements of other methods (mainly probabilistic) but increased computational costs (LIU, 2019).

Once the switching instants of each electrical charge are known, some disaggregation methods can be used to determine the individual electrical signals from the aggregated signal. However, first, the feature extraction must be performed, detailed as follows.

2.1.3 Feature Extraction

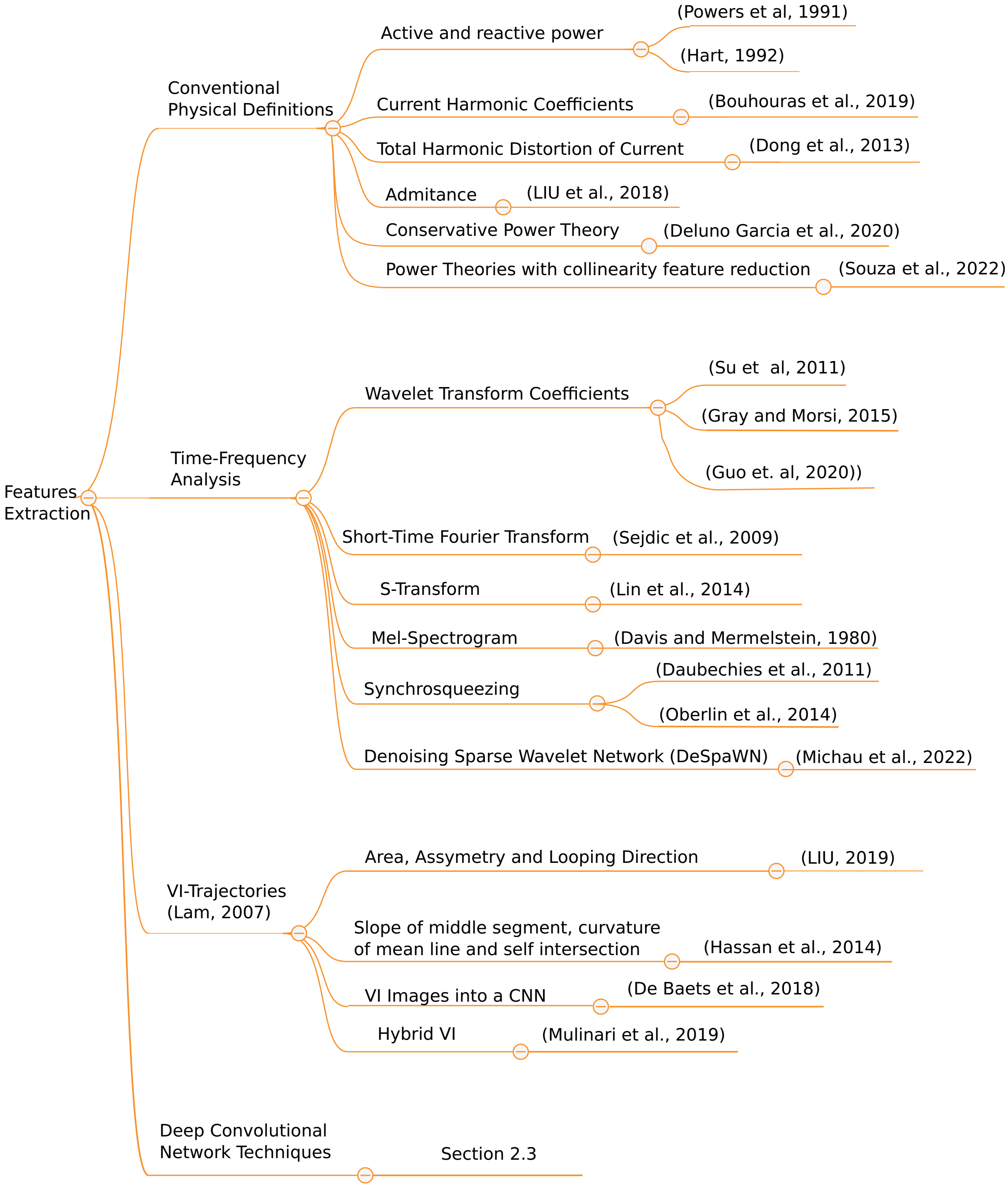
The feature extraction stage shown in Figure 3 maps a vector of one-dimensional signal $\mathbf{x} \in \mathbb{R}^{n_x}$ into a feature space $\mathbf{f} \in \mathbb{R}^{n_f}$, where n_x is the number of samples of each signal and n_f is the number of features.

The number of features and the meaning of each feature depend on the selected extraction method. For the classification task, the primary function of the feature extractor is to unscramble patterns in $\mathbf{x}$ so that the class separation is more evident in $\mathbf{f}$.

As we discussed in the Introduction, based on Sadeghianpourhamami *et al.* (2017), we categorize the feature extractors into (i) Conventional Physical Definitions; (ii) Time-Frequency Analysis; (iii) Voltage-Current (VI)-Trajectories. In the last five years, however, the scientific community has presented many feature extractors based on *deep Convolutional Networks*, which we number as category (iv). We dedicate Section 2.3 exclusively to discussing the related works regarding this type of feature extractor. Figure 4 shows the main categories of feature extractors presented in the literature. Besides that, we will also dedicate the Section 2.2 to introduce the *Scattering Transform*, which is the time-frequency feature extraction method used by us in our recent publications.

The first NILM feature extractors were based on the electrical characteristics of the signals. As shown in the Figure 4, in Powers *et al.* (1991) and Hart (1992) the authors used the average powers obtained directly from the voltage and current curves. The authors decomposed the current harmonic coefficients and used them as features in Bouhouras *et al.* (2019). Similarly, in Dong *et al.* (2013), the authors used THDi (Total Harmonic Distortion of Current) to define the feature map. In (LIU *et al.*, 2018), the admittance was used as a parameter to define the NILM

Figure 4 – NILM Feature Extractor Categorization.


Features Extraction
Conventional Physical Definitions
Active and reactive power
(Powers et al, 1991)
(Hart, 1992)
Current Harmonic Coefficients
(Bouhouras et al., 2019)
Total Harmonic Distortion of Current
(Dong et al., 2013)
Admitance
(LIU et al., 2018)
Conservative Power Theory
(Deluno Garcia et al., 2020)
Power Theories with collinearity feature reduction
(Souza et al., 2022)
Time-Frequency Analysis
Wavelet Transform Coefficients
(Su et al, 2011)
(Gray and Morsi, 2015)
(Guo et. al, 2020))
Short-Time Fourier Transform
(Sejdic et al., 2009)
S-Transform
(Lin et al., 2014)
Mel-Spectrogram
(Davis and Mermelstein, 1980)
Synchrosqueezing
(Daubechies et al., 2011)
(Oberlin et al., 2014)
Denoising Sparse Wavelet Network (DeSpaWN)
(Michau et al., 2022)
VI-Trajectories (Lam, 2007)
Area, Assymetry and Looping Direction
(LIU, 2019)
Slope of middle segment, curvature of mean line and self intersection
(Hassan et al., 2014)
VI Images into a CNN
(De Baets et al., 2018)
Hybrid VI
(Mulinari et al., 2019)
Deep Convolutional Network Techniques
Section 2.3

Source: Own authorship.

feature extractor. Deluno Garcia *et al.* (2020) proposed a heterogeneous approach for obtaining real-world data and extracting features from these data. The feature extraction stage was made with Conservative Power Theory (CPT). Although the hardware proposed by Deluno Garcia *et al.* (2020) allows sampling frequencies above 15kHz, the features were calculated in windows of 256 samples, corresponding to a 60Hz. Souza *et al.* (2022) proposed a strategy to obtain the most representative features for classifying electrical charges from high-frequency NILM datasets. The authors used power theories and active-reactive power (PQ) indicators to extract features and collinearity to obtain representative feature filters. The authors obtained a reduced set of low-frequency features from the high-frequency dataset, and the classification results (F1-Score above 95%) obtained with the reduced set of features (11 features or less) proposed by Souza *et al.* (2022) surpassed the state-of-the-art results. Despite the good classification results and the notable reduction of computational complexity in training, the authors did not present discussions regarding load detection.

Time-frequency-based feature extractors use time-frequency transforms to define the features space. Well known time-frequency transform are: Wavelet Transform (BURRUS *et al.*, 1998), Short-Time Fourier Transform (STFT) (SEJDIC *et al.*, 2009) and S-Transform (LIN; TSAI, 2014). One can find feature extractors for NILM applying Wavelet Transform Coefficients in SU *et al.* (2011), Gray and Morsi (2015), and Guo *et al.* (2020). The main disadvantage of these feature extraction methods is the dependency on the switching (turn-on or turn-off of each appliance) time. Some classical feature spectral extraction methods, such as the STFT (SEJDIC *et al.*, 2009), were time-shifting invariant representations but were not stable to time-warpings (MALLAT, 2012). Based on auto-correlation, the Wigner-Ville (SCHOLL, 2021) distribution has finer resolution than methods based on wavelets both for time and frequency but presents cross-terms. Cross-terms can be confused with auto-terms and penalty classification. The Hilbert-Huang transform (HHT) is based on Empirical Mode Decomposition (EMD); therefore, its base is expandable, allowing a good representation with physical meaning for non-stationary processes. However, the adaptive and empirical nature of HHT undermines a firm theoretical foundation (HUANG; SHEN, 2014). Methods based on Mel-Spetrograms (DAVIS; MERMELSTEIN, 1980) extracted features more closely related to human perception of sound, but their particular filter bank structure may not adequately represent electrical signals. Two reassignable Synchrosqueezing approaches were presented by Daubechies *et al.* (2011) and Oberlin *et al.* (2014). Authors in Daubechies *et al.* (2011) proposed the Synchrosqueeed Wavelet Transform (SSWT), which was based on the Continous Wavelet Transform (CWT). They added CWT components with the same instantaneous frequency to obtain a more focused representation. The method Daubechies *et al.* (2011) considerably decreased the dimensionality of the CWT representation but depended on prior knowledge of the nature of the input signal. Also, unlike

ST, method (SSWT) Daubechies *et al.* (2011) was a time-shifting variant representation. The authors at (OBERLIN *et al.*, 2014) proposed a synchrosqueezing method based on the STFT, named Fourier-Based Synchrosqueezing Transform (FSST). The approach of Oberlin *et al.* (2014) outperformed existing time-frequency analysis techniques in terms of its ability to extract detailed information about the frequency content of a signal. However, unlike ST, FSST was not stable to time warpings. Recently Michau *et al.* (2022) proposed the Denoising Sparse Wavelet Network (DeSpaWN) method, composed of a cascade architecture using a deep learning framework. The DeSpaWN method, unlike SSWT Daubechies *et al.* (2011), did not need a prior, as the coefficients were learned, and hard-thresholding allowed for a sparse representation. DeSpaWN, however, requires training a reduced number of parameters, unlike ST. Gauthier *et al.* (2022) proposed the Parametric Scattering Network (PSN), a version of ST with learned coefficients. PSN achieved state-of-the-art results for classifying two-dimensional signals, but applicability to time series was not addressed by the authors.

Several authors used the VI trajectories to determine extractors of NILM features. These strategies map electrical signals onto a Cartesian plane of voltage and current. From this mapping, a 2D image is obtained, and from this image, the features for NILM are extracted. In Liu (2019), the authors use area, asymmetry, and looping detection to determine features. In Hassan *et al.* (2014), the authors use the slope of the middle segment, the curvature of the mean line, and self-intersection. Baets *et al.* (2018) determined a weighted pixelated image for each electrical signal, and these images were placed at the input of a convolutional neural network (CNN), which performed the classification. Mulinari *et al.* (2019) proposed new features from VI images taken from both transient and steady-state periods.

Mallat (2012) proposed the Scattering Transform (ST) to define a time-frequency representation of a one-dimensional signal. ST has an architecture analogous to a Convolutional Neural Network (CNN). However, in the ST, the filter coefficients are not trained but are analytically determined using wavelets. In addition, the structure of the ST (number of layers and convolutional filters) can be also determined analytically. As a consequence, ST needs less training data than CNN for classification tasks (AGUIAR *et al.*, 2021b; BRUNA; MALLAT, 2012). ST is invariant to time-shiftings and stable to time-warpings, and these properties are particularly interesting for classifying electrical charges (AGUIAR *et al.*, 2021b). ST dealt with the time-shifting variance of the wavelet transform, improving pattern recognition tasks results by reducing variability (BRUNA; MALLAT, 2012). The structure of ST is analogous to the well-known Convolutional Neural Network (CNN). Deep CNN-based feature extractors for NILM have been proposed in the last years, as we show in Section 2.3.

2.1.4 Load Identification

The Load identification step determines which loads are present in the aggregated signal, taking the features as input. Generally, classification models with machine learning are used for this stage. Some classical methods used for load identification are k-Nearest-Neighbors, Decision Tree, Support Vector Machine (SVM), Ensemble Method, and Linear Discriminant Analysis (LDA) (CHERKASSKY; MULIER, 1998; VAPNIK, 1998), briefly detailed as follows:

- k-Nearest-Neighbors: this method classifies the test example by comparing it to the training dataset. The comparison is based on the Euclidean distance (CHERKASSKY; MULIER, 1998). This method fully relies on the distances between test samples and the training set, and due to that, it can be considered one of the simplest classification approaches (CHERKASSKY; MULIER, 1998). However, it requires the storage of all training examples, which may be a limitation for embedded systems with more restrictive memory requirements (LAZZARETTI *et al.*, 2020).

- Decision Tree: this method employs several concatenated binary splits arranged in a tree structure. Each split (node of the tree) refers to a particular feature and the corresponding parameter value for the comparison. In the test stage, the example is evaluated at each node of the tree, and the corresponding class is the majority class in the leaf node. The training process and the related splits are based on the Information Gain (CHERKASSKY; MULIER, 1998).

- Support Vector Machine: this classifier was proposed to maximize the separation margin between pairs of classes, based on a linear model (hyperplane) (VAPNIK, 1998). One of the significant advantages of the SVM formulation is that it can be formulated as a convex optimization. Besides, the problem can be defined in terms of the dot product between the features. Therefore, by using the *kernel trick*, the dot product can be replaced by a dot product kernel in feature space using the kernel, allowing nonlinear separation between classes (CHERKASSKY; MULIER, 1998). Here, we used the Gaussian kernel to evaluate nonlinear separations.

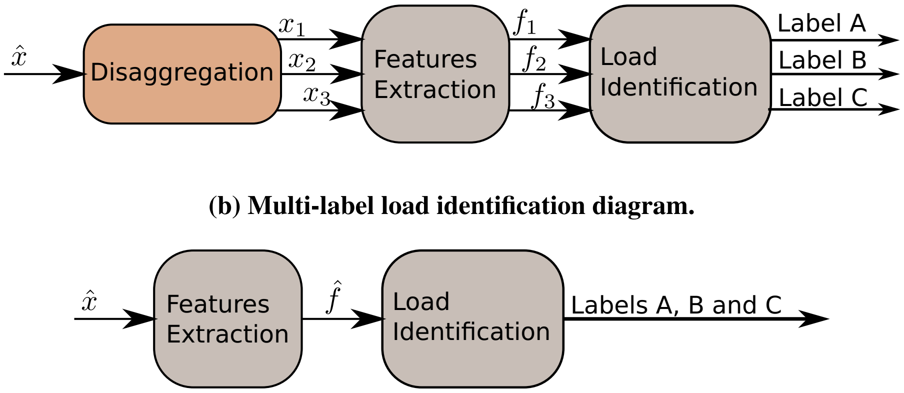
- Ensemble Method: this method combines different weak learner models by averaging the output of individual classifiers, improving the final accuracy (CHERKASSKY; MULIER, 1998). It is important to note that weak models usually have a poor individual response in terms of accuracy. However, combining the various classifiers tends to improve overall accuracy. One example is the random forest method that combines several decision trees to compose a final classification. The ensemble normally uses the following weak models: *AdaBoostM1, AdaBoostM2, Bag, GentleBoost, LogitBoost, LPBoost, LSBoost, RobustBoost, RUSBoost, Subspace*, and *TotalBoost* (CHERKASSKY; MULIER, 1998).

- Linear Discriminant Analysis: The LDA is a linear classifier that employs hyperplanes to differentiate data from two different classes. It assumes a normal distribution with an equal covariance matrix and equal priors for both classes. With that, the separating hyperplane is defined by reducing the dimensionality in such a way that it maximizes the separation between classes and minimizes the intraclass variance. So, its complexity and, consequently, overfitting is reduced (CHERKASSKY; MULIER, 1998).

Another strategy to identify the aggregated loads is to use multi-label classification archit-ectures, as presented in Nolasco *et al.* (2022). In multi-label approaches, an instance can be associated with multiple labels (READ *et al.*, 2011). Figure 5 compares traditional (multi-class) methods with multi-label methods for identifying electrical loads.

Figure 5 – Comparison between traditional and multi-label load identification methods.

(a) Traditional load identification diagram.



(b) Multi-label load identification diagram.

Source: Own authorship.

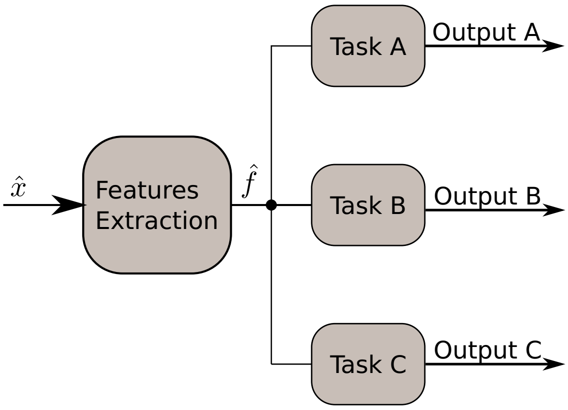
Note in Figure 5a, that traditional methods apply a disaggregation step to the aggregated signal $\hat{x}$, in order to define the individual electrical signals x_1,x_2,x_3. The features vectors f_1,f_2,f_3

are related to x_1,x_2, and x_3, respectively, in Figure 5a. Each individual features vector will be labeled one by one after passing through the classifier (load identification) stage.

We show a scheme of multi-label approach in Figure 5b. In figure 5b, the aggregated signal $\hat{x}$ enters directly in the feature extractor, without going through the disaggregation step shown in the figure 5a. The output of the feature extractor is the aggregated vector of features $\hat{f}$, which also enters directly into the classification stage (load identification). Finally, the aggregated signal $\hat{x}$ is labeled with three different classes: A, B, *and* C.

The NILM problem can also be addressed using multi-task learning (MTL) models. Caruana (1997) defines MTL as a mechanism that uses training tasks in parallel while using a shared representation. MTL allows for inductive transfer between tasks, improving learning performance on the current task (CARUANA, 1997). Figure 6 presents an example block diagram of an MTL approach with three tasks. Note that each task shares the same features $(\hat{f}$ extracted from signal $\hat{x}$. In deep learning applications, MTL consists of learning the weights of the features extractors simultaneously.

Figure 6 – Diagram of a multi-task learning (MTL) approach.



Source: Own authorship.

We apply traditional classification methods for NILM in Chapters 3 and 4. We propose a multi-label and multi-task approach for NILM in Chapter 5.

2.2 SCATTERING TRANSFORM

As we showed in section 2.1.3, the state-of-the-art presents several promising time-frequency approaches for feature extraction. Given that, we choose ST for the following reasons: (i) The time-shifting invariance and the time-warping stability properties are interesting for classifying appliances of the same category but different brands; (ii) The compact support and time-frequency resolution of the wavelets allow the loads detection; (iii) The exponential decay of the energy of the ST coefficients across the layers allows representing NILM signals with fewer coefficients than other time-frequency approaches; (iv) The non-trained and deterministic characteristics of ST allows a physically meaningful interpretation of the feature space.

Consider a set of features determined by the representations of two signals f and g. Let $\Phi(f)$ and $\Phi(g)$ representations of the signals f and g signals, respectively. Then, the Euclidean distance $d(f,g)$, defined by $d(f, g) = ||\Phi(f) - \Phi(g)||$ must be small for elements of the same class and large for elements of different classes (BRUNA; MALLAT, 2012). The similarity measure between $\Phi(f)$ and $\Phi(g)$ depends on the inner product of the two representations. The central question of the classification is to define a good kernel[1], which allows for a reliable measurement of similarity (MALLAT, 2012).

Consider a discrete-time signal $x[n]$, a translated signal $x_c[n] = x[n - c]$, $c \in \mathbb{N}$ and a deformed (time-warped) signal $x_d[n] = x[n - \tau[n]]$, $\tau[n] \in \mathbb{N}$ from the same type of electrical appliance. In real cases, two signals from the same appliance may both be translated and deformed. This occurs, for example, when the same load is switched on at different times in the same sampling window (translation) or when there is measurement noise (time-warping). The classifier should be invariant to translation and also to the small time-warping.

Let a wavelet $\Psi_\lambda[n]$ be defined by $\Psi_\lambda[n] = \lambda\Psi[n]$, where $\Psi[n]$ is the discrete time mother wavelet, $\lambda = 2^{-jQ}$, Q is the number of wavelets per octave, n is an integer that represent the n-th sample, and j is the scale factor. So the wavelet transform of $x[n]$ is:

$$Wx[n] = \{x[n] * \Phi[n], x[n] * \Psi_\lambda[n]\}_\lambda. \tag{1}$$

The wavelet transform has the following advantages: (i) it is stable for small-time deformations; (ii) it is well located both in time and in frequency, but it has the disadvantage of being a time-shifting variant. This happens because the Wavelet transform is calculated using convolutions (BURRUS *et al.*, 1998). To solve that problem, Mallat et al. (MALLAT, 2012) used the coefficients module, followed by the average in time:

$$\{|x[n] * \Psi_\lambda| * \Phi[n]\}_\lambda, \tag{2}$$

being $\Phi[n]$ a low-pass filter that implements the average. The modulus and average operators guarantee the time-shifting invariance, but results in loss of information (BRUNA; MALLAT, 2012; MALLAT, 2012).

[1] Let an operator f that takes elements from set A to set B, which preserves the form (homeomorph). The kernel of this operator f is the inverse image of $0 \in B$.

Mallat (2012) proposed applying successive modulus and average operations to new layers of convolutions with wavelets starting from $\{|x * \Psi_\lambda| * \Phi(t)\}_\lambda$. This gives rise to the Scattering Path, in discrete time, of sequence $x[n]$, given by:

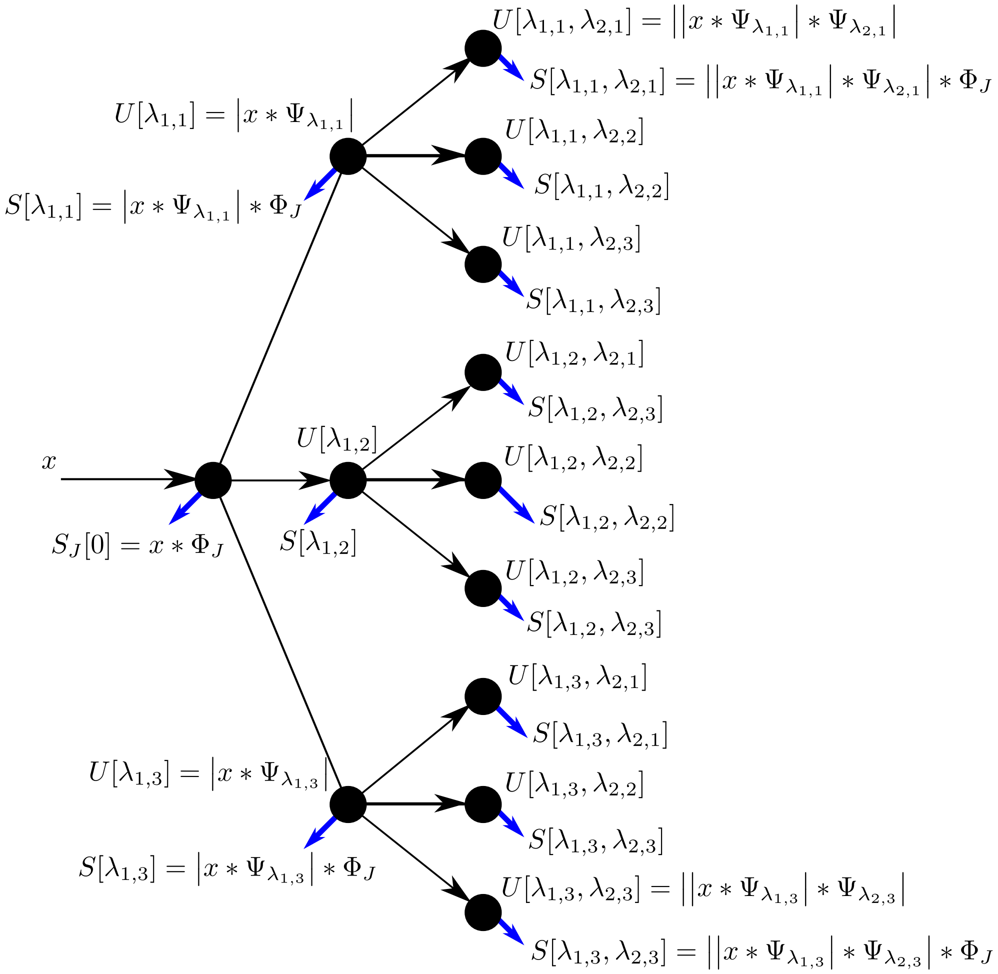
$$S_m[n,\lambda_1,\lambda_2,\ldots,\lambda_m] = ||x[n] * \Psi_{\lambda_1}| * \Psi_{\lambda_2}| \cdots * \Psi_{\lambda_m} | * \Phi[n], \tag{3}$$

depending on the order m and the frequency scales $\lambda_1,\ldots,\lambda_m$.

The total number of coefficients for the Scattering transform of a discrete signal x is $Q^m log_m N_s$, being N_s the total number of samples of x.

Figure 7 shows the structure of the ST graphically. The blue arrows in Figure 7 represent the convolution operation with the low-pass filter Φ_T. The set of all coefficients obtained as a result of the average operation (the tip of the arrows in Fig 7) composes the *Scattering Transform*. The discrete electrical signal $x[n]$ passes through the first layer of convolutions with the wavelets $\Psi_{q,i} = \Psi_{1,i}$. The subscript i is the frequency scale of the first filter bank. The convolution module is taken at each tree node in the first layer. The output of each node in the first layer is used to calculate the second level of the convolutional network.

Figure 7 – Scattering Transform Structure.


$U[\lambda_{1,1}, \lambda_{2,1}] = \big||x * \Psi_{\lambda_{1,1}}| * \Psi_{\lambda_{2,1}}\big|$
$S[\lambda_{1,1}, \lambda_{2,1}] = \big||x * \Psi_{\lambda_{1,1}}| * \Psi_{\lambda_{2,1}}\big| * \Phi_J$
$U[\lambda_{1,1}] = |x * \Psi_{\lambda_{1,1}}|$
$U[\lambda_{1,1}, \lambda_{2,2}]$
$S[\lambda_{1,1}] = |x * \Psi_{\lambda_{1,1}}| * \Phi_J$
$S[\lambda_{1,1}, \lambda_{2,2}]$
$U[\lambda_{1,1}, \lambda_{2,3}]$
$S[\lambda_{1,1}, \lambda_{2,3}]$
$U[\lambda_{1,2}, \lambda_{2,1}]$
$S[\lambda_{1,2}, \lambda_{2,3}]$
$U[\lambda_{1,2}]$
$U[\lambda_{1,2}, \lambda_{2,2}]$
x
$S[\lambda_{1,2}, \lambda_{2,2}]$
$S_J[0] = x * \Phi_J$
$S[\lambda_{1,2}]$
$U[\lambda_{1,2}, \lambda_{2,3}]$
$S[\lambda_{1,2}, \lambda_{2,3}]$
$U[\lambda_{1,3}, \lambda_{2,1}]$
$S[\lambda_{1,3}, \lambda_{2,1}]$
$U[\lambda_{1,3}] = |x * \Psi_{\lambda_{1,3}}|$
$U[\lambda_{1,3}, \lambda_{2,2}]$
$S[\lambda_{1,3}, \lambda_{2,3}]$
$S[\lambda_{1,3}] = |x * \Psi_{\lambda_{1,3}}| * \Phi_J$
$U[\lambda_{1,3}, \lambda_{2,3}] = \big||x * \Psi_{\lambda_{1,3}}| * \Psi_{\lambda_{2,3}}\big|$
$S[\lambda_{1,3}, \lambda_{2,3}] = \big||x * \Psi_{\lambda_{1,3}}| * \Psi_{\lambda_{2,3}}\big| * \Phi_J$

Source: Based on (ANDÉN *et al.*, 2019)

At the second layer, the convolution of $|x * \Psi_{1,i}|$ with a second set of wavelets, $\Psi_{2,k}$ results in a new set of coefficients. The index k represents the second frequency scale of the transform ($q = 2$), implemented by the second filter bank. Each second layer node comes from convolution-modulus with $\Psi_{2,k}$.

The cutoff frequency of Φ_T is half the bandwidth on the J-th sub-band. We propose to use the results of those convolutions to construct features used for classification. We detail extracting and selection features with the Scattering Transform in Subsection 3.1.3 of Chapter 3, Subsection 4.1.4 of Chapter 4, and Subsection 5.1.2.2 of Chapter 5.

2.2.1 Properties of Scattering Transform

The ST proposition is based on building a representation invariant to time-shifting. The convolution $x * \Psi_\Lambda$ is covariant to time-shifting. To make it invariant, one could directly apply the average, but $\int(x * \Psi_\Lambda)dt = 0$; therefore, the average is also zero, making the representation uninformative. To make it informative, Mallat (2012) shows that we must apply a nonlinearity to $x * \Psi_\Lambda$ before making it invariant. Among the possible nonlinearities, Mallat (2012) uses the modulus and presents it axiomatically as the only option.

The scattering transform is contractive. This means that with each iteration of average and modulus, the data get closer. This property comes from the fact that, given two complex numbers a and b, then $|a| + |b| \geq |a + b|$.

Given the signs x and y, and their transforms Sx and Sy, then the contraction property is:

$$||Sx - Sy|| \leq ||x - y|| \,. \tag{4}$$

The transform norm *scattering*, defined by $||Sx||^2$, is equal to the sum of all output signals from each of the layers shown in the figure 7, that is:

$$||Sx||^2 = \sum_{p \in \mathcal{P}} ||S[p]x||^2 \,. \tag{5}$$

Another property is that *scattering* preserves signal energy, i.e.:

$$||Sx||^2 = ||x||^2 \,. \tag{6}$$

Furthermore, the *scattering* transform is stable to small perturbations, so that:

$$||Sx - Sx_\tau|| \leq C\, sup\, |\nabla\tau(t)|\; ||x|| \,. \tag{7}$$

Note that the interpretation of the Equation 7 is that the distance between the transform of a signal x and a distorted signal x_τ is not zero, but has the size of the perturbation. This shows the stability of the transformation against deformation.

2.2.2 Analogy between Scattering Transform (ST) and Convolutional Neural Networks (CNN)

Although ST and CNN are different structures, with different distributions of numbers of filter components and energy concentration in the layers and topology, the fact that the two share a convolutional core (the convolution operation itself) encourages us to draw an analogy between the two.

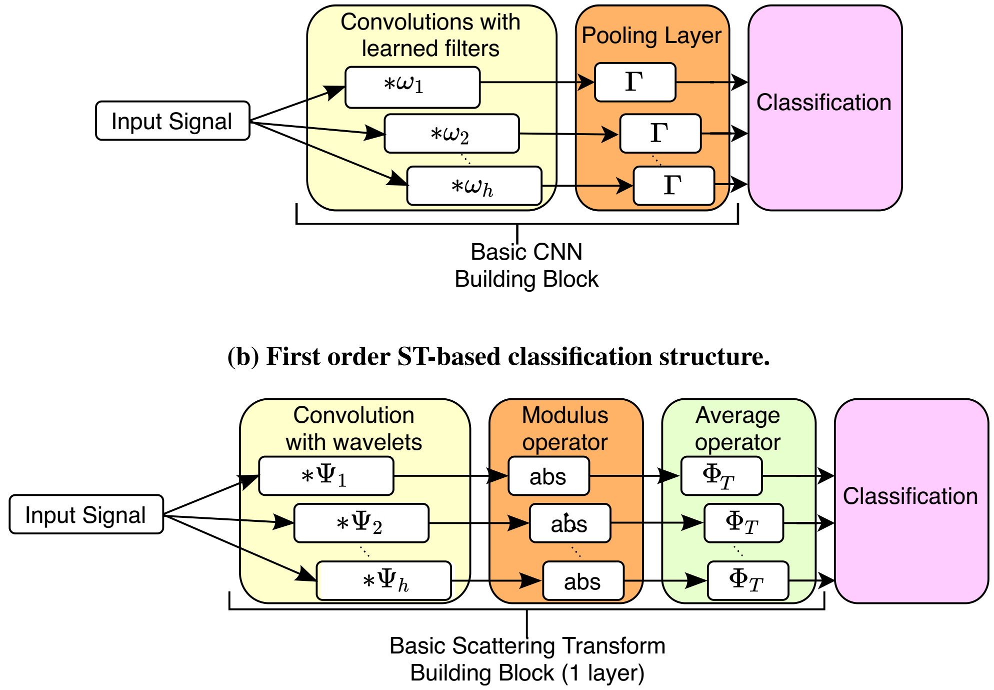
We explain here the structural differences between classical CNN and ST. The typical structure of a CNN comprises, as shown in Figure 8a, an input layer followed by one or more stages of activation, convolutional, and pooling layers (subsampling) (KIRANYAZ *et al.*, 2021). The discrete convolution operation applied in each feature extraction stage takes an input signal $x[k]$,$k \in \mathbb{N}$ and results (KHAN *et al.*, 2018) in the sequence $s_h[k]$, given by:

$$s_h[k] = \sum_{a=-\infty}^{\infty} x[a]\omega_h[k-a] = x[k] * \omega_h[k]. \tag{8}$$

The sequence $\omega_h[k]$ contains the convolutional filters (or kernels) coefficients for the h-th convolutional filter of the related feature extraction stage. These filter coefficients are generally learned in the training process (KHAN *et al.*, 2018; KIRANYAZ *et al.*, 2021). The pooling stage, on the other hand, applies the non-linear operator Γ to $s_h[k]$ (typically Γ_h is the maximum value or average value), and decreases the dimension of $s_h[k]$.

Figure 8 – Typical simplified structures of Convolutional Networks. We show both CNN and ST structures with only one convolutional layer. Both approaches may have other topologies based on the basic building blocks presented in this figure.

(a) Simple CNN-based classification Structure.



(b) First order ST-based classification structure.

Source: Own authorship.

Unlike CNN, the ST convolutional layer filters, represented by $\Psi_n, n \in [1:h]$, in Figure 8b are already predetermined by *wavelets* (BURRUS *et al.*, 1998), and therefore there is no need for training. Besides that, both CNN and ST structures have non-linearities Layers. In Figure 8a, CNN presents a pooling Layer and in Figure 8b, ST has a modulus and an average layers. The modulus and the average provide to ST the *stability* to small time-warpings and local time-shifting invariance properties, but there is a loss of information (MALLAT, 2012). To recover the lost information, the modulus layer output must be convoluted with a new set of *wavelet* filters, at a different scale (MALLAT, 2012). These properties are detailed in Subsection 2.2.1.

2.2.3 Observations about Scattering Transform

The Scattering Transform consists of convoluting an input signal with a set of *wavelets* (MALLAT, 2012) and then applying modulus and average operator to the convolutional stage output. Figure 8b represents the average operator by Φ_T.

The Discrete Time Wavelet Transform (DWT), the core of ST, maps an one-dimensional discrete signal $x[n]$ into a two-dimensional array of coefficients using a multirate digital filter bank (BURRUS *et al.*, 1998). Consider that n_x is the number of samples of x, $\varphi[n]$ is the scaling function, $\psi[n]$ is the mother wavelet, c_k is the k-th scaling coefficient, and $d_{j,k}$ is the detail coefficient for the scale j and the discrete-time k. The mother wavelet is translated by k and scaled by j, and this gives both ST and DWT good localization in time and frequency domains (BURRUS *et al.*, 1998). The multiresolution filter banks approach is suitable for computational implementation of DWT (BURRUS *et al.*, 1998).

The Scattering transform is based on the continuous wavelet transform (CWT). We commonly use filter banks to implement the discrete version of the Wavelet transform (BURRUS *et al.*, 1998). These filter banks decompose the input signal into other filtered signals. Each DWT filter bank consists of a low-pass filter (which results in *approximation* coefficients) and a high-pass filter (which results in *detail* coefficients). The DWT implementation uses frequency scales multiples of 2, and there is subsampling. This leads to the fact that the filter passbands of each layer of the DWT are multiple pairs of the previous layers. The consequence is that there is only one convolutional filter in the range of one octave of frequency for the DWT.

Unlike DWT, the computational implementation of CWT assumes a number of convolutional filters greater than 1 for an octave interval. The greater the number of convolutional filters per octave, the closer to the theoretical CWT the implementation will be.

Scattering is based on CWT, but computational practical implementations are discrete. In the computing context, when we refer to a CWT, we are talking about a technique in which there is more than one wavelet per octave (each time the frequency doubles), and in which there is a single sample displacement between the wavelets in the time domain (minimum shift on the time axis).

The Q parameter defines the number of wavelets per octave for each Scattering filter bank. Each layer of the ST has a bank of filters, and therefore a set Q_i of wavelets per octave for the i-th layer. Complex wavelets such as Morlet are preferable to CWTs as they carry phase information. As the CWT is the basis of the ST, this conclusion is also reasonable for the ST.

CWTs have constant Q, and it makes the bandwidth of each filter that compose the bank proportional to the center frequency. Consequently, we conclude that filters with higher center frequencies have greater bandwidth (less frequency localization), and filters with lower center frequencies have smaller bandwidth (better frequency localization). Due to the time/frequency reciprocity, the higher frequency wavelets have a better location in time, and those with a lower frequency have a worse location in time.

Each filter bank has Q_i center frequencies, separated by logarithmic intervals (not linear). The reason for this stems from the issue of the constant quality factor. For CWT (and ST, therefore), the interval is a multiple of $2^{\frac{1}{Q_i}}$.

The difference between CWT and ST occurs by the ST invariance range. While CWT has logarithmic intervals between the center frequencies for the entire spectrum, ST has to ensure that the time support of the lowest frequency filters does not exceed the invariance range. The consequence is that algorithms must detect this limitation and impose linear spacing for lower frequencies (the lower the frequency, the tighter the passband, and the greater the temporal support, because of time-frequency duality).

The three transforms (CWT, DWT and ST) are implemented through convolutions with wavelets. The CWT and DWT transforms are time-shifting variants, since $x[n] * \Psi \neq x[n-k] * \Psi, k \in \mathbb{N}$ (BURRUS *et al.*, 1998; MALLAT, 2012). On the other hand, for the case of ST, it is a time-shifting invariant representation. Such invariance comes from the successive averaging operations of each set of coefficients of each layer (MALLAT, 2012).

Table 2 compares DWT, CWT, and ST, given what we expose in this Section. Note that sparsity is related to the redundancy of the representation. CWT has more coefficients (and more sparsity) than DWT, as DWT only takes the power of two discrete scales from the mother wavelets (BURRUS *et al.*, 1998). One can truncate ST in a particular order level and then control the number of transform coefficients (sparsity). Because of these considerations, we consider in Table 2 that CWT has high sparsity, DWT has low sparsity, and ST has intermediate sparsity.

Table 2 – Comparison among CWT, DWT and ST.

Method	Sparsity	Time-shifting invariance	Computational Complexity
CWT	High	No	High
DWT	Low	No	Low
ST	Mid	Yes	Low

2.3 WORKS RELATED TO FEATURES EXTRACTION WITH DEEP CONVOLUTIONAL NETWORKS

The relationship between Feature Extraction for NILM and Deep Convolutional Networks started with the work of Kelly and Knottenbelt (2015a). In that work, the authors proposed three deep networks for NILM, applied to a low-frequency dataset. Each proposed architecture had a dedicated output network to classify each load. The authors proposed: (i) an architecture based on recurrent networks (bidirectional LSTM + CNN); (ii) Denoising Autoencoder; (iii) Regress Start Time, End Time, and Power. The three proposed networks extract NILM features and also perform disaggregation. In strategies (i) and (ii), padding and sliding windows are used, and strategy (iii) uses a probabilistic output for each appliance's power demand, converting this to a single vector per appliance.

Since Kelly and Knottenbelt (2015a), many other papers proposed deep Convolutional Network architectures for feature extraction for NILM tasks. Several works used low frequency datasets to train CNN architectures for NILM feature extraction (CHEN *et al.*, 2018; CHEN *et al.*, 2020; MORADZADEH *et al.*, 2021; CHEN *et al.*, 2020; MASSIDDA *et al.*, 2020; KASELIMI *et al.*, 2019). In Chen *et al.* (2018), for instance, the authors proposed a sequence-to-sequence 1D-CNN architecture for NILM with superior disaggregation results for high power loads, but with an inability to disaggregate low power loads. Similarly, in Chen *et al.* (2020), a low-frequency load disaggregation architecture called CAEBN-HC was proposed, which was based on 1D-CNN with batch normalization (BN) and Hill Climbing (HC). The authors extracted the temporal features with a CNN, applying BN to avoid explosion or vanishing gradient in the training process. HC was applied to adjust the hyperparameters, and the results obtained were promising, i.e., from 7 to 10W of Mean Absolute Error (MAE). However, the authors only addressed the disaggregation of few loads with high power and did not present results in terms of low power and several aggregated loads. Chen *et al.* (2020) addressed the intention to use data augmentation to improve the results with more aggregated loads. In Moradzadeh *et al.* (2021), the authors proposed a deep CNN architecture for NILM classification, with accuracy results greater than 96% for the REDD dataset. Moradzadeh *et al.* (2021) proposed to classify appliances of households not included in the training stage, but the disaggregated load curves were not available, and the results are limited to low-frequency REDD dataset only for three selected loads.

The SCANnet proposed in Chen *et al.* (2020) used Context-Aware Feature Integration, which is a map of additional features used to learn the contextual information for NILM disaggregation. The disaggregation results, Mean Absolute Error (MAE) in the range of 9-16W, were comparatively superior to the literature but limited to only six electrical loads with relatively high power, and dependent on data augmentation with Wasserstein Generative Adversarial Network (WGAN). The work proposed in Massidda *et al.* (2020) presented a CNN architecture called TP-NILM for load classification. The authors used electrical power as an input signal for the CNN. The feature extraction technique only detected whether the appliance is active and what was its average consumption in that mode. A stage called temporal pooling, which aggregates the features of different resolutions, was added in Massidda *et al.* (2020), which improved the temporal context. Both disaggregation and classification were performed in Massidda *et al.* (2020). The accuracy results were up to 97% for seen classes and up to 78% for previously unseen ones. The approach discussed in Massidda *et al.* (2020), on the other hand, is based on a multi-label strategy to classify multiple loads in the UK-DALE dataset. However, only three electrical appliances were simultaneously analyzed. A Multi-Channel Recurrent Tapped Delay Line CNN network (MR-TDLCNN) was proposed in Kaselimi *et al.* (2019), with training and testing in the AMPds dataset. The authors used three input channels on a CNN for disaggregation: active and reactive power, and current. This approach increased the discriminability, and the MAE results varied between 4.8W and 18W. The drawback was the need for more input data w.r.t. the other compared methods due to the overall size of the architecture.

In Zhou *et al.* (2021b), the authors applied a spatial clustering using density-based for applications with noise to classify different load curves. The idea of combining expert knowledge and deep learning models led up to 95% accuracy, overcoming other state-of-the-art deep learning methods. However, the method in Zhou *et al.* (2021b) needed a multi-feature method to transform the 1D load data into 2D matrix data. Authors in Matindife *et al.* (2021) proposed a 2D CNN structure that recognizes the load status. The proposed method used Gramian angular fields (GAFs) for encoding appliance low-frequency power series (1D) to an image (2D). The authors produced their own dataset and, consequently, proper comparisons are hampered.

The CNN-Long Short Term Memory (LSTM) hybrid model proposed in Zhou *et al.* (2021a) overcame CNN and LSTM methods for UK-DALE dataset in terms of accuracy (up to 98.87%). Despite these good results, the proposed CNN-LSTM method presented a much longer test time than CNN and LSTM ($1031\mu s$ vs $23\mu s$ and $27\mu s$ respectively). Furthermore, CNN-

LSTM needed a large amount of data for training to avoid overfitting, besides the performance depended on the depth of the neural network. Ding *et al.* (2021) proposed a disaggregation method independent of the depth of the CNN. The proposed method was based on multiple overlapping sliding windows that avoid overfitting and gradient vanishing in NILM. Authors in Ding *et al.* (2021) introduced a new extension of CNN called inception structured CNN to deal with NILM. F1-Score results (up to 70.7%) and the accuracy (up to 76.7%) overcame other deep learning methods when considering different sliding windows, but the proposed structure is complex and has a large set of trained filters. A comprehensive up-to-date review of deep models applied to low-frequency NILM can be found in Huber *et al.* (2021).

Although low sampling frequency data are easily obtained directly from smart meters, the extracted features are not as discriminative as those obtained with high-frequency datasets (RUANO *et al.*, 2019). Discriminative features for low power consumption appliances or transient features for devices containing switched static converters, for instance, cannot be achieved with low-frequency methods. In the sequence, we present methods that use data with high sampling frequency to overcome those limitations.

In different feature extraction methods, a 2D feature image is generated from the one-dimensional NILM signal, allowing the use of well-known image processing and deep learning techniques for NILM classification. In (HOUIDI *et al.*, 2020), the 2D image was generated by a time-frequency STFT. The spectrogram was applied as the input of a CNN, particularly designed for that work. The good location both in time and frequency allowed to deal with non-stationary multi-component signals, but some classification accuracy results were below the average of other methods, i.e., around 70% for the PLAID dataset. A similar approach is presented in (WU; WANG, 2019), in which spectrograms obtained from an STFT were used as input of the CNN. The strategy was devised to filter out background noise caused by other loads in the target load. In (BAETS *et al.*, 2018), the 2D representation was weighted pixelated VI images, obtained from the normalized VI curve in steady-state. The image was then inserted as the entry of a CNN, which performs the classification. The overall F1-Score was also below the average of the other compared methods (<78%), and the authors needed to use 2 datasets together (PLAID and WHITED) to reach those results. (MULINARI *et al.*, 2022) proposed to use the 2D Fourier transform to extract features from the VI curve of electric loads. FSscores of up to 97.7% were achieved by (MULINARI *et al.*, 2022), but the proposed method did not contemplate load detection and disaggregation.

Two multi-label approaches were presented in (FAUSTINE; PEREIRA, 2020a) and (FAUSTINE; PEREIRA, 2020b). Both have the advantages of being multi-label classification strategies, presented as alternative approaches to the traditional 2D image applied to the input of a CNN. In (FAUSTINE; PEREIRA, 2020b), a transition event is first located, followed by the Fryze Power Theory (STAUDT, 2008) in the aggregated current to extract the features together with a similarity matrix based on the Euclidean distance to reinforce the discriminability. A 2D image is then generated, being the input of a CNN with multi-label classification. In (FAUSTINE; PEREIRA, 2020a), on the other hand, a multi-label approach was proposed to improve the classification of loads of the same type but different brands. Given one cycle of the voltage and current, a Weighted Recurrent Graph (WRG) generates a 2D image for posterior classification. Accuracy results were better than other baseline methods that produce 2D images from V-I trajectories, but still lower than those presented in (FAUSTINE; PEREIRA, 2020b) (both for PLAID dataset). Moreover, results of (FAUSTINE; PEREIRA, 2020a) were obtained from submetered data (i.e., not aggregated).

Accuracy results above 98% were obtained in (HIMEUR *et al.*, 2021), which presented a technique called 2D phase encoding (2DPEP) to generate a 2D image from the NILM signal. Using time-domain feature extractors, the authors overcame state-of-the-art classification metrics for several distinct datasets based on sliding windows. Despite these promising results, authors in (HIMEUR *et al.*, 2021) had a high dependency on the event detection algorithm and the proposed approach used other classification methods based on classical Machine Learning, increasing the complexity. Authors in (CHEN *et al.*, 2022) proposed both temporal and spectral to define a dual power signature for each load. Each PS, represented as 2D image, enters a DCN structure followed by a fully connected network. The F1-Score results overcame VI method, but the authors did not compare these results with another CNN-based approach. Furthermore, the proposal of (CHEN *et al.*, 2022) was not time-shifting invariant.

In Yang *et al.* (2020), a discrete wavelet transform to obtain a 2D image from the aggregated current signal is proposed. The authors used the image as an input to a sequence-to-sequence CNN. The wavelet transform has the advantage of time-warping stability, but the data used in Yang *et al.* (2020) was submetered, and not naturally aggregated. The same limitation found in Yang *et al.* (2020) was noted in Morán *et al.* (2020). Particularly in Morán *et al.* (2020), the authors employed a deep convolutional autoencoder to extract features from individual hospital loads. Nevertheless, there is no disaggregation since the CNN input is obtained from a submetering network. In Lazzaretti *et al.* (2020), a multi-agent strategy was proposed to improve NILM classification, achieving accuracy results above 95% on LIT-dataset. Although this result was superior to the related literature, applying the multi-agent strategy in realistic cases may be compromised due to the high computational complexity. The authors in Mukaroh *et al.* (2020) also presented a CNN-based classification model for NILM, reaching 92% global accuracy on LIT-dataset, but with limitations in the feature extraction for loads with similar transitory shapes.

A real-time CNN-based method proposed in Athanasiadis *et al.* (2021) reached up to 99.2% accuracy, with a 100Hz sampling frequency data. This method used a three-stage structure: (i) Event detection; (ii) CNN Classification; and (iii) Power estimation, applying machine learning to detect the turn-on events and an heuristic algorithm to estimate the real time power. However, there were some limitations, such as: (i) Authors used a private dataset, which reduces reproducibility; (ii) Only three appliances of greater power consumption were used in the tests; (iii) The algorithm had problems in identifying loads with steep step-up transients.

Authors in Gomes and Pereira (2020) applied a pinball loss function (PB) to different DCN architectures, and compared the results with the mean squared error (MSE) loss function. Authors performed experiments with manually summed data and native aggregated data. Both with pinball loss function and MSE, all results were better with the sum of loads. As the DCN was not time-shifting invariant, authors in Gomes and Pereira (2020) reported problems with time-shifting signals. Authors in Jia *et al.* (2021) extracted features with bidirectional dilated convolution networks, increasing the length of receptive fields. Although the results outperformed other methods, the authors did not address the computational cost of training, and the non-causal attribute of the extractor did not allow real-time application. Both Hwang and Kang (2022) and Laouali *et al.* (2022) applied Long Short Term Memory (LSTM) to disaggregate NILM loads. Hwang and Kang (2022) and Laouali *et al.* (2022) presented promising applications for real-time. However, the lack of reproducibility of the private dataset of Laouali *et al.* (2022), the average F1-Score below 90% of Laouali *et al.* (2022) and the low-frequency data features discriminability are challenging to overcome.

Nolasco *et al.* (2022) proposed the DeepDFML architecture that integrated disaggregation, event detection, and multi-label classification. The DeepDFML had a shared DCNN stage, with three fully connected sub-networks, each one for one specific task. The authors proposed new metrics, as the work was pioneering in solving all three tasks at once. The authors needed external methods to generate more training data (data augmentation) despite the promising results – state-of-the-art for LIT dataset.

Chen *et al.* (2022) proposed two signatures for each appliance: one temporal and other spectral. The authors converted disaggregated 1D signals of voltage and current into images and defined those images as the signature of each load. Both temporal and Spectral signatures, 2D images, were inputs of two CNN structures, followed by a shared fully connected layer. The F1-Score validation results overcame VI and weighted recurrence graph methods, but the authors did not compare their proposal with other state-of-the-art CNN-based methods. Furthermore, the authors depended on the single load switching point detection (time-shifting covariant).

In Aguiar *et al.* (2021b), we expanded the analysis of Aguiar *et al.* (2021a) to multiple aggregated loads, using LIT-SYN dataset and Scattering Transform-based feature extractors. We compile the methodology and the results presented in Aguiar *et al.* (2021b) in Chapter 4.

We proposed in Aguiar *et al.* (2023) a framework based on Scattering Transform to extract features and classify electrical loads signals from COOLL Dataset Picon *et al.* (2016) and obtained state-of-the-art results for this dataset.

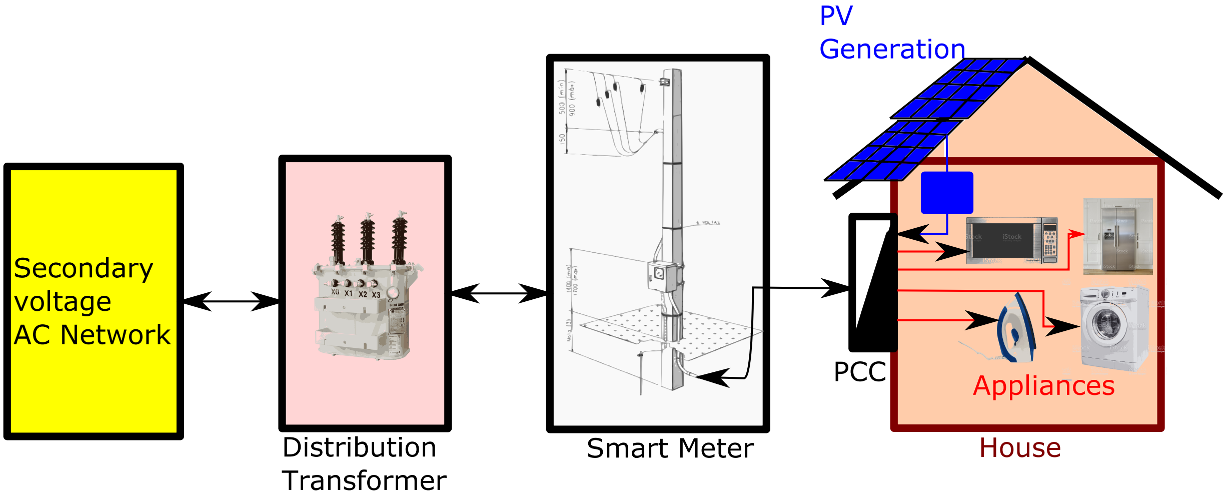
2.4 LINKING NILM, DISTRIBUTED GENERATION AND BEHIND THE METER ESTIMATION

Our work also addresses the identification and classification of photovoltaic inverters (PV inverters) connected to the secondary distribution network. The secondary distribution network is connected to the low voltage windings of the distribution transformer, corresponding, in Paraná, to 220V RMS for the two-phase system and 127V RMS for the single-phase system.

Our study considers a typical residence with a nominal voltage supply of 220V RMS. We also consider a photovoltaic (PV) distributed micro-generation (DG) source in this residence. In Brazil, distributed generation units with an installed capacity of less than 75kW are called “micro-generation”. We show this type of connection in figure 9.

Figure 9 shows a house with several appliances connected to a common connection point (PCC). This PCC comprises a circuit breaker distribution board, voltage surge protection devices (DPS), and other protective elements. The AC output of the photovoltaic generation is also connected to the PCC. The photovoltaic generation unit installed in the house, shown in blue in the figure 9, is composed of photovoltaic panels and a photovoltaic micro-inverter. The micro-inverter is a static converter that converts the continuous output voltage of the photovoltaic panels into alternating voltage, synchronized with the voltage in the PCC.

Figure 9 – Typical house with PV distributed generation.



Source: Own authorship.

The power flow in the network shown in figure 9 is represented by unidirectional and bidirectional arrows. Inside the house, electrical loads consume energy, so the arrows point from the PCC to them. On the other hand, PV generation only supplies energy to the grid, and therefore the arrow points to the PCC. Between the PCC and the Smart Meter, there is a two-way link. That means that both the PV generation can supply power to the utility grid and the utility can deliver power to the house. Likewise, the smart meter (typically installed on the pole at the utility's delivery point) connects to the distribution transformer. The distribution transformer converts the secondary distribution voltage (220V two-phase or 127V single-phase) into the primary distribution voltage (typically 13.8kV). This link between the smart meter and distribution transformer can also have a two-way power flow. Finally, the distribution transformer connects to the distribution secondary voltage network. Although it is not explicit in figure 9, the secondary voltage distribution network is connected to the National Integrated System (SIN) through transmission transformers located in specific substations.

Figure 9 also shows that measurements by the utility take place after the PCC and, therefore, outside the house. This means the utility does not have direct, real-time access to separate data on load consumption and the amount of instantaneous generation of the PV generation units. This feature motivated several literature research lines, generally called "Behind the Meter" Estimation (BTM). One of the main objectives of the BTM is to determine, from the concessionaire's point of view, the consumption of electrical loads and the energy production of the PV generators separately for each residence.

On the other hand, we saw in Section 2.1 that the most common definitions and classifications in the context of NILM were made concerning electrical appliances. Given the BTM problem, the increased penetration of distributed generation in the distribution network, and what we have already discussed about NILM, we feel encouraged to expand the concepts of NILM to the case of consumption units that produce energy, or *prosumers*. The primary Brazilian regulation for prosumers is ANEEL Normative Resolution number 1,059 of February 2023.

In general terms, ANEEL Normative Resolution 1,059 of February 2023 establishes the following:

- The installed power limits to define distributed micro and mini generation;
- The primary forms of energy considered as distributed generation;
- The procedures for installation, commissioning, and billing of micro and mini distributed generation;
- Guidelines for concessionaires and prosumers, given the possibilities of power flow inversion caused by prosumers;
- Transition rules from old regulations.

The high penetration of distributed generation in the distribution system can lead to some management, planning, and interconnection problems in the distribution network, such as:

- **Aspect 1:** Dimensioning of network protection systems;
- **Aspect 2:** Islanding detection;
- **Aspect 3:** Voltage control on substation buses;
- **Aspect 4:** Harmonic content;
- **Aspect 5:** Identification of Power flow inversion;
- **Aspect 6:** Clandestine connections of distributed generation units.

Our efforts for merging NILM to BTM estimation could potentially contribute to mitigating the problems related to the six aspects above.

2.4.1 Frameworks to classify and detect DG in the NILM context

Most micro and mini generation photovoltaic units are residential and are located *behind the meter* (BTM). Photovoltaic (PV) generation units of the BTM type may not be individually observable by the distribution or transmission systems. This occurs for the following reasons: (i) not every residential PV unit has the measurement of the electric current or generated power; (ii) there are privacy issues in data sharing; (iii) energy utilities may not have access to individual *smart meters* data. With this in mind, the disaggregation (or forecasting) of photovoltaic energy (BTM photovoltaic forecasting) is presented. A complete and up-to-date review on the topic is presented by Erdener *et al.* (2022).

The area of PV forecasting is active in the literature and consists precisely in predicting energy generation curves for PV BTM units. Lin *et al.* (2022) categorize the methods for forecasting PV BTM into *data-based* and *model based*. *Model based* are dependent on specific information of each PV unit (model, power, and physical characteristics). In contrast, *data-based* needs local measurement units, quality, and availability of measurement data locations of each unit.

Existing methods up to (LIN *et al.*, 2022) considered data availability from each prosumer by a centralized system. The problem with this is the privacy of each prosumer's data. To deal with this, Lin *et al.* (2022) proposed a probabilistic method that does not use private information from each prosumer to train the prediction model but federated learning. The *Federated Learning Bayesian neural network (FL-BNN)* is implemented in a decentralized way, respecting the federated learning framework.

Federated learning involves training a global model without sharing private data from individual installations. Training is carried out locally in a decentralized manner. FL-BNN is a regression problem that intends to discover the *net load* and the *generation load* of unobservable prosumers (those that do not have smart meters).

Brown *et al.* (2021) proposed a pioneering method of PV generation disaggregation based on real and censored measurements of smart meters. A censored measurement occurs when the PV unit generates more than the household consumption; consequently, the smart meter reading is zero. This is a limitation presented by the authors for the precise measurement of smart meters when there is BTM generation. The method proposed by Brown *et al.* (2021) was able to disaggregate the consumption of BTM PV generation in homes (locally) using real data from smart meters at low frequencies. For this, the authors used two combined datasets.

In Wang *et al.* (2022), the authors expanded the disaggregation analysis in PV to cases where prosumers also have a storage system. Such systems are named *hybrid rooftop solar battery systems (HRSBS)*. The authors mentioned that it would be easy to obtain the disaggregated powers simply by measuring the current of the battery and the PV system. However, this is not feasible in most cases, either because of the cost or legal issues related to privacy. The only information available is the smart meter energy curve, so the authors used *net load* to disaggregate. A weakness of Wang *et al.* (2022) method is the reliability of context data (observable data), which are available in a restricted scenario of high PV penetration. Another weakness of the work is the reliability of a physical model to generate energy storage data (as if it were a data augmentation). Wang *et al.* (2022) applied three feature vectors (PV generation, Load, and Battery Storage System feature vectors) as input to an optimization problem. In this way, the authors determined the disaggregated signals.

The method proposed in (PAN *et al.*, 2022) clustered the *net load* curves of the PV units together with the irradiance data. A model sensitive to energy consumption was developed from the temperature of days with similar irradiation. The authors improved by almost six percent the accuracy compared to state-of-the-art methods. Pan *et al.* (2022) main contribution was to propose an unsupervised disaggregation method that relies heavily on observable data (climatic and electrical) and not on proxy and physical modeling.

Wu *et al.* (2022) proposed an unsupervised disaggregation method on the neighborhood scale (an entire area of Taiwan). They proposed a fuzzy algorithm with four steps. First, they perform a pre-processing of the data. Later they clustered the PV installations. Next, they proposed a method for selecting the most representative PV points. Finally, they used the historical data of the most representative PV places to build a fuzzy model to estimate the PV generation in the cluster or region.

The authors in Abbood and Benigni (2018) used data-driven modeling to create a dynamic model of a PV microinverter without knowing the device's internal characteristics (such as topology, control, drive, etc.). The authors used neural networks from an FFT, trained with data emulated by a Power Hardware in the Loop. Despite having modeled the microinverter, the purpose in Abbood and Benigni (2018) was not precisely to extract the device's power signature but to contribute to commercial emulation tools. Furthermore, the lack of real data compromised the obtained results.

Mason *et al.* (2020) proposed an unsupervised regression model to disaggregate the generated power and the azimuthal angle of the panels in units of PV generation. The authors achieved a mean absolute error of 2.09% for disaggregation, but disaggregation and load detection were not performed.

Authors proposed in (JARAMILLO *et al.*, 2020) an extension of the typical NILM approaches conventionally used for electrical loads to identify residential DG. Jaramillo *et al.* (2020) determined the PV DG signature from PMU data and proposed a statistical features extractor along with an SVM classifier. Jaramillo *et al.* (2020) used a dataset that does not have transient information but synchrophasors at 50Hz: They measured both PV and loads currents and determined the grid current by simple subtraction. They used a sliding window of 50 samples (1s). From this windowed sample, the authors extracted the statistical features. The authors obtained F1-Scores of up to 96.16% but did not compare these results with other methods in the literature. The classification model had low dimensionality (5 variables), but the temporal window resolution was low (1s), and the method did not detect loads individually.

Jaramillo *et al.* (2021b) proposed a regression method to disaggregate the consumption curve of a residential PV system from the low voltage distribution power bar. Jaramillo *et al.* (2021b) used the SMART (BARKER *et al.*, 2012) public dataset. The authors achieved mean absolute errors of 5.2%, but the model had the following limitations: (i) low sampling frequency (1s); (ii) lack of comparison with other methods in the literature; (iii) Evaluation of feature extraction with kNN and RF only, and other methods were not presented in the paper.

According to Jaramillo *et al.* (2021c), there is a lack of observability in power distribution systems due to the increased penetration of DG and electric cars (EV), and NILM methods can help fill this gap. The authors proposed a method to identify both PV generation and EV consumption in low-voltage distribution buses and disaggregate the DG electrical profile pattern. They used the phasor data from the IEEE European Low Voltage test Feeder low voltage standard bars, using kNN for classification. The authors achieved F1-Scores of up to 92% but with the following limitations: (i) The 100W threshold used for noise may be considered too high for residential units with low consumption; (ii) The low sampling frequency (1/60Hz) limited the detection time resolution for the method, in addition to impairing the classifier's sensitivity to transient events; (iii) the method used more than one time-series to extract features (active, reactive, apparent power, voltage and current), increasing the dependence on data acquisition. Jaramillo *et al.* (2023) extended the work done in Jaramillo *et al.* (2021c), increasing the penetration of PV, including the random forest and multilayer perceptron classifiers, optimizing the hyperparameters of the classifiers, including statistical variables, in addition to testing three methodologies for selection of features. The authors proposed a new method for extracting the electrical signature of DG and EV applied at low voltage, obtaining up to 93% of F1-score for PV detection.

The methods proposed by (JARAMILLO *et al.*, 2021b; JARAMILLO *et al.*, 2021c; JARAMILLO *et al.*, 2020; JARAMILLO *et al.*, 2023) used classification and regression frameworks made with classic machine learning methods, not addressing deep learning strategies for DG classification or disaggregation. Deep learning methods have achieved state-of-the-art results for NILM in classifying and disaggregating electrical loads. These facts encourage us to apply NILM methods to DG using deep learning.

In order to expand the previous works by (JARAMILLO *et al.*, 2020) and (JARAMILLO *et al.*, 2021b), the authors proposed in (JARAMILLO *et al.*, 2022) a framework that contemplated both the disaggregation and the classification of EV and PV, using an ANN as a classifier for both tasks, in addition to the inclusion of statistical variables of active power as features. The authors achieved state-of-the-art results for both classification (99% F1-score for PV identification) and disaggregation (11% MAE considering the PV as target), but the proposed method presented the following limitations: (i) The approach considered neither load classification nor load detection; (ii) The sampled data, at low frequency, limited the time resolution in event detection; (iii) the authors did not test other features extraction methods besides the statistical one (frequency domain or time-frequency); (iv) the authors did not address other state-of-the-art frameworks for classification and disaggregation based on deep learning.

Goncalves *et al.* (2022) proposed a strategy to generate a dataset with residential and PV data. From the few available external examples, the (GONCALVES *et al.*, 2022) method allows the creation of training examples for both loads and DG. Despite being a practical method, which excludes the need for expensive and time-consuming measurements, the need for real high-frequency NILM and PV data is a problem for validating classification methods trained and tested by the generated dataset.

Jaramillo *et al.* (2021a) used an SVM classifier to detect aggregated appliances' on/off state and a 3.5kWp PV system from a residential installation in the United Kingdom. The authors obtained current and voltage samples from an OpenPMU module and got synchro phasors. The methodology of Jaramillo *et al.* (2021a) consisted in: (i) preprocessing the data, downsampling the original sample rate (12.8kHz) to the synchro phasors frequency (50Hz); (ii) extracting features from a moving windows containing 50 samples of synchro phasors sampled at 50Hz; (iii) obtain five statistic features from the synchro phasors; (iv) train an SVM classifier with a Gaussian kernel; (v) predict if each test sample has PV status "turned on" or "turned off. Despite being a pioneer work in obtaining the PV system signature, the work of Jaramillo *et al.* (2021a) leaves some gaps: (i) the proprietary dataset does not allow the reproduction of the results for comparison with other methods in future works; (ii) there is no standardization in the procedures for obtaining measurements; (iii) despite being computationally efficient, the proposed method is poorly discriminative, as it uses data sampled at low frequency.

2.4.2 Public NILM datasets related to DG

We have reserved this Subsection for discussing the primary public datasets for NILM related to DG. Typically one could classify the NILM datasets as High Frequency (represented by the letter H, with a sampling frequency higher than 50Hz, as defined in Basu *et al.* (2016)) or Low Frequency (represented by the letter L, with a sampling frequency lower than 50Hz, also as defined in Basu *et al.* (2016)). Besides that, Angelis *et al.* (2022) classified the datasets for NILM into three categories: residential (R), industrial (I), and commercial (C). Since the first publication of NILM methods by Hart (1992), most public datasets have been residential. A few examples of NILM datasets have Distributed Generation (typically PV) data.

Even with the storage capacity limitation, high sampling frequency datasets allow a more discriminative classification since the features extracted from them contain more transient information (AGUIAR *et al.*, 2021b). On the other hand, datasets with a low sampling frequency, even though they are not so discriminative, allow for looking at features with longer time windows (long-term properties). Considering these considerations, we create Table 3, which presents the main characteristics of high and low-frequency datasets related to residential, commercial, and industrial loads and cases with DG data.

Both low (L) and high-frequency (H) datasets previously published in the literature for NILM raised the following limitations:

- Most of the datasets presented only electrical measurements of residential appliances and did not present data from distributed generation;
- High-frequency datasets generally had short samples (less than 10s), while datasets with

Table 3 – Primary public datasets available in the literature. We highlight the following characteristics: Sampling Frequency Class (SFC) as High-Frequency (H) or Low-Frequency (L); Type, as Residential (R), Commercial (C), Industrial (I), Photovoltaic (PV) or the possible combinations; The time period that the dataset was obtained; The presence or absence of Multiple Sampled Loads (MSL); The aggregated appliance sampling frequency (AS); The individual device sampling frequency (DS); The total number of appliances (NoA); The total number of buildings where the data was obtained (B.) and the Electrical Characteristics (Charac.) measured (the active power P, reactive power Q, apparent power S, current I, voltage V, energy E and electromagnetic interference EMI)

Reference	Name	SFC	Type	Period	MSL	AS	DS	NoA	B.	Charac.
Kolter (2011)	REDD	H	R	119 days	yes	15kHz/ 0.5kHz	0.33Hz	92	6	P,V,I
Anderson *et al.* (2012)	BLUED	H	R	1 week	yes	12kHz	60Hz	50	1	P,Q,V,I
Gao *et al.* (2014)	PLAID	H	R	1-20s	yes	30kHz	30khz	1876	65	V,I
Kelly and Knottenbelt (2015b)	UK-Dale	H	R	2247 days	yes	16kHz & 167Hz	1Hz	109	5	P,Q,S,V,I
Shin *et al.* (2019)	ENERTALK	H	R	1714 days	yes	15Hz	15Hz	75	22	P,Q
Gulati *et al.* (2014)	HFED	H	R+C	–	yes	10kHz-5MHz	–	24	1	EMI
Picon *et al.* (2016)	COOLL	H	R	840 waveforms (6s each)	no	–	100kHz	42	1	V,I
Ribeiro *et al.* (2016)	SustData	H	R	10 days	yes	12.8kHz	0.5Hz	17	1	V,I,Q
Kahl *et al.* (2016)	WHITED	H	R+I	5123 waveforms (5s each)	no	–	44.1kHz	110	1	V,I
Kriechbaumer and Jacobsen (2018)	BLOND	H	R	50-213 days	yes	50-250kHz	50kHz	53	1	V,I
Renaux *et al.* (2020)	LIT	H	R	–	yes	15.6kHz	15.6kHz	26	1	V,I
Klemenjak *et al.* (2020)	SYND	L	R	180 days	yes	5Hz	5Hz	21	1	P
Batra *et al.* (2014)	COMBED	L	C	1 month	yes	33.3mHz	33.3mHz	–	8	P,I
Martins *et al.* (2018)	IMD	L	I	111 days	yes	1Hz	1Hz	8	1	P,Q,S,V,I
Bischof *et al.* (2018)	HIPE	L	I	92 days	yes	0.2Hz	0.2Hz	10	1	P,Q,S,V,I
Hebrail (2012)	UCI	L	R	4 years	yes	16.7mHz	16.7mHz	9	1	P,Q,V,I
Makonin *et al.* (2013)	AMPds	L	R	360 days	yes	16.7mHz	16.7mHz	20	1	P,Q, S,V,I
Nambi *et al.* (2015)	DRED	L	R	6 months	yes	1Hz	1Hz	12	1	P
Beckel *et al.* (2014)	ECO	L	R	8 months	yes	1Hz	1Hz	45	6	P,V,I
Makonin *et al.* (2013)	AMPdsV2	L	R	720 days	yes	16.7mHz	16.7mHz	20	1	P,Q,S,V,I
Murray *et al.* (2017)	REFIT	L	R	2 years	yes	0.125Hz	0.125Hz	177	20	P
Parson *et al.* (2016)	Dataport	L	R+C	4 years	yes	16.7mHz	16.7mHz	8598	722	P,S
Barker *et al.* (2012)	SMART	L	R+PV	3 months	yes	1Hz/ 0.2Hz	1Hz/ 0.2Hz	25	3	P,Q
Zimmermann *et al.* (2012)	HES	L	R+PV	1 year/1 Month	yes	8.33mHz	8.33mHz	251	26	E
Reinhardt *et al.* (2012)	Tracebase	L	R	1 day	yes	1Hz	1Hz	158	1	P
Batra *et al.* (2013)	iAWE	L	R	73 days	yes	1Hz	1Hz	63	1	P,Q,S,V,I
Monacchi *et al.* (2015)	GREEND	L	R	3-6 months	yes	1Hz	1Hz	–	7	P
Makonin *et al.* (2018)	RAE	L	R	72 days	yes	1Hz	–	–	2	V,I,P,Q,S
This Work	**DG-NILM**	H	R+PV	**open set**	yes	1kHz	1kHz	4	1	V,I

longer samples had low sampling frequency.

In Chapter 6, we propose a new dataset with the following advantages related to previous published datasets:

- The proposed dataset has high frequency aggregated measurements, adding multiple appliances with distributed generation;
- We present high-frequency samples of long duration (more than 10s);
- To evaluate the dataset, we present classification and disaggregation results with state-of-the-art NILM methods.

2.5 CONTRIBUTIONS

Bearing in mind the limitations and research gaps presented so far, the main contributions of this work are:

1. New feature extraction and selection methods for NILM using ST;
2. State-of-the-art classification results for both load disaggregation and individual electrical loads using ST;
3. Proposal for a new convolutional network architecture, more computationally efficient and less dependent on large datasets, with untrained filters using ST;
4. Ablation study on ST parameters variations on classification performance;
5. Construction of dedicated hardware for NILM and DG data collection;
6. Proposal for a new public dataset with aggregated data on electrical loads and PV distributed generation;
7. Ablation study on the impact of DG penetration on electrical loads classification and the impact of the aggregated loads on the DG identification.

We highlight contribution 7 (Ablation study on the impact of DG penetration on electrical loads classification and the impact of the aggregated loads on the DG identification) as the most significant contribution of our study.

3 FEATURE EXTRACTION WITH SCATTERING TRANSFORM FOR NON-INTRUSIVE LOAD MONITORING

We discussed in previous chapters that:

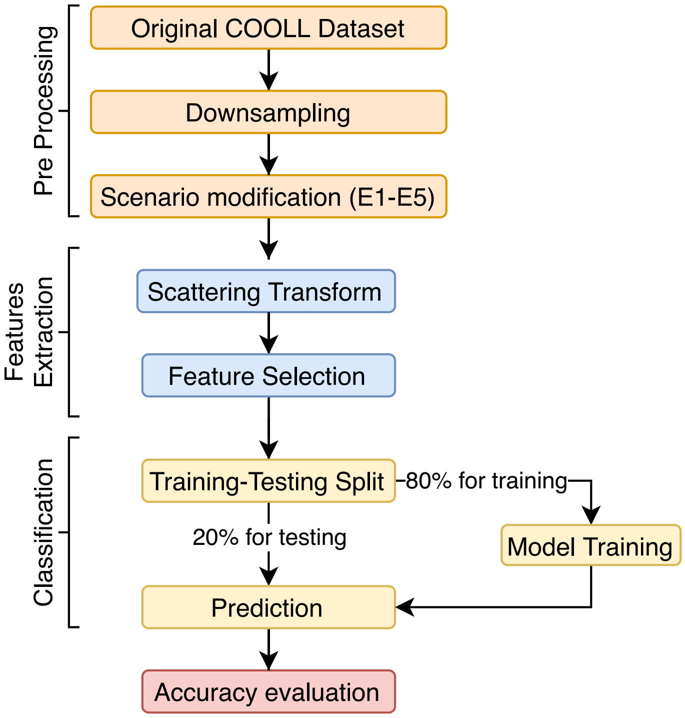
- In recent years, several feature extraction methods have been proposed for NILM using Deep Convolutional Networks;
- A recurring problem encountered in this type of method is the necessity to a large amount of data for training, once the network weights are trained;
- The Scattering Transform has an architecture analogous to CNN, but it does not have trained weights (wavelets analytically determine them).

With these considerations in mind, we propose in this chapter a framework based on ST for extracting features for NILM. We also propose six ways to select the features resulting from the ST in this proposed framework. We tested classification models on the COOLL (PICON *et al.*, 2016) dataset and obtained state-of-the-art results for that dataset. We tested our framework under reduced and subsampling dataset conditions. Furthermore, we present in this chapter, a study of the class separability, taking the reduced order features obtained by the proposed framework using the t-SNE (MAATEN; HINTON, 2008) method for dimensionality reduction.

3.1 PROPOSED FRAMEWORK

Figure 10 shows the proposed classification framework for NILM using Scattering Transform. The following steps constitute the proposed framework: (i) Pre-processing, (ii) Feature Extraction, (iii) Classification, and (iv) Evaluation of Results. In the following Subsections, we initially detail the dataset used, and later we will discuss each of the steps of the proposed framework (Figure 10).

Figure 10 – Proposed framework for NILM Classification. Our proposed framework could be used in other applications by replacing the COOLL dataset with another database.



Source: Own authorship.

3.1.1 COOLL Dataset

Considering that the features extracted from high-frequency data are more discriminative than low-frequency data (RUANO *et al.*, 2019), we chose the Controlled On/Off Loads Library dataset (COOLL), proposed in (PICON *et al.*, 2016). The COOLL dataset has submetered instantaneous current records from 42 different electrical appliances at a sampling frequency of 100kHz. Each electric load has 20 samples of 6s duration, at 100kHz, totaling 840 samples, as presented in Table 4.

Figure 11 shows the original current waveform from the Drill 1 single appliance. The red curve in figure 11 indicates the anottated turn-on and turn-off events of the appliance.

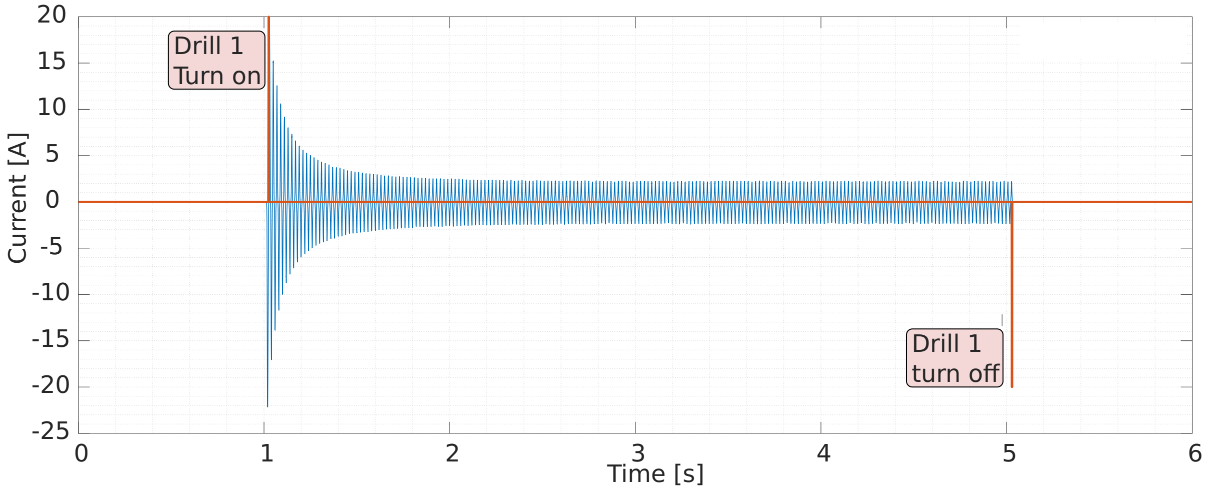
3.1.2 Preprocessing

The filter banks that implement the ST are multi-scale operators, and their implementation uses downsampling by two. For each downsampling by two, one of every two samples in the time domain is discarded, and the sampling frequency drops by half. For this reason, we changed the sampling frequency of the original input signal, coming from the COOLL dataset in

Table 4 – COOLL Dataset: Description of the appliances and categories.

Class Number	Appliance	Category	Class Number	Appliance	Category
1	Drill 1	Drills	22	Paint Stripper 1	Paint Strippers
2	Drill 2		23	Planer 1	Planers
3	Drill 3		24	Router 1	Routers
4	Drill 4		25	Sander 1	Sanders
5	Drill 5		26	Sander 2	
6	Drill 6		27	Sander 3	
7	Fan 1	Fans	28	Saw 1	Saws
8	Fan 2		29	Saw 2	
9	Grinder 1	Grinders	30	Saw 3	
10	Grinder 2		31	Saw 4	
11	Hair Dryer 1	Hair Dryers	32	Saw 5	
12	Hair Dryer 2		33	Saw 6	
13	Hair Dryer 3		34	Saw 7	
14	Hair Dryer 4		35	Saw 8	
15	Hedge Trimmer 1	Hedge Trimmers	36	Vacuum Cleaner 1	Vacuum Cleaners
16	Hedge Trimmer 2		37	Vacuum Cleaner 2	
17	Hedge Trimmer 3		38	Vacuum Cleaner 3	
18	Lamp 1	Lamps	39	Vacuum Cleaner 4	
19	Lamp 2		40	Vacuum Cleaner 5	
20	Lamp 3		41	Vacuum Cleaner 6	
21	Lamp 4		42	Vacuum Cleaner 7	

Figure 11 – Example of a Drill 1 waveform from the COOLL dataset, highlighting the times when the turn-on and turn-off events occurred.



Source: Own authorship.

the downsampling stage. The signal resulting from this step has a sampling frequency that is a power of 2 (8 192Hz), which makes the filter banks of the Scattering Transform feasible. After downsampling, we changed the dataset based on five different scenarios:

- **Scenario 1 (SC1):** Different number of cycles for the signal window, *with* turn-on events detection;

- **Scenario 2 (SC2):** Different number of cycles for the signal window, *without* turn-on events detection;
- **Scenario 3 (SC3):** Different number of examples per class;
- **Scenario 4 (SC4):** Different sampling frequency;
- **Scenario 5 (SC5):** Whole signal-length, at 8 192Hz of sampling frequency.

3.1.3 Feature Extraction of Individual Loads

We apply the Scattering Transform to extract features for NILM signals. We explain the mathematical definitions needed to clarify the Scattering transform in Subsection 2.2.

3.1.4 Feature Selection Techniques

We determine the features matrix employing six different strategies, briefly described below and detailed as follows:

- **Method A:** by taking the average of first order-path Scattering coefficients;
- **Method B:** by taking the energy of the first order-path Scattering Coefficients;
- **Method C:** by taking all the first order-path Scattering Coefficients;
- **Method D:** concatenating the averages of the first-order paths with the averages of the second-order paths;
- **Method E:** concatenating the energies of the first-order paths with the energies of the second-order paths;
- **Method F:** concatenating all first-order coefficients with all second-order coefficients.

3.1.4.1 Method A: Averages of first order-path Scattering coefficients

Let $S_{1,i} = |x * \Psi_{1,i}| * \Phi$ the first order-path coefficients of the i-th sub-band. Then, we compute the features by the averaging method ($A_{1,i}$), for the i-th sub-band, as

$$A_i = \frac{1}{N} \sum_{m=1}^{N} S_{1,i}[m], \tag{9}$$

where N is the number of coefficients at the i-th sub-band. We define the set of selected features for Method A as:

$$\mathcal{A} = \{A_i : i = 1, \cdots, J\}, \tag{10}$$

in which J_1 is the number of sub-bands at the first layer.

3.1.4.2 Method B: Energies of the first order-path Scattering Coefficients

The Method B of feature selection consists of taking the energy of the first order-path Scattering Coefficients. We compute the energy of the first-order coefficients, B_i, as:

$$B_i = \sum_{m=1}^{N} S_{1,i}^2[m], \tag{11}$$

where N is the number of coefficients at the i-th sub-band. Then, we use the B_i to select features for each example. We define the set of selected features for Method B as:

$$\mathcal{B} = \{B_i : i = 1, \cdots, J_1\}, \tag{12}$$

in which J_1 is the number of sub-bands at the first layer.

3.1.4.3 Method C: All the first order-path Scattering Coefficients

For Method C, we apply all the first order-path Scattering Coefficients as features. Let $\mathcal{C}$ be all the set of all first-order Scattering Coefficients, given by concatenating each first order sub-band Scattering Coefficients, as follows:

$$\mathcal{C} = \{S_{1,i} : i = 1, \cdots, J_1\}, \tag{13}$$

in which J_1 is the number of sub-bands at the first layer.

3.1.4.4 Method D: Concatenation of the first order and second order-path averages of the Scattering Coefficients.

Let $S_{2,i,k} = ||x * \Psi_{1,i}| * \Psi_{2,k}| * \Phi$ be the second order-path coefficients of the i first order sub-band and k second order sub-band. Then, for Method D, we compute ($D_{i,k}$), as:

$$D_{i,k} = \frac{1}{M} \sum_{m=1}^{M} S_{2,i,k}[m], \tag{14}$$

in which M is the number of coefficients at the i-th sub-band from first layer and k-th sub-band from second layer. Hence, we define the features vector for Method D as:

$$\mathcal{D} = \mathcal{A} \cup \{D_{i,k} : i = 1, \ldots, J_1;\;\; k = 1, \ldots, J_2\}, \tag{15}$$

in which J_2 is the total number of second-order wavelets filters.

3.1.4.5 Method E: Concatenation of the first order and second order-path averages of the Scattering Coefficients.

Let $S_{2,i,k} = ||x * \Psi_{1,i}| * \Psi_{2,k}| * \Phi$ be the second order-path coefficients of the i first order sub-band and k second order sub-band. Then, for Method E, we compute ($E_{i,k}$), as:

$$E_{i,k} = \sum_{m=1}^{M} S^2_{2,i,k}[m], \tag{16}$$

in which M is the number of coefficients at the i-th sub-band from first layer and k-th sub-band from second layer. Therefore, we define the features vector for Method E as:

$$\mathcal{E} = \mathcal{B} \cup \{E_{i,k} : i = 1, \ldots, J_1;\;\; k = 1, \ldots, J_2\}, \tag{17}$$

in which J_2 is the total number of second-order wavelets filters.

3.1.4.6 Method F: All first and second-order Scattering Coefficients

For Method F, we compose the feature vector $\mathcal{F}$ by taking all the first and all the second order-path Scattering Coefficients, as follows:

$$\mathcal{F} = \{S_{1,i} : i = 1, \ldots, J_1\} \cup \{S_{2,i,k} : i = 1, \ldots, J_1;\;\; k = 1, \ldots, J_2\}. \tag{18}$$

3.1.5 Classification

We perform the classification task considering the scenarios presented in Subsection 3.1.2 and the strategies of feature selection presented in Subsection 3.1.4. The training and test sets are separated from the feature matrix, and the label vector, i.e., 80% were used to train the classifier from the total number of instances and 20% for testing with a five-fold cross-validation.

We train the classification models using Ensemble Method (ENS), which presents the best results for different NILM evaluations, as presented in (LAZZARETTI *et al.*, 2020).

This method comprises a set of classifiers whose individual decisions are combined in some way – normally averaging – to classify new examples. In classification methods, ensembles are often much more accurate than the individual classifiers that make them up (CHERKASSKY; MULIER, 1998). Ensemble classification combines a set of trained weak learner models. It can predict ensemble responses for new data by aggregating predictions from its weak learners. This method can use different algorithms for sequential learning (weaker learning models), such as *AdaBoostM1, AdaBoostM2 Bag, GentleBoost, LogitBoost, LPBoost, LSBoost, RobustBoost, RUSBoost, Subspace*, and *TotalBoost*.

After training the classification model, we perform the prediction with the test subset (for each scenario). Let n_c be the number of classes of the dataset. For COOLL, $n_c = 42$. The prediction, for each experiment, results in a $\mathrm{M}_{n_c \times n_c}$ confusion matrix:

$$\mathrm{M} = \begin{bmatrix} a_{1,1} & \dots & a_{1,n_c} \\ \vdots & \ddots & \vdots \\ a_{n_c,1} & \dots & a_{n_c,n_c} \end{bmatrix}, \tag{19}$$

whose rows represent the predicted classes and the columns represent the actual classes. From the confusion matrix we calculated two performance metrics: F1-Score and Accuracy. The F1-Score, for each i-th class, is defined by

$$\text{F1-Score}_i = \frac{2 \times \text{Recall}_i \times \text{Precision}_i}{\text{Recall}_i + \text{Precision}_i}, \tag{20}$$

in which

$$\text{Precision}_i = \frac{a_{i,i}}{\sum_{k=1}^{n_c} a_{i,k}}, \tag{21}$$

and

$$\text{Recall}_i = \frac{a_{i,i}}{\sum_{k=1}^{n_c} a_{k,i}}. \tag{22}$$

The accuracy for each class is

$$\text{Accuracy}_i = \frac{\text{TP}_i + \text{TN}_i}{\text{TP}_i + \text{TN}_i + \text{FN}_i + \text{FP}_i}, \tag{23}$$

in which $\text{FN}_i = (\sum_{k=1}^{n_c} a_{k,i}) - a_{i,i}$, $\text{TN}_i = (\sum_{k=1}^{n_c} a_{k,k}) - a_{i,i}$, $\text{TP}_i = a_{i,i}$, and $\text{FP}_i = (\sum_{k=1}^{n_c} a_{i,k}) - a_{i,i}$.

From the F1-Score$_i$ and Accuracy$_i$ per class, we calculate the macro F1-Score and macro Accuracy by the expressions:

$$\text{F1-Score}_{\text{macro}} = \frac{1}{n_c} \sum_{i=1}^{n_c} \text{F1-Score}_i, \tag{24}$$

Table 5 – Scenarios and experimental cases details.

		Number of 60Hz Cycles						
Scn.	Case	N_{before}	N_{after}	Total time per example	Examples per class	Samples per example	Total of examples	Fs [Hz]
SC1	1	5	5	166.67ms	20	1 365	840	8 192
	2	10	10	333.33ms		2 731		
	3	20	20	666.67ms		5 461		
SC2	1	0	5	83.33ms	20	683	840	8 192
	2		10	166.67ms		1 365		
	3		50	833.33ms		6 827		
	4		100	1.67s		13 653		
		Number of Cycles:						
SC3	1	240		4s	10	32 768	420	8 192
	2				15		630	
	3				20		840	
SC4	1	240		4s	20	16384	840	4 096
	2			4s		8 192		2 048
SC5	1	360		6s	20	49 152	840	8 192

and

$$\text{Accuracy}_{\text{macro}} = \frac{1}{n_c}\sum_{i=1}^{n_c}\text{Accuracy}_i. \tag{25}$$

For simplification purposes, we refer to F1-Score$_{\text{macro}}$ and Accuracy$_{\text{macro}}$ as F1-Score and Accuracy, respectively, in the next Sections.

3.2 EXPERIMENTAL ANALYSIS

We use the library *Wavelet Scattering*, from Matlab® r2021, to implement the Scattering transform. We show in this Section the results obtained from the five scenarios derived from the COOLL dataset and described in Subsection 3.1.2. We performed the experiments using both the proposed method (ST) and the Discrete Wavelet Transform (DWT) as a baseline comparison. The investigations follow the structure of Figure 10. The choice of DWT as a baseline is because both ST and DWT are based on wavelet filters (BURRUS *et al.*, 1998). We modify the input signal from the original dataset for all experiments to analyze the performance metrics for the classification process. The modifications are detailed in Table 5.

First, we present the setup of the Scattering Transform for the experiments. Then, we show the setup of the DWT baseline extraction method. We present the classification results and discussions at the end of this Section.

3.2.1 Scattering Transform (ST) Experimental Setup

For the feature extraction, we parameterized the ST as follows:

- **Sampling Frequency (Fs):** 8 192Hz for Scenarios SC1, SC2, SC3, and SC5. 4 096Hz and 2 048Hz for SC4;
- **Number of Layers (m):** 2 layers;
- **Number of Filter Banks:** 2;
- **Number of wavelets per-octave, or Quality Factor (Q):** 8 for first layer, and 1 for second layer;
- **Type of Wavelet Filters:** Complex Morlet.

3.2.2 Discrete Wavelet Transform (DWT) Baseline Experimental Setup

For DWT, we use ten layers of detail signals. For each layer, we compute the energy of the wavelet coefficients. Then, we use the ten energies of each detail layer as features for the baseline classification model. We add to these features the energy of the approximation coefficients (BURRUS *et al.*, 1998), totaling 11 features for DWT baseline. The algorithm we use for DWT implementation is based on Meyer (1993), Mallat (1989), Daubechies (1990), implemented by function *wavedec*, on Matlab® r2021.

3.2.3 Feature Extraction and Classes Separability: Qualitative Analysis

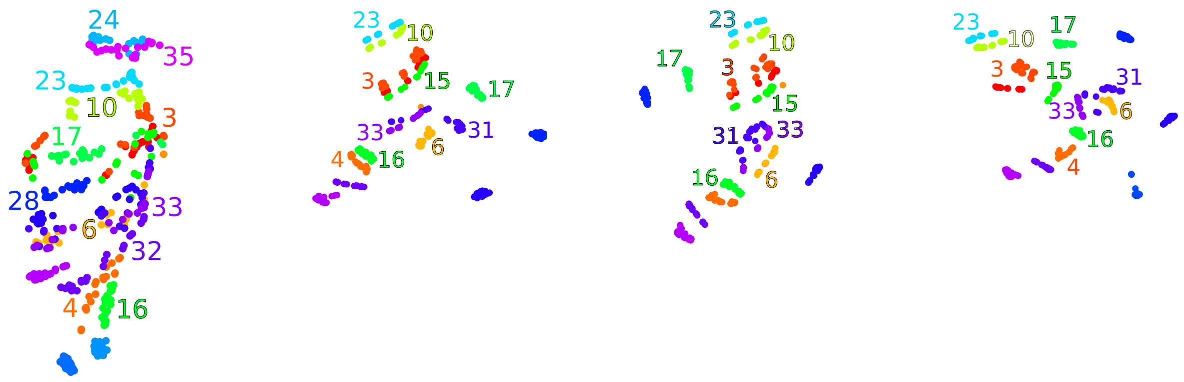
We are interested in showing the class separability when using ST for feature extraction and comparing the results with the features obtained with the Wavelet baseline. We use the t-SNE method, proposed in Maaten and Hinton (2008), to visualize the features, which were originally in the high-dimensional domain, in the Cartesian plane (dimension 2).

We apply both ST and Wavelet baseline to extract features from each example, each scenario, and each case presented in the table 5. Once this is done, we apply the t-SNE method to reduce high-dimensional features to the (2D) plane. We use $Perp = 30$ for the t-SNE experiments. As a result, we obtain a 2D representation for each case, scenario, and feature selection method. We choose the case of scenario 5 to show in Figure 12(a-d) because it represents the full dataset.

Also, we show only Methods A, B, and C in Figure 12 because these cases obtained better visual separation than Methods D, E, and F, in addition to having fewer features in original space (before t-SNE). Figure 12(a-d) shows the regions of interest, in which there is a more significant intersection between classes for each method of extracting and selecting features.

The numbers indicated in Figure 12 correspond to the classes with the highest correlation in the feature space. Each of these class numbers corresponds to a specific appliance in the COOLL dataset, as shown in Table 4.

Figure 12 – Analysis of the separability between classes, using the t-SNE method. We obtain the figures 12(b), 12(c) and 12(d) with Methods A, B and C for feature selection, respectively. The figure 12(a) was obtained with the traditional Wavelet transform. We noticed that the classes are considerably more separated in the Scattering Transform figures.



(a) Scenario 5, with Traditional Wavelet (baseline) for Feature Selection.

(b) Scenario 5, with Method A for Feature Selection.

(c) Scenario 5, with Method B for Feature Selection.

(d) Scenario 5, with Method C for Feature Selection.

Source: Own authorship.

The appliances involved in the separability analysis of Figure 12 correspond to class numbers 3 (Drill 3), 4 (Drill 4), 6 (Drill 6), 10 (Grinder 2), 15 (Hedge Trimmer 1), 16 (Hedge Trimmer 2), 17 (Hedge Trimmer 3), 23 (Planer 1), 24 (Router 1), 28 (Saw 1), 31 (Saw 4), 32 (Saw 5), 33 (Saw 6), and 35 (Saw 8) from Table 4. We observe that classes 3, 4, and 6 belong to the Drills category and classes 15, 16, and 17 to the Hedge Trimmer category. This behavior also occurs with the other Feature Selection Methods, as one can observe in Figs. 12(b), 12(c), 12(d) . Classes 28, 31, 32, 33, and 35 belong to the Saw category.

From Figure 12(a), one can observe that appliances 16 and 17 are located distant from each other, despite being from the same category (Hedge Trimmer). We obtain the same conclusion from 3 and 4 (Drills). In Figure 12(a), the Saws 28, 32, and 35 are close to each other, but there are intersections. The other Saw (35) is separated from 28 and 32. 24 (Router 1) and 35 (Saw 5) are appliances with completely different operation principles, but they are overlapped in Figure 12(a). 10 (Grinder 2) and 23 (Planer 1) are close in all sub-figures, unlike 16 (Hedge Trimmer 2) and 17 (Hedge Trimmer 3), which are far from each other. There is an evident intersection among 3 (Drill 3), 10 (Grinder 2), and 28 (Saw 1), which belong to different categories. The same occurs with 6 (Drill 6), 28 (Saw 1), and 33 (Saw 6).

Figs. 12(b), 12(c) and 12(d) have fewer overlapping regions than Figure 12(a). In Figure 12(b) the pairs 23 (Planer 1) and 10 (Grinder 2), 3 (Drill 3) and 15 (Hedge Trimmer 1), 4 (Drill 4) and 16 (Hedge Trimmer 2) are close to each other, but there is no overlap. On the other hand, there is an intersection between 33 (Saw 6) and 31 (Saw 4).

One can notice that the appliances 23 (Planer 1) and 10 (Grinder 2) are better separated when using Method B in Figure 12(c) than when using other methods in Figs. 12(a), 12(b) and 12(d). There is an intersection between appliances 31 (Saw 4) and 33 (Saw 6) with Method B, but the separability is better than Methods A and the Traditional Wavelet.

The separation of 3 (Drill 3) and 15 (Hedge Trimmer 1) is more prominent with Method C in Figure 12(d) than Methods A, B, and the Traditional Wavelets. Furthermore, 31 (Saw 4) and 33 (Saw 6) are not intersected, and the separation between the appliances 3 and 15 is much more evident than the other cases shown in Figs. 12(a), 12(b), 12(c) and 12(d).

In summary, the conclusions of the separability analysis are:

- The intersection regions between appliances are less frequent in ST selection methods;
- The tested ST-based feature selection methods showed similar visual separability characteristics;
- Separability between appliances of the same category was met with ST methods.

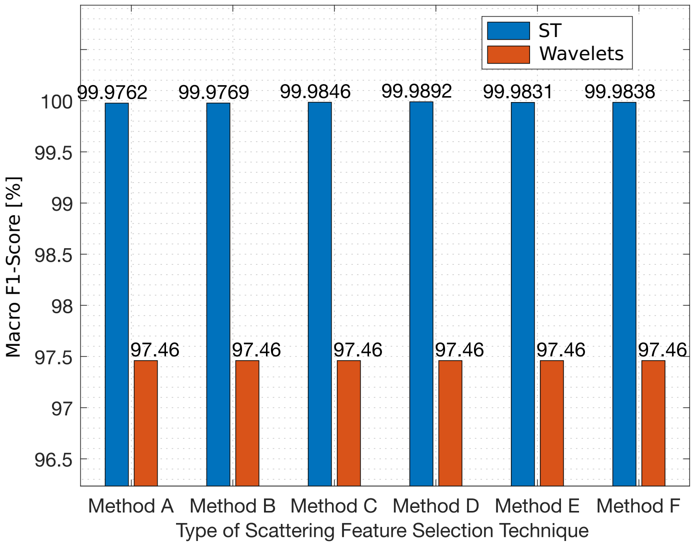
3.2.4 Results and Discussions

We follow the structure of Figure 10 and trained five classification models for each experiment. We use different training-test sets for each one of those classification models (five-fold cross-validation). Then, we perform the prediction five times for the Ensemble classifier, one for each different training-test set. Finally, we compute F1-Score and Accuracy for each trained model and each class. With these metrics, we calculate the macro F1-Score and Macro Accuracy.

Initially, we evaluate the influence of the feature selector type on classification metrics. For that, we perform the experiments according to Figure10 of each scenario and each case of

Tab.5 using the six feature selection methods presented in the Subsection 3.1.4. For each Feature Selection Method (A to F), we average the macro F1-Scores for all cases and Scenarios. Hence, we show these Global F1-Scores (all cases average metric) in Figure 13.

Figure 13 – Average of the macro F1-Scores for each method and feature extractor. One can note that ST overcame the Wavelet Baseline for all proposed Feature Selection Methods. There is no significant variation among the obtained global F1-Scores when we modify the selection Method.



Source: Own authorship.

The results in Figure 13 show that global F1-Score values (the average of all $F1 - Score_{macro}$ from all cases and all scenarios) do not depend on the method. In other words, there is no significant variation of Global F1-Score among all blue bars.

We show in Table 6, a comparison of the classification results obtained with each feature selection method for the five analyzed scenarios. We obtain the value of each bar in this figure by averaging all $F1 - Score_{macro}$, for each case, in the relative scenario.

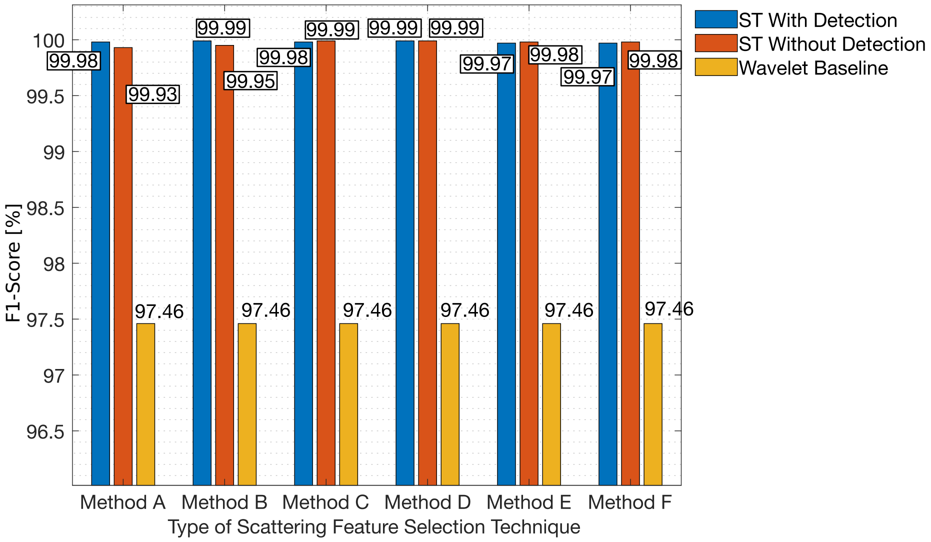
Table 6 – Comparison of the average F1-Score for each Scenario considering different feature selection methods. We obtain each of the bars in this figure by averaging the $F1 - Score_{macro}$ of all cases in each Scenario. We notice that even the maximum variation of F1-Score is relatively small (0.06%). This indicates that the global classification result is independent of the feature selection method.

	Method A	**Method B**	**Method C**	**Method D**	**Method E**	**Method F**
Scn.	F1-Score	F1-Score	F1-Score	F1-Score	F1-Score	F1-Score
SC1	99.98 %	99.98 %	99.98 %	99.99 %	99.98 %	99.98 %
SC2	99.95 %	99.96 %	99.98 %	99.98 %	99.76 %	99.98 %
SC3	100.00 %	99.99 %	99.99 %	100.00 %	99.99 %	99.99 %
SC4	99.99 %	99.99 %	99.99 %	99.99 %	99.98 %	99.99 %
SC5	99.98 %	99.98 %	99.98 %	100.00 %	100.00 %	99.98 %

We can observe in Table 6 that there is less than 0.06% variation of F1-Score in the classification, regardless of the scenario and the feature extraction method. Such a low difference in the classification metric reinforces the argument that the performance of the ST for classification in the COOLL dataset is independent of the evaluated feature selection methods.

We show in Figure 14, the global F1-Score results for each feature selection method, considering Scenarios 1 and 2. The objective of these experiments is to verify the influence of event detection on classification results. We obtain the blue bars in Figure 14 from the average of the 5-folds of case 1, Scenario 1 (in which case there are five cycles before the turn-on annotation and five cycles after the turn-on annotation, totaling ten cycles). We obtain the red bars of Figure 14 by averaging the 5-folds of case 2 of Scenario 2.

Figure 14 – Comparison of F1-Score between ST with event detection (Scenario 1, case 1, represented with blue bars) and ST without event detection (Scenario 2, case 2, represented with red bars) for different feature selection methods. The existence or not of event detection does not significantly interfere in the F1-Score obtained with the ST since the variation between the blue and red bars in the graph is less than 0.05%. In the figure, we also represent, in yellow, the mean value of the F1-Score obtained with the DWT (Wavelet Baseline).



Source: Own authorship.

One can observe from Figure 14 that the proposed ST framework overcomes the DWT baseline for all Methods of features selection analyzed. The minimum increase of F1-Score of ST over the baseline is 2.46%, when comparing ST without detection with the baseline applying Method A for feature selection. Besides that, the maximum variation between ST With and ST Without Detection is 0.07%, which indicates that the classification results were practically independent concerning the turn-on time-stamp.

So far, observe that:

- F1-Score does not significantly vary when using the different proposed ST feature selection methods;

- The F1-Scores with the ST surpass the results of the Wavelet baseline for all the cases tested;
- The position in time (time-stamp) at which the load is turned on does not significantly interfere with the classification results.

From Subsection 3.1.4, one can observe that Method A and Method B produce smaller features vectors. This implies simpler training stages, smaller overall complexity, and faster processing time. Considering that we verified in Figs. 13, 14 and Table 6 that the classification performance does not vary substantially with the feature selection method, we follow the experiments considering Method B for detailed analysis in Subsection 3.2.4.1.

3.2.4.1 Detailed Experiments with Method B

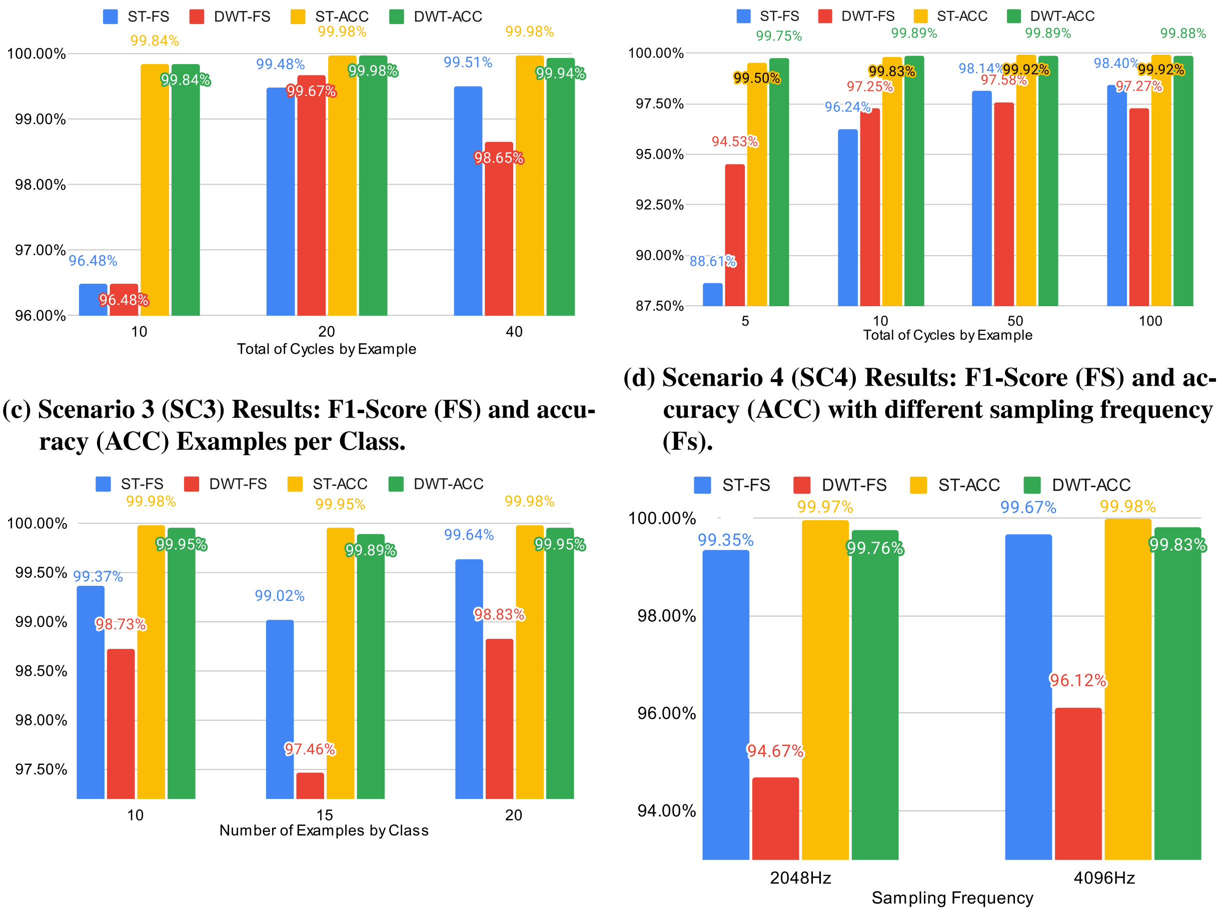
Table 7 shows detailed results obtained with Method B. We obtain these results considering the 5-fold classification models, with different examples for training and validation. We take the average of these five folds for each case and scenario. We also calculate accuracy for comparison purposes.

Table 7 – Macro Accuracies and F1-Scores for all Scenarios and cases with Method B.

			Accuracy		**F1-Score**	
Scn.	**Case**	**Description**	**ST**	**DWT**	**ST**	**DWT**
SC1	1	5 cycles	99.84%	99.84%	96.48%	96.48%
	2	10 cycles	99.98%	99.98%	99.48%	99.67%
	3	20 cycles	**99.98%**	99.94%	**99.51%**	98.65%
SC2	1	5 cycles	99.50%	99.75%	88.61%	94.53%
	2	10 cycles	99.83%	99.89%	96.24%	97.25%
	3	50 cycles	**99.92%**	99.89%	**98.14%**	97.58%
	4	100 cycles	**99.92%**	99.88%	**98.40%**	97.27%
SC3	1	10 examples	**99.98%**	99.95%	99.37%	98.73%
	2	15 examples	**99.95%**	99.89%	**99.02%**	97.46%
	3	20 examples	**99.98%**	99.95%	99.64%	98.83%
SC4	1	2048Hz	**99.97%**	99.76%	**99.35%**	94.67%
	2	4096Hz	**99.98%**	99.83%	**99.67%**	96.12%
SC5	1	6s samples	**99.95%**	99.91%	**98.86%**	98.02%

Figure 15(a) shows that with the smaller cycle window (10 cycles), both DWT and ST have the same accuracy (99.84%) and F1-Score (96.48%). For 20 cycles window, both methods present the same accuracy (99.98%), but DWT shows slightly better F1-Score than ST (99.66% vs. 99.48%). On the other hand, with 40 cycles, the ST method shows marginally better accuracy than DWT (99.98% vs. 99.94%). In this case, ST presents a significantly better global F1-Score than DWT (99.51% vs. 98.55%).

Figure 15 – Macro Accuracy (ACC) and Macro F1-Score (FS) results for different experimental Scenarios using Scattering Transform with Method B (ST) and Discrete Wavelet Transform (DWT) feature extractors.



(a) Scenario 1 (SC1) Results: F1-Score (FS) and accuracy (ACC) with different number of cycles of the time window surrounding the *turn-on* event.

(b) Scenario 2 (SC2) Results: F1-Score (FS) and accuracy (ACC) with different number of cycles per example *after* the *turn-on* event.

(c) Scenario 3 (SC3) Results: F1-Score (FS) and accuracy (ACC) Examples per Class.

(d) Scenario 4 (SC4) Results: F1-Score (FS) and accuracy (ACC) with different sampling frequency (Fs).

Source: Own authorship.

Similar to SC1 results, in scenario 2 (Figure 15(b)), DWT presents slightly better F1-Scores and Accuracy than ST for five cycles: F1-Score 94.53% vs. 88.61% and Accuracy 99.75% vs. 99.50%. For 10 cycles per sample: F1-Score 97.25% vs. 96.24% and Accuracy 99.89% vs. 99.83%. For the cases with 50 and 100 cycles per sample, ST outperforms DWT in both F1-Score and Accuracy. F1-Score 98.14% vs. 97.58% and Accuracy 99.92% vs. 99.89% for 50 cycles, and F1-Score 98.40% vs. 97.27% and Accuracy 99.92% vs. 99.88% for 100 cycles per sample.

Figure 15(c) shows that ST has better performance than DWT for all cases of Scenario 3. The macro F1-Score is 0.65% better with ten examples per class, 1.6% with 15 examples, and 0.82% with 20 samples per class. The improvements in accuracy are smaller than the improvements in F1-Score: 0.03% in 10 examples per class, 0.06% in 15 examples, and 0.03% in 20 examples.

ST presents better F1-Score and Accuracy than DWT for the 2 cases of Scenario 4, as shown in Figure 15(d). This indicates that ST deals better with sub-sampling conditions than DWT for COOLL dataset (considering Ensemble classifier). Figure 15(d) also shows that both ST and DWT increase their performance metrics (F1-Score and Accuracy) when the sampling frequency rises. From 99.35% to 99.67% and from 99.97% to 99.98% for ST (F1-Score and Accuracy, respectively), and from 94.67% to 96.12% of F1-Score and from 99.76% to 99.83% of Accuracy for DWT.

3.3 COMPARISONS WITH STATE-OF-THE-ART APPROACHES

Other state-of-the-art works presented feature extraction strategies applied to COOLL dataset. Table 8 shows the comparison of the classification results metrics between those approaches and the proposed method. For all values of F1-Score and Accuracy in Table 8, the best scenario for each method is considered.

Table 8 – Comparison between literature approaches and the proposed method.

Reference	Method	F1-Score [%]	Accuracy [%]
(RENAUX *et al.*, 2018)	Traditional Wavelets	-	92.00%
(MULINARI *et al.*, 2019)	Hybrid V-I Trajectory	-	99.44%
(ANCELMO *et al.*, 2019)	Prony	-	98.00%
Proposed	Scattering Transform	**99.51%**	**99.98%**

As shown in Table 8, the proposed method presents, in terms of accuracy, an improvement of 8.67% in relation to (RENAUX *et al.*, 2018), 0.54% to (MULINARI *et al.*, 2019), and 2.02% in relation to (ANCELMO *et al.*, 2019).

3.4 CONCLUSIONS OF THIS CHAPTER

The classification results of NILM signal feature extraction and selection methods based on Deep CNN depend on the amount of data available for training. We propose a classification framework using a Scattering Transform-based feature extraction method applied to NILM. In this framework, the weights of the convolutional network are not trained but are analytically calculated using wavelets. We included variations in the dataset properties in the framework to test the performance of our proposal against reductions in the amount of training data. We vary the signal length, the number of examples per class, and sampling frequency. The main contributions of this chapter are: (i) apply the Scattering transform to improve NILM state-of-the-art classification results; (ii) validate these improvements under dataset properties variations; (iii) visual low-dimensional separability analysis of ST-based feature extraction techniques; and (iv) proposal and evaluation of six different feature selectors based on ST applied to NILM signals.

In Section 3.2.3, we presented a reduced-order representation for the features extracted from all examples of the COOLL dataset, both with the proposed method and with the baseline method (traditional wavelet). These 2D representations, obtained with the t-SNE (MAATEN; HINTON, 2008) method, showed that the intersection regions between appliances are smaller in the proposed method than in the baseline for all tested feature selectors. This verification indicates that the separability of the proposed method is better than the baseline for all tested selectors. In addition, all the proposed feature selection methods showed similar visual separability characteristics, which indicates robustness concerning the number of features chosen. Finally, proposed ST-based feature selection methods separated in the 2D plane the appliances of the same category (different brands, for example), which is desirable in the NILM classification.

We tested the six proposed feature selection methods with the five scenarios in the table 5, using the structure of Figure 10. The average of all macro F1-Scores from each case and scenario for each proposed feature selection method did not vary significantly depending on the different proposed feature selection methods. This result corroborates the separability inspection of Section 3.2.3. The results presented in Table 6 showed that the F1-Score of proposed feature selectors surpasses the results of the wavelet baseline for all scenarios. Furthermore, the results from Table 6 showed that the appliance turn-on event does not significantly interfere with the classification results. This last verification is critical considering that many methods in the literature have such dependence.

We perform detailed experiments with Method B, applied to five different scenarios and several cases per scenario. From SC1, the proposed method outperformed the DWT baseline for cases with more cycles per example (20 and 40 for SC1; 50 and 100 for SC2). The DWT baseline and ST are architectures that use multi-resolution filter banks and, therefore, intrinsically use downsampling. In our case, the ST has more subbands than the DWT, so there are more downsampling operations for the ST. As a consequence of this, for examples with fewer cycles, DWT outperforms ST. This trend reverses when the sampling window is larger. The experiments with Scenario SC3 showed that the proposed method outperformed DWT when reducing the number of examples per class for all evaluated cases. We also conclude, with SC4 results, that Scattering Transform presented better classification metrics for downsampling conditions in the dataset. Finally, the proposed method resulted in better macro accuracy when compared to state-of-the-art methods of feature extraction for NILM in the literature, as shown in Table 8.

4 CLASSIFICATION WITH SCATTERING TRANSFORM FOR NON-INTRUSIVE LOAD MONITORING

In Chapter 3, we presented a framework that investigated the application of ST as a feature extractor for NILM with single appliances. The results were better than state-of-the-art methods for the COOLL dataset. We showed that ST has better separation (in reduced order 2D plane) than the traditional wavelet transforms for all proposed feature selection proposals. This chapter extends the approach of chapter 3 to aggregated NILM signals. In this chapter, we are interested in evaluating ST for the classification of aggregated loads, applying different signal regions (steady-state, transitory, and both regions) to the feature extractor.

The results showed that the proposed ST-Based method is more robust in terms of signal length, sampling frequency, and event location. Furthermore, the aggregated and single electrical loads results showed the best generalization capability of the proposed method compared to baselines.

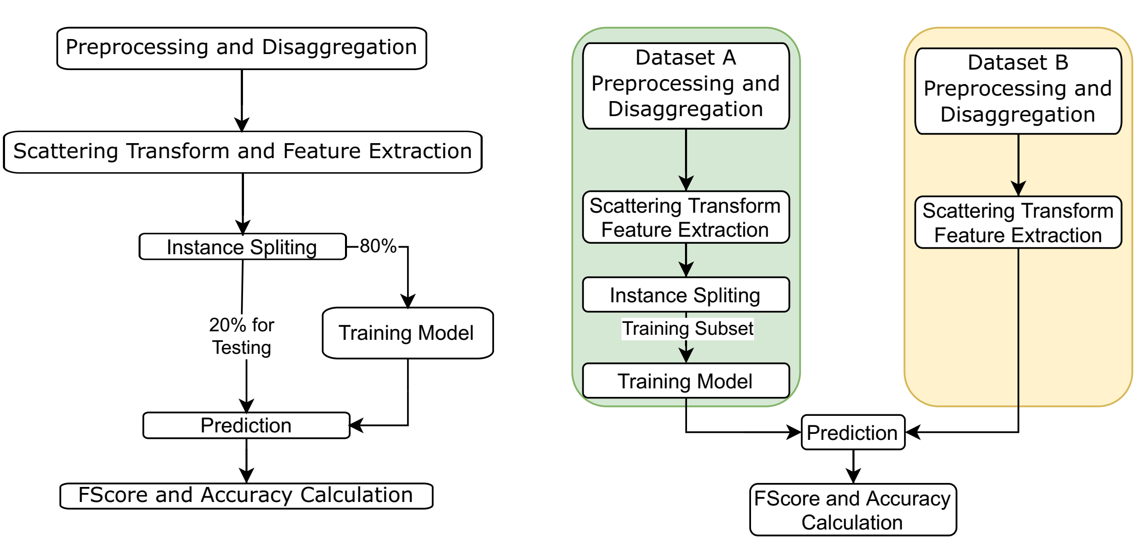
4.1 PROPOSED CLASSIFICATION STRATEGY

We employ two different approaches and present the results and comparisons with related works. The feature extraction, the training of the classifiers, and evaluation for the test set are carried out with the same dataset in the first approach, which is the most conventional analysis (RUANO *et al.*, 2019). In the second approach, the feature extraction and the training are conducted in one dataset, and the prediction is evaluated in a different dataset. In this second case, we train models with single and three loads and evaluate the classification performance of the model with two, three, and eight loads. Therefore, it is possible to analyze the generalization of the classifier, as proposed in Lazzaretti *et al.* (2020). Figure 16 shows the two approaches, detailed as follows.

4.1.1 LIT Syntetic Dataset

The first dataset used in this work is the LIT Synthetic (LIT-SYN) (RENAUX *et al.*, 2020). This dataset is a subset of the full LIT-Dataset, and it refers to acquisitions collected from a bench, in which actual loads are connected, but the network loads' switching instant is controlled. This set contains 1 664 waveform acquisitions sampled at 15 360Hz, with precisely

Figure 16 – Proposed Framework Experimental Setup


(a) With the same datasets.
Preprocessing and Disaggregation
Scattering Transform and Feature Extraction
Instance Spliting
80%
20% for Testing
Training Model
Prediction
FScore and Accuracy Calculation
(b) With different datasets.
Dataset A Preprocessing and Disaggregation
Scattering Transform Feature Extraction
Instance Spliting
Training Subset
Training Model
Dataset B Preprocessing and Disaggregation
Scattering Transform Feature Extraction
Prediction
FScore and Accuracy Calculation

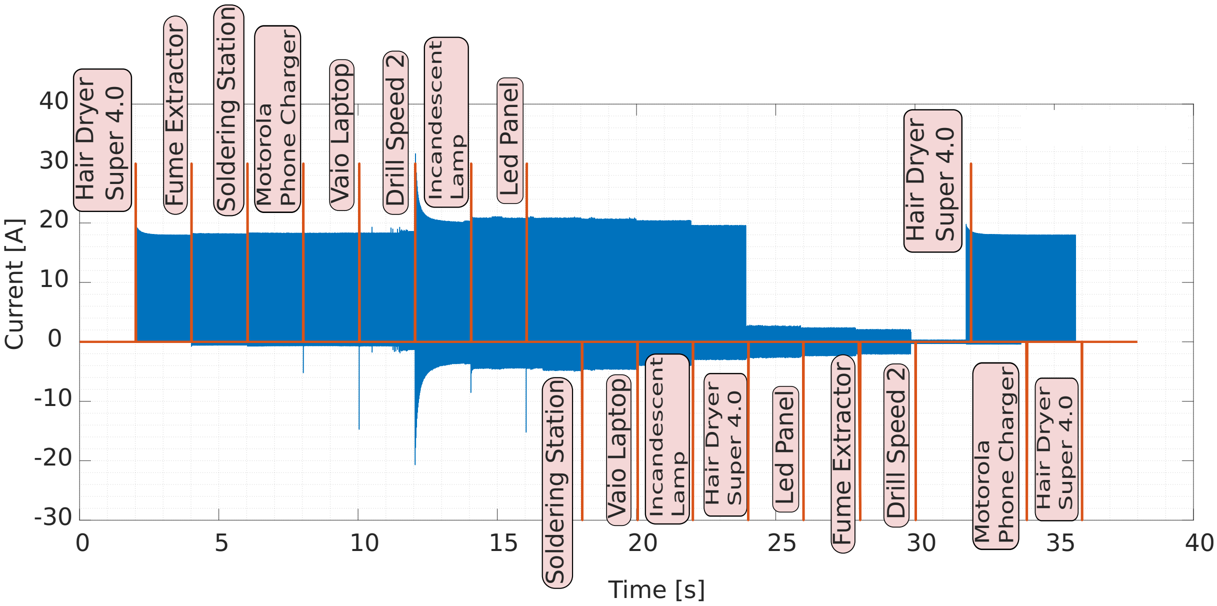
Source: Own authorship.

annotated (< 5ms) switching events of 26 classes (loads). The aggregated AC grid voltage and current acquisitions are monitored for periods of up to 40 seconds. The devices used in this subset are summarized in Table 9.

Table 9 – Appliances in LIT Synthetic Subset (LIT-SYN)

ID	Device	Rated Power (W)
1	Microwave Standby	4.5
2	LED Lamp	6
3	CRT Monitor	10
4	LED Panel	13
5	Fume Extractor	23
6	LED Monitor	26
7	Cell phone charger Asus	38
8	Soldering station	40
9	Cell phone charger Motorola	-
10	Laptop Lenovo	70
11	Fan	80
12	Resistor	80
13	Laptop Vaio	90
14	Incandescent Lamp	100
15	Drill Speed. 1	165
16	Drill Speed. 2	350
17	Oil heater power 1	520
18	Oil heater power 2	750
19	Microwave On	950
20	Air heater Nilko	1 120
21	Hair dryer Eleganza - Speed 1	365
22	Hair dryer Eleganza - Speed 2	500
23	Hair dryer Super 4.0 - Speed 1 - Heater 1	660
24	Hair dryer Super 4.0 - Speed 1 - Heater 2	1 120
25	Hair dryer Parlux - Speed 1 - Heater 1	660
26	Hair dryer Parlux - Speed. 2 - Heater 1	885

Figure 17 – Example of a waveform from the LIT-SYN-8 subset, highlighting the times when the turn-on and turn-off events occurred.



Source: Own authorship.

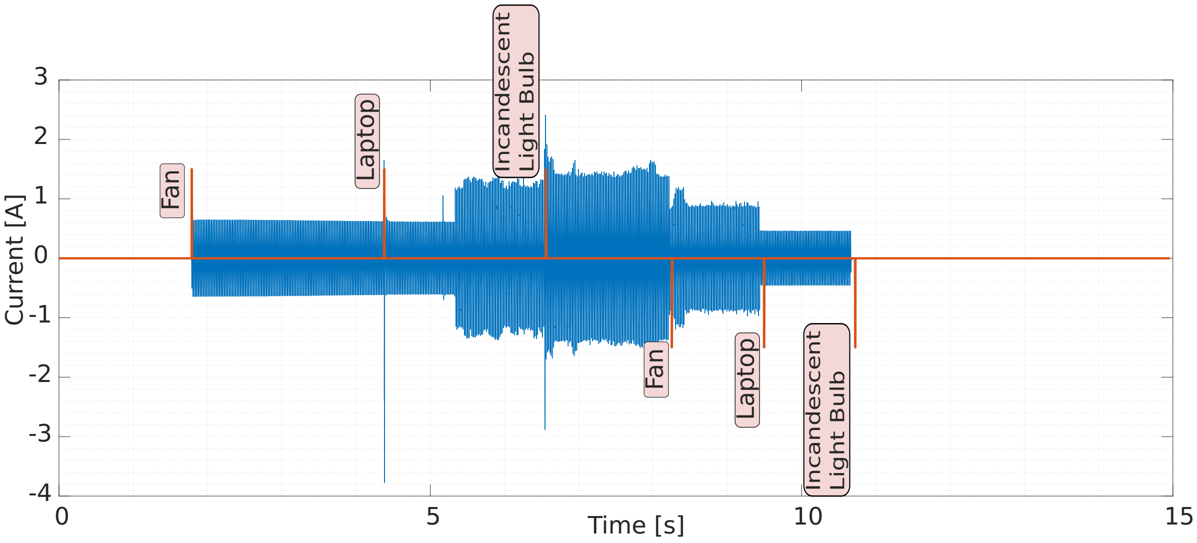
LIT-SYN contains different subsets with different multiple aggregated loads, starting with isolated loads (LIT-SYN-1), two combined loads (LIT-SYN-2), three (LIT-SYN-3), and eight (LIT-SYN-8). The authors of the dataset also make available a subset with all combinations, defined as LIT-SYN-Total.

We show in figure 17 an example of the electric current curve of the LYT-SYN-8 subset, with eight aggregated signals from different household appliances. The red segments in the figure 17 indicate the moments at which the turn-on and turn-off events occurred, annotated in the dataset. When the red segment is greater than zero, it is a turn-on event, and when it is less than zero, it is a turn-off event. To facilitate the visualization, we indicate the name of the class next to each segment. In the same figure, the blue curve represents the aggregated current.

4.1.2 PLAID Dataset

The second subset is the Plug-Load Appliance Identification Dataset (PLAID), proposed in Medico *et al.* (2020). It is a high-frequency (30kHz) public dataset, and it has submetered data for 17 electrical appliances, totaling 1 876 measurements, in 65 different homes. PLAID also has a subset of aggregated data, composed of 1 314 waveforms of 13 electrical appliances (classes) present in a test laboratory, which is used here. The aggregated measurements are obtained in combinations of two or three electrical loads, with annotations of connection and disconnection

Figure 18 – Example of a waveform from the PLAID dataset, highlighting the times when the turn-on and turn-off events occurred.



Source: Own authorship.

events for each load. Table 10 shows the set of aggregated electrical loads from the PLAID dataset. We use the aggregate subset of measurements in this work.

Table 10 – Appliances in aggregated subset PLAID

ID	Device	Rated Power (W)
1	Compact Fluorescent Lamp	13
2	Fridge	Not available
3	Fridge Defroster	Not available
4	Air Conditioner	740
5	Laptop	Not available
6	Vacuum Cleaner	Not available
7	Fan	38
8	Incandescent Light Bulb	100
9	Blender	Not available
10	Coffee Maker	975
11	Water Kettle	1500
12	Hair Iron	Not available
13	Iron Solder	60

We show in figure 18 an example of the electric current curve of the PLAID dataset, with three aggregated signals from different household appliances. The red segments in the figure 18 indicate the moments at which the turn-on and turn-off events occurred, annotated in the dataset. When the red segment is greater than zero, it is a turn-on event, and when it is less than zero, it is a turn-off event. To facilitate the visualization, we indicate the name of the class next to each segment. In the same figure, the blue curve represents the aggregated current.

4.1.3 Preprocessing and Disaggregation

For the preprocessing and disaggregation, different scenarios are considered, as summarized in Table 11. Each chosen scenario has one particular time region of the electrical NILM signal, detailed as follows.

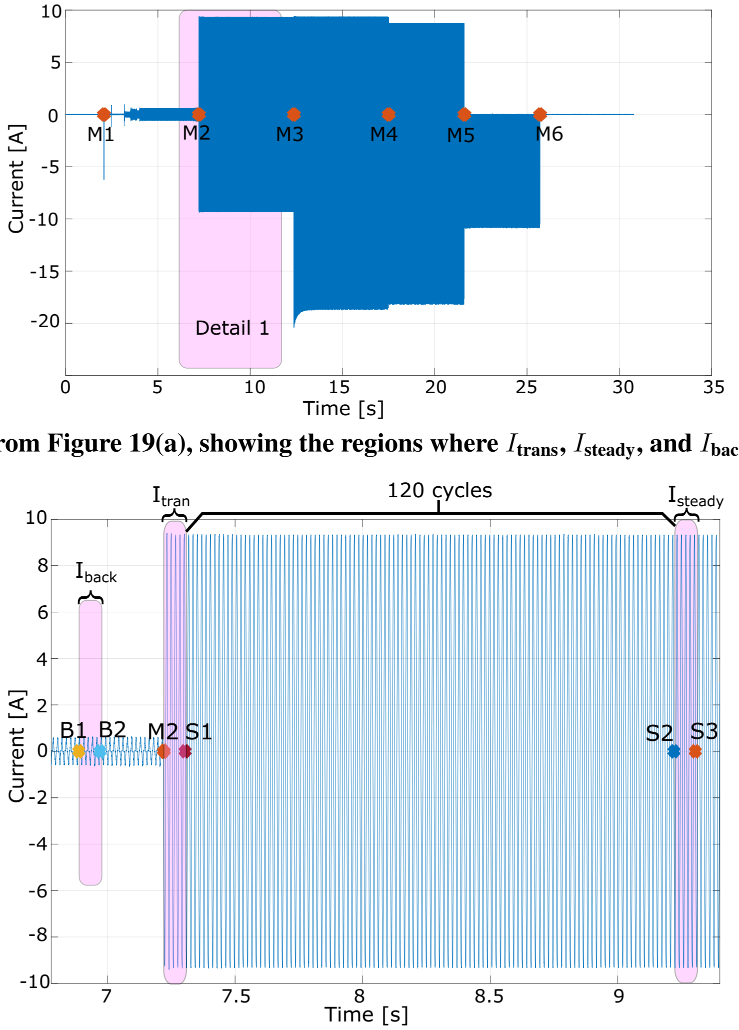
Table 11 – Experimental Scenarios Description.

Scenario	n_{back}	n_{cycles}	Selected Region	n_{feat}
A	30	20	SS + T	118
B	30	20	T	59
C	30	20	SS	59
D	20	5	SS + T	86
E	20	5	T	43
F	20	10	SS + T	102
G	20	10	T	51
H	20	10	SS	51
I	40	40	SS + T	134
J	40	40	T	67
K	40	40	SS	67

The selected regions may be in *Steady State* (SS), *Transient* (T), or both states. Scenarios A, D, F, and I consider SS and T regions by concatenating the energy coefficients from those regions. Scenarios B, E, G, and J consider only transient instants. Scenarios C, H, and K consider steady-state regions. Each scenario assesses a different number of cycles, n_{back}, before the turn-on event, considered for the disaggregation procedure. With those scenarios, the authors intend to verify the robustness of the proposed feature extraction strategy for variations applied to the ST input signals. The region selected for each scenario has n_{cycles} cycles of the aggregated signal. The n_{feat} will be detailed in Subsection 4.1.4.

To exemplify, we present how we construct *Scenario D* and *E*. In Figure 19(a), the first three red markers (M1-M3) indicate the turn-on events for a sample extracted from the LIT-SYN-3 subset. The last three red markers (M4-M6) indicate the turn-off events. First, our algorithm defines a point (B1), with $n_{\text{back}} = 20$ cycles before M2. Then, we take a window (I_{back}) from B1 to B2, with $n_{\text{cycles}} = 5$ cycles. In the sequence, we compute two signal windows after M2. The first is the transient current region (I_{tran}), and the other is the steady-state current window (I_{steady}). The I_{tran} is between the first zero-crossing after M2 and $n_{\text{cycles}} = 5$ after its zero-crossing, reaching S1. The I_{steady} is located between 120 cycles after P1 (point S2) and S3, that is located $n_{\text{cycles}} = 5$ cycles after S2. The windows I_{steady} and I_{tran} are then used to compute the disaggregated data, I^d_{tran} and I^d_{steady}. As in Mulinari *et al.* (2019), the transient and steady-state disaggregated data are defined as $I^d_{\text{tran}} = I_{\text{tran}} - I_{\text{back}}$ and $I^d_{\text{steady}} = I_{\text{steady}} - I_{\text{back}}$, respectively. For

Figure 19 – Example of the preprocessing and disaggregation.



(a) An original waveform of LIT-SYN3. The markers in red are the load turn-on or turn-off annotations (both considered previously known in our case).

(b) Detail 1 from Figure 19(a), showing the regions where I_{trans}, I_{steady}, and I_{back} are located.

Source: Own authorship.

Scenario D, we concatenate both signal windows I^d_{tran} and I_{steady}. For *Scenario E*, we directly apply the feature extraction on I^d_{tran}.

4.1.4 Feature Extraction

We use the *Wavelet Scattering* library, from Matlab® R2021a to implement the ST. We consider different scenarios for the extraction and selection of features from the ST of the electrical signal. The number of features (n_{feats}) for each scenario is the number of wavelets filters (J) for the first layer of Scattering Transform (order 1). This number depends on the quality factors of the first-order filter bank (Q) and the total number of samples for each input signal (T). With Algorithm 1 we compute the approximated number of Scattering Coefficients (n^k_j), resulting of j-th wavelet filter convolution for the scenario $k \in \{A,B,\ldots,K\}$.

Algorithm 1 – Number of Scattering Coefficients

```
Input: $Q$ and probability parameter ($PP$).
Output: $n_j^k$
for each scenario $k \in \{A,B,\ldots,K\}$ do
    Compute T, the number of samples per examples.
    Compute $cv$, the gaussian critical value for the probability parameter $PP$, as detailed in Bonamente (2022).
    Compute $\sigma_{\Phi}^{f} = \frac{T}{2 \times cv}$, the frequency standard deviation for the scaling function.
    Compute the frequency support for the scaling function ($\Phi_{support}$), as discussed in Mallat (2012).
    Compute the highest wavelet center frequency $\omega_{\Psi} = \frac{2\pi}{1+2^{\frac{1}{Q}}}$.
    Compute the frequency standard deviation for the wavelets $\sigma_{\Psi}^{f} = \omega_{\Psi} \frac{1-2^{-\frac{1}{Q}}}{1+2^{-\frac{1}{Q}}} \frac{1}{\sqrt{\frac{3}{10} ln(10)}}$.
    $J \leftarrow \text{floor}\left\{Q\left[log_2\left(\sigma_{\Psi}^{f}\right) - log_2\left(\sigma_{\Phi}^{f}\right)\right]\right\}$.
    Compute $\Psi_{tsupport}^{f}$, the time support of the j-th first layer wavelet filter.
    $\text{res} \leftarrow \min\left[log_2\left(\frac{2\pi}{\Psi_{tsupport}^{f}}\right), 0\right]$.
    $n_j^k \leftarrow \text{round}\left[\frac{T}{2^{\text{res}}}\right]$.
    return $n_j^k$
end for
```

4.1.4.1 Feature calculation from Scattering Coefficients

We verified in chapter 3 that F1-Score does not vary significantly when using the different proposed ST feature selection methods. Similar to when we performed the detailed experiments in Subsection 3.2.4.1, here we also use Method B (Energies of the first-order Scattering Coefficients, from Eq. 11) for selecting the features. We apply this method to each scenario presented in Table 11.

Finally, it is noteworthy that both rigid translation and time-warping variabilities in the NILM signals need to be mitigated to represent the signals. The invariance to translation is desirable to NILM signals as the event detection is not always accurate or available. Both Deep CNN and ST have these properties, but in the case of ST: (i) it is possible to optimize filters and pooling non-linearities; (ii) the multiple layers are analytically determined; and (iii) there is no training stage for the filters coefficients (BRUNA; MALLAT, 2012).

4.1.5 Classification

The electrical appliance classification task follows the steps described in Figure 16a and 16b. The training and test sets are separated from the feature matrix and the vector of labels. 80% of the instances were used to train the classifier and select parameters using five-fold cross-validation from the complete matrix. For testing, the remaining 20% were evaluated. It is noteworthy that all the features are normalized in $[-1,1]$.

We train five different classification methods, typically used in NILM approaches (LAZZARETTI *et al.*, 2020): K-Nearest Neighbors (k-NN), Ensemble Method (ENS), Support Vector Machines (SVMs), Decision Trees (DTs), and Linear Discriminant Analysis (LDA). We use Matlab® to implement all these classifiers.

4.2 RESULTS

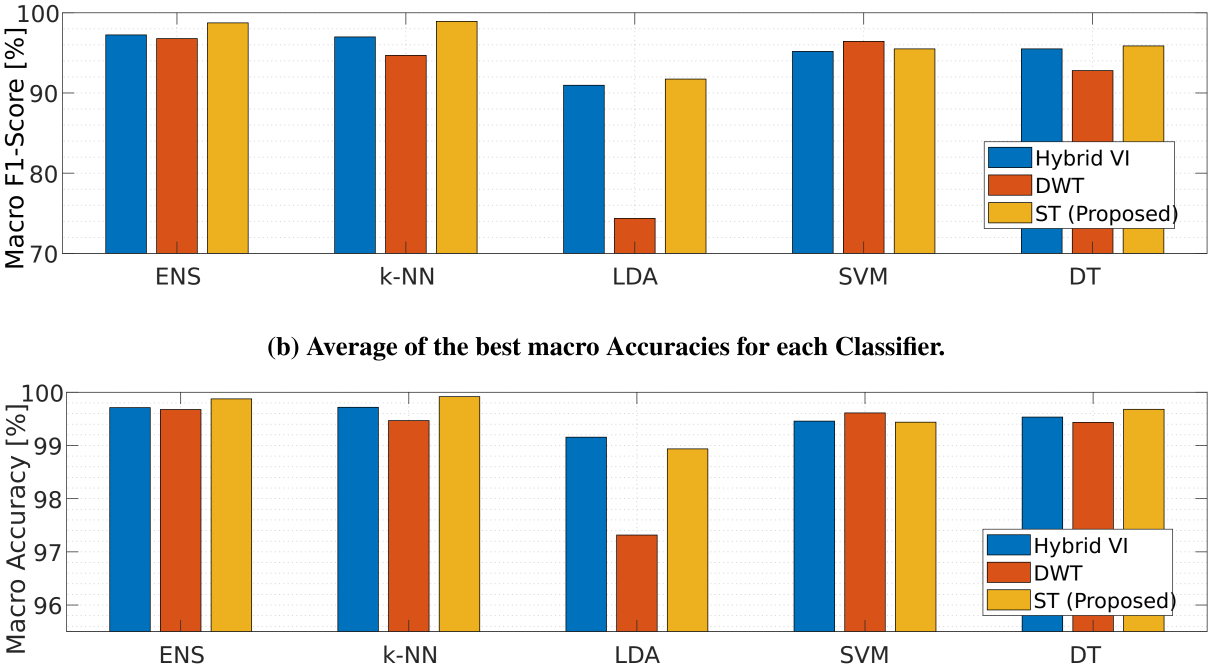
Initially, in Subsection 4.2.1, we perform the two approaches presented in Figure 16. In the first approach, the experiments consider the same subsets of LIT-SYN for training and testing. Subsequently, as shown in Figure 16b, the experiments are conducted varying the number of loads for training and testing (different subsets), aiming at evaluating different scenarios and generalizations for the model. Since the PLAID dataset does not contain subsets with a different number of aggregated loads, in Subsection 4.2.2, we present only the results for the approach presented in Figure 16a. For all the experiments in Subsection 4.2.1 and 4.2.2, we include other two baseline methods for comparisons: (i) V-I trajectory (MULINARI *et al.*, 2019) and (ii) Discrete Wavelet Transform (DWT) (CHANG *et al.*, 2014). Those methods are selected since they are part of the state-of-the-art results presented in Lazzaretti *et al.* (2020) and both extract transient and steady-state features, allowing direct comparisons for all experimental scenarios described in Table 11. In Section 4.3, a discussion and a comparison with deep learning-based and state-of-the-art results for LIT-SYN and PLAID is shown to present further the positive aspects of the ST feature extraction and classification.

4.2.1 LIT-SYN Dataset

4.2.1.1 Results using the same subset for training and testing

We performed the strategy proposed in Figure 16-a, training the five classification models for each scenario of Table 11. We show in Figure 20 the comparison of the macro F1-Scores and Accuracies among the proposed method and the baseline methods. The results were obtained as follows: (i) for each scenario and each LIT-SYN subset, we obtained the average of the five-fold macro F1-Score and Accuracy for five classification models; (ii) for each LIT-SYN

Figure 20 – Average of the best macro F1-Scores and Accuracies for each classifier.



(a) Average of the best macro F1-Scores for each Classifier.

(b) Average of the best macro Accuracies for each Classifier.

Source: Own authorship.

subset, we choose the best scenario results for each classifier; and (iii) we performed the averages of the best macro F1-Scores and Accuracies among the LIT-SYN subsets.

From Figure 20, we can conclude that the proposed method presented better performances than baseline methods for ENS, k-NN, and DTs. Additionally, ST reaches the two highest macro F1-Score and Accuracy averages, with 98.74% (F1-Score) and 99.87% (Accuracy) for ENS, and 98.93% (F1-Score) and 99.91% (Accuracy) for k-NN.

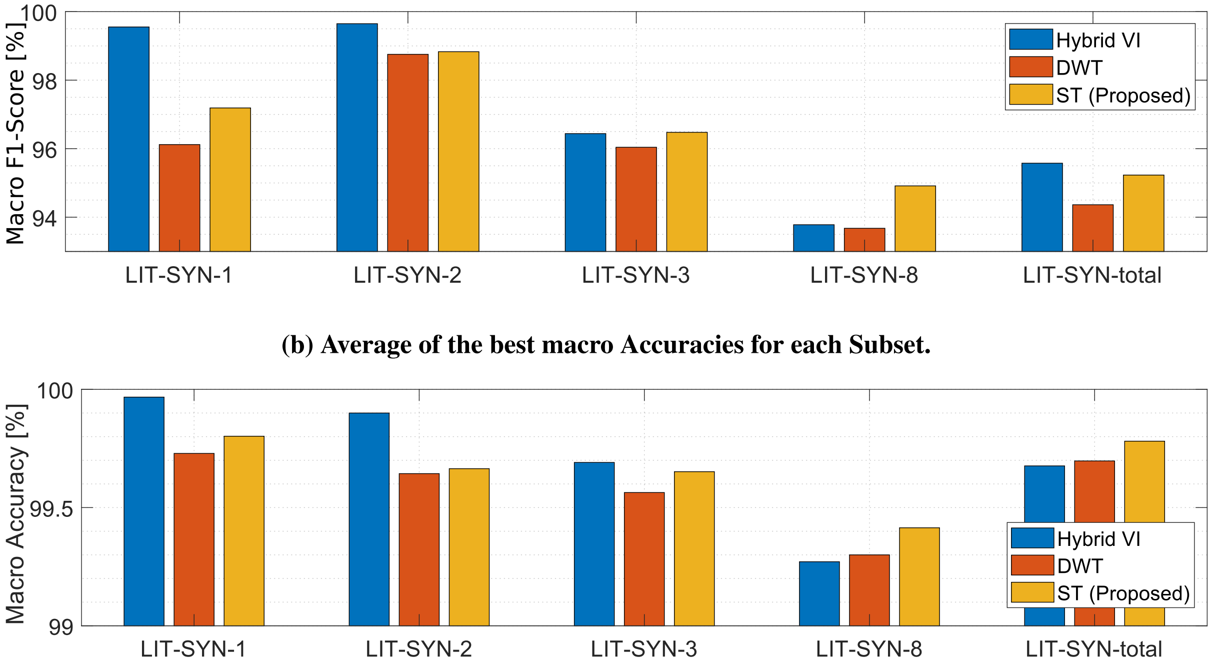
Based on the results for the best classifier (ENS), for each subset of the LIT dataset, and considering all the scenarios of Table 11, we calculated the average macro F1-Score and Accuracy. The results are shown in Figure 21.

To verify the influence of the scenarios presented in Table 11, Figure 22(b) shows macro accuracies and F1-Scores for the LIT-SYN-8.

4.2.1.2 Results using different subsets for training and testing

As presented in Lazzaretti *et al.* (2020), the actual scenario of data collection in NILM systems may involve the acquisition of single or multiple loads, depending on the availability and difficulty in acquiring samples. Hence, to verify in terms of classification accuracy how much

Figure 21 – Average of best macro F1-Scores and Accuracies for each LIT subset. Observe that proposed method becomes better than baselines as the number of aggregated loads increases. Each bar represents the average of all scenarios for the correspondent LIT-SYN subset.

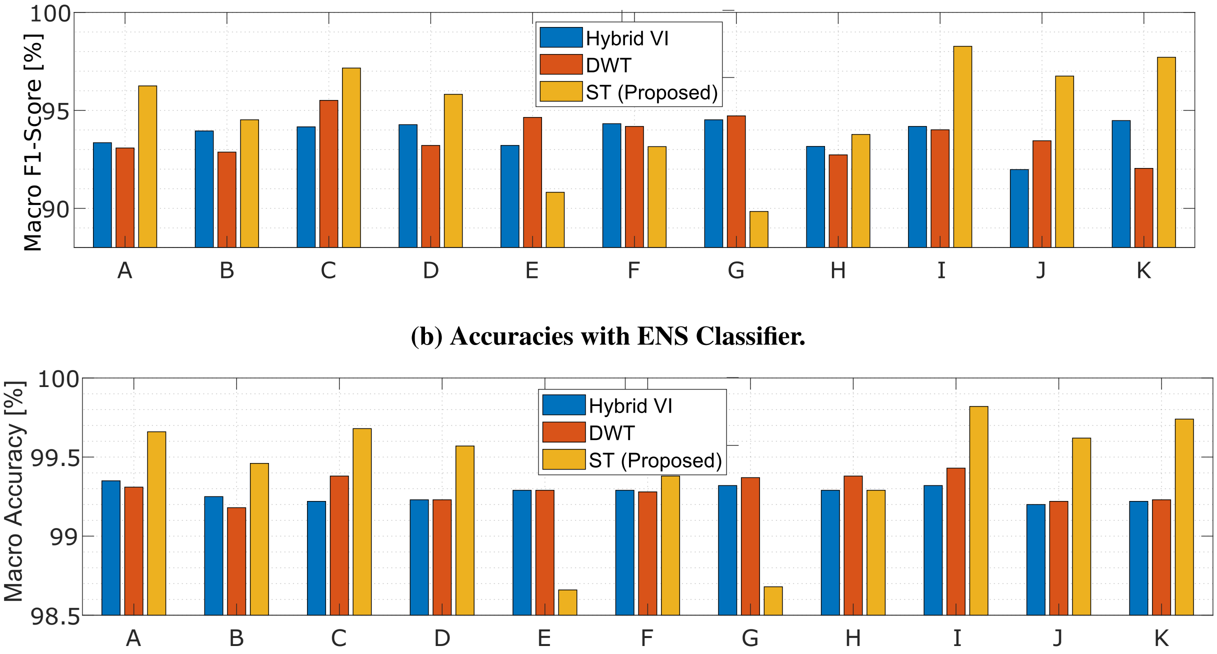


(a) Average of the best macro F1-Scores for each Subset.

(b) Average of the best macro Accuracies for each Subset.

Source: Own authorship.

Figure 22 – Best accuracies and F1-Scores for each scenario with subset LIT-SYN-8.



(a) F1-Scores for each scenario with ENS Classifier.

(b) Accuracies with ENS Classifier.

Source: Own authorship.

the typical signature of an electric load is maintained as multiple loads are added, we present the generalization analysis in this Subsection.

We first extract features with ST considering the best accuracy and F1-Score performance previously obtained with the same subsets for training and testing. For comparison purposes, classification models are also trained using the V-I and DWT methods for the same training and test subsets, with the best classifiers for each case. The average macro accuracy obtained with different subsets for training and test is shown in Table 12.

Table 12 – Macro Accuracy for different subsets of LIT-SYN

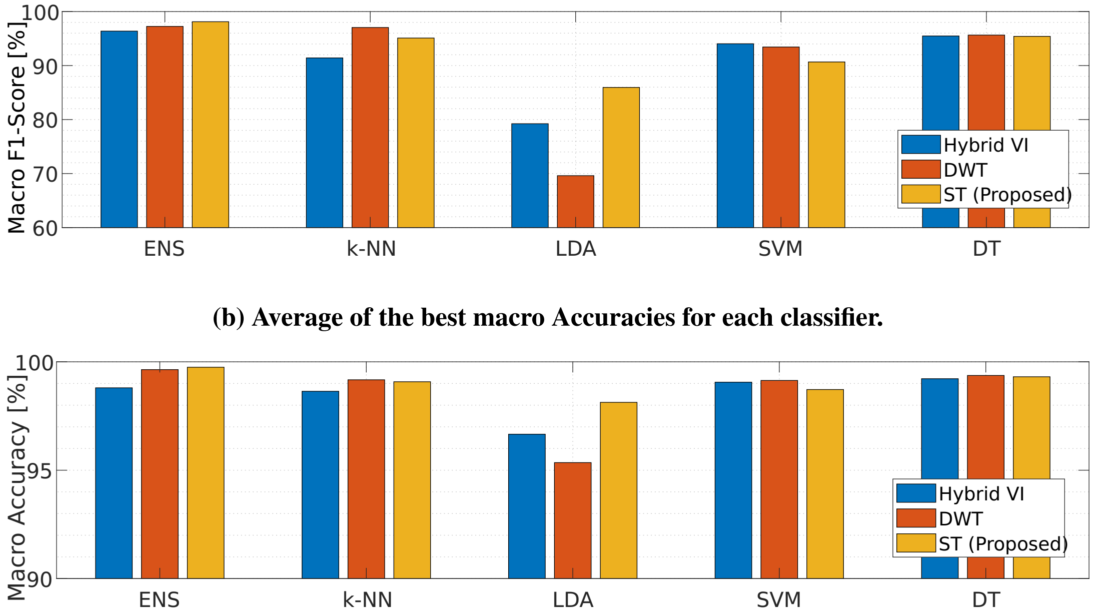
Training Dataset	Testing Dataset	Accuracy [%]		
		ST	DWT	V-I
LIT-SYN-1	LIT-SYN-2	86.50	**92.10**	89.55
LIT-SYN-1	LIT-SYN-3	**73.60**	62.14	67.98
LIT-SYN-1	LIT-SYN-8	**58.68**	41.34	54.72
LIT-SYN-T	LIT-SYN-1	**99.88**	99.16	99.76
LIT-SYN-T	LIT-SYN-2	**99.63**	99.41	99.52
LIT-SYN-T	LIT-SYN-3	**99.56**	99.02	98.42
LIT-SYN-T	LIT-SYN-8	**99.42**	98.90	97.75

4.2.2 PLAID Dataset

For the PLAID dataset, experiments were performed with scenarios A-H from Table 11. The I-K were not applied because the turn-on events in that dataset are located less than $n_{back} = 40$ cycles from the reference. The experiments for PLAID were conducted according to the diagram in Figure 16a. The best results for accuracy and F1-Score for each classifier are shown in Figure 23.

Figure 23 – Average of best macro F1-Scores and Accuracies for each classifier for PLAID dataset.

(a) Average of the best macro F1-Scores for each classifier.



(b) Average of the best macro Accuracies for each classifier.

Source: Own authorship.

4.3 DISCUSSION AND COMPARISON WITH RELATED WORKS

In this Section, we discuss the results presented in Section 4.2 sequentially, making comparisons with baseline literature methods, as suggested in Saura *et al.* (2021).

As one can observe in Figure 21, the proposed method has similar macro F1-Score and Accuracy (but slightly inferior to DWT and Hybrid VI) for subsets with a smaller number of aggregated loads (up to 3). For the cases with more aggregated loads (LIT-SYN-8 and LIT-SYN-total), the ST overcomes the Accuracy of DWT and V-I. The macro F1-Score also surpasses baseline F1-Scores for LIT-SYN-8. These results are significant since, in real-world NILM applications, aggregated scenarios are more frequent.

As one can observe in Figure 22(b), scenarios *E* and *G* present lower accuracies for the ST method, indicating that a lower number of n_{cycles} for transient regions may decrease the performance. However, this is not a relevant variation ($< 2\%$ for accuracy), and the ST presents higher overall results for most cases, both for accuracy and F1-Score. This reinforces the robustness of the ST for different waveform parameters, such as signal length and event location, making the ST more tolerant to the reference point given by a previous detection method.

When training and testing with different subsets, the results obtained in Table 12 show superior results for the proposed method compared to the baselines, for the LIT-SYN-3 and

LIT-SYN-8 subsets – 73.60% and 58.68%, respectively. Using LIT-SYN-T as a training subset classifier, the ST presented better accuracy than baselines in all test subsets (LIT-SYN-1, LIT-SYN-2, LIT-SYN-3, and LIT-SYN-8). It stands out the average accuracy of 99.88% achieved when the subset LIT-SYN-1 was used as the test subset. As one can observe, the ST presents better classification results for testing cases with different datasets for single and multiple loads available during training. In other words, the proposed feature extraction method has a more powerful generalization capability than baselines.

The results with PLAID dataset (Figure 23), demonstrate that ST presented both accuracy and F1-Score higher than the baselines for the ENS and the LDA classifiers. For ENS, the proposed method reached an accuracy of 99.75% (against 98.80% and 99.64%) and F1-Score of 98.12% (against 96.38% and 97.15%), for scenario *F*. For the LDA classifier, ST reached the best metrics among the baselines methods with scenario *B* (Accuracy= 98.13% and F1-Score= 85.95%). The accuracy of the ST surpassed V-I method for DT classifiers with scenario *G* and k-NN with scenario *F*. Also, the accuracy of the proposed method exceeded the accuracy of the DWT method for the SVM classifier using scenario *A*. The results in Figure 23 show that: (i) ST has both F1-Score and Accuracy superior or in the same range as the baselines for all applied classifiers; (ii) ST has the best-of-all results for the PLAID dataset, with ENS classifier and scenario *F*.

Finally, the classification results obtained with the proposed method are compared with the state-of-the-art results on NILM classification in Table 13.

Table 13 – Comparison with State-of-the-art Approaches.

Method	Dataset	Accuracy (%)	F1-Score (%)
Proposed Method	PLAID	**99.75**	**98.12**
Houidi *et al.* (2020)	PLAID	97.70	97.50
Baets *et al.* (2018)	PLAID	-	77.60
Proposed Method	LIT	**99.87**	**97.42**
Lazzaretti *et al.* (2020)	LIT	99.80	-
Mukaroh *et al.* (2020)	LIT	92.0	-

With PLAID dataset, the proposed method presented an F1-Score 0.64% greater than Houidi *et al.* (2020) and 26.44% greater than Baets *et al.* (2018). The accuracy of the proposed method with the PLAID dataset was 2.10% greater than Houidi *et al.* (2020). With the LIT-SYN dataset, our approach showed accuracy approximately equivalent to Lazzaretti *et al.* (2020) and 8.48% greater than Mukaroh *et al.* (2020).

Note that the results of Houidi *et al.* (2020) were obtained from a two-channel complex CNN, whose signal length (input of the time-frequency transform) was determined empirically since the authors faced a loss of relevant information when using the time-frequency representation with the reassignment process. In contrast to Houidi *et al.* (2020), our proposed method: (i) resulted in better classification results; (ii) had no significant loss of information, since the signal energy is almost entirely concentrated in the first-order coefficients of the ST (BRUNA; MALLAT, 2012); (iii) presented a good localization both in time and frequency, obtained from the wavelet-based structure, that contributed to better results; and (iv) comprised a feature extraction and selection structure analytically determined, and there are no learned coefficients in the convolutional network.

ST reaches significantly better F1-Score and Accuracy when compared with Baets *et al.* (2018). In Baets *et al.* (2018), the authors extracted features from weighted pixelated V-I images of NILM signals. This extraction process presented poor F1-Score results for high energy consumption appliances like washing machines, fans, fridges, and air-conditioners. Our $F1 - Score = 0.9812$ on PLAID overcame Baets *et al.* (2018), and this shows that our proposed feature selection method (Eq. 11) provides greater discriminability for the classification of high energy consumption appliances.

The results of the ST with the LIT-SYN dataset reached F1-Scores and Accuracies close to Lazzaretti *et al.* (2020). Still, the strategy proposed in Lazzaretti *et al.* (2020), different from the one proposed for ST, is based on a complex multi-agent approach, which raises the overall complexity of the classification process.

The proposal of Mukaroh *et al.* (2020) dealt with the load classification task as a denoising problem, and the authors proposed a Generative Adversarial Network to generate the noise distribution of background to clear the target load. This proposed structure is quite complex and requires two learning processes: one for the GAN and another for the CNN applied to classification. We reach an accuracy 8.48$\%$ greater than Mukaroh *et al.* (2020) with a more straightforward approach.

4.4 CONCLUSIONS OF THIS CHAPTER

NILM represents a set of essential tools for managing the consumption and production of electrical energy. Strategies for classifying and disaggregating NILM electrical signals are necessary to build such tools. The proposed ST approach reached better results than state-of-the-art CNN-based techniques with PLAID and LIT-SYN datasets with a convolutional network not dependent on training the feature extraction layers. The results showed that the accuracy of

ST framework is more stable in terms of signal length, sampling frequency, and event location. Also, ST presented better generalization capability in classification since the proposed method overcame the baselines using single and multiple loads as the training set.

The main contributions presented in this chapter are:

- A NILM classification framework using a convolutional-based network with the Scattering Transform, without the need to learn filter coefficients to extract features, reducing the amount of data required in the training process;
- An approach with better classification performance compared to the state-of-the-art methods for different publicly available datasets (LIT and PLAID);
- A time-frequency feature extraction technique that directly uses one-dimensional data as input, reducing the overall complexity and increasing the class discriminability.

5 NOVEL DEEP LEARNING ARCHITECTURE BASED ON SCATTERING TRANSFORM

We presented in Chapters 3 and 4 approaches based on ST to extract features and classify NILM signals. In Chapter 3, we evaluated the performance of ST under non-ideal dataset conditions: sub-sampling, reduced number of examples, and reduced number of measurements.

We explained ST in section 2.2, as a network similar to CNN (AGUIAR *et al.*, 2021b), sharing its advantages (high discriminability), but with the following advantages: (i) there is no need for weight training; (ii) the ST coefficients have physical meaning (location in time and frequency), which is particularly interesting in the NILM problem; (iii) as a consequence of the coefficients not being trained, ST needs less training data than CNN to reach equivalent evaluation performance.

The mentioned advantages of DCN methods (highly discriminative features and no need for pre-knowledge about the appliances) combined with the ST, encouraged us to propose a framework for extracting features from high-frequency NILM signals, using a new architecture with the ST replacing the CNN in a multi-task application, called ST-NILM. Hence, the main contributions of this chapter are:

- We propose a single framework that integrates detection and classification of aggregated loads;
- Our proposal uses a shared network that does not need a training step, thus requiring less data and no need for data augmentation (DA) approaches;
- Unlike other state-of-the-art approaches, our proposal utilizes a multi-label, non-trainable feature-extraction framework to disaggregate and classify multiple aggregated loads;
- The architecture of the proposed convolutional network is determined analytically through the selection of the appropriate ST parameters for NILM;
- Our proposal requires less computational cost than the state-of-the-art methods, with superior results for most cases;
- An ablation study on ST-NILM parameters that evaluates the impact of varying the Scattering transform parameters on the classification and detection of electrical loads;

- An embedded version of our proposed framework was also developed for demonstrating its feasibility for a real-time application.

The remaining of this chapter is organized as follows. We introduce ST-NILM in Section 5.1, highlighting all the steps that compose it. We show the preprocessing process in Section 5.1.1, and present the scattering network (and its parameterization) in Section 5.1.2. We detail the dataset used in Section 5.2. The results obtained are shows in Section 5.3, including an ablation study on the ST-NILM parameters, and the embedded implementation of our proposed framework. Finally, general conclusions and future works are presented in Section 5.4.

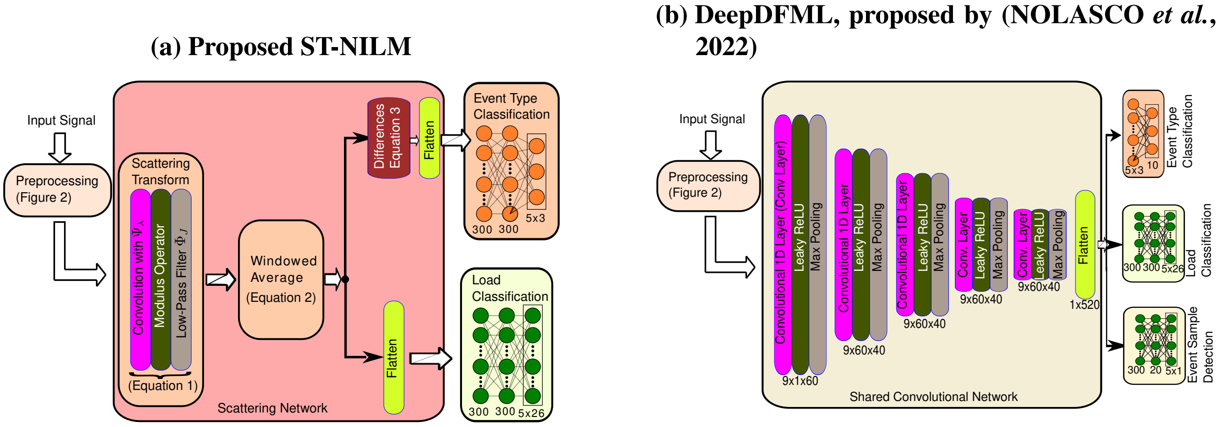
DeepDFML-NILM method (NOLASCO *et al.*, 2022) used a DCN for detection, feature extraction, and multi-label classification of high-frequency NILM signals for the publicly available LIT-Dataset. The detection results were above 90% for most cases, whilst the classification accuracies were comparable with other state-of-the-art works (around 97%). However, due to the high number of parameters, the proposed method requires data augmentation strategies, in addition to significantly increasing the computational complexity.

Despite the relatively good results presented so far, some of the main limitations with the architectures based on DCN applied to NILM are: (i) The proposed architectures generally do not integrate detection, disaggregation, and classification, i.e., the network performs only one or two of these tasks – with the exception of DeepDFML (NOLASCO *et al.*, 2022); (ii) As they use deep learning techniques, these solutions demand a large amount of data for training, since the computational effort grows with the increase in the number of coefficients (weights) of learned convolutional filters. The ST, on the other hand, contributes to reduce the computational effort, by replacing the convolutional weights by analytical wavelet coefficients. Hence, this chapter proposes an original and unified architecture for high-frequency NILM signals, combining the advantages of ST with DCN approaches for signal detection (identification) and classification.

5.1 THE PROPOSED ST-NILM

Since our approach is based on the DeepDFML (NOLASCO *et al.*, 2022), we first present in this Section a brief overview of that architecture. The DeepDFML was inspired by YOLO (REDMON; FARHADI, 2017), and it detects the instant in time when each electrical load is turned on or off, and predicts the combination of loads present in the input signal. The architecture of DeepDFML has a shared network with five layers and several learned coefficients.

Figure 24 – (a) Proposed ST-NILM and (b) DeepDFML architectures: The proposed ST-NILM architecture is inspired by DeepDFML, but by replacing the Shared Convolutional Network of Figure 24(b) by the Scattering Network of Figure 24(b) We also adapt the fully connected network for event type classification to obtain better results.



(a) Proposed ST-NILM

(b) DeepDFML, proposed by (NOLASCO *et al.*, 2022)

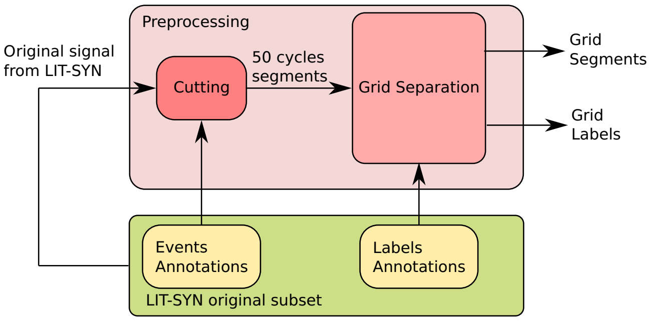
Source: Own authorship

The size of the convolutional filters, the number of layers, the pool size, and all other parameters were empirically determined.

We present DeepDFML in Figure 24(b). The DeepDFML method has a Shared Convolutional Network after the preprocessing stage and three sub-networks (one for each task) at the output. Two sub-networks deal with classification tasks (event type and load identification). The event type sub-network performs the binary classification of the turn-on or turn-off event of an appliance. The load identification sub-network performs the multi-label binary classification of the aggregated appliances in the input sample. The event sample detection sub-network deals with a regression task, and its output is a real number from 0 to 1 that informs the event's position inside a grid time interval.

Figure 24(a) shows the proposed ST-NILM architecture. In the ST-NILM, we replace the shared deep convolutional network of the DeepDFML with the Scattering Network. The Scattering Network is a network in which the filters are fixed, not learned, unlike CNN. Figure 24 shows the proposed ST-NILM architecture. The most relevant differences between our proposal (ST-NILM) and DeepDFML-NILM are (i) ST-NILM does not have trained weights on the shared convolutional network (named Scattering Network in Figure 24(a)); (ii) The determination of the Scattering Network weights is analytical (determined by wavelets) and, therefore, it is possible to calculate them *a priori*; We perform load detection through the Event Type Classification network. This network detects whether the load has been turned on or off within a grid interval (less than 166ms). We consider this detection resolution sufficient, so we do not use DeepDFML's Event Sample Detection network on ST-NILM.

Figure 25 – Preprocessing Strategy Applied to ST-NILM and proposed in (NOLASCO *et al.*, 2022).



Source: Own authorship

We detail all the parts that compose the proposed architecture as follows[1].

5.1.1 Preprocessing

A fim de manter a comparabilidade dos resultados, we use the same preprocessing strategy used in DeepDFML-NILM applied also to ST-NILM. To illustrate the preprocessing strategy proposed in (NOLASCO *et al.*, 2022) and used for ST-NILM, we present the block diagram of Figure 25.

In Figure 25, the green blocks represent the original information contained in the LIT-SYN subset, and the red blocks represent the preprocessing stage. The original LIT-SYN subset signals are cut into smaller segments of approximately 15600 samples (this is around 50 cycles of 60Hz plus two unmapped margins). The unmapped regions are a result of the labeling strategy proposed in Nolasco *et al.* (2022), whose effectiveness worsened in samples close to the beginning and end edges of the 50 cycles. The cutoff points of the original signals come from the electrical loads connection annotations contained in the original subset. Each output segment of the cutting block must have at least one turn-on event. The 50-cycles segments go through a grid separation block (Figure 25), which divides them into five regions of 10 cycles, each called *grids*. From the labels and events annotations originally contained in the dataset, we determine multi-task labels for each of the grids as follows:

- *Event type label:* 3-bit binary one-hot-encoding that determines whether a load has turned on or off in the range of a grid;
- *Multi-label Classification:* Binary one-hot-encoding that indicates which loads are connected in the range of a grid.

[1] All the codes for ST-NILM are publicly available at https://github.com/LucasNolasco/ST-NILM.

5.1.2 Scattering Network

The proposed method replaces the shared Deep Convolutional Neural Network applied in (NOLASCO *et al.*, 2022) with the Scattering Network, which has no learned coefficients. The path of the input signal in the Scattering Network block follows (i) Scattering Transform; and (ii) Windowed Average; (iii) Flatten. We will address each of these steps as follows.

5.1.2.1 Scattering Transform Stage

Let a time-domain electrical current signal. Guth *et al.* (2021) showed that phase collapse improves the classification performance of convolutional networks in which there are time-shifting invariant classes (one-dimensional case). Phase collapse is the phase information elimination performed by some non-linearity. The ST performs phase collapse explicitly through the modulus applied to the convolution of the time-domain signal the wavelet (GUTH *et al.*, 2021). Real-valued CNNs, for example, perform phase collapse implicitly (GUTH *et al.*, 2021).

Now we will define the main parameters of the Scattering transform: the bandwidth and central frequencies of the filters, the number of wavelets per octave Q, the maximum scale (J) and the order (m) (or the number of filter banks):

- *Choosing the number of wavelets per octave Q:* The more wavelets per octave, the greater the number of ST coefficients, and the more selective the filters. Studies in the literature using ST chose Q values between 8 and 12. We chose $Q = 10$ for the ST-NILM;

- *Choosing maximum scale J:* We choose J based on the nature of the harmonic content of electrical signals. Consider a discrete signal $x[n]$, with sampling frequency f_s and a total number of samples equal to T. So, the highest representable frequency, by Nyquist's theorem, is $\frac{f_s}{2}$. Because of that, the maximum frequency scale (f_{max}) analyzed must represent a frequency 2^J, lower than half the sampling frequency. We choose $J = 10$, so that the maximum frequency scale is $f_{max} = 2^{10} = 1024Hz$. This choice guarantees that harmonic currents (multiples of the fundamental frequency of 60Hz) of orders lower than 17 are represented by the ST. The authors in (RAMíREZ-RAMíREZ *et al.*, 2019) showed that the harmonic distortion of the combination of residential loads for frequencies greater than 1020Hz (order 17) represents less than 4% of the distortion found, for example, at 180Hz (order 3). With that in mind, we consider the choice of $J = 10$, representing a maximum frequency scale of 1024Hz, reasonable for our application in electrical signals;

- *Choosing order m:* Literature shows that the Scattering Coefficients (S) with two layers already represent up to 98% of the energy of the input signal, and all second-order coefficients together represent only 20% of the first-order coefficients' total energy (BRUNA; MALLAT, 2011). In addition, equivalent electrical appliances classification results were presented in (AGUIAR *et al.*, 2023) for $m = 1$ and $m = 2$. We chose $m = 1$, considering that the ST with $m = 1$ has fewer coefficients than the ST with $m = 2$, resulting in less computational effort.

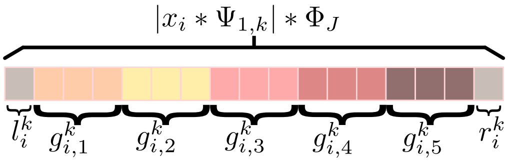
5.1.2.2 Windowed Average

We use a windowed average to select the features shared by the two output networks. The first-order coefficients of the ST, in each path, are signals located in time, given by:

$$S_i^k = |x_i * \Psi_{1,k}| * \Phi_J. \tag{26}$$

Figure 26 shows the division for windowed average. Let K be the number of samples of S_i^k. We divide the range $n \in [0,N]$ into five integer parts ($g_{i,1}^k \dots g_{i,5}^k$), centered on the same range. The remaining samples from this entire division were at the ends of the range, and we named them *unmapped region* (represented with l_i^k,r_i^k). The idea is to combine the result of the ST using the same grid division applied by DeepDFML when encoding the model outputs. We maintained the unmmaped regions representation into the grids division for windowed average to maintain compatibility of temporal location with annotations made with the same preprocessing of Nolasco *et al.* (2022).

Figure 26 – Grids division for windowed average.



Source: Own authorship

Next, the average of each l_i^k, r_i^k, and $g_{i,n}^k$ segments are calculated. We define $f_n^k = \bar{g}_{i,n}^k$, $f_{i,l}^k = \bar{l}_i^k$, and $f_{i,r}^k = \bar{r}_i^k$, therefore the selected features for each wavelet are:

$$Sf_i^k = \begin{bmatrix} f_{i,l}^k & f_{i,1}^k & \cdots & f_{i,5}^k & f_{i,r}^k \end{bmatrix}. \tag{27}$$

5.1.2.3 Event type Classification and Load Classification Networks

The Event Type classification network determines what type of event is found within a grid's time interval using one-hot encoding on a multi-class classification task. The input signal for this task is derived from the differences between consecutive grid interval averages. We define the differences for the k-th wavelet filter and the i-th sample as:

$$D_i^k = \begin{bmatrix} d_{i,0}^k \\ d_{i,1}^k \\ d_{i,2}^k \\ d_{i,3}^k \\ d_{i,4}^k \\ d_{i,5}^k \\ d_{i,6}^k \end{bmatrix} = \begin{bmatrix} f_{i,1}^k - f_{i,l}^k \\ f_{i,2}^k - f_{i,l}^k \\ f_{i,3}^k - f_{i,1}^k \\ f_{i,4}^k - f_{i,2}^k \\ f_{i,5}^k - f_{i,3}^k \\ f_{i,r}^k - f_{i,4}^k \\ f_{i,r}^k - f_{i,5}^k \end{bmatrix}. \tag{28}$$

For the event type classification the feature set is formed by concatenating the coefficients D_i^k, $k = [0 \ldots K]$ such that K is the total number of wavelets with order less than or equal to one. A Flatten layer is applied to D_i^k in order to obtain a 1D vector with all the selected features.

The output for this sub-task has dimension ($n_{grids} \times 3$), where n_{grids} is the number of grids, and 3 is the number of possibilities for classifying events: 1 for turn-on, 2 for turn-off and 0 for a false event. This sub-task network uses a sigmoid layer as an activation function and the categorical cross-entropy as a loss function. This cost function, in turn, takes the following form:

$$CCE = -\sum_{i=1}^{M} p(x_i) \cdot \log q(x_i), \tag{29}$$

in which M is the total number of examples, $p(x_i)$ is the expected probability of x_i to be from the class being analyzed, and $q(x_i)$ is the probability found by the model.

Also, there is a Load Classification network, which serves to determine which loads are present in the sample to be classified. This network represents a multi-label classification task, which allows the identification of multiple loads simultaneously. The input signal of this network is the set of concatenated coefficients Sf_i^k, given by applying a Flatten layer to Sf_i^k in order to obtain a 1D vector with all the selected features. The output has dimension $(n_{grid} \times n_{loads})$, where n_{grids} is the number of grids obtained from the windowed average, and $n_{loads} = 26$ is the number of electrical appliances in the LIT-SYN dataset. Since a multi-label classification was implemented, sigmoid was chosen as the activation function using the binary cross-entropy as a loss function. This cost function takes the following form:

$$BCE = -\frac{1}{M}\sum_{i=1}^{M} p(x_i)\log q(x_i) + (1 - p(x_i))\log(1 - q(x_i)). \tag{30}$$

It is worth noting that this sub-task did not require weighting to improve the classification performance.

Subsections 5.3.1 and 5.3.2 provide a more detailed explanation of the training and evaluation processes.

5.2 DATASET

For the experiments, we use LIT-SYN dataset, previously presented in subsection 4.1.1, for the following reasons: (i) The subset contains annotated samples of up to eight aggregate loads; (ii) the Natural subset was not available by the time this work was done; (iii) the Simulated subset does not contain real-world data; (iv) the Synthetic subset contains real-world data, from real-world appliances, with precise annotations.

5.3 RESULTS

This Section presents the results obtained with ST-NILM, comparing them with DeepDFML and related works. In addition, we present the procedures adopted for the experiments, and a discussion of the results obtained.

5.3.1 Experiments Setup

The dataset was divided into training and test sets, with a holdout test set containing 10% of the data. For training, with 90% of the entire dataset, we used a 10-fold cross-validation procedure. The model was trained for each fold considering a validation subset with 10% of the training subset size. The best model was chosen considering the average between the F_1 scores for binary classification on the validation subsets for each fold.

Three detection metrics were used, based on (NOLASCO *et al.*, 2022), as follows:

i. the $PC_{on} = \frac{A_{on}}{N_{on}}$ (Percentage of Correction of an ON event for a given method), such that A_{on} is the number of correct ON detections and N_{on} is the total number of turning on events;

ii. the $PC_{off} = \frac{A_{off}}{N_{off}}$ (Percentage of Correction Detection of an OFF event for a given method), such that A_{off} is the number of correct OFF detections and N_{off} is the total number of turning off events.

iii. the $PC_{av} = \frac{PC_{on}+PC_{off}}{2}$ (The arithmetic mean of PC_{on} and PC_{off}), which indicates the average detection performance.

For classification performance evaluation, we use the $\mathrm{F1-Score} = \mathrm{F1-Score}_{\mathrm{macro}}$ metric, given by Equation 24.

5.3.2 Results using a reduced LIT-SYN dataset

We present both detection and classification metrics considering reduced subsets for the training step. We perform the analyzes on reduced datasets for the following reasons: (i) In real-world cases of disaggregation, the amount of data can be limited, so an analysis of the performance of the NILM techniques under these conditions is necessary; (ii) To the best of our knowledge, related works do not address the effect of reducing the number of examples on the performance of NILM strategies; (iii) Methods with better performance with reduced datasets tend to be more easily implemented in real-time.

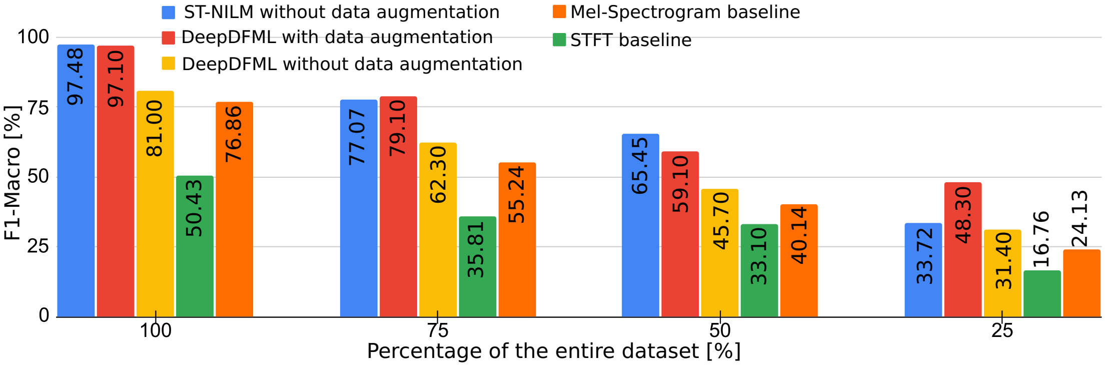
Table 14 shows the detection metrics for the reduced subset scenarios for training, and without data augmentation (DA). The detection metrics are comparable between ST-NILM and DeepDFML for large training sets (100% and 75%). For reduced training sets (50% and 25%), ST-NILM outperforms DeepDFML on all calculated detection metrics. We also inserted in Table 14 the metrics obtained with DeepDFML with data augmentation. ST-NILM, even without data augmentation, obtained detection results equivalent to DeepDFML with data augmentation (maximum difference in the PC metric of 4.8%).

Table 14 – Detection results for reduced training subsets.

Percentage	$PC_{on}(\%)$			$PC_{off}(\%)$			$PC_{av}(\%)$		
	ST-NILM	DeepDFML		ST-NILM	DeepDFML		ST-NILM	DeepDFML	
		with DA	without DA		with DA	without DA		with DA	without DA
100%	**97.84%**	93.50%	95.10%	91.04%	96.50%	**95.90%**	94.44%	95.00%	**95.50%**
75%	**95.68%**	90.90%	83.50%	84.20%	**90.80%**	81.80%	89.94%	**90.85%**	82.70%
50%	**91.13%**	83.50%	64.10%	**82.31%**	80.20%	64.10%	**86.72%**	81.85%	64.10%
25%	**72.66%**	60.20%	51.40%	**82.08%**	73.30%	47.40%	**77.37%**	66.75%	49.30%

Figure 27 shows the classification results obtained for reduced training datasets, comparing ST-NILM, DeepDFML (NOLASCO *et al.*, 2022), in addition to two baselines using STFT (SEJDIC *et al.*, 2009) and Mel-Spectrogram (DAVIS; MERMELSTEIN, 1980) instead of ST as feature extractors. Notice that the classification metrics are better for ST-NILM in all non-data-augmented scenarios with reduced training dataset. For comparison purposes, we also insert in Figure 27 the FScore-Macro obtained with DeepDFML with data augmentation. The classification result obtained with ST-NILM without data augmentation is equivalent or higher to that obtained by DeepDFML with data augmentation for the scenarios with more training examples (maximum difference of 2%) and lower (difference of 14.58%) for the scenario with 25% of training data. The detection performance using the spectral extractors STFT (SEJDIC *et al.*, 2009) and Mel-Spectrogram (DAVIS; MERMELSTEIN, 1980) was low (reached a maximum of $PC_{av} = 49\%$, similar to the values found by (LAZZARETTI *et al.*, 2020)) and for this reason, we chose not to include them as detection performance baselines in the Tables 14 and 15.

Figure 27 – Classification results: F1-Score results for reduced training subset scenarios.



Source: Own authorship

5.3.3 Comparison with Related Works

The most similar architecture to ST-NILM is DeepDFML (NOLASCO *et al.*, 2022) and, therefore it was used as the primary baseline for comparing results. This Section also compares the results obtained with other related works. The best results achieved by DeepDFML considered a data augmentation strategy to expand the training set. This strategy increased the overall complexity of the model. On the other hand, our proposed method, ST-NILM, does not need data augmentation. We present in Table 15 the detection results comparing ST-NILM with DeepDFML in all subsets of the LIT-SYN dataset, also considering scenarios with and without data augmentation. The results of Table 15 show that: (i) The best ST-NILM results are achieved without data augmentation; (ii) ST-NILM detection results are more affected by using data augmentation than DeepDFML; (iii) ST-NILM without data augmentation is better than DeepDFML without data augmentation, i.e., PC_{av} is higher, for tripe loads (LIT-SYN-3); (iii) Except for LIT-SYN-2, the metric PC_{on} obtained by ST-NILM without data augmentation outperforms DeepDFML with data augmentation for all other subsets. (iv) Notwithstanding, ST-NILM without data augmentation surpassed DeepDFLM and correctly detected all events for LIT-SYN-1. For LIT-SYN-2 and LIT-SYN-3 the detection performances of ST-NILM and DeepDFLM were equivalent (maximum variation of 5.3%). For LIT-SYN-8 (the case with more aggregated loads), ST-NILM detected better turn-on events and, conversely, DeepDFML was better for turn-off events.

Table 15 – Comparison of detection results between methods considering each subset of the LIT-SYN dataset

Method	**Subset**	$PC_{on}(\%)$		$PC_{off}(\%)$		$PC_{av}(\%)$	
		Data augmentation?		**Data augmentation?**		**Data augmentation?**	
		No	Yes	No	Yes	No	Yes
ST-NILM	LIT-SYN-1	**100.0**	88.6	**100.0**	**100.0**	**100.0**	94.3
	LIT-SYN-2	91.1	92.1	93.5	90.5	95.3	91.3
	LIT-SYN-3	**98.7**	88.7	94.3	94.0	**96.5**	91.4
	LIT-SYN-8	**96.5**	80.5	77.5	73.6	87.0	77.0
		No	Yes	No	Yes	No	Yes
DeepDFML (NOLASCO *et al.*, 2022)	LIT-SYN-1	100.0	**98.9**	100.0	99.6	100.0	**99.3**
	LIT-SYN-2	**97.8**	**97.9**	**98.3**	**98.7**	**98.2**	**98.3**
	LIT-SYN-3	92.6	**94.2**	**98.1**	**98.3**	95.4	**96.3**
	LIT-SYN-8	88.4	**89.0**	**94.4**	**90.2**	**91.4**	**89.6**

Table 16 presents a comparison between our work and the main state-of-the-art methods, taking into account the following criteria: (i) *the maximum number of multiple loads*, which is the maximum number of appliances added together in a single aggregated signal used for the proposed framework; (ii) whether the method considers multi-label classification or not; (iii) whether the method uses *data augmentation* or not, which directly impacts the overall complexity of the proposal; (iv) whether the method is *embedded* or not, which indicates the feasibility of real-time implementation; (v) a classification or disaggregation metric. Table 16 shows that ST-NILM is the only method (among the analyzed methods) with an embedded application that performs multi-label disaggregation with FScore above 95%, without using data augmentation.

5.3.4 Ablation study on ST-NILM parameters

In order to verify the impact of the variation of parameters m, J, and Q on the ST-NILM performance, we perform a series of training processes with different combinations of these parameters. We reproduced all the experiments previously presented with $m = 1$, now with

Table 16 – Comparisons With State-Of-The-Art Methods. The ✓ symbol indicates that the method has the related characteristic and the ✗ symbol indicates that the method does not have that characteristic.

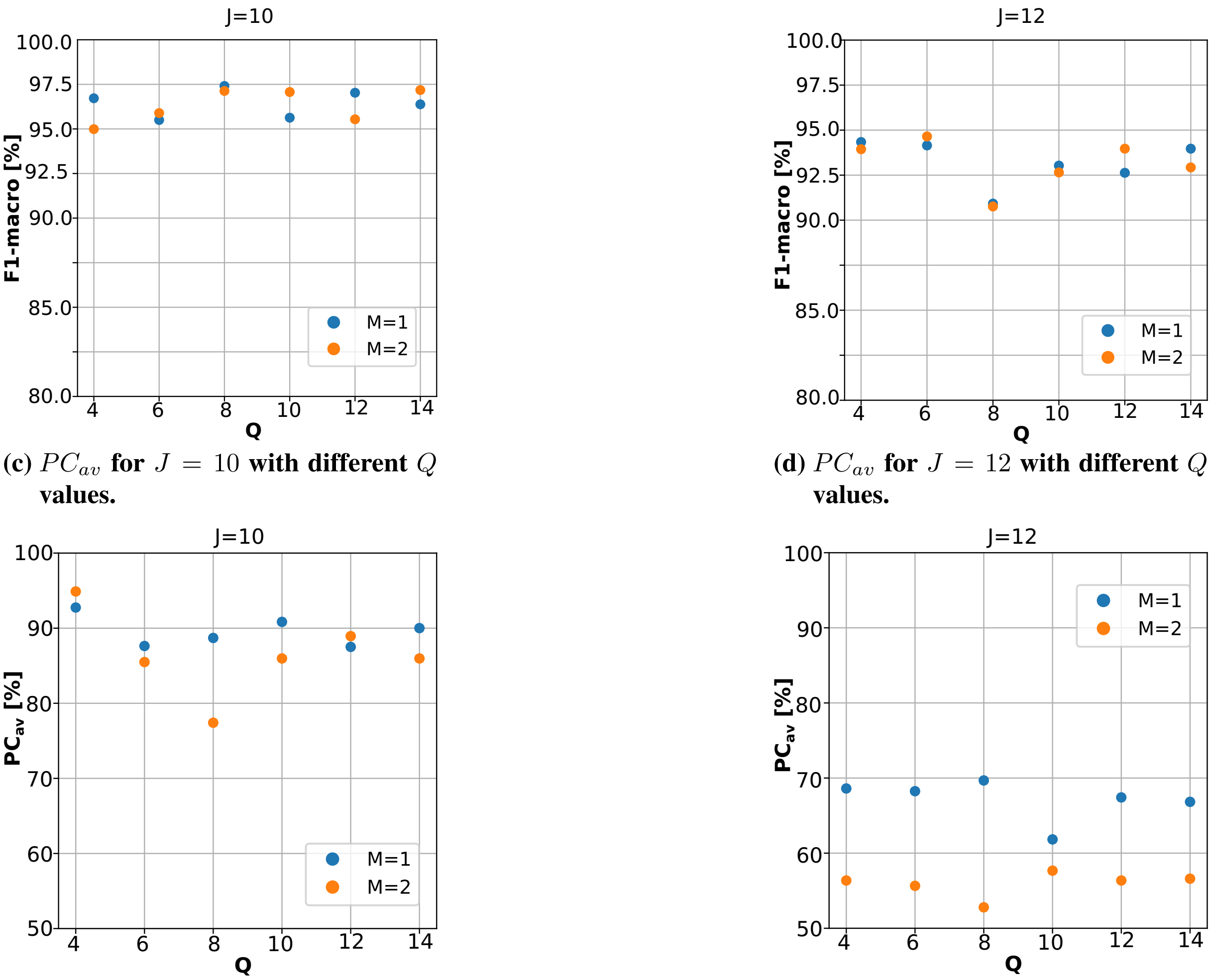
Method	Multiple loads	Multi-label?	Dataset	Augm.?	Embedded?	Best FScore (%)
(HOUIDI *et al.*, 2020)	1	✗	PLAID	✗	✗	97%
(WU; WANG, 2019)	5	✗	UK-DALE and BLUED	✗	✗	89%
(BAETS *et al.*, 2018)	-	✗	PLAID + WHITED	✗	✗	75%
(FAUSTINE; PEREIRA, 2020a)	1	✓	PLAID	✗	✗	94%
(FAUSTINE; PEREIRA, 2020b)	3	✓	PLAID	✗	✗	94%
(NOLASCO *et al.*, 2022)	8	✓	LIT-dataset	✓	✓	90%
(CHEN *et al.*, 2022)	3	✗	PLAID and WHITED	✗	✗	96%
(AGUIAR *et al.*, 2021a)	1	✗	COOLL	✗	✗	99%
(AGUIAR *et al.*, 2021b)	8	✗	LIT-dataset and PLAID	✗	✗	97%
(HWANG; KANG, 2022)	5	✗	REDD	✓	✓	88%
(MULINARI *et al.*, 2022)	8	✗	LIT-dataset	✗	✓	96%
This work	**8**	✓	**LIT-dataset**	✗	✓	**97%**

$m = 2$, with Kymatio library (ANDREUX *et al.*, 2020). For $m = 2$, for each time series, we extract the features considering the following:

- For the classification model, we use the averages of each first-order grid as features. For the detection model, we apply a flatten layer, as in Figure 24, and concatenate the difference between the averages of each first and second-order grid, in all sub-bands;
- We apply the same fully connected layers used for $m = 1$ in the tests for $m = 2$;
- For $m = 2$ we assume one wavelet per octave in the second-order filter bank and Q filters per octave in the first-order filter bank. For $m = 1$ we assume Q wavelets per octave in the first-order filter bank;
- We used the same classification metrics for $m = 1$ and $m = 2$.

Figure 28 presents the classification and detection results obtained by tests with parametric variation. Comparing the classification results in Figures 28(a) and (b), we notice that the F1-Score is not significantly impacted when choosing $m - 2$ instead of $m = 1$ (maximum variation less than 1.8%). We also notice that the average classification performance is better for $J = 10$ than for $J = 12$ (3.2% higher F1-Score average value) and that parameter Q has more impact on classification performance for J=12 than for J=10. The detection results presented in Figure 28(c) and (d) show that the J parameter is the most impactful for the detection since the average PC_{av} drops more than 25% when exchanging $J = 10$ for $J = 12$. Furthermore, the variance of PC_{av} with different Q is greater with $J = 12$ than with $J = 10$. We noticed that

Figure 28 – (a) Classification and (b) Detection results considering different ST parameters. Note that classification results do not variate significantly by increasing J. Detection metric PC_{av} deterioration, on the other hand, is much more evident when exchanging $J = 10$ by $J = 12$.



(a) F1-Score for $J = 10$ with different Q values.

(b) F1-Score for $J = 12$ with different Q values.

(c) PC_{av} for $J = 10$ with different Q values.

(d) PC_{av} for $J = 12$ with different Q values.

Source: Own authorship.

the metrics PC_{av} are better for $J = 10$ in most analyzed scenarios. Finally, the analysis of the results leads us to conclude that: (i) Increasing the value of J does not necessarily improve the classification performance and aggravates the detection performance; (ii) Using $m = 2$ instead of $m = 1$ does not result in significant improvement in classification performance, and deteriorates detection performance.

5.3.5 Embedded System

With the objective of evaluating the practical implementation of the proposed method, a subject also underexplored in the literature (Table 16), ST-NILM was tested on a Jetson NVIDIA TX1 to evaluate its performance in an embedded system environment. This platform has a

256-core GPU, a 4-core ARM Cortex-A57 CPU, and 4GB of RAM under a Linux operating system. The inference performance, presented in Table 17, was obtained from an average of 841 examples. Compared to DeepDFML, ST-NILM had lower RAM usage (62.7% vs 94.8%), lower GPU load and substantially fewer floating point operations (0.264 GFLOPS for ST- NILM and 7.643 GFLOPS for DeepDFLM).

Table 17 – Resources consumption on the embedded system.

Resource monitored	System state	Value
RAM	Idle	808/3984 Mb (20.3%)
	ST-NILM	**3379/3984 Mb (84.8%)**
	(NOLASCO *et al.*, 2022)	3777/3984 Mb (94.8%)
Avg. CPU load	Idle	8.2%
	ST-NILM	44.0%
	(NOLASCO *et al.*, 2022)	30.9%
Avg. GPU load	Idle	0.0%
	ST-NILM	**38.0%**
	(NOLASCO *et al.*, 2022)	50.6%
FLOPS	(NOLASCO *et al.*, 2022)	7.643 G
	ST-NILM	**0.264 G**

5.4 CONCLUSIONS OF THIS CHAPTER

In this chapter, we presented an architecture to classify electrical signals in an unified way. The proposed network, ST-NILM is based on the DeepDFML architecture, which has a shared CNN with many learned parameters. However, ST-NILM has an untrained shared network based on the Scattering Transform that has analytically computed filters given by wavelets.

We tested ST-NILM with the LIT-SYN dataset and repeated the same tests with the DeepDFML architecture for comparison purposes. Classification results with smaller datasets were substantially better for ST-NILM than DeepDFML. Furthermore, classification results show that ST-NILM, even without data augmentation, outperforms DeepDFML with data augmentation. The good classification performance of ST-NILM, unlike DeepDFML, is not dependent on data augmentation. The improvements in classification performance over DeepDFML can be explained by the fact that there is no need for the Scattering network to learn the filter coefficients, only the output networks. The lack of training in the features extraction process contributed to the ST-NILM obtaining better results than DeepDFML for smaller training sets. ST-NILM achieved detection results equivalent to DeepDFML for LIT-SYN2 and LIT-SYN-3, and performed better for single loads (LIT-SYN-1).

Performance gains of ST-NILM over DeepDFML in both classification and detection, despite being an important indicator of the method's applicability, in themselves may not represent a significant impact in practical applications. On the other hand, the resources used by the hardware implementation showed that ST-NILM consumes less memory, less GPU load, and consumes substantially less computational effort. As shown in Table 17, ST-NILM used less than 4% of the DeepDFML floating point operations, representing an essential step towards making the architecture compatible with a cheaper and more efficient hardware. The results obtained for the embedded system with LIT-SYN (a real-world dataset) indicate that future tests with ST-NILM and real-time data can be promising.

6 NILM FOR PV INVERTER IDENTIFICATION

Bearing in mind the scenario of distributed micro and mini generation and also the concepts previously discussed related to NILM, in this chapter, we intend to answer the following research questions (RQ):

- **RQ1:** *Does the presence of DG interfere with the loads classification?*
- **RQ2:** *Is it possible to identify the presence of the inverter aggregated with the loads?*

The answer to RQ1 reinforces the advantages related to traditional NILM, such as energy savings, planning, and monitoring of energy consumption by the consumer. In addition, RQ1 helps with distribution transformer sizing and distribution network protection. The answer to RQ2 helps in all aspects mentioned in Section 2.4. A specific literature review for these works is presented in Subsections 2.4.1 and 2.4.2. However, to the best of our knowledge, no works involving these themes use real-world data sampled at high frequency in a system that contains both electrical loads and distributed photovoltaic generation.

In this way, our work intends to fill the following Research Gaps (RG):

- *RG1:* Lack of public datasets with real-world validated data that contain both Distributed Generation and aggregate electrical load signals;
- *RG2:* Lack of frameworks for classifying and detecting Distributed Generators aggregated with NILM loads.

6.1 CONTRIBUTIONS OF THIS CHAPTER

Based on the research gaps RG1 and RG2, our main contributions arising from this chapter are:

- Analysis, design, and implementation of an experimental arrangement to obtain NILM samples with Distributed Generation, allowing the reproducibility and collaborative future expansion of the proposed dataset;
- Construction and validation of a novel public dataset with electrical signals from both electrical loads and Distributed Generation;

- Proposition of frameworks for disaggregation and detection of Distributed Generation in a high frequency dataset aggregated with electrical loads;
- Get state-of-the-art detection and disaggregation results for the proposed dataset.

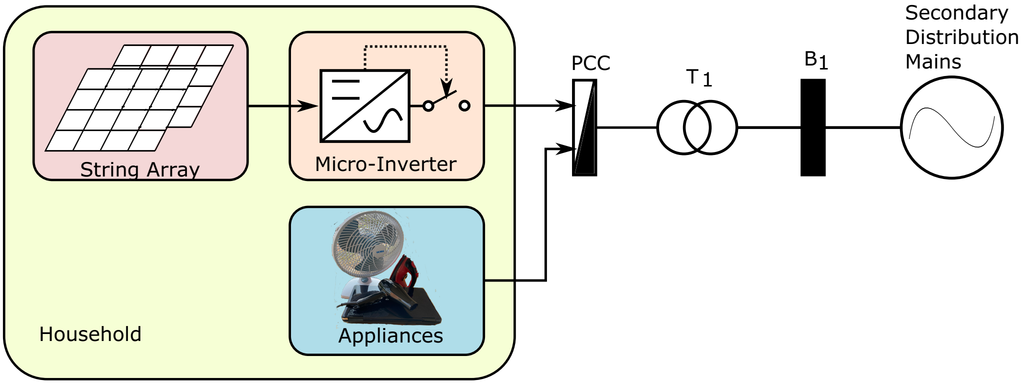
Section 6.2 presents the development process of our proposed dataset, highlighting the design and implementation of the hardware and embedded software. Section 6.3 shows our proposed frameworks applied to our dataset, obtaining novel NILM and DG results. Answers to RQ1 and RQ2 are discussed in Section 6.4. We reserve Section 6.5 for the conclusions of this chapter.

6.2 CONSTRUCTION OF A NOVEL HIGH-FREQUENCY DATASET WITH NILM AND PV DATA

We propose a new dataset with high-frequency data for NILM containing electrical loads and distributed PV generation measurements. For this, we design an experimental arrangement and collect real-world data. The Figure 29 diagram represents the experimental arrangement. Figure 29 shows a typical residential micro photovoltaic distributed generation (DG) system. This system has: (i) a set of photovoltaic panels (reference) called *string array*, which generate, from solar irradiation, a continuous voltage through the photovoltaic effect; (ii) a micro-inverter, which is a direct current to alternating current static converter. The micro-inverter internally has an automatic synchronization structure, which connects the DG to the mains. Figure 29 shows the connection of the household appliances to the Point of Common Connection (PCC). The energy consumed by the appliances comes from both the DG system and the distribution mains. Modern micro-inverters already certified for commercial use already have static converters to regulate the DC bus from the string-array voltage, in addition to automatic real-time control of generated active and reactive power and the maximum power point tracking (MPPT).

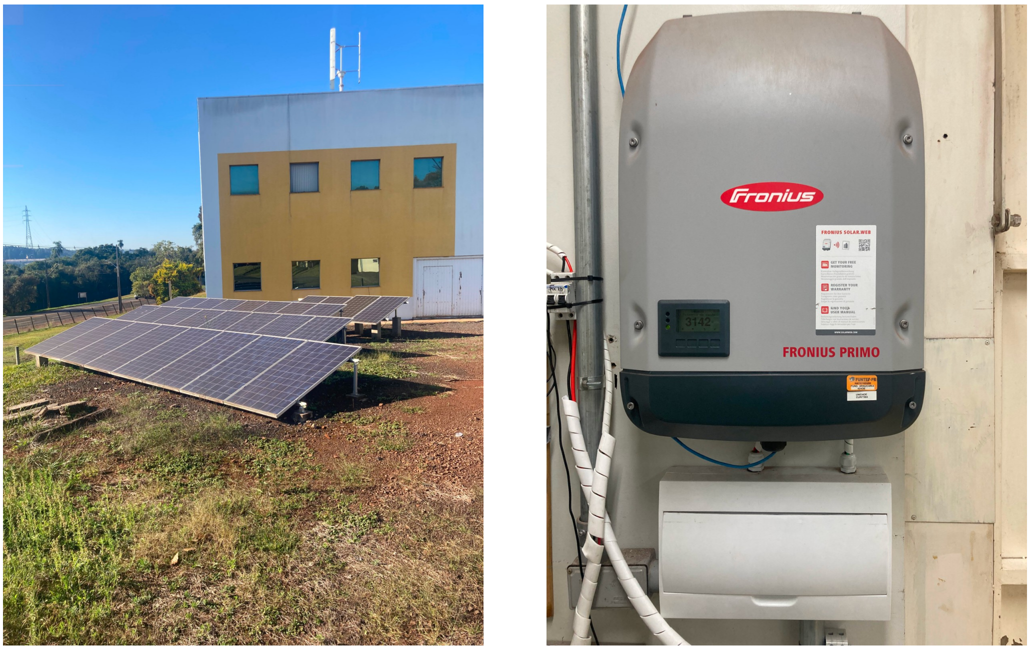
We partnered with campus Pato Branco of the Federal University of Technology (UTFPR) to access the facilities of the photovoltaic generation unit and measure energy generation and consumption. The laboratory of the graduate program in electrical engineering (PPGEE) at UTFPR Pato Branco, POLITEC, provides an arrangement of photovoltaic panels, as shown in Figure 30a. A Fronius Primo 8.2-1 micro-inverter, which allows remote control, was also used, as shown in Figure 30b. We show in Figure 30c the laboratory installations where we developed this part of the work.

Figure 29 – Typical Micro Grid-Tie Photovoltaic Distrituted Generation System



Source: Own authorship.

Figure 30 – Installations of the Graduate Program in Electrical Engineering at UTFPR-PB.



(a) Photovoltaic panel arrays.

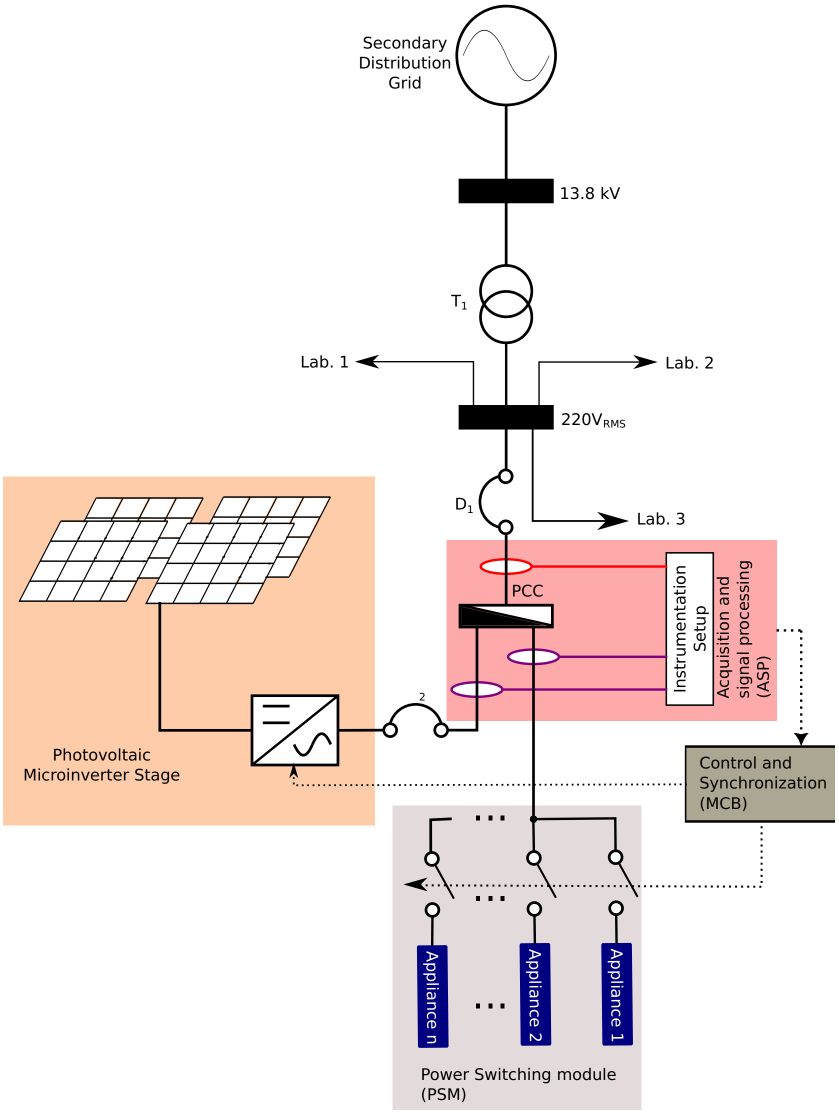
(b) Fronius Microinverter.

(c) Overview of the entire laboratory.

Source: Own authorship.

Fig 31 shows the single-line diagram of the experimental setup assembled in POLITEC, at campus Pato Branco of the Federal University of Technology (UTFPR), to develop the proposed DG-NILM-v1 dataset. We measured the single-phase current in the low-voltage distribution circuit breaker. This distribution circuit breaker is shared with three laboratories. We developed a panel that contains an instrumentation unit to collect these electrical variables measurements. At the same time, we developed an electrical load-switching system to turn on and off with pre-determined patterns to the low-voltage bus. The remote control system also controls the activation of a microinverter (Fronius Primo 8.2k), which is also switched at known times.

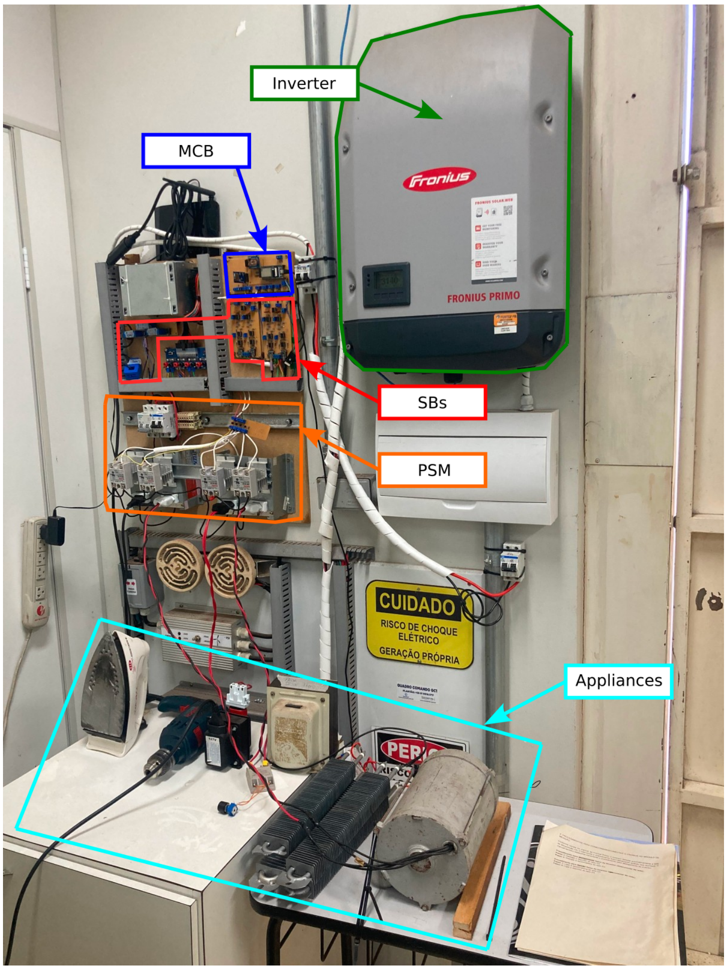
Figure 31 – Single-line diagram of the experimental setup designed by us and implemented in POLITEC.


Secondary Distribution Grid
13.8 kV
T_1
Lab. 1
Lab. 2
$220V_{RMS}$
D_1
Lab. 3
PCC
Instrumentation Setup
Acquisition and signal processing (ASP)
2
Photovoltaic Microinverter Stage
Control and Synchronization (MCB)
Appliance n
Appliance 2
Appliance 1
Power Switching module (PSM)

Source: Own authorship.

Figure 32 shows the experimental setup assembled. We present in this Section the implementation of the setup shown in Figure 31 and Figure 32, dividing it into three parts: *Hardware* design, *software* design, and *methodology for data logging*. The 6.2.1 Subsection presents the *hardware* design, which comprises the projects for the Acquisition and Signal Processing (ASP) module, the Control and Synchronization module, and the Power Switching Module (PSM). In the Subsection 6.2.2, we present the *software* design, which comprises the routines for interconnection between modules and command of devices. Finally, in Subsection 6.2.3, we present the methodology for obtaining the data that make up our proposed dataset.

Figure 32 – Overview of the assembled setup in POLITEC. We highlight the main structures with different colors.



Source: Own authorship.

6.2.1 Hardware Design

We divide the hardware design into three parts:

- *Control and Synchronization (MCB)*, presented in Subsection 6.2.1.1;

- *Acquisition and Signal Processing (ASP)*, presented in Subsection 6.2.1.2;
- *Power Switching Module (PSM)*, presented in Subsection 6.2.1.3.

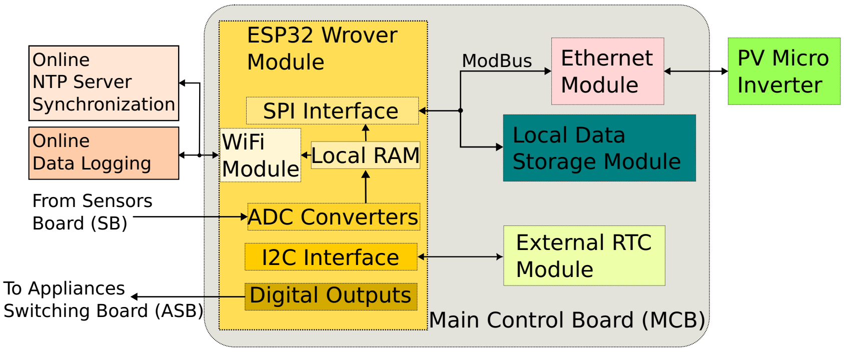
6.2.1.1 Control and Synchronization

We develop the Main Control Board (MCB) for system control and synchronization. The MCB is responsible for switching the electrical loads, activating the PV micro-inverter, and the analog-digital conversions of measured voltages and currents.

The MCB consists of a printed circuit board that uses the ESP32 WROVER microcontroller, 8Mb of pseudo-static RAM and a 240MHz dual-core processor. This microcontroller also has a WiFi module, SPI, and I2C communication. In the MCB, we use a I2C external Real-Time Control (RTC) module, model DS1307, synchronized with an online *Network Time Protocol (NTP)* server in addition to an SD card recording module HW111. We use Arduino IDE and FreeRTOS for MCB programming.

The Main Control Board block diagram consists of the structure shown in Figure 33.

Figure 33 – Main Control Board (MCB).



Source: Own authorship.

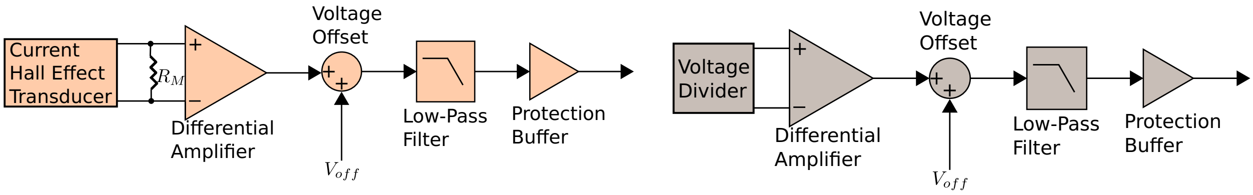
As shown in the diagram in Figure 33, the MCB has a wired connection with the PV micro-inverter through the Ethernet Module. The Ethernet Module connects to the ESP32 via the SPI protocol. We carry out the analog-digital conversions with the 12 bits internal converters of the ESP32 WROVER module. The sampled analog signals (voltage and current) come from the Sensor Boards (SB, detailed in Subsection 6.2.1.2). We store the sampled quantities, in the form of a buffer, in the ESP32 WROVER's RAM. We periodically save data from RAM locally (on the SD card, via SPI interface and Local Data Storage Module) and online (via WiFi module and Online Data Logging).

6.2.1.2 Acquisition and signal processing

We developed Sensor Boards (SB) for acquiring and conditioning NILM electrical signals. Each SB consists of a printed circuit board that deals with NILM analog signals (voltage or current) from electrical appliances or PV.

The SBs have the same circuit topology for both current and voltage signals. Figure 34 presents two general SB block diagrams. Each analog signal (voltage or current) that enters the SB passes through the following stages: (i) A differential amplifier with gain 3, which consists of a two-stage instrumentation amplifier topology, using the NE5534 operational amplifier, trimpots and precision resistors; (ii) A stage to insert voltage offset, in order to guarantee the full scale strict positive voltage signal at the output. We implemented this step with the NE5532 in a non-inverting adder configuration; (iii) A low-pass anti-aliasing filter with $8.84kHz$ of cutting-frequency; (iv) A rail-to-rail operational amplifier based protection voltage buffer, to keep the output voltage to the limits specified by the ADC of the microcontroller.

Figure 34 – Functional diagram of the two different SB (Current and Voltage). Both SB has the same circuit topology. The difference between them is the usage or not of a Hall current transducer.



(a) Current Instrumentation Sensor Board.

(b) Voltage Instrumentation Sensor Board.

Source: Own authorship.

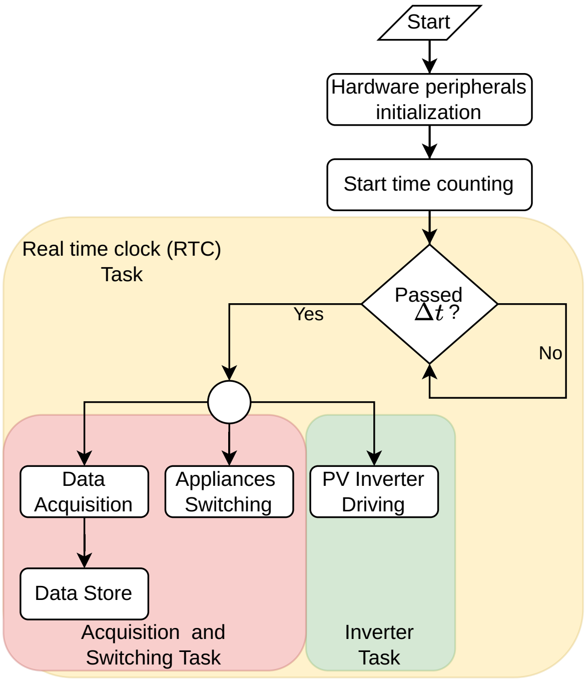
6.2.1.3 Power Switching Module

The Power Switching Module (PSM) is responsible for turning the appliances on and off at periods predetermined by the MCB. At PSM we use EARU SSR-40DA solid state relays. We drive the relays with the signals from the digital outputs of the ESP32 (0 and +3.3V). Each solid state relay, responsible for switching each appliance, has $40A_{rms}$ of rated current.

6.2.2 Software Design

We develop the embedded software based on a FreeRTOS framework. We define a main task for Real-time clock control (RTC) with a maximum priority level. The RTC task uses an internal ESP32 timer to generate a time reference synchronized by an NTP server, which is valid for the entire system. At every Δt time interval, the Acquisition and Switching Task (AST) initiates a sequence of conversions (called the Acquisition Window, or AW) at high frequency. AST also performs relay switching, turning electrical loads on and off at pre-programmed times. The embedded software design consists of the structure shown in Figure 35.

Figure 35 – General structure of the embedded algorithm.



Source: Own authorship.

Δt, in Figure 35, defines the interval between two consecutive high-frequency acquisition windows (AW). Consequently, one can define $f_{low} = (\Delta t)^{-1}$ as a low-frequency acquisition value for our framework. In other words, there are two main frequencies involved in data acquisition: f_{hight}, or the high-frequency sampling rate for the AST, and f_{low}, which is the low-frequency sampling rate that represents the inverse of the time interval between two AW.

Every Δt s, we also read the registers of the Fronius Primo inverter via the Modbus protocol using the WiFi module of the ESP32. These registers contain generation data (at low frequency), such as active power, reactive power, and RMS current.

The Algorithm 2 presents our strategy to implement the software project using FreeR-

TOS and the tasks we describe previously.

Algorithm 2 – Algorithm of the interconnection between the tasks we implemented to acquire the data for our dataset.

```
Input: $\Delta t$, $f_{high}$, window_size, SWtimes.
Output: $I_{agg}$, $I_{inv}$, $V_{pcc}$, Labels
Setup and initialize the microcontroller peripherals (ADC, timers, SD card storage, WiFi, and Modbus)
Setup a timer interruption for the ADC task
Initialize and synchronize RTC task with an NTP server
Initialize the Acquisition and Switching Task
Initialize the Inverter Task
while Not manually or remotely interrupted do
    for each $\Delta t$ time interval do
        Start the ADC acquisition window (AW) conversion for $I_{agg}$, $I_{inv}$ and $V_{pcc}$
        Store temporarily the NTP time correspondent to the start of the AW
        Access the Fronius Primo registers via WiFi, with the Modbus protocol
        Get the generation data from the Fronius Primo registers
        Store temporarily the samples in the RAM
        Start the Solid State Relays switching, based on SWtimes
        Store temporarily the states of the switches in a binary array named Labels, in the RAM
        if the number of conversions reaches window_size then
            Stop the ADC conversions
            Stop the Switching process
            Store the RAM values into the SD card
        end if
    end for
end while
return $I_{agg}$, $I_{inv}$, $V_{pcc}$, Labels.
```

We reserve the Subsection 6.2.3 to explain the methodology for obtaining data for our proposed dataset. We highlight in Subsection 6.2.3 the loads' selection process and the format of the sampled data, as well as the choice of parameters Δt, f_{high}, f_{low}, SWtimes.

6.2.3 Methodology for data logging

The main parameters for data sampling are:

- *High sampling frequency* $(f_{high} = 1000Hz)$: Sampling frequency for each voltage or current measurement contained in an AW;
- *Low sampling frequency* $(f_{low} = \frac{1}{60})$: Sampling frequency of the PV inverter registers, or the inverse of the period between two consecutive AWs $(\Delta t)^{-1}$;
- *Acquisition window time interval (AWTI)*: This measure is given in seconds and represents the time interval corresponding to an AW;
- *Number of samples per acquisition window* $n_{aw} = 16000$: Knowing the previous definitions, $n_{aw} = f_{high} \times$ AWTI.

Figure 36 shows a set of typical waveforms for two consecutive AWs, highlighting the abovementioned parameters. Figure 36(a) shows in green two consecutive acquisition windows. In seconds, the duration of a particular acquisition window is $AWTI$. Two consecutive AW start=points are separated by an interval of $\Delta t = t_2 = t_{ini}$ seconds. In Figure 36(a), $AWTI = t_1 - t_{ini} = t_3 - t_2$. We also show in figure 36(a), in red, an example of the interval $t_{inv} - t_{ini}$ in which the PV inverter is turned on (generating energy). After the instant $t = t_{inv}$, the PV inverter is off (not generating energy). For our example case, in which we show two AW, we have the inverter on (*on* state) in the first AW (left green area) and off (*off* state) in the second AW (right green area).

Figure 36(b) shows an example of a set of switching patterns for four relays (switches a, b, c, and d), each controlling an individual appliance. Our hardware controls each switch with a predetermined switching interval. Note that the total time interval showed in Figure 36(b) is $AWTI$, being Figure 36(b) a zoomed version of Figure 36(a). In figure 36(b), we represent each of these switching intervals with different colors: pink for switch a, blue for switch b, yellow for switch c, and grey for switch d.

Our hardware allows us to choose the instants at which each load is triggered within an AW. This flexibility allows us to assemble the load combinations we want. Some possible examples are:

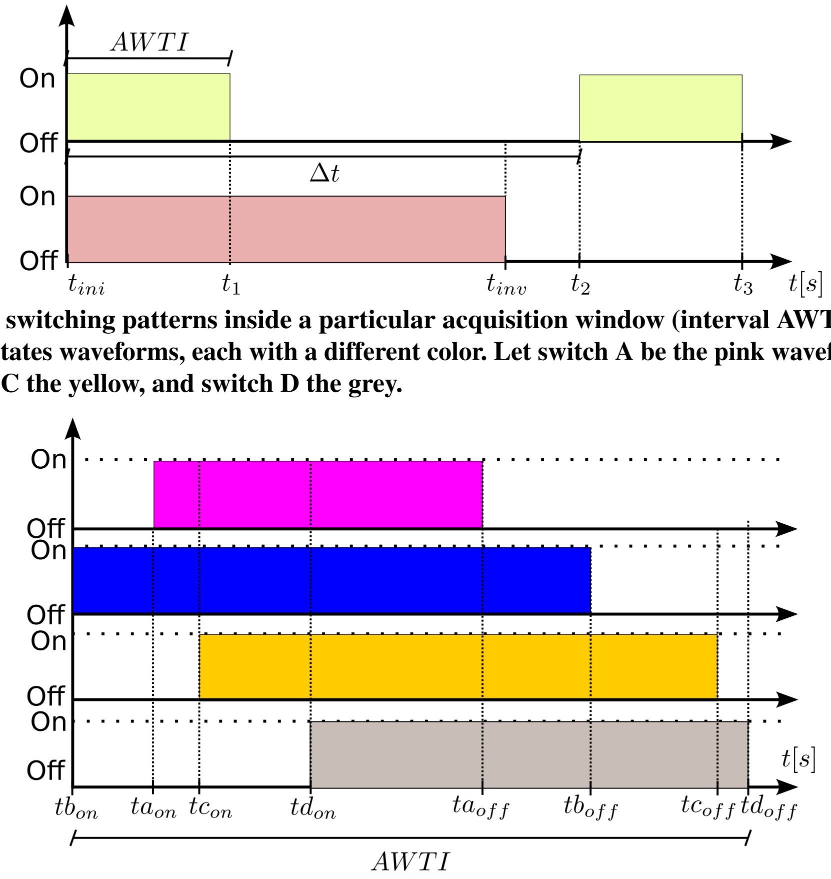
- *Sequential triggering of individual loads (TIL):* In this case, we choose a switching interval for each load so that only one load is activated at a time, without overlapping more than one load at the same time;

- *Aggregate triggering with fixed intervals (TFI):* In this case, we choose fixed switching intervals and always starting at the same times for all switches, with overlapping loads;

- *Aggregate triggering with a finite set of different intervals (TDI):* In this case, we choose a finite set of switching patterns for the relays, generating a finite set of different switching patterns for different AWs;

- *Aggregate triggering with a random set of different intervals (TRI):* In this case, we choose a random switching pattern for each load, and there is an infinite set of possible combinations between these patterns, generating all AW with different switching patterns.

Figure 36 – Set of typical waveforms for an acquisition window for the proposed dataset



(a) In green, we show two acquisition windows whose start point is separated by a Δt time interval. In red, we show an example of the state transition in PV inverter conditions, from on to off, if $t > t_{inv}$.

(b) Detail of the switching patterns inside a particular acquisition window (interval AWTI). We show here four switch states waveforms, each with a different color. Let switch A be the pink waveform, switch B the blue, switch C the yellow, and switch D the grey.

Source: Own authorship.

6.2.3.1 Generation profile and PV inverter labeling

We show in figure 36(a) a typical switch pattern for the PV invert binary state logging. In DG-NILM-V1, we create a binary variable specifically to signal whether or not the inverter is injecting active power into the distribution network. If the power generated by the DG inverter is greater than zero, the binary variable $y_{inv} = 1$ (shown in red in the figure 36(a)). If no active power is injected into the grid (in cases where the inverter is off, nighttime or rainy weather), $y_{inv} = 0$. The AW for DG-NILM-V1 was obtained under varying meteorological regimes to generate variability in the collected data.

6.2.3.2 Electrical appliances choosing

We choose four electrical loads of different natures to compose our dataset. We describe the loads chosen in the items below:

- *Electric Iron:* We choose an electric iron Black & Decker Quick'nEasy 315 with a nominal power of 900W. The electric iron is a predominantly linear (resistive) load, prevalent in most residential consumption units, and with relatively high power (above 500W);
- *Single Phase Induction Motor:* We chose a 0.5hp single-phase motor, with start capacitor. This type of load has a non-linear behavior caused by the magnetic characteristics of the iron and the air gap, resulting in a specific power signature. In Brazilian homes, it is common for this type of load to be used in washing machines that do not use frequency inverters;
- *Driller + Transformer:* We built an arrangement by connecting a driller Bosch 3601B185D0 with 127V nominal voltage to a single-phase 127V/220V transformer. The drill has a universal motor (DC motor in series configuration). This arrangement, with a total nominal power of 700W, has a non-linear behavior caused both by the motor action and by the magnetic saturation and in-rush current of the transformer;
- *Dimmer:* Bearing in mind that more than 70% of Brazilian homes have an electric shower (SANGOI; GHISI, 2019), that an electric shower is a resistive load, and that the temperature control of these devices is usually carried out employing thyristor switching systems, we build an arrangement of resistive loads commanded by a dimmer. We control the average power delivered to the resistive array through the dimmer firing angle. This adjustment leads to a non-linear behavior and adds considerable harmonic content to the residential network.

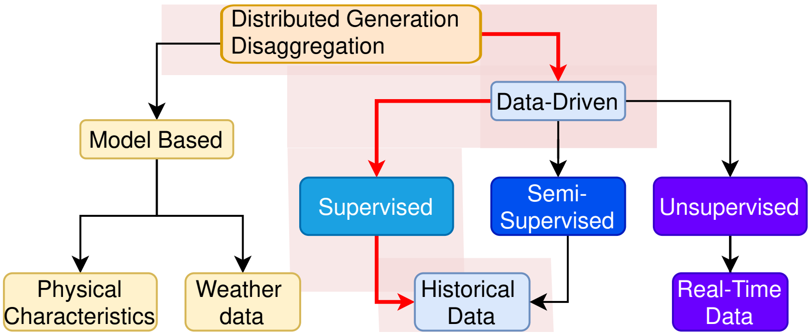
6.3 PROPOSED NILM AND PV DISAGGREGATION FRAMEWORKS

Our lines of investigation proposed in the disaggregation of PV-DG are based on the classification proposed by Wu *et al.* (2022), and shown in Figure 37.

Figure 37 shows our proposal category classification with bolted red lines. We propose a Data-Driven approach. Data-Driven approaches use feature extraction from measured electrical data. We propose a Supervised framework with labeled data available for training. As discussed in the introduction to this chapter, we also present a new dataset to store Historical Data and use them for training DG-PV disaggregation and classification models.

Considering our new high-frequency dataset, we propose a framework for evaluating electrical load and distributed generation disaggregation problems. Figure 38 presents the pro-

Figure 37 – Classification of DG-PV Disaggregation Methods in the Literature, as proposed by Wu *et al.* (2022)



Source: Adapted from Wu *et al.* (2022)

posed framework, highlighting the three basic experiments we consider in this Section: Load and Inverter Disaggregation Experiment (LIDE), Load Disaggregation Experiment (LDE) and Inverter Disaggregation Experiment (IDE). We intend to answer RQ1 with the LDE and RQ2 with the results of the IDE and LIDE.

6.3.1 Segmentation

We use a segmentation stage applied to the aggregated signals from the DG-NILM Dataset. This segmentation process returns chunks (time windows) of the aggregated signal, which must meet the following requirements:

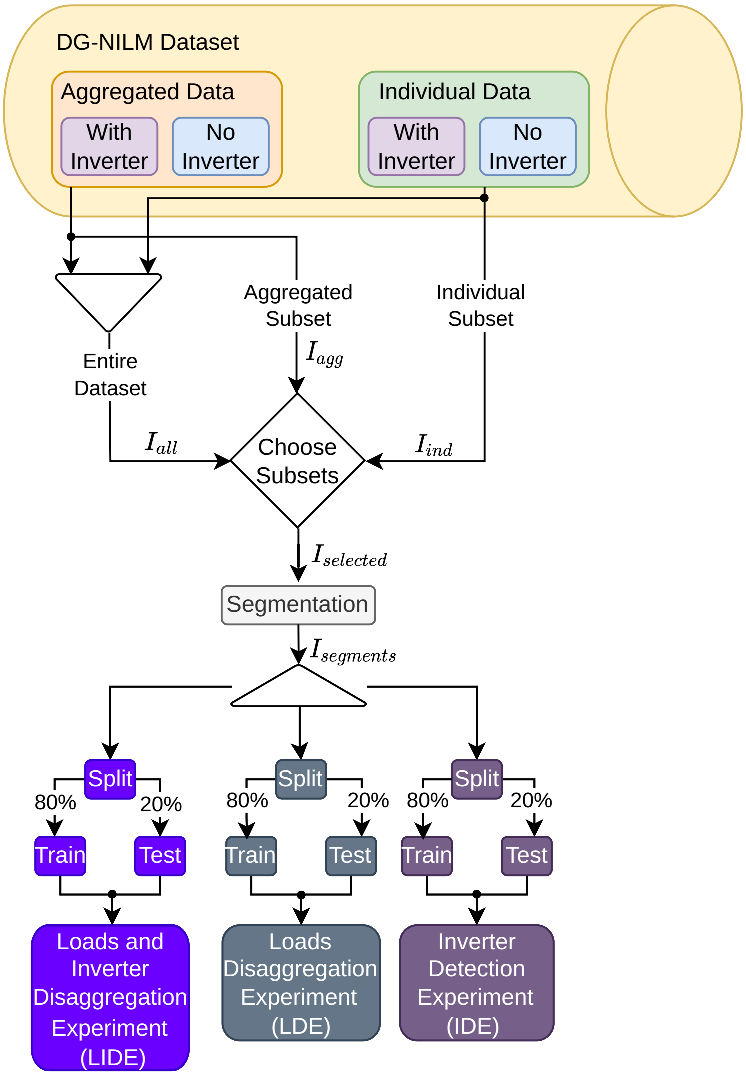
- *Requirement 1:* Every segment must have n_{window} samples; ;
- *Requirement 2:* Every segment must have a unique label for each appliance.

To achieve requirements 1 and 2, we use Algorithm 3.

Algorithm 3 – Segmentation of the DG-NILM dataset for the LIDE, LDE and IDE experiments.

```
Input: $I_{selected} = \{i_0, \ldots, i_N\}, Y_{selected} = \{y_0, \ldots, y_N\}$.
Output: $I_{segments} = \{i_{s0}, \ldots, i_{sS}\}, Y_{segments} = \{y_{s0}, \ldots, y_{sS}\}$
for each original DG-NILM dataset waveform $i_k \in \{i_0, \ldots, i_N\}$ do
    for each appliance and PV inverter binary output do
        Find a transition from zero to 1, defining its index as $i$.
        Select $n_{window}$ samples from $i$, defining this chunck as $i_{si}$.
        if there is no other transitions along the selected $n_{window}$ samples then
            Append $i_{si}$ to the output segments vector $I_{segments}$.
            Append the corresponding label logits $y_{si}$ to the output labels vector $Y_{segments}$.
        end if
    end for
end for
return $I_{segments}, Y_{segments}$.
```

Figure 38 – Proposed framework to evaluate PV Inverter and Electrical Loads Disaggregation in NILM Context. We train the models with high-frequency data from our proposed DG-NILM Dataset.

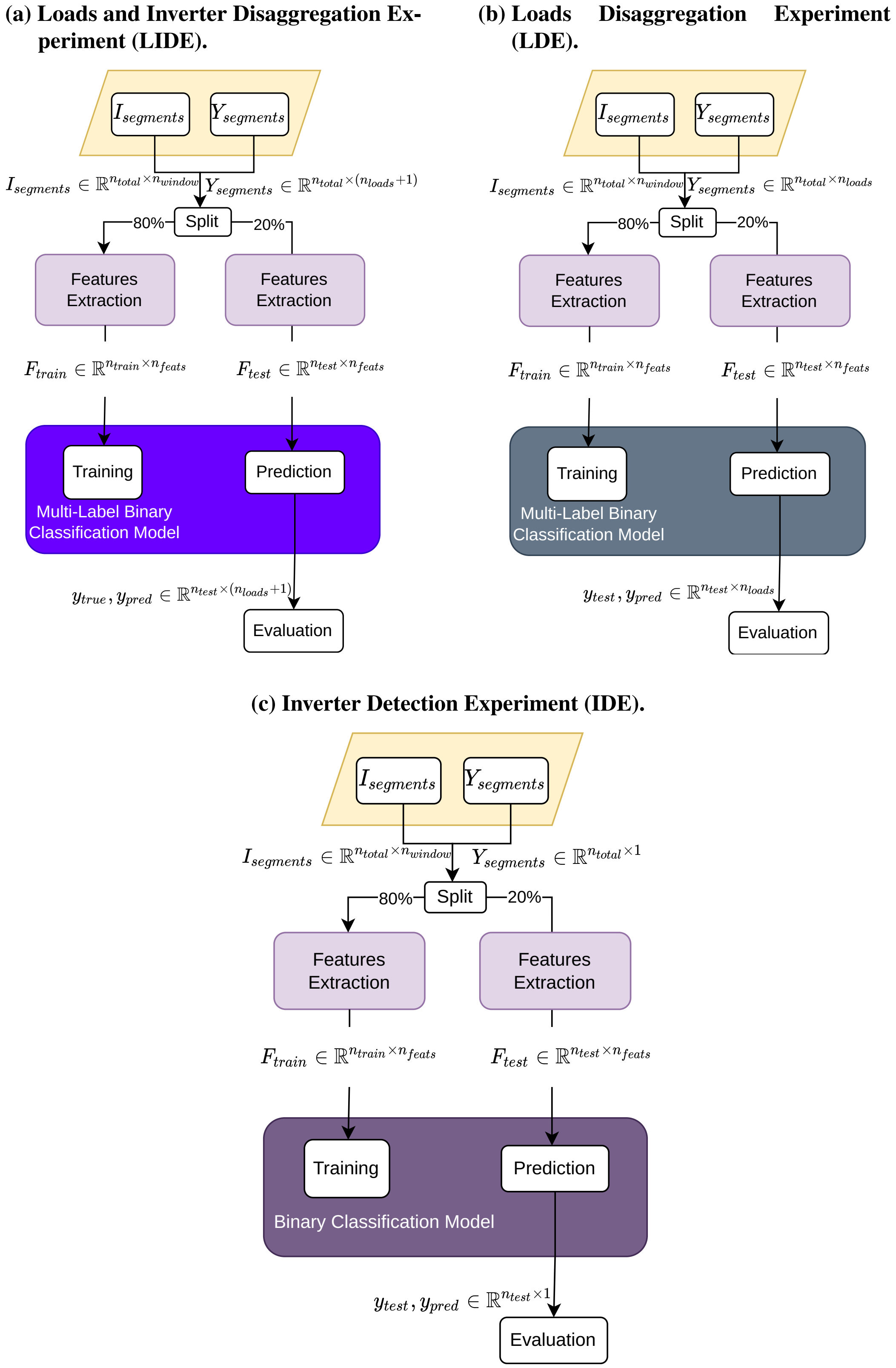


Source: Own authorship

6.3.2 The experiments LIDE, LDE and IDE.

Figure 39 shows the block diagrams of the LIDE, LDE, and IDE experiments. Note that each case has different dimensions of the annotated and predicted label arrays. As a consequence of this, we create different classification tasks for each of the experiments. For LIDE and LDE, we train multi-label binary classification models. In LIDE and LDE, we consider the presence or absence of each electric load in the current $I_{segments}$ as binary outputs for the classification task. Thus, the matrix of labels for LDE has dimension $(n_{total} \times n_{loads})$, where n_{total} is the total number of selected segments and n_{loads} is the total number of loads in the DG-NILM dataset. In addition to the n_{loads} outputs corresponding to the electrical loads, the output of the LIDE classification task has one more binary output corresponding to the presence or absence of the PV inverter. For this reason, the experiments LIDE have an array of labels with dimension $[n_{total} \times (n_{loads} + 1)]$.

Figure 39 – Frameworks to implement the Loads and Inverter Disaggregation Experiments (LIDE), Loads Disaggregation Experiments (LDE), and Inverter Disaggregation Experiments. LIDE and LDE are multi-label classification tasks, with label dimensions according to the number of loads and the presence or absence of the PV inverter. IDE is a binary classification task that only detects the presence or absence of the inverter.


$I_{segments}$
$Y_{segments}$
$I_{segments} \in \mathbb{R}^{n_{total} \times n_{window}}$
$Y_{segments} \in \mathbb{R}^{n_{total} \times (n_{loads}+1)}$
80%
Split
20%
Features Extraction
Features Extraction
$F_{train} \in \mathbb{R}^{n_{train} \times n_{feats}}$
$F_{test} \in \mathbb{R}^{n_{test} \times n_{feats}}$
Training
Prediction
Multi-Label Binary Classification Model
$y_{true}, y_{pred} \in \mathbb{R}^{n_{test} \times (n_{loads}+1)}$
Evaluation
$I_{segments}$
$Y_{segments}$
$I_{segments} \in \mathbb{R}^{n_{total} \times n_{window}}$
$Y_{segments} \in \mathbb{R}^{n_{total} \times n_{loads}}$
80%
Split
20%
Features Extraction
Features Extraction
$F_{train} \in \mathbb{R}^{n_{train} \times n_{feats}}$
$F_{test} \in \mathbb{R}^{n_{test} \times n_{feats}}$
Training
Prediction
Multi-Label Binary Classification Model
$y_{test}, y_{pred} \in \mathbb{R}^{n_{test} \times n_{loads}}$
Evaluation
$I_{segments}$
$Y_{segments}$
$I_{segments} \in \mathbb{R}^{n_{total} \times n_{window}}$
$Y_{segments} \in \mathbb{R}^{n_{total} \times 1}$
80%
Split
20%
Features Extraction
Features Extraction
$F_{train} \in \mathbb{R}^{n_{train} \times n_{feats}}$
$F_{test} \in \mathbb{R}^{n_{test} \times n_{feats}}$
Training
Prediction
Binary Classification Model
$y_{test}, y_{pred} \in \mathbb{R}^{n_{test} \times 1}$
Evaluation

(a) Loads and Inverter Disaggregation Experiment (LIDE).

(b) Loads Disaggregation Experiment (LDE).

(c) Inverter Detection Experiment (IDE).

Source: Own authorship.

Table 18 – F1-Score for the Inverter Detection Experiment (IDE).

Method	Individual F1-Score	Aggregated F1-Score	Entire F1-Score
ST-NILM	100%	98%	97%
DeepDFML (NOLASCO *et al.*, 2022)	100%	99%	99%

For each experiment (LIDE, LDE and IDE), we extract the features of $I_{segments}$ using both the ST-NILM method (which we propose in Chapter 5) and the DeepDFML (NOLASCO *et al.*, 2022) (previously presented in Chapter 5) architecture. We performed the experiments LDE, LIDE, and IDE with a personal computer with the following configurations: Processor Intel Core I7-4790 (four cores and eight threads); 24Gb of DDR3 RAM; GPU NVidia Quadro K620; Linux Ubuntu 20.04 operational system.

6.3.2.1 Inverter Detection Experiment (IDE)

The Inverter Detection Experiment consists of a binary classification problem in which the only binary output represents the presence or absence of the inverter. To assemble this strategy, we consider the binary labels that represent the presence of the inverter in each subset of segments of the original dataset (aggregated, individual, or entire dataset). We separate 80% from segments for training and 20% for testing into each subset of the original dataset. For example, the binary vectors of training and test labels for the aggregated subset are represented, respectively, by Y_{train}^{agg} and Y_{test}^{agg}.

Table 18 shows the F1-Scores for each subset of our proposed dataset and obtained for each method for IDE. The results show that all methods can precisely identify the presence of the inverter since the lowest F1-Score obtained with the IDE was 97$\%$. ST-NILM achieves classification performance equivalent to DeepDFML, considering that ST-NILM's F1-Score was at most 2$\%$ lower than DeepDFML (for the case of the Entire Dataset).

6.3.2.2 Loads Disaggregation Experiment (LDE)

For the LDE experiment, we intend to answer whether or not the DG inverter presence impairs the classification performance of electrical loads. We separate training and testing subsets for each condition (aggregated, individual or entire dataset) as follows:

- We separate the n_{train}^{with} segments obtained with DG from $I_{segments}$ and create the matrix $I_{train}^{dg} \in \mathbb{R}^{n_{train}^{with} \times n_{window}}$;

- We separate the $n_{train}^{without}$ segments obtained with DG from $I_{segments}$ and create the matrix $I_{train}^{ndg} \in \mathbb{R}^{n_{train}^{without} \times n_{window}}$;

- We separate the n_{test}^{with} segments obtained with DG from $I_{segments}$ and create the matrix $I_{test}^{dg} \in \mathbb{R}^{n_{test}^{with} \times n_{window}}$;

- We separate the $n_{test}^{without}$ segments obtained with DG from $I_{segments}$ and create the matrix $I_{test}^{ndg} \in \mathbb{R}^{n_{test}^{without} \times n_{window}}$;

- We define Y_{train}^{dg} and Y_{train}^{ndg} as the training labels matrices, related to I_{train}^{dg} and I_{train}^{ndg}, respectively;

- We define Y_{test}^{dg} and Y_{test}^{ndg} as the training labels matrices, related to I_{test}^{dg} and I_{test}^{ndg}, respectively;

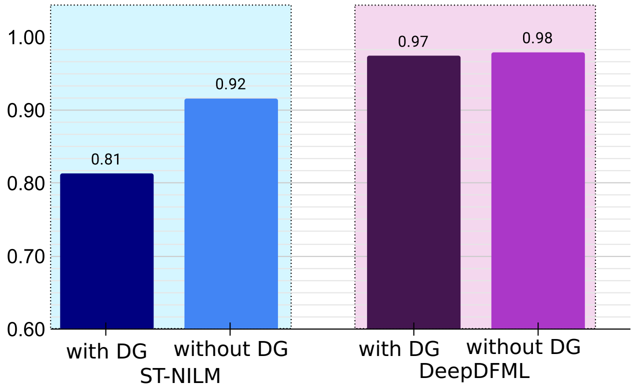
We set up the LDE classification structure as a multi-label binary classification problem with four binary targets (one per electrical load) without considering the inverter as a target. From the subsets obtained with the described procedure, we trained and tested some models according to the following methodology:

- ***Scenario LDE with DG:*** We use I_{train}^{dg}, with its labels Y_{train}^{dg} to train different classification models (with different tested feature extractors), and we test the trained models in the subset I_{test}^{dg} (with the respective labels Y_{test}^{dg});

- ***Scenario LDE without DG:*** We use I_{train}^{ndg}, with its labels Y_{train}^{ndg} to train different classification models (with different tested feature extractors), and we test the trained models in the subset I_{test}^{ndg} (with the respective labels Y_{test}^{ndg}).

Figure 40 shows the average F1-Score values obtained for the LDE. For the figure 40 we consider the average value of F1-Score, calculated from all classes, for each method.

Figure 40 – Average F1-Score for each method obtained from LDE.

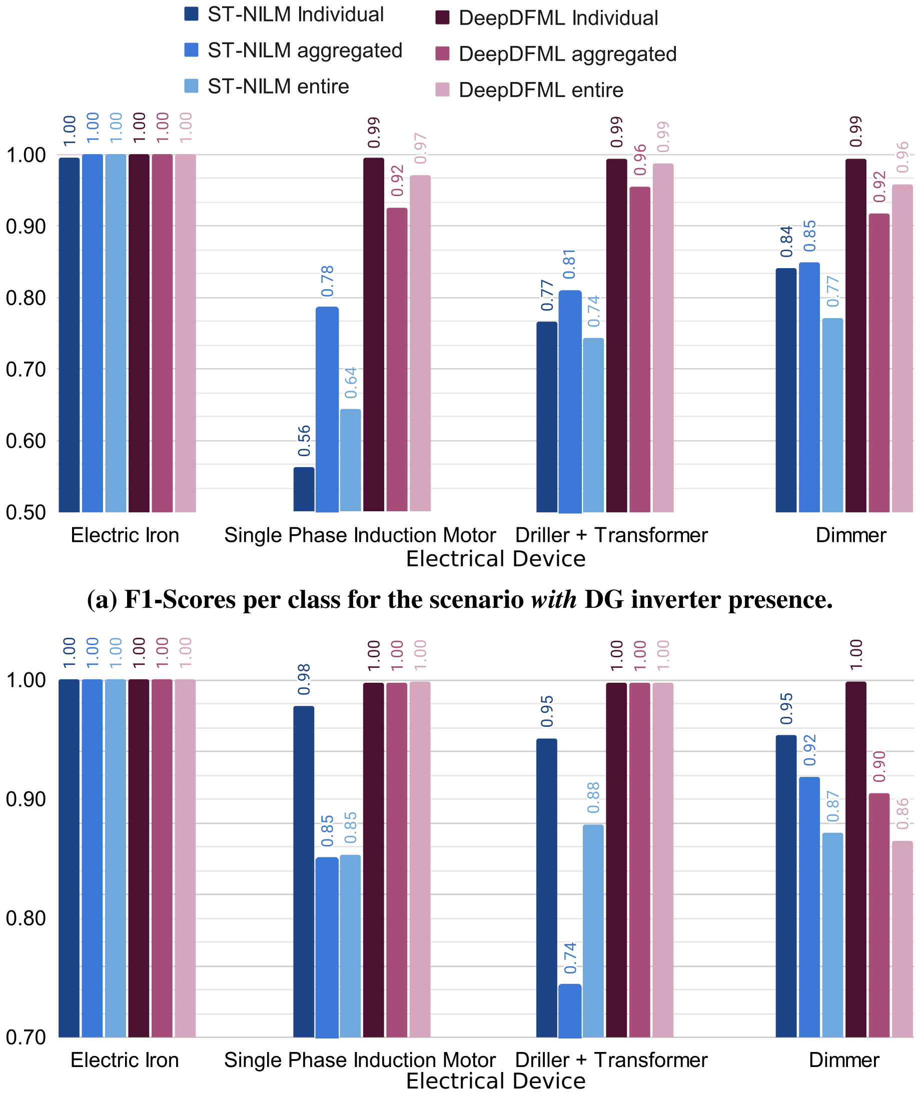


Source: Own authorship.

The results presented in Figure 40 show that: (i) The presence of DG impairs the classification performance for both methods; (ii) DeepDFML method presents better average performance than ST-NILM; (iii) The presence of the DG inverter affects less the classification performance for the DeepDFML than ST-NILM.

Figure 41 extends the analysis of Figure 40 to each class, showing the F1-Scores for each electric load in a stratified way. Figure 41(a) shows the results by class *with* DG and Figure 41(b) shows the results *without* DG.

Figure 41 – Classification per-class results for LDE, for both scenarios with and without DG presence.



(a) F1-Scores per class for the scenario *with* DG inverter presence.

(b) F1-Scores per class for the scenario *without* DG inverter presence.

Source: Own authorship.

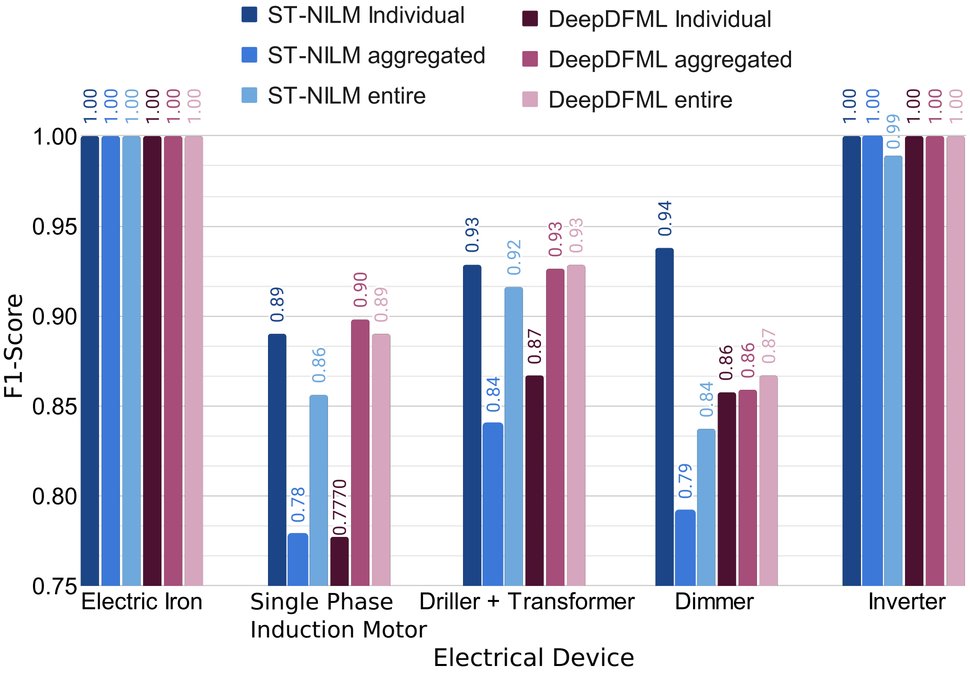
From Figure 41, one can conclude that: (i) The classification of the linear load (Electric Load) does not depend on the presence of the inverter, presenting F1-Score=1 for both methods, scenarios, and subsets; (ii) The methods suffered performance reduction in the classification of non-linear loads (Single-Phase Induction Motor, Driller+Transformer and Dimmer) when there is the presence of DG. The exception is Dimmer classification performance with the DeepDFML; (iii) The presence of the inverter had a less significant impact on DeepDFML than ST-NILM for classifying non-linear loads.

6.3.2.3 Loads and Inverter Disaggregation Experiment (LIDE)

The LIDE experiment makes it possible to analyze, together, the performance of the classification frameworks in the following aspects: (i) the ability to detect the presence or not of the inverter (DG); (ii) performance in the disaggregation of electrical loads in a residential installation with the presence of distributed generation.

Figure 42 presents the classification results for the LIDE tests. We trained four different frameworks for load classification together with the inverter. The results in shades of blue correspond to the ST-NILM, which we proposed in Chapter 5. The bars in shades of purple represent the F1-Scores obtained with DeepDFML (NOLASCO *et al.*, 2022).

Figure 42 – F1-Score Results for the Loads and Inverter Disaggregation Experiment (LIDE).



Source: Own authorship.

When analyzing the figure 42 one can notice that:

- Both methods achieved maximum performance in electrical iron classification. This result was expected since this appliance has linear characteristics and high power, which facilitates discriminability in the classification;

- The performances of both methods were worse for nonlinear loads (driller+transformer and dimmer) than linear loads. For these cases, our ST-NILM method surpassed DeepDFML (at least 6% higher F1-Score);

- For aggregated nonlinear loads, ST-NILM presented the worst performance among the analyzed methods. The worst performance under these conditions occurred for Dimmer (F1-Score of 79%);
- For the entire dataset and nonlinear appliances, the performance difference between our method (ST-NILM) and DeepDFML decreased. This behavior occurred both for Driller + Transformer, in which case ST-NILM performed equivalent to DeepDFML (1% lower);
- Both methods analyzed successfully identified the presence of the inverter.

When comparing the performance of the methods in LIDE among the three subsets (aggregate, individual, and integer), we can conclude that:

- For the single-phase induction motor, DeepDFML achieved the best performance for the aggregated subset. ST-NILM performed better with the individual dataset;
- The proposed method (ST-NILM) presented the same performance pattern for the Single Phase Induction Motor, Driller + Transformer, and Dimmer, presenting a better performance for the individual subset, a worse result for the aggregated subset, and an intermediate performance for the entire dataset;
- Differently from ST-NILM, DeepDFML presented the worst result for the individual subset for Single Phase Induction Motor, Driller + Transformer, and Dimmer loads.

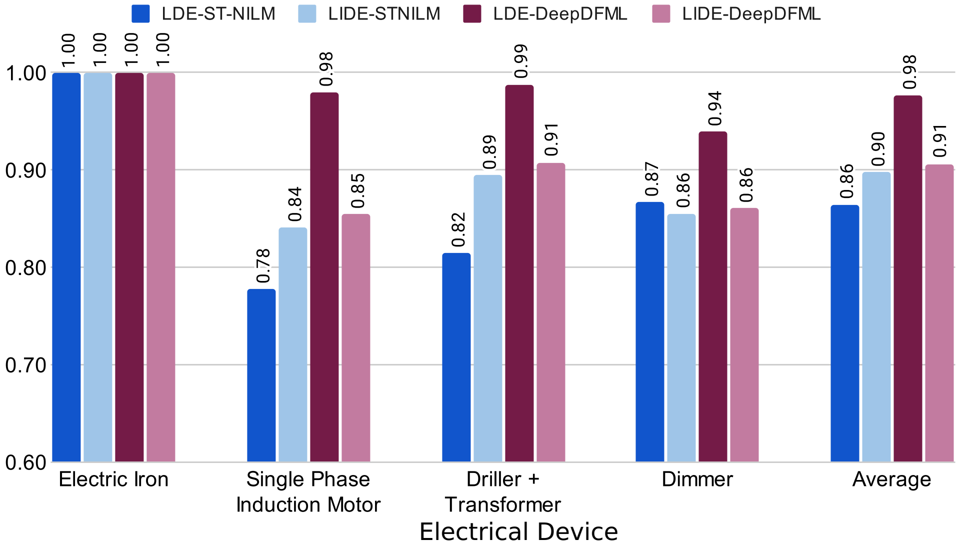
6.3.2.4 Comparisons between LDE, IDE, and LIDE

This Subsection compares the results obtained with the LDE, IDE, and LIDE experiments. Comparing the classification results obtained for load identification (LDE) with those obtained with loads *and* DG identification (LIDE), we can verify whether the presence of DG interferes with the multi-label classification of loads. On the other hand, when we compare the binary inverter identification experiments (IDE) with the inverter *and* loads detection experiment (LIDE), we can verify the performance variation in the inverter detection when we change the classification framework (IDE is a single-output binary classification framework and LIDE is a multi-output multi-label binary classification framework).

Figure 43 compares the average F1-Scores (obtained by averaging the F1-Scores between the three subsets for each appliance) of the LDE and LIDE experiments. As shown in

Figure 43: (i) Both methods achieved maximum performance for Electric Iron classification, which is a linear load; (ii) Unlike ST-NILM, DeepDFML performed worse (up to 13% worse) for LIDE than for LDE for the classification of Single Phase Induction Motor and Driller + Transformer (two non-linear loads with magnetic saturation); (iii) On average, classification performance worsens with the presence of the inverter for DeepDFML, but improves for ST-NILM.

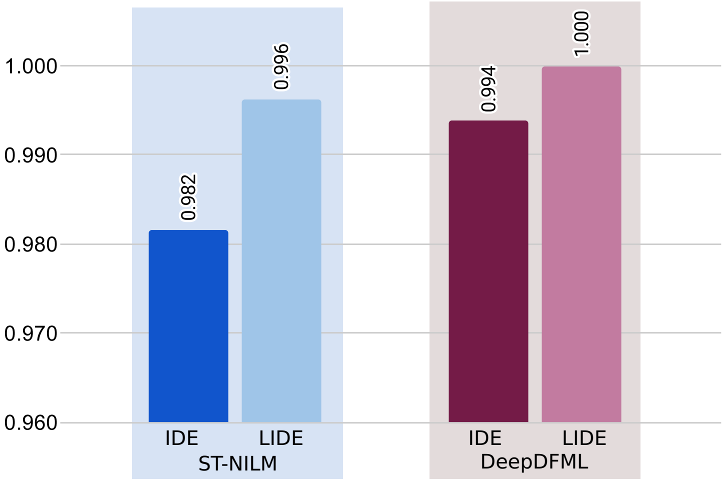
Figure 43 – FScore comparison between LDE and LIDE to determine the influence of the presence of the inverter on the classification of electrical loads



Source: Own arthorship.

Figure 44 compares the average F1-Scores, calculated among all subsets and for each method, obtained with the IDE and LIDE experiments. Each average F1-Score represents the classification performance of the DG inverter identification only, even for LIDE. The results indicate that: (i) Both tested methods identified the inverter with F1-Score greater than 98%, regardless of the framework; (ii) The multi-label framework, resulting in LIDE, leads to better classification performance for ST-NILM and DeepDFML.

Figure 44 – FScore comparison between IDE and LIDE to determine the influence of the classification framework on the inverter classification performance. The IDE results are obtained from a single output binary classification framework, and LIDE are obtained from a multi-label binary classification framework.



Source: Own authorship.

6.4 ANSWERS TO RESEARCH QUESTIONS RQ1 AND RQ2

Here we discuss the answers to RQ1 and RQ2:

- *Discussion regarding RQ1:* When we compare the average performance of ST-NILM with (LIDE) and without (LDE) inverter, we notice that we achieved better performance with inverter (LIDE) than without inverter (LDE). The opposite behavior occurred with DeepDFML, and for this method, the average FScore was better for the case without an inverter (LDE) than for the case with an inverter (LIDE). In addition, we observed the highest average performance variation between LIDE and LDE for the DeepDFML method (8%), indicating that the classification of loads for the DeepDFML method is more affected by the presence of the inverter than ST-NILM;

- *Discussion regarding RQ2:* The results of IDE and LIDE, compiled in the comparison of Figure 44, show that identifying the DG inverter was possible and doable, with an average F1-Score greater than 98% for both methods tested.

The average classification performance of ST-NILM was worse than DeepDFML for the performed experiments. It should be noted that ST-NILM is a smaller network than DeepDFML, with fewer layers and fewer convolutional filters. Despite having proven to be less discriminative than DeepDFML for NILM + BTM classification, ST-NILM is a convolutional network with untrained filters, which requires less computational effort and may be more feasible for real-time applications.

6.5 CONCLUSION OF THIS CHAPTER

NILM and BTM identification have a lot in common. We propose a new dataset with data from both electrical loads and DG generation, allowing us to unite the discussions of these two areas. We also propose three experiments to validate our proposed dataset and present state-of-the-art classification results for our dataset. We chose state-of-the-art classification methods involving both spectral (ST-NILM) and DCN-based (DeepDFML) strategies. Our experiments innovatively showed the influence of the presence of DG along with the aggregate loads on the load classification performance. Our results showed that the PV inverter impairs the classification of loads with magnetic saturation. One possibility for this is due to the coupling of the intrinsic inductances of these loads with the characteristic inductance of the weak network Aguiar (2014) that represents the local distribution network of the POLITEC laboratory and with the inductance resulting from the DC filter of the inverter.

We also evaluated the influence of aggregate loads on classification performance in identifying the presence of DG. We obtained a DG identification performance (average F1-Score) higher than 98% for all tested methods. This result points to possible applications of IDE for energy management systems in electricity utilities, in order to: (i) detect clandestine distributed generation connections; (ii) increase the observability of the generation network connected to the distribution bus; (iii) collaborate in the sizing, planning and expansion of the distribution network; (iv) generate information for power system operating agents; (v) improve the selectivity of protection.

The most significant impact on the average load classification performance by the presence of the inverter is 8%, which occurred with the DeepDFML method. Our results also showed that load classification performance is more affected by the presence of DG than DG identification is affected by the presence of aggregated loads.

The NILM-DG-V1 dataset has AWs of 16s. Although there are 16s of information for each example, our frameworks used smaller windows, the size of a grid (166ms). Other frameworks based on our experiments could also use context information to extract characteristics from the aggregated signal, contributing to the possible increase in discriminability. Furthermore, another valid discussion is about the resolution of analog-to-digital converters: we used the internal converters of the ESP32 microcontroller, which present nonlinearities close to the maximum and minimum allowable voltage limits, in addition to having a maximum resolution of 12 bits. Although many converters on the market have better specifications than this, the classification results, even with relatively low resolution and nonlinearities in acquisitions, are valid since the cost of implementing practical solutions with higher specifications is also significantly higher.

The results of LDE, LIDE, and IDE frameworks constitute the main contribution of this thesis. These results can serve as a benchmark for future comparisons with other load identification and DG detection methods, in addition to serving as a basis for restructuring the frameworks to include other loads, other forms of DG, and other tasks.

7 CONCLUSIONS AND FUTURE WORKS

In Chapter 3, we proposed six feature selectors and a framework for extracting features with the Scattering Transform. We used subsets of training data with fewer examples per class and subsampling. The classification results for simple loads proved to be better than state-of-the-art methods for most tested scenarios. Furthermore, we showed that the separability of the classes with the features obtained by the Scattering Transform is better than the traditional wavelet (baseline) for the tested cases. These promising results encouraged us to investigate further, including multiple loads and disaggregation. The results presented in Chapter 3 meet specific objectives numbers 1-5 and 8.

In Chapter 4, we proposed a new framework to extract features with the Scattering Transform. In this framework we used multiple loads (aggregated). We tested the performance of the Scattering Transform considering the electrical signals input in different temporal regions (steady-state, transient regime, or both). The classification results showed that the ST is more robust than the baselines regarding signal length, sampling frequency, and event location. The results shown in Chapter 4 meet specific objectives numbers 6-8.

The frameworks proposed in Chapters 3 and 4 considered the traditional diagram for NILM shown in Figure 3. Our proposal in the Chapter 5 improved the traditional NILM strategy since it is a multi-task and multi-label approach. In this approach, we perform classification and disaggregation in a single step. We proposed the ST-NILM architecture, based on DeepDFML (NOLASCO *et al.*, 2022). We showed that ST-NILM is less computationally costly than DeepDFML. The FLOPS for ST-NILM, as shown in Table 17, are still high compared to some embedded methods in the literature. However, we significantly reduced computational complexity compared to DeepDFML (0.264G for ST -NILM versus 7.643G for DeepDFML). Also, the classification results were better in ST-NILM than in DeepDFML, especially in cases where we used less data for training. This result corroborates the theory presented by (MALLAT, 2012), and meet specific objectives numbers 8-9.

In Section 6.2, Figure 31 and 32, we proposed an experimental arrangement that allowed flexible NILM and DG variables measuring. Our proposal made it possible to choose the exact times for connecting and disconnecting multiple loads, in addition to allowing the connection of a PV micro-inverter to the distribution voltage PCC. This development meets specific objective 10.

Motivated by the lack of datasets mixing NILM and DG acquisitions discussed in Subsection 2.4.2, in the Subsection 6.2.2, we selected a set of appliances and elaborated a methodology for data logging. As a result, we constructed a novel public dataset named DG-NILM-V1. We provided a public repository containing our dataset to meet specific objective 11.

In Section 6.3, we validated our proposed dataset by comparing the results of NILM load classification and disaggregation using DG-NILM-V1 subsets for training and testing. We performed feature extraction and classification with four state-of-the-art methods. This validation meets the specific objective 12.

To meet specific objective 13, we divided the proposed dataset validation into three experiments: IDE, LIDE, and IDE, building an ablation study (Subsection 6.3.2) to investigate the influence of DG in NILM classification and vice versa. We obtained classification F1-Scores above 98% with our dataset for DG identification, and we concluded in Section 6.5 that load classification performance is more affected by the presence of DG than DG identification is affected by the presence of aggregated loads.

We showed in Table 1 that the results presented in Chapter 4 were published in a scientific paper in the journal Energies MDPI. Furthermore, the results of Chapter 3 were presented in a scientific paper published in the journal LNLM. The results presented in Chapter 5 were compiled into a paper submitted to IEEE Sensors journal. Finally, our novel dataset, ablation study, and the related conclusions in Chapter 6 were compiled into a paper and submitted to the IEEE Transactions on Smart Grids. These submissions meet specific objective 14.

7.1 FUTURE WORKS AND OPEN CHALLANGES

Even though our proposal in Chapter 3 has obtained state-of-the-art results for feature selection with ST, in future works, we intend to evaluate the general performance of extraction methods, also considering the performance of a detection method. Furthermore, we suggest a future ablation study to evaluate the power signature of the inverter analogously to an electrical load. A possible approach is to extract features with the methods proposed in Chapter 3 and compare them with other features selection methods.

Despite the good results of ST-NILM compared to other state-of-the-art methods based on Deep Learning, some future improvements can be considered: (i) Improve the GPU implementation of ST-NILM; (ii) Improve the feature selection techniques in order to improve the PC_{off} in the case of more aggregate loads; (iii) Improve the classification and detection comparisons with other time-frequency-based feature extractors; (iv) Evaluate the ST-NILM with other real-world datasets; (v) Evaluate the ST-NILM with real-time data.

Although the results presented in Chapter 6 have shown encouraging endeavors in this field, including DG with NILM still requires numerous efforts. We focus our results on the disaggregation and classification of appliances and DG. In the future, it is desirable to expand the analysis to detect the instants of connection and disconnection of the appliances and the inverter. Detection of the *on* and *off* time instants is a different challenge than the one we address, which is also essential in the context of NILM and DG. In addition, we intend to obtain classification results with other state-of-the-art methods, including methods based on Transformers (VASWANI *et al.*, 2017).

With continuity and new versions of the NILM-DG-v1 dataset, some open challenges can be raised: (i) expanding our dataset, inserting more samples in different meteorological conditions and with an unbalanced number of examples per class; (ii) Increasing the number of appliances, allowing the evaluation of life-long learning methods; (iii) Evaluating the classification performance under unbalanced conditions; (iv) Expanding the number of AW, and testing the classification methods without using stride; (v) Identify the DG's modes of operation; (vi) Elaborate and evaluate a framework considering both I_{inv} and I_{agg} for training (instead of only I_{agg}) as a regression task to estimate the amount of distributed generated power from the aggregated current load.

In Brazil, the entity responsible for managing, dispatching energy generation and monitoring the interconnected system of generation and transmission of electrical energy is the ONS (acronym in Portuguese for "National System Operator"). Currently, ONS does not directly observe micro and mini DG systems connected to the distribution network. This means that, for example, when a sufficient number of prosumers stop generating energy at the same time, this is noticed by the ONS only due to the effect caused by the imbalance between generation and consumption, characterized in this case by a perception of increased load. This perception of an increase in load can lead to mistaken decision-making, since in fact what happened was a drop in generation, not an increase in load. For these cases, the DG detection that we proposed in this work using the LIDE and IDE frameworks can be useful, helping to mitigate the reported problem.

We use the aggregated current for the frameworks with the NILM-DG-V1 dataset. In many real cases, however, the available quantity is power, not current. We intend to address this line of investigation in the future, also involving the inclusion of power measurements in new versions of our dataset. Besides that, we intend to get a new subset for NILM-DG dataset with an external bidirectional power alternated current source, emulating an ideal distribution network scenario. One may better understand the local distribution line characteristics and parameters when comparing LIDE, IDE, and LDE results with this new subset.

REFERENCES

ABBOOD, Hayder D.; BENIGNI, Andrea. Data-Driven Modeling of a Commercial Photovoltaic Microinverter. **Modelling and Simulation in Engineering**, v. 2018, 2018. ISSN 16875605.

AFZALAN, Milad; JAZIZADEH, Farrokh; WANG, Jue. Self-configuring event detection in electricity monitoring for human-building interaction. **Energy and Buildings**, Elsevier B.V., v. 187, p. 95–109, 2019. ISSN 03787788. Available at: https://doi.org/10.1016/j.enbuild.2019.01.036.

AGUIAR, Everton Luiz de. **Análise, projeto e implementação de um sistema monofásico de geração distribuída conectado a redes fracas**. 2014. Master's Thesis (Master's Thesis) — Universidade Tecnológica Federal do Paraná, 2014.

AGUIAR, Everton Luiz de; LAZZARETTI, Andre Eugenio; MULINARI, Bruna Machado; PIPA, Daniel Rodrigues. Scattering transform for classification in non-intrusive load monitoring. **Energies**, v. 14, n. 20, 2021b. ISSN 1996-1073. Available at: https://www.mdpi.com/1996-1073/14/20/6796.

AGUIAR, Everton Luiz de; LAZZARETTI, Andre Eugenio; PIPA, Daniel Rodrigues. Performance of scattering transform feature extraction for electrical load classification. *In*: FILHO, Carmelo Jose Albanez Bastos; SIQUEIRA, Hugo Valadares; FERREIRA, Danton Diego; BERTOL, Douglas Wildgrube; OLIVEIRA, Roberto C'elio Lim ao de (Ed.). **Anais do 15 Congresso Brasileiro de Inteligência Computacional**. Joinville, SC: SBIC, 2021a. p. 1–8.

AGUIAR, Everton Luiz de; LAZZARETTI, Andre Eugenio; PIPA, Daniel Rodrigues. Features extraction and selection with the scattering transform for electrical load classification. **Learning & Nonlinear Models**, SBIC, v. 21, n. 1, p. 19–35, 2023.

ALCALA, Jose; URENA, Jesus; HERNANDEZ, Alvaro; GUALDA, David. 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, 2017. ISSN 00189456.

ANCELMO, H. C.; Grando, F. L.; Mulinari, B. M.; da Costa, C. H.; Lazzaretti, A. E.; Oroski, E.; Renaux, D. P. B.; Pottker, F.; Lima, C. R. E.; Linhares, R. R. A transient and steady-state power signature feature extraction using different prony's methods. *In*: **2019 20th International Conference on Intelligent System Application to Power Systems (ISAP)**. *[S.l.: s.n.]*, 2019. p. 1–6.

ANDÉN, Joakim; LOSTANLEN, Vincent; MALLAT, Stéphane. Joint Time-Frequency Scattering. **IEEE Transactions on Signal Processing**, v. 67, n. 14, p. 3704–3718, 2019. ISSN 19410476.

ANDERSON, Kyle D.; OCNEANU, Adrian; CARLSON, Derrick R.; ROWE, Anthony G.; BERGES, Mario E. Blued : A fully labeled public dataset for event-based non-intrusive load monitoring research. *In*: . *[s.n.]*, 2012. Available at: https://api.semanticscholar.org/CorpusID: 25397318.

ANDREUX, Mathieu; ANGLES, TomÃ¡s; EXARCHAKIS, Georgios; LEONARDUZZI, Roberto; ROCHETTE, Gaspar; THIRY, Louis; ZARKA, John; MALLAT, Stèphane; ANDEN, Joakim; BELILOVSKY, Eugene; BRUNA, Joan; LOSTANLEN, Vincent; CHAUDHARY, Muawiz; HIRN, Matthew J.; OYALLON, Edouard; ZHANG, Sixin; CELLA, Carmine; EICKENBERG, Michael. Kymatio: Scattering transforms in python. **Journal of Machine Learning Research**, v. 21, n. 60, p. 1–6, 2020. Available at: http://jmlr.org/papers/v21/19-047.html.

ANGELIS, Georgios Fotios; TIMPLALEXIS, Christos; KRINIDIS, Stelios; IOANNIDIS, Dimosthenis; TZOVARAS, Dimitrios. NILM applications: Literature review of learning approaches, recent developments and challenges. **Energy and Buildings**, Elsevier B.V., v. 261, n. 636302, p. 111951, 2022. ISSN 03787788. Available at: https://doi.org/10.1016/j.enbuild.2022.111951.

ATHANASIADIS, Christos; DOUKAS, Dimitrios; PAPADOPOULOS, Theofilos; CHRYSOPOULOS, Antonios. A scalable real-time non-intrusive load monitoring system for the estimation of household appliance power consumption. **Energies**, v. 14, n. 3, 2021. ISSN 19961073.

BAETS, Leen De; RUYSSINCK, Joeri; DEVELDER, Chris; DHAENE, Tom; DESCHRIJVER, Dirk. Appliance classification using VI trajectories and convolutional neural networks. **Energy and Buildings**, Elsevier B.V., v. 158, p. 32–36, 2018. ISSN 03787788.

BARKER, S.; MISHRA, A.; IRWIN, D.; CECCHET, E.; SHENOY, P.; ALBRECHT, J. Smart*: An Open Data Set and Tools for Enabling Research in Sustainable Homes. **SustKDD**, n. August, p. 6, 2012. Available at: http://lass.cs.umass.edu/papers/pdf/sustkdd12-smart.pdf\%5Cnhttp: //traces.cs.umass.edu/index.php/Smart/Smart.

BASU, Kaustav; DEBUSSCHERE, Vincent; DOUZAL-CHOUAKRIA, Ahlame; BACHA, Seddik. Time series distance-based methods for non-intrusive load monitoring in residential buildings. **Energy and Buildings**, v. 96, p. 109 – 117, 2015. ISSN 0378-7788. Available at: http://www.sciencedirect.com/science/article/pii/S0378778815002133.

BASU, Kaustav; HABLY, Ahmad; DEBUSSCHERE, Vincent; BACHA, Seddik; DRIVEN, Geert Jan; OVALLE, Andres. A comparative study of low sampling non intrusive load dis-aggregation. **IECON Proceedings (Industrial Electronics Conference)**, p. 5137–5142, 2016.

BATRA, Nipun; GULATI, Manoj; SINGH, Amarjeet; SRIVASTAVA, Mani B. It's different: Insights into home energy consumption in India. **BuildSys 2013 - Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings**, n. August, 2013.

BATRA, Nipun; PARSON, Oliver; BERGES, Mario; SINGH, Amarjeet; ROGERS, Alex. A comparison of non-intrusive load monitoring methods for commercial and residential buildings. **arXiv:1408.6595**, 2014.

BECKEL, Christian; KLEIMINGER, Wilhelm; CICCHETTI, Romano; STAAKE, Thorsten; SANTINI, Silvia. The ECO data set and the performance of non-intrusive load monitoring algorithms. **BuildSys 2014 - Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings**, p. 80–89, 2014.

BERGES, Mario; GOLDMAN, Ethan; MATTHEWS, H. Scott; SOIBELMAN, Lucio; ANDERSON, Kyle. User-Centered Nonintrusive Electricity Load Monitoring for Residential Buildings. **Journal of Computing in Civil Engineering**, v. 25, n. 6, p. 471–480, 2011. ISSN 0887-3801.

BISCHOF, Simon; TRITTENBACH, Holger; VOLLMER, Michael; WERLE, Dominik; BLANK, Thomas; BöHM, Klemens. Hipe: An energy-status-data set from industrial production. *In*: **Proceedings of the Ninth International Conference on Future Energy Systems**. New York, NY, USA: Association for Computing Machinery, 2018. (e-Energy '18), p. 599–603. ISBN 9781450357678. Available at: https://doi.org/10.1145/3208903.3210278.

BONAMENTE, Massimiliano. **Statistics and Analysis of Scientific Data**. 3. ed. *[S.l.]*: Springer Singapore, 2022. (Graduate Texts in Physics). ISBN 978-981-19-0365-6.

BOUHOURAS, Aggelos S.; GKAIDATZIS, Paschalis A.; PANAGIOTOU, Evangelos; POULAKIS, Nikolaos; CHRISTOFORIDIS, Georgios C. A NILM algorithm with enhanced disaggregation scheme under harmonic current vectors. **Energy and Buildings**, Elsevier B.V., v. 183, p. 392–407, 2019. ISSN 03787788. Available at: https://doi.org/10.1016/j.enbuild.2018.11.013.

BROWN, Joe; ABATE, Alessandro; ROGERS, Alex. Disaggregation of household solar energy generation using censored smart meter data. **Energy and Buildings**, Elsevier, v. 231, p. 110617, 2021. ISSN 03787788. Available at: https://doi.org/10.1016/j.enbuild.2020.110617.

BRUNA, Joan; MALLAT, Stéphane. Classification with scattering operators. *In*: **CVPR 2011**. *[S.l.: s.n.]*, 2011. p. 1561–1566.

BRUNA, Joan; MALLAT, Stéphane. **Invariant Scattering Convolution Networks**. 2012.

BURRUS, Charles; GOPINATH, R.; GUO, H. Introduction to wavelets and wavelet transform—a primer. **Recherche**, v. 67, 01 1998.

CARUANA, Rich. Multitask learning. **Machine Learning**, v. 28, n. 1, p. 41–75, 1997. ISSN 1573-0565. Available at: https://doi.org/10.1023/A:1007379606734.

CHANG, Hsueh-Hsien; LIAN, Kuo-Lung; SU, Yi-Ching; LEE, Wei-Jen. Power-spectrum-based wavelet transform for nonintrusive demand monitoring and load identification. **IEEE Transactions on Industry Applications**, v. 50, n. 3, p. 2081–2089, 2014.

CHEN, Huan; WANG, Yue Hsien; FAN, Chun Hung. A convolutional autoencoder-based approach with batch normalization for energy disaggregation. **Journal of Supercomputing**, Springer US, n. 0123456789, 2020. ISSN 15730484.

CHEN, Junfeng; WANG, Xue; ZHANG, Xiaotian; ZHANG, Weihang. Temporal and Spectral Feature Learning with Two-Stream Convolutional Neural Networks for Appliance Recognition in NILM. **IEEE Transactions on Smart Grid**, IEEE, v. 13, n. 1, p. 762–772, 2022. ISSN 19493061.

CHEN, Kunjin; WANG, Qin; HE, Ziyu; CHEN, Kunlong; HU, Jun; HE, Jinliang. Convolutional sequence to sequence non-intrusive load monitoring. **arXiv**, p. 1–5, 2018. ISSN 23318422.

CHEN, Kunjin; ZHANG, Yu; WANG, Qin; HU, Jun; FAN, Hang; HE, Jinliang. Scale- And Context-Aware Convolutional Non-Intrusive Load Monitoring. **IEEE Transactions on Power Systems**, IEEE, v. 35, n. 3, p. 2362–2373, 2020. ISSN 15580679.

CHERKASSKY, Vladimir S.; MULIER, Filip. **Learning from Data: Concepts, Theory, and Methods**. 1st. ed. New York, NY, USA: John Wiley & Sons, Inc., 1998. ISBN 0471154938.

DAUBECHIES, Ingrid. The Wavelet Transform, Time-Frequency Localization and Signal Analysis. **IEEE Transactions on Information Theory**, v. 36, n. 5, p. 961–1005, 1990. ISSN 15579654.

DAUBECHIES, Ingrid; LU, Jianfeng; WU, Hau Tieng. Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool. **Applied and Computational Harmonic Analysis**, Elsevier Inc., v. 30, n. 2, p. 243–261, 2011. ISSN 10635203. Available at: http://dx.doi.org/10.1016/j.acha.2010.08.002.

DAVIS, S.; MERMELSTEIN, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 28, n. 4, p. 357–366, 1980.

Deluno Garcia, Fernando; Angelino De Souza, Wesley; Pinhabel Marafao, Fernando. Embedded NILM as Home Energy Management System: A Heterogeneous Computing Approach. **IEEE Latin America Transactions**, v. 18, n. 2, p. 360–367, 2020. ISSN 15480992.

DING, Dong; LI, Junhuai; ZHANG, Kuo; WANG, Huaijun; WANG, Kan; CAO, Ting. Non-intrusive load monitoring method with inception structured. Applied Intelligence, 2021.

DONG, M.; Meira, P. C. M.; Xu, W.; Chung, C. Y. Non-intrusive signature extraction for major residential loads. **IEEE Transactions on Smart Grid**, v. 4, n. 3, p. 1421–1430, 2013.

ERDENER, Burcin Cakir; FENG, Cong; DOUBLEDAY, Kate; FLORITA, Anthony; HODGE, Bri Mathias. A review of behind-the-meter solar forecasting. **Renewable and Sustainable Energy Reviews**, Elsevier Ltd, v. 160, n. November 2021, p. 112224, 2022. ISSN 18790690. Available at: https://doi.org/10.1016/j.rser.2022.112224.

FAUSTINE, Anthony; PEREIRA, Lucas. Improved Appliance Classification in Non-Intrusive Load Monitoring Using Weighted Recurrence Graph and Convolutional Neural Networks. **Energies**, v. 13, n. 13, 2020. ISSN 1996-1073. Available at: https://www.mdpi.com/1996-1073/13/13/3374.

FAUSTINE, Anthony; PEREIRA, Lucas. Multi-label learning for appliance recognition in NILM using fryze-current decomposition and convolutional neural network. **Energies**, v. 13, n. 6, 2020. ISSN 19961073.

GAO, Jingkun; GIRI, Suman; KARA, Emre Can; BERGÉS, Mario. PLAID: A public dataset of high-resolution electrical appliance measurements for load identification research. **BuildSys 2014 - Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings**, p. 198–199, 2014.

GAUTHIER, Shanel; THERIEN, Benjamin; ALSENE-RACICOT, Laurent; CHAUDHARY, Muawiz; RISH, Irina; BELILOVSKY, Eugene; EICKENBERG, Michael; WOLF, Guy. Parametric Scattering Networks. **Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, v. 2022-June, p. 5739–5748, 2022. ISSN 10636919.

GOMES, Eduardo; PEREIRA, Lucas. PB-NILM: Pinball guided deep non-intrusive load monitoring. **IEEE Access**, IEEE, v. 8, p. 48386–48398, 2020. ISSN 21693536.

GONCALVES, Calvin; BARRETO, Ruben; FARIA, Pedro; GOMES, Luis; VALE, Zita. Energy community consumption and generation dataset with appliance allocation. **IFAC-PapersOnLine**, v. 55, n. 9, p. 285–290, 2022. ISSN 2405-8963. 11th IFAC Symposium on Control of Power and Energy Systems CPES 2022. Available at: https://www.sciencedirect.com/science/article/pii/S2405896322004359.

GULATI, Manoj; RAM, Shobha Sundar; SINGH, Amarjeet. An in depth study into using EMI signatures for appliance identification. **BuildSys 2014 - Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings**, p. 70–79, 2014.

GUO, Luyang; WANG, Shouxiang; CHEN, Haiwen; SHI, Qingyuan. A load identification method based on active deep learning and discrete wavelet transform. **IEEE Access**, v. 8, p. 113932–113942, 2020.

GUTH, Florentin; ZARKA, John; MALLAT, Stéphane. Phase collapse in neural networks. **CoRR**, abs/2110.05283, 2021. Available at: https://arxiv.org/abs/2110.05283.

HART, George W. Nonintrusive Appliance Load Monitoring. **Proceedings of the IEEE**, v. 80, n. 12, p. 1870–1891, 1992. ISSN 15582256.

HASSAN, Taha; JAVED, Fahad; ARSHAD, Naveed. An empirical investigation of v-i trajectory based load signatures for non-intrusive load monitoring. **IEEE Transactions on Smart Grid**, v. 5, n. 2, p. 870–878, 2014.

HEBRAIL, G. **Individual household electric power consumption data set**. 2012. Available at: https://archive.ics.uci.edu/ml/datasets/.

HIMEUR, Yassine; ALSALEMI, Abdullah; BENSAALI, Faycal; AMIRA, Abbes. An intelligent nonintrusive load monitoring scheme based on 2D phase encoding of power signals. **International Journal of Intelligent Systems**, v. 36, n. 1, p. 72–93, 2021. ISSN 1098111X.

HOUIDI, Sarra; FOURER, Dominique; AUGER, François. On the use of concentrated time-frequency representations as input to a deep convolutional neural network: Application to non intrusive load monitoring. **Entropy**, v. 22, n. 9, p. 1–17, 2020. ISSN 10994300.

HUANG, Norden E; SHEN, Samuel S P. **Hilbert-huang Transform And Its Applications**. 2nd. ed. Singapore: World Scientific Publishing Co, 2014. 400 p. ISBN 978-981-4508-23=0.

HUBER, Patrick; CALATRONI, Alberto; RUMSCH, Andreas; PAICE, Andrew. Review on deep neural networks applied to low-frequency nilm. **Energies**, v. 14, n. 9, 2021. ISSN 19961073.

HWANG, Hyeontaek; KANG, Sanggil. Nonintrusive load monitoring using an lstm with feedback structure. **IEEE Transactions on Instrumentation and Measurement**, v. 71, p. 1–11, 2022.

JARAMILLO, Andres F. Moreno; LAVERTY, David M.; MORROW, D. John; RINCON, Jesùs Del; FOLEY, Aoife M. Load modelling and non-intrusive load monitoring to integrate distributed energy resources in low and medium voltage networks. **Renewable Energy**, Elsevier Ltd, v. 179, p. 445–466, 2021a. ISSN 18790682. Available at: https://doi.org/10.1016/j.renene.2021.07.056.

JARAMILLO, Andres F. Moreno; LAVERTY, David M.; RINCON, Jesùs Martínez Del; BROGAN, Paul; MORROW, D. John. Non-intrusive load monitoring algorithm for pv identification in the residential sector. *In*: **2020 31st Irish Signals and Systems Conference (ISSC)**. *[S.l.: s.n.]*, 2020. p. 1–6.

JARAMILLO, Andres F Moreno; LOPEZ-LORENTE, Javier; LAVERTY, David; RINCON, Jesus Martinez-del; FOLEY, Aoife M. Identification of distributed energy resources in low voltage distribution networks. *In*: **2021 IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)**. *[S.l.: s.n.]*, 2021b. p. 1–6.

JARAMILLO, Andres F. Moreno; LOPEZ-LORENTE, Javier; LAVERTY, David M.; RINCON, Jesus Martinez-del; MORROW, D. John; FOLEY, Aoife M. Effective identification of distributed energy resources using smart meter net-demand data. **IET Smart Grid**, v. 5, n. 2, p. 120–135, 2022. Available at: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/stg2.12056.

JARAMILLO, Andres F. Moreno; LOPEZ-LORENTE, Javier; LAVERTY, David M.; BROGAN, Paul V.; VELASQUEZ, Santiago H. Hoyos; MARTINEZ-DEL-RINCóN, Jesus; FOLEY, Aoife M. Distributed energy resources electric profile identification in low voltage networks using supervised machine learning techniques. **IEEE Access**, v. 11, p. 19469–19486, 2023.

JARAMILLO, Andres F Moreno; MOHAMED, Ahmed A. Raouf; LAVERTY, David; RINCON, Jesus Martinez del; FOLEY, Aoife M. Photovoltaic power disaggregation using a non-intrusive load monitoring regression model. *In*: **2021 IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)**. *[S.l.: s.n.]*, 2021c. p. 1–6.

JIA, Ziyue; YANG, Linfeng; ZHANG, Zhenrong; LIU, Hui; KONG, Fannie. Sequence to point learning based on bidirectional dilated residual network for non-intrusive load monitoring. **International Journal of Electrical Power and Energy Systems**, Elsevier Ltd, v. 129, n. January, p. 106837, 2021. ISSN 01420615. Available at: https://doi.org/10.1016/j.ijepes.2021.106837.

KAHL, Matthias; HAQ, Anwar Ul; KRIECHBAUMER, Thomas; JACOBSEN, Hans-arno. WHITED - A Worldwide Household and Industry Transient Energy Data Set. **3rd International Workshop on Non-Intrusive Load Monitoring (NILM2016)**, n. April, p. 1–4, 2016.

KASELIMI, Maria; PROTOPAPADAKIS, Eftychios; VOULODIMOS, Athanasios; DOULAMIS, Nikolaos; DOULAMIS, Anastasios. Multi-channel recurrent convolutional neural networks for energy disaggregation. **IEEE Access**, IEEE, v. 7, p. 81047–81056, 2019. ISSN 21693536.

KELLY, Jack; KNOTTENBELT, William. Neural NILM: Deep neural networks applied to energy disaggregation. **BuildSys 2015 - Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built**, p. 55–64, 2015.

KELLY, Jack; KNOTTENBELT, William. The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. **Scientific Data**, v. 2, n. 150007, 2015.

KHAN, Salman; RAHMANI, Hossein; SHAH, Syed Afaq Ali; BENNAMOUN, Mohammed. A Guide to Convolutional Neural Networks for Computer Vision. **Synthesis Lectures on Computer Vision**, v. 8, n. 1, p. 1–207, 2018. ISSN 2153-1056.

KIRANYAZ, Serkan; AVCI, Onur; ABDELJABER, Osama; INCE, Turker; GABBOUJ, Moncef; INMAN, Daniel J. 1D convolutional neural networks and applications: A survey. **Mechanical Systems and Signal Processing**, The Author(s), v. 151, p. 107398, 2021. ISSN 10961216. Available at: https://doi.org/10.1016/j.ymssp.2020.107398.

KLEMENJAK, Christoph; KOVATSCH, Christoph; HEROLD, Manuel; ELMENREICH, Wilfried. A synthetic energy dataset for non-intrusive load monitoring in households. **Scientific Data**, Springer US, v. 7, n. 1, p. 1–17, 2020. ISSN 20524463. Available at: http://dx.doi.org/10.1038/s41597-020-0434-6.

KOLTER, J. Z. Redd : A public data set for energy disaggregation research. *In*: . *[S.l.: s.n.]*, 2011.

KRIECHBAUMER, Thomas; JACOBSEN, Hans Arno. BLOND, a building-level office environment dataset of typical electrical appliances. **Scientific Data**, v. 5, p. 1–14, 2018. ISSN 20524463.

LAOUALI, Inoussa; RUANO, Antonio; RUANO, Maria da Graça; BENNANI, Saad Dosse; El Fadili, Hakim. Non-Intrusive Load Monitoring of Household Devices Using a Hybrid Deep Learning Model through Convex Hull-Based Data Selection. **Energies**, v. 15, n. 3, 2022. ISSN 19961073.

LAUGHMAN, Christopher; LEE, Kwangduk; COX, Robert; SHAW, Steven; LEEB, Steven; NORFORD, Les; ARMSTRONG, Peter. Power signature analysis. **IEEE Power and Energy Magazine**, v. 1, n. 2, p. 56–63, 2003. ISSN 15407977.

LAZZARETTI, André Eugenio; RENAUX, Douglas Paulo Bertrand; LIMA, Carlos Raimundo Erig; MULINARI, Bruna Machado; ANCELMO, Hellen Cristina; OROSKI, Elder; PÖTTKER, Fabiana; LINHARES, Robson Ribeiro; da Silva Nolasco, Lucas; LIMA, Lucas Tokarski; OMORI, Júlio Shigeaki; SANTOS, Rodrigo Braun dos. A multi-agent NILM architecture for event detection and load classification. **Energies**, v. 13, n. 17, p. 1–37, 2020. ISSN 19961073.

LEEB, S. B.; SHAW, S. R.; KIRTLEY, J. L. Transient event detection in spectral envelope estimates for nonintrusive load monitoring. **IEEE Transactions on Power Delivery**, v. 10, n. 3, p. 1200–1210, 1995.

LIN, Jun; MA, Jin; ZHU, Jianguo. A Privacy-Preserving Federated Learning Method for Probabilistic Community-Level Behind-the-Meter Solar Generation Disaggregation. **IEEE Transactions on Smart Grid**, IEEE, v. 13, n. 1, p. 268–279, 2022. ISSN 19493061.

LIN, Yu Hsiu; TSAI, Men Shen. Development of an improved time-frequency analysis-based nonintrusive load monitor for load demand identification. **IEEE Transactions on Instrumentation and Measurement**, IEEE, v. 63, n. 6, p. 1470–1483, 2014. ISSN 00189456.

LIU, Bo; LUAN, Wenpeng; YU, Yixin. Dynamic time warping based non-intrusive load transient identification. **Applied Energy**, Elsevier Ltd, v. 195, p. 634–645, 2017. ISSN 03062619. Available at: http://dx.doi.org/10.1016/j.apenergy.2017.03.010.

LIU, Hui. **Non-intrusive load monitoring: Theory, technologies and applications**. *[S.l.: s.n.]*, 2019. 1–277 p. ISBN 9789811518607.

LIU, Yanchi; WANG, Xue; ZHAO, Lin; LIU, Youda. Admittance-based load signature construction for non-intrusive appliance load monitoring. **Energy and Buildings**, Elsevier B.V., v. 171, p. 209–219, 2018. ISSN 03787788. Available at: https://doi.org/10.1016/j.enbuild.2018.04.049.

LU, Mengqi; LI, Zuyi. A Hybrid Event Detection Approach for Non-Intrusive Load Monitoring. **IEEE Transactions on Smart Grid**, v. 11, n. 1, p. 528–540, 2020. ISSN 19493061.

LUO, Dong; NORFORD, Leslie; SHAW, Steven; LEEB, S.B.; DANKS, R.; WICHENKO, G. Monitoring hvac equipment electrical loads from a centralized location - methods and field test results. **ASHRAE Transactions**, v. 108, p. 841–857, 01 2002.

MAATEN, Laurens van der; HINTON, Geoffrey. Visualizing data using t-SNE. **Journal of Machine Learning Research**, v. 9, p. 2579–2605, 2008. Available at: http://www.jmlr.org/papers/v9/vandermaaten08a.html.

MAKONIN, Stephen; POPOWICH, Fred; BARTRAM, Lyn; GILL, Bob; BAJIĆ, Ivan V. AMPds: A public dataset for load disaggregation and eco-feedback research. **2013 IEEE Electrical Power and Energy Conference, EPEC 2013**, n. Section III, 2013.

MAKONIN, Stephen; WANG, Z. Jane; TUMPACH, Chris. RAE: The rainforest automation energy dataset for smart grid meter data analysis. **Data**, v. 3, n. 1, p. 1–9, 2018. ISSN 23065729.

MALLAT, Stéphane. Group Invariant Scattering. **Communications on Pure and Applied Mathematics**, v. 65, n. 10, p. 1331–1398, 2012. ISSN 00103640.

MALLAT, Stephane G. A theory for multiresolution signal decomposition: The wavelet representation. **Fundamental Papers in Wavelet Theory**, II, n. 7, p. 494–513, 1989.

MARTINS, Pedro B. M.; GOMES, José G. R. C.; NASCIMENTO, Vagner B.; FREITAS, Antonio R. de. Application of a deep learning generative model to load disaggregation for industrial machinery power consumption monitoring. *In*: **2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)**. *[S.l.: s.n.]*, 2018. p. 1–6.

MASON, Karl; RENO, Matthew J.; BLAKELY, Logan; VEJDAN, Sadegh; GRIJALVA, Santiago. A deep neural network approach for behind-the-meter residential pv size, tilt and azimuth estimation. **Solar Energy**, v. 196, p. 260–269, 2020. ISSN 0038-092X. Available at: https://www.sciencedirect.com/science/article/pii/S0038092X19312101.

MASSIDDA, Luca; MARROCU, Marino; MANCA, Simone. Non-intrusive load disaggregation by convolutional neural network and multilabel classification. **Applied Sciences (Switzerland)**, v. 10, n. 4, 2020. ISSN 20763417.

MATINDIFE, Liston; SUN, Yanxia; WANG, Zenghui. Image-based mains signal disaggregation and load recognition. **Complex & Intelligent Systems**, Springer International Publishing, v. 7, n. 2, p. 901–927, 2021. ISSN 2199-4536. Available at: https://doi.org/10.1007/s40747-020-00254-0.

MEDICO, Roberto; De Baets, Leen; GAO, Jingkun; GIRI, Suman; KARA, Emre; DHAENE, Tom; DEVELDER, Chris; BERGÉS, Mario; DESCHRIJVER, Dirk. A voltage and current measurement dataset for plug load appliance identification in households. **Scientific Data**, v. 7, n. 1, p. 1–10, 2020. ISSN 20524463.

MEYER, Yves. **Wavelets and Operators**. *[S.l.]*: Cambridge University Press, 1993. v. 1. (Cambridge Studies in Advanced Mathematics, v. 1).

MICHAU, Gabriel; FRUSQUE, Gaetan; FINK, Olga. Fully learnable deep wavelet transform for unsupervised monitoring of high-frequency time series. **Proceedings of the National Academy of Sciences of the United States of America**, v. 119, n. 8, 2022. ISSN 10916490.

MONACCHI, Andrea; EGARTER, Dominik; ELMENREICH, Wilfried; D'ALESSANDRO, Salvatore; TONELLO, Andrea M. GREEND: An energy consumption dataset of households in Italy and Austria. **2014 IEEE International Conference on Smart Grid Communications, SmartGridComm 2014**, n. November, p. 511–516, 2015.

MORADZADEH, Arash; MOHAMMADI-IVATLOO, Behnam; ABAPOUR, Mehdi; ANVARI-MOGHADDAM, Amjad; Gholami Farkoush, Saeid; RHEE, Sang Bong. A practical solution based on convolutional neural network for non-intrusive load monitoring. **Journal of Ambient Intelligence and Humanized Computing**, v. 8, 2021. ISSN 18685145.

MORÁN, A; ALONSO, S; PÉREZ, D; PRADA, M A; FUERTES, José J; DOMÍNGUEZ, Manuel. Feature extraction from building submetering networks using deep learning. **Sensors**, v. 20, n. 13, 2020. ISSN 1424-8220.

MUKAROH, Afifatul; LE, Thi Thu Huong; KIM, Howon. Background load denoising across complex load based on generative adversarial network to enhance load identification. **Sensors (Switzerland)**, MDPI AG, v. 20, p. 1–23, 10 2020. ISSN 14248220.

MULINARI, B. M.; DE CAMPOS, D. P.; DA COSTA, C. H.; ANCELMO, H. C.; LAZZARETTI, A. E.; OROSKI, E.; LIMA, C. R. E.; RENAUX, D. P. B.; POTTKER, F.; LINHARES, R. R. A new set of steady-state and transient features for power signature analysis based on V-I trajectory. *In*: **IEEE PES Innovative Smart Grid Technologies Conference - Latin America (ISGT Latin America)**. *[S.l.: s.n.]*, 2019. p. 1–6.

MULINARI, Bruna Mac Hado; NOLASCO, Lucas Da Silva; OROSKI, Elder; LAZZARETTI, Andre Eugenio; LINHARES, Robson Ribeiro; RENAUX, Douglas Paulo Bertrand. Feature Extraction of V-I Trajectory Using 2-D Fourier Series for Electrical Load Classification. **IEEE Sensors Journal**, v. 22, n. 18, p. 17988–17996, 2022. ISSN 15581748.

MURRAY, David; STANKOVIC, Lina; STANKOVIC, Vladimir. An electrical load measurements dataset of United Kingdom households from a two-year longitudinal study. **Scientific Data**, v. 4, p. 1–12, 2017. ISSN 20524463.

NAMBI, Akshay S.N. Uttama; LUA, Antonio Reyes; PRASAD, Venkatesha R. Loced: Location-aware energy disaggregation framework. *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. 45–54. ISBN 9781450339810. Available at: https://doi.org/10.1145/2821650.2821659.

NOLASCO, Lucas Da Silva; LAZZARETTI, Andre Eugenio; MULINARI, Bruna Machado. DeepDFML-NILM: A New CNN-Based Architecture for Detection, Feature Extraction and Multi-Label Classification in NILM Signals. **IEEE Sensors Journal**, v. 22, n. 1, p. 501–509, 2022. ISSN 15581748.

OBERLIN, T.; MEIGNEN, S.; PERRIER, V. The fourier-based synchrosqueezing transform. *In*: **2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. *[S.l.: s.n.]*, 2014. p. 315–319.

PAN, Keda; CHEN, Zhaohua; LAI, Chun Sing; XIE, Changhong; WANG, Dongxiao; LI, Xuecong; ZHAO, Zhuoli; TONG, Ning; LAI, Loi Lei. An unsupervised data-driven approach for behind-the-meter photovoltaic power generation disaggregation. **Applied Energy**, Elsevier Ltd, v. 309, n. November 2021, p. 118450, 2022. ISSN 03062619. Available at: https://doi.org/10.1016/j.apenergy.2021.118450.

PARSON, Oliver; FISHER, Grant; HERSEY, April; BATRA, Nipun; KELLY, Jack; SINGH, Amarjeet; KNOTTENBELT, William; ROGERS, Alex. Dataport and NILMTK: A building data set designed for non-intrusive load monitoring. **2015 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2015**, IEEE, p. 210–214, 2016.

PICON, Thomas; MEZIANE, Mohamed Nait; RAVIER, Philippe; LAMARQUE, Guy; NOVELLO, Clarisse; BUNETEL, Jean-Charles Le; RAINGEAUD, Yves. COOLL: Controlled on/off loads library, a public dataset of high-sampled electrical signals for appliance identification. **arXiv preprint arXiv:1611.05803 [cs.OH]**, 2016.

POWERS, John; MARGOSSIAN, Bedig; SMITH, Bruce. 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, 1991. ISSN 08950156.

RAMíREZ-RAMíREZ, Alfredo; JIMéNEZ-REYES, Víctor F.; VéLEZ-ENRíQUEZ, Jorge A.; LEAL-ORTIZ, Simón; GARCíA-GUZMáN, Jesús. Study of harmonic content and influence of common electrical appliances used in residential installations. *In*: **2019 IEEE International Conference on Engineering Veracruz (ICEV)**. *[S.l.: s.n.]*, 2019. I, p. 1–8.

READ, Jesse; PFAHRINGER, Bernhard; HOLMES, Geoff; FRANK, Eibe. Classifier chains for multi-label classification. **Machine Learning**, v. 85, n. 3, p. 333–359, 2011. ISSN 08856125.

REDMON, Joseph; FARHADI, Ali. YOLO9000: Better, faster, stronger. **Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017**, v. 2017-January, p. 6517–6525, 2017.

REINHARDT, Andreas; BAUMANN, Paul; BURGSTAHLER, Daniel; HOLLICK, Matthias; CHONOV, Hristo; WERNER, Marc; STEINMETZ, Ralf. On the accuracy of appliance identification based on distributed load metering data. **2012 Sustainable Internet and ICT for Sustainability, SustainIT 2012**, IEEE, 2012.

RENAUX, D. P. B.; Lima, C. R. E.; PATTKER, F.; OROSKI, E.; LAZZARETTI, A. E.; LINHARES, R. R.; ALMEIDA, A. R.; COELHO, A. O.; HERCULES, M. C. Non-intrusive load monitoring: an architecture and its evaluation for power electronics loads. *In*: **2018 IEEE International Power Electronics and Application Conference and Exposition (PEAC)**. *[S.l.: s.n.]*, 2018. p. 1–6.

RENAUX, Douglas Paulo Bertrand; POTTKER, Fabiana; ANCELMO, Hellen Cristina; LAZZARETTI, André Eugenio; LIMA, Carlos Raiumundo Erig; LINHARES, Robson Ribeiro; OROSKI, Elder; da Silva Nolasco, Lucas; LIMA, Lucas Tokarski; MULINARI, Bruna Machado; SILVA, José Reinaldo Lopes da; OMORI, Júlio Shigeaki; SANTOS, Rodrigo Braun dos. A dataset for non-intrusive load monitoring: Design and implementation. **Energies**, 2020. ISSN 19961073.

RIBEIRO, Miguel; PEREIRA, Lucas; QUINTAL, Filipe; NUNES, Nuno. SustDataED: A Public Dataset for Electric Energy Disaggregation Research. n. Ict4s, p. 244–245, 2016.

RUANO, Antonio; HERNANDEZ, Alvaro; UREÃ±A, Jesus; RUANO, Maria; GARCIA, Juan. NILM techniques for intelligent home energy management and ambient assisted living: A review. **Energies**, MDPI, Open Access Journal, v. 12, n. 11, p. 1–29, 2019.

SADEGHIANPOURHAMAMI, N; RUYSSINCK, J; DESCHRIJVER, D; DHAENE, T; DEVELDER, C. Comprehensive feature selection for appliance classification in NILM. **Energy and Buildings**, v. 151, p. 98–106, 2017. ISSN 0378-7788. Available at: https://www.sciencedirect.com/science/article/pii/S0378778817314366.

SANGOI, Juliana May; GHISI, Enedir. Energy efficiency of water heating systems in single-family dwellings in brazil. **Water**, v. 11, n. 5, 2019. ISSN 2073-4441. Available at: https://www.mdpi.com/2073-4441/11/5/1068.

SAURA, José Ramón; PALACIOS-MARQUÉS, Daniel; ITURRICHA-FERNÁNDEZ, Agustín. Ethical design in social media: Assessing the main performance measurements of user online behavior modification. **Journal of Business Research**, v. 129, n. February, p. 271–281, 2021. ISSN 01482963.

SCHOLL, Stefan. **Fourier, Gabor, Morlet or Wigner: Comparison of Time-Frequency Transforms**. 2021.

SEJDIC, Ervin; DJUROVIC, Igor; JIANG, Jin. Time–frequency feature representation using energy concentration: An overview of recent advances. **Digital Signal Processing**, v. 19, n. 1, p. 153–183, 2009. ISSN 1051-2004. Available at: https://www.sciencedirect.com/science/article/pii/S105120040800002X.

SHAW, Steven R.; LEEB, Steven B.; NORFORD, Leslie K.; COX, Robert W. Nonintrusive load monitoring and diagnostics in power systems. **IEEE Transactions on Instrumentation and Measurement**, v. 57, n. 7, p. 1445–1454, 2008. ISSN 00189456.

SHIN, Changho; LEE, Eunjung; HAN, Jeongyun; YIM, Jaeryun; RHEE, Wonjong; LEE, Hyoseop. The enertalk dataset, 15 hz electricity consumption data from 22 houses in Korea. **Scientific Data**, Springer US, v. 6, n. 1 1, p. 1–13, 2019. ISSN 20524463. Available at: http://dx.doi.org/10.1038/s41597-019-0212-5.

SOUZA, Wesley A.; ALONSO, Augusto M.S.; BOSCO, Thais B.; GARCIA, Fernando D.; GONCALVES, Flavio A.S.; MARAFAO, Fernando P. Selection of features from power theories to compose nilm datasets. **Advanced Engineering Informatics**, v. 52, p. 101556, 2022. ISSN 1474-0346. Available at: https://www.sciencedirect.com/science/article/pii/S1474034622000301.

STAUDT, Volker. Fryze - buchholz - depenbrock: A time-domain power theory. *In*: **2008 International School on Nonsinusoidal Currents and Compensation**. *[S.l.: s.n.]*, 2008. p. 1–12.

SU, Y.; LIAN, K.; CHANG, H. Feature selection of non-intrusive load monitoring system using stft and wavelet transform. *In*: **2011 IEEE 8th International Conference on e-Business Engineering**. *[S.l.: s.n.]*, 2011. p. 293–298.

VAPNIK, Vladimir N. **Statistical Learning Theory**. *[S.l.]*: Wiley-Interscience, 1998.

VASWANI, Ashish; SHAZEER, Noam; PARMAR, Niki; USZKOREIT, Jakob; JONES, Llion; GOMEZ, Aidan N.; KAISER, Lukasz; POLOSUKHIN, Illia. **Attention Is All You Need**. 2017.

WANG, Fei; GE, Xinxin; DONG, Zengbo; YAN, Jichuan; LI, Kangping; XU, Fei; LU, Xiaoxing; SHEN, Hongtao; TAO, Peng. Joint Energy Disaggregation of Behind-the-Meter PV and Battery Storage: A Contextually Supervised Source Separation Approach. **IEEE Transactions on Industry Applications**, IEEE, v. 58, n. 2, p. 1490–1501, 2022. ISSN 19399367.

WILD, Benjamin; BARSIM, Karim Said; YANG, Bin. A new unsupervised event detector for non-intrusive load monitoring. *In*: **2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)**. *[S.l.: s.n.]*, 2015. p. 73–77.

WU, Qian; WANG, Fei. Concatenate convolutional neural networks for non-intrusive load monitoring across complex background. **Energies**, v. 12, n. 8, 2019. ISSN 19961073.

WU, Yuan Kang; LAI, Yi Hui; HUANG, Cheng Liang; PHUONG, Nguyen Thi Bich; TAN, Wen Shan. Artificial Intelligence Applications in Estimating Invisible Solar Power Generation. **Energies**, v. 15, n. 4, 2022. ISSN 19961073.

YANG, Chuan Choong; SOH, Chit Siang; YAP, Vooi Voon. Comparative study of event detection methods for nonintrusive appliance load monitoring. **Energy Procedia**, Elsevier B.V., v. 61, p. 1840–1843, 2014. ISSN 18766102. Available at: http://dx.doi.org/10.1016/j.egypro.2014.12.225.

YANG, Dongsheng; GAO, Xiaoting; KONG, Liang; PANG, Yongheng; ZHOU, Bowen. An Event-Driven Convolutional Neural Architecture for Non-Intrusive Load Monitoring of Residential Appliance. **IEEE Transactions on Consumer Electronics**, IEEE, v. 66, n. 2, p. 173–182, 2020. ISSN 15584127.

ZHAO, Bochao; STANKOVIC, Lina; STANKOVIC, Vladimir. On a Training-Less Solution for Non-Intrusive Appliance Load Monitoring Using Graph Signal Processing. **IEEE Access**, IEEE, v. 4, p. 1784–1799, 2016. ISSN 21693536.

ZHENG, Zhuang; CHEN, Hainan; LUO, Xiaowei. A supervised event-based non-intrusive load monitoring for non-linear appliances. **Sustainability (Switzerland)**, v. 10, n. 4, p. 1–28, 2018. ISSN 20711050.

ZHOU, Xinxin; FENG, Jingru; LI, Yang. Non-intrusive load decomposition based on CNN–LSTM hybrid deep learning model. **Energy Reports**, Elsevier Ltd, v. 7, p. 5762–5771, 2021. ISSN 23524847. Available at: https://doi.org/10.1016/j.egyr.2021.09.001.

ZHOU, Xiao; LI, Shujian; LIU, Chengxi; ZHU, Haojun; DONG, Nan; XIAO, Tianying. Non-intrusive load monitoring using a cnn-lstm-rf model considering label correlation and class-imbalance. **IEEE Access**, v. 9, p. 84306–84315, 2021.

ZIMMERMANN, Jean-Paul; EVANS, Matt; LINEHAM, Tim; GRIGGS, Jonathan; SURVEYS, Griggs; HARDING, Les; KING, Nicola; ROBERTS, Penelope. Household Electricity Survey: A study of domestic electrical product usage. **Intertek**, p. 600, 2012. Available at: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/208097/10043_R66141HouseholdElectricitySurveyFinalReportissue4.pdf.