HAL open science

Signal processing techniques for disaggregation of smart-meter energy data

Alaa Saleh

To cite this version:

Alaa Saleh. Signal processing techniques for disaggregation of smart-meter energy data. Electric power. Université de Haute Alsace - Mulhouse, 2019. English. NNT : 2019MULH2908 . tel-03576804

HAL Id: tel-03576804

https://theses.hal.science/tel-03576804v1

Submitted on 16 Feb 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université de Haute-Alsace

Université de Strasbourg

Année 2019 N° d'ordre :

Thèse

Présentée pour l'obtention du grade de

Docteur de l'Université de Haute-Alsace

MATHÉMATIQUES, SCIENCES DE L'INFORMATION ET DE L'INGÉNIEUR

Discipline : Électronique, Électrotechnique et Automatique

par

Alaa SALEH

Signal Processing Techniques for Disaggregation of smart-meter energy Data

Sous la Direction de : MCF HDR Djaffar OULD-ABDESLAM

et au sein de Institute for Smart Systems (ISS)

à Furtwangen University sous la co-direction de

Professeur Dirk BENYOUCEF

Soutenue publiquement le 18 Décembre 2019 devant le jury composé de:

M 1,	Professeur : Nadia STEINER	Rapporteur
M 2,	Professeur : Michael SCHMIDT	Rapporteur
M 3,	Professeur : Serge PIERFEDERICI	Examinateur
M 4,	Professeur : Patrice WIRA	Examinateur
M5,	Professeur : Dirk Benyoucef	
M6,	MCF HDR : Djaffar OULD-ABDESLAM	

Signal Processing Techniques for Disaggregation of smart-meter energy Data

Alaa Saleh

This dissertation is submitted for the degree of Doctor of Philosophy

November 2019

IRIMAS UNIVERSITÉ HAUTE-ALSACE
Institut de Recherche en Informatique, Mathématiques, Automatique et Signal

To my soulmate

Aula

and my little ones

Daniel & Celina

Acknowledgements

I would like to thank the following people:

- My supervisor Prof. Dirk Benyoucef, who gave me the freedom to explore new research areas and to experiment with different techniques, while providing frequent guidance and excellent advice.
- My supervisor Prof. Djaffar Ould Abdeslam, for his infinite kindness, clever insights and constant encouragement, especially in the last “meters”.
- My colleague and friend at ISS institute during my time there: Pirmin Held, for interesting discussions and fun time in the office and on the road trips.
- My friends: Hazem Sulyman, Somar Karheily, Mouhammad Alhumaidi, Majdi Msallam and Mahmood Eissa for the beneficial discussions in signal processing and machine learning topics, I learnt a lot from each of you.
- My brothers and sister: Zain, Salman and Bayan for being there for me whenever I ask, and for proofreading this dissertation.
- Most of all: I owe the greatest debt of gratitude to my mother, for all the support and love throughout my life.
- Last but not least: To my wonderful wife Aula, for her love, support and patience.

The work of this thesis was funded by the BMBF[a] as part of the I-Mon[b] project at Furtwangen University.

a) BMBF: Federal ministry of education and research (in German: BundesMinisterium für Bildung und Forschung)

b) I-Mon: Intelligent energy monitoring of electrical consumers

Contents

1 INTRODUCTION **1**

1.1 ENERGY DISAGGREGATION IN A NUTSHELL 1

1.2 POTENTIAL APPLICATIONS OF NILM 2

1.3 AIMS AND OBJECTIVES 4

1.4 THESIS CONTRIBUTIONS 5

1.5 THESIS OUTLINES 6

1.6 PUBLICATIONS 7

2 BACKGROUND & RELATED WORK **9**

2.1 NILM TERMINOLOGY 9

2.2 NILM WORKFLOW & LITERATURE REVIEW 11

2.2.1 Data Acquisition *12*

2.2.2 Event Detection *12*

2.2.3 Feature extraction *13*

2.2.4 Classification *14*

2.3 NILM DATASETS 15

2.3.1 HELD1: Home Equipment Laboratory Dataset [14] *17*

3 NILM SIGNAL ANALYSIS WITH TIME-FREQUENCY TOOLS **19**

3.1 INTRODUCTION 19

3.2 BACKGROUND AND PROBLEM STATEMENT 20

3.3 HUANG-HILBERT TRANSFORM 23

3.3.1 Empirical mode decomposition *23*

3.3.1.1 Stopping criteria 25

3.3.1.2 Ensemble Empirical Mode decomposition 26

3.3.2 Hilbert spectrum Analysis (HSA) *28*

3.3.2.1 Marginal Hilbert spectrum 28

3.4 ON THE FEASIBILITY OF HHT FOR BETTER NILM 29

3.4.1 Methodology *29*

3.4.2 Filter-Based EMD inspired algorithm *30*

3.4.2.1 Detailed description of FBE algorithm 31

3.4.2.2 FBE outputs in steady state and Transient state 35

3.4.2.3 FBE-based event detector 37

3.4.2.4 Numerical evaluation of the FBE-based event detector 41

3.4.2.5 FBE for NILM feature extraction 45

3.4.3 NILM Feature extraction with classical HHT *45*

3.4.3.1 Parameter tuning 46

3.4.3.2 Empirical NILM signal segmentation 48

3.4.3.3 numerical simulations 53

3.4.3.4 Exploring HSA outputs and candidate Features 56

3.5 CONCLUSIONS AND CLOSING NOTES 60

4 FEATURE SELECTION FOR NILM SYSTEMS 62

4.1 MATHEMATICAL BACKGROUND AND PROPOSED ALGORITHM 64

4.1.1 Euclidean Distance, Similarity, and Entropy *64*

4.1.2 NEFSA Design Rules *65*

4.1.3 NEFSA principle *66*

4.1.4 Calculating Number of Clusters *71*

4.1.4.1 Using K-means Algorithm 71

4.1.4.2 Using Histogram Peaks Detection (CHIPD) 72

4.2 NEFSA ASSESSMENT AND NUMERICAL SIMULATIONS 74

4.2.1 Experimental setup *74*

4.2.2 NEFSA Efficiency Assessment *77*

4.2.3 Testing NEFSA on real home measurements *81*

4.2.4 Computational cost *82*

4.3 DISCUSSION AND CLOSING NOTES 84

5 SCENARIO-BASED NILM SIGNALS CONSTRUCTION 88

5.1 NILM SIGNAL SYNTHESIS OUT OF SINGLE-DEVICE MEASUREMENTS 90

5.1.1 Intuition and phase preservation *90*

5.1.2 Operational modes *93*

5.1.3 Numerical comparisons *94*

5.1.3.1 Event Detection 96

5.1.3.2 Comparing extracted features 97

5.1.3.3 Comparing classification results 99

5.2 NILM SIGNAL SYNTHESIS OUT OF AGGREGATE MEASUREMENTS 102

5.2.1 Steady-state subtraction *103*

5.2.2 Extraction of Operational Modes *104*

5.2.3 Construction Procedure *105*

5.2.3.1 Modelling of steady-state variations 107

5.2.3.2 Signal-Generation of individual appliance 111

5.2.4 Numerical comparisons *112*

5.2.4.1 Event Detection 113

5.2.4.2 Feature selection 114

5.2.4.3 Classification 116

5.3 POSSIBLE APPLICATIONS OF THE CONSTRUCTION SCHEME 117

5.3.1 providing more Data for Training/Testing NILM algorithms *117*

5.3.2 Tracking classified appliances for abnormal behaviours *119*

5.3.2.1 Tracking principle 120

5.3.2.2 Evaluation 121

5.3.2.3 Use-cases for Tracking scheme 123

5.4 GRAPHICAL TOOL FOR NILM SIGNALS ANALYSIS AND CONSTRUCTION 123

5.4.1 objectives of the tool *124*

6 SUMMARY AND PERSPECTIVES 126

6.1 CHAPTER SUMMARIES 126

6.2 CONCLUSIONS & FUTURE WORK 128

7 BIBLIOGRAPHY 130

8 APPENDICES 142

APPENDIX 1 NILM SIGNAL SEGMENTATION PSEUDO CODE 143

8.1 PSEUDOCODE AND EXPLANATION 144

APPENDIX 2: EXAMPLES OF IF CURVES FOR VARIOUS APPLIANCES 146

APPENDIX 3: SEQUENTIAL FEATURE SELECTION ALGORITHMS AND MORE COMPARISONS 147

APPENDIX 4: A VISUAL USER-GUIDE FOR THE SIGNAL CONSTRUCTION GUI 149

List of Tables

Table 2.1: Public NILM Datasets with V, I measurements 16

Table 2.2: List of all appliances in HELD1 18

Table 3.1: Short comparison among TFA tools 22

Table 3.2: Results of the FBE-based Event detector on BLUED Dataset 44

Table 3.3: Performance comparison with [93] 45

Table 3.4: Classification rates with extracted marginal features only 60

Table 4.1: List of Appliances under Test 74

Table 4.2: Bank of NILM features, First part 75

Table 4.3: Bank of NILM features, second part 76

Table 4.4: The 7 selected features for G1, ranked 81

Table 4.5: Intersections of best subsets with increased overlapping 87

Table 5.1: Public Datasets with measurement types 89

Table 5.2: List of considered Appliances 95

Table 5.3: Comparison of Event detection results, Hart 97

Table 5.4: Comparisons of Event Detection results, FBE 97

Table 5.5: FFNN performance comparison 100

Table 5.6: Side by side comparison of confusion matrices, Original vs constructed signals..... 101

Table 5.7: SVM performance comparison (3 kernels)..... 102

Table 5.8: Hart Event Detector performance comparison 113

Table 5.9: FBE Event Detector performance comparison 114

Table 5.10: Best 20 features per NEFSA algorithm: Constructed vs Original 115

List of Figures

Figure 1.1: Typical residential NILM configuration 1

Figure 1.2: Thesis contributions to NILM tasks in green 6

Figure 2.1: Aggregate power consumption with marked switching events (arrows), [6] 10

Figure 2.2: Hart's "steady states and changing states", [6] 11

Figure 2.3: Classic processing chain of an event-based NILM system 12

Figure 2.4: Block diagram of the measurement system, [14] 18

Figure 3.1: Two steps of HHT 23

Figure 3.2: Flowchart of the sifting process for EMD algorithm 25

Figure 3.3: Methodology scheme for exploring HHT in NILM applications..... 30

Figure 3.4: First stage of FBE algorithm, [92]..... 32

Figure 3.5: FBE algorithm 34

Figure 3.6: Example of FBE application (steady state) 35

Figure 3.7: Example of FBE application (Transient state, first block) 36

Figure 3.8: Example of FBE application (Transient state, second block) 37

Figure 3.9: (Top) a subset of aggregate current NILM signal around an Event in time domain (middle) spectrogram with zoom-in on the event area (bottom) cardinality of modes versus time 39

Figure 3.10: Visualization of some criteria 40

Figure 3.11: Data preparation 42

Figure 3.12: Initial research scheme for HHT feature extraction 46

Figure 3.13: Mode mixing example (highlighted), Device: Fan, 2 Events 47

Figure 3.14: Example event with left and total windows in power and current signals . 49

Figure 3.15: Checking event type based on comparing left and total windows 50

Figure 3.16: Transient state limits along with the event index (GT) 51

Figure 3.17: 100 Events of Hairdryer together 52

Figure 3.18: 100 Event of a Hairdryer, transient ends with proposed method 53

Figure 3.19: More mode mixing due to higher Nstd value 54

Figure 3.20: Effect of tuning MaxItr on 50 Hz modes 54

Figure 3.21: Examples of final IMFs after fixing Parameters 55

Figure 3.22: Inspecting IF curves for 2 devices at 4 random events each 57

Figure 3.23: IF envelopes for 6 appliances 58

Figure 3.24: marginal spectrum for 4 Events, Fan 58

Figure 3.25: comparison of marginal spectra of 4 devices 59

Figure 4.1: Example of performance plateau after using 10% of the total feature set.... 63

Figure 4.2: Flowchart of NEFSA Algorithm 67

Figure 4.3: Visualising the defined distances in 2D feature space 68

Figure 4.4: Similarity vs distance Type 1 69

Figure 4.5: Similarity vs distance Type 2 70

Figure 4.6: Calculating the number of clusters using k-means algorithm 72

Figure 4.7: (a) Histogram of feature values, (b) smoothed envelope 73

Figure 4.8: Classification Rate as a function of subset size with all algorithms..... 77

Figure 4.9: Performance of all feature selectors against randomly drawn subsets..... 78

Figure 4.10: Several comparisons of NEFSA performance with various classifiers..... 79

Figure 4.11: Evaluation of NEFSA ranking on BLUED Dataset 82

Figure 4.12: time-efficiency of NEFSA vs. SFFS and SFS..... 83

Figure 4.13: Entropy evaluation of all features on G1..... 84

Figure 4.14: NEFSA Feature ordering with 3 overlapping degrees 85

Figure 5.1: NILM signals construction out of single-device measurements 91

Figure 5.2: Aggregation algorithm 92

Figure 5.3: Operational modes of a fridge 93

Figure 5.4: Typical sequence of processes in a NILM system 94

Figure 5.5: Constructed current signal vs Original one, magnifying 2 regions..... 96

Figure 5.6: comparison of steady state features for a Kettle (top), and a Radio (bottom) 98

Figure 5.7: NILM signals construction out of aggregate measurements 103

Figure 5.8: Steady-State subtraction (left) Flow-chart (right) example for a Dremel event 104

Figure 5.9: comparisons of operational modes (out of individual measurement vs aggregate measurement) for two appliances 105

Figure 5.10: Construction algorithm with noise modelling 106

Figure 5.11: Steady-state envelope, Kettle 107

Figure 5.12: steady-state modelling, first fitting process 108

Figure 5.13: steady-state modelling, second fitting process 109

Figure 5.14: (Left) Histogram of Nsig values with fitted gaussian curve (Right) Autocorrelation of Nsig 109

Figure 5.15: Modelling steady-state variations of a Lamp 110

Figure 5.16: Modelling steady-state variations of a Dremel 111

Figure 5.17: Signal-Generation of each individual appliance 112

Figure 5.18: Entropy comparison: constructed vs original measurements 115

Figure 5.19: Testing classification performance on constructed and original measurements 116

Figure 5.20: Combining different public datasets with NILM construction procedure 118

Figure 5.21: Random generation of NILM events-scenarios..... 118

Figure 5.22: simple tracking scheme 119

Figure 5.23: Testing the resulted indivdual signals 122

Figure 5.24: Comparing current signals: Tracker output vs Original individual measurements..... 122

Figure 5.25: Sample interfaces from the graphical tool..... 125

Figure 8.1: inputs and outputs of the Signal Segmentation procedure 143

Figure 8.2: IF curves for 6 devices, 20 events for each one; taking first period of Transient state 146

List of Appendices

Appendix 1 NILM signal segmentation Pseudo code 143

Appendix 2: Examples of IF curves for various appliances 146

Appendix 3: Sequential Feature selection algorithms and more comparisons 147

Appendix 4: A visual User-Guide for the signal construction GUI 149

1 Introduction

1.1 Energy Disaggregation In a Nutshell

Energy disaggregation is a set of computational tools for estimating the individual power consumption of each appliance from aggregate current and voltage measurements of a single meter. Figure 1.1 illustrates a typical residential configuration with a NILM smart meter installed at one point (ideally the main).

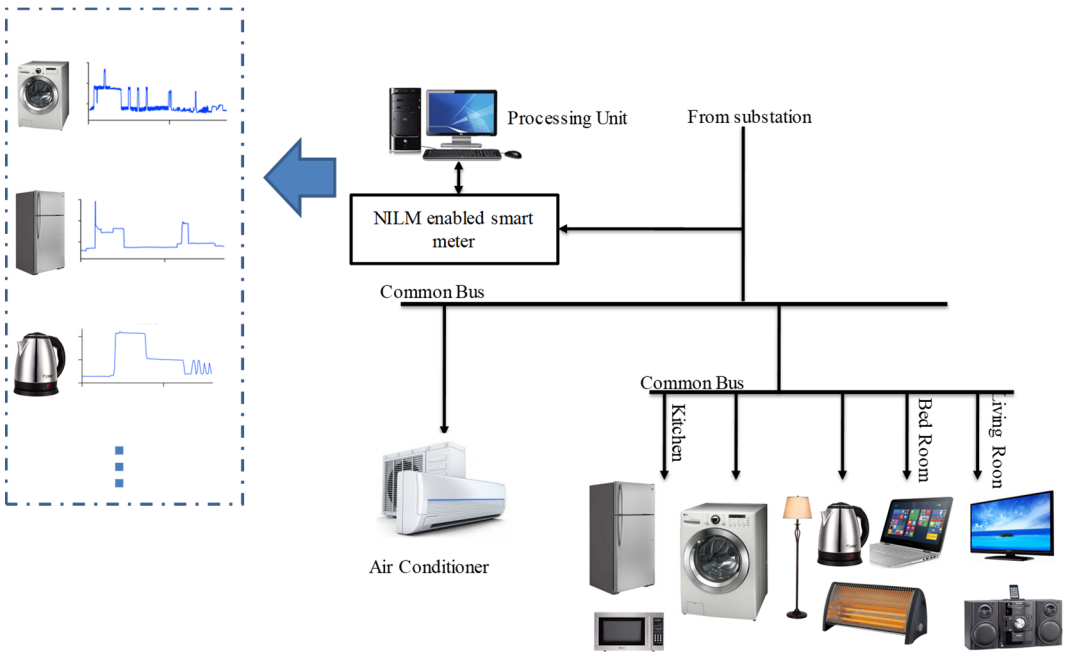


Figure 1.1: Typical residential NILM configuration

George Hart started researching this topic in the 1980s [1], [2], when he invented the term “NILM”, which stands for: Non-Intrusive Load Monitoring. Nowadays, NILM is the most used term in the scientific research community of energy disaggregation field. An exemplary use-case for NILM is providing the layman with a detailed electricity bill, which can assist him in taking informed decisions about his power consumption. High hopes [3]–[5] pinned on such feedback mechanism for reducing energy consumption, more NILM use-cases in section 1.2.

NILM has drawn a lot of attention in the last decade; the most cited NILM article [6] by Hart himself has 2035 citations in total, 1920 of which are since 2010 (according to google scholar engine, accessed 5 November 2019). This attention was accompanied with the new regulations that impose installing smart meters in many countries. More than 20 companies in Europe alone [7] are already offering NILM products and services. Several NILM workshops were held recently (6 in Europe, 4 in north America), and many conferences haave a dedicated track for NILM or smart energy in general.

1.2 Potential applications of NILM

NILM has a broad range of use-cases, this section mentions some of them with a brief context:

- Generating an itemized electricity bill with individual energy-use of each electrical device can lead to saving energy/money

Context: A handful of studies (e.g. [8], [9]) showed that providing people with detailed appliance-by-appliance list of consumed power/cost can give them incentive to reduce their power consumption, possibly by changing their

behavioural patterns or by replacing inefficient appliances with better energy efficient ones (e.g. newer fridges or LED lights consume/cost considerably less power/money). Furthermore, the effects of those replacements can be tracked with NILM logs.

- ❖ Quantifying power consumption is beneficial for Grid operators

Context: Grid operators and electrical-power providers need end-user data for different goals; with NILM meters, a wealth of data is available and statistical analysis can be automatically generated with high sample size, such data can also enhance demand response greatly.

- ❖ Enabling new services for smart buildings

Context: IoT sensors and NILM meters can be well integrated to advance smart building/home technologies and create new possibilities, to name a few examples: sending alarm notification for “on” devices in case no body was home, or for possible water leakage (as the water sensor has no activity while the washing machine is active all the time), another example would be ocupancy detection in active assisted living environments.

- ❖ Diagnostics and preventive maintenance of electrical appliances

Context: One of the ambitions for NILM is to identify problems and malfunctions in electrical appliances before they totally break, or detect a mistake in their settings. For example a stand-by appliance consuming a high portion of power in the night can be reported; a dish-washer with an unusual power profile should prompt the NILM meter to generate a message to the user. more thoughts on this application in chapter 5.

- Raising energy-consumption awareness among the general public can lead to better climate policies

Context: The adverse effects of human activities on climate changes are still debated despite the general agreement of scientific society on the urgent need for preventive actions on a global scale. While some researchers are skeptical about the NILM usefulness for convincing people to save power [10], [11] and hence contribute to mitigating the climate change; still NILM can enhance the perception of energy consumption and increase energy literacy[12] among people, which can bring them closer to understanding the energy-climate dilemma and eventually mobilize them towards better political choices.

1.3 Aims and Objectives

The growth of renewable energy generation can be the solution to achieve the desired reductions in CO_2 emissions by 2050 [13], one of the critical factors for this transition to clean energy is the flexibility of the power grid as the sources of renewable energy cannot be controlled due to their weather dependency. A flexible grid requires a constant flow of data about the network and its demand, on the other hand, clients who produce electrical power can be an active part of the demand response if they are informed about the power needs of their appliances.

My aim throughout this work is to contribute to the efforts of developing NILM technologies so that it can be an efficient tool for a flexible grid, leading to a cleaner environment for our children.

The concrete objectives of this work are:

1- To explore the advantages and possibilities of time-frequency representation of NILM signals for increasing the accuracy of "On/Off" events detection and extracting effective features for better disaggregation results.

2- To develop a novel algorithm for features selection suitable for low-cost NILM meters, enabling them to adapt automatically for defined changes in a certain building/facility.

3- To design a data aggregation scheme as a response to the problem of incomplete publicly available datasets.

4- To propose a method for tracking classified devices and detect unusual consumptions.

1.4 Thesis Contributions

The main contributions of this thesis to the field of NILM systems are the following (Figure 1.2 illustrates the chain of NILM tasks, I contributed to the green ones):

- A detailed investigation of Huang-Hilbert Transform capabilities for NILM signals, resulted in a novel event detection approach for low-cost non-intrusive load monitoring system, motivated by Empirical Mode Decomposition and its modes extraction procedure; and a promising set of features for future "NILM on the cloud" solutions.
- A new Entropy based feature selection algorithm for NILM features, designed to give NILM meters the ability of adjusting load signatures on the fly.

- A scenario based NILM signals construction, which extend any set of limited NILM measurements for training and testing NILM algorithms. The concept has other beneficial applications like tracking classified devices for unusual activity detection or supporting the design process of energy-efficient buildings.

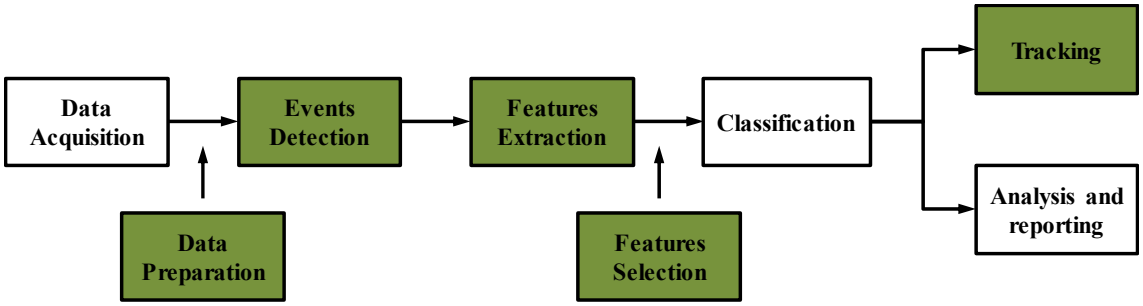


Figure 1.2: Thesis contributions to NILM tasks in green

1.5 Thesis Outlines

The remaining chapters of this thesis are organized as follows:

In Chapter 2, we define the basic terms and concepts of NILM systems, then provide a block-oriented literature review and an overview of the publicly available datasets for NILM research, with detailed description of HELD1[14] and BLUED [15] datasets.

Following that, the main body of this thesis will divided in three chapters:

In chapter 3 we explore empirical mode decomposition (EMD) and Huang-Hilbert Transform (HHT) for NILM signals and propose an event detector and a group of features. In Chapter 4 we propose a novel features-selection algorithm for NILM systems with numerical evaluations, while in chapter 5, we present a new scheme of

NILM signal construction for generating any desired NILM scenario out of limited number of real measurements.

Lastly, in chapter 6 we summarize the presented work in this thesis and discuss the possible directions for future work.

1.6 Publications

The following list of publications have emerged from the work in this thesis in collaboration with my colleagues in UHA and HFU universities, most recent ones are mentioned first:

1) Alaa Saleh, Pirmin Held, Djaffar Ould Abdeslam, and Dirk Benyoucef "On the Feasibility of Scenario-Based NILM Signals Construction". Submitted to *IEEE Transactions on Instrumentation and Measurement*.

2) Alaa Saleh, Djaffar Ould Abdeslam, and Dirk Benyoucef "An Entropy Based Feature Selection Algorithm for Smart Meter Data Disaggregation". Submitted to *IEEE Transactions on Smart Grid*.

3) Pirmin Held, Steffen Mauch, Alaa Saleh, Djaffar Ould Abdeslam, Dirk Benyoucef "Frequency Invariant Transformation of Periodic Signals (FIT-PS) for Classification in NILM. *IEEE Transactions on Smart Grid*.

4) Alaa Saleh, Pirmin Held, Dirk Benyoucef, Djaffar Ould Abdeslam. "A Novel Procedure for Virtual Measurements Generation suitable for Training and Testing in the context of Non-Intrusive Load Monitoring". *SIGNAL 2018, The Third International Conference on Advances in Signal, Image and Video Processing, Nice, France, May 2018.*

5) Pirmin Held, Steffen Mauch, Alaa Saleh, Djaffar Ould Abdeslam, Dirk Benyoucef. HELD1: Home Equipment Laboratory Dataset for Non-Intrusive Load Monitoring. *SIGNAL 2018, The Third International Conference on Advances in Signal, Image and Video Processing, Nice, France, May 2018.*

6) Alaa Saleh, Pirmin Held, Dirk Benyoucef, and Djaffar Ould Abdeslam. "EMD inspired Filtering algorithm for signal analysis in the context of Non-Intrusive Load Monitoring". *IECON 2017 43rd Annual Conference of the IEEE Industrial Electronics Society. Beijing, China, November 2017.*

7) Pirmin Held, Alaa Saleh, Djaffar Abdeslam Ould, and Dirk Benyoucef. "Frequency Invariant Transformation of Periodic Signals (FIT-PS) for high frequency Signal Representation in NILM". *In 3rd Baden-Württemberg Center of Applied Research Symposium on Information and Communication Systems, pages 1-6, Karlsruhe, December 2016.*

2 Background & Related Work

In this chapter, we present an overview of the basic terms and subtasks of a NILM system through a historical example from Hart's pioneer work, then a review of the NILM state-of-the-art is provided, where we explore the on-going research efforts in the field and how they are related to the proposed developments in this Thesis.

2.1 NILM Terminology

The earliest NILM prototype was published by Hart in 1985 [2] then improved in a newer article in 1992 [6], where he defined some of the basic terms of the field. In Figure 2.1 he showed the aggreate power signal of a household, where each power change is linked to a switching event of an appliance (arrows mark the detected events in the figure). An **Event** is defined as a power change that exceeds a predefined threshold and lasts for a predefined period of time. **Event detection** is the first task in a typical event-based NILM system: In Figure 2.1 examples of

detected ON events are marked with green ovals, detected OFF events are marked with red ovals, missed (undetected) events are marked with blue ovals.

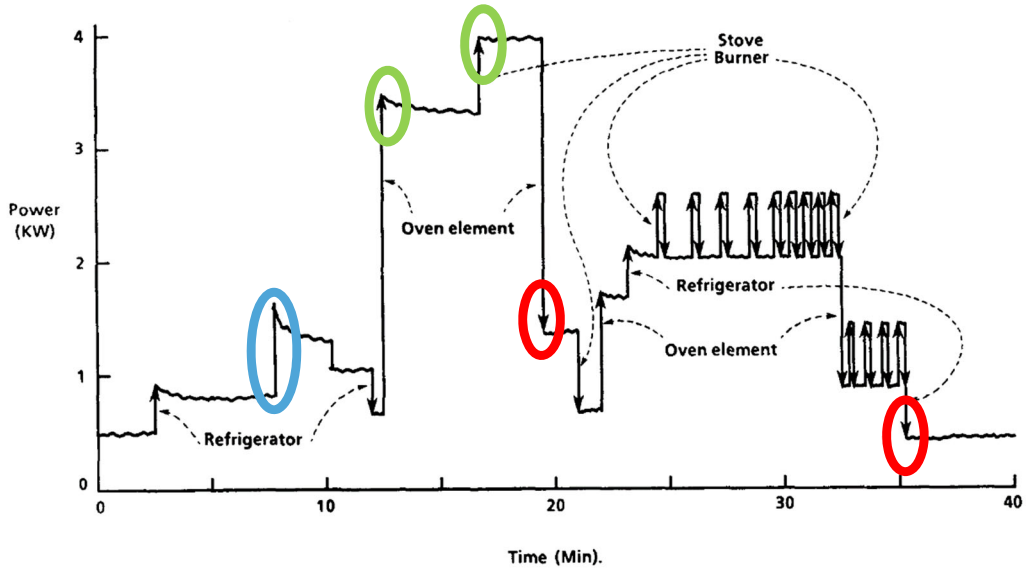


Figure 2.1: Aggregate power consumption with marked switching events (arrows), [6]

In the given example of Hart, he ignored the fast transients in the power signal by segmenting the power signal to **steady states** and "changing" states, as illustrated in Figure 2.2. Changing states are referred to as **transient states** in the recent NILM literature. Steady state was characterized by a minimum of 3 seconds (3 samples as Hart sampled with 1 Hz) and restricting the power change to 15 W (VARs in reactive power). **Features** (active and reactive powers only) were calculated for steady states in this case, this feature calculation step is the second NILM subtask and is known as "**Feature extraction**". The next step is to cluster those events according to the values of their features, then match the clusters that have similar values but opposite signs (ON/OFF).

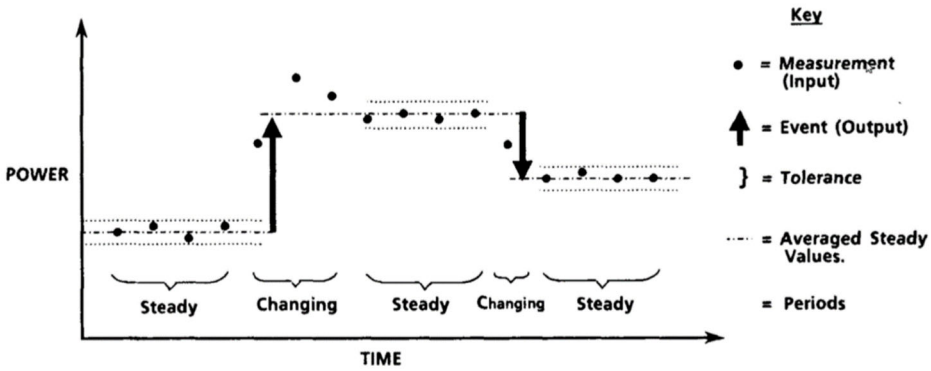


Figure 2.2: Hart's "steady states and changing states", [6]

In the final **Load identification** step, each appliance is assigned to one of those pairs of clusters according to its **Load signature**, which is defined as a set of features associated with this load (in this case active and reactive powers are the only considered features). The signatures of the loads are known beforehand or gathered during a **training phase**. If each load were to be uniquely identified by its signature, i.e. all clusters are well separated in the **signature space**, then the considered features would comprise an **optimal set of features**; otherwise a **features-selection** operation can be added to enhance the overall disaggregation result.

2.2 NILM Workflow & Literature Review

Since we have already introduced the basic sub-processes of NILM systems in the last section, let us explore the classic processing chain of event-based NILM systems in Figure 2.3. We will discuss those tasks one by one with an overview of the related research work in the following.

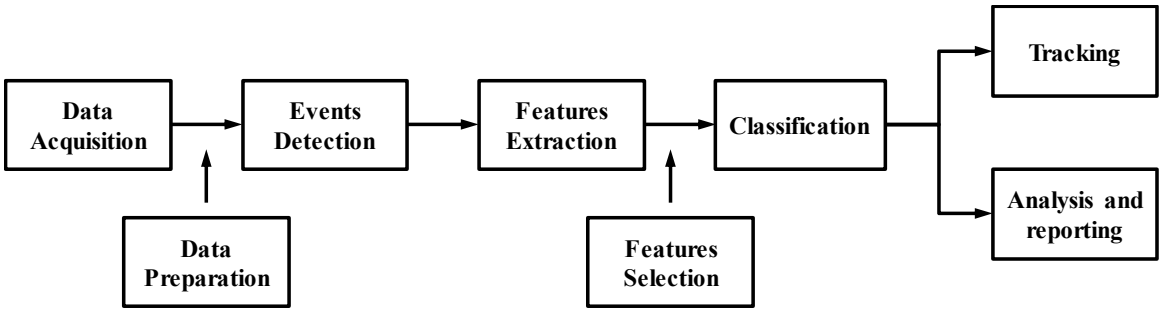


Figure 2.3: Classic processing chain of an event-based NILM system

2.2.1 Data Acquisition

The first task of any NILM smart meter is acquiring and sampling the current and voltage signals. In [16], the authors differentiated between low-frequency solutions and high-frequency-solutions, listing the additional possible features if the sampling rate is increased till 1 KHz or more. Higher sampling rates reduce the number of simultaneous events[17]. The authors of [5] argued that distinguishing 20-40 appliances require a sampling frequency of 10 – 40 KHz, while going beyond 1 MHz can raise the uniquely identified appliances to 100, yet the memory needs for such configuration should be addressed with care, Ul Haq proposed using audio compression algorithms to mitigate this problem [18]. The ADC (Analog to Digital Converter) should be also selected carefully, i.e. a 10 bits ADC can make the smart meter blind for all small consumers under 11.23 W [19].

2.2.2 Event Detection

An accurate event detector is critical for the success of the whole NILM task. This fact motivated many researchers to develop their own detectors: heuristic methods were used in [20], [21] to scan the current signal for noticeable changes (above predefined threshold). Other methods based on probabilistic models were reported

in [22]–[24], where statistical tests are applied on each sample of the power signal to calculate the chance of an event, the resulting probabilities will be compared to a predefined threshold for the final decision. Other approaches used time-frequency analysis tools like STFT and wavelet transform [25], with better results for wavelet transform in detecting transient states (only 3 appliances were tested).

An event-detector based on a novel representation of NILM signals is presented in [26]. Chapter 3 will present my proposed algorithm for event detection in low-cost NILM systems with evaluation tests.

Lastly, we point out to the existence of event-less NILM approaches, in which each sample is processed and compared to the consumption of various combinations of aggregate loads. Such approaches rely on probabilistic method like Hidden Markov models (see [27] and references within) or they formulate NILM as a blind source separation problem [28], [29], These methods are generally computationally-demanding compared to the event-based ones and less accurate as well.

2.2.3 Feature extraction

Many efforts were made towards extracting effective features for improving the classification accuracy of loads, most of them relied on previous experiences in electrical engineering [30]–[32], or were transferred from other domains like audio signal processing [33]. Various ways to categorize NILM features were proposed in the literature: high frequency vs low frequency features [16], manually engineered vs data driven [34], steady-state vs transient state features [35].

Data-driven features are getting more attention as deep learning algorithms are getting more effective in all fields; yet still we believe that manual extraction can be

very useful for specific cases, an interesting example can be seen in [36] (page 63), where EMI (electromagnetic interference) signals were used to discriminate 4 identical PC power-supplies.

To the best of our knowledge, there is still no "one size fits all" in terms of an optimal set of features for all appliances. The best attainable solution is to exploit the available features and have a fast features-selection algorithm for different scenarios. In this thesis a bank of 84 NILM features were gathered in Table 4.2 and Table 4.3 for the sake of developing entropy-based feature selection algorithm.

2.2.4 Classification

The features from the last stage are to be used for identifying the working appliances, researchers formulated this in either of two ways: a classification problem, or an optimization problem.

Starting with classification methods: nearly all classical supervised classifiers were applied, like Support Vector Machines (SVM) in [37]–[39] , decision trees [40], [41], K-Nearest Neighbor (K-NN) [37], [40], [42], Linear Discriminant Analysis (LDA) [40], [43]. Deep Neural Networks were also employed [44]–[47], and Convolutional Neural Networks (CNN)[48], [49]. Unsupervised methods were also applied, [23], [28], [36], [50], with the added advantage of not requiring a training phase. An interesting approach of Semi-supervised learning was followed in [51], where the authors make use of the often incomplete labelled data points to infer the label of the other unknown (clustered) data points.

Optimization methods generally employ combinatorial search (matching aggregate consumption by finding the set of appliances that best approach the measured

values) with different tools: genetic algorithms [52], Integer programming [53], [54]. in addition to the high complexity issue (especially with higher number of appliances), it is hard to discriminate between similar signatures using such methods[35].

2.3 NILM Datasets

A NILM Dataset is defined as a collection of electrical energy measurements that may include current and voltage measurements and/or power measurements; those measurement can be aggregate measurements (overlapping group of appliances measured at the main level) and/or individual measurements (stand-alone measurements measured at the plug level).

One of the main reasons for growing research efforts in the NILM field since 2010 is the availability of public datasets. In Table 2.1, we listed all public datasets that include current and voltage measurements from the most recent to the oldest. EB stands for event-based (i.e. the dataset has records for all or most of the switching events). For more information on other datasets with power measurements alone we refer to [55] (and references within). Event-less datasets are more convenient to collect, which may explain why they were dominant till 2018 . The only dataset prior to that with whole-house measurements fully labeled (identified power changes were listed with the corresponding appliaces) is BLUED dataset [15].

While PLAID, COOLL and WHITED [56]–[58] are event-based datasets, yet they are limited to short transients (2 - 6 seconds) from various individual measurements, they can be used for evaluating feature extraction and classification algorithms using cross-validation [59], [55].

Table 2.1: Public NILM Datasets with V, I measurements

Dataset	Resolution	EB	Year	country	Duration
HELD1 [14]	4 KHz	Yes	2018	Germany	Scenarios
BLOND [60]	50/250 KHz	Yes	2018	Germany	213/50 days
RAE [61]	1 Hz	NO	2018	Canada	10 weeks
WHITED [57]	44.1 KHz	Yes	2016	Germany, Austria, Indonesia	Transients
SustDataED [62]	12.8 KHz	Yes	2016	Portugal	10 days
COOLL [63]	100 KHz	Yes	2016	France	Transients
ECO [64]	1 Hz	NO	2014	Switzerland	8 months
UK Dale [65]	16 KHz	NO	2014	UK	655 days
PLAID I&II [56]	30 KHz	Yes	2014	US	Transients
COMBED [66]	1/30 Hz	NO	2014	India	1 month
AMPds [67]	1 Hz	NO	2013	Canada	2 years
IHEPCDS [68]	1/60 Hz	NO	2013	France	4 years
iAWE [69]	1 Hz	NO	2013	India	74 days
ACS-F2 [70]	1/10 Hz	NO	2013	Switzerland	2 hours
BLUED [15]	12 KHz	Yes	2012	US	1 week
REDD [71]	15 KHz	NO	2011	US	2 - 4 weeks

To solve the problem of scarce labelled data in the NILM field, we proposed two approaches in our lab: the first one is a data generation scheme for constructing "virtual" aggregate measurements from real individual measurements, which is the subject of the fifth chapter. The other approach is to collect a new dataset in our lab, namely HELD1 dataset [14]. In the following sections we describe HELD1 in detail as it was used in different parts of this thesis along with BLUED dataset [15].

2.3.1 HELD1: Home Equipment Laboratory Dataset [14]

The HELD1 dataset was collected in a controlled environment (lab) with the ability to determine the exact scenario of measurements. The measurement system (designed by Thomas Bier [72], extended to 4 kHz by Pirmin Held [14]) is equipped with a switching box that generates switching events for the connected appliances, thus selecting a predefined sequence of events is possible and the reference data can be recorded with high precision. A block diagram of the measurement system is depicted in Figure 2.4.

As listed in, 18 appliances were considered in the measurement campaign (23 internally, unreliable measurements were eventually discarded). All of them were used to collect 4 groups of measurements according to the following scenarios:

G0: individual measurements for the same appliance ,100 pairs of ON/OFF events.

G1: up to one active appliance at a time, randomly picked.

G2: up to 4 appliances can be active at the same time (random switching).

G3: up to 6 appliances can be active at the same time (random switching).

HELD1 provides fully labelled aggregate and individual measurements at relatively high sampling rate (4 kHz). which makes it the first suitable dataset for evaluating event-based and event-less approaches (BLOND shares the same advantage).

Table 2.2: List of all appliances in HELD1

ID	Name	P(W)
1	Toaster	998
2	Black hairdryer (setting 2)	1155
3	Radio	6.2
5	Vacuum cleaner	424
7	Hairstraightener	56
9	Heat gun (setting 1)	820
10	Router	9.2
11	Black desk lamp	20
13	Refrigerator (white)	170
14	Refrigerator (blue)	190
15	Fluorescent Lamp	40
16	Light bulb box	20
17	Kettle	2100
19	Black hairdryer (setting 1)	500
20	Heat gun (setting 2)	1603
21	Fan	22
22	Dremel (Multifunction tool)	30
23	LED lamp	1

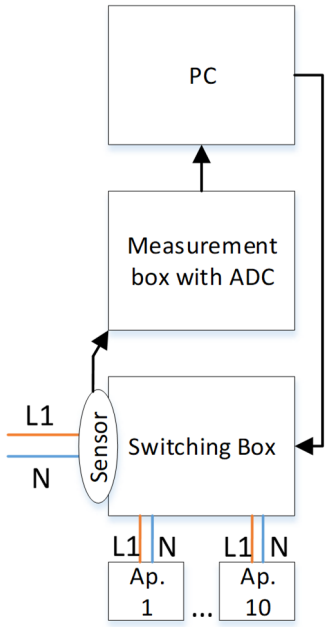


Figure 2.4: Block diagram of the measurement system, [14]

3 NILM Signal Analysis With Time-Frequency Tools

3.1 Introduction

Analysing discrete and digital sampled signals grew rapidly from a mere application of mathematical "numerical" techniques to be a huge independent field of engineering science in the last 7 decades, simply known now as signal processing. Digital signal processing (DSP) algorithms "like filtering, estimation, detection, recognition, transmission, analysis, synthesis ...etc" are cornerstones in almost all scientific fields: communications, control, economics, biology, computer science...etc.

In this chapter, we will discuss the application of a special branch of digital signal processing: utilizing signal analysis techniques to gain deeper insights into the underlying processes of NILM signals. Traditionally, time domain analysis and frequency domain analysis were used, we will take a glance upon those methods, briefly discuss the need of better tools and introduce the concept of time-frequency distributions, in particular, Huang Hilbert Transform "HHT"; then we proceed to

NILM feature extraction with EEMD and HHT, after that we present our approach "Filter based EMD" with its application in event detection and feature extraction.

3.2 Background and problem Statement

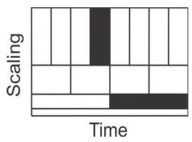
Early approaches of NILM signal analysis [6], [73], [74] were in time domain, where different tools were used to harness the information content of power signals (real, active, reactive). Those methods are simple and suitable for low sampling rates as well [75]. However, they are sensitive to noise and power disturbances, which can yield substantially sub-optimal forecasts (i.e. high misclassification rate)[76]. Additionally, the low sampling rate (usually paired with such methods) decreases the temporal accuracy of load-detection. On the other hand, frequency domain signal analysis approaches looked for measurable changes in the spectral domain (harmonic components in particular) [77], which can be very useful for loads that draw the same active power and have unique harmonic signature [78]; but they are not sufficient for loads with poor harmonic profile (resistive loads). Applications of standard spectral analysis (e.g. Fourier decomposition) provide us with spectral signal components (frequencies) along with their relative intensities, yet it does not tell us anything about the specific time where those frequencies were dominant [79].

Time-frequency analysis (TFA) can be a powerful solution as it will give us a representation of energy (or power) of a signal in time and frequency domains simultaneously. In most cases a short-time Fourier transform (STFT) is used, which is equivalent to obtaining the spectrum in time intervals (windows $\boldsymbol{w}$) with T duration: $X(T,w) = \int x(t)\boldsymbol{w}(t-T)e^{-jwt}dt$ (3.1)

but this assumption of piecewise stationarity is not always valid (e.g. change rate in a given signal is very high, i.e. the time resolution T is bounded). Furthermore, frequency resolution is inversely proportional to the duration T of the time window. Those drawbacks led to a trade-off consideration between temporal and spectral resolutions for any signal representation (a form of Heisenberg-principle[79]). Several TFDs were developed over the last century; the most recent one is Huang-Hilbert transform (HHT) is an adaptive data-driven approach which does not suffer the shortcomings (i.e. limits) caused by STFT window functions, the various kernels of GTFD (generalized TFDs in the Cohen class) or the choice of mother functions of wavelet transform [80]. In our work we focused on studying the feasibility of such tool for better NILM solutions. A comparison is shown in Table 3.1

For NILM signal analysis, a desired TFD should be able to extract the raw information in current and voltage waveforms, including the non-stationary (and nonlinear) transient part of the current signal. Features from this part can have a considerable positive effect on the load-discrimination rate (classification rate) [81]. Since the computational cost of this TFA in general (and for HHT in particular) is high; we developed a “customized” signal analysis method for NILM signals suitable for low cost NILM systems, in parallel we explored the capabilities of HHT as a tool with NILM signals if cloud-based NILM solutions is considered in the near future.

Table 3.1: Short comparison among TFA tools

	Fourier Transform	**Wavelet Transform**	**S Transform**	**Huang-Hilbrt Transfrom**
building blocks	frequencies	the wavelets	win func	IMF modes
superposition	Linear	Linear	Linear	Linear
Completeness	For stationary signals	+	+	+
Orthogonality	+	+	+	- (almost)
TF space discretization	quasi-Complete for stationary signals only		complete	uniform and complete
fast (RT) application	+	+	+	-
Inverse Transform	+	+	+	-
flexibility	adaptive	A priori given	adaptive	adaptive, Data driven

3.3 Huang-Hilbert Transform

While most natural processes are nonlinear and nonstationary, yet the available methods to handle such data are quite limited and either valid for linear but nonstationary signals (as Wavelet transform), or vice versa. This fact motivated the development of HHT, as introduced recently by Huang [82], as a new tool suitable for filling this gap. HHT can inform us also about the inner dynamics in the processes (Huang presented the concept of intra-wave frequency modulation as the change of the instantaneous frequency within one oscillation cycle).

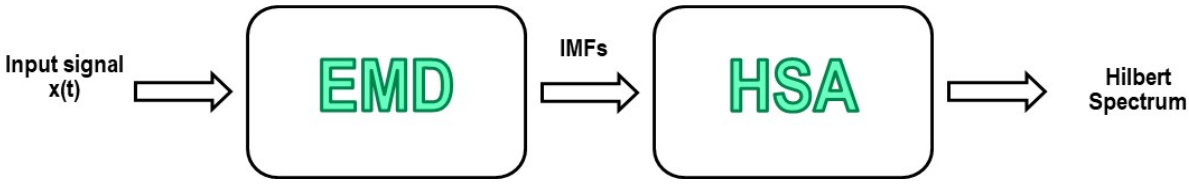


Figure 3.1: Two steps of HHT

Huang-Hilbert Transform is a two-steps procedure. The first step is an empirical mode decomposition (EMD) of the given input signal, the second step is Hilbert spectral analysis (HSA), as in Figure 3.1.

3.3.1 Empirical mode decomposition

Direct application of Hilbert transform to the input signal may yield negative (non-physical) frequencies [83], to avoid such outcome we apply EMD at first, which will decompose the data (i.e. signal) into **I**ntrinsic **M**ode **F**unctions (IMFs), from which the Hilbert transform will give us positive instantaneous frequencies (that hopefully correspond to physically meaningful oscillatory behaviour).

The necessary conditions to define an IMF is to have[82]:

i. Symmetry with respect to the local zero mean;
ii. The number of local extrema and zero crossings differs at most by one.

Putting that in mind now we can describe the "sifting" procedure for extracting IMFs[82], [84]:

1. Identify all local extrema (maxima and minima) in the given signal.
2. Interpolate all the local maxima with a cubic spline line as the upper envelope.
3. Interpolate all the local minima with a cubic spline line as the lower envelope, the upper and lower envelopes should cover the entire signal between them.
4. Compute the mean of both envelopes as $m_1(t)$, calculate the difference between the signal $x(t)$ and $m_1(t)$ as the first component $h_1 = x(t) - m_1(t)$.

This process can face pre-existing overshoots and undershoots in the original signal, which could be classified as new extrema [85], Therefore we repeat the sifting process as many times as required to eliminate all possible riding waves and make the wave profiles more symmetric.

5. Check the resulted $h_1(t)$,
    a. if the requirements of an IMF (stopping criteria) are met, then it is denoted as the first IMF, and $x(t)$ is replaced with $x(t) - h_1(t)$
    b. if $h_1(t)$ does not comprise an IMF, then assign $h_1(t)$ to $x(t)$
6. repeat steps 1-5 till the residual signal becomes a monotonic function or the number of extrema does not exceed one, from which no further IMF can be extracted.

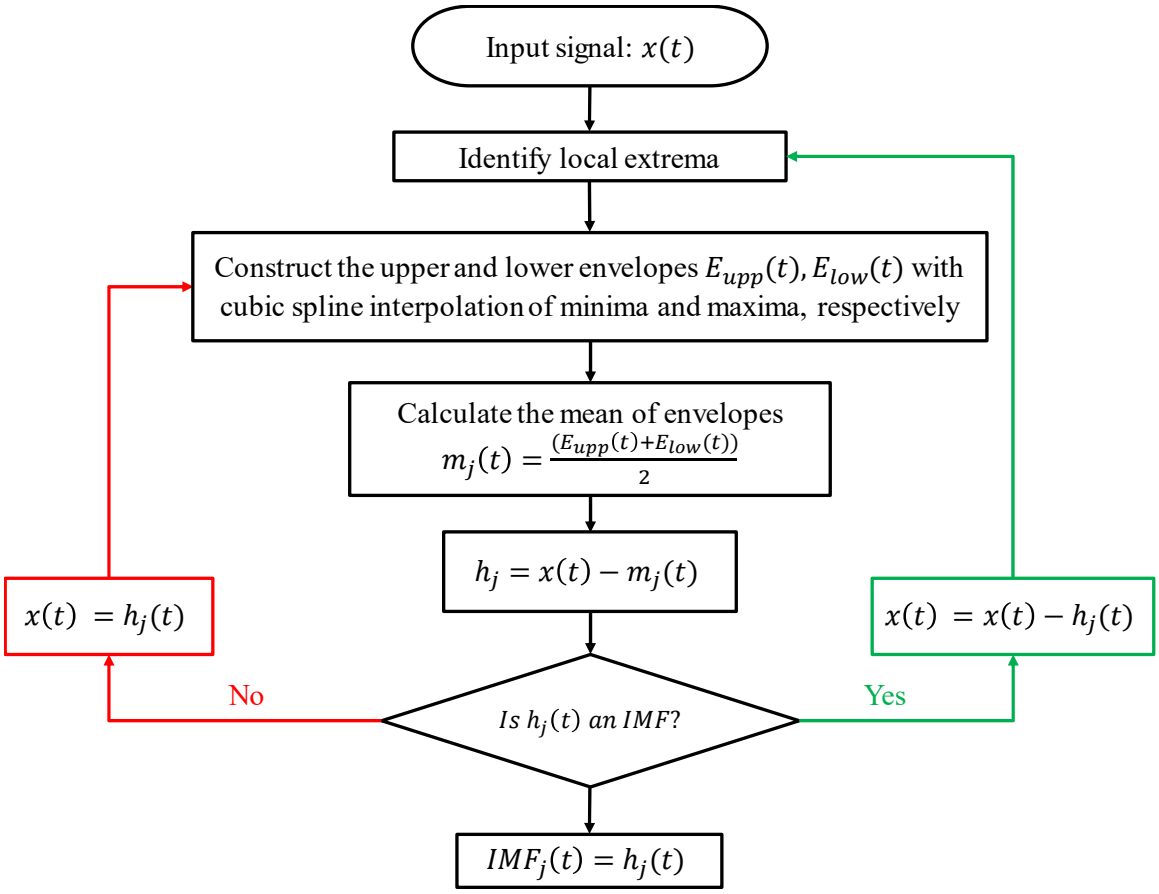


Figure 3.2: Flowchart of the sifting process for EMD algorithm

The sifting process is an iterative procedure that extracts the oscillations in a given signal, starting with highest frequency. Combining IMFs and the remaining residue should give us the inspected signal back (comprising an orthogonal representation):

$$x(t) = \sum_{j=1}^{n} IMF_j + r_n \tag{3.2}$$

3.3.1.1 Stopping criteria

The most critical aspect of EMD is how to choose practical criterion for stopping the sifting process. In the best-case scenario, the process stops when the resulting signal satisfies the IMF conditions; However, reaching zero mean value for the upper and lower envelopes is hardly fulfilled in practice, which leads to a high

number of sifting iterations that in turn wear out the informative (meaningful) amplitude fluctuations[82]. Since the sifting process is an converging process (changes made on the signal should decrease in total in the next new iteration), Huang [83] proposed to limit the size of the standard deviation, denoted as SD, between the results of two consecutive sifting steps:

$$SD = \sum_{t=0}^{N} \left[\frac{\left| h_{j(k-1)}(t) - h_{jk}(t) \right|^2}{h_{j(k-1)}^2(t)} \right] \tag{3.3}$$

The choice of the limit (threshold) can be defined according to application and length of the input data *N*, a value of 0.2–0.3 is a common starting point in most applications and it also was a reasonable initial value in our simulations, yet it should be tuned afterwards according to the application.

The sifting process can also be stopped by any of the following predetermined criteria (among others): absolute amplitude of the remaining signal, mean value of the envelope, cross correlation coefficients between the remaining signal and the original signal.

3.3.1.2 Ensemble Empirical Mode decomposition

A frequently encountered problem with the resulting intrinsic modes is the so-called phenomenon “mode mixing”; where a signal of a similar oscillation (time scale) appears in two or more IMFs, or getting a mode with widely separated scales. Mode mixing has been a major challenge and was first diagnosed by Rilling and Flandrin in [86], Geldhill [87] and Flandrin [88] experimented with adding noise to the original signal, then Wu and Huang [89] proposed an improved version of EMD that depends on an “Ensemble” strategy, in which several realizations of white noise are

introduced to the signal before applying EMD, then averaging over all trials will ensure that the uncorrelated whit noise will cancel out. This method showed great potential in reducing mode mixing (in some cases to 0). To get a better sense of EEMD let us take a glance on its steps:

1) build a group of signals $x_i(t)$, where each one is composed of the original signal $x(t)$ and added white noise (N independent realizations)

$$x_i(t) = x(t) + \varepsilon_i(t), \quad where \ 1 \leq i \leq N \tag{3.4}$$

2) apply the EMD sifting process on each $x_i(t)$ to get its IMFs: $\{m_i^k\}_{1 \leq k \leq K}$ and residue r_i:

$$x_i(t) = \sum_{k=1}^{K} m_i^k(t) + r_i(t) \tag{3.5}$$

3) get mean IMFs and mean residue by averaging over the ensemble of extracted modes:

$$\begin{cases} m^k(t) = \frac{1}{N}\sum_{i=1}^{N} m_i^k \\ r(t) = \frac{1}{N}\sum_{i=1}^{N} r_i(t) \end{cases} \tag{3.6}$$

to reconstruct the original signal :

$$x(t) = \sum_{k=1}^{K} m^k(t) + r(t) \tag{3.7}$$

The standard deviation of the aggregate introduced error by EEMD (added noise) was shown [89] to be $\alpha/\sqrt{N}$, where α is the amplitude of the original added white noise (higher numbers of ensembles reduce this error substantially).

3.3.2 Hilbert spectrum Analysis (HSA)

The second step in HHT is to apply Hilbert transform on the desired mode (or modes depending on the application) and get the instantaneous frequency and instantaneous amplitude (time-frequency-energy plane).

Let us start by forming the analytical form $Z_j(t)$ of each IMF $m_j(t)$ and its Hilbert Transform $\widehat{m}_j(t) = HT\left(m_j(t)\right) = \frac{P}{\pi} \int \frac{m_j(t)}{t-\tau}\, d\tau$:P is Cauchy principle value:

$$Z_j(t) = m_j(t) + i\, \widehat{m}_j(t) \tag{3.8}$$

From this equation we can rewrite the analytical signal as:

$$Z_j(t) = a_j(t)e^{i\theta_j(t)},\ where \begin{cases} a_j(t) = \sqrt[2]{\left[C_j^2(t) + \hat{C}_j^2(t)\right]} \\ \theta_j(t) = \arctan\left(\frac{\hat{C}_j(t)}{C_j(t)}\right) \end{cases} \tag{3.9}$$

The instantaneous frequency is given by $\omega_j(t) = \frac{d\theta_j(t)}{dt}$, and Hilbert spectrum of each IMF can be written as:

$$H_j(\omega, t) = a_j(t)e^{i\int \omega_j(t)dt} \tag{3.10}$$

3.3.2.1 Marginal Hilbert spectrum

now we can define marginal Hilbert spectrum of any mode as:

$$h_j(w) = \int_0^T H_j(w,t)dt \quad : T \text{ is the mode duration} \tag{3.11}$$

at a first glance, one can find the marginal spectrum analogous to Fourier decomposition, but the spectral components are different; as Fourier frequencies are constant over the sinusoidal harmonics in the data window [19] [20], thus providing

an energy distribution to trigonometric component with associated amplitude and frequency through the whole data window. On the other side, the ones that appear in (4.11) are time-dependant, thus providing an energy distribution in a probabilistic sense to local appearances of associated frequencies.

3.4 On the feasibility of HHT for better NILM

The quest for the perfect "signatures" to identify each device has been always the hottest debate topic in NILM community. Whether to invest more effort in engineering a few physical features that are extremely useful for all or most categories of devices or to better direct those efforts to enhance "learning" algorithms that look for learned features for every case on its own.

One of our goals in this work is to explore how to exploit empirical mode decomposition and Hung-Hilbert transform to get

- Better NILM event detection: more temporal accuracy and less false alarms
- Better features and thus better signatures. Analysing the transient part of the current signal at "ON" events can be very helpful to build a hybrid signature (combining steady and transient features) for various appliances.

3.4.1 Methodology

We followed two parallel tracks to approach the concepts of EMD and HHT for NILM signals: In the first track (green coded in Figure 3.3): the goal is to customize EMD for low cost NILM solution while taking the nature of NILM signals into consideration; an algorithm (FBE: **F**ilter **B**ased **E**MD) was designed for NILM signal analysis with applications in Event detection and feature extraction.

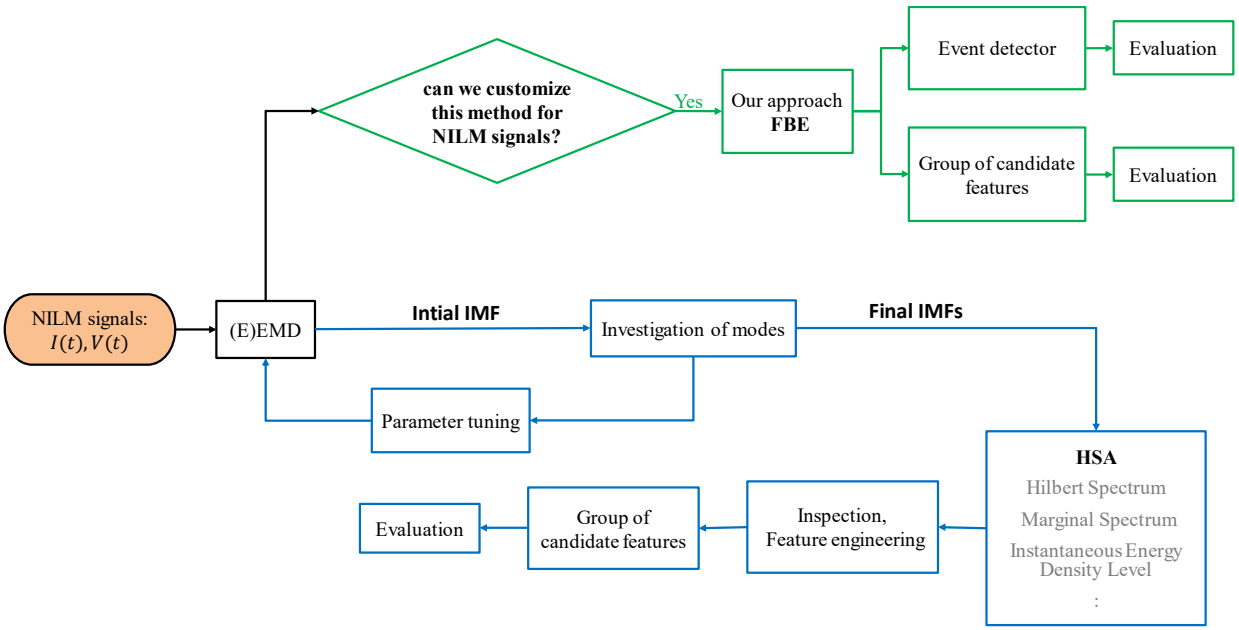


Figure 3.3: Methodology scheme for exploring HHT in NILM applications

In the second track (blue coded in Figure 3.3): the goal is to apply the classical HHT and use the physical quantities after the Hilbert spectrum analysis (marginal spectrum, instantaneous frequency curves ..) to get a group of useful features and evaluate them. In the next paragraphs both tracks are discussed in detail.

3.4.2 Filter-Based EMD inspired algorithm

A novel algorithm for NILM signal analysis is proposed, suitable for low-cost smart meters solution. We call it filter-based EMD inspired algorithm (FBE) as its first phase was motivated by Empirical Mode Decomposition and its modes extraction procedure, yet FBE is not equivalent to EMD and FBE-modes are different from the EMD intrinsic mode functions. FBE algorithm aims to break down the overall multi-component current signal into several mono-component signals using adaptive filters, whose cut-off frequencies are calculated from the detected extrema (no cubic curve interpolation as in EMD) and depend on the power of the signal.

3.4.2.1 Detailed description of FBE algorithm

The FBE algorithm [92] has two stages: initial isolation and refinement. A flow chart of the first stage is shown in Figure 3.4. The total current signal is divided into blocks of M points, and the algorithm is applied on each block separately.

Mode extraction follows these steps:

Step 1) Detect indices of the local maxima $[i_1, i_2, \dots, i_L]$ in the current signal block, then estimate the highest frequency f_{max} of the signal using the mean distance between those indices, if $D_k = [i_2 - i_1, i_3 - i_2, \dots, i_L - i_{L-1}]$ is the difference vector of the local indices then this frequency is calculated as:

$$f_{max} = \frac{F_s}{\overline{D}_k} \tag{3.12}$$

where F_s is the sampling frequency and $\overline{D}_k$ is the mean value of the difference vector D_k.

Step 2) The estimation of f_{max} allows us to separate the input signal into 2 parts in the spectral domain; the first one contains the frequencies above f_{max}, while the second one is composed of the frequencies below f_{max}. The separation is implemented using a low pass filter LPF with a cut-off frequency $f_{max} - \Delta_1$, where Δ_1 is a small margin to ensure that f_{max} is included in the higher frequency part after separation.

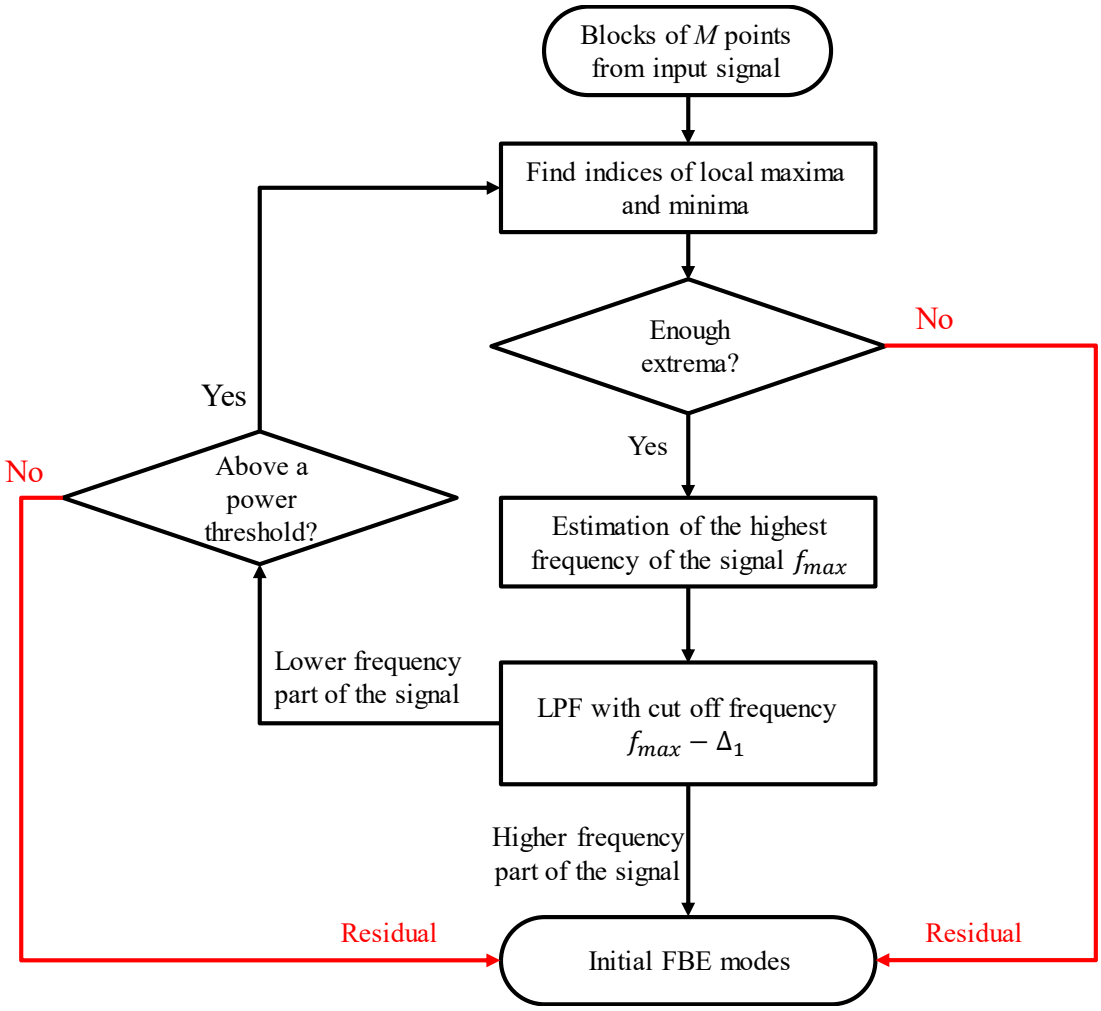


Figure 3.4: First stage of FBE algorithm, [92]

Step 3) The output of the LPF is the residual signal r. if the total signal is x then the first mode is: $m_1 = x - r$, contains the frequency f_{max} along with all spectral components below it, we set $f_1 = f_{max}$ as the frequency of this mode.

Step 4) Extract other modes using the same procedure in steps 1-3 from the residual signal. Extraction stops when the power of the residual signal is below a certain threshold, which is relative to the power of the input signal, or when the number of

local maxima in the residual signal is smaller than a threshold (we chose 3 in our case).

The desired output of FBE algorithm is the initial modes $m_1, m_2, \dots, m_N$ and their estimated frequencies $f_1, f_2, \dots, f_N$ where N is the number of the extracted modes, so the current signal can be written as:

$$I[n] = \sum_{k=1}^{N} m_k[n] + r[n] \tag{3.13}$$

Our goal here is to isolate mono-component signals. However, the peaks of the input signal are close to each other and different in amplitude, so each resulted initial mode till now may contain a substantial part of the signal spectrum with multiple frequencies, and the estimated frequency for this mode can be any of those in this interval, so in the next stage we re-apply the first phase on each resulted mode, this step allows us to refine the separation as illustrated in Figure 3.5 .

A better estimation of the frequencies can be calculated using the indices of the local extrema in the modes rather than the original current signal. For each mode, let f_k be the estimated frequency. The algorithm continues as follows:

Step 5) Check if there is a peak at the f_k, by passing the mode through a (narrow) band pass filter BPF $[f_k - \Delta_2,\ f_k + \Delta_2]$, where Δ_2 is a small margin, the original mode can be written as:

$$m_k = m_{\mathrm{BPF}} + m_{rest} \tag{3.14}$$

m_{BPF} is the BPF output while m_{rest} is the part that does not pass through the filter.

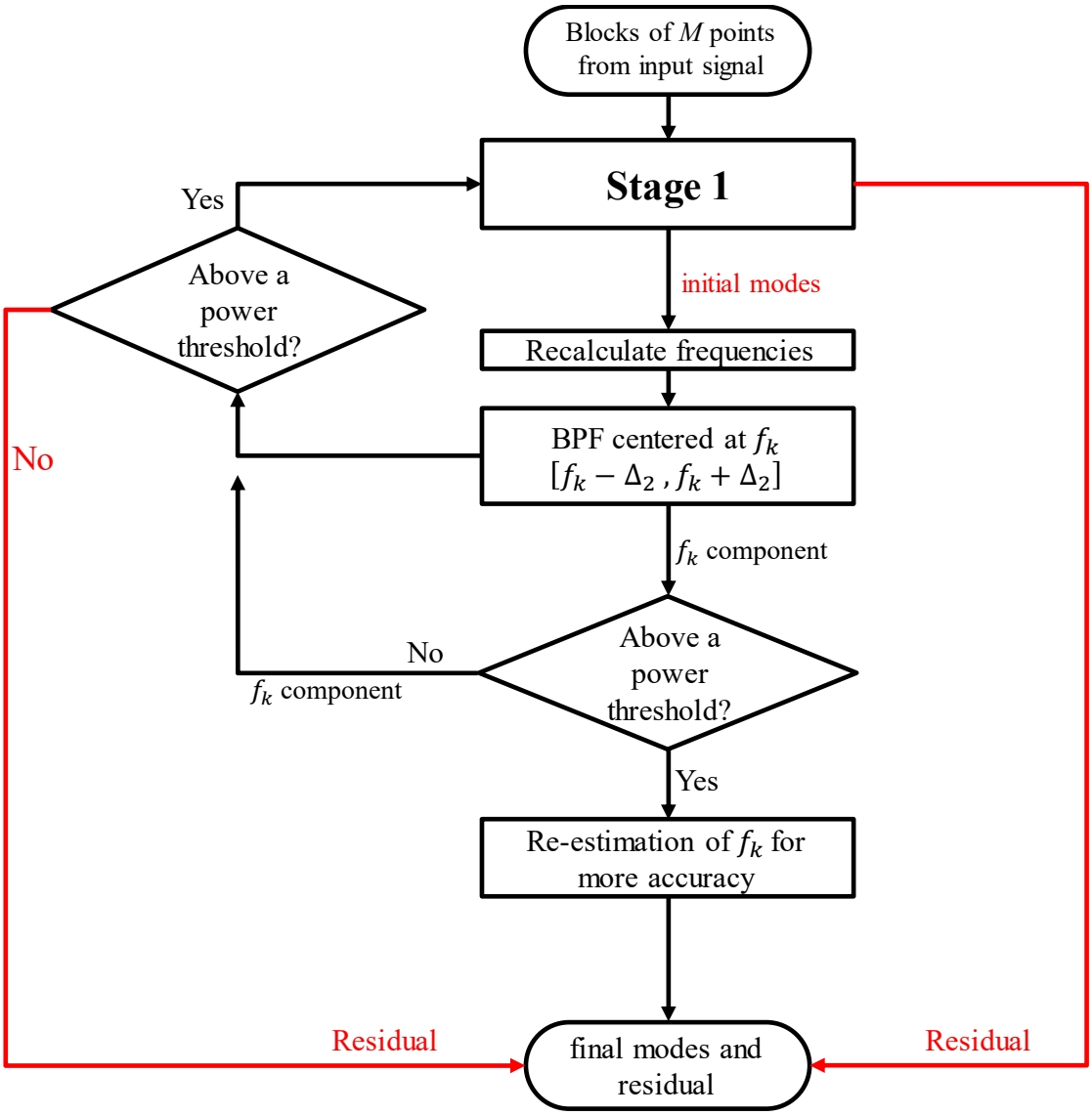


Figure 3.5: FBE algorithm

Step 6) If the power of m_{BPF} is larger than a defined threshold, then there is a mono-component mode at f_k, and the rest signal m_{rest}, whose spectrum lies outside the interval $[f_k - \Delta_2 , f_k + \Delta_2]$, is passed again to stage1 to check if there is any possible modes left.

Step 7) If the power of m_{BPF} does not exceed the threshold, then there is no peak at f_k, and the whole mode is passed back to stage 1.

The previous steps allow us to decompose the input current signal into a set of mono-component modes, in addition to a residual signal. This residual signal contains the low frequency part of the current signal and possibly some low power high frequencies resulted from the step 7 in case their power is less than the threshold. Now let us look at an example application of FBE on real NILM current signal.

3.4.2.2 FBE outputs in steady state and Transient state

A subset of current signal from BLUED dataset is addressed here, where FBE is applied on the steady state area before a "switch on" event of a fridge, see Figure 3.6. The plots in the left column are in the time domain, while the plots in the right column are in the frequency domain (with Fourier transform), the first row is the original current signal, and the resulted modes are in the next rows, the algorithm decomposes the input current signal into three modes at frequencies 60, 180, 300 Hz, which are the usual harmonics as expected.

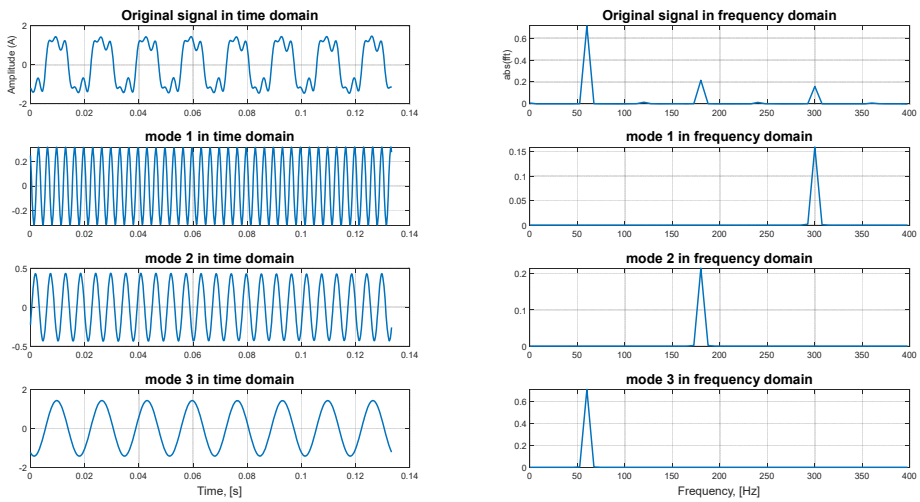


Figure 3.6: Example of FBE application (steady state)

Then moving to the transient state in Figure 3.7 and Figure 3.8, where two successive blocks, each with a duration of 125 *ms*, were considered. The algorithm produces only one mode in Figure 3.7 at the fundamental frequency 60 *Hz*, the amplitudes of the peaks at the third and fifth harmonics are very small compared to the amplitude of the peak at 60 Hz. It is important to indicate that these small peaks will join the residual signal at the output of the FBE algorithm, which results in a high-power residual signal.

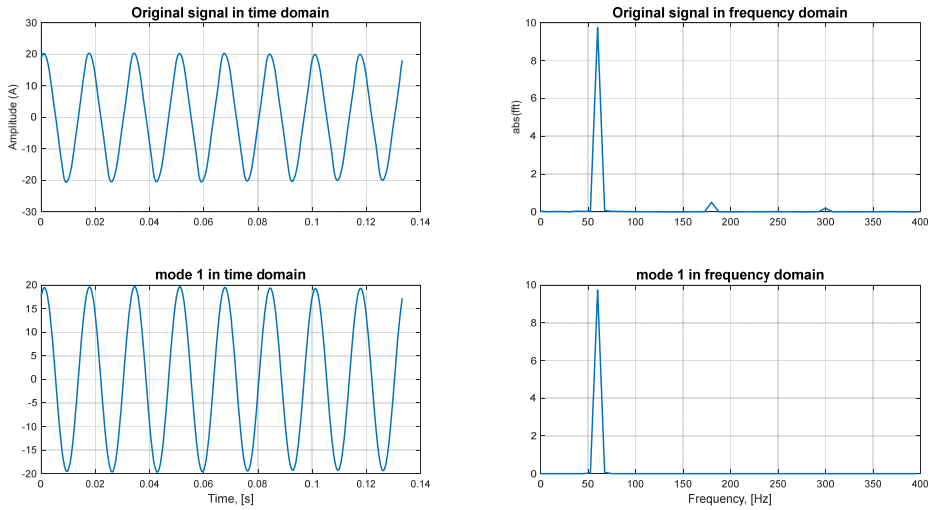


Figure 3.7: Example of FBE application (Transient state, first block)

In Figure 3.8, the algorithm produces seven modes for the second block of the transient signal. The fundamental mode extends over a wide interval [25, 110] Hz, so the algorithm divides this interval into several small intervals; each of them is considered a separate mode.

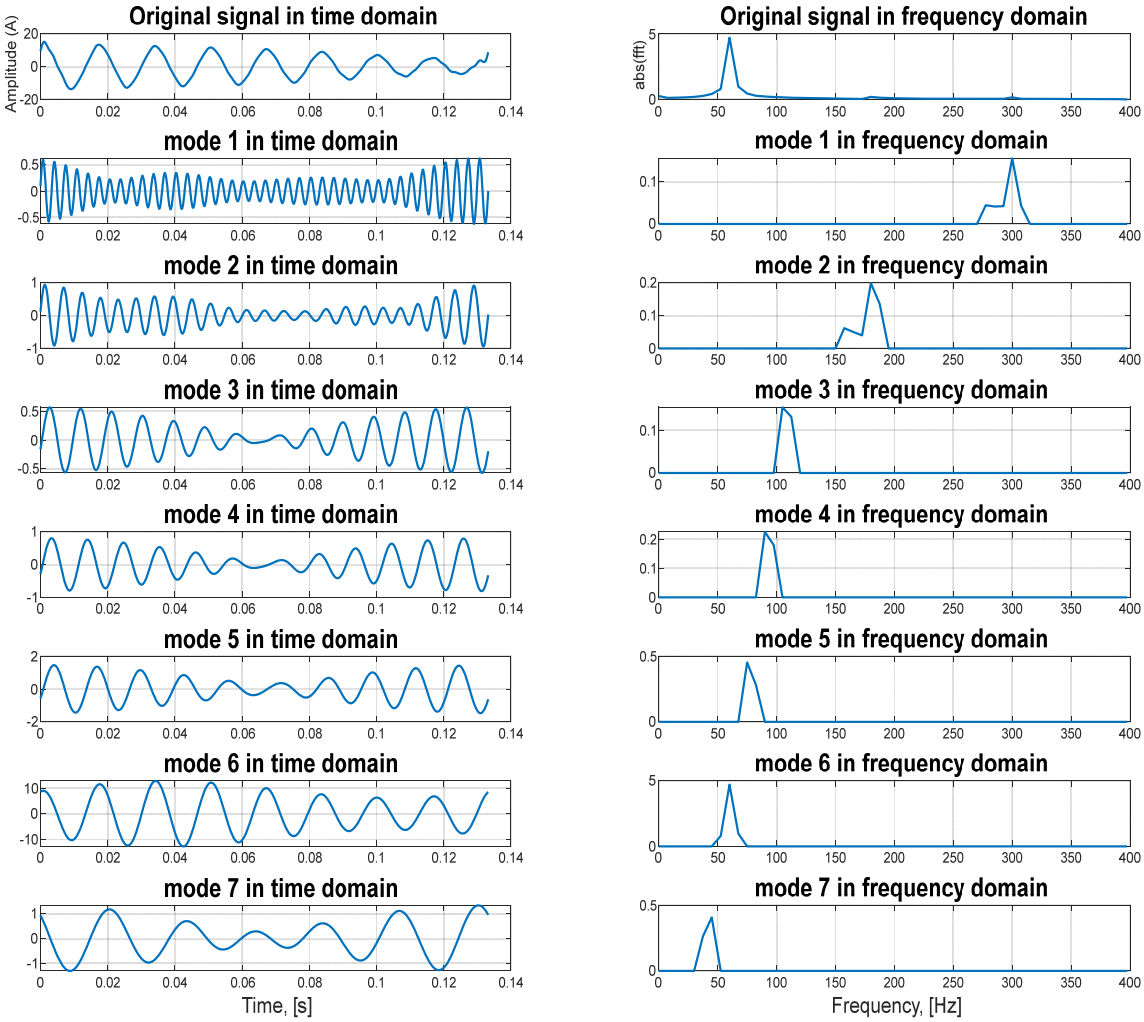


Figure 3.8: Example of FBE application (Transient state, second block)

In the next section, an event-detection method will be developed depending on the information extracted from the modes and their spectral content.

3.4.2.3 FBE-based event detector

The event detection FBE based procedure works in two steps: decomposition and examination. The decomposition stage starts with applying FBE algorithm to break down the overall multi-component current signal into several mono-component

signals, whose cut-off frequencies are calculated from the detected extrema and depend on the power of the signal.

During the examination stage: a few criteria are applied on the modes in order to detect possible events in the current signal and ensure their correctness. An event is defined as a change in the signal power above a specific power threshold, and the change duration exceeds a defined time threshold as well.

Current signal consists of steady state intervals and transient short intervals in time domain. The transient state starts when an event occurs, and continues for a short time (in the order of 1 second) while steady state prevails for the other ~98% of the time and its spectrum composed primarily of a fundamental frequency (usually 50 Hz or 60 Hz) and its harmonics.

In the transient state the signal behaves differently. Experiments show that some harmonics may disappear from the signal spectrum, while new spectral components may emerge between harmonics or below the fundamental frequency; accordingly, we can define the best rules to apply on the extracted modes and use the them to solve our problem (detect events, extract features, track special frequency and so on). In the following we will show the criteria we chose to construct an event detector for NILM signals.

3.4.2.3.1 Criteria for FBE-Event-detector

The choice of the desired criteria depends on the observations made by visual and numerical inspections of a variety of events. To clarify the intuition behind this process let us start by examining a practical example shown in Figure 3.9: this signal is taken from the aggregate current measurements of BLUED Dataset , where

an event "switch ON" took place at $t_{ev} = 94.3\ sec$ in time domain (top), this event caused a local turbulence around t_{ev} as can be seen from the spectrogram in the middle (where the turbulence is also magnified), the bottom plot shows how did the cardinality of the modes change from 3 to 10 then back to 3 again in a very short duration (less than 1 s).

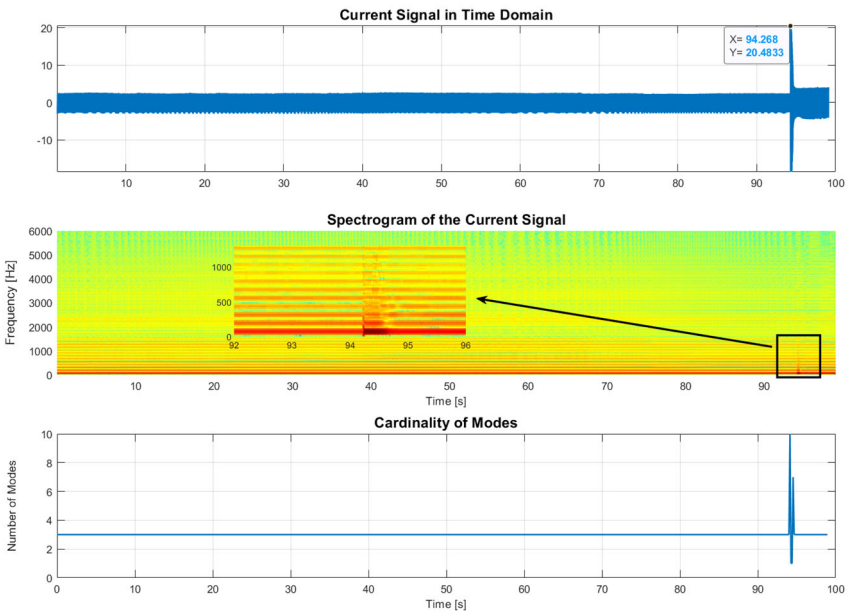


Figure 3.9: (Top) a subset of aggregate current NILM signal around an Event in time domain (middle) spectrogram with zoom-in on the event area (bottom)

The following criteria are chosen building on these observations (higher number of modes at an event, new spectral content) among others from various events:

1st Criterion) Cardinality of the modes

The number of modes in steady state is equal to the number of harmonics in the current signal, while in the transient state, the number of modes may increase or decrease, as in Figure 3.9, depending on the emerging spectral components/modes and disappearance of the harmonics peaks.

2nd Criterion) Existence of newly emerged frequencies

Since the FBE algorithm provides an estimation of the mode frequencies, we can check for the presence of new spectral components between harmonics and below the fundamental frequency as follows:

Let $F_{mod} = \{f_1, f_2, \dots, f_N\}$ be a vector of the frequencies of the resulting modes, check if : $\boldsymbol{f_{min} = min(F_{mod}) < f_0}$ (3.15)

Or $\boldsymbol{\exists\, f_i \in\,]kf_0\,, (k+1)f_0[}$ (3.16)

where f_0 is the frequency of the fundamental mode. The existence of such new emerging frequencies would indicate the presence of events, as in Figure 3.10.

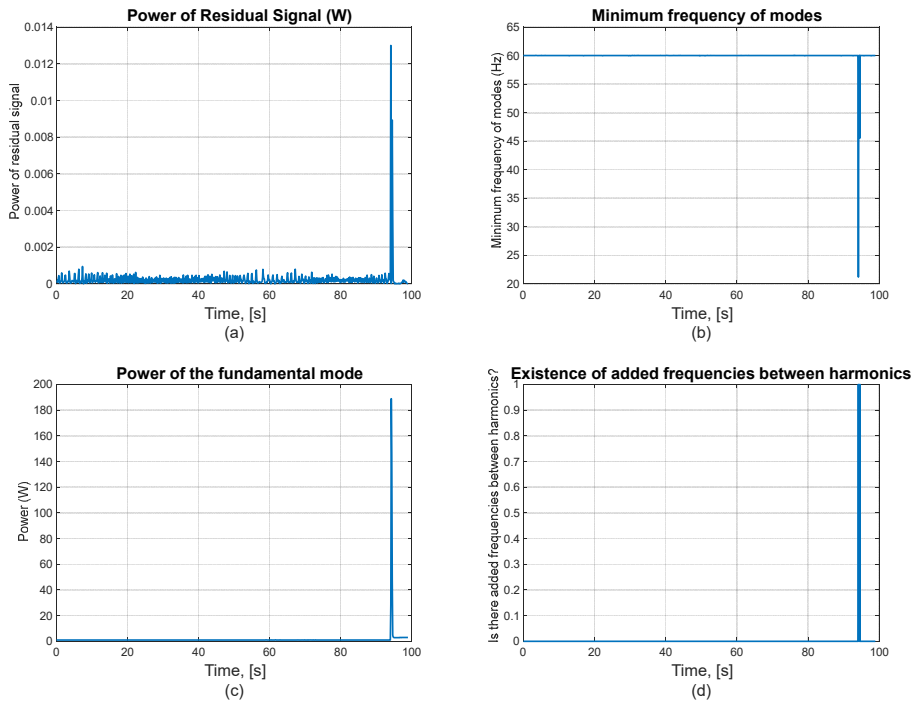


Figure 3.10: Visualization of some criteria

3rd Criterion) Mean power fluctuations of the modes

Most of the signal power is concentrated around the fundamental frequency component, so calculating the instantaneous power of this mode is approximately equivalent to calculating its power, and events can therefore be detected by observing the changes in the power level of this mode.

The power of other modes at the harmonics may also change at the event. This change depends also on other factors like the event type (ON or OFF), which appliance caused the event...etc, so we can use these changes for event detection.

4th Criterion) Power fluctuations of residual signal

The power of the residual signal can also be used to detect events. As the results show, residual signal power may increase in the transient state compared to the steady state (as the harmonics will probably join it in the first transient blocks, see Figure 3.7).

3.4.2.4 Numerical evaluation of the FBE-based event detector

A pre-processing operation for the real data (BLUED Dataset [15]) was done at first, then the event-detector was tested. We will describe the data preparation at first so that the interested reader can fully reproduce the results.

3.4.2.4.1 Data preparation

Before applying FBE algorithm on the measured current signal, an initial preparation process is conducted on current and voltage signals to efficiently reduce algorithm complexity as shown in Figure 3.11, this process can be adapted according to the target application. In event detection for example: it is known that

the time duration of the event is much smaller than that of steady state, so the mode extraction is only applied when necessary. This process is as follows:

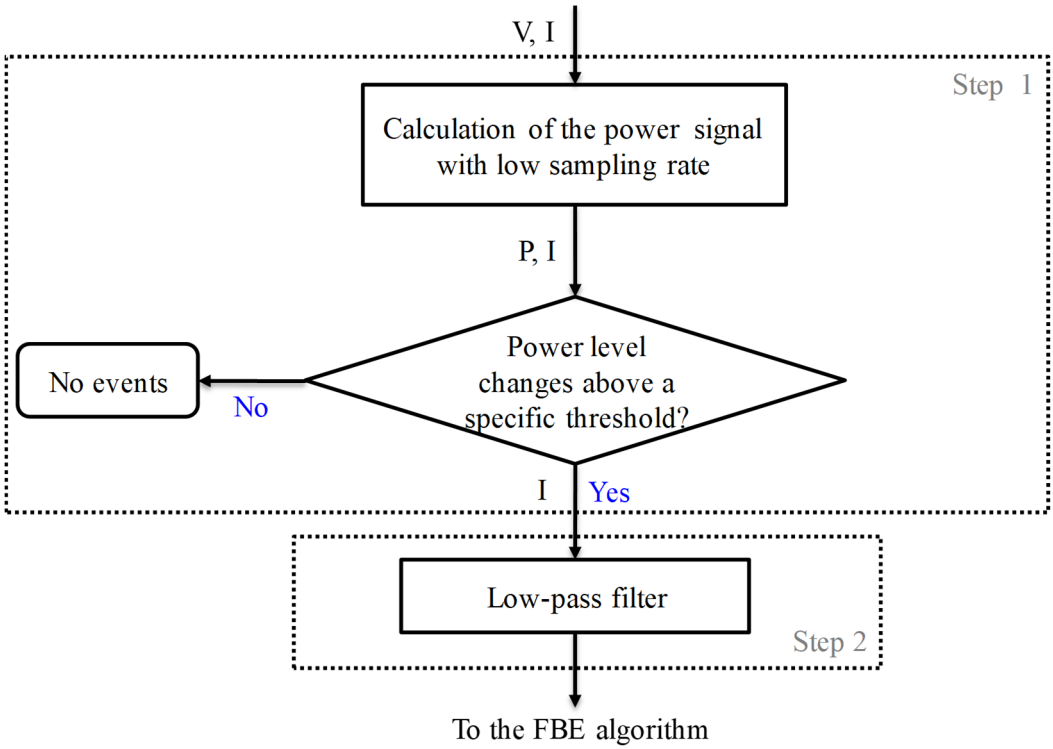


Figure 3.11: Data preparation

Firstly, we start by calculating and down-sampling the power signal, such power signal can be provided in the ground truth data, as in BLUED Dataset, where the current signal sampling rate was $12\,KHz$, whereas the power signal is down-sampled to 60 Hz only.

Next, we check the fluctuations in the power signal

$$P[k] = \frac{1}{M}\sum_{n=kM}^{(k+1)M-1} V[n] \times I[n] \qquad (3.17)$$

where $P[k]$ is the mean power over the time interval $[kT, (k+1)T]$, corresponding to the discrete interval $[kM, (k+1)M-1]$, M is the block length and $V[n]$, $I[n]$ are the sampled versions of the measured voltage and current, respectively.

If the changes in the power signal were negligible (under a defined threshold), then the observed time interval does not contain any events, and the algorithm skips the corresponding current block, the threshold should be chosen carefully here as events of small consumers under it will be lost. On the other hand, if a noticeable change in the power level is detected, then an event or more can be the reason for it, but it can be also a false alarm since large changes in the power level can occur even in the steady state of the signal depending on the devices working at this particular moment.

Secondly, we apply a low pass filter on the current signal, which allows us to reduce the number of spectral components and results in a smaller number of modes after extraction. Small values of the cut-off frequency will reduce complexity greatly, yet there is a possibility of information loss, while larger values of the cut-off frequency increase the complexity. For the event detector, we chose 400 Hz as a cut-off frequency of this initial LPF whose output is passed to the FBE algorithm.

3.4.2.4.2 Event-detection performance

FBE algorithm was applied on the signal blocks that may have events (after the two steps of data preparation in Figure 3.11). We utilize the previously chosen criteria in the next step to reduce false alarms and detect any possible missed events, a final test on the events is conducted, in which the duration of the event as well as the power levels around it are quickly checked by means of a median filter, such tests

can be done on the fly in real time applications. The final results are presented in Table 3.2 and Table 3.3 for both phases A and B of BLUED dataset; we used the following performance metrics:

$$\boldsymbol{P_{recall}} = \frac{\boldsymbol{TP}}{\boldsymbol{FN+TP}} \tag{3.18}$$

$$\boldsymbol{P_{precision}} = \frac{\boldsymbol{TP}}{\boldsymbol{FP+TP}} \tag{3.19}$$

Where TP is the number of true detected events, FN is the number of missed events, FP is the number of false alarms (detecting events while they are not present).

Table 3.2: Results of the FBE-based Event detector on BLUED Dataset

Phase	Total Events	TP	FN	FP	P_{recall} %	$P_{precision}$ %
A	904	897	7	17	99.23	98.14
B	1578	1390	188	197	88.09	87.59

The measurements in phase A have relatively small variations compared to those of phase B due to the number and nature of the appliances connected to each phase [15], as a result, most of event detectors have excellent results when applied to phase A, while the benefit of the FBE based event detector is clear in Table 3.3 when looking at its results on phase B in comparison with the event detector of [93] (by the team that collected BLUED dataset).

Table 3.3: Performance comparison with [93]

BLUED Dataset	Phase A		Phase B	
%	FBE	[93]	FBE	[93]
Precall	99.23	98.16	88.09	70.40
Pprecision	98.14	97.94	87.59	87.29

3.4.2.5 FBE for NILM feature extraction

Several features were extracted from the classical EMD modes in other fields of research, which makes it tempting to experiment with those features on FBE modes as well. However, the critical point is how to evaluate such features. That is the next step in our research in chapter 4, where we look closely at feature selection for NILM application. For now let us just define exemplary features that we had actually experimented with later on: $\{F_i^{FBE}\}_{1\leq i\leq 5}$ are a group of binary values represent the existence of a mode with a new emerging frequency other than the usual harmonics as a result of FBE decomposition of the transient part of an event, in mathematical terms:

$$\boldsymbol{F_i^{FBE}} = \begin{cases} \mathbf{1} & \text{if } \exists\, \boldsymbol{f_i} \in [(\boldsymbol{i}-\mathbf{1})\times \boldsymbol{f_0} + \boldsymbol{\Delta f}, \boldsymbol{i}\times \boldsymbol{f_0} - \boldsymbol{\Delta f}] \\ \mathbf{0} & \boldsymbol{otherwise} \end{cases} \tag{3.20}$$

3.4.3 NILM Feature extraction with classical HHT

Huang proposed the concept of intra-wave frequency modulation as a better physical approach to represent nonlinearly deformed waves [94] instead of resorting to higher harmonics per Fourier decomposition. In this section we use this perspective and consider the effect of each “ON” event as a nonlinearity introduced

to the aggregate current signal. Our goal is to "mine" the transient part for effective features. Initially, the scheme of research process was set as in Figure 3.12:

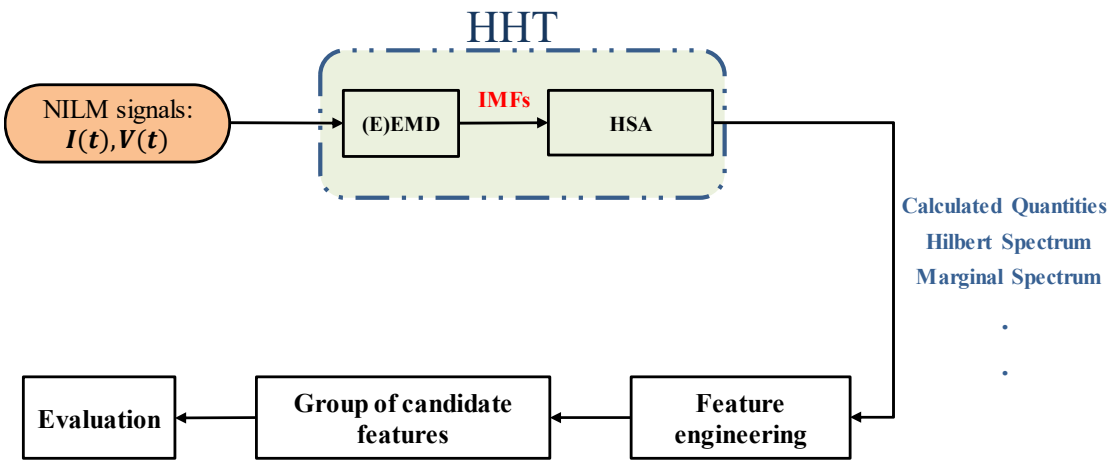


Figure 3.12: Initial research scheme for HHT feature extraction

However, the application faced some practical issues:

- the EEMD is empirical by definition and has several parameters left to the user to choose (mostly in an ad-hoc manner according to the application at hand).
- The temporal length of transient state varies greatly among devices.

So as can be seen in Figure 3.3, another in-between step for parameter tuning was added, each step will be discussed in the following paragraphs.

3.4.3.1 Parameter tuning

Looking for the most "suitable" intrinsic modes depends greatly on the purpose of the analysis. EMD has a lot of room for customization as it is an algorithm with two linked loops by definition (i.e. HHT is not a transform in the wide accepted sense as it lacks a solid mathematical basis), and many recent papers [95]–[97] presented insights for how to choose the parameters of sifting process in particular. In [98],

Wu and Huang demonstrated that for a low number of sifting iterations (ca. 10), the EMD procedure can be still considered equivalent to a bank of dyadic filters with a minimum mean frequency ratio of 2 between the neighbouring components; on the other hand, a high number of sifting iterations leads the IMFs to approach constant amplitude, thus losing the desired physical sense of the resulting modes.

In our study, we aimed for the mode that contains the fundamental grid frequency (50 Hz in Europe). One of the issues for an "out of the box" application (per [99] implementation) is the appearing of the same spectral content in more than one mode (as in Figure 3.13: modes 4 & 5 in 2 events of a Fan Device), which indicates a "mode mixing problem", even though EEMD is said to significantly reduce the mixing percentage in most cases.

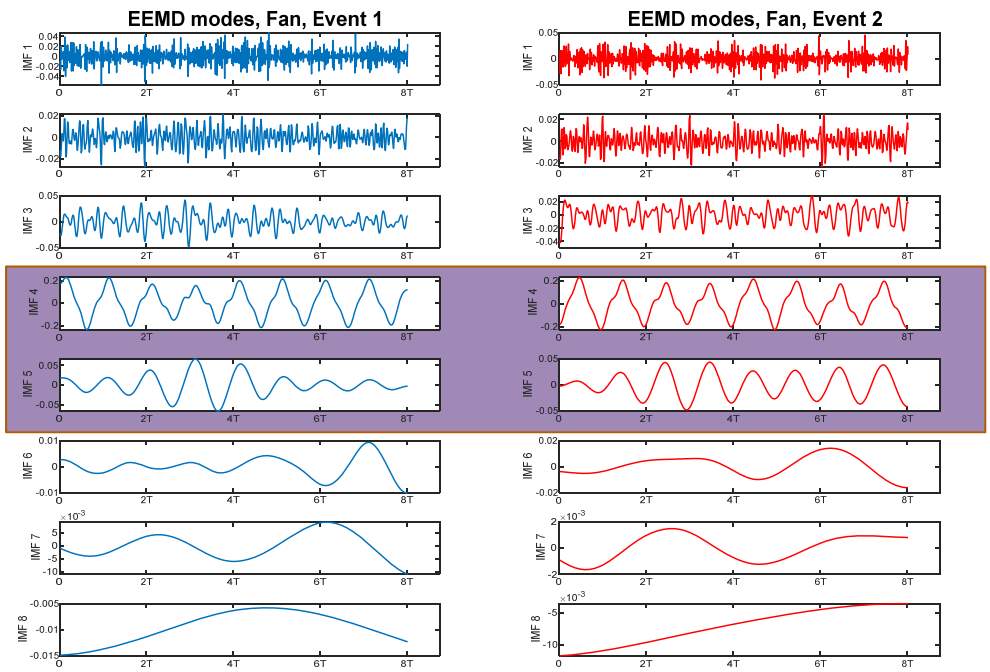


Figure 3.13: Mode mixing example (highlighted), Device: Fan, 2 Events

The goal of the tuning operation is to get modes with good frequency separation (i.e. alleviate mode mixing as much as possible) while keeping the required properties of the modes, the important parameters for our application are:

a. length of the addressed part of the signal **TranL**, as an integer number of periods (a special parameter for our NILM application)
b. maximum number of sifting iterations **MaxItr** (general EMD parameter)
c. number of realizations **NR** (EEMD parameter)
d. standard deviation of the added noise **Nstd** (EEMD parameter)

For exploring the possible values of the first parameter TranL, we designed an empirical method for NILM signal segmentation to accurately determine the transient state boundaries in the next section; numerical experiments to find suitable values for the other three parameters are discussed afterwards.

3.4.3.2 Empirical NILM signal segmentation

One of the challenges in NILM research field is determining the limits of transient and steady states precisely. While this method was designed for tuning the parameter TranL at first, yet it was applied more than once later for different purposes in our research work (Pseudo code can be also found in appendix 1). similar “expert heuristics” approaches in the literature can be found in [20], [21], yet they were used as event detectors, while in here this method is applied after event detection and takes events indices as input and outputs the limits of transient state, along with event type (as a sanity check). The segmentation procedure can be divided into six steps as follows.

For each event in the signal do the following:

[step 1] Take a window from the steady state before the event. This window is defined by its start index and its length. The default value of the start index is 1 second before the event, if a close event exists (less than 1 seconds), then set start index as the middle point between the two events (same process for the index of the window end as well). The window length is set to 200 milliseconds by default and can be smaller for close events; let us call the resulted window "left window". The minimum required length of the left window is N: the number of samples per period.

[step 2] Take another window around the event, let us call it the "total window". It starts at the end of the left window and extends to the steady state after the event, as illustrated in Figure 3.14, its length is set to the value of maximum possible transient length of all appliances (3-7 seconds from manual inspections). This window should not reach the next steady state if the next event is too close.

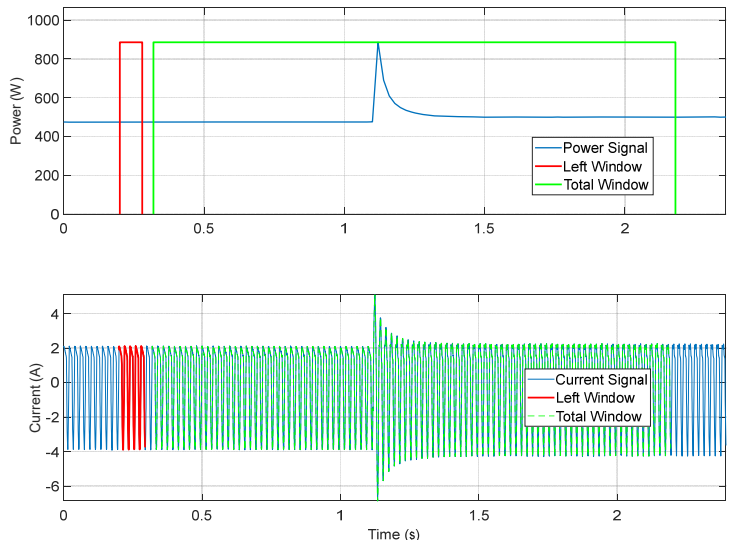


Figure 3.14: Example event with left and total windows in power and current signals

[step 3] Determine the type of the event by comparing the changes in power levels before and after the event. For this purpose, we check the mean power of the left window P_{left} (ca. 23 W in Figure 3.15) and the power level at the end of the total window P_{right} (ca. 35 W). Declare an 'ON' event if the power level increased after the event ($P_{left} < P_{right}$), as displayed in Figure 3.15.

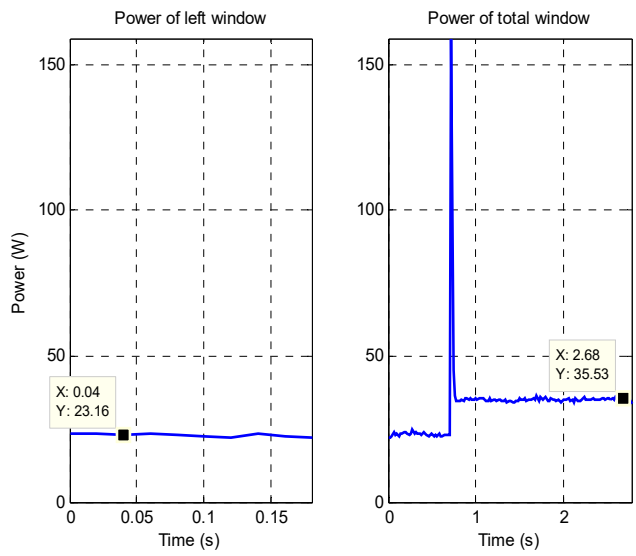


Figure 3.15: Checking event type based on comparing left and total windows

[step 4] Subtract a repeated version of the left window from the total window, the type of event is irrelevant here because when the event is 'ON' we get the appliance signal I_{app}, and when it is 'OFF' we get a reversed version of the appliance signal $-I_{app}$. In this step we applied FIT-PS representation [26], [47], [100] to preserve the phase during subtraction

[step 5] Determine the transient start index by calculating a threshold as the average between the max value of the first *k* periods of resulted appliance signal I_{app} (we took *k=3* periods) and the maximum of the whole signal, then searching for the first sample whose amplitude exceeds this threshold.

[step 6] Determine the end index of the transient state: search for the first group of *M* samples with a smaller mean power difference than a predefined threshold (calculated as a percentage of the change in power level caused by the event). which is equivalent to finding the start of the next steady state. Set the end index as the first sample of those *M* samples.

Two examples of transient state limits are illustrated in Figure 3.16, start index (red marker) and end index (black marker) along with the "old" index of event from ground truth information (HELD1 dataset [14]). This method will be used for correcting the indices of the events (replacing by the index of the transient state start index).

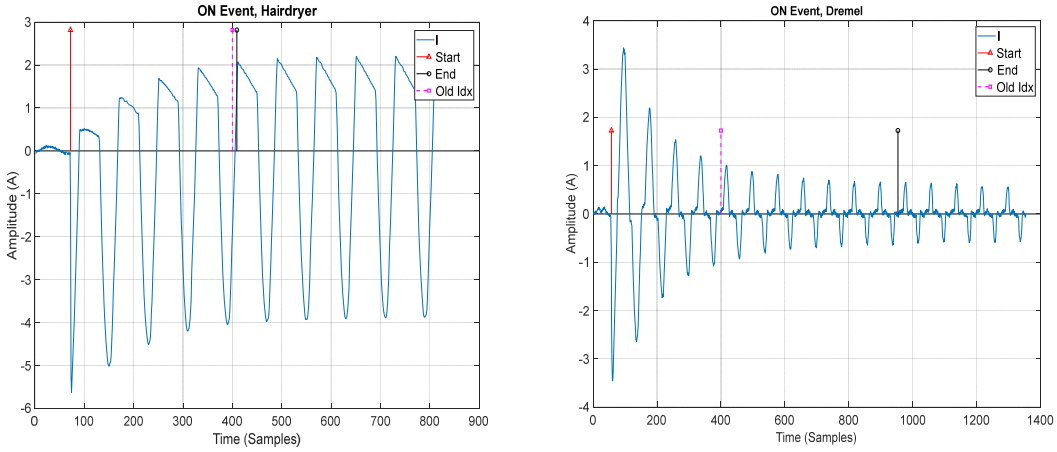


Figure 3.16: Transient state limits along with the event index (GT)

3.4.3.2.1 Temporal accuracy

One way to estimate the maximum error in detecting the start index of the transient state is to draw several events from single appliance measurements together and observe the span of the signals at the start of the event (in analogy to eye diagram concept). An example is shown in Figure 3.17, where the time axis starts 2 periods

before the event (1 period is $\frac{F_S}{f_0} = \frac{4000}{50} = 80$ samples); the red arrow represents the index of all events.

One can easily observe the maximum error in this case is about 40 samples. By repeating the same experiment on other devices in HELD1 Dataset, the worst case (Radio) was 72 samples (less than a period).

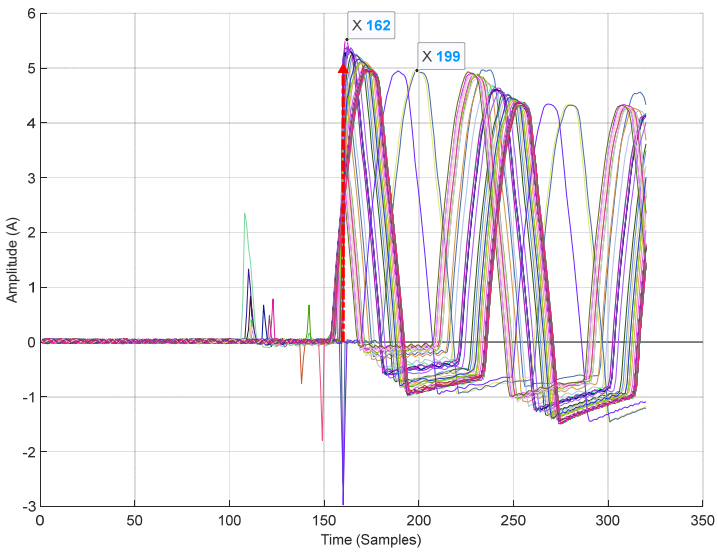


Figure 3.17: 100 Events of Hairdryer together

we consider the temporal accuracy of the start index of transient state to be 1 period ($20\ ms$). First period will be discarded later as to avoid this error. For the end index, the same strategy showed a span of 2-4 periods. An example is illustrated in Figure 3.18 with black arrows drawn at the detected end indices. The transient length TranL will be chosen as 12 periods (the longest detected transient state for the devices in our experiment).

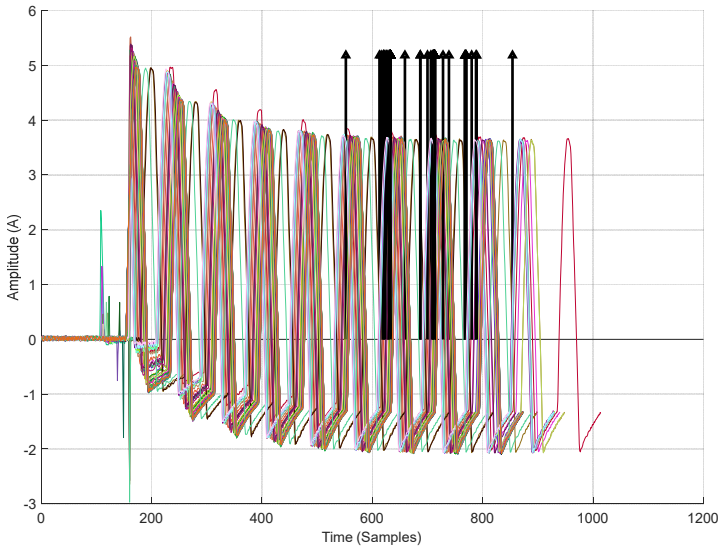


Figure 3.18: 100 Event of a Hairdryer, transient ends with proposed method

3.4.3.3 numerical simulations

A group of simulations with several parameters configurations were oriented towards exploring the best properties of the resulting mode as well as the limits and desired values of parameters, Getting one mode around grid frequency was unattainable for all devices with the same parameters so we relaxed the requirements to accepting a second mode with that frequency as long as its amplitude is less than 10% of the amplitude of main mode.

Example Insights: Figure 3.19 shows that increasing *Nstd* (0.55) intensifies the mode mixing problem (a value of 0.2 is chosen); another example in Figure 3.20 illustrates an interesting point about the maximum number of iterations: the shape of main 50 Hz mode is the same for any number above 10, while this was not the case for other types of signals reported in literature, as in [98], yet for NILM signals going beyond 10 is counter-productive.

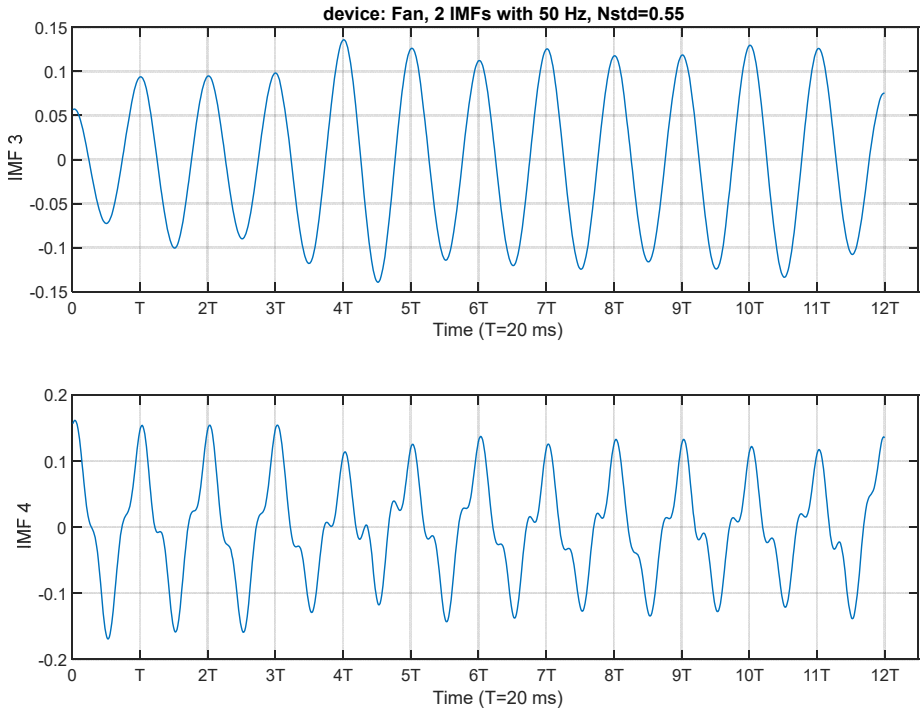


Figure 3.19: More mode mixing due to higher Nstd value

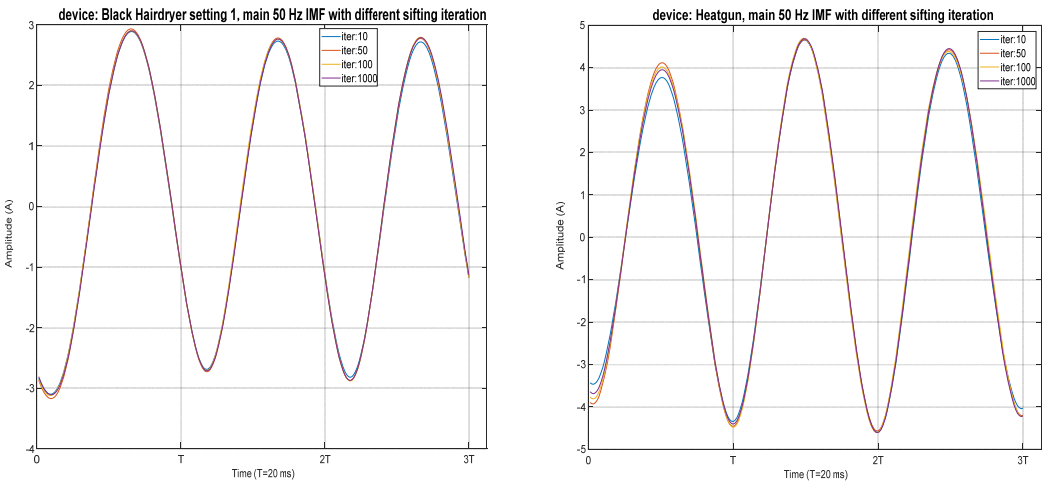


Figure 3.20: Effect of tuning MaxItr on 50 Hz modes

To sum up: The desired outcome is having as little sifting iteration as possible while maintaining mode mixing phenomena to a minimum. The chosen parameters are

{Nstd=0.2, NR=200, MaxItr=10, TranL=12), Figure 3.21 shows two examples of resulted IMFs.

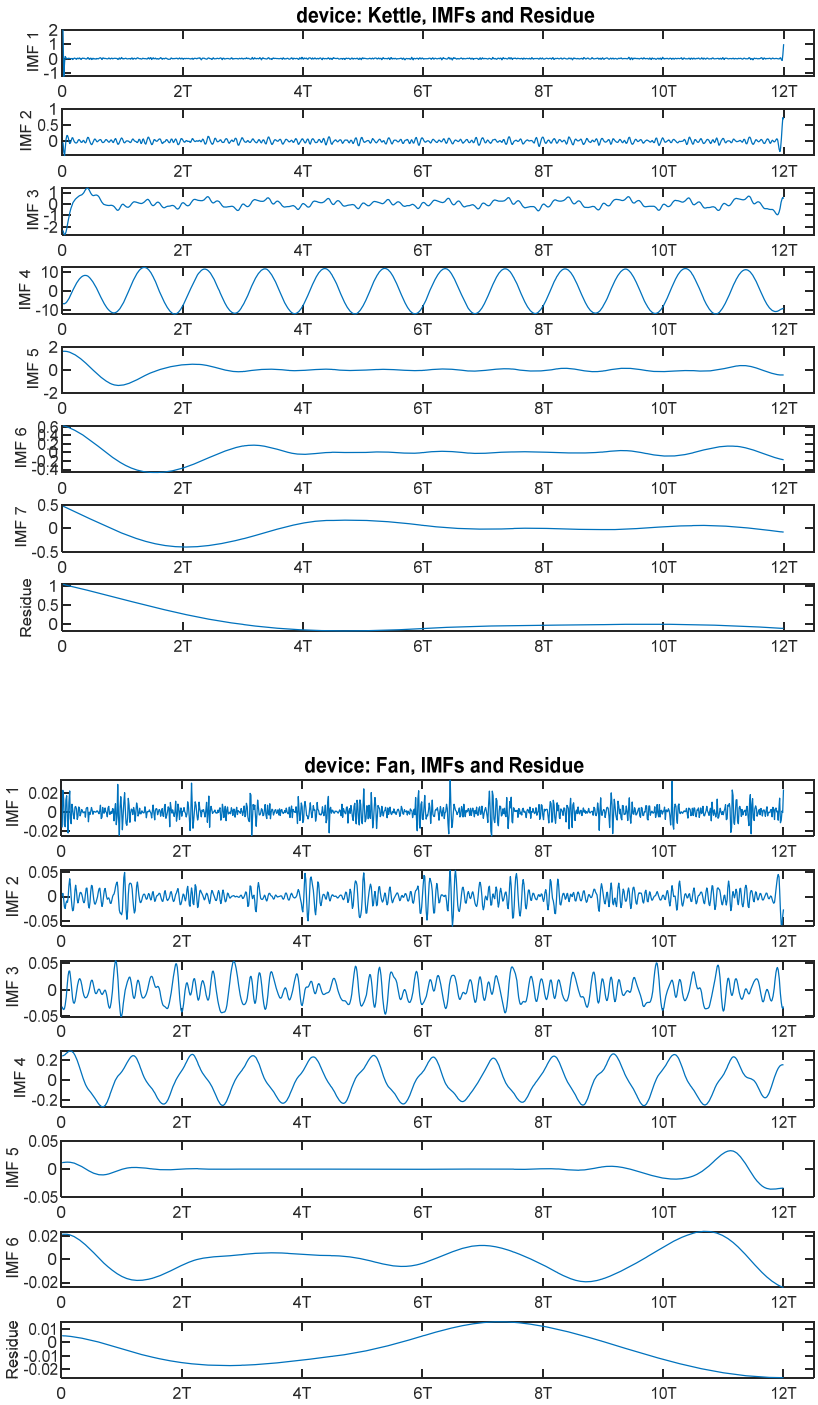


Figure 3.21: Examples of final IMFs after fixing Parameters

3.4.3.4 Exploring HSA outputs and candidate Features

Engineering features the "manual" way is losing popularity against the "relatively new" deep learning algorithms for the general NILM applications [34], yet it is still the way to go when the NILM problem target special devices or certain environments (commercial buildings. factories ...); in such cases domain knowledge and specifically-designed features can be of great value.

To motivate the process of feature extraction we start by looking at two main quantities that Hilbert spectum analysis provides us with:

1) "IF" – Time plane; its accuracy is in the order of a sampling period (IF stands for instantaneous frequency).
2) The marginal spectrum of the main mode (the instantaneous amplitude of main mode as function of IF).

3.4.3.4.1 IF-Time plane

Figure 3.22 presents IF as a function of time for 2 devices at 4 random events taken from HELD1 dataset (G3: aggregate measurements with up to 6 devices). The addressed signal is a subset of 12 periods of the current signal taken one period after the start of the transient state. The signal of the device was isolated by in-phase steady state subtraction and HHT was applied (EEMD then HSA). By looking at all events-curves: the behaviour of IF is consistent: IF took higher frequencies at first, then stabilized approaching steady state (with values around Grid frequency), the "shape" in both regions is approximately the same for each device. In terms of features candidates: the level and duration of IF oscillation (average) can be interesting, the shape (envelope) of oscillation from the very start seems also

promising. In Figure 3.23, the upper envelopes (peaks) of IF(t) is shown for 6 appliances (with a period duration at the start of the transient state start.

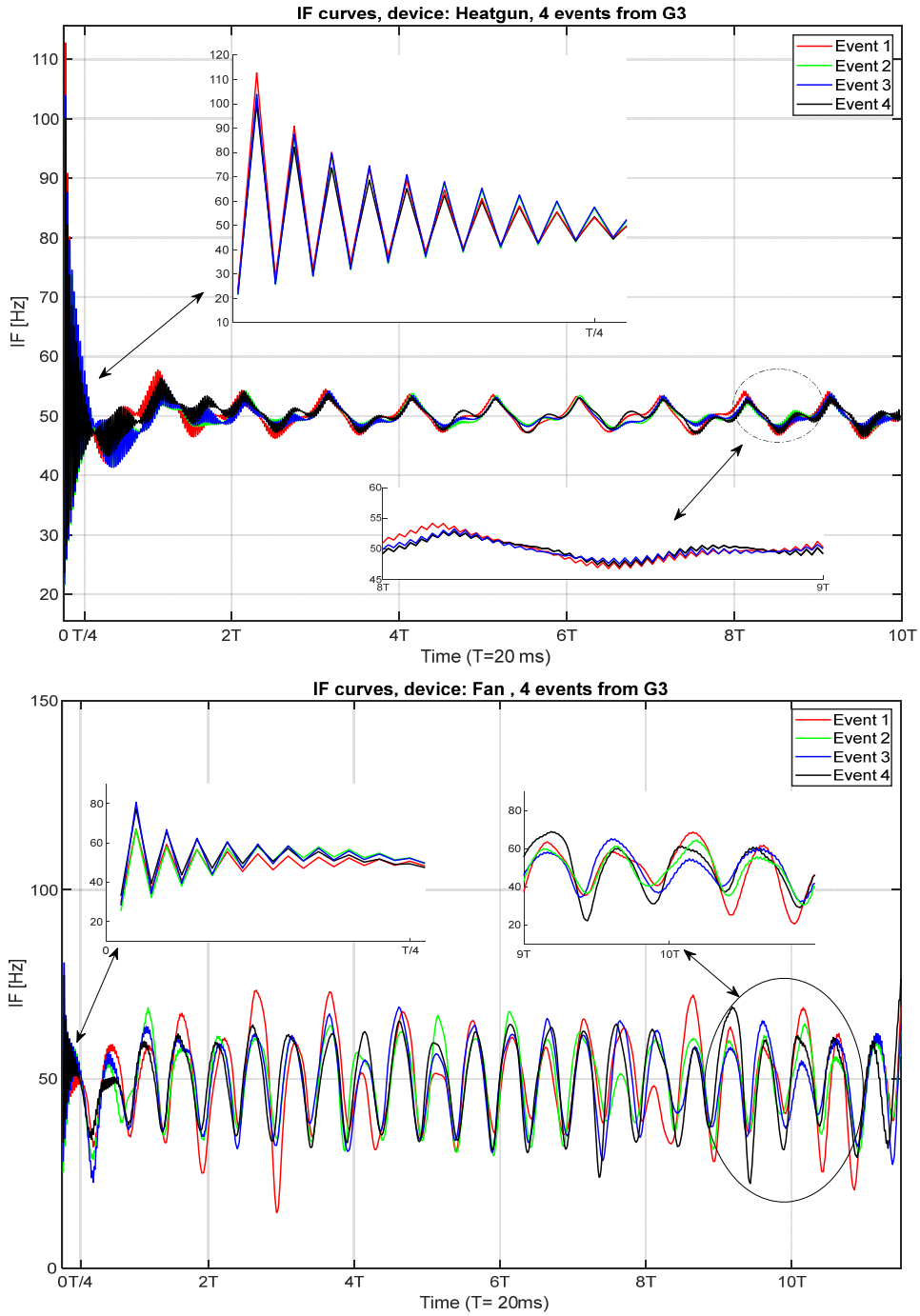


Figure 3.22: Inspecting IF curves for 2 devices at 4 random events each

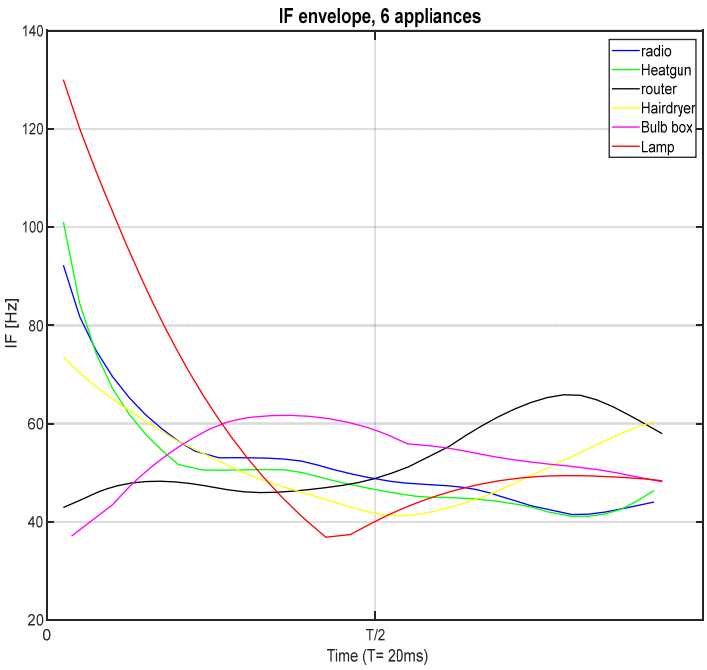


Figure 3.23: IF envelopes for 6 appliances

One approach to make use of this noticeable difference is to fit the IF curves with a second order polynom and consider the coefficients as candidate features as well.

3.4.3.4.2 marginal Hilbert spectrum

Marginal spectrum of the main mode bears important information about the probabilities of local appearances of a specific frequency (as discussed earlier), In Figure 3.24 we show that the bandwidth of the main mode spectrum as well as the amplitude (can be averaged for more accuracy) are consistent among all instances.

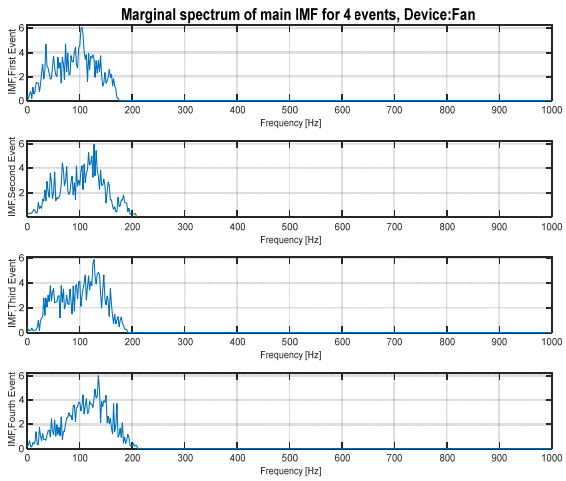


Figure 3.24: marginal spectrum for 4 Events, Fan

Another comparison is made in Figure 3.25 among the marginal spectra of 4 devices, where the bandwidth and the amplitudes are also promising as candidate features.

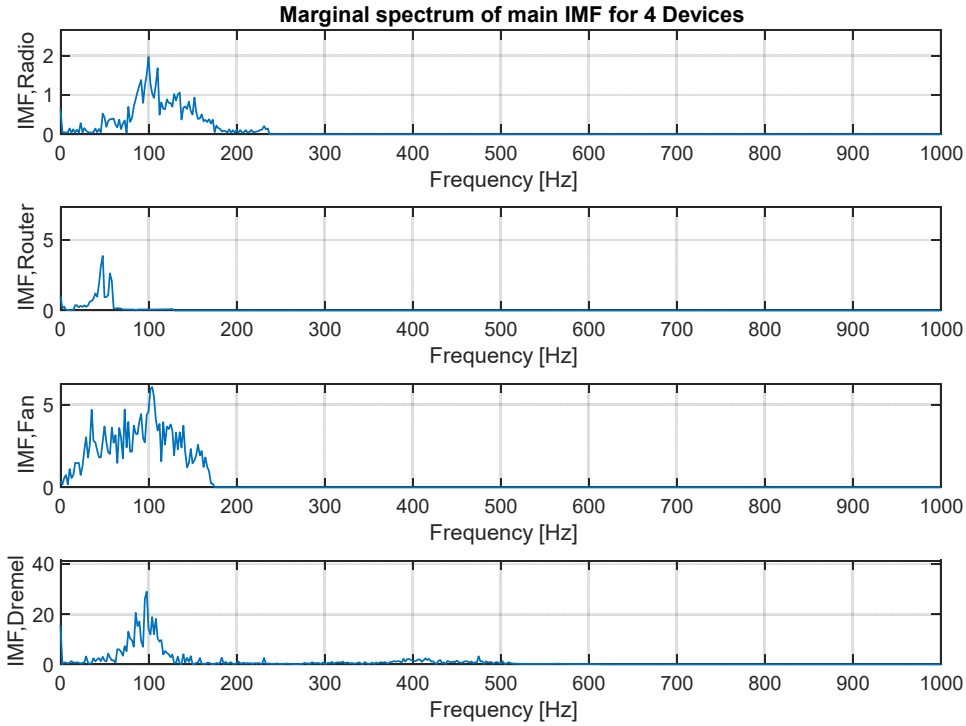


Figure 3.25: comparison of marginal spectra of 4 devices

3.4.3.4.3 Intial evaluation

An assesment of a certain feature is to be done with respect to other features, which is the subject of the next chapter; yet to test the feasability of example feature extracted in this study, we make a simple calssification experiment with 2 different classifiers: Linear Discriminant Analysis classifier LDA and SVM (linear kernel) on G1 & G3 of measurements from HELD1 dataset (G1 is a collection of measurements with 1 active device at most, randomly selected from the list in , while G3 allows up to 6 active random appliances at the same time).

The input features are (pairs of them and all 3 are tested):

- BW: the bandwidth span of the marginal spectrum of main mode,
- AA: the average amplitude of the spectral components,
- MA: the maximum amplitude in the marginal spectrum,

Table 3.4: Classification rates with extracted marginal features only

Classifier/Group		Kernel		Correct classifications (%)	
		BW & AA	BW & MA	MA & AA	AA & MA & AA
LDA/G1	pseudolinear	43.5	78.3	68.8	77.8
SVM/G1	Linear	44.5	81	70.3	78
LDA/G3	pseudolinear	37.5	60.3	51.3	66.2
SVM/G3	Linear	38.4	67.2	60.1	71.2

while those features alone can not be used as an “optimal” set of load signatures for classification, yet the pair (BW & MA) has potential and can be added to other features for better classification rate, the systematic way to do that is discussed in the next chapter.

3.5 conclusions and closing notes

One of the limitations of HHT transform for our application is its high computational cost due to the EMD nature, for example: processing a 40 minutes measurements file (with 100 “ON” events) from HELD1 dataset for extracting the modes takes around 40 minutes (parallel processing on a computer with 6 core i7

2.2 GHz processor). The other calculations of Hilbert spectrum analysis and proposed feature extraction methods take negligible time compared to EMD execution time (less than 1 %), in spite of our efforts to shorten the calculations: iterations were limited to 10 as discussed previously, and the extraction procedure is stopped at the fifth mode (IMF_5) as the required mode around grid frequency is always in the first five modes (depending on the noise level and perturbations in the input signal), yet higher modes (lower number of extrema) needs less time of processing per sifting step.

In this chapter, we investigated the possible advantages of applying HHT on NILM signals. As a solution for high-computational EEMD modes extraction, we proposed another approach for NILM signal analysis and presented a novel event detector based on this approach, the numerical results showed a state-of-the-art performance on BLUED dataset. In parallel, we applied the classical HHT and analysed the resulted IF curves and marginal Hilbert spectrums, then proposed a set of interesting features and made an initial evaluation for three of them.

The usability of HHT features is limited by the high computational cost of their extraction, A “NILM on the cloud” solutions may benefit from such features, yet they are impractical for any offline real-time NILM smart meter in the time being. An extensive test of extracted features from IF curves and marginal spectrum for discrimintation of identical devices is one of the near future tasks as well as addresseing their “importance” against other features with respect to classification rate, processing power and sampling frequency.

4 Feature Selection For NILM Systems

Choosing the best set of features can be essential for the success of any machine learning algorithm. The hard quest for such features in the NILM field started more than two decades ago and still comprises a hot research topic nowadays. Active and reactive powers were sufficient features for Hart [6] in the early nineties, since then several approaches were developed [30], [81], [101], [102] to extract a lot more features towards finding the "optimal" load signature; which motivated the use of various ad-hoc combinations of features from different categories in the recent works of [40], [103], [104]. However, employing all those available features in designing the decision rule can add an avoidable complexity, e.g., classification performance reached its maximum after a small subset of features in Figure 4.1 with negligible improvement after that, going further might even lead to worse classification results as [105] pointed out.

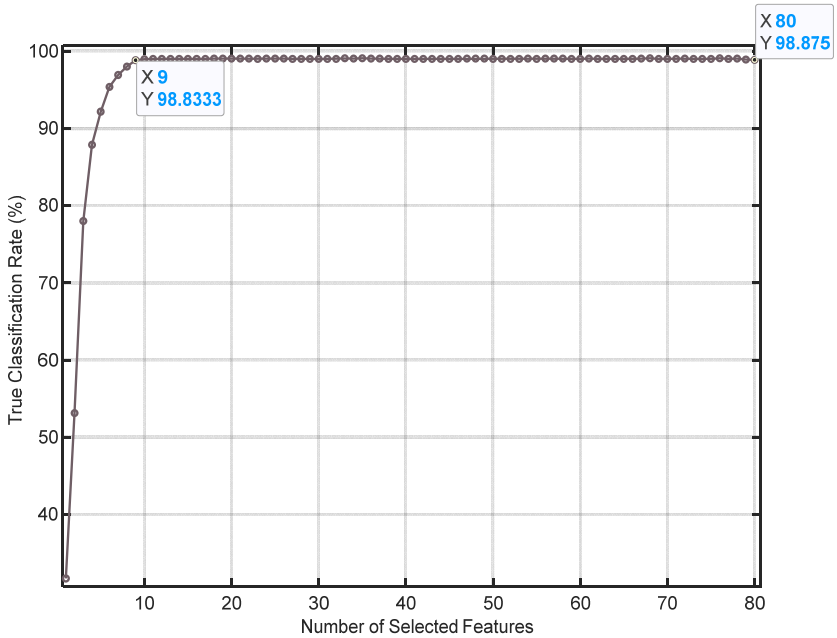


Figure 4.1: Example of performance plateau after using 10% of the total feature

A recursive feature elimination process was proposed in [106] and tested on single measurements Dataset (PLAID [56]), while [102] deployed the sequential Forward Selection algorithm "SFS", which assumes a certain classifier, to choose the best set of features for different types of Datasets.

In this chapter, A modified entropy-based feature selection algorithm is proposed and tested on single and aggregate measurements from Held01 Dataset [14], where the same set of appliances are present on both types and the number of ON/OFF events is the same for every appliance, then the tests are extended to BLUED Dataset [15]. Several classifiers are used after ranking to validate the approach on both datasets.

The remainder of this chapter is organized as follows. First section will present the mathematical background, then continues to explain the novel modifications to

entropy measures for NILM feature selection in detail. The second section describes the experimental setup, then presents a demonstration of concept and numerical simulations. Finally, section 3 briefly discusses some insights from the results and presents the main conclusion of our work.

4.1 mathematical Background and proposed algorithm

An effective set of features helps groups 'similar' objects together, this concept of similarity will be discussed further with a mathematical interpretation and proper tools to measure and exploit it.

4.1.1 Euclidean Distance, Similarity, and Entropy

An unsupervised feature selection algorithm for clustering is proposed in [107], where similarity $S_{i,j}$ between two data points P_i and P_j in the feature space depends simply on the normalized Euclidean distance $D_{i,j}$ between them, and is defined as:

$$S_{i,j} = e^{-\alpha D_{i,j}} \tag{4.1}$$

where α is a damping parameter.

For numeric multi-dimensional data points, the Euclidean distance $D_{i,j}$ is defined as:

$$D_{i,j} = \sqrt{\sum_{m=1}^{N_f} \left(\frac{x_{i,m} - x_{j,m}}{L_m}\right)^2} \tag{4.2}$$

where N_f is the number of features, $x_{i,m}$ is the mth feature of the ith data point, and L_m is the length of the space of the mth feature among all data points.

$$L_m = \max_i(x_{i,m}) - \min_i(x_{i,m}) \tag{4.3}$$

Similarity$S_{i,j}$ is high (≈1) for points close to each other, i.e. same cluster, and low (≈0) if they are far from each other. Entropy is then given as:

$$E_{Total} = \sum_{i=1}^{N} \sum_{j=1}^{N} E(S_{i,j}) \tag{4.4}$$

where *N* is the number of data points, and the function *E* is defined as [108]:

$$E(x) = -(x \times log\, x + (1 - x) \times log(1 - x)) \tag{4.5}$$

Entropy is low for both low and high values of $S_{i,j}$, but will be maximized for moderate values (around 0.5) which would result if the points are uniformly distributed in the feature space, yielding poor clustering possibilities.

According to [107], ranking of a feature is based on the calculated entropy after eliminating this feature; then the feature is important if the entropy has a much higher value without it, i.e. the feature reduces the ambiguity of the clustering result. In the second case, if the entropy remains unchanged after removing the feature, then it is not so important. Finally, if the entropy is smaller after removing the feature, then this feature is not important at all and should be ignored because it has a negative effect on the clustering performance. let us tweak this concept for NILM case.

4.1.2 NEFSA Design Rules

we call our proposed algorithm NEFSA (stands for **N**ILM **E**ntropy based **F**eature **S**election **A**lgorithm), where the key difference to the clustering problem is:

- A class may be composed of one or more clusters.
- A cluster may contain points from one or more classes.

So, by design, NEFSA should adhere to the following "rules" for calculating the similarity $S_{i,j}$:

1. If the distance $D_{i,j}$ between two points P_i, P_j is small and the points belong to the same class, then the similarity $S_{i,j}$ is high ($S_{i,j}$≈1, Entropy → 0).
2. If $D_{i,j}$ is large and the points belong to the same class, then $S_{i,j}$ is moderate ($S_{i,j}$≈ 0.5, Entropy ↑).
3. If $D_{i,j}$ is small and the points belong to different classes, then $S_{i,j}$ is moderate ($S_{i,j}$≈ 0.5, Entropy ↑).
4. If $D_{i,j}$ is large and the points belong to different classes, then $S_{i,j}$ is low ($S_{i,j}$≈0, Entropy → 0).

As a result, similarity takes its values in the interval [0.5, 1] if the two corresponding points P_i, P_j belong to the same class, while it takes values in the interval [0, 0.5] if they belong to different classes.

4.1.3 NEFSA principle

The importance of each feature is inversely proportional to the value of its entropy; NEFSA measures this importance by evaluating the distances among data points for each separate feature at a time and calculating the similarities according to the design rules, then measuring its entropy. A feature is considered more important if that entropy is smaller. This approach has the advantage of reducing complexity compared to the one in [107].

Features are ordered according to their entropy values in an ascending manner, where the first feature in the ordered list is the most important one. In the last step, we choose a subset of the best features from the head of the ordered list. The number of chosen features is preferable to be as small as possible to reduce complexity, but large enough to give the best achievable classification performance. The algorithm calculates the entropy of features one-by-one, separately, as shown in Figure 4.2 below.

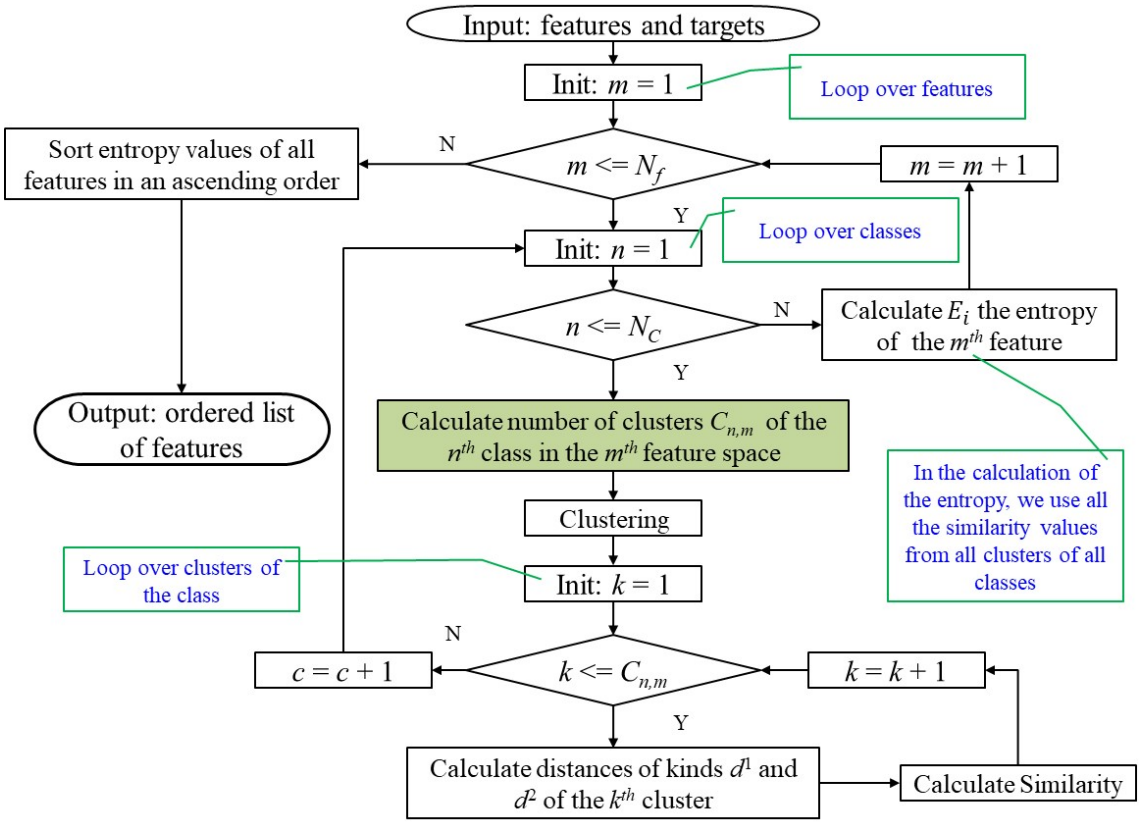


Figure 4.2: Flowchart of NEFSA Algorithm

Let F_m be the mth feature, first step is to calculate the number of clusters $C_{n,m}$ of each class in the feature space, where $n = 1,2, \dots, N_C$; N_C is the number of classes (appliances in our application); In addition, points of each class are grouped into clusters using a clustering algorithm (more on that in the following section).

We omit subscripts *m* and *n* for simplicity. In the second step, clusters of **each class** are processed separately, **where for the kth cluster**, we distinguish between two types of distances $d^{k,1}$ and $d^{k,2}$ as follows (Figure 4.3 for visualisation):

- $d^{k,1}$ is calculated between the points of the addressed cluster (i.e. same class and same cluster) and the centroid of this cluster; distances of this kind are preferred to be as small as possible, so that the cluster points are concentrated (coded green in the Figure 4.3).
- $d^{k,2}$ is calculated between the centroid of the cluster and the points of other classes. (**i.e. points of that class that lay in other clusters are not considered here**). $d^{k,2}$ distances are preferred to be as large as possible, so that we can distinguish points of different classes from the cluster under process (coded red in the Figure 4.3).

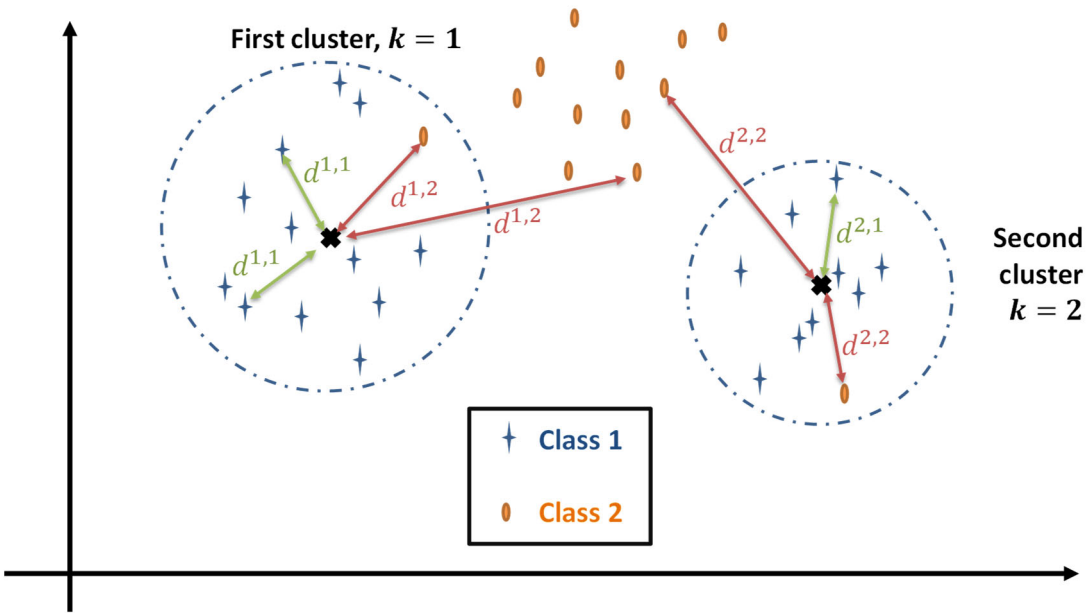


Figure 4.3: Visualising the defined distances in 2D feature space

The next step is to calculate similarity values using distances $d^{k,1}$and $d^{k,2}$.

for distances $d^{k,1}$ we define the similarity as:

$$S_i^{k,1} = \frac{1+e^{-\alpha^{k,1} d_i^{k,1}}}{2} \tag{4.6}$$

where $\alpha^{k,1}$ is a damping parameter. $i = 1,2,\dots,N_k$: N_k is the number of points of the kth cluster.

$S_i^{k,1} = 1$ for $d_i^{k,1} = 0$ ($E_m \downarrow$), and $S_i^{k,1} = 0.5$ for $d_i^{k,1} \to +\infty$ ($E_m \uparrow$)

The damping parameter $\alpha^{k,1}$ is determined by the allowed values for similarity after the maximum limits of feature span of the addressed cluster $L_m = max(d_i^{k,1}) - \min(d_i^{k,1})$, i.e. $d^{k,1}$ distance does not exceed $\frac{L_m}{2}$:

$$S_i^{k,1}\left(d_i^{k,1} = \frac{L_m}{2}\right) = 0.5 + \varepsilon \tag{4.7}$$

$$\text{then}: \alpha^{k,1} = \frac{2}{L_m} log \frac{1}{2\varepsilon} \tag{4.8}$$

For visualization: Figure 4.4 shows the relation between $S_i^{k,1}$ and $d_i^{k,1}$ for $L_m = 25$, and $\varepsilon = 0.01$.

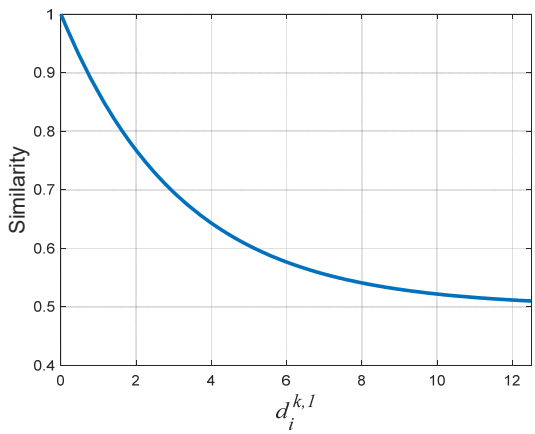


Figure 4.4: Similarity vs distance Type 1

The similarity $S^{k,2}$ for the distance $d^{k,2}$ is defined as:

$$S_i^{k,2} = \begin{cases} 0.5 & d_i^{k,2} \leq R_k \\ \frac{e^{-\alpha^{k,2}(d_i^{k,2}-R_k)}}{2} & otherwise \end{cases} \tag{4.9}$$

where R_k is the radius of the kth cluster (mean of $d^{k,1}$ values), $i = 1,2,\dots,M_n$, and M_n is the number of points that are not from the nth class. The case $d_i^{k,2} \leq R_k$ indicates that the ith point is located close to the cluster centroid, although this point belongs to another class, so in this case, similarity must be 0.5 and entropy should be enlarged. The parameter $\alpha^{k,2}$ is chosen so that:

$$S_i^{k,2}\left(d_i^{k,2} = l * R_k\right) = \varepsilon : l \in \mathbb{N} \setminus \{0,1\} \tag{4.10}$$

then $\alpha^{k,2} = \frac{2}{R_k(l-1)} log \frac{1}{2\varepsilon}$ (4.11)

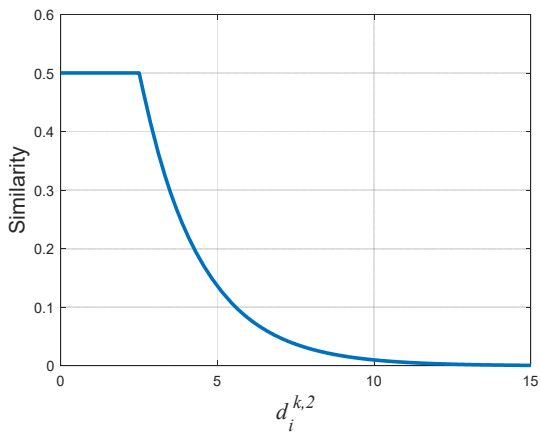


Figure 4.5: Similarity vs distance Type 2

Figure 4.5 shows the relation between $S_i^{k,2}$ and $d_i^{k,2}$ for $L_m = 25, l = 4, R_k = 0.1, and\ \varepsilon = 0.01$. the last step of the algorithm is to calculate the entropy E_m of the mth feature as follows:

$$E_m = \sum_{n=1}^{N_C} \sum_{k=1}^{C_{n,m}} E^{k,n} \tag{4.12}$$

where $E^{k,n}$ is the entropy calculated from the kth cluster of the nth class as:

$$E^{k,n} = \sum_{i=1}^{N_k} E\left(S_i^{k,1}\right) + \sum_{i=1}^{M_n} E\left(S_i^{k,2}\right) \tag{4.13}$$

Next paragraphs are dedicated to our preferred choices for determining the clusters.

4.1.4 Calculating Number of Clusters

4.1.4.1 Using K-means Algorithm

Based on the intuitive choice K-means algorithm, to determine the number of clusters $C_{n,m}$ of the nth class in the mth feature space (as in the flow chart in Figure 4.6 below) :

1) Set $C_{n,m} = 2$.
2) Perform clustering on the points of the nth class using the K-means algorithm.
3) Calculate distances between the resulted clusters, and compare them with a predefined distance threshold.
   a) If the smallest distance is less than the threshold, then the value of $C_{n,m}$ is not suitable; the algorithm stops and outputs the previous value ($C_{n,m} - 1$).
   b) If the smallest distance between clusters is larger than the threshold, then the class has at least $C_{n,m}$ clusters; increase the number of clusters ($C_{n,m} = C_{n,m} + 1$) and move back to step 2.

After calculating the number of clusters, we can use the K-means algorithm to group the class points into clusters.

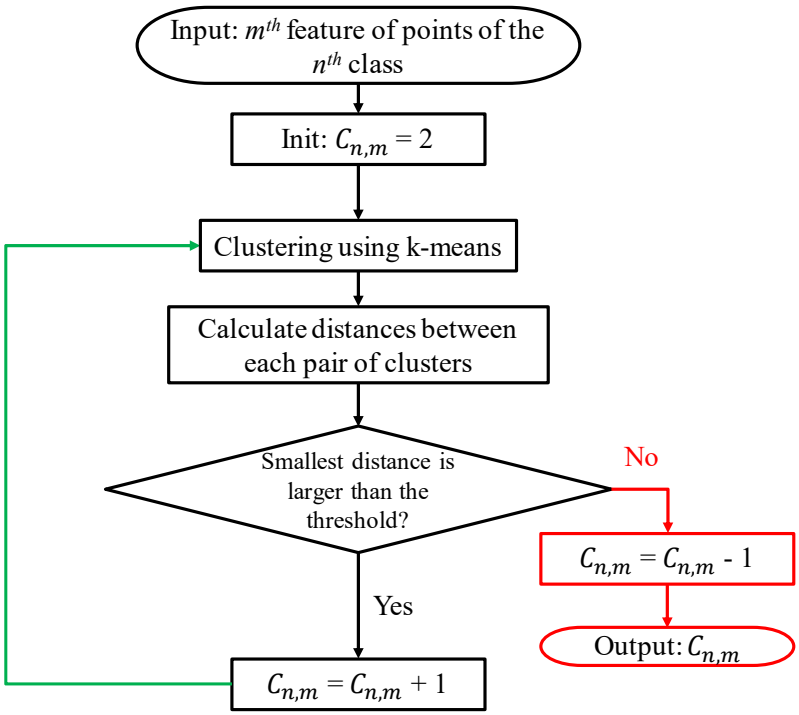


Figure 4.6: Calculating the number of clusters using k-means algorithm

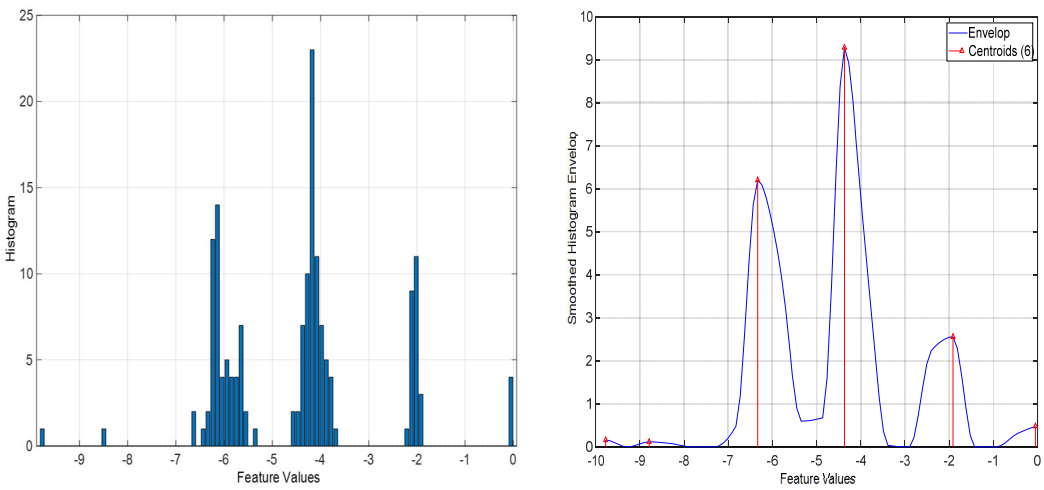
The random selection of initial centroids can lead to slightly different ordering of features if we repeat the algorithm, so another novel approach for our specific application is proposed in the next paragraph.

4.1.4.2 Using Histogram Peaks Detection (CHIPD)

To reduce complexity and get a more robust procedure than *K*-means for our purpose, we propose a 1-D **C**lustering Algorithm based on **HI**stogram **P**eaks **D**etection (CHIPD). The proposed algorithm calculates the histogram of the feature values, then try to detect its peaks. We assume that each peak is a cluster. The steps of the CHIPD algorithm are as follows:

1) Calculate the histogram of a vector (1-D) of feature values. In Figure 4.7 (a) an example of the resulted histogram from a real dataset is shown.

2) Smooth the envelope of the histogram in two phases.
   a) Apply a series of consecutive averaging filters as needed. Two filters were enough for our application.
   b) Sample the envelope of the histogram, then produce a new one using interpolation. In Figure 4.7(b) the resulted envelop after applying the two steps of smoothing on the histogram is shown.
3) Find the indices of the local maximums of the envelope, and set them as the initial values of the centroids. The number of the detected peaks is the number of clusters. In Figure 4.7 six peaks were detected.
4) Cluster all points based on the weighted Euclidian distance: calculate a threshold between each two consecutive clusters; which should be at the local minimum between the two consecutive peaks. Each feature value (point) is then compared with the calculated thresholds and assigned to the nearest centroid.
5) Calculate the final centroids as the expected value of the points in each cluster.



(a) (b)

Figure 4.7: (a) Histogram of feature values, (b) smoothed

4.2 NEFSA Assessment and Numerical Simulations

4.2.1 Experimental setup

NEFSA was evaluated on a chosen group of measurements from HELD1 Dataset [14], then was tested on BLUED Dataset [15], along with other feature selection algorithms.

The evaluation on HELD1[14] considered three groups of measurements:

1. G1: where only one appliance is active at a time.
2. G2: where up to 4 appliances can be active at a time.
3. G3 :where up to 6 appliances can be active at a time.

The goal is to evaluate the feature ordering performance in various scenarios of overlapping while keeping the same set of appliances, which is listed in Table 4.1.

Table 4.1: List of Appliances under Test

ID	Name	Power (W)
1	Radio	6.2
2	Heat gun (setting 1)	820
3	Router	9.2
4	Black desk lamp	20
5	Light bulb box	20
6	Kettle	2100
7	Black hairdryer (setting 1)	500
8	Fan	22
9	Rotary tool (Dremel)	30
10	LED lamp	1

A bank of 84 features is gathered at each event in this partial dataset. The addressed features are a group of frequently reported features in NILM research, along with some features under development in our lab (FBE and Marginal HHT features from chapter 3, entropy of band energies and statistical properties of the Hilbert envelope of the current signal, based on the work of [109], [110]).

Table 4.2 and Table 4.3 list all NILM features included in the experimental study, we will refer to these features by their given indices in the next figures and tables of this chapter so the reader can track any relevant feature by its index.

Table 4.2: Bank of NILM features, First part

Features	Indices	References
Active, Reactive and Apparent Powers (P,Q,S)	1-3	[6], [101], [30], [81]
Resistance, Admittance (mean and median)	4-7	[102]
Phase Shift	8	[6], [101], [30]
OER[1], Tristimulus[2] (T1, T2, and T3), THD[3]	9-13	[33], [102], [111]
Crest Factor[4], Form Factor, ICR[5], LogTimeAttack[6] ,Temporal Centroid[7]	14 - 18	[102],[33], [31]

[1] *Mean of odd harmonics over mean of even ones.*

[2] *Three features for lower, medium and higher harmonics (normalized by the sum of all harmonics).*

[3] *Total harmonic distortion.*

[4] *The ratio of peak values of a waveform to the effective value.*

[5] *Ratio of RMS values of first period over the last period for an addressed region of current signal.*

[6] *Measure of the needed time to reach maximum intensity (of the envelope).*

[7] *Weighted sum to identify the point in time where most of the energy of the signal is located (in average).*

Table 4.3: Bank of NILM features, second part

Features	Indices	References
PMR[8], MIR[9], MVR[10], MAMI[11], SSMR[12]	19 - 23	[102]
Various features[13] from the evelope of the current signal	24 - 28	Based on [109], [110]
First three temporal Moment-measures	29 - 31	[112]
Marginal HHT features	32-34	Chapter 3
Harmonics	35 - 54	[101], [30], [81]
FBE analysis	55 - 59	Chapter 3
Wavelets coefficients[14]	60 - 68	[81], [113]
Features of VI-trajectory[15]	69 - 76	[101], [114], [115]
VI-trajectory (non active current[16])	77 - 84	[30]

[8] *Peak- mean Ratio.*

[9] *Max Inrush ratio: first sample of the period of interest over the max value.*

[10] *Mean-variance Ratio.*

[11] *Max-Min Ratio.*

[12] *Max-Mean-Mean Ratio, highest spectral amplitude over the spectral mean.*

[13] *Entropy, skewness, kurtosis and centroid of the relative energies of 10 equal bands of the envelope of current signal.*

[14] *Discrete wavelet coefficients, mother wavelet: Daubechies 3. 8 levels.*

[15] *Graphical features from the V-I trajectory: asymmetry, 3 area features, slope of middle segment, self-intersection, looping direction and mean line curvature.*

[16] *Graphical features from the V-I_{NA} trajectory where I_{NA} is the non-active current defined per Fryze theory as the orthogonal potion of the current to the source voltage.*

4.2.2 NEFSA Efficiency Assessment

The desired properties of NEFSA are:

- To rank features according to its importance (entropy values), regardless of the classification mechanism that follows.
- To determine the lowest possible number of features needed to obtain the best possible performance (as compared to well-known algorithms).
- To be feasible for NILM embedded- systems, i.e. low computational load.

so in our assessment we will compare NEFSA performance with the performance of sequential forward selection method (SFS) [116] and its improved version: the sequential forward floating selection (SFFS) [117] (more information in appendix 3). SFFS is generally accepted to deliver the best performance in several applications. we start the discussion by presenting Figure 4.8.

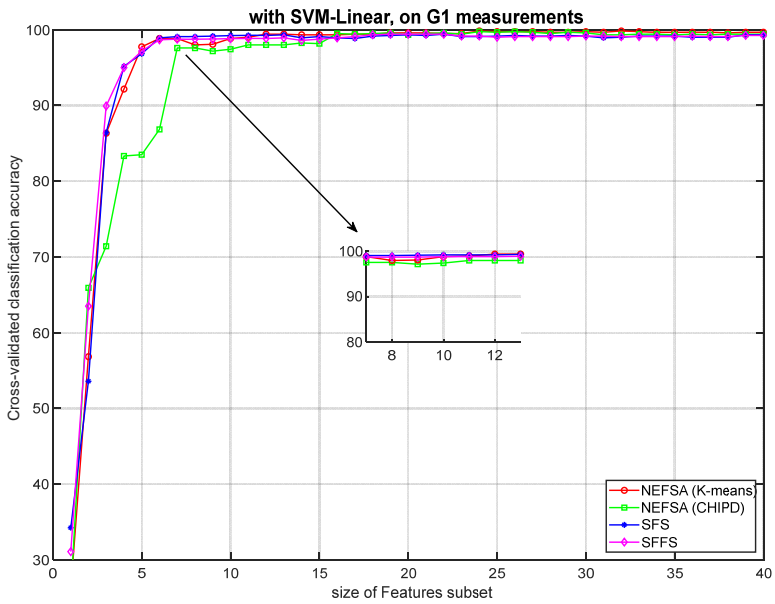


Figure 4.8: Classification Rate as a function of subset size with all algorithms

The performance of the SVM classifier (percentage of correct classification) is plotted vs. the size of features subset (the number of features fed to the classifier for

each value). one can notice that after a size of 7 features, the classification rate improvements are only marginal, to get a better understanding, we built on this observation and tested the goodness of NEFSA by the following experiment:

- The size of selected subset is fixed to 7 features for all algorithms (NEFSA in both versions, SFS and SFFS) .
- A 10000 subsets of 7 are randomly selected features and fed to 4 different classifiers: SVM with linear and gaussian kernels, linear discriminant analysis classifier (LDA) and k nearest neighbor (k-NN).

In Figure 4.9 the performance of a linear-kernel SVM classifiers is shown with chosen subsets of NEFSA-Kmeans (red), NEFSA-CHIPD (pink), SFS (black), SFFS (green) and all randomly-drawn subsets (blue), the left plot shows how many subsets can deliver a certain classification rate, for example in 493 subsets (ca. 5% of all subsets) delivered an 80% accuracy, the result using the same fixed size of features subset for other classifiers is shown in Figure 4.10.

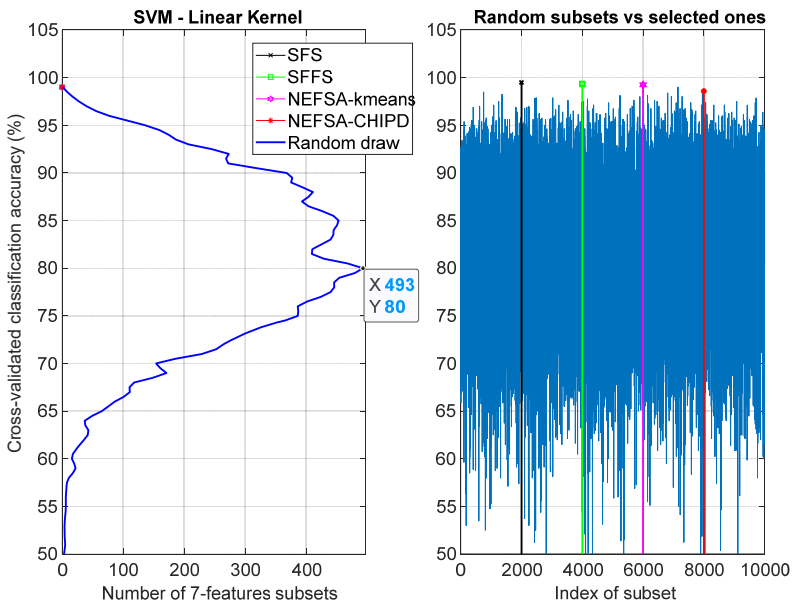


Figure 4.9: Performance of all feature selectors against randomly drawn

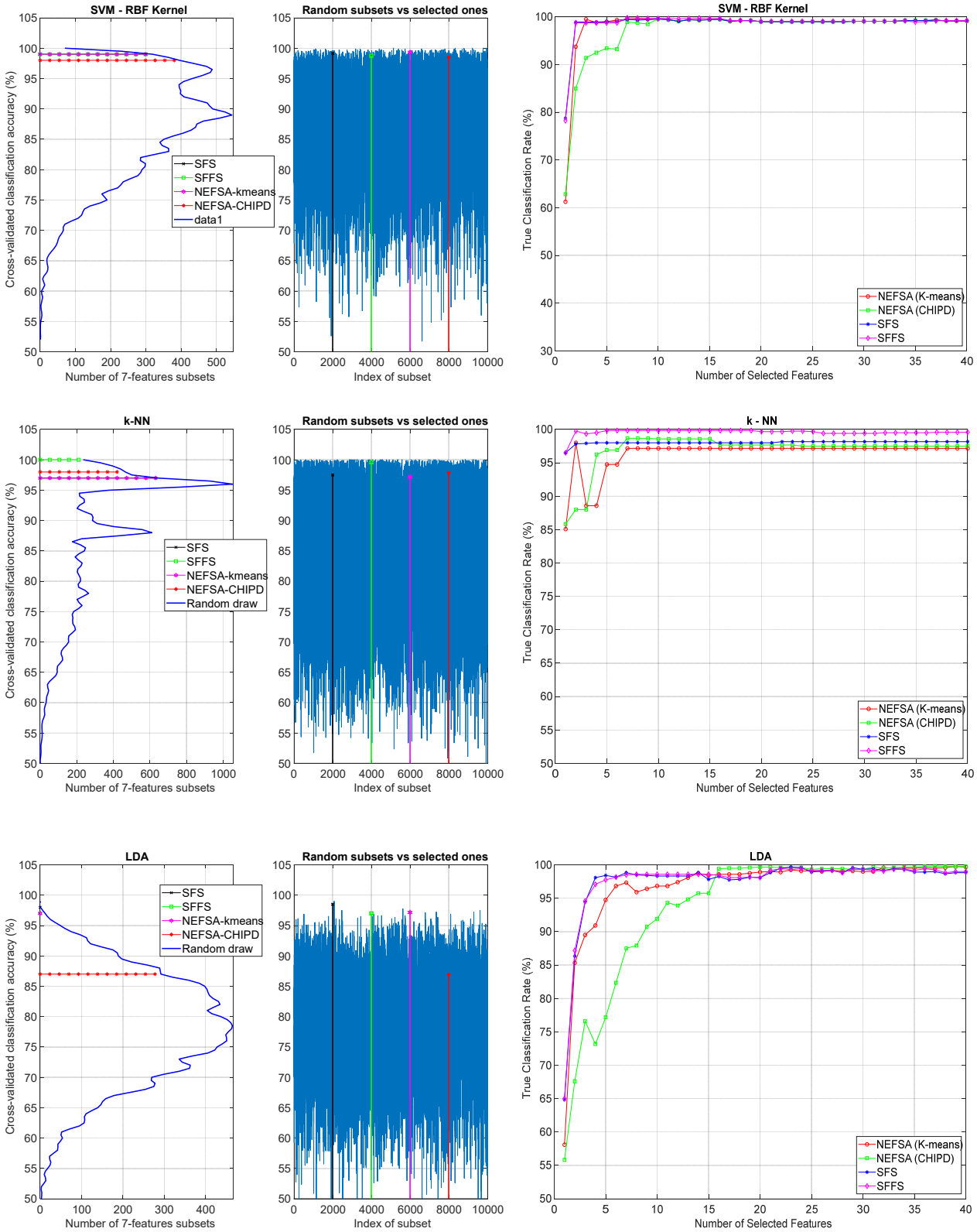


Figure 4.10: Several comparisons of NEFSA performance with various classifiers

The subsets of chosen features by NEFSA, SFS and SFFS algorithms are different, as listed in Table 4.4, yet there is no contradiction; firstly, the best subset of features is not unique, i.e. there can be other subset that deliver the same performance as we saw in Figure 4.10, secondly, both SFFS and SFS suffer the problem of nesting[105], i.e. the best subset of size n (features) is not necessarily included in the best subset of n+1 (for example the best 8 features subset for SFFS/ G1/SVM-linear: 6,5,73,70,74,30,27,33).

SFS and SFFS are "wrapper based" which means they calculate the classification rate for a given feature rate in the process of assessment. On the other hand, NEFSA does not depend on the classifier type, such approaches (called "filter based") are always considered inferior towards the wrapper based in terms of performance, yet we argue that NEFSA has comparable results with those of SFS and SFFS, as presented for G1 in Figure 4.9 and Figure 4.10.

Fixing the size of feature subset to 7 in Figure 4.10 was based on the observation of SVM results in Figure 4.8, the results for each classifier can be improved if the subset size was based on the performance of that particular classifier.

In general, the pre-selection of the subset size (number of features) is a common practice for most feature selection algorithms (including SFS and SFFS), several methods were developed over the past decades to estimate the optimal dimensionality of the feature set (see [118]–[120] and references within), but since the computational cost is an important factor for our specific NILM application, it will suffice to do a simple (binary) search operation for the smallest subset with the best possible performance, taking into account that our case is far from the boundaries of which the "curse of dimensionality" appears, as we stick to feature

sets 10 times smaller (or less) than the number of observations (see [121] for more information).

Table 4.4: The 7 selected features for G1, ranked

	SVM-linear	SVM-RBF	LDA	KNN
SFS	66-6-11-76-71-70-27	4-2-1-3-5-7-8	4-73-36-8-9-1-2	4-76-2-23-1-3-5
SFFS	66-6-37-71-73-76-1	4-2-8-1-3-5-6	4-73-36-8-9-1-2	76-2-23-1-3-5-7
NEFSA(K-means)	23-37-6-73-76-70-4			
NEFSA (CHIPD)	23-73-74-76-79-70-37			

We will not emphasise the importance of the features given in this table, as they are suitable for the experimental setup (V-I trajectory features were ranked among the best here, yet they are known to perform poorly on loads with similar front end power supply [114]).

let us move to numerical tests of NEFSA on real house measurements (BLUED), then discuss the advantage of NEFSA in reducing the computational cost, the overlapping effect on entropy values and on the resulting features subset in general.

4.2.3 Testing NEFSA on real home measurements

Both versions of NEFSA along with SFFS algorithm were applied on BLUED [15] dataset (BLUED measurements are divided to two phases A and B with a total of 43 appliances). The same bank of features (in Table 4.2 and Table 4.3) was calculated and the features sets were always fixed after training on 50% of the data while testing on the other 50%, the imbalance in the number of events per appliance is

high like most of real datasets, so the classification rate is always bounded, solutions for such cases are proposed in the next chapter. In Figure 4.11 the classification rates of an SVM classifier (linear kernel) fed with increasing size of features subsets from all three lists of ordered features are illustrated: first one is the result of applying SFFS (blue) and the other two are the results of NEFSA (both versions).

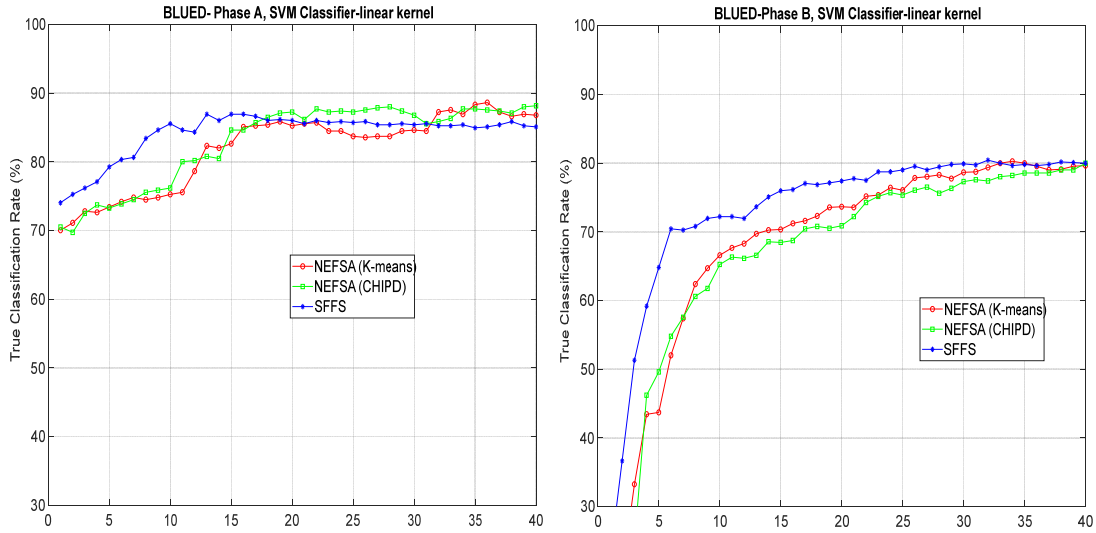


Figure 4.11: Evaluation of NEFSA ranking on BLUED

Although the SFFS algorithm performed better on smaller sets of features (less than 15), yet the evaluations on both phases show an equivalent performance when considering the best possible classification rate (89% for phase A, 80% for phase B).

4.2.4 Computational cost

The second aspect for comparison is the computational cost of NEFSA compared to the other algorithms: SFFS, SFS. A computer equipped with an Intel i7 6 core 2.2 GHz processor is employed for all tests, and all programs were written in Matlab [122].

Since both SFS and SFFS depend on misclassification rate as its figure of merit (evaluation criterion for choosing one subset over another one), their execution will depend heavily on the number of classification operations needed till convergence. NEFSA on the other hand calculates the entropies once, clustering can take tens of seconds if done with K-means, or a lot less if CHIPD is employed. The execution times on all 3 groups of measurements from HELD1 are measured and bar-plotted against a logarithmic time scale in Figure 4.12 as the difference is enormous (numbers of seconds are printed over each bar for convenience).

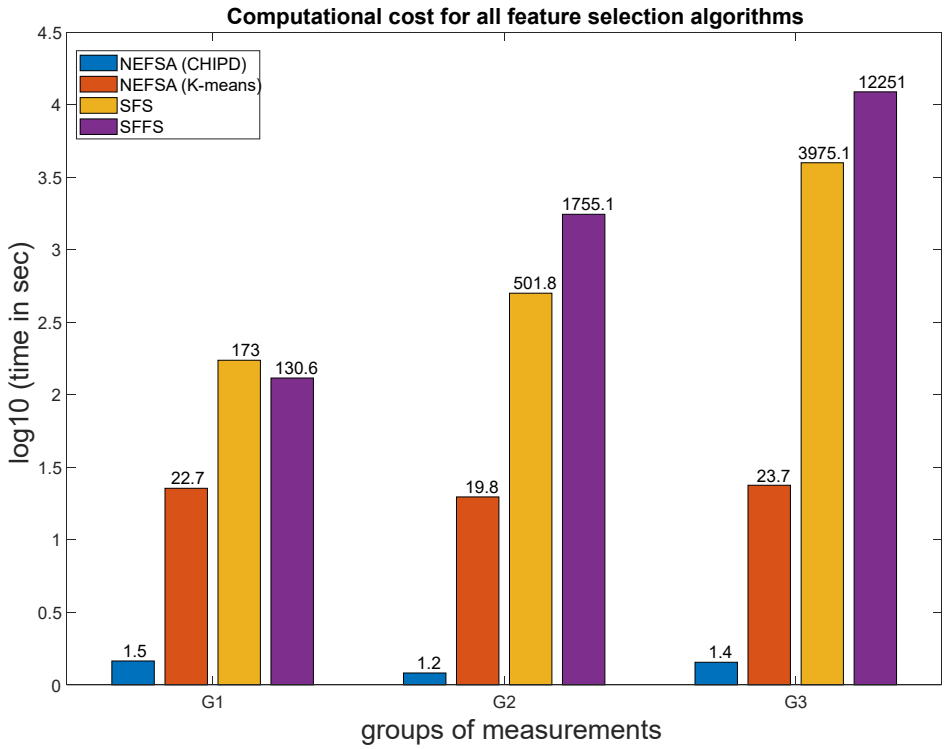


Figure 4.12: time-efficiency of NEFSA vs. SFFS and SFS

For the third group of measurements, NEFSA (with CHIPD clustering) is 10^4 faster than SFFS algorithm. The ranked list of features is available to classifiers within 1.4 seconds.

4.3 Discussion and closing notes

Most of the features in the collected bank were developed and extracted in the hope of enabling the best disaggregation possible, we argue that feature selection is essential as changing the configuration of appliances (adding or omitting appliances) will change the order of the best features and considerably affect the classification accuracy.

Both versions of NEFSA were applied on a subset of the first group G1 of appliances (Table 4.1); where measurements contain single measurements (i.e. no overlapping) with equal number of events (80 ON/80 OFF per appliance) for all devices. Figure 4.13 demonstrates the resulted values of Entropy on the vertical axis, while the horizontal axis represents the indices of the all tested features.

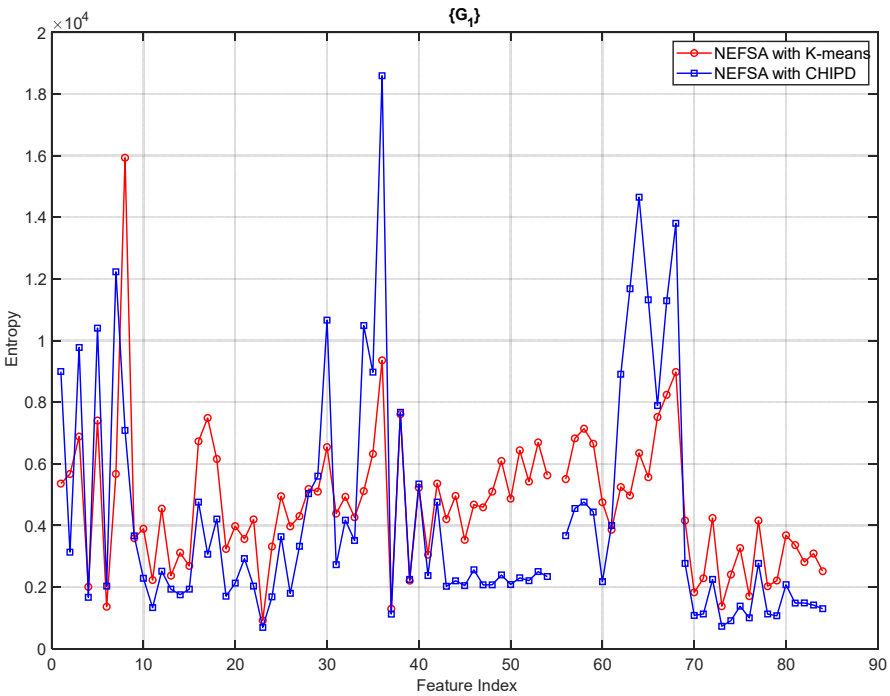


Figure 4.13: Entropy evaluation of all features on G1

Ideally, the best feature is the one with the lowest Entropy value, so we are more interested in the lower part of the plot, both versions of NEFSA (with K-means and with CHIPD) gave approximately the same set of 20 best features, yet there are differences in the general order, this can be partially attributed the initial assigning of centroids with K-means version; one should also keep in mind that both versions delivered identical performance in our tests with 3 classifiers. The Insight here is that all tested feature selection algorithms (SFS, SFFS, NEFSA-CHIPD, NEFSA-K-means) are ranking with different methodologies but they result in reliable ordering. Also, to study the effect of overlapping on Features-ordering we add G2 to G1, then we take all 3 groups together in Figure 4.14.

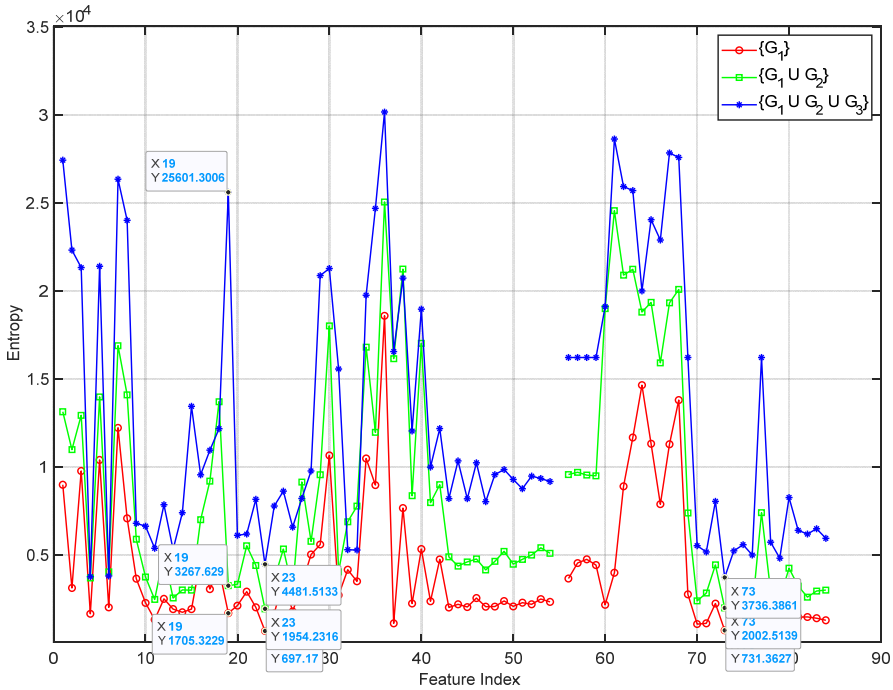


Figure 4.14: NEFSA Feature ordering with 3 overlapping degrees

while adding G2 to G1 will double the number of events (data points), yet the entropy differences are not uniformly the same and it is not only doubled after

adding G2 (or tripled with G3), which indicates the effects of overlapping, it is also noted that some of the features keep their importance as Feature 23 (shown with a data tip in Figure 4.14, the feature is SSMR: highest spectral peak over spectral mean [102]) while others are affected greatly with increased overlapping as Feature 19 (PMR: highest temporal peak over temporal mean).

An important insight can be drawn from this result: one “best” set of features is not enough if the environment will be drastically changing its composition, i.e. factories with shifts for different production lines); such environments will be better monitored with a smart meter that can adapt its set of features or equipped with ready-to-use list of features for the targeted configurations.

On the other hand, for residential environments, best attainable load disaggregation depends on adequate training on the typical routines of power usage and calibrating the smart meter on regular basis is also important to account for new appliances. As can be seen from, the features that stayed in the best 25 features are color-coded in three cases: NEFSA with both versions and SFFS/SVM-linear, We observe an advantage for NEFSA (both versions) as its features-subset will not change greatly with overlapping compared to SFFS for example (19/25 for NEFSA-CHIPD,20/20 for NEFSA-K-means compared to only 7/25 for SFFS/SVM-linear).

Table 4.5: Intersections of best subsets with increased overlapping

	G#	First 25 Features, Ranked
NEFSA CHIPD	G1	23,73,74,76,79,70,37,78,71,84,11,75,83,81,82,4,24,19,14,26,15,13,43,6,22
	G1 & G2	76,23,73,74,70,11,79,13,82,71,78,83,15,84,75,14,26,81,19,20,24,4,10,6,31
	All	73,4,6,23,79,76,71,74,33,32,13,11,70,75,78,84,20,82,21,81,83,26,10,9,14
NEFSA K-means	G1	23,37,6,73,76,70,4,78,39,79,11,71,13,74,84,15,82,41,83,14,19,75,24,81,45
	G1 & G2	23,4,76,6,78,73,70,71,21,41,11,39,74,82,9,79,13,84,75,83,45,15,19,10,81
	All	6,4,76,23,11,39,83,41,84,45,74,73,75,71,9,79,70,43,34,13,78,82,81,47,10
SFFS with SVM_linear	G1	66,6,37,71,73,76,1,2,3,4,5,7,8,9,10,11,12,13,14,15,16,17,18,19,20
	G1 & G2	6,5,73,70,74,30,27,33,78,1,79,32,10,3,2,7,11,4,9,15,8,16,12,23,20
	All	73,2,4,11,70,81,37,75,84,66,16,74,12,60,54,49,17,7,30,55,67,3,1,5,35

5 Scenario-Based NILM Signals Construction

The majority of NILM Datasets, which were made publicly available in the last two decades, had been collected with a specific application in mind; they were still not optimal for extracting important features, such as harmonic properties in transient states of the current signal [123]. They were either limited to aggregate measurements only or single measurements only, or lacking enough sampling rate in one of them, as provided in Table 5.1.

Nowadays, Machine learning algorithms along with advanced signal processing techniques are ready to exploit such features to deliver the best possible identification of electrical appliances, so all we need is other “extensive” data sources or a smart way to construct “quasi-equivalent” sources out of limited measurements at hand. The first question we raised: if only single-device measurements are available, could we use them for 'synthesizing' whole measurements without measuring again? it will be sufficient for the producers to provide interested customers with a list of stand-alone measurements of their device,

a sort of "extended digital datasheet"; enabling them to test many scenarios of consumption before putting any constellation of devices together, which is of a high importance for big factories and sensitive facilities. A more preferable second solution would be to rely on few non-intrusive aggregate measurements to synthesis these desired NILM scenarios.

In this chapter, we will start by presenting a procedure to generate an approximately identical version of the aggregate real measurements using single-device measurements alone as an answer to the first question. Then we develop a generalized version of it to be able to use any aggregate measurements to construct any desired NILM signal.

Table 5.1: Public Datasets with measurement types

Dataset	Type of measurements	References
REDD	whole-house measurements only	[124]
BLUED	whole-house measurements only	[125]
PLAID	individual measurements only	[56]
UKDALE	whole-house measurements, single measurements with low sampling rate	[126]
WHITED	individual measurements only	[127]
COOLL	individual measurements only	[63]

5.1 NILM Signal synthesis out of single-device measurements

5.1.1 Intuition and phase preservation

The intuition to construct aggregate measurements out of single-device measurements was always encountered with the problem of phase difference among different measurements for different devices. In the following we present a simple example to clarify this problem:

The general model for the reconstructed current signal I_t can be written as:

$$I_t = \sum_{i=1}^{N} \delta_i I_i + v \tag{5.1}$$

Where N is the number of the appliances.

I_i , $i = 1, \dots, N$: are the current signals of the individual appliances.

$\boldsymbol{v}$ is a noise term. δ_{i} , $\mathrm{i} = 1, \cdots, N$ indicates the state of the corresponding appliance as:

$$\delta_i = \begin{cases} 1, & if\ the\ i^{th}\ appliance\ is\ 'ON' \\ 0, & if\ the\ i^{th}\ appliance\ is\ 'OFF' \end{cases} \tag{5.2}$$

Now let us look at the case of two appliances (*N=2*) where they are represented as complex loads resulting in a phase shift between the voltage and the current signals; their currents can be written as:

$$I_1 = A_1 e^{j\theta_1} \quad , \quad I_2 = A_2 e^{j\theta_2} \tag{5.3}$$

The total current signal can be then obtained as ($A_1 = A_2 = A$ for simplicity):

$$I_t = 2A\, e^{j\left(\frac{\theta_1+\theta_2}{2}\right)} cos\left(\frac{\theta_1-\theta_2}{2}\right) \tag{5.4}$$

$$\Rightarrow |I_t| = 2A\left|cos\left(\frac{\theta_1-\theta_2}{2}\right)\right| \quad (5.5)$$

The amplitude of the total signal $|\boldsymbol{I_t}|$ depends on the phase difference $\theta_1 - \theta_2$, so a simple addition as in relation (5.1) will not be correct unless all current signals are added with emphasis on their relative phase.

Fortunately, the voltage signal stays intact and can be considered a reliable phase reference, so current signals of the individual appliances are converted to FIT-PS [100], [128] representation, which takes the zero-crossing point (rising edge: from negative to positive) in the voltage signal as a phase reference. In the following, we build on this observation, firstly let us present a flow chart for the proposed procedure in Figure 5.1 below:

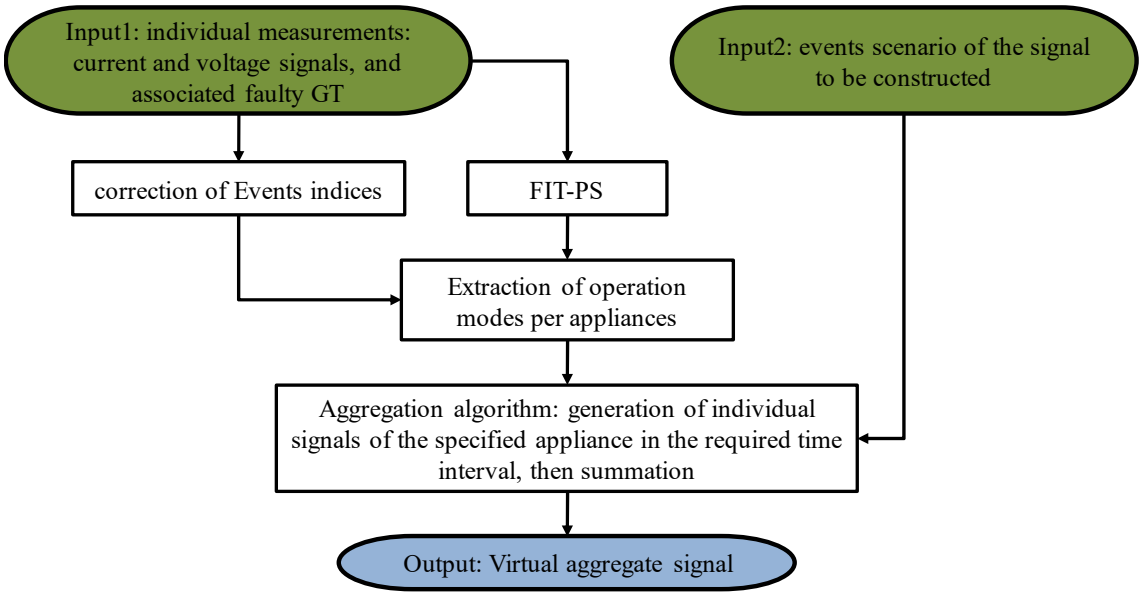


Figure 5.1: NILM signals construction out of single-device measurements

Five distinctive steps are followed in this procedure, let us list them then explain new ones:

[step 1] Convert single measurements to "Frequency Invariant Transformation of Periodic Signals" representation (FIT-PS) [100].

[step 2] Correct event indices (using our proposed method for transient state start detection in 3.4.3.2).

[step 3] Extract operational modes for each appliance.

[step 4] Generate the targeted current signals according to the desired scenario, adding steady-state periods to meet the desired length, as detailed in Figure 5.2.

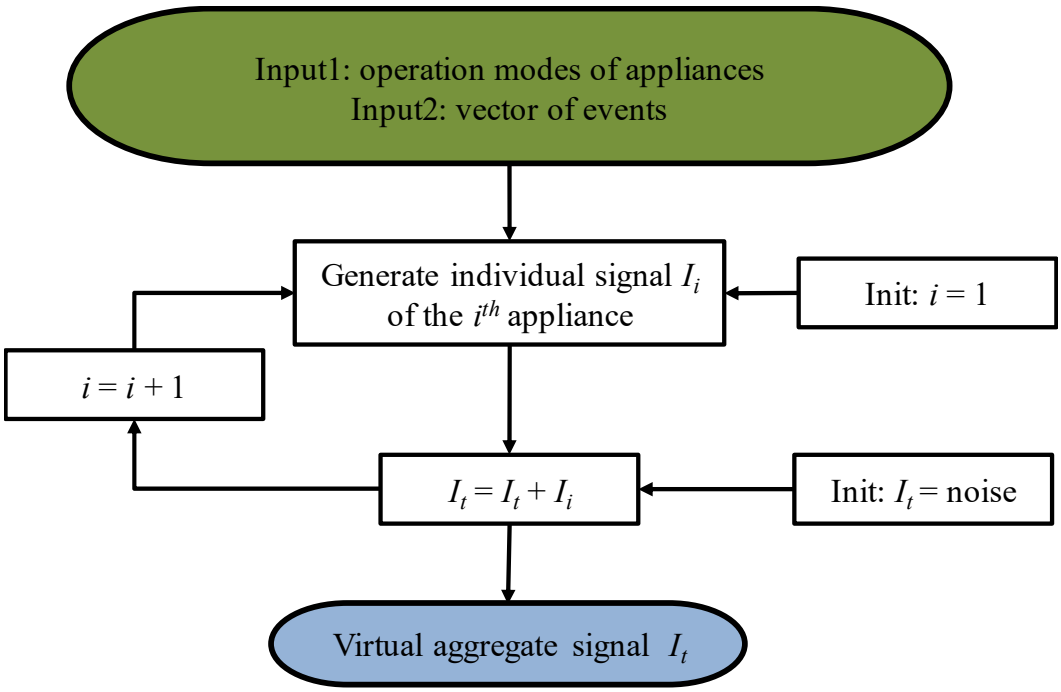


Figure 5.2: Aggregation algorithm

[step 5] Sum the generated current signals according to the desired scenario.

We will clarify the concept of operational modes and their extraction in the next sections.

5.1.2 Operational modes

Some appliances have more than one operational mode in general, where we define an operational mode as a distinctive subset of the current signal of a certain device; that can comprise both transient state and part of a steady state. An example is

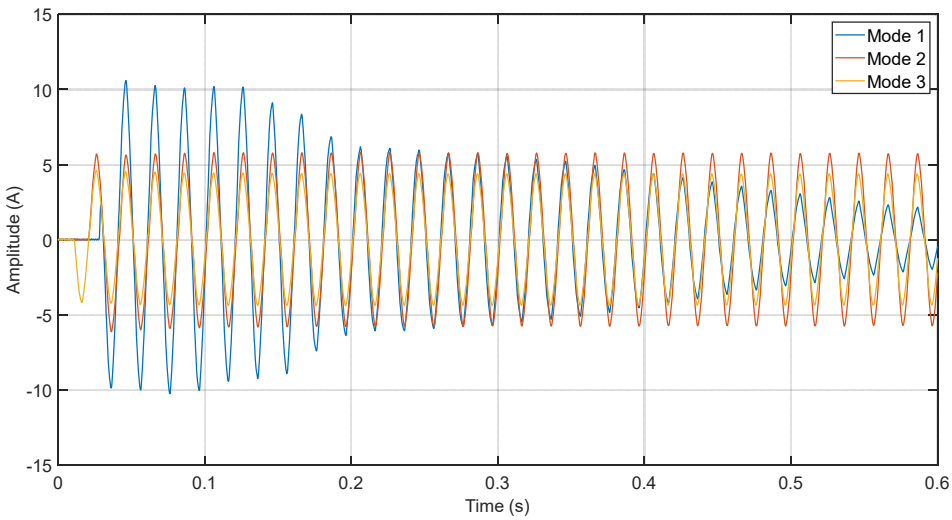


Figure 5.3: Operational modes of a fridge

depicted in Figure 5.3 for a fridge.

As can be seen, these operational modes differ from each other and must be considered when constructing the virtual aggregate signal. Each operational mode has a different power level in its transient or steady state or both, which can be used to choose the most appropriate one in the construction procedure. Simple structure appliances have only one operational mode in general, e.g., a lamp.

Extraction of operational modes from single-device measurements can be done as follows:

- For each event, take a window starting at the event index and containing both transient state and a part of the steady state.

- Compare the power level in the steady state of that window with the power levels of the previously extracted operational modes of the appliance; it will be added as a new mode if it differs from those previously detected.
- a new sequence number is assigned to the new mode and will be checked again in other occurrences (as the current signal drawn by some appliances will change its operational modes in a sequential manner).

5.1.3 Numerical comparisons

Numerical experiments were conducted to test the hypothesis of "suitability of the constructed aggregate signal for training and testing NILM algorithms".

To do that, the performance of standard NILM algorithms on such generated data will be compared against their performance on an original aggregate signal that has the same scenario of events and appliances. Figure 5.4 shows a classical chain of processes for typical NILM system.

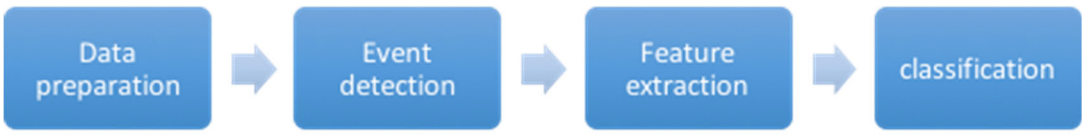


Figure 5.4: Typical sequence of processes in a NILM system

The comparison will be conducted between the results of standard event detection, feature extraction and classification algorithms on the original aggregate measurements as well as the constructed aggregate measurements. For that purpose, the set of single measurements in HELD1 dataset are employed along with aggregate measurements for the same set of appliances in Table 5.2.

The real measurements were conducted as follows:

- Single measurements for each appliance were taken with 100 pairs of On-Off events, each active cycle was kept running for 15-25 seconds to get the transient state and part of the steady state.
- Aggregate measurements, where each appliance was switched On and Off randomly, but only 4 appliances can be active on the same time at most.
- A minimum temporal distance of 3 seconds was kept between events.

Table 5.2: List of considered Appliances

ID	Name	Nominal Power (W)
1	Radio	6.2
2	Heat gun (setting 1)	820
3	Router	9.2
4	Black desk lamp	20
5	Light bulb box	20
6	Kettle	2100
7	Black hairdryer (setting 1)	500
8	Fan	22
9	Rotary tool (Dremel)	30
10	LED lamp	1

The virtual aggregate measurements were generated according to the same scenarios of events followed in real ones, an example of those measurements is depicted in Figure 5.5, where both constructed and original signals are plotted together in the upper side, while the other 2 plots are magnified regions, one where the signals were identical (middle), and the other one where the MSE was at its highest. in the

coming paragraphs we propose a hybrid method for measuring the similarity of two signals in this particular application.

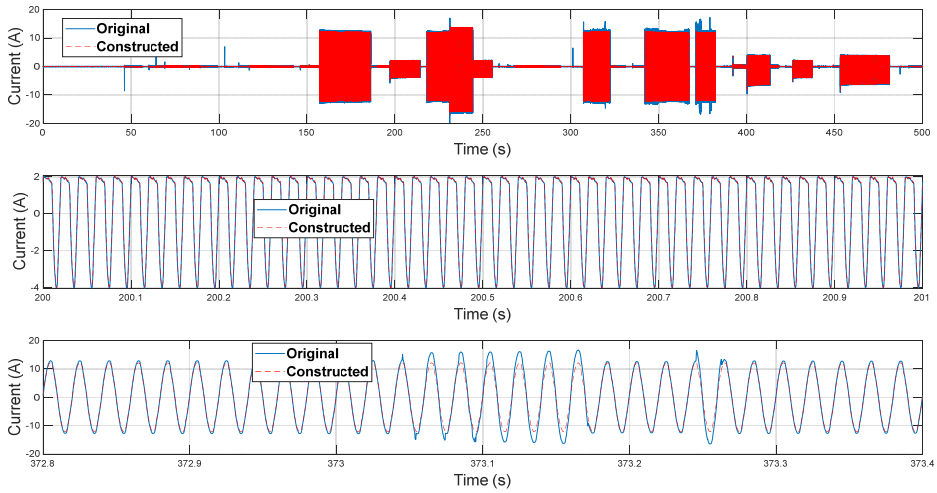


Figure 5.5: Constructed current signal vs Original one, magnifying 2 regions

5.1.3.1 Event Detection

Two different event detectors were used, the first one is based on the work of Hart [6] , while the second is our FBE Event detector, as designed in third chapter. Both were applied on the original measurements as well as the constructed ones, and the results are given in Table 5.3 and Table 5.4 respectively. The threshold for Hart Event detector was set to 15 W, as going below that will result in detecting too many false events due to the high variations of the noise (1 W is also presented in the Table 5.3 for demonstration), appliances below the chosen threshold were treated as noise in all event-detection tests since the comparison is the main goal here.

Table 5.3: Comparison of Event detection results, Hart

Signal Type	Thr (W)	Total Event	TP	FN	FP	P_{recall} %	$P_{precision}$ %
Original	1	1600	1411	189	3417	88.19	29.23
Constructed	1	1600	1448	152	5542	90.50	20.72
Original	15	1120	958	162	31	85.54	96.87
Constructed	15	1120	959	161	0	85.62	100

For FBE detector also, several thresholds were chosen, and the results are presented in Table 5.4.

Table 5.4: Comparisons of Event Detection results, FBE

Signal Type	Thr (W)	Total Event	TP	FN	FP	Precall %	Pprecision %
Original	1	1600	1439	161	135	89.94	91.42
Constructed	1	1600	1480	120	97	92.50	93.85
Original	5	1440	1285	155	53	89.24	96.04
Constructed	5	1440	1322	118	7	91.81	99.47
Original	15	1120	1113	7	37	99.38	96.78
Constructed	15	1120	1090	26	0	97.32	100

Looking at the tables, both event detectors show a slightly better performance on the constructed measurements (less than 5%), a more realistic solution with modelling the steady state variations will be presented in the generalised construction model.

5.1.3.2 Comparing extracted features

A full comparison of all extracted features is a tedious task and more meaningful conclusions can be drawn from the performance of a certain classifier fed with all or an identical subset of NILM features extracted from both sides, nonetheless, let us take a glance at some of the steady state features for high and low consumers. Active and reactive powers P and Q were chosen, along with the mean power of the harmonics. If multiple appliances are active at the same time, a subtraction procedure is done at each event to extract a steady state window suitable for

calculating those features of interest (while taking the phase and the transient state length into account, assuming that events are well separated which is the case for these measurements). A band-pass filter is applied on the current signal for each harmonic component $f_{c,n} = n * f_0$, where n is the harmonic order, f_0 is the fundamental frequency, then the mean harmonic power is given by:

$$P_n = V_{RMS} * I_{n,RMS} \tag{5.6}$$

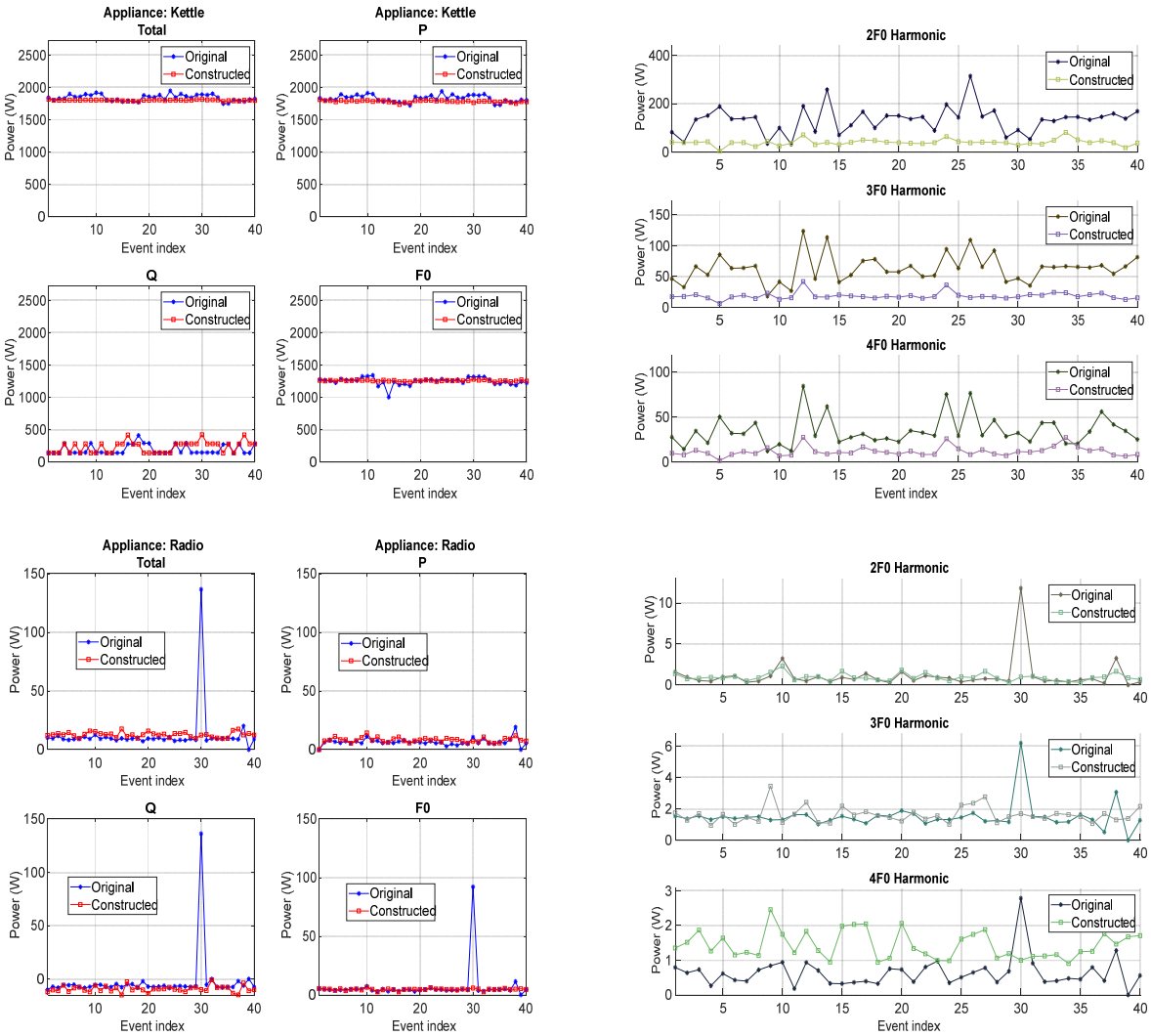


Figure 5.6: comparison of steady state features for a Kettle (top), and a Radio (bottom)

In Figure 5.6, we compare the selected steady-state features: Apparent power P_A, active power P and reactive powers Q for two different appliances, along with P_2, P_3, P_4 as in equation (5.6). The kettle which consumes high power (ca. 2 kW), and a Radio that consumes 6 W only, where in each sub-plot, the horizontal axis is for event indices (took 40 events) while the vertical is for the power.

This "visual" comparison is to get a feeling for the values, while constructed signals mostly show a homogenous behaviour, original real signal has few outliers for some events, which can be attributed to the sensitive Sine function for reactive power Q calculation (small deviation in phase difference has a big impact). Let us see how these "close results" can be reflected in classification results in the next section.

5.1.3.3 Comparing classification results

Two standard classifiers [129] were implemented to compare their performance on both types of measurements, the first one is based on a feed-forward neural network (FFNN), while the other one is a support vector machine (SVM) classifier (various kernels). Both classifiers were trained using extracted features from individual measurements (active and reactive powers in addition to the harmonics), then tested on the original and constructed virtual measurements respectively. To simplify the classification process, a power threshold of $25W$ was applied to divide the appliances into two groups: high power appliances and low power appliances; the features are then fed to the corresponding FFNN subnet (or SVM sub-classifier respectively), comparisons between the performance of those classifiers are listed in Table 5.5 and Table 5.7.

Table 5.5: FFNN performance comparison

%	High power appliances	Low power appliances
Original	96.9	87.9
Constructed	100	89.6

Starting with the FFNN in Table 5.5: while performance on both sides is comparatively similar, the classifier delivered slightly better results when applied on constructed measurements; one reason for that is the mismatch between the possible variations in steady state and the few periods being constantly repeated, this will addressed in the general model. also, the the added noise to the constructed signal is a simple additive white noise which is rarely the case for real ones. Another reason is that steady-state features from real measurements are more prone to variations and outliers. For a deeper analysis per appliance one can check the confusion matrices, as in Table 5.6.

FFNN correctly classified nearly all the samples in the test set on both types of measurements in the high-power category. Although the overall accuracy rate is close in low power category, yet a closer look yields differences in misclassification rate among the individual appliances, for example: The Radio (ID=1) was misclassified as Router only once in the original measurements table, while it was misclassified 8 times as Router in the constructed measurements table. This can be attributed to the higher noise level in real measurement which can be of great impact for low-power consumers (as both appliances have a less than 10 W nominal power).

Table 5.6: Side by side comparison of confusion matrices, Original vs

Original Signal Classification, Low-power consumers

Output classes: appliances IDs \ Target classes: appliances IDs	1	3	4	5	8	10	Total
1	67 14.0%	2 0.4%	0 0.0%	1 0.2%	0 0.0%	3 0.6%	91.8% 8.2%
3	1 0.2%	70 14.6%	0 0.0%	1 0.2%	1 0.2%	5 1.0%	89.7% 10.3%
4	0 0.0%	1 0.2%	73 15.2%	0 0.0%	3 0.6%	1 0.2%	93.6% 6.4%
5	5 1.0%	2 0.4%	1 0.2%	68 14.2%	0 0.0%	1 0.2%	88.3% 11.7%
8	5 1.0%	4 0.8%	4 0.8%	0 0.0%	76 15.8%	2 0.4%	83.5% 16.5%
10	2 0.4%	1 0.2%	2 0.4%	10 2.1%	0 0.0%	68 14.2%	81.9% 18.1%
Total	83.8% 16.2%	87.5% 12.5%	91.3% 8.8%	85.0% 15.0%	95.0% 5.0%	85.0% 15.0%	**87.9% 12.1%**

Constructed Signal Classification, Low-power consumers

Output classes: appliances IDs \ Target classes: appliances IDs	1	3	4	5	8	10	Total
1	58 12.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.4%	96.7% 3.3%
3	8 1.7%	69 14.4%	1 0.2%	2 0.4%	0 0.0%	0 0.0%	86.3% 13.7%
4	0 0.0%	0 0.0%	75 15.6%	6 1.3%	0 0.0%	0 0.0%	92.6% 7.4%
5	3 0.6%	0 0.0%	2 0.4%	71 14.8%	0 0.0%	1 0.2%	92.2% 7.8%
8	0 0.0%	11 2.3%	2 0.4%	0 0.0%	80 16.7%	0 0.0%	86.0% 14.0%
10	11 2.3%	0 0.0%	0 0.0%	1 0.2%	0 0.0%	77 16.0%	86.5% 13.5%
Total	72.5% 27.5%	86.3% 13.7%	93.8% 6.3%	88.8% 11.3%	100% 0.0%	96.3% 3.7%	**89.6% 10.4%**

Original Signal Classification, High-power consumers

Output classes: appliances IDs \ Target classes: appliances IDs	2	6	7	9	Total
2	78 24.4%	0 0.0%	4 1.3%	0 0.0%	95.1% 4.9%
6	0 0.0%	77 24.1%	0 0.0%	0 0.0%	100% 0.0%
7	1 0.3%	2 0.6%	75 23.4%	0 0.0%	96.2% 3.8%
9	1 0.3%	1 0.3%	1 0.3%	80 25.0%	96.4% 3.6%
Total	97.5% 2.5%	96.3% 3.7%	93.8% 6.3%	100% 0.0%	**96.9% 3.1%**

Constructed Signal Classification, High-power consumers

Output classes: appliances IDs \ Target classes: appliances IDs	2	6	7	9	Total
2	80 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
6	0 0.0%	80 25.0%	0 0.0%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	80 25.0%	0 0.0%	100% 0.0%
9	0 0.0%	0 0.0%	0 0.0%	80 25.0%	100% 0.0%
Total	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	**100% 0.0%**

same approach is valid for the results of SVM performance in Table 5.7, the differences in classification rates are relatively small (worst case encountered is a difference of 8.4%) and can be attributed for the most part to misclassifications of small-power consumers due to unrealistic noise model, other improvements can be introduced to operational modes extraction and selection, next section will discuss the possibility of generalizing the construction concept to aggregate-to-aggregate

NILM signals with proposed modelling to steady state variations, then some closing insights and notes are presented.

Table 5.7: SVM performance comparison (3 kernels)

	High power appliances			Low power appliances		
%	RBF	Linear	Polynomial	RBF	Linear	Polynomial
Original	91.3	90.6	96.3	85.4	81.9	86.3
Constructed	99.4	82.2	100	90.6	78.8	90.6

5.2 NILM Signal synthesis out of aggregate measurements

The next milestone in NILM signal synthesis research is: How feasible is it to use limited aggregate NILM signals only (instead of single-device measurements) to generate any desired NILM scenario. A systematic method to answer this question would be to find out if it were possible to harvest the operational modes of individual appliances from a real aggregate NILM signal and use them to build any NILM signal.

In analogy to the followed approach in the previous section. This section will present a novel method to isolate those operational modes and evaluate them before discussing the possible applications of the method. Figure 5.7 illustrates the general scheme for this approach.

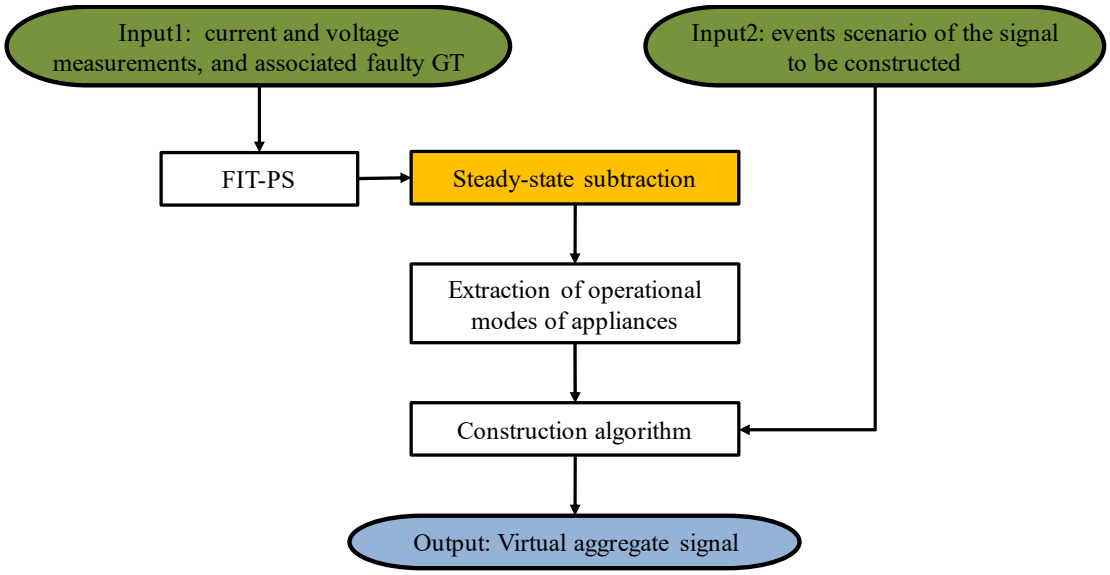


Figure 5.7: NILM signals construction out of aggregate measurements

5.2.1 Steady-state subtraction

The highlighted box in Figure 5.7 (steady-state subtraction) will be clarified first as other steps prior to it are similar to the ones previously discussed in the preceding section.

The objective of this step is to extract, for each event, the individual current signal of the appliance, we eliminate the effect of the other 'on' appliances (if exists) on the total current signal as much as possible, thus keeping only the signal of the appliance under interest. In principle: the other 'on' appliances are in their steady states before the event, and it is assumed that they're still in the same state after it, so we can compose a new signal by taking an integer number of periods from the steady state before the event and then subtracting them from the total signal.

The resulted signal is supposed to be an individual current signal of the appliance that caused the event. This process will be referred to as "SSS" in the following, a flow-chart is presented in Figure 5.8 along with an example of a subtracted signal of

a Dremel, it should be noted that a correction of event index is executed beforehand if needed.

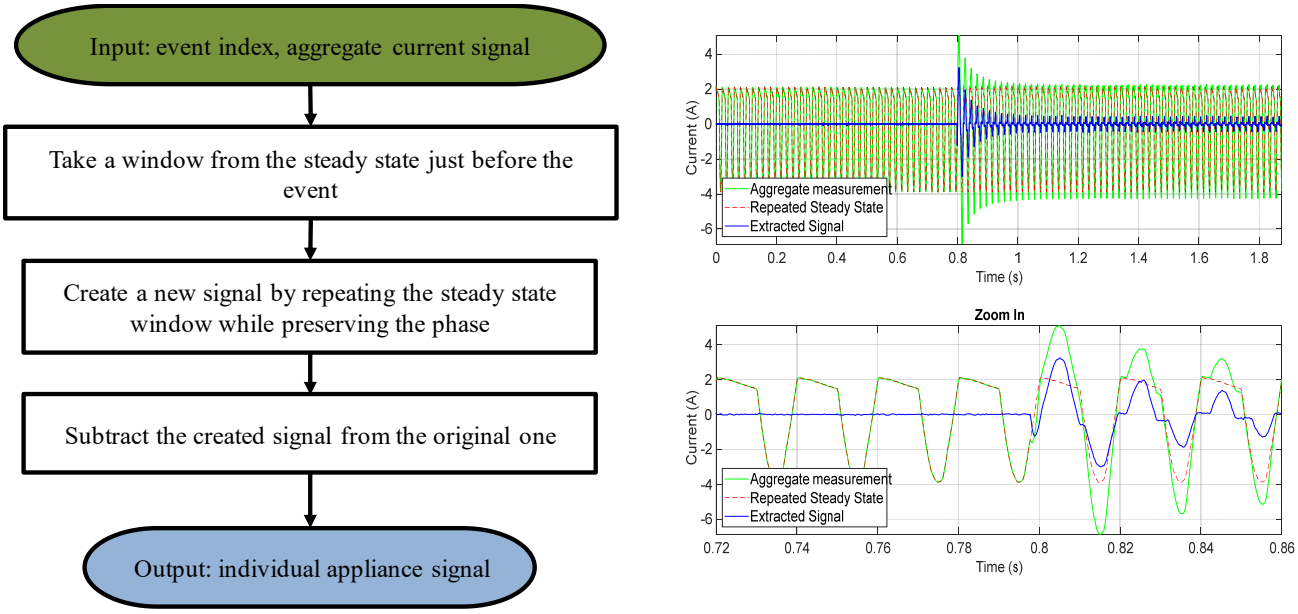


Figure 5.8: Steady-State subtraction (left) Flow-chart (right) example for a Dremel event

5.2.2 Extraction of Operational Modes

The output of SSS method is a set of extracted individual signals for each appliance in a given dataset or aggregate measurement. Some of those signals may be "similar" in their transient and steady states, so they are represented by a single operational mode. To measure similarity between extracted signals we use power change parameter and compare the length of the transient state. Two signals are considered similar if the difference between their power change values is smaller than a predefined threshold (1-3% of the nominal power consumption of the appliance) and their transient state length differ by 3 periods or less ($T = \frac{3}{F_{Grid}}$).

A comparison between the operational modes extracted from both individual and aggregate measurements for two different appliances is given in Figure 5.9, where we note that the number of operational modes extracted from aggregate measurements can be larger than that of individual measurements, depends on the length of the measurement and how much "distortion" did the transient state of that particular appliance suffer due to the presence of other devices, in the steady state they have the same amplitude and waveform.

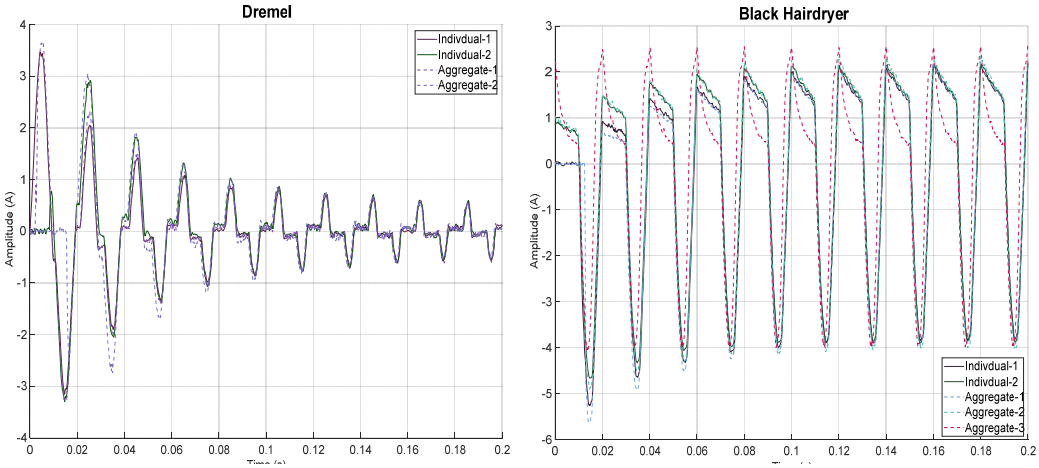


Figure 5.9: comparisons of operational modes (out of individual measurement vs aggregate measurement) for two appliances

5.2.3 Construction Procedure

The construction algorithm is detailed in Figure 5.10, it has three inputs:

The first two inputs are similar to those of last section (Figure 5.2): the set of extracted operational modes for each appliance and the desired events scenario. The third input is the calculated parameters of steady-state variations that describes the ripples in signals envelop.

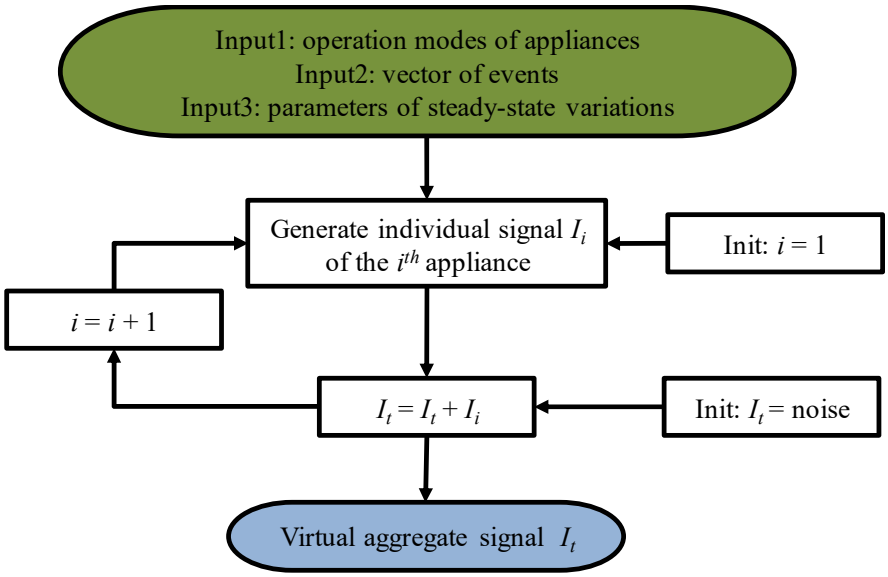


Figure 5.10: Construction algorithm with noise modelling

The construction procedure has two stages:

- **Initialization**: the constructed current signal I_t is initialized with randomly selected samples of a noise signal, preferably taken from the available measurements where all appliances are 'off', assuming that these samples are IID (independent and identically distributed), otherwise an additive white noise is generated.
- **Loop over appliances**: create an individual signal for each appliance based on its events and operational modes. The outputted aggregate signal I_t is the accumulation of all individual signals for all appliances.

In the following, we describe the steady-state variations model then walk through the steps of individual-appliance signal generation.

5.2.3.1 Modelling of steady-state variations

The purpose of NILM signal construction scheme is to be able to generate any desired NILM scenario with realistic characteristics. Careful analysis of the difference signal between the modelled sum of extracted operational modes and the original real aggregate measurement displayed inconsistent properties (spectral peaks at grid frequency and harmonics) of what was assumed white (or at least coloured) noise previously. One way to mitigate this difference is to model the variations of steady-state regions and incorporate this model in the construction procedure. In Figure 5.11, the behaviour of steady-state peaks (envelope) of current signal suggests a little "decay" which can be seen in current and in power signals as well.

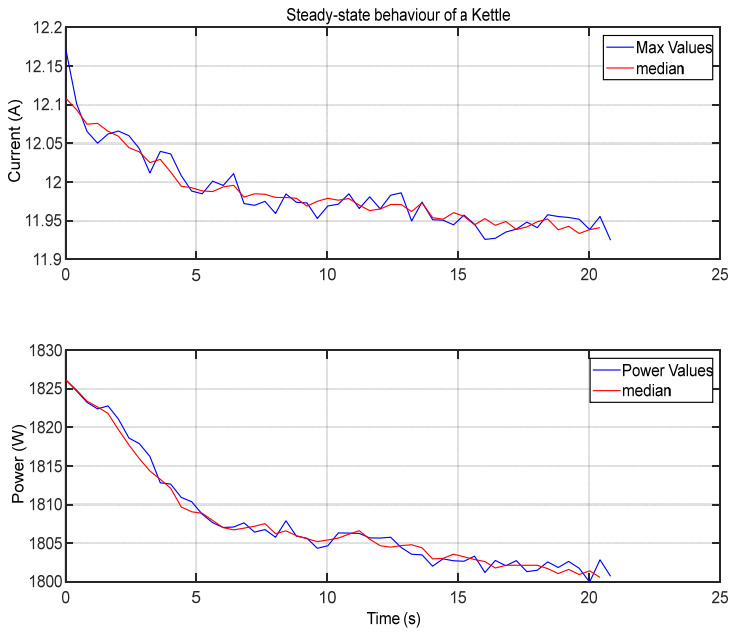


Figure 5.11: Steady-state envelope, Kettle

We propose the following method for modelling those variations:

Let I be a segment of the current signal starting at the end of the transient state.

1. calculate the signal I_p, where $I_p(m)$ is the maximum value in the mth period of I,

$$I_p(m) = \max I((m-1) * N + 1 : m * N) \quad (5.7)$$

N is the number of points per period. The signal I_p is then fitted with a power function of the form $f_1(m) = a \times m^b + c$ as shown in Figure 5.12.

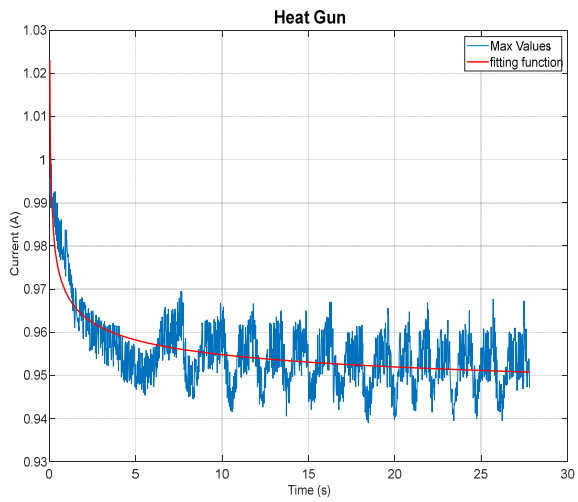


Figure 5.12: steady-state modelling, first fitting process

2. calculate the difference between the two signals as

$$Diff_1(m) = I_p(m) - f_1(m) \quad (5.8)$$

The second fitting operation is applied on the signal $Diff_1$ using sin_5 function defined as : $\sin_5(m) = \sum_{i=1}^{5} a_i \times \sin(b_i \times m + c_i)$ (5.9)

The result of second fitting step is shown in Figure 5.13.

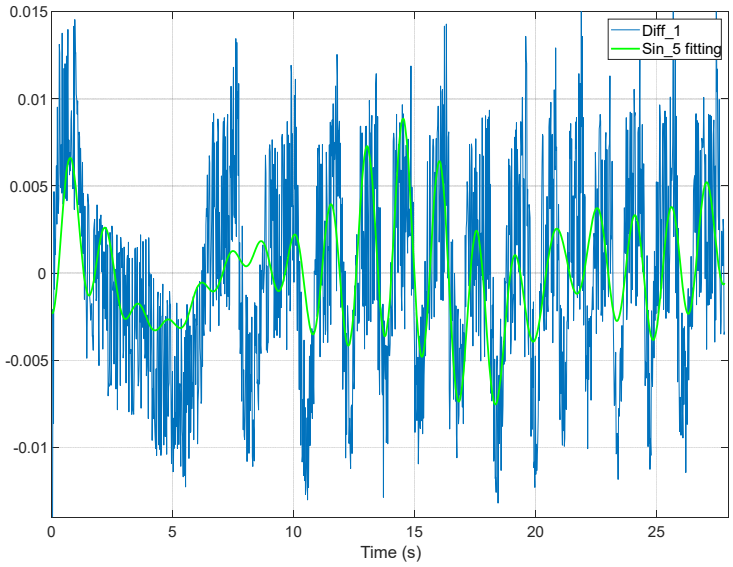


Figure 5.13: steady-state modelling, second fitting process

The difference between the signal $Diff_1$ and its fitted curve is the signal *Nsig*, which is desired to be random with Gaussian distribution. In Figure 5.14 (Left) we show the histogram of *Nsig* and the PDF of a Gaussian fitting (using the calculated mean and variance of *Nsig*), while right plot illustrates the autocorrelation function of *Nsig*, where the samples of this signals behave as aapproximately independent.

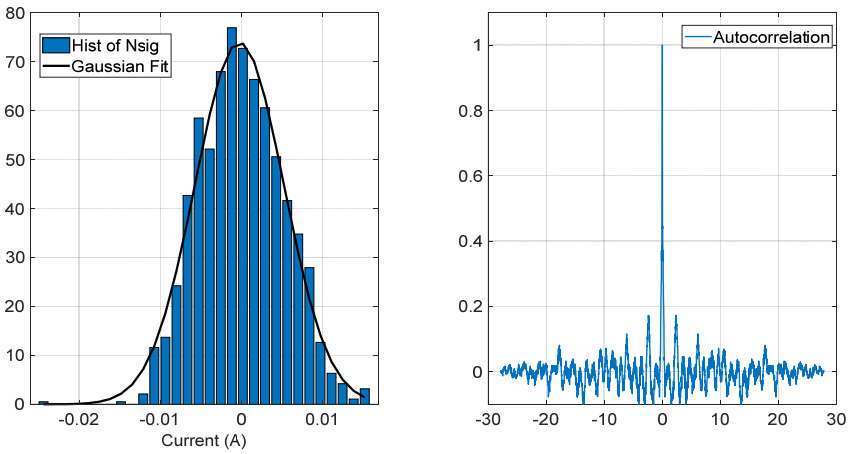


Figure 5.14: (Left) Histogram of Nsig values with fitted gaussian curve (Right) Autocorrelation of Nsig

To sum up: 20 parameters are used for the **S**teady-**S**tate **V**ariations **M**odel (SSVM):

- 3 coefficients of $f_1(m)$, the first fitting curve.
- 15 coefficients of $\sin_5(m)$, for the second fitting curve.
- 2 coefficients for the Gaussian fitting (mean and variance).

Each appliance may have several sets of parameters calculated at different events, and these can be used to generate realizations when needed. In our construction algorithm we selected the parameters-set randomly at each event. In Figure 5.15 and Figure 5.16 we show some realizations of the model for two more appliances. next paragraph will provide a plan for constructing the signals for each appliance.

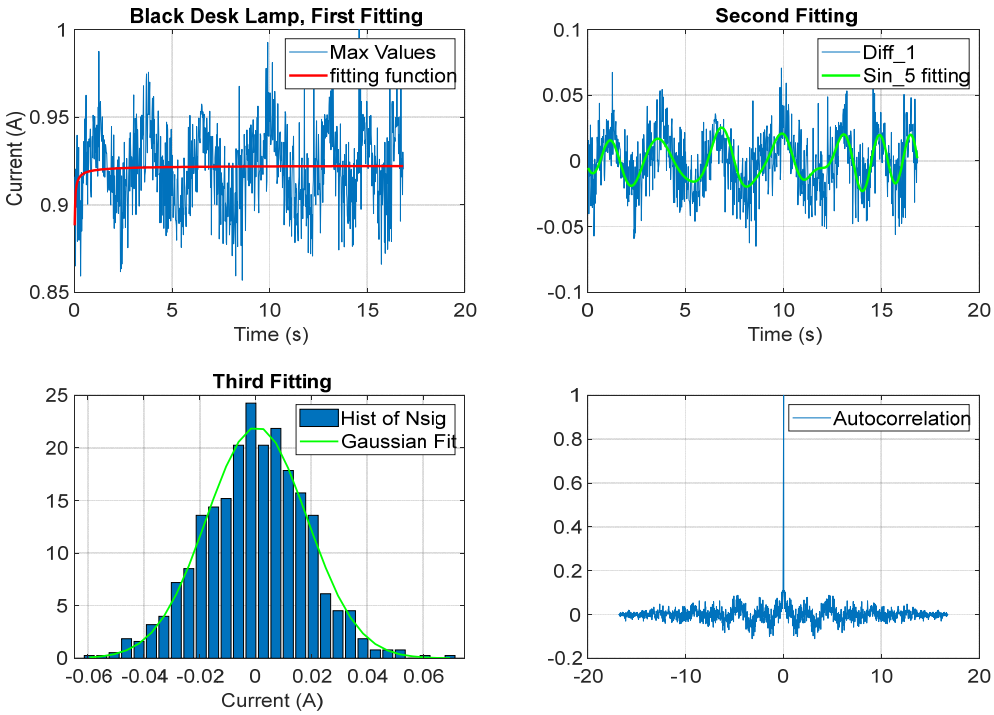


Figure 5.15: Modelling steady-state variations of a Lamp

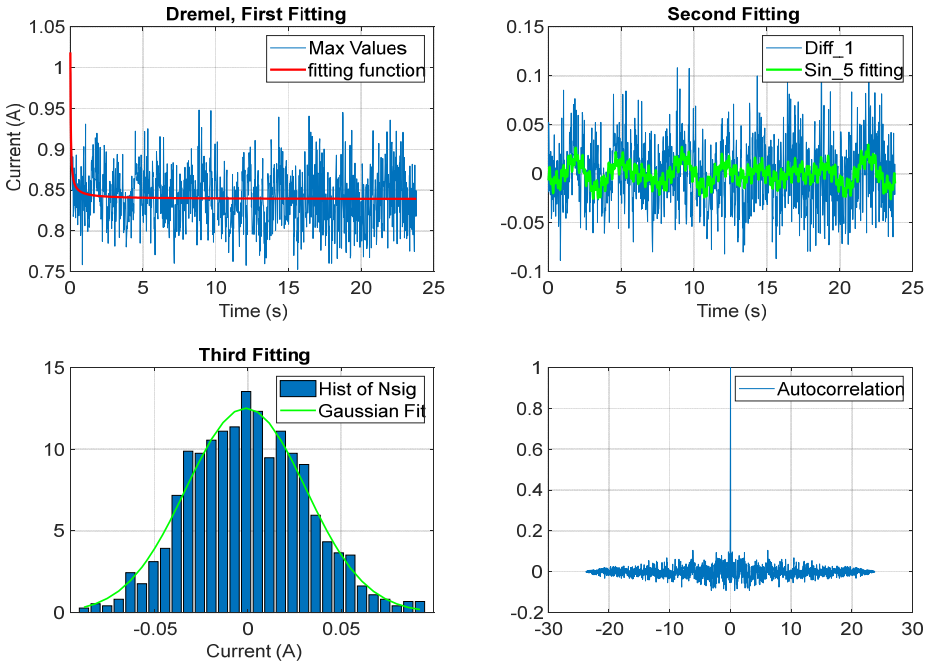


Figure 5.16: Modelling steady-state variations of a Dremel

5.2.3.2 Signal-Generation of individual appliance

The algorithm used to construct the individual signal I_i of the ith appliance is shown in Figure 5.17:

1) The signal I_i is initialized with zero values.
2) First event of the appliance is assumed 'on', otherwise a dummy one is added at the beginning of the events list. we also assume that each 'on' event is followed by an 'off' event, and vice versa. For the kth 'on' event, we create the signal $I_{i,k}$ which starts at the index of this event and ends at the index of the next 'off' event. Values of $I_{i,k}$ are created using the operational mode of the kth event. The transient state of $I_{i,k}$ is the same as that of the operational mode, and the steady state is a repeated version of the mode's steady state. The number of repetitions is dependent on the number of samples between k^{th} and $(k+1)^{th}$ events.

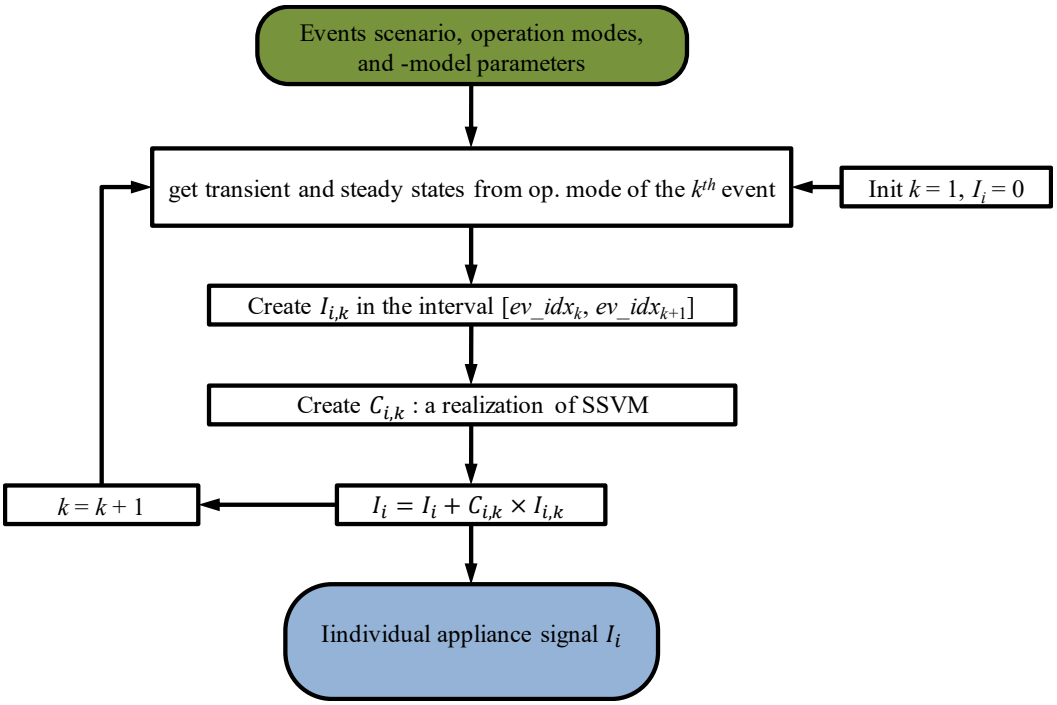


Figure 5.17: Signal-Generation of each individual appliance

3) Application of a steady-state variations model (SSVM): We randomly choose a set from the available sets of parameters of the ith appliance, then we generate a realization $C_{i,k}$ of the noise model. The number of samples in $C_{i,k}$ is equal to the number of periods in $I_{i,k}$. The nth period of $I_{i,k}$ is then multiplied by the sample $V_{i,k}(n)$ as:

$$I'_{i,k}(m) = I_{i,k}(m) \times C_{i,k}(\text{modulo}(m, N)) \tag{5.10}$$

4) The signal I_i is the sum of all $I'_{i,k}(m)$ signals.

5.2.4 Numerical comparisons

Once again, the performance of NILM algorithms will be compared at all stages of a typical NILM system between the original aggregate measurements and constructed aggregate measurements (out of aggregate real measurements).

5.2.4.1 Event Detection

A comparison of the performance of Hart and FBE event detectors on original measurement from HELD1 Dataset [14] against constructed aggregate signals out of real individual measurements (denoted Constructed_ind) and constructed aggregate signals out of real aggregate measurements (denoted Constructed_sub). The results are shown in Table 5.8 and Table 5.9, where P_r (recall) and P_{pre} (precision) are calculated for different values of detection parameters: T_1 : Step-size threshold (W), T_2: Min-steady length (Periods), T_3: Steady-state tolerance (W).

Results show that the performance of constructed signals out of individual measurements is generally higher than that of original signal, especially in P_{pre}values (less false events), whereas the performance of constructed signal based on aggregate measurements is closer to that of original signal. As a result, events detector's performance on Constructed_sub signals give quite a good idea about its performance on the original measurements.

Table 5.8: Hart Event Detector performance comparison

T_1	Original		Constructed_ind		Constructed_sub	
	P_r	P_{pre}	P_r	P_{pre}	P_r	P_{pre}
3	84.63	58.78	90.00	87.70	89.34	52.69
10	65.66	90.56	72.13	100	67.44	78.62
15	57.34	92.96	59.25	100	57.53	93.07

Table 5.9: FBE Event Detector performance comparison

T_2	T_3	Original		Constructed_ind		Constructed_sub	
		P_r	P_{pre}	P_r	P_{pre}	P_r	P_{pre}
5	5	85.28	83.43	87.88	90.01	89.59	80.44
5	10	77.28	90.22	75.56	100	79.94	84.90
10	5	85.19	85.62	87.63	90.95	88.25	91.66

5.2.4.2 Feature selection

Since the generalised construction approach can depend on real aggregate measurements, we can extend our comparisons to BLUED dataset. The collection of 84 features from chapter 4 (listed in Table 4.2 and Table 4.3) are calculated at each event of original and constructed signals. Best set of features are then selected using two algorithms, SFFS and our NEFSA.

The interesting comparison is between the resulted entropy values of all features on original and constructed signals, as to assess if the construction procedure affected the "ability of separation" of these features. Figure 5.18 illustrates the entropy curves of both types of measurements on phases A and B of BLUED dataset, where the values are relatively close, V-I trajectory features with indices (69-84) are of special interest since they are related to the signal shape and they exhibit the same behaviour for entropy calculations.

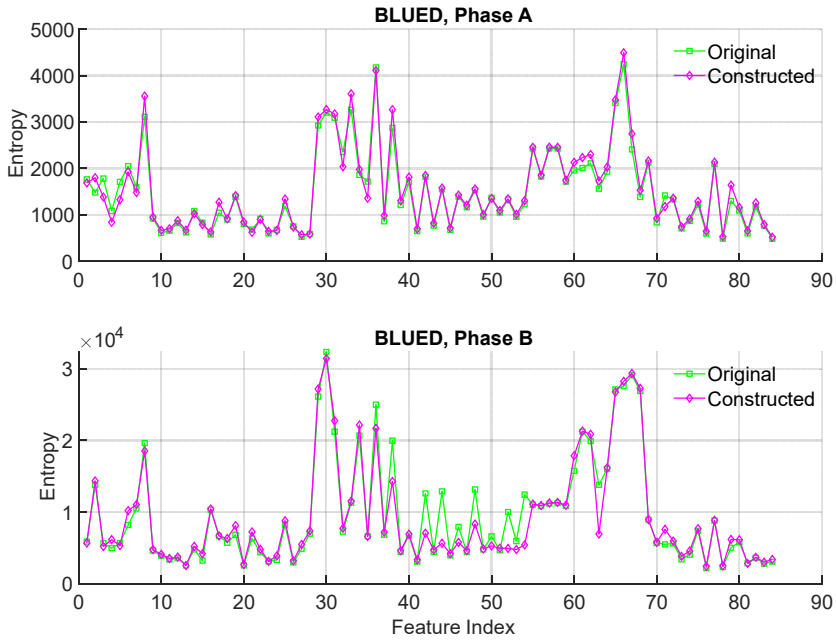


Figure 5.18: Entropy comparison: constructed vs original measurements

We list the NEFSA order of first 20 features also in Table 5.10, where only 1 feature was different (Bold) while all the other 19 are identical in both phases. This result is of special importance as it allows us to run NEFSA feature selector on the constructed signal to predict the suitable features subset for the original one.

Table 5.10: Best 20 features per NEFSA algorithm: Constructed vs Original

BLUED	**Best ranked 20 Features with NEFSA algorithm**	
Phase A	Original	84,78,27,16,28,76,81,23,10,13,41,11,45,24,21,73,26,43,83,**20**
	Constructed	84,78,27,28,21,16,23,76,81,10,24,13,41,11,45,26,73,83,**15**,43
Phase B	Original	76,78,20,13,83,81,26,41,84,23,15,24,11,73,12,82,10,45,74,**22**
	Constructed	76,78,13,20,81,83,23,26,41,84,11,82,12,73,24,10,45,15,74,**47**

5.2.4.3 Classification

The performance of SVM classifier (linear kernel) is evaluated on (BLUED, both phases) with three cases:

1- The constructed measurements (using ranked features with SFFS and NEFSA).

2- The original measurement (using ranked features with SFFS and NEFSA).

3- The original measurement (using ranked features of constructed measurements).

The third case is to test the suitability of constructed features-order for the original measurements, Figure 5.19 shows the resulted curves with respect to varying features-subset size (horizontal axis), while the vertical axis represents the true classification percentage.

SVM performed better on constructed measurements on phase A (4-8% more), yet the performance of original measurements had the same trends and values with both features-subsets, which prove the constructed measurement suitability for predicting one of the best features-subset for the original measurements (in other words, simulating the desired scenario before hand can help us to fix the features subset beforehand).

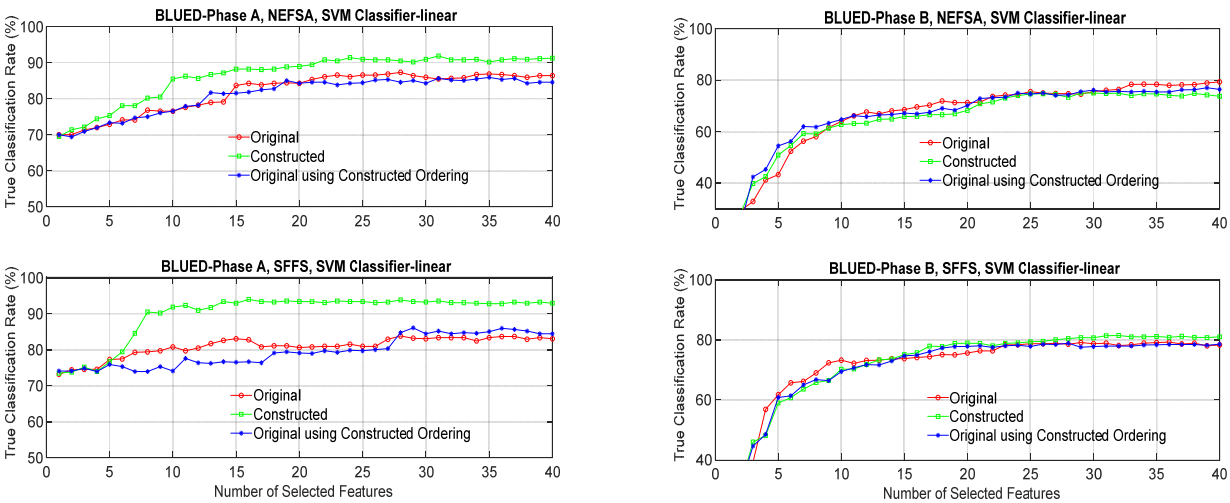


Figure 5.19: Testing classification performance on constructed and original measurements

5.3 Possible Applications of the Construction scheme

In the last sections, the feasibility of constructing NILM scenarios from a limited of group of measurements, the size and variability of this training group will affect the accuracy of the constructed scenarios as some operational modes may not appear in short measurements.

This scheme of construction has many potential applications, in this section we discuss some of those possibilities

5.3.1 providing more Data for Training/Testing NILM algorithms

One of the long-standing problems in the NILM research field is "useful" data scarcity. In chapter 2 we already mentioned that most public datasets were event-less (no ground-truth as in REDD [71]) and/or measured with a very low sampling rate (RAE [61]) and/or measured for a very short period (PLAID, COOLL, WHITED [56]–[58]). Another issue with the rest of real-world datasets (UKDALE [65], BLUED [15]) is the vast imbalance in the events among different appliances (e.g. the TV has 20 ON events per day while the dish washer has one event per day and the vacuum cleaner has one event per week); such usage patterns may result in biasing the classifiers towards majority classes during training phase.

The straightforward application of the construction scheme is to provide more suitable data for training and testing NILM algorithms out of the present problematic or incomplete ones. In Figure 5.20, a new NILM dataset can be created by extracting the operational modes from several datasets and then design a new one according to desired scenarios or randomly as proposed in Figure 5.21.

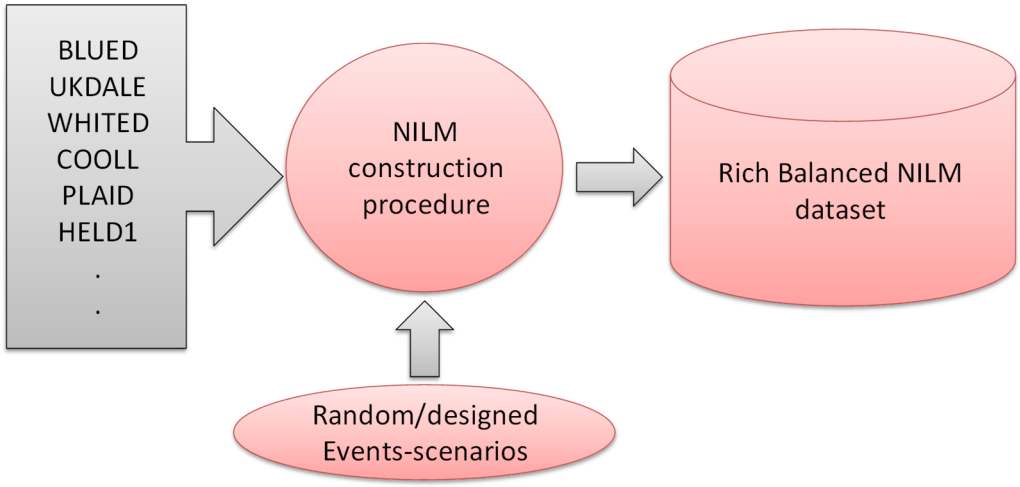


Figure 5.20: Combining different public datasets with NILM construction

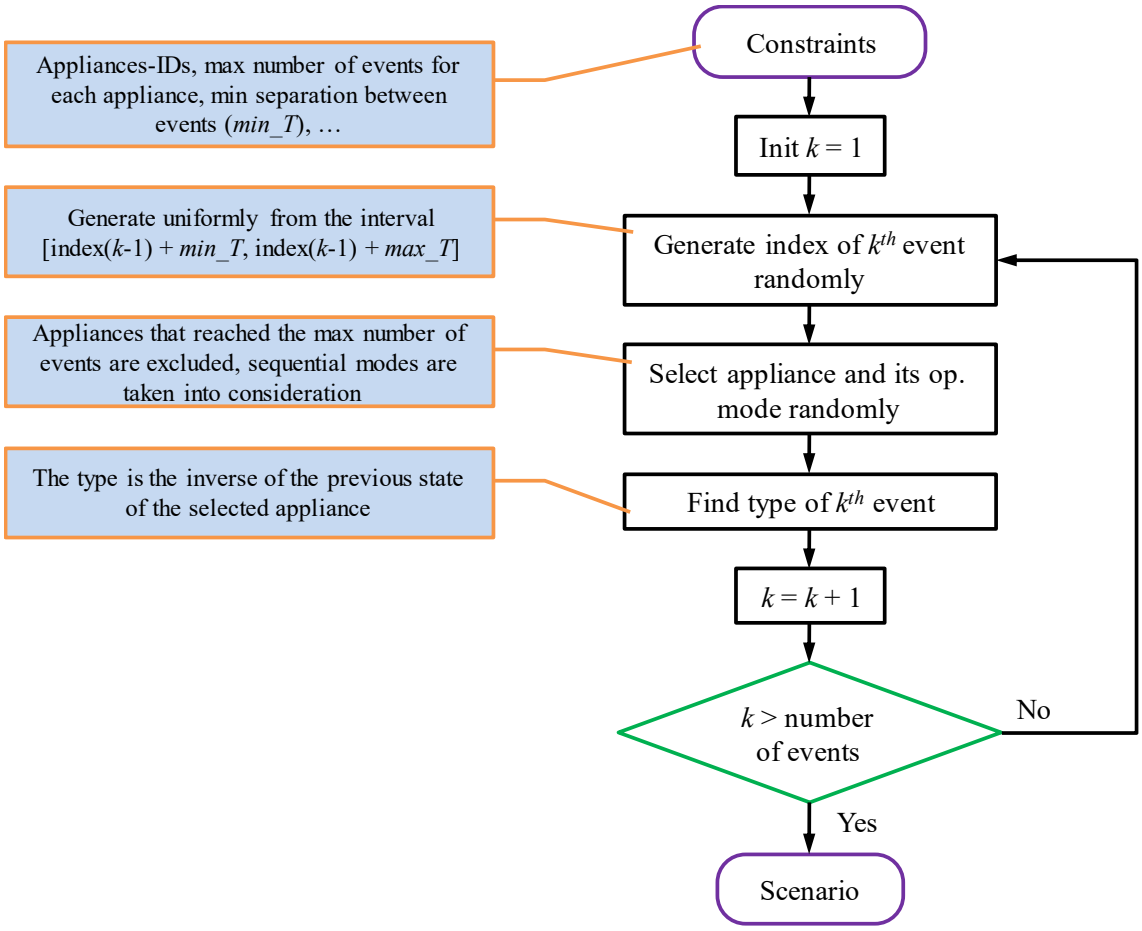


Figure 5.21: Random generation of NILM events-scenarios

5.3.2 Tracking classified appliances for abnormal behaviours

After having classified all appliances correctly (in the ideal case), what would be possible to do with such information besides the obvious generation of detailed power consumption bill?

One of the ambitious goals of NILM research is to enable anomalous behaviour detection, with several approaches in [130]–[132] for detecting unusual power consumption in behaviour-time context (training on past days or weeks to learn behavioural patterns, learning from other buildings in the area as the weather effect will be similar).

One potential application of NILM construction procedure is to generate tracking reports to point out abnormal patterns. We propose a tracking concept based on our construction procedure. A simple flow chart is shown in Figure 5.22.

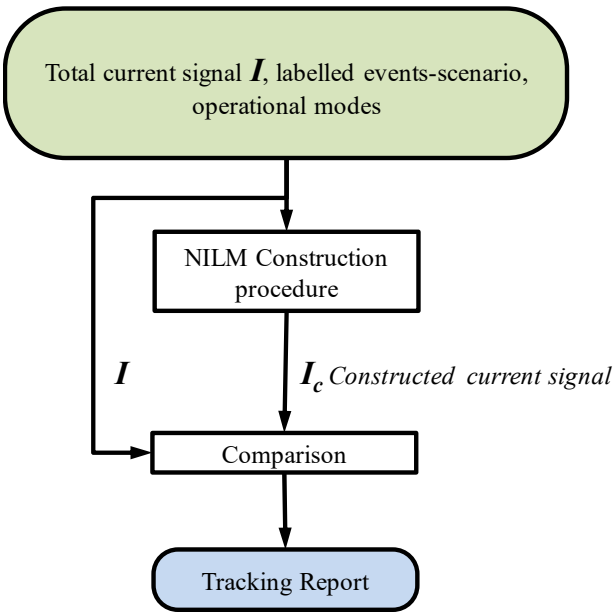


Figure 5.22: simple tracking scheme

5.3.2.1 Tracking principle

As can be seen in Figure 5.22, the tracking block has the following inputs:

- The aggregate current signal $I(t)$ at the input of the smart meter.
- Operation modes of all appliances (should be stored in a dynamic list after a training phase).
- Labelled sequence of events (as outputs from event detector and classifier).

The constructed signal I_c can be written as:

$$I_c(t) = \sum_{i=1}^{M} f_i(t) \tag{5.11}$$

where f_i is the operational mode of the ith appliance, and M is the number of active appliances at time t. Each operational mode can be written as:

$$f_i(t) = g_i(t) + v_i(t) \tag{5.12}$$

where $g_i(t)$ is the noise-free operational mode, and $v_i(t)$ is the remaining noise after extracting this operational mode; the constructed signal can be rewritten as:

$$I_c(t) = \sum_{i=1}^{M} g_i(t) + \sum_{i=1}^{M} v_i(t) \tag{5.13}$$

On the other hand, the original signal can be written as

$$I(t) = \sum_{i=1}^{M} g_i(t) + v(t) \tag{5.14}$$

where v is the noise term. The difference signal can be then calculated as follows:

$$d(t) = I(t) - I_c(t) = \sum_{i=1}^{M} g_i(t) + v(t) - \sum_{i=1}^{M} g_i(t) - \sum_{i=1}^{M} v_i(t) \tag{5.15}$$

$$\Leftrightarrow d(t) = v(t) - \sum_{i=1}^{M} v_i(t) \tag{5.16}$$

The difference term is only related to the noise (noise is broadly defined here to include all contributions from different sources: electrical network and appliances and measurement system). Practical tests show an increase in the amplitude of the difference signal when a high-power appliance is turned on, so we "loosely" assume that the difference signal can divided among the active appliances with ratios proportional to their nominal powers. In this case, a high-power appliance will be given a high portion of the difference signal. According to this proposed model, the estimated individual signals can be written as

$$I_i(t) = f_i(t) + a_i \times d(t), \;\; i = 1,2,\dots,M \tag{5.17}$$

where a_i is a coefficient that is calculated as follows

$$a_i = \frac{P_i}{\sum_{j=1}^{M} P_j} \quad : P_j \text{ is the power of the } j^{th} \text{ appliance} \tag{5.18}$$

5.3.2.2 Evaluation

Since HELD1 dataset has labelled individual measurements as well as labelled aggregate measurements for the same group of devices, we can test the proposed concept of dividing the difference signal among active devices for tracking as in the illustrated methodology in Figure 5.23.

On one hand, the tracker receives a set of aggregate measurements with their ground truth, i.e. labelled events. The tracker can access a list of operation modes extracted from half the individual measurements. The other half of the individual measurements are compared with the individual current signals at the output of the tracker as described in equation (5.19).

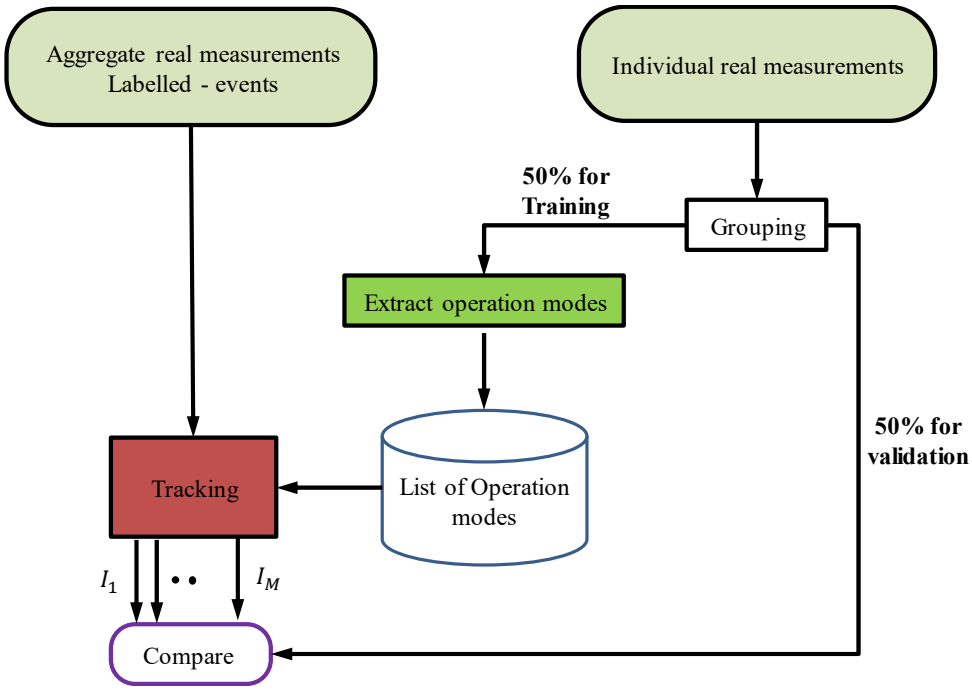


Figure 5.23: Testing the resulted indivdual signals

Visual comparison is displayed in Figure 5.24, where the signals of three devices are compared, distributing the difference signal did not affect the shape of the current signals.

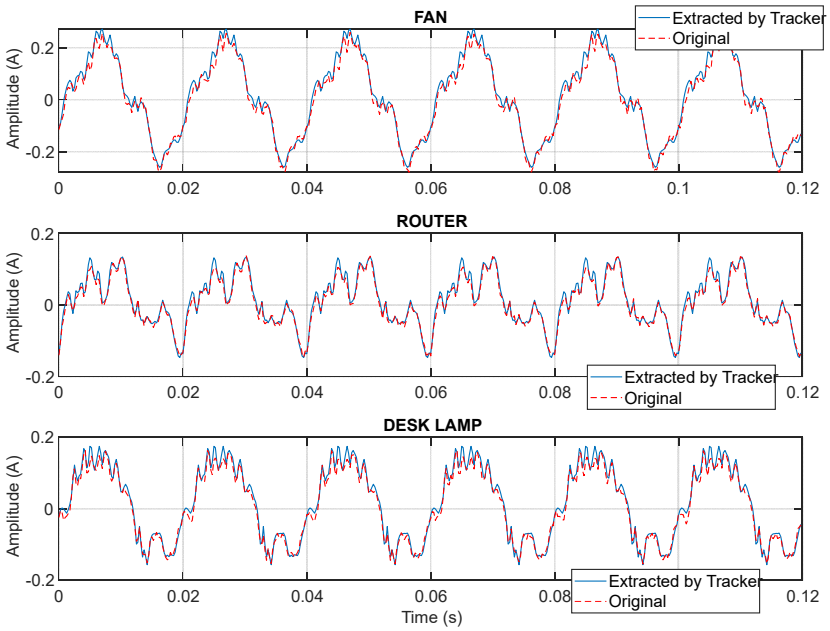


Figure 5.24: Comparing current signals: Tracker output vs Original individual measurements

5.3.2.3 Use-cases for Tracking scheme

Most use-case for tracker are generally use-cases for the smart meter.

1. Providing the smart meter with pattern information (typical usage of appliances in the day and at night) and the number of residents enable the tracker to compare this profile against the real time consumption pattern and report any excessive consumption in an unusual time.

2. Tracker can point out power wastage through comparing with general consumption average per appliance, e.g. alarm the parents of long active TV hours can help them plan their children activities better.

3. Directed maintenance: If a specific defect is defined, the tracker can compare the resulted signal shape of this device in the normal consumption circumstances

4. predicting condition deteriorations: Aging appliances like an old fridge can exhibit unusual power profile. Tracker can compare it to its history records to detect differences, yet it is not straightforward as the signal difference will be distributed to other appliances as well.

5.4 Graphical tool for NILM signals analysis and construction

Several tools were developed recently to establish a cumulative research effort in the NILM field, Batra et al [135] developed NILMTK as a tool-kit for benchmarking NILM algorithms, Kelly [136] and Pereira [137] proposed separately two interesting metadata schemes for unifying public datasets. nonetheless, one of the hardest problems that we have encountered was replicating other algorithms described in

NILM research papers (mostly due to lack of implementation details). As a contribution to NILM research community we provide a Graphical tool (GUI) for NILM signal construction (some of its interfaces are illustrated in Figure 5.25), all employed algorithms (signal construction, feature selection, event detection, modelling of steady-state variations) were fully explained in chapters (3-4-5) of this Thesis, we only refer here to its existence and objectives while appendix 4 provides more details for all capabilities of the tool and how to use it properly.

5.4.1 objectives of the tool

the main objectives are to

- give interested NILM researcher/enthusiast the ability to generate any desired scenario of NILM signals and explore its abilities, provided that he has a very limited set of measurements of the target appliances, those measurements can be
    - ❖ from digital datasets of specific devices provided by the producers.
    - ❖ A combination of measurements from several public datasets (Ukdale [65], HELD1 [14], COOLL[63], Plaid [56], among others).
    - ❖ self-gathered with a data logger/smart meter.
- provide graphical user interfaces to compare all the steps of NILM process on both sides: the original provided signals, and the constructed ones.
- Supply the user with a rich bank of NILM features, with the ability for him to add/edit the features as he sees fit.

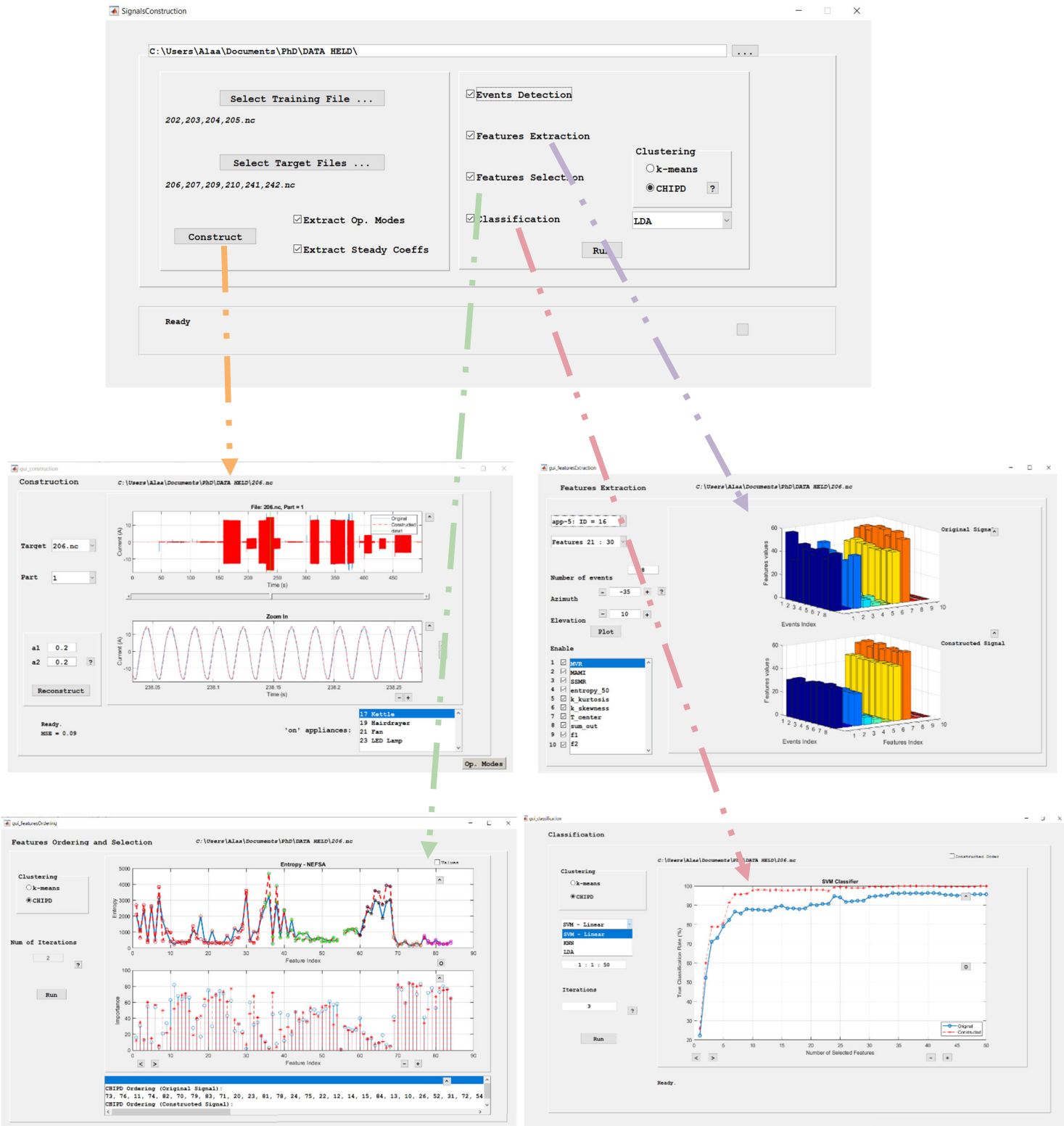


Figure 5.25: Sample interfaces from the graphical tool

6 Summary and Perspectives

In this chapter, we summarize the content of this thesis and discuss the possible future steps in the development of NILM systems.

6.1 Chapter Summaries

We commenced in chapter 1 by introducing NILM as an important technology for optimizing energy consumption and a hot research field. We then listed the research contributions of this work. Chapter 2 started with defining the basic concepts of NILM through an example from Hart's work, then we reviewed the state-of-the-art of NILM research and looked at the NILM public datasets.

In Chapter 3, we introduced Huang-Hilbert transform (HHT) and presented the two separate research methods we followed: the first one is motivated by the empirical mode decomposition (which comprises the first step of HHT), where we adaptively extracted narrow-banded modes from the NILM current signal. Those modes were used to build an effective event detector that performed well on real house

measurements. the resulted modes were also used to extract spectral features from the transient states of some appliances.

The second method looked at the final outcome of classical HHT application on aggregate current signals, where we discussed the tuning method of sifting process along with an empirical method to detect the limits of transient states in NILM signals; then presented the instantaneous frequency curves with time and proposed to use their characteristics as features if the computational cost is not a problem. other statistical features from our observations were also presented with classification results with those features only.

Chapter 4 addressed the feature selection problem for getting the best load signature (e.g. set of features) from available NILM features. We proposed a novel entropy-based algorithm for NILM feature selection and presented several comparisons against the well-known sequential feature selection algorithms (SFS, SFFS). the assessment showed a comparable classification performance with several classifiers while the computational cost for our algorithm is much lower than other computation hungry algorithms.

Chapter 5 was dedicated to a new scheme of NILM signal construction, we began with constructing aggregate measurements out of single-device measurements (i.e. measurements with only that device present), then we generalised the method for constructing any type of measurement out of any set of limited. We presented detailed numerical comparisons and discussed the possible applications of such scheme; then show a use-case for tracking classified devices with its different benefits.

6.2 Conclusions & Future Work

To extend the research topics addressed in this thesis, we provide some thoughts for "near and far" future steps:

(1) Extending FBE event detector for simultaneous events detection

Almost all event detection approaches assume the switch continuity principle [17]. In non-residential environments, simultaneous events can be a serious challenge for smart meters. as a future step, we intend to test (and further develop) the FBE event detector presented in chapter 3 on such events in different sampling rates scenarios.

(2) Implementing the proposed feature selection algorithm NEFSA on a low-cost embedded system.
(3) Exploiting the latest advances in machine learning to identify the most effective set of features

Recently, black-box machine learning models are examined by new methods [138], [139] with the intention of making those models "explainable". The reported success with deep learning models in NILM experiments can be extended by examining the important features used by such models to focus our data analysis. Tools like SHAP values [140] are already in use in other machine learning applications and can help us to gather optimized features that can be later provided to smart meters. Those features along with quick feature selection algorithms as the one we proposed would guarantee the best attainable energy disaggregation possible.

(4) Considering other models of devices in the NILM signals construction scheme

The operational modes concept in chapter 5 is not valid for continuously varying devices like dimmers, nor for always "switched ON" devices (devices with constant power consumption like security cameras and alarm systems). The operational modes model can be extended to account for them (learning a development track of power consumption for example).

(5) Incorporating other data formats used in publicly available datasets (FLAC, csv, txt) for scenario based NILM construction graphical tool, the input files are expected in NetCDF format for now.

(6) Testing the derived HHT features in this thesis for discriminating among identical devices if "Cloud NILM" is adopted

As we saw in chapter 3, interesting instantaneous frequency curves can be calculated by applying HHT, yet the computational cost is high and not feasible for low-cost NILM solutions. If a "Cloud NILM" model is adopted to ignore the complexity drawback and more accurate methods are employed for determining the limits of the transient state, then an automatic extraction will be possible for such curves. Experimenting with those features for identical devices identification can yield good results in my opinion.

7 Bibliography

[1] G. W. Hart, *Nonintrusive Appliance Load Data Acquisition Method: Progress Report*. MIT Energy Laboratory, 1984.

[2] G. W. Hart, 'Prototype nonintrusive appliance load monitor', in *MIT Energy Laboratory Technical Report, and Electric Power Research Institute Technical Report*, 1985.

[3] D. Parker, D. Hoak, A. Meier, and R. Brown, 'How much energy are we using? Potential of residential energy demand feedback devices', *Proceedings of the 2006 Summer Study on Energy Efficiency in Buildings, American Council for an Energy Efficient Economy, Asilomar, CA*, 2006.

[4] C. Fischer, 'Feedback on household electricity consumption: a tool for saving energy?', *Energy efficiency*, vol. 1, no. 1, pp. 79–104, 2008.

[5] K. C. Armel, A. Gupta, G. Shrimali, and A. Albert, 'Is disaggregation the holy grail of energy efficiency? The case of electricity', *Energy Policy*, vol. 52, pp. 213–234, 2013.

[6] G. H. Hart, 'Nonintrusive appliance load monitoring', vol. 80, no. 12, pp. 1870–1891.

[7] 'Companies offering NILM products and services', *Companies | NILM Wiki*. [Online]. Available: http://wiki.nilm.eu/companies.html. [Accessed: 07-Nov-2019].

[8] R. H. Socolow, 'The Twin Rivers program on energy conservation in housing: Highlights and conclusions', *Energy and Buildings*, vol. 1, no. 3, pp. 207–242, 1978.

[9] J. Seryak and K. Kissock, 'Occupancy and behavioral affects on residential energy use', in *Proceedings of the Solar conference*, 2003, pp. 717–722.

[10] L. Pereira, F. Quintal, M. Barreto, and N. J. Nunes, 'Understanding the limitations of eco-feedback: a one-year long-term study', in *International Workshop on Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data*, 2013, pp. 237–255.

[11] J. Kelly and W. Knottenbelt, 'Does disaggregated electricity feedback reduce domestic electricity consumption? A systematic review of the literature', *arXiv preprint arXiv:1605.00962*, 2016.

[12] T. Schwartz, S. Denef, G. Stevens, L. Ramirez, and V. Wulf, 'Cultivating energy literacy: results from a longitudinal living lab study of a home energy management system', in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013, pp. 1193–1202.

[13] D. Gielen, F. Boshell, D. Saygin, M. D. Bazilian, N. Wagner, and R. Gorini, 'The role of renewable energy in the global energy transformation', *Energy Strategy Reviews*, vol. 24, pp. 38–50, 2019.

[14] P. Held, S. Mauch, A. Saleh, D. Benyoucef, and Ould Abdeslam, Djaffar, 'HELD1: Home Equipment Laboratory Dataset for Non-Intrusive Load Monitoring', presented at the SIGNAL 2018, Nice, France, 2018.

[15] K. Anderson, A. Ocneanu, D. Benitez, A. Rowe, and and M. Berges, 'BLUED: A fully labeled public dataset for event-based non-intrusive load monitoring research', in *Proceedings of the 2nd KDD Workshop on Data Mining Applications in Sustainability (SustKDD)*, 2011.

[16] M. Zeifman and K. Roth, 'Nonintrusive appliance load monitoring: Review and outlook', *IEEE transactions on Consumer Electronics*, vol. 57, no. 1, pp. 76–84, 2011.

[17] S. Makonin, 'Investigating the switch continuity principle assumed in Non-Intrusive Load Monitoring (NILM)', in *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2016, pp. 1–4.

[18] A. U. Haq, 'Appliance Event Detection for Non-Intrusive Load Monitoring in Complex Environments', PhD Thesis, Technische Universität München, 2018.

[19] A. U. Haq and H.-A. Jacobsen, 'Prospects of appliance-level load monitoring in off-the-shelf energy monitors: A technical review', *Energies*, vol. 11, no. 1, p. 189, 2018.

[20] P. Meehan, C. McArdle, and S. Daniels, 'An efficient, scalable time-frequency method for tracking energy usage of domestic appliances using a two-step classification algorithm', *Energies*, vol. 7, no. 11, pp. 7041–7066, 2014.

[21] M. Weiss, A. Helfenstein, F. Mattern, and T. Staake, 'Leveraging smart meter data to recognize home appliances', in *2012 IEEE International Conference on Pervasive Computing and Communications*, 2012, pp. 190–197.

[22] K. N. Trung, E. Dekneuvel, B. Nicolle, O. Zammit, C. N. Van, and G. Jacquemod, 'Event detection and disaggregation algorithms for nialm system', in *Proceedings of 2nd International Non-Intrusive Load Monitoring (NILM) Workshop*, 2014.

[23] B. Wild, K. S. Barsim, and B. Yang, 'A new unsupervised event detector for non-intrusive load monitoring', in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2015, pp. 73–77.

[24] L. De Baets, J. Ruyssinck, C. Develder, T. Dhaene, and D. Deschrijver, 'On the Bayesian optimization and robustness of event detection methods in NILM', *Energy and Buildings*, vol. 145, pp. 57–66, 2017.

[25] Y.-C. Su, K.-L. Lian, and H.-H. Chang, 'Feature selection of non-intrusive load monitoring system using STFT and wavelet transform', in *2011 IEEE 8th International Conference on e-Business Engineering*, 2011, pp. 293–298.

[26] P. Held, F. Laasch, D. O. Abdeslam, and D. Benyoucef, 'Frequency invariant transformation of periodic signals (FIT-PS) for signal representation in NILM', in *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 5149–5154.

[27] M. Bergés and Z. Kolter, 'Non-intrusive load monitoring: A review of the state of the art', in *International Workshop on Non-Intrusive Load Monitoring, Pittsburgh, PA, USA*, 2012.

[28] H. Gonçalves, A. Ocneanu, M. Bergés, and R. Fan, 'Unsupervised disaggregation of appliances using aggregated consumption data', in *The 1st KDD Workshop on Data Mining Applications in Sustainability (SustKDD)*, 2011.

[29] J. Z. Kolter, S. Batra, and A. Y. Ng, 'Energy disaggregation via discriminative sparse coding', in *Advances in Neural Information Processing Systems*, 2010, pp. 1153–1161.

[30] D. F. Teshome, T. D. Huang, and K.-L. Lian, 'Distinctive load feature extraction based on fryze's time-domain power theory', *IEEE Power and Energy Technology Systems Journal*, vol. 3, no. 2, pp. 60–70, 2016.

[31] T. J. Rouphael, *RF and digital signal processing for software-defined radio: a multi-standard multi-mode approach*. Newnes, an imprint of Elsevier, 2009.

[32] S. Gupta, M. S. Reynolds, and S. N. Patel, 'ElectriSense: single-point sensing using EMI for electrical event detection and classification in the home', in *Proceedings of the 12th ACM international conference on Ubiquitous computing*, 2010, pp. 139–148.

[33] G. Peeters, 'A large set of audio features for sound description (similarity and classification) in the CUIDADO project', Paris, France, Technical, 2004.

[34] D. Kelly, 'Disaggregation of domestic smart meter energy data', PhD Thesis, Imperial College London, 2016.

[35] A. Zoha, A. Gluhak, M. Imran, and S. Rajasegarar, 'Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey', *Sensors*, vol. 12, no. 12, pp. 16838–16866, 2012.

[36] T. Bernard, 'Non-Intrusive Load Monitoring (NILM): combining multiple distinct electrical features and unsupervised machine learning techniques', PhD Thesis, 2018.

[37] M. B. Figueiredo, A. De Almeida, and B. Ribeiro, 'An experimental study on electrical signature identification of non-intrusive load monitoring (nilm) systems', in *International Conference on Adaptive and Natural Computing Algorithms*, 2011, pp. 31–40.

[38] H. Chang, K. Lian, Y. Su, and W. Lee, 'Power-Spectrum-Based Wavelet Transform for Nonintrusive Demand Monitoring and Load Identification', *IEEE Transactions on Industry Applications*, vol. 50, no. 3, pp. 2081–2089, May 2014.

[39] A. Saleh, P. Held, D. Benyoucef, and D. O. Abdeslam, 'A Novel Procedure for Virtual Measurements Generation suitable for Training and Testing in the context of Non Intrusive Load Monitoring', *SIGNAL 2018 Editors*, p. 36, 2018.

[40] J. Gao, E. C. Kara, S. Giri, and M. Bergés, 'A feasibility study of automated plug-load identification from high-frequency measurements', in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2015, pp. 220–224.

[41] G. Lin, S. Lee, J. Y. Hsu, and W. Jih, 'Applying power meters for appliance recognition on the electric panel', in *2010 5th IEEE Conference on Industrial Electronics and Applications*, 2010, pp. 2254–2259.

[42] S. Giri, M. Bergés, and A. Rowe, 'Towards automated appliance recognition using an EMF sensor in NILM platforms', *Advanced Engineering Informatics*, vol. 27, no. 4, pp. 477–485, 2013.

[43] J. Yu, Y. Gao, Y. Wu, D. Jiao, C. Su, and X. Wu, 'Non-Intrusive Load Disaggregation by Linear Classifier Group Considering Multi-Feature Integration', *Applied Sciences*, vol. 9, no. 17, p. 3558, 2019.

[44] J. Kelly and W. Knottenbelt, 'Neural nilm: Deep neural networks applied to energy disaggregation', in *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, 2015, pp. 55–64.

[45] M. Xia, K. Wang, X. Zhang, Y. Xu, and others, 'Non-intrusive load disaggregation based on deep dilated residual network', *Electric Power Systems Research*, vol. 170, pp. 277–285, 2019.

[46] J. Kim, T.-T.-H. Le, and H. Kim, 'Nonintrusive load monitoring based on advanced deep learning and novel signature', *Computational intelligence and neuroscience*, vol. 2017, 2017.

[47] P. Held, S. Mauch, A. Saleh, D. O. Abdeslam, and D. Benyoucef, 'Frequency Invariant Transformation of Periodic Signals (FIT-PS) for Classification in NILM', *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5556–5563, Sep. 2019.

[48] Q. Wu and F. Wang, 'Concatenate Convolutional Neural Networks for Non-Intrusive Load Monitoring across Complex Background', *Energies*, vol. 12, no. 8, p. 1572, 2019.

[49] K. Chen, Q. Wang, Z. He, K. Chen, J. Hu, and J. He, 'Convolutional sequence to sequence non-intrusive load monitoring', *The Journal of Engineering*, vol. 2018, no. 17, pp. 1860–1864, 2018.

[50] R. Bonfigli, S. Squartini, M. Fagiani, and F. Piazza, 'Unsupervised algorithms for non-intrusive load monitoring: An up-to-date overview', in *2015 IEEE 15th International Conference on Environment and Electrical Engineering (EEEIC)*, 2015, pp. 1175–1180.

[51] K. S. Barsim and B. Yang, 'Toward a semi-supervised non-intrusive load monitoring system for event-based energy disaggregation', in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2015, pp. 58–62.

[52] M. Baranski and J. Voss, 'Genetic algorithm for pattern detection in NIALM systems', in *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, 2004, vol. 4, pp. 3462–3468.

[53] K. Suzuki, S. Inagaki, T. Suzuki, H. Nakamura, and K. Ito, 'Nonintrusive appliance load monitoring based on integer programming', in *2008 SICE Annual Conference*, 2008, pp. 2742–2747.

[54] F. M. Wittmann, J. C. López, and M. J. Rider, 'Nonintrusive load monitoring algorithm using mixed-integer linear programming', *IEEE Transactions on Consumer Electronics*, vol. 64, no. 2, pp. 180–187, 2018.

[55] L. Pereira and N. Nunes, 'Performance evaluation in non-intrusive load monitoring: Datasets, metrics, and tools—A review', *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 6, p. e1265, 2018.

[56] J. Gao, S. Giri, E. C. Kara, and M. Bergés, 'PLAID: a public dataset of high-resoultion electrical appliance measurements for load identification research: demo abstract', in *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings(BuildSys '14)*, Memphis, Tennessee, 2014, pp. 10–1145.

[57] M. Kahl, A. U. Haq, T. Kriechbaumer, and H.-A. Jacobsen, 'Whited-a worldwide household and industry transient energy data set', in *3rd International Workshop on Non-Intrusive Load Monitoring*, 2016.

[58] T. Picon *et al.*, 'COOLL: Controlled on/off loads library, a public dataset of high-sampled electrical signals for appliance identification', *arXiv preprint arXiv:1611.05803*, 2016.

[59] K. S. Barsim, L. Mauch, and B. Yang, 'Neural network ensembles to real-time identification of plug-level appliance measurements', *arXiv preprint arXiv:1802.06963*, 2018.

[60] T. Kriechbaumer and H.-A. Jacobsen, 'BLOND, a building-level office environment dataset of typical electrical appliances', *Scientific data*, vol. 5, p. 180048, 2018.

[61] S. Makonin, Z. J. Wang, and C. Tumpach, 'RAE: The rainforest automation energy dataset for smart grid meter data analysis', *data*, vol. 3, no. 1, p. 8, 2018.

[62] M. Ribeiro, L. Pereira, F. Quintal, and N. Nunes, 'SustDataED: A public dataset for electric energy disaggregation research', in *ICT for Sustainability 2016*, 2016.

[63] T. Picon *et al.*, 'COOLL: Controlled On/Off Loads Library, a Public Dataset of High-Sampled Electrical Signals for Appliance Identification', *CoRR*, vol. abs/1611.05803, 2016.

[64] C. Beckel, W. Kleiminger, R. Cicchetti, T. Staake, and S. Santini, 'The ECO data set and the performance of non-intrusive load monitoring algorithms', in *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, 2014, pp. 80–89.

[65] J. Kelly and W. Knottenbelt, 'The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes', *Scientific data*, vol. 2, p. 150007, 2015.

[66] N. Batra, O. Parson, M. Berges, A. Singh, and A. Rogers, 'A comparison of non-intrusive load monitoring methods for commercial and residential buildings', *arXiv:1408.6595*, 2014.

[67] S. Makonin, F. Popowich, L. Bartram, B. Gill, and I. V. Bajić, 'AMPds: A public dataset for load disaggregation and eco-feedback research', in *2013 IEEE Electrical Power & Energy Conference*, 2013, pp. 1–6.

[68] M. Lichman, *UCI machine learning repository, 2013*. 2013.

[69] N. Batra, M. Gulati, A. Singh, and M. B. Srivastava, 'It's Different: Insights into home energy consumption in India', in *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, 2013, pp. 1–8.

[70] A. Ridi, C. Gisler, and J. Hennebert, 'ACS-F2—A new database of appliance consumption signatures', in *2014 6th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, 2014, pp. 145–150.

[71] J. Z. Kolter and M. J. Johnson, 'REDD: A public data set for energy disaggregation research', in *Workshop on Data Mining Applications in Sustainability (SIGKDD), San Diego, CA*, 2011, vol. 25, pp. 59–62.

[72] T. Bier, 'Disaggregation of Electrical Appliances using Non-Intrusive Load Monitoring', PhD Thesis, 2014.

[73] A. Marchiori, D. Hakkarinen, Q. Han, and L. Earle, 'Circuit-Level Load Monitoring for Household Energy Management', *IEEE Pervasive Computing*, vol. 10, no. 1, pp. 40–48, Jan. 2011.

[74] L. Farinaccio and R. Zmeureanu, 'Using a pattern recognition approach to disaggregate the total electricity consumption in a house into the major end-uses', *Energy and Buildings*, vol. 30, no. 3, pp. 245–259, 1999.

[75] J. Froehlich, E. Larson, S. Gupta, G. Cohn, M. Reynolds, and S. Patel, 'Disaggregated End-Use Energy Sensing for the Smart Grid', *IEEE Pervasive Computing*, vol. 10, no. 1, pp. 28–39, Jan. 2011.

[76] H. Chang, M. Lee, N. Chen, C. Chien, and W. Lee, 'Feature extraction based hellinger distance algorithm for non-intrusive aging load identification in residential buildings', in *2015 IEEE Industry Applications Society Annual Meeting*, 2015, pp. 1–8.

[77] D. Jung, H. H. Nguyen, and D. K. Y. Yau, 'Tracking appliance usage information using harmonic signature sensing', in *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2015, pp. 459–465.

[78] J. A. Hoyo-Montaño, C. A. Pereyda-Pierre, J. M. Tarín-Fontes, and J. N. Leon-Ortega, 'Overview of Non-Intrusive Load Monitoring: A way to energy wise consumption', in *2016 13th International Conference on Power Electronics (CIEP)*, 2016, pp. 221–226.

[79] J. L. Rojo-Alvarez, M. Martinez-Ramon, J. Munoz-Mari, and G. Camps-Valls, *Digital Signal Processing with Kernel Methods*, 1st ed. Wiley-IEEE Press, 2018.

[80] R. Yan and R. X. Gao, 'Hilbert-huang transform based time-frequency distribution and comparisons with other three', *IEEE Instrumentation Measurement Magazine*, vol. 10, no. 5, pp. 40–45, Oct. 2007.

[81] J. Gillis and W. G. Morsi, 'Non-intrusive load monitoring using orthogonal wavelet analysis', in *Electrical and Computer Engineering (CCECE) 2016 IEEE Canadian Conference*, 2016, pp. 1–5.

[82] N. E. Huang, Z. Shen, and S. R. Long, 'The Empirical Mode Decomposition and The Hilbert Spectrum for Nonlinear Non stationary Time Series Analysis', *the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 454, pp. 903–995, 1998.

[83] N. E. Huang and N. O. Attoh-Okine, *The Hilbert-Huang transform in engineering*. CRC Press, 2005.

[84] N. Agana and A. Homaifar, 'EMD-based predictive deep belief network for time series prediction: an application to drought forecasting', *Hydrology*, vol. 5, no. 1, p. 18, 2018.

[85] H. Dong, K. Qi, X. Chen, Y. Zi, Z. He, and B. Li, 'Sifting process of EMD and its application in rolling element bearing fault diagnosis', *Journal of Mechanical Science and Technology*, vol. 23, no. 8, pp. 2000–2007, Aug. 2009.

[86] G. Rilling, P. Flandrin, P. Goncalves, and others, 'On empirical mode decomposition and its algorithms', in *IEEE-EURASIP workshop on nonlinear signal and image processing*, 2003, vol. 3, pp. 8–11.

[87] R. J. Gledhill, 'Methods for investigating conformational change in biomolecular simulations.', 2004.

[88] P. Flandrin, P. Gonçalves, and G. Rilling, 'EMD equivalent filter banks, from interpretation to applications', in *Hilbert-Huang transform and its applications*, World Scientific, 2005, pp. 57–74.

[89] Z. Wu and N. E. Huang, 'Ensemble empirical mode decomposition: a noise-assisted data analysis method', *Advances in adaptive data analysis*, vol. 1, no. 01, pp. 1–41, 2009.

[90] A. D. Veltcheva, 'An application of HHT method to the nearshore sea waves', in *The Hilbert–Huang transform in engineering*, CRC Press, 2005, pp. 97–119.

[91] R. R. Zhang, '8 An HHT-Based Approach to Quantify Nonlinear Soil Amplification and Damping', in *The Hilbert-Huang Transform in Engineering*, CRC Press, 2005, pp. 161–191.

[92] A. Saleh, P. Held, D. Benyoucef, and D. Ould Abdeslam, 'EMD inspired filtering algorithm for signal analysis in the context of non intrusive load monitoring', in *Proceedings IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, Beijing, China, 2017, pp. 3615–3620.

[93] K. D. Anderson, M. E. Bergés, A. Ocneanu, D. Benitez, and J. M. F. Moura, 'Event detection for Non Intrusive load monitoring', in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, 2012, pp. 3312–3317.

[94] N. E. Huang, Z. Shen, and S. R. Long, 'A new view of nonlinear water waves: the Hilbert spectrum', *Annual review of fluid mechanics*, vol. 31, no. 1, pp. 417–457, 1999.

[95] P. Flandrin, G. Rilling, and P. Goncalves, 'Empirical mode decomposition as a filter bank', *IEEE signal processing letters*, vol. 11, no. 2, pp. 112–114, 2004.

[96] J.-R. Yeh, J.-S. Shieh, and N. E. Huang, 'Complementary ensemble empirical mode decomposition: A novel noise enhanced data analysis method', *Advances in adaptive data analysis*, vol. 2, no. 02, pp. 135–156, 2010.

[97] T. Y. Hou, M. P. Yan, and Z. Wu, 'A variant of the EMD method for multi-scale data', *Advances in Adaptive Data Analysis*, vol. 1, no. 04, pp. 483–516, 2009.

[98] Z. Wu and N. E. Huang, 'On the filtering properties of the empirical mode decomposition', *Advances in Adaptive Data Analysis*, vol. 2, no. 04, pp. 397–414, 2010.

[99] G. Rilling and P. Flandrin, *implementation of EEMD in Matlab*. Lyon: Ecole Normale Supérieure de Lyon, 2007.

[100] P. Held, A. Saleh, D. O. Abdeslam, and D. Benyoucef, 'Frequency Invariant Transformation of Periodic Signals (FIT-PS) for high frequency Signal Representation in NILM', *BW-CAR| SINCOM*, p. 1, 2016.

[101] T. Hassan, F. Javed, and N. Arshad, 'An Empirical Investigationof V-I Trajectory Based Load Signatures for Non-Intrusive Load', *Monitoring.IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 870–878, Mar. 2014.

[102] M. Kahl, A. U. Haq, T. Kriechbaumer, and H.-A. Jacobsen, 'A Comprehensive Feature Study for Appliance Recognition on High Frequency Energy Data', in *Proceedings of the Eighth International Conference on Future Energy Systems (e-Energy '17). ACM*, Shatin, Hong Kong, 2017, pp. 121–131.

[103] J. Liang, S. K. K. Ng, G. Kendall, and J. W. M. Cheng, 'Load Signature Study—Part I: Basic Concept, Structure, and Methodology', *IEEE Transactions on Power Delivery*, vol. 25, no. 2, pp. 551–560, Apr. 2010.

[104] S. Lin, L. Zhao, F. Li, Q. Liu, D. Li, and Y. Fu, 'A nonintrusive load identification method for residential applications based on quadratic programming', *Electric Power Systems Research*, vol. 133, pp. 241–248, 2016.

[105] R. F. ADAC, 'System Design and its Application to Mine Hunting Using SAS Imagery', PhD thesis, TU Darmstadt, 2012.

[106] N. Sadeghianpourhamami, J. Ruyssinck, D. Deschrijver, T. Dhaene, and C. Develder, 'Comprehensive feature selection for appliance classification in NILM', *Energy and Buildings*, vol. 151, pp. 98–106, 2017.

[107] M. Dash and H. Liu, 'Feature Selection for Clustering', in *Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, London, UK, UK, 2000, pp. 110–121.

[108] C. E. Shannon, 'A mathematical theory of communication', *Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[109] A. kulkarni and S. M, 'Empirical Mode Decomposition based Feature Extraction Method for the Classification of EEG Signal', *International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC)*, pp. 04–08, Jun. 2016.

[110] A. S. Bouhouras, A. N. Milioudis, and D. P. Labridis, 'Development of distinct load signatures for higher efficiency of NILM algorithms', *Electric Power Systems Research*, vol. 117, pp. 163–171, 2014.

[111] D. Shmilovitz, 'On the Definition of Total Harmonic Distortion and Its Effect on Measurement Interpretation', *IEEE Transactions on Power Delivery*, vol. 20, no. 1, pp. 526–528, Jan. 2005.

[112] H.-K. Min, T. An, S. Lee, and I. Song, 'Non-intrusive Appliance Load Monitoring with Feature Extraction from Higher Order Moments', in *2013 IEEE 6th International Conference on Service-Oriented Computing and Applications*, 2013, pp. 348–350.

[113] H.-H. Chang, K.-L. Lian, Y.-C. Su, and W.-J. Lee, 'Power-spectrum-based wavelet transform for nonintrusive demand monitoring and load identification', *IEEE Transactions on Industry Applications*, vol. 50, no. 3, pp. 2081–2089, 2013.

[114] L. Du, 'Advanced classification and identification of plugged-in electric loads', PhD Thesis, Georgia Institute of Technology, 2013.

[115] H. Y. Lam, G. Fung, and W. Lee, 'A novel method to construct taxonomy electrical appliances based on load signaturesof', *IEEE Transactions on Consumer Electronics*, vol. 53, no. 2, pp. 653–660, 2007.

[116] A. W. Whitney, 'A direct method of nonparametric measurement selection', *IEEE Transactions on Computers*, vol. 100, no. 9, pp. 1100–1103, 1971.

[117] P. Pudil, J. Novovičová, and J. Kittler, 'Floating search methods in feature selection', *Pattern recognition letters*, vol. 15, no. 11, pp. 1119–1125, 1994.

[118] H. Liu and L. Yu, 'Toward integrating feature selection algorithms for classification and clustering', *IEEE Transactions on Knowledge & Data Engineering*, no. 4, pp. 491–502, 2005.

[119] A. Jain and D. Zongker, 'Feature selection: Evaluation, application, and small sample performance', *IEEE transactions on pattern analysis and machine intelligence*, vol. 19, no. 2, pp. 153–158, 1997.

[120] R. Fandos and A. M. Zoubir, 'Optimal feature set for automatic detection and classification of underwater objects in SAS images', *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 3, pp. 454–468, 2010.

[121] B. Kim and D. Landgrebe, 'Prediction of optimal number of features', in *10th Annual International Symposium on Geoscience and Remote Sensing*, 1990, pp. 2393–2396.

[122] MATLAB, *version 9.6.0 (R2019a)*. Natick, Massachusetts: The MathWorks Inc., 2019.

[123] Y. Du, L. Du, B. Lu, R. Harley, and and T. Habetler, 'A review of identification and monitoring methods for electric loads in commercial and residential buildings.', in *Energy Conversion Congress and Exposition (ECCE), 2010 IEEE*, pp. 4527–4533.

[124] J. Z. Kolter and M. J. Johnson, 'REDD: A public data set for energy disaggregation research', in *Workshop on Data Mining Applications in Sustainability (SIGKDD), San Diego, CA*, 2011, vol. 25, pp. 59–62.

[125] K. Anderson, A. Ocneanu, D. Benitez, A. Rowe, and M. Berges, 'BLUED: A Fully Labeled Public Dataset for Event-Based Non-Intrusive Load Monitoring Research', in *Proceedings of the 2nd KDD Workshop on Data Mining Applications in Sustainability (SustKDD)*, Beijing, China, 2012.

[126] J. Kelly and W. Knottenbelt, 'The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes', *Scientific data*, vol. 2, 2015.

[127] M. Kahl, A. U. Haq, T. Kriechbaumer, and H.-A. Jacobsen, 'WHITED-A Worldwide Household and Industry Transient Energy Data Set'.

[128] P. Held, F. Laasch, A. Djaffar Ould, and D. Benyoucef, 'Frequency Invariant Transformation of Periodic Signals (FIT-PS) for Signal Representation in NILM', *Industrial Electronics Society, IECON 2016-42th Annual Conference of the IEEE*, no. 42, 2016.

[129] K. Basu, 'Classifcation techniques for non-intrusive load monitoring and prediction of residential loads', PhD Thesis, 2014.

[130] H. Rashid, N. Batra, and P. Singh, 'Rimor: Towards Identifying Anomalous Appliances in Buildings', in *Proceedings of the 5th Conference on Systems for Built Environments*, New York, NY, USA, 2018, pp. 33–42.

[131] G. Bellala, M. Marwah, M. Arlitt, G. Lyon, and C. E. Bash, 'Towards an Understanding of Campus-scale Power Consumption', in *Proceedings of the Third ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, New York, NY, USA, 2011, pp. 73–78.

[132] P. Arjunan, H. D. Khadilkar, T. Ganu, Z. M. Charbiwala, A. Singh, and P. Singh, 'Multi-User Energy Consumption Monitoring and Anomaly Detection with Partial Context Information', in *Proceedings of the 2Nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, New York, NY, USA, 2015, pp. 35–44.

[133] A. Darvishi, 'Translation invariant approach for measuring similarity of signals', *Journal of advances in Computer Research*, vol. 1, no. 1, pp. 19–27, 2010.

[134] T. K. Moon, 'Similarity methods in signal processing', *IEEE transactions on signal processing*, vol. 44, no. 4, pp. 827–833, 1996.

[135] N. Batra *et al.*, 'NILMTK: an open source toolkit for non-intrusive load monitoring', in *Proceedings of the 5th international conference on Future energy systems*, 2014, pp. 265–276.

[136] J. Kelly and W. Knottenbelt, 'Metadata for energy disaggregation', in *2014 IEEE 38th International Computer Software and Applications Conference Workshops*, 2014, pp. 578–583.

[137] L. Pereira, 'EMD-DF: A data model and file format for energy disaggregation datasets', in *Proceedings of the 4th ACM International Conference on Systems for Energy-Efficient Built Environments*, 2017, p. 52.

[138] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, 'Explaining explanations: An overview of interpretability of machine learning', in *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, 2018, pp. 80–89.

[139] C. Molnar and others, 'Interpretable machine learning: A guide for making black box models explainable', *E-book at< https://christophm. github. io/interpretable-ml-book/>, version dated*, vol. 10, 2018.

[140] S. M. Lundberg and S.-I. Lee, 'A unified approach to interpreting model predictions', in *Advances in Neural Information Processing Systems*, 2017, pp. 4765–4774.

8 Appendices

Appendix 1 NILM signal segmentation Pseudo code..... 143

Appendix 2: Examples of IF curves for various appliances 146

Appendix 3: Sequential Feature selection algorithms and more comparisons 147

Appendix 4: A visual User-Guide for the signal construction GUI..... 149

Appendix 1 NILM signal segmentation Pseudo code

The block diagram of the procedure is shown below.

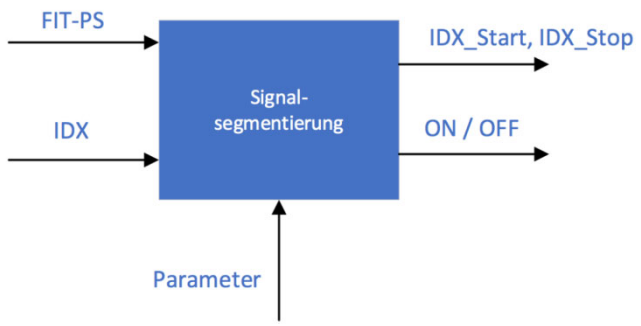


Figure 8.1: inputs and outputs of the Signal Segmentation procedure

Calling the function has the following syntax:

[Idx_start, Idx_stop, Type] = SignalSegmentation(ncFileName);

Function inputs:

- *ncFileNam*: name of the .nc file (has nc format) containing the voltage and current signals after conversion to FIT-PS representation.

Function outputs:

- *Idx _ start* : a vector whose elements are the corrected indices of the events, these indices represent the starts of the transient states in the signal.
- *Idx _ stop* : a vector of the transient states end indices.
- *Type* : events type on/off.

Assumptions:

- We assume that current and voltage signals are stored in a .nc file.

- Events indices Ev_idx, sampling frequency f_s, and grid frequency f_0 are assumed to be available in the .nc file.

8.1 Pseudocode and explanation

```
/* Mainline */

READ v, i, idx, fs

len = 0.2 * fs // default length of left window (samples)

maxTrLen = 4 * fs // max expected transient length of all appliances (samples)

count = 1 // loop counter

WHILE count <= numOfEvents

   // Step 1) cut a window of length "len" from steady state before the event

   DO [vL, iL] = CutLeftWindow(v, i, len)

   // Step 2) cut a total window around the event

   DO [vT, iT] = CutTotalWindow(v, i, maxTrLen)

   // Step 3)

   // find power change around the event

   DO [pBefore, pAfter] = CalculatePowerChange

   // determine event type using power levels before and after it

   IF pBefore < pAfter
```

```
        evType = 1 // on
    ELSE
        evType = 0 // off
    ENDIF
    // Step 4) extract individual appliance signal
    DO appI = SubtractWindows(iT, iL)
    // Step 5) correct event index
    DO idx1 = FindFirstOnEvent(appI)
    // Step 6) calculate power signal and find transient end index
    DO p = CalculatePower(appI)
    DO idx2 = FindTransientEnd(p)
    WRITE idx1, idx2, evType
ENDWHILE
```

Appendix 2: Examples of IF curves for various appliances

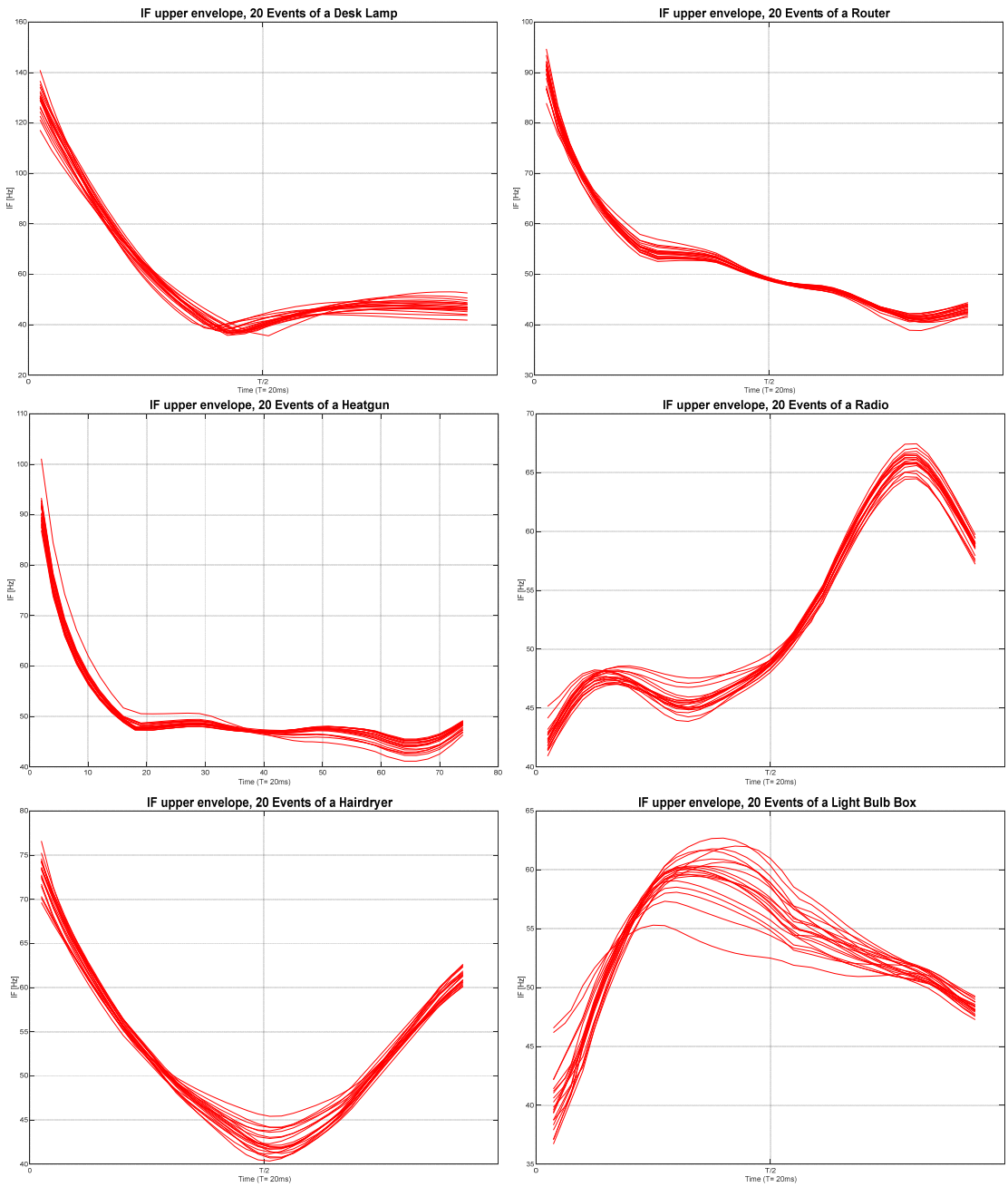


Figure 8.2: IF curves for 6 devices, 20 events for each one; taking first period of Transient state

Appendix 3: Sequential Feature selection algorithms and more comparisons

Let $\boldsymbol{t} = \{t_1, t_2, \ldots, t_N\}$ be the set of all available features, and N is the cardinality of this set, every feature selection algorithm aims to find a subset

$$\boldsymbol{t}_n^* = \{t_1^*, t_2^*, \ldots, t_n^*\}, \qquad 1 \leq n \leq N$$

of features which gives the best performance over any other subset of n features. Let f be a function that represents the classification error, then

$$\boldsymbol{t}^* = arg \min_{\boldsymbol{t}_n^*} \{f(\boldsymbol{t}_n^*)\}, \qquad 1 \leq n \leq N$$

where $f(\boldsymbol{t}_n^*)$ is the classification error rate using the set $\boldsymbol{t}_n^*$.

Sequential forward selection (SFS) algorithm finds the set $\boldsymbol{t}^*$ using the following steps:

SFS Algorithm

1. Find the best feature t_1^*, this feature gives the minimum error rate when the number of selected features is 1, then set $\boldsymbol{t}_1^* = \{t_1^*\}$ where

$$t_1^* = arg \min_{t_j} \{f(t_j)\}, \quad 1 \leq j \leq N$$

2. Iterate for the values of $2 \leq n \leq N$. In each iteration, add one feature t_n^* to the selected set $\boldsymbol{t}_{n-1}^*$. The added feature t_n^* must satisfy the following conditions

$$\begin{cases} t_n^* \notin \boldsymbol{t}_{n-1}^* \\ t_n^* = arg \min_{t_j \notin \boldsymbol{t}_{n-1}^*} \{f(\{\boldsymbol{t}_{n-1}^*, t_j\})\}, 1 \leq j \leq N \end{cases}$$

Then set $\boldsymbol{t}_n^* = \{\boldsymbol{t}_{n-1}^*, t_n^*\}$.

3. Find the best number of features $n^* = arg\,\min_n \{f(\boldsymbol{t}_n^*)\}, \quad 1 \leq n \leq N$

   and set $\boldsymbol{t}^* = \boldsymbol{t}_{n^*}^*$

SFFS Algorithm

Sequential forward floating selection (SFFS) algorithm is similar to SFS algorithm. It only adds an extra step called sequential backward selection (SBS) to SFS to try solve the nesting problem. SFFS finds its best subset of features $\boldsymbol{t}^*$using the following steps:

1. Same as steps 1 in SFS.

2. Same as steps 2 in SFS.

3. SBS: after finding $\boldsymbol{t}_n^*$ from the previous step, check the performance after removing one feature at a time from this set, and find the feature t_n^{**} which satisfies $t_n^{**} = arg\,\min_{t_j \in \boldsymbol{t}_n^*} \{f(\{\boldsymbol{t}_n^* \backslash t_j\})\}$

4. Compare the best value $f(\{\boldsymbol{t}_n^* \backslash t_n^{**}\})$ with the value $f(\{\boldsymbol{t}_{n-1}^*\})$,

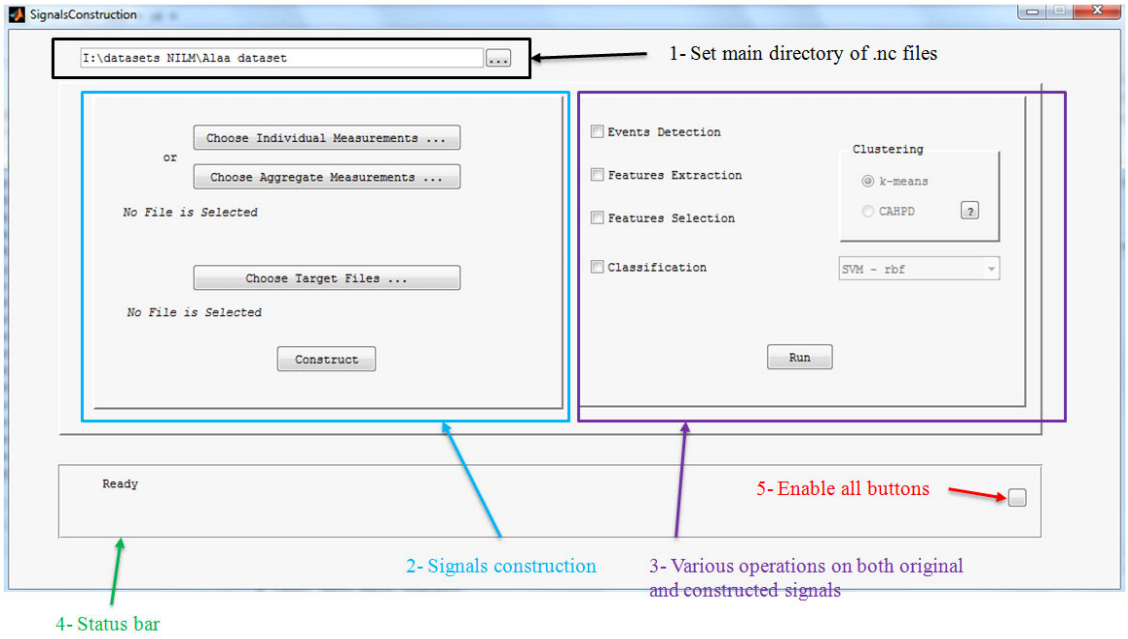
   if $f(\{\boldsymbol{t}_n^* \backslash t_n^{**}\}) > f(\{\boldsymbol{t}_{n-1}^*\})$ then set $\boldsymbol{t}_{n-1}^* = \{\boldsymbol{t}_n^* \backslash t_n^{**}\}$, repeat step 3 for $n-1$

   Otherwise, move to step 2.

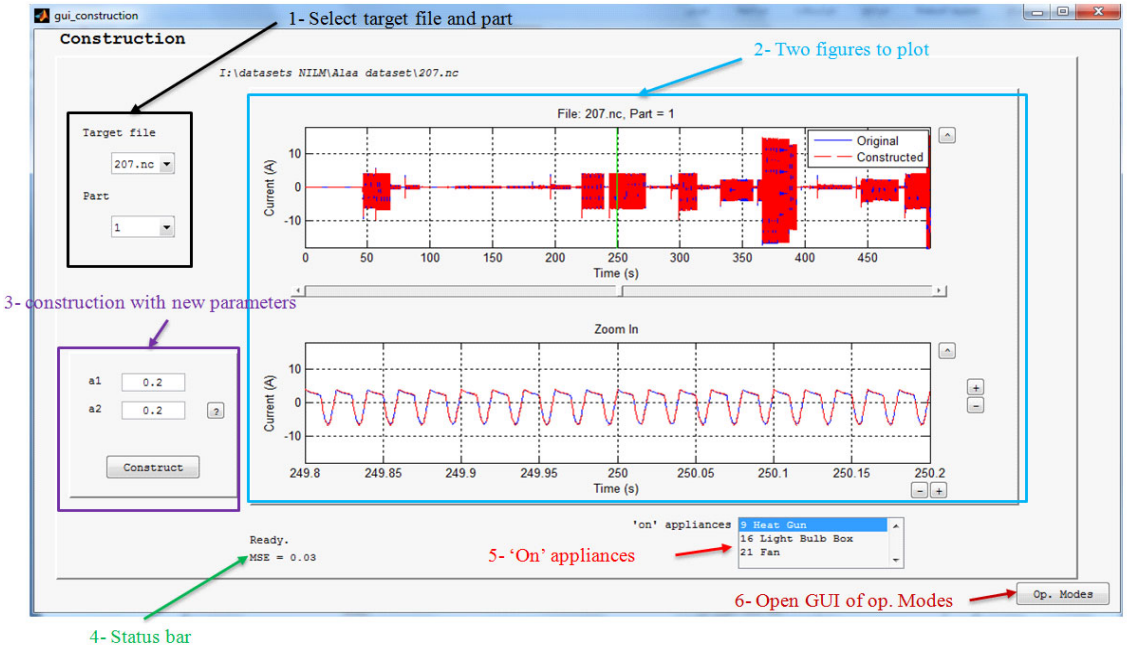
5. Same as step 3 in SFS.

Appendix 4: A visual User-Guide for the signal construction GUI

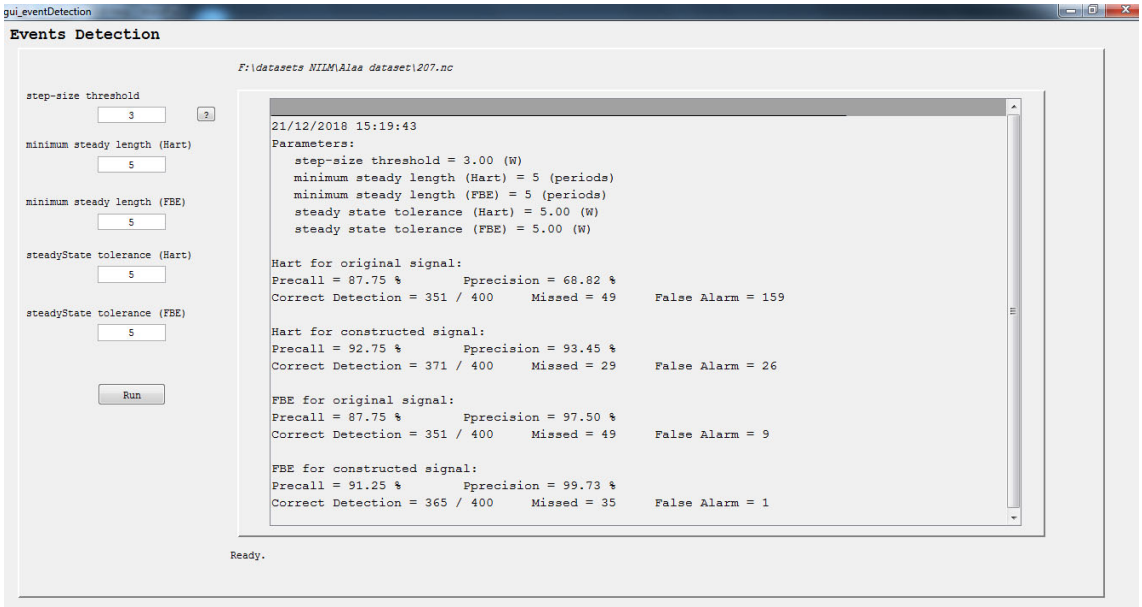
1- Main Interface:



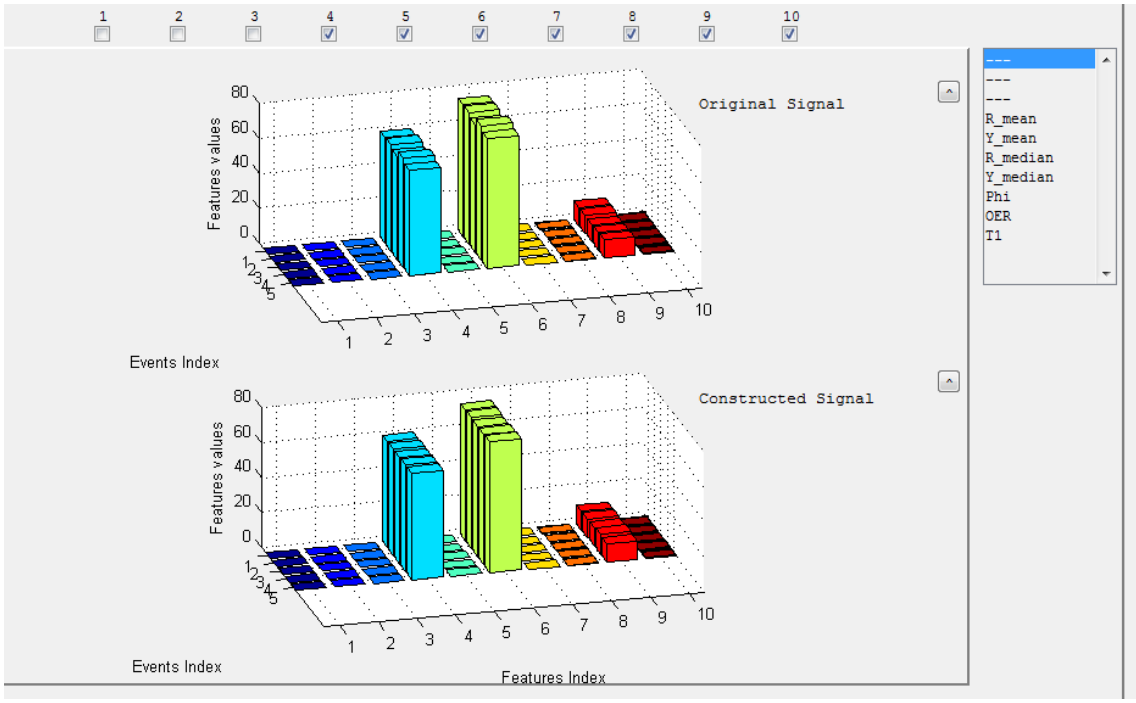
2- Signal construction Interface:



3- Event detection Interface:

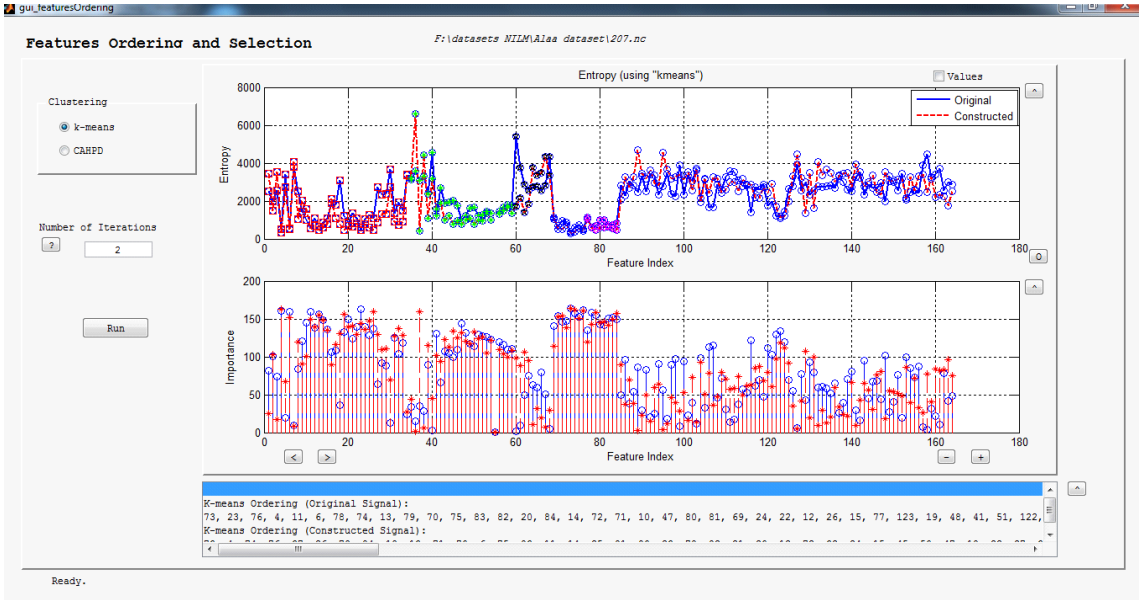


4- Feature extraction Interface:

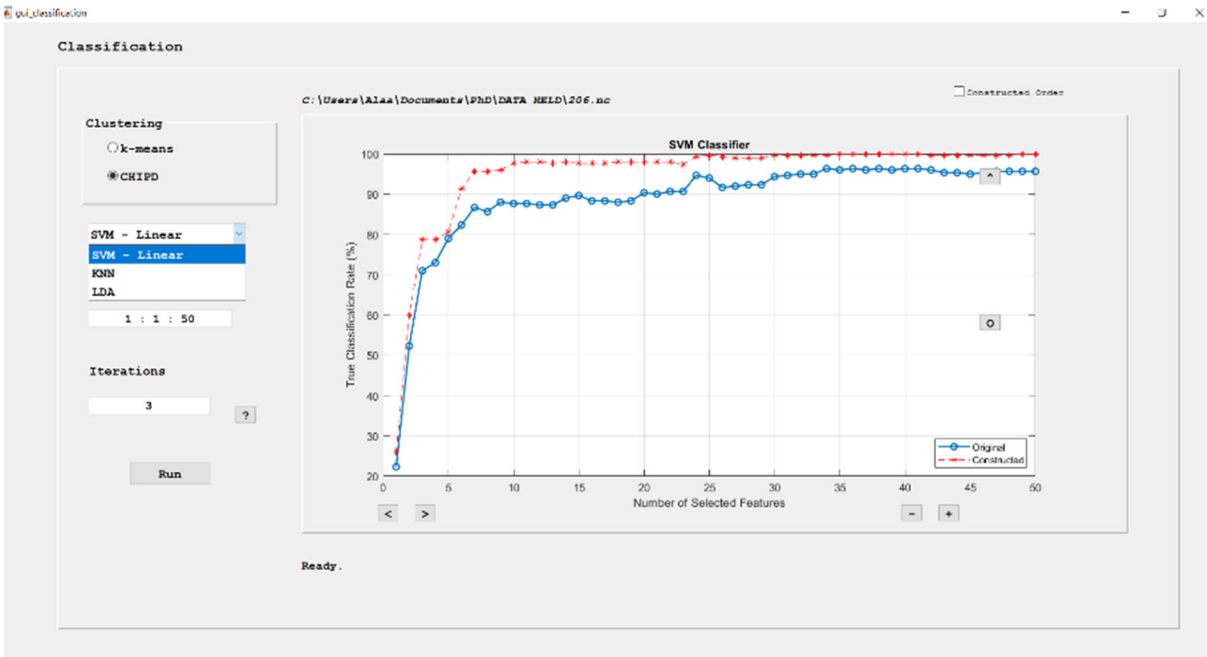


Features 1 to 10 of the appliance ID = 9, but with features 1, 2, and 3 deactivated

4- Feature Selection Interface



5- Classification Interface



List of Abbreviations and Acronyms

NILM	Non-Intrusive Load Monitoring
HHT	Huang-Hilbert Transform
EMD	Empirical Mode Decomposition
EEMD	Ensemble Empirical Mode Decomposition
FBE	Filter-Based EMD inspired algorithm
SD	Standard deviation
NEFSA	NILM Entropy based Feature Selection Algorithm
CHIPD	Clustering based on HIstogram Peaks Detection
FFNN	feed-forward neural network
SVM	support vector machine
STFT	Short Time Fourier Transform
FIT-PS	Frequency Invariant Transformation of periodic Signals
NETCDF	Network Common Data Form