Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell'Ingegneria
Curriculum in Ingegneria Biomedica, Elettronica e delle Telecomunicazioni

Machine Learning approaches for Non-Intrusive Load Monitoring

Ph.D. Dissertation of:
Roberto Bonfigli

Advisor:
Prof. Stefano Squartini

Curriculum Supervisor:
Prof. Francesco Piazza

XVI edition - new series

Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell'Ingegneria
Curriculum in Ingegneria Biomedica, Elettronica e delle Telecomunicazioni

Machine Learning approaches for Non-Intrusive Load Monitoring

Ph.D. Dissertation of:
Roberto Bonfigli

Advisor:
Prof. Stefano Squartini

Curriculum Supervisor:
Prof. Francesco Piazza

XVI edition - new series

Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell'Ingegneria
Facoltà di Ingegneria
Via Brecce Bianche – 60131 Ancona (AN), Italy

Acknowledgments

I acknowledge the CINECA award under the ISCRA initiative, for the availability of high performance computing resources and support, and Netribe Business Solution srl for supporting this research. Last but not the least, I would like to acknowledge my supervisor Prof. Stefano Squartini, for the constant support on my research, and all my colleagues, for this time spent together.

Ancona, Febbraio 2018

Roberto Bonfigli

Abstract

Research on Smart Grids has recently focused on the energy monitoring issue, with the objective to maximize the user consumption awareness in building contexts on one hand, and to provide a detailed description of customer habits to the utilities on the other. One of the hottest topic in this field is represented by Non-Intrusive Load Monitoring (NILM): it refers to those techniques aimed at decomposing the consumption aggregated data acquired at a single point of measurement into the diverse consumption profiles of appliances operating in the electrical system under study.

This work reports an up-to-date state of the art of most promising NILM methods, with an overview of the public available dataset used on purpose and the list of all the evaluation metrics used in this research field. Within all the proposed methods, the Hidden Markov Model (HMM) based and the Deep Neural Network (DNN) based ones have been detected as the most performing and most interesting from the future improvement point of view. In this work, one method for each category has been selected and the performance improvement achieved are described.

In the HMM based approaches, the Additive Factorial Approximate MAP (AFAMAP) algorithm has shown outstanding capabilities and, therefore, it is nowadays regarded as a reference model. In this work, the AFAMAP algorithm has been extended, by means of a differential forward model, thus complementing the existing differential backward model. Furthermore, an aggregated data examination method has been employed, aimed to the detection of inadmissible working state combinations of appliances, as well as the constraints setting based on the reactive power disaggregation feedback. In a second step, an alternative formulation of the same algorithm is presented, in order to deal with Additive Factorial Hidden Markov Models (FHMM) framework based on bivariate HMM, whose emitted symbols are the joint active-reactive power signals. The experiments are conducted on the AMPds dataset, in noised and denoised conditions. Additionally, a user-aided footprint extraction procedure is presented as a facilitated procedure, in order to obtain a clean footprint from the aggregated power signal in real scenario.

In the DNN based approaches, the Denoising Autoencoder (dAE) represents one of the most performing approaches. In this work, this method is extended and improved by conducting a detailed study on the topology of the network,

and by intelligently recombining the disaggregated output with a median filter. An exhaustive comparative evaluation is conducted with respect to the AFAMAP algorithm. The experiments have been conducted on the AMPds, UK-DALE, and REDD datasets in seen and unseen scenarios both in presence and in absence of noise. Furthermore, the same method is explored when the input size is increased, including the reactive power component near the active power consumption.

Finally, similar computational intelligence techniques are applied in other field, i.e. the smart water and gas grid, and audio application.

Contents

List of Figures

List of Tables

List of Tables

Chapter 1

Introduction

In the recent years, the public awareness on energy saving themes has been constantly increasing. Indeed, the consequences of global warming are now tangible and studies have demonstrated that they are directly related to humans activities and their inefficient use of energy and natural resources [2, 3, 4]. The response of governments and public institutions to counteract this trend is to promote policies for reducing energy waste and intelligently use natural resources. The electricity grid is a key component in this scenario: the original electromechanical grid, where the information flow was one-directional, is transforming into the new digital *smart grid* [5] where the information flows from the energy provider to distributed sensors and generator stations and vice-versa. Part of this change involves the integration of smart meters in the grid in order to provide detailed consumption information both to the consumers and to the energy provider.

Indeed, recent studies demonstrated that this fine-grained information is able to provide significant energy savings [6]. On the consumers side, the knowledge of the energy consumption of individual appliances establishes a virtuous behaviour towards a wiser use of electric energy [7, 8]. Studies showed that this can lead to savings greater than 12% with specific appliance feedback and personalised recommendations [6, 9, 10, 11, 12]. On the energy provider side, fine-grained information enables the prediction of the power demand, the application of management policies and the prevention of overloading or blackouts over the energy network [13].

Providing detailed consumption information without installing several dedicated meters requires intelligent methods able to infer the energy consumed by individual appliances with minimal metering points. Non-intrusive load monitoring (NILM) denotes the class of methods and algorithms able to perform this task by using the electrical parameters measured in a single-point [6, 14, 15]. Originally developed in the seminal work by Hart [16], NILM has been an active area of research in the last years. The most promising approaches recently presented in the literature are based on machine learning algorithms, and their general scheme consists in extracting significant features from the measured

electrical parameters and then estimating the appliance specific active power signal by using a supervised or unsupervised algorithm [14, 17].

As aforementioned, machine learning techniques have become a popular choice for NILM, since they showed significant disaggregation performance: in particular Hidden Markov models (HMMs) [18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30] and Neural Networks (NN) [31, 32, 33, 34, 35], despite other approaches as graph-based signal processing [36], Support Vector Machines (SVM) [37], k-Nearest Neighbours [37], and Decision Trees [38] have been successfully employed for NILM. This dissertation is focused on the first two categories.

The majority of the approaches employ the active power (P_a) consumption, but other signals can be also effectively used, in order to have a better representation of the electric load, such as reactive power (P_r). This dissertation is focused on the exploitation and the integration of the reactive component of the power consumption within the approaches under study, in order to improve their performance.

Additionally, similar techniques based on computational intelligence approaches are exploited in other fields, as smart water and gas grid, and audio application.

The outline of the dissertation is the following. In Chapter 2, the NILM is introduced, with an update state of the art of the approaches in literature and the dataset publicly available for the experiments. Chapter 3 describes the fundamental notion on the Hidden Markov Model e Neural Network paradigm, entering in details for the models parameters meaning and the training algorithm for their estimation. The details of the proposed disaggregation algorithm are presented in Chapter 4 and Chapter 5, respectively, AFAMAP [22] and the denoising Auto Encoder [32]. In both chapters, the improvement of the method and the experimental setup are described, with a discussion on the related results. For both the approaches, the integration of the reactive power component has been proposed. Furthermore, advancement in the fields of smart water and gas grid, and audio application are presented in Chapter 6. Finally, Chapter 7 concludes this dissertation and presents future developments.

Chapter 2

Non-Intrusive Load Monitoring

The issues relating to the energy conservation and efficiency have gained a role of great importance, both from the point of view of the consumer and the energy provider. Furthermore, over the years, the infrastructures for energy distribution have undergone an ageing process, which have led to the study of the possibility in smart grids implementation, in which a set of information from detection and network management systems can be transmitted in addition to energy [39, 40].

Useful information, about the characteristics and operating behaviour of an electrical system, can be obtained by means of the power consumption analysis, in order to predict the power demand (load forecasting), to apply management policies and to avoid overloading or blackouts over the energy network. Similarly, from the user perspective, the lifestyle of the people in a house can be predicted by the energy consumption analysis, allowing to implement policies for advantageous time tariffs [41].

Over the years, several studies have demonstrated that the energy consumption awareness (i.e., which appliances are operating at a certain time instant and how much electrical power they are consuming) influences the user behaviour [7]. Specifically, the awareness conducts to moderate energy consumption, resulting in monetary savings and reduction of the energy required to the provider. Furthermore, applying this consideration to commercial or industrial environments, it may provide larger energy saving [42].

In the struggle to improve the energy efficiency of residential environments, the availability of information about the appliances in use can support automated optimization approaches [43, 44].

Load monitoring has become a challenging problem, and several techniques have been studied to solve it. This work is focused on Non-Intrusive Load Monitoring (NILM) algorithms, introduced by [16], which aim to separate the aggregated energy consumption signal, measured in a single centralized point, in the individual signals from each appliance, using a simple hardware but smart software algorithms. This solution replaces a distributed smart socket grid inside the house, resulting in lower implementation costs and less invasive

solutions for the end user.

2.1 Problem statement

The NILM problem can be formulated as follows: let $y(t)$ be the aggregated signal measured at the time index t. Without lack of generality, here it is supposed that $y(t)$ represents the active power. $y(t)$ can be expressed as the sum of the active power contributions of each appliance:

$$y(t) = \sum_{i=1}^{N} y_i(t) + e(t), \tag{2.1}$$

where N is the number of appliances, $y_i(t)$ is the individual contribution of appliance i, and $e(t)$ is a noise term. The NILM problem is, thus, the task of finding the individual appliance contributions $y_i(t)$ given only the aggregated measurement $y(t)$. In a *denoised* scenario [28], the term $e(t)$ is zero, while in a *noised* scenario $e(t)$ can comprise both measurement noise and the contributions of other appliances (e.g., unknown or always-on appliances). The noise term can be treated as a single additional appliance or as an actual noise contribution.

The NILM is classified as a *Blind Source Separation* (BSS) problem. Specifically, it is categorized as a single-channel overcomplete BSS, since the signals, i.e. the power consumptions, flows through the electric line from the multiple loads to the unique sensor, i.e. the smart meter. In the case of analysing the active power consumption, the meter samples the aggregate current flowing in the electric line, and multiplying it with the voltage values, which is approximately a fixed value, it allows to calculate the aggregate power consumption. In order to exploit the reactive power data, both current and voltage have to be sampled in the electric line, in order to recover the phase between them, which is the crucial information for the reactive power calculation. The introduction of the second meter allows to reach an higher level of representation of the problem, which reverse in a more accurate disaggregation results.

2.2 State of the Art

This section presents an overview of the recent literature on NILM.

Several approaches are proposed in the literature, which could be gathered in two main categories, as discussed in [45] (Figure 2.1): *load classification* and *source separation*. In the former, the disaggregation is achieved by a first step of signature detection, which corresponds to the activation of a specific appliance, and a second step of event classification by means of appliance model,

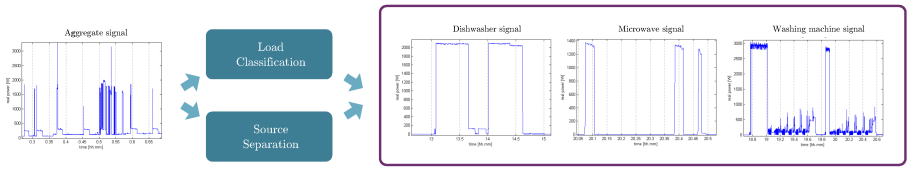


Figure 2.1: NILM paradigm: the overall power load, given as input, is disaggregated in output signals, each one representing an appliance contribution (i.e., dishwasher, microwave and washing machine).

previously trained over some training data. In the latter approach, the disaggregation is achieved by recovering the source signals, which in this case correspond to the electrical consumption of each appliance in the network.

For the load disaggregation purpose, defining how the specific appliance in the circuit can be identified within the aggregated signal is fundamental: for this reason a *signature* is defined as a particular trait over the aggregated signal that can be associated to a specific appliance, which can be exploited to permit the disaggregation goal.

For different application, signature is defined in different ways. Two main signature categories can be found in the literature: *steady-state* and *transient* signature. The former [16, 18, 22, 25, 26, 19, 32, 33, 34, 23, 36, 27] relates to changing operation state of the appliance (i.e., when an appliance is turned on/off), which is reflected on the power characteristics: the value of power measurement is stable in time until the appliance changes operation state, thus this kind of signature can be captured with low frequency sampling (respectively in the order of Hz). Nevertheless, low resolution may results unsuitable if rapid state changes occur. The latter [46, 14, 17, 47, 48, 49, 35, 50, 51] is based on the transient phenomena between steady-states: high frequency noise in electrical current or voltage, as a result of an appliance changing operation state, can be exploited to recognize the different appliances. For this purpose, an high sampling rate is required (respectively in the order of kHz), with a more complex and costly hardware equipment [17]. This explains why the scientific community devoted particular attention to steady state approaches.

The necessity of the user intervention for creating appliance models distinguishes supervised from unsupervised approaches [52]. The first implies the availability of the individual signals of each appliance. In a real operating scenario, this translates into requiring support by the user, that should sequentially switch on the appliance of interest and switch off the remaining [16]. In this dissertation, this requirement has been partially reduced by allowing selected appliances (e.g., the fridge) to remain operational while signatures of the other appliances are being created, as described in Section 4.4. The latter have been the preferred choice in the literature, since they represent the

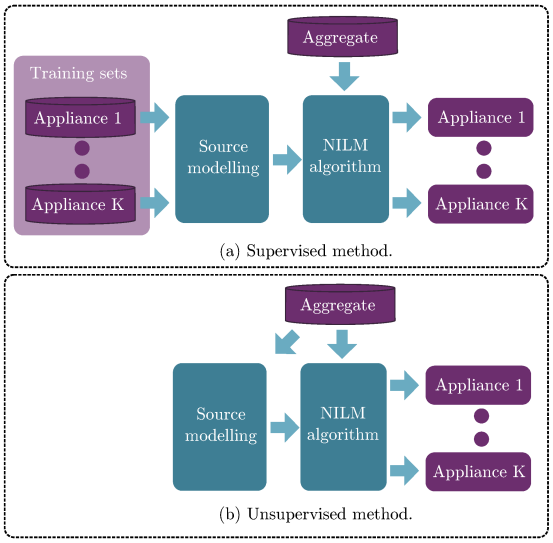


Figure 2.2: Comparison of supervised (a) and unsupervised (b) method.

most convenient approach for end-users. Unsupervised techniques provide the means to automate the learning process, thus being completely transparent to the user. Furthermore, they are capable of dynamically adapting to the power system changes over time (i.e., addition, removal, or substitution of appliance) [53]. However, their major shortcoming is represented by the inability to apply an appropriate label to the disaggregated signals. Different approaches try to overcome these limitations by exploiting the information contained on a generic labelled dataset and generalising to unseen household data by using an unsupervised algorithm [19].

A comparison between the steps required for a supervised and an unsupervised approach is depicted in Figure 2.2: in order to achieve the load disaggregation purpose, for the former approach individual appliance data are necessary to create models used by the NILM algorithm, while for the latter approach no information other than the aggregate data is required. Although various techniques have been already presented in the literature, which obtain reasonable performance, most of them are based on supervised algorithms (i.e., require individual appliance data for model training, prior to the system deployment), thus their functioning depend on the user intervention and the a-priori knowledge of the power system parameters in which they are working. In order to prevent these inconveniences, unsupervised NILM techniques have been developed: these approaches do not require individual appliance data and the models information is captured only using the aggregated load, without the user intervention. Furthermore, the unsupervised approaches are independent

from the number of the appliances forming the aggregated load and capable of dynamically adapt to the power system changes over time (i.e., addition, removal or substitution of appliance).

For a recent review and a taxonomy, please refer to [54, 55, 14, 17]. In [56] different approaches are described, also with an overview over metering equipment for data logging.

Some techniques are publicly available implemented within the NILMTK toolkit [1] and the NILM Eval framework [57].

Among unsupervised approaches, the ones based on FHMMs have been devoted particular attention in the last years. One of the earliest work on the topic has been presented in [18] by Kim and colleagues. The key idea is to model each appliance with independent parallel HMM each contributing to the aggregate power. The framework is assessed by using the steady-state real power signal, but it allows multidimensional features as input. In [19], the authors employ HMMs in a Bayesian framework in order to combine multiple models and form a general model of an appliance. Labelled data are required in the training phase and then appliance specific models are tuned on aggregate data without requiring user intervention. In the literature, particular attention has been devoted to the algorithm proposed by Kolter and Jaakkola [22], since it showed noteworthy performance with a reasonable computational complexity. The Additive Factorial Approximate Maximum a Posteriori (AFAMAP) algorithm is an efficient method, based on an optimization problem, for the inference of the working states combination in the Factorial Hidden Markov Model framework. The authors introduced the AFAMAP algorithm, where they constrain the posterior probability to require only one HMM change state at any given time. Semi-Markov models are combined with Hierarchical Dirichlet Process in [29] for inferring both the state complexity of the models and the duration of the distributions. The authors use the active power as input feature and evaluate the performance on the five most consuming appliances of the REDD dataset [30]. Makonin and colleagues in [28] proposed the sparse Viterbi algorithm for disaggregating the active power online and in real-time. Sparse Viterbi exploits the matrix sparsity in HMMs and it was evaluated on the AMPds [58] and REDD [30] datasets. Aiad and Lee [52] augmented FHMMs with additional chains for modelling possible interactions among the appliances. The algorithm operates on the active power input feature and it was evaluated on the REDD dataset. The work in [23] introduces an FHMM model with unbounded number of chains, and states for each chain as well. In [24] the authors introduce Hierarchical FHMM with the aim of overcoming the device independence assumption and the one-at-time condition. The algorithm operates on the steady-state active power signal by clustering the signals of correlated devices and then by training HMM models on the identified

clusters (denoted as "super devices"). In the disaggregation phase, inference is performed with AFAMAP on the super devices, and the result is mapped back to the original device by using the state relation table learned during the training phase. Compared to the original AFAMAP algorithm on the REDD and Pecan datasets, the method proposed by the authors provides significant performance improvements. Zhong *et al.*, [25] incorporate domain knowledge in the FHMM in the form of signal aggregate constraint. In the NILM scenario, this translates into constraining the total energy consumed in a day by an appliance to be close to a predefined value. The algorithm was assessed on the Household Electricity Survey dataset and compared to the Additive Factorial HMM and the AFAMAP algorithms. The results showed that the method indeed achieves better performance in terms of disaggregation error. In a different work [26], the same authors introduce interleaved factorial non-homogeneous hidden Markov model (IFNHMM), where the transition probabilities of the models are supposed time variant in order to represent the different pattern of usage of an appliance during the day. In addition, at each time step only one chain is allowed to change. The algorithm presented in [27] combine FHMM and Subsequence Dynamic Time Warping (SDTW). The FHMM is employed in the first stage to identify only the ON and OFF state of each appliance. SDTW, then, is applied iteratively to extract the final output. The authors propose both a supervised and semi-supervised version of the algorithm, with the latter employing the aggregate signal and consumption diaries to extract the appliance signatures.

The works presented above perform load disaggregation by using the active power as the only input feature. Differently, in [21], the authors propose a structural variational approximation method and they evaluated the combination of five features: active and reactive power, power factor, and the active and reactive power standard deviation calculated in a window of five samples. The algorithm is evaluated in a "denoised scenario", for different combinations of low-power appliances (e.g., laptop, desk lamp, LCD monitor). Instead of using only electrical parameters, in [59] the authors proposed the inclusion of contextual information represented by the timing-usage statistics and the presence of the user in the house. The disaggregation algorithm is based on AFAMAP and Conditional FHMMs, and the experiments are conducted on the Tracebase dataset augmented with synthetic contextual information.

Among the techniques appeared in the literature, Deep Neural Networks (DNN) have been devoting particular attention in the last years, since they exhibited noteworthy performance for load disaggregation [32, 33, 34]. In [33], the authors proposed an approach based on Long Short-Term Memory (LSTM) neural networks [60]. The algorithm consists in training a neural network for each appliance in order to predict a sample of the disaggregated active power

from a segment of aggregated data. Neural networks have been combined with HMMs in [34]: the emission probabilities of the HMM are modelled by a Gaussian distribution for state representing the single load, and by a DNN for state representing the aggregated signal. Similarly to [33], LSTMs have been also employed in [32], this time combined with convolutional layers at the input of the network to extract the features of the signal directly from raw data. In the same paper, NILM is treated also as a noise reduction problem, where the clean signal is represented by the disaggregated appliance profile, and the noise signal by the remaining profiles and the measurement noise. Noise reduction is performed by using a denoising autoencoder (dAE) composed of convolutional and fully connected layers that estimates the appliance profile from the aggregated noisy signal. An additional approach proposed in [32] uses a neural network that estimates the start time, the end time, and the mean power demand of each appliance. In the experiments conducted by the authors on the UK recording Domestic Appliance-Level Electricity dataset (UK-DALE) [61], they demonstrated that the most performing approach is represented by the dAE network, that outperformed both the other DNN architectures, and the FHMM method proposed in [30].

A different approach has been proposed in [62], where the algorithm employs motif mining to identify recurring events. In particular, based on the a-priori knowledge of the number of devices, it operates by firstly removing the appliances that are always on. Then, it identifies the steady-states power levels with a Dirichlet process Gaussian Mixture Model, and it detects repetitive sequences of power level changes. The probabilistic sequential mining stage discovers devices with several sequential power levels. The algorithm operates by firstly clustering power levels according to time of day and day of the week. Finally, the motif mining stage finds repetitive episodes in the time series. On average, the results obtained on the REDD dataset showed a superior performance with respect to the AFAMAP algorithm [22]. In [36], the authors propose a graph signal processing (GSP) approach that do not require training data. The GSP paradigm is employed for event detection, clustering and feature matching.

Although in the majority of the approaches the active power (P_a) consumption is employed, other signals can be also effectively used, in order to have a better representation of the electric load, such as reactive power (P_r) consumption, current (I) and voltage (V) signal. Beside using the raw signal, better performance can be achieved introducing a feature extraction stage, in order to represent information at an higher level: different kind of ensemble averages (i.e., mean, variance) or the application of transform operator (i.e., Fourier, Wavelet, ST, Hilbert) are the main features employed. In addition, other quantities can be extracted to represent specific information about the appliance usage, such as cycling frequency and temporal duration usage, or in-

Table 2.1: Comparison of household energy data sets.

Contribution	Dataset	Location	Duration per house	Number of houses	Appliance sample resolution	Aggregate sample resolution
[30]	REDD	USA	3-19 days	6	3 sec	1 sec & 15 kHz
[63]	BLUED	USA	8 days	1	transition label	12 kHz
[64]	UMass Smart	USA	3 months	3	1 sec	1 sec
[65]	Tracebase	DE	N/A	15	1-10 sec	N/A
[66]	Pecan Street	USA	7 days	10	1 min	1 min
[67]	HES	UK	1 or 12 months	251	2 or 10 min	2 or 10 min
[58]	AMPds	CDN	1 year	1	1 min	1 min
[68]	iAWE	IND	73 days	1	1 or 6 sec	1 sec
[61]	UK-DALE	UK	3-17 months	4	6 sec	1-6 sec & 16 kHz
[69]	GreenD	AT/IT	1 year	9	1 sec	1 sec
[70]	COMBED	IND	18 months	8	30 sec	30 sec
[57]	ECO	CH	8 months	6	1 sec	1 sec
[71]	BERDS	USA	1 year	N/A	20 sec	20 sec
[72]	SustData	PT	5 years	50	50 Hz	50 Hz

dicator representative of the appliance electric circuit, such as current/voltage harmonic distortion.

2.3 Datasets

Every problem to be solved with machine learning and data mining techniques requires the availability of data for algorithm parametrization: the ability to access public dataset, representative of a real scenario, allows to test the approaches, in order to evaluate the effective benefit in real applications, and to compare the performance of existing approaches on a common comparison basis. In order to evaluate the effectiveness of the algorithms and the performance about the disaggregation task, both aggregate and appliance specific data, which represent the ground truth, are required.

Comparison between the datasets, highlighting their main characteristics, such as duration, number of houses and signal sampling frequency is shown in Table 2.1. This comparative table is an extension of the proposed one in [1], with an update considering the recent datasets published in the last year.

From a geographic point of view, in most cases the datasets are recorded in USA, with some examples for European countries (i.e., Germany, United Kingdom, Austria, Italy, Switzerland and Portugal), besides Canada and India. The recording coming from different country could lead to mismatching between electric quantity (i.e., the RMS voltage value is 220 V in Europe and 110 V in USA), thus attention needs to be paid when different datasets are used in the same system development. It can be noticed that the consumption recordings last several days or few months for many contributions. Nevertheless, several datasets contain recordings one or more years long: in these cases is possible to study the human behaviour over a long time, comprising the effect of seasonal changes on consumption. In addition, only in [67, 72] an high number of houses is present, which lead to studies about power circuit behaviour

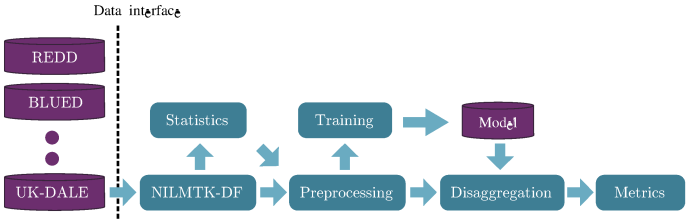


Figure 2.3: The processing pipeline of NILMTK. Courtesy of Batra *et al.* [1].

in different households. Regarding the sampling frequency of the aggregate data and specific appliance signals, there is a common trend about using a sampling interval between 1 sec and 1 min. Only in [30, 63, 73] an higher sampling frequency, in order of kHz, is used, which allows the development of more sophisticated algorithm: the availability of an higher data resolution allows to examine transient phenomena, which can be used for a more complete description of the problem.

In [1] an open source toolkit is presented, called *NILMTK*, useful to evaluate NILM algorithms in a simple way over different datasets. The toolkit contains a data importer for each dataset, a set of preprocessing and statistics functions, a list of some disaggregation algorithms and a set of metrics to evaluate the performance of such algorithms. The complete processing pipeline is reported in Figure 2.3.

2.4 Evaluation metrics

The metrics chosen for the performance evaluation have to represent both the aspects of the disaggregation problem: the classification of the switching activity of the appliances and the accuracy of the disaggregated profiles compared to the ground truth appliance consumption [73].

In order to evaluate both aspects of the NILM problem, algorithms have been evaluated by using the following metrics:

- Energy-based Precision ($P^{(E)}$), Recall ($R^{(E)}$), and F_1-Measure ($F_1^{(E)}$) [22];
- Normalized Disaggregation Error (NDE) [22];
- Normalised Error in Assigned Power (NEP) [73];
- State-based Precision ($P^{(S)}$), Recall ($R^{(S)}$), F_1-Measure ($F_1^{(S)}$);
- Matthews Correlation Coefficient (MCC) [73, 74].

Energy-based Recall measures the part of the power consumption that has been correctly classified, whereas the Precision measures the amount of power assigned to an appliance that actually belongs to it. Considering the i-th appliance, $P_i^{(E)}$ and $R_i^{(E)}$ are calculated as follows:

$$P_i^{(E)} = \frac{\sum_{t=1}^{T} \min\left(\hat{y}_i(t), y_i(t)\right)}{\sum_{t=1}^{T} \hat{y}_i(t)}, \quad R_i^{(E)} = \frac{\sum_{t=1}^{T} \min\left(\hat{y}_i(t), y_i(t)\right)}{\sum_{t=1}^{T} y_i(t)}, \tag{2.2}$$

where $\hat{y}_i(t)$ is the disaggregated power consumption signal, $y_i(t)$ is the ground truth appliance power consumption signal, and T is the total number of samples. In order to evaluate the total performance of the disaggregation algorithm, the metric average across the appliances is computed as follows:

$$P^{(E)} = \frac{1}{N} \sum_{i=1}^{N} P_i^{(E)}, \quad R^{(E)} = \frac{1}{N} \sum_{i=1}^{N} R_i^{(E)}. \tag{2.3}$$

The F_1-Measure is calculated as the geometric mean between Precision and Recall:

$$F_1^{(E)} = 2 \frac{P^{(E)} R^{(E)}}{P^{(E)} + R^{(E)}}. \tag{2.4}$$

The Normalized Disaggregation Error (NDE) [22] provides a direct measure of the ability of the algorithm of reconstructing the active power profiles, and it is defined as:

$$NDE = \sqrt{\frac{\sum_{t=1}^{T} \sum_{i=1}^{N} \left(y_i(t) - \hat{y}_i(t)\right)^2}{\sum_{t=1}^{T} \sum_{i=1}^{N} \left(\hat{y}_i(t)\right)^2}}. \tag{2.5}$$

The Normalised Error in Assigned Power (NEP) measures the deviation of the estimated power $\hat{y}_i(t)$ from the true power $y_i(t)$ normalised by the total energy consumption of the appliance. Considering appliance i, NEP is calculated as follows:

$$\mathrm{NEP}_i = \frac{\sum_{t=1}^{T} |y_i(t) - \hat{y}_i(t)|}{\sum_{t=1}^{T} y_i(t)}. \tag{2.6}$$

State-based metrics are defined based on the actual and predicted state of an appliance. More in details, considering appliance i, true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN) are defined as

follows:

$$TP_i = \sum_{t=1}^{T} \text{AND}(x_i(t) = \text{on}, \hat{x}_i(t) = \text{on}), \tag{2.7}$$

$$FP_i = \sum_{t=1}^{T} \text{AND}(x_i(t) = \text{off}, \hat{x}_i(t) = \text{on}), \tag{2.8}$$

$$FN_i = \sum_{t=1}^{T} \text{AND}(x_i(t) = \text{on}, \hat{x}_i(t) = \text{off}), \tag{2.9}$$

$$TN_i = \sum_{t=1}^{T} \text{AND}(x_i(t) = \text{off}, \hat{x}_i(t) = \text{off}), \tag{2.10}$$

where $x_i(t)$ and $\hat{x}_i(t)$ are respectively the actual and the predicted state of appliance i at the time index t. Appliance i is considered in the "on" state if $y_i(t)$ exceeds a predefined threshold. Generally, the threshold varies with the appliance and it assumes the same value used for extracting the activations within the ground truth power consumption [32]. State-based Precision and Recall are defined as:

$$P_i^{(S)} = \frac{TP_i}{TP_i + FP_i}, \quad R_i^{(S)} = \frac{TP_i}{TP_i + FN_i}, \tag{2.11}$$

In the case of multi-state models, e.g. HMM or FSM, the *state based* metric, considers the ability of the system to infer the exact state of evolution of each HMM in the model: for the i-th appliance, the multiclass confusion matrix is built by comparing, for each time instant $t = 1, 2, \ldots, T$, the disaggregation variables $\boldsymbol{\xi_t^{(i)}}$ value assumed in the problem solution, with the exact evolution state $x_t^{(i)}$, defined as the *ground truth*. Each class corresponds to a state $j = 1, \ldots, m_i$ of the i-th HMM. Since that the values in $\boldsymbol{\xi_t^{(i)}}$ are not-integral, the computed confusion matrix is soft weighted, similar to the fuzzy-logic [75]. For each class, the Precision $P_i^{(j)}$ and Recall $R_i^{(j)}$ are computed, then the average between the classes evaluates the medium performance for each HMM:

$$P_i^{(S)} = \frac{1}{m_i} \sum_{j=1}^{m_i} P_i^{(j)}, \quad R_i^{(S)} = \frac{1}{m_i} \sum_{j=1}^{m_i} R_i^{(j)}. \tag{2.12}$$

Finally, state-based F_1-Measure is given by:

$$F_1^{(S)} = \frac{2P^{(S)}R^{(S)}}{P^{(S)} + R^{(S)}}, \quad \text{with} \quad P^{(S)} = \frac{1}{N} \sum_{i=1}^{N} P_i^{(S)}, \; R^{(S)} = \frac{1}{N} \sum_{i=1}^{N} R_i^{(S)}. \tag{2.13}$$

The Matthews Correlation Coefficient is defined as:

$$\mathrm{MCC}_i = \frac{TP_i TN_i - FP_i FN_i}{\sqrt{(TP_i + FP_i)(TP_i + FN_i)(TN_i + FP_i)(TN_i + FN_i)}}, \tag{2.14}$$

and

$$\mathrm{MCC} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{MCC}_i. \tag{2.15}$$

MCC assumes values in the range $[-1, 1]$, with $+1$ representing perfect prediction, 0 random prediction, and -1 total disagreement between the ground truth and the prediction.

In the case of the metrics are evaluated for a signal window w_f with $f = 1, 2, \ldots, F$, the metrics are averaged over the windows, since the performance are evaluated over the entire dataset:

$$P_i^{\{S,E\}} = \frac{1}{F} \sum_{f=1}^{F} P_i^{\{S_f, E_f\}}, \quad R_i^{\{S,E\}} = \frac{1}{F} \sum_{f=1}^{F} R_i^{\{S_f, E_f\}}. \tag{2.16}$$

2.5 Remarks

Regarding the datasets, the difference among the many appeared in the literature is highlighted, in terms of amount of recorded data, number of houses and sampling frequency. All these parameters, together with the characteristics of the available smart meter providing the aggregate consumption data in the operating scenario, strongly influence the choice of the NILM technique and therefore the dataset for algorithm design and optimization must be carefully selected.

Chapter 3

Background

The Computers are able to perform complex calculus operations in a short amount of time. However computers cannot compete with humans in dealing with: common sense, ability to recognize people, objects, sounds, comprehension of natural language, ability to learn, categorize, generalize.

Therefore, why does the human brain show to be superior w.r.t common computers for these kind of problems? Is there any chance to mimic the mechanisms characterizing the way of working of our brain in order to produce more efficient machines?

In the field of signal analysis, the aim is the characterization of such real-world signals in terms of *signal models*, which can provide the basis for a theoretical description of a signal processing system. They are potentially capable of letting us learn a great deal about the signal source, without having to have the source available.

Therefore, in this chapter two family of modelling technique are described, i.e., the Hidden Markov Models (HMM) and the Deep Neural Network (DNN). After a theoretical description, the algorithms used for their parameter estimation are described, with a focus on the most widely model structure used in the field of the NILM.

3.1 Hidden Markov Model (HMM)

Within the multiple technique available, there are several possible choices for what type of signal model is used for characterizing the properties of a given signal. The most widely used categorization gather the methods in deterministic models and statistical models. In this chapter, it is interesting to explore the statistical models, which try to characterize only the statistical properties of the signal.

The underlying assumption of the statistical model is that the signal can be well characterized as a parametric random process, and that the parameters of the stochastic process can be determined (estimated) in a precise, well-defined manner. One type of stochastic signal model is the *Hidden Markov model*

(HMM). This model is based on some theoretical fundamentals. The treatise followed in this chapter is inspired by [76].

Firstly, a *discrete Markov process* need to be introduced. It is a system which may be described at any time as being in one of a set of N distinct states, $S_1, S_2, \ldots, S_N$. The time instants associated with state changes are defined as $t = 1, 2, \ldots$, while the actual state at time t as q_t.

In a discrete, first order, Markov chain, this probabilistic description is truncated to just the current and the predecessor state:

$$P[q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, \ldots] = P[q_t = S_j | q_{t-1} = S_i]. \tag{3.1}$$

In those processes, the right-hand side of the equation is independent of time. Additionally, a set of state transition probabilities a_{ij} is defined in the form:

$$a_{ij} = P[q_t = S_j | q_{t-1} = S_i] \text{ for } 1 \leq i, j \leq N. \tag{3.2}$$

The state transition coefficients follow the properties:

$$\sum_{j=1}^{N} a_{ij} = 1 \text{ with } a_{ij} \geq 0 \tag{3.3}$$

since they obey to standard stochastic constraints.

The notation to denote the initial state probabilities is the following:

$$\pi_i = P[q_1 = S_i] \text{ for } 1 \leq i \leq N \tag{3.4}$$

The model defined above is classified ad an *observable Markov chain*, since the output of the process is the set of states at each instant of time. The extension to *Hidden Markov models* (HMM) introduces the fundamental that the observation is a probabilistic function of the state, i.e., the resulting model (which is called a hidden Markov model) is a doubly embedded stochastic process with an underlying stochastic process that is not observable (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observations.

Therefore, the elements which constitute an HMM are the following:

- N, the number of states in the model. Generally the states are interconnected in such a way that any state can be reached from any other state (e.g., an ergodic model) individual states as $S = \{S_1, S_2, \ldots, S_N\}$, and the state at time t as q_t.

- M, the number of distinct observation symbols per state, i.e., the discrete alphabet size. The individual symbols is denoted as $V = \{v_1, v_2, \ldots, v_M\}$.

- The state transition probability distribution $A = \{a_{ij}\}$, where:

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i] \text{ for } 1 \leq i, j \leq N \tag{3.5}$$

  For the special case where any state can reach any other state in a single step, we have $a_{ij} > 0$ for all i, j. For other types of HMMs, we would have $a_{ij} = 0$ for one or more (i, j) pairs.

- The observation symbol probability distribution in state j, $B = \{b_j(k)\}$, where:

$$b_j(k) = P[v_k \text{ at } t | q_t = S_j] \text{ for } 1 \leq j \leq N, 1 \leq k \leq M \tag{3.6}$$

- The initial state distribution $\pi = \{\pi_i\}$, where:

$$\pi_i = P[q_1 = S_i] \text{ for } 1 \leq i \leq N \tag{3.7}$$

The HMM can be used as a generator to give an observation sequence:

$$O = O_1, O_2, \ldots, O_T \tag{3.8}$$

where each observation O_t is one of the symbols from V, and T is the number of observations in the sequence.

A complete specification of an HMM requires the definition of two model parameters (N and M), specification of observation symbols, and of the three probability measures A, B, and π

$$\lambda = (A, B, \pi). \tag{3.9}$$

The probability of the observation sequence, $O = O_1, O_2, \ldots, O_T$, given the model λ, i.e., $P(O|\lambda)$ for the state sequence $Q = q_1, q_2, \ldots, q_T$ is defined as:

$$P(O|Q, \lambda) = \prod_{t=1}^{T} P(O_t | q_t, \lambda) = \tag{3.10}$$

$$= b_{q_1}(O_1) b_{q_2}(O_2) \cdots b_{q_T}(O_T) \tag{3.11}$$

in which it is assumed the statistical independence of observations.

The probability of such a state sequence Q is defined as:

$$P(Q|\lambda) = \pi_{q_1} \, a_{q_1 q_2} \, a_{q_2 q_3} \cdots a_{q_{T-1} q_T}. \tag{3.12}$$

Therefore, the joint probability of O and Q is:

$$P(O, Q|\lambda) = P(O|Q, \lambda)\, P(Q|\lambda) \tag{3.13}$$

The probability of O (given the model) is obtained by summing this joint probability over all possible state sequences q:

$$P(O|\lambda) = \sum_{allQ} P(O|Q, \lambda)\, P(Q|\lambda) = \tag{3.14}$$

$$= \sum_{q_1, q_2, \ldots, q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \cdots a_{q_{T-1} q_T} b_{q_T}(O_T) \tag{3.15}$$

and an efficient procedure to solve the problem is the *Forward-Backward* procedure.

The forward variable $\alpha_t(i)$ is defined as:

$$\alpha_t(i) = P(O_1 O_2 \cdots O_t, q_t = S_i|\lambda) \tag{3.16}$$

and the probability of the partial observation sequence, $O_1 O_2 \cdots O_t$ (until time t) and state S_i at time t, given the model λ, is solved exploiting $\alpha_t(i)$ inductively, following the procedure:

1. Initialization:
$$\alpha_1(i) = \pi_i b_i(O_1) \text{ for } 1 \leq i \leq N. \tag{3.17}$$

2. Induction:
$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}) \text{ for } 1 \leq t \leq T-1, 1 \leq j \leq N. \tag{3.18}$$

3. Termination:
$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i). \tag{3.19}$$

On the other hand, a backward variable $\beta_t(i)$ is defined as:

$$\beta_t(i) = P(O_{t+1} O_{t+2} \cdots O_T|q_t = S_i, \lambda) \tag{3.20}$$

and the probability of the partial observation sequence from $t+1$ to the end, given state S_i at time t and the model λ, is solved exploiting the $\beta_t(i)$ inductively, following the procedure:

1. Initialization:
$$\beta_T(i) = 1 \text{ for } 1 \leq i \leq N. \tag{3.21}$$

2. Induction:

$$\beta_t(i) = \sum_{i=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \text{ for } t = T-1, T-2, \ldots, 1, 1 \leq i \leq N \tag{3.22}$$

Solution to Problem 2: how do we choose a corresponding state sequence

Additionally, finding the *optimal* state sequence $Q = q_1, q_2, \ldots, q_T$ associated with the given observation sequence is defined exploiting several possible optimality criteria.

The variable $\gamma_t(i)$ defines the probability of being in state S_i at time t, given the observation sequence O, and the model λ:

$$\gamma_t(i) = P(q_t = S_i | O, \lambda) = \tag{3.23}$$

$$= \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} \tag{3.24}$$

where $\alpha_t(i)$ accounts for the partial observation sequence $O_1 O_2 \cdots O_t$ and state S_i at t, while $\beta_t(i)$ accounts for the remainder of the observation sequence $O_{t+1} O_{t+2} \cdots O_T$ given state S_i at t. The normalization factor $P(O|\lambda) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i)$ makes $\sum_{i=1}^{N} \gamma_t(i) = 1$.

The most widely used criterion is to find the *single* best state sequence (path) $Q = \{q_1 q_2 \cdots q_T\}$ for the given observation sequence $O = \{O_1 O_2 \cdots O_T\}$, i.e., to maximize $P(Q|O, \lambda)$ which is equivalent to maximizing $P(Q, O|\lambda)$. This criterion is satisfied by the *Viterbi algorithm*. The quantity $\delta_t(i)$ is defined as the best score (highest probability) along a single path, at time t, which accounts for the first t observations and ends in state S_i:

$$\delta_t(i) = \max_{q1, q2, \cdots, q_{t-1}} P[q_1 q_2 \cdots q_t = i, O_1 O_2 \cdots O_t | \lambda], \tag{3.25}$$

and, by induction:

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] b_j(O_{t+1}) \tag{3.26}$$

To actually retrieve the state sequence, we need to keep track of the argument which maximized $\delta_{t+1}(j)$, for each t and j, via the array $\psi_t(j)$. The procedure follows the steps:

1. Initialization:

$$\delta_1(i) = \pi_i b_i(O_1) \text{ for } 1 \leq i \leq N \tag{3.27}$$

$$\psi_1(i) = 0 \text{ for } 1 \leq i \leq N \tag{3.28}$$

2. Recursion:

$$\delta_t(j) = \max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}] b_j(O_t) \text{ for } 2 \le t \le T, 1 \le j \le N \tag{3.29}$$

$$\psi_t(j) = \arg \max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}] \text{ for } 2 \le t \le T, 1 \le j \le N \tag{3.30}$$

3. Termination:

$$P^* = \max_{1 \le i \le N} [\delta_T(i)] \tag{3.31}$$

$$q_T^* = \arg \max_{1 \le i \le N} [\delta_T(i)] \tag{3.32}$$

4. Path (state sequence) backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \text{ for } t = T-1, T-2, \ldots, 1 \tag{3.33}$$

3.1.1 Baum-Welch algorithm

Finally, a method to adjust the model parameters (A, B, π) to maximize the probability of the observation sequence given the model $P(O|\lambda)$ is defined. Given any finite observation sequence as training data, there is no optimal way of estimating the model parameters. One solution is to choose $\lambda = (A, B, \pi)$ such that $P(O|\lambda)$ is locally maximized using an iterative procedure such as the *Baum-Welch* method.

The variable $\xi_t(i, j)$ is defined as the probability of being in state S_i at time t, and state S_i, at time $t + 1$, given the model and the observation sequence:

$$\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda) = \tag{3.34}$$

$$= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)} = \tag{3.35}$$

$$= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \tag{3.36}$$

where the numerator term is just $P(q_t = S_i, q_{t+1} = S_j, O|\lambda)$ and the division by $P(O|\lambda)$ gives the desired probability measure.

Since $\sum_{t=1}^{T-1} \gamma_t(i)$ represents the expected number of transitions from S_i, and $\sum_{t=1}^{T-1} \xi_t(i, j)$ the expected number of transitions from S_i to S_j, the two variable $\gamma_t(i)$ and $\xi_t(i, j)$ are related by summing over j:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i, j). \tag{3.37}$$

Using the concept of counting event occurrences, the estimated parameters

are defined as follow:

$$\overline{\pi}_i = \gamma_1(i) \tag{3.38}$$

$$\overline{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{3.39}$$

$$\overline{b}_j(k) = \frac{\sum_{t=1,s.t.O_t=v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \tag{3.40}$$

The model $\overline{\lambda}$ is more likely than the model λ in the sense that $P(O|\overline{\lambda}) > P(O|\lambda)$, i.e., we have found a new model $\overline{\lambda}$ from which the observation sequence is more likely to have been produced. If $\overline{\lambda}$ is iteratively used in the place of λ and repeat the reestimation calculation, the probability of O being observed from the model can improved, until some limiting point is reached. The final result of this reestimation procedure is called a *maximum likelihood* estimate of the HMM. It has to be highlighted that the forward-backward algorithm leads to local maxima only.

The Baum-Welch reestimation equations are essentially identical to the EM steps for this particular problem, and the stochastic constraints of the HMM parameters are automatically satisfied at each iteration:

$$\sum_{i=1}^{N} \overline{\pi}_i = 1 \tag{3.41}$$

$$\sum_{j=1}^{N} \overline{a}_{ij} = 1 \text{ for } 1 \leq i \leq N \tag{3.42}$$

$$\sum_{k=1}^{M} \overline{b}_j(k) = 1 \text{ for } 1 \leq j \leq N \tag{3.43}$$

3.1.2 Factorial HMM

In an HMM, information about the past is conveyed through a single discrete variable, e.g., the hidden state. A generalization of HMMs in which this state is factored into multiple state variables and is therefore represented in a distributed manner.

An HMM encodes information about the history of a time series in the value of a single multinomial variable, e.g., the hidden state, which can take on one of K discrete values. This multinomial assumption supports an efficient parameter estimation algorithm, the Baum-Welch algorithm, which considers each of the K settings of the hidden state at each time step.

An HMM with a *distributed* state representation let the model automatically decompose the state space into features that decouple the dynamics of the

process that generated the data, therefore the task of modelling time series that are known a priori to be generated from an interaction of multiple, loosely-coupled processes.

The treatise followed in this chapter is inspired by [77].

The generalization of the HMM state representation let the state be represented by a collection of state variables:

$$S_t = S_t^{(1)}, \ldots, S_t^{(m)}, \ldots, S_t^{(M)} \tag{3.44}$$

where M is the number of underlying distributed variables, each of which can take on $K^{(m)}$ values.

This model is defined as *Factorial Hidden Markov model* (FHMM), as the state space consists of the cross product of these state variables. The number of state combination is equal to $\prod_{m=1}^{M} K^{(m)}$.

A natural structure to consider is one in which each state variable evolves according to its own dynamics, and is *a priori* uncoupled from the other state variables:

$$P(S_t|S_{t-1}) = \prod_{m=1}^{M} P(S_t^{(m)}|S_{t-1}^{(m)}) \tag{3.45}$$

The observation at time step t can depend on all the state variables at that time step. For continuous observations, as a linear Gaussian, the observation Y_t is a random vector whose mean is a linear function of the state variables. Representing the state variables as $K \times 1$ vectors, where each of the K discrete values corresponds to a 1 in one position and 0 elsewhere. The probability density for a $D \times 1$ observation vector Y_t:

$$P(Y_t|S_t) = |C|^{-\frac{1}{2}}(2\pi)^{-\frac{D}{2}} \exp\left\{-\frac{1}{2}(Y_t - \mu_t)'C^{-1}(Y_t - \mu_t)\right\}, \tag{3.46}$$

where

$$\mu_t = \sum_{m=1}^{M} W^{(m)} S_t^{(m)}. \tag{3.47}$$

Each $W^{(m)}$ matrix is a $D \times K$ matrix whose columns are the contributions to the means for each of the settings of $S_t^{(m)}$, C is the $D \times D$ covariance matrix, $'$ denotes matrix transpose, and $|\cdot|$ is the matrix determinant operator.

The inference problem consists of computing the probabilities of the hidden variables given the observations. This problem can be solved efficiently via the forward-backward algorithm. In some cases, it is desirable to infer the single most probable hidden state sequence. This can be achieved via the Viterbi algorithm.

The learning problem consists of learning the parameters for a given struc-

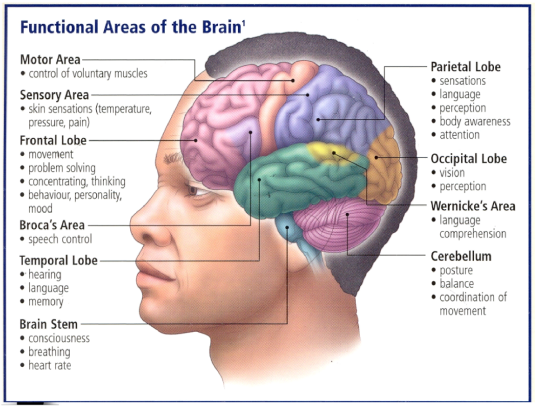


Figure 3.1: The human brain.

ture. The parameters of a factorial HMM can be estimated via the Expectation Maximization (EM) algorithm, which in the case of classical HMMs is known as the Baum-Welch algorithm. This procedure iterates between a step that fixes the current parameters and computes posterior probabilities over the hidden states (the E step) and a step that uses these probabilities to maximize the expected log likelihood of the observations as a function of the parameters (the M step). The exact M step for factorial HMMs is simple and tractable, whilst the exact E step for factorial HMMs is computationally intractable. Rather than computing the exact posterior probabilities, one can approximate them using a Monte Carlo sampling procedure, avoid the sum over exponentially many state patterns at some cost in accuracy. Within many possible sampling schemes, the *Gibbs* sampling is the simplest. A second approach is the *Completely factorized variational* inference, which results to be both tractable and deterministic. A third approximation, the *Structured variational* inference, is both tractable and preserves much of the probabilistic structure of the original system.

3.2 Deep Neural Network (DNN)

A *biological Neural Networks* is a big set of specialized cells (*neurons*) connected among them, which memorize and process information, thus controlling the body activities they belong to.

The *neuron* model is composed of:

- DENDRITE: input terminal
- CELL BODY (Nucleus): processing core

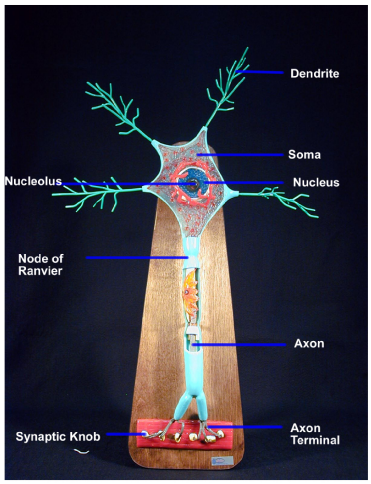


Figure 3.2: The neuron model.

- AXON: output way-out
- SYNAPSES: output terminal (with weight)

The *neuron* properties can be described in:

- LOCAL SIMPLICITY: the neuron receives stimuli (excitation or inhibition) from dendrites and produces an impulse to the axon which is proportional to the weighted sum of the inputs;
- GLOBAL COMPLEXITY: the human brain possess $\mathcal{O}(10^{10})$ neurons, with more than 10K connections each;
- LEARNING: even though the network topology is relatively fixed, the strength of connections (synaptic weights) can change when the network is exposed to external stimuli;
- DISTRIBUTED CONTROL: no centralized control, each neuron reacts only to its own stimuli;
- TOLERANCE TO FAILURES: performance slowly decrease with the increase of failures.

The biological Neural Networks are able to solve very complex tasks in few time instants (like memorization, recognition, association, and so on.)

The *Artificial Neural Networks* (ANNs) are defined as *Massively parallel distributed processors made up of simple processing units having a natural propensity for storing experiential knowledge and making it available for use* (Haykin, 2008).

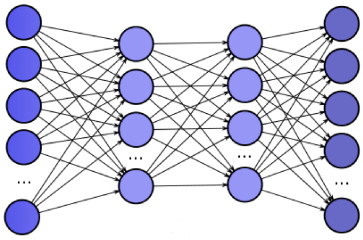


Figure 3.3: The Artificial Neural Network.

An ANN resembles the brain in two aspects:

1. Knowledge is acquired by the network from its environment through a learning process;

2. Synaptic weights are used to store the acquired knowledge.

A *neuron* is an information-processing unit that is fundamental to the operation of a neural network. The model of a neuron is composed of three basic elements of the neural model:

- a *set of synapses*, or connecting links, each of which is characterized by a weight or strength of its own, w_{kj};

- an *adder* for summing the input signals, weighted by the respective synaptic strengths of the neuron; the operations described here constitute a linear combiner;

- an *activation function* for limiting the amplitude of the output of a neuron. Typically, the normalized amplitude range of the output of a neuron is written as the closed unit interval [0,1], or, alternatively, [-1,1].

The neural model also includes an externally applied *bias*, denoted by b_k.

Therefore, the mathematical description of neuron activity can be defined as:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{3.48}$$

$$y_k = \varphi\left(u_k + b_k\right) \tag{3.49}$$

where:

- $x_1, x_2, \cdots, x_m$ are the input signals;

- $w_{k1}, w_{k2}, \cdots, w_{km}$ are the respective synaptic weights of neuron k;

- u_k is the linear combiner output due to the input signals;

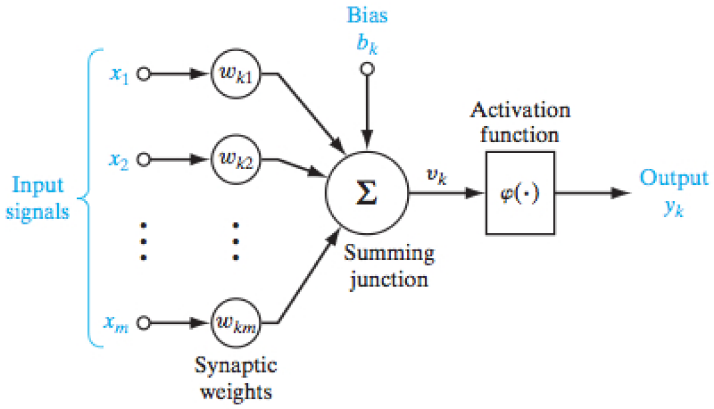


Figure 3.4: The artificial neuron model.

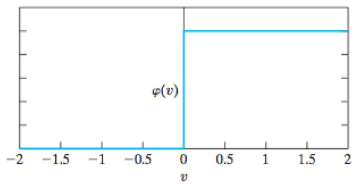


Figure 3.5: The threshold non-linear function.

- b_k is the bias;
- $\varphi(\cdot)$ is the activation function;
- y_k is the output signal of the neuron.

The types of *activation non-linear functions* $\varphi(x)$ are:

- the *threshold function*: in engineering, this form of a threshold function is commonly referred to as a Heaviside function;

$$\varphi(v) = 1 \quad if \quad v \geq 0 \tag{3.50}$$

$$\varphi(v) = 0 \quad if \quad v < 0 \tag{3.51}$$

- the *sigmoid function*: it is defined as a strictly increasing function that exhibits a graceful balance between linear and nonlinear behavior; an example of the sigmoid function is the *logistic function* defined by:

$$\varphi(v) = \frac{1}{1 + exp(-av)} \tag{3.52}$$

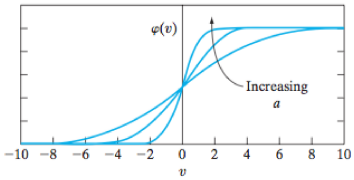


Figure 3.6: The sigmoid non-linear function.

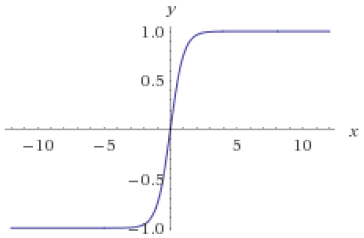


Figure 3.7: The *tanh* non-linear function.

- the *hyperbolic tangent* (*tanh*): it is simply a scaled and shifted version of the sigmoid function:

$$\varphi(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{3.53}$$

- the *Rectifier Linear Unit* (*ReLU*):

$$\varphi(x) = \max(0, x) \tag{3.54}$$

- the $softmax$: it is used on the last layer of a classifier setup: the outputs of the softmax layer represent the probabilities that a sample belongs to the different classes. Indeed, the sum of all the output is equal to 1.

$$\varphi(x_k) = \frac{e^{x_k}}{\sum_{j=1}^{N} e^{x_j}} \text{ for } k = 1, \ldots, K \tag{3.55}$$

The manner in which the neurons of a neural network are structured is intimately linked with the learning algorithm used to train the network. There, the *network architectures* (structures) is defined. In general, two different classes of network architectures are identified:

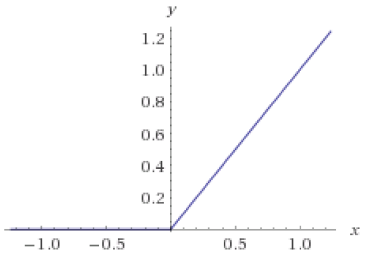


Figure 3.8: The *ReLU* non-linear function.

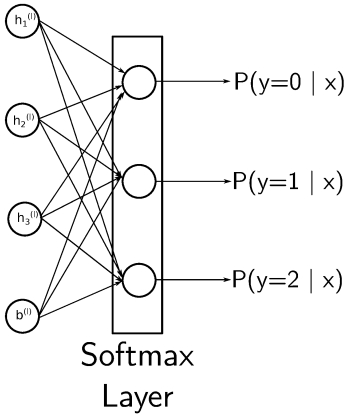


Figure 3.9: The *softmax* layer in a neural network classifier.

1. *Multilayer Feedforward Networks* - (FFNN):

   it is characterized by the presence of one or more hidden layers, whose computation nodes are correspondingly called *hidden neurons* (or hidden units); the term *hidden* refers to the fact that this part of the neural network is not seen directly from either the input or output of the network. The function of hidden neurons is to intervene between the external input and the network output in some useful manner. By adding one or more hidden layers, the network is enabled to extract higher-order statistics from its input.

   The MLP is a well known kind of artificial neural network introduced in 1986 [78]. Each node applies an activation function over the weighted sum of its inputs. The units are arranged in layers, with feed forward connections from one layer to the next. The stochastic gradient descent with error back-propagation algorithm is used for the supervised learning of the network. In the forward pass, input examples are fed to the input

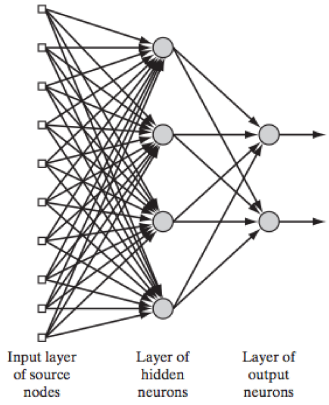


Figure 3.10: The Multilayer Feedforward Network.

layer, and the resulting output is propagated via the hidden layers towards the output layer. At the backward pass, the error signal originating at the output neurons is sent back through the layers and the network parameters (i.e., weights and biases) are tuned.

A single neuron can be formally described as:

$$g(\mathbf{u}[n]) = \varphi\left(\sum_{j=1}^{D} w_j u_j[n] + b\right), \tag{3.56}$$

where $\mathbf{u}[n] \in \mathbb{R}^{D\times 1}$, the bias b is an externally applied term and $\varphi(\cdot)$ is the non-linear activation function. Thus, the mathematical description of a one-hidden-layer MLP is a function $\mathbf{f} : \mathbb{R}^D \to \mathbb{R}^{D'}$, where D' is the size of the output vector, so:

$$\mathbf{f}(\mathbf{u}[n]) = \varphi\left(\mathbf{b}_2 + \mathbf{W}_2\left(\varphi\left(\mathbf{b}_1 + \mathbf{W}_2 \cdot \mathbf{u}[n]\right)\right)\right), \tag{3.57}$$

where $\mathbf{W}_i$ and $\mathbf{b}_i$ are the respective synaptic weights matrix and the bias vector of the i-th layer. The behaviour of this architecture is parametrized by the connection weights, which are adapted during the supervised network training.

2. *Convolutional Neural Networks*(CNN)

   Convolutional neural networks are feedforward neural networks similar to multilayer perceptron, with some special layers.

   Convolution kernels process the input data matrix by dividing it in *local receptive fields*, a region of the same size of the kernel, and sliding the

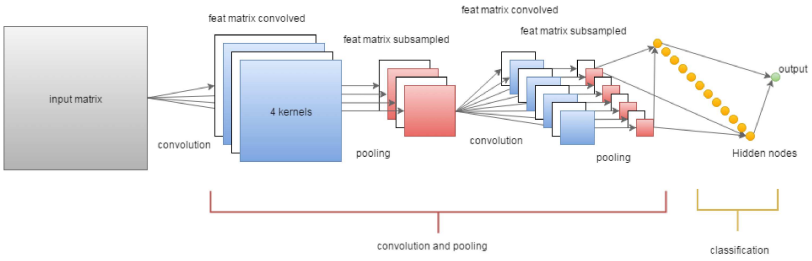


Figure 3.11: The Convolutional Neural Network.

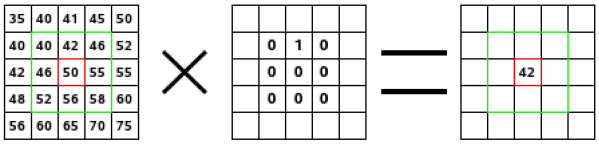


Figure 3.12: The convolution operation.

local receptive field across the entire input. Each hidden neuron is thus connected to a local receptive field, and all the neurons form a matrix called *feature map*. The weights in each *feature map* are *shared*: all hidden neurons are aimed to detect exactly the same pattern just at different locations in the input image.

The main advantages of this network is the robust pattern recognition system characterized by a strong immunity to pattern shifts.

Pooling layer just reduces the dimension of the matrix by a rule: a submatrix of the input is selected, and the output is the maximum value of this submatrix.

The pooling process introduces tolerance against shifts of the input patterns. Together with convolution layer it allows the CNN to detect if a particular event occurs, regardless its deformation or its position.

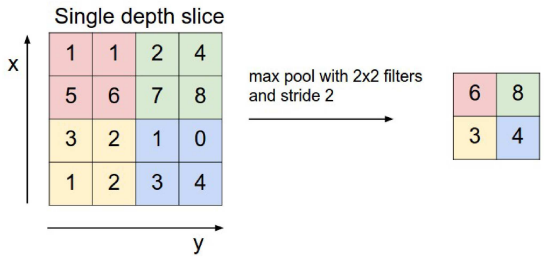


Figure 3.13: The max-pooling layer.

CNN is a feed-forward neural network [79] usually composed of three types of layers: convolutional layers, pooling layers and layers of neurons. The convolutional layer performs the mathematical operation of convolution between a multi-dimensional input and a fixed-size kernel. Successively, a non-linearity is applied element-wise. The kernels are generally small compared to the input, allowing CNNs to process large inputs with few trainable parameters. Successively, a pooling layer is usually applied, in order to reduce the feature map dimensions. One of the most used is the *max-pooling* whose aim is to introduce robustness against translations of the input patterns. Finally, at the top of the network, a layer of neurons is applied. This layer does not differ from MLP, being composed by a set of activation and being fully connected with the previous layer. For clarity, the units contained in this layer will be referred as *Hidden Nodes* (HN).

Denoting with $\mathbf{W}_m \in \mathbb{R}^{K_{1m} \times K_{2m}}$ the m-th kernel and with $\mathbf{b}_m \in \mathbb{R}^{D_1 \times D_2}$ the bias vector of a generic convolutional layer, the m-th feature map $\mathbf{h}_m \in \mathbb{R}^{D_1 \times D_2}$ is given by:

$$\mathbf{h}_m = \varphi\left(\sum_{d=1}^{D_3} \mathbf{W}_m * \mathbf{u}_d + \mathbf{b}_m\right), \tag{3.58}$$

where $*$ represent the convolution operation, and $\mathbf{u}_d \in \mathbb{R}^{D_1 \times D_2}$ is a matrix of the three-dimensional input tensor $\mathbf{u} \in \mathbb{R}^{D_1 \times D_2 \times D_3}$. The dimension of the m-th feature map $\mathbf{h}_m$ depends on the zero padding of the input tensor: here, padding is performed in order to preserve the dimension of the input, i.e., $\mathbf{h}_m \in \mathbb{R}^{D_1 \times D_2}$. Please note that for the sake of simplicity, the time frame index n has been omitted. Commonly, (3.58) is followed by a pooling layer in order to be more robust against patterns shifts in the processed data, e.g. a max-pooling operator that calculates the maximum over a $P_1 \times P_2$ matrix is employed.

A *Deep Learning* definition: *A class of machine learning techniques that exploit many layers of non-linear information processing for supervised or unsupervised feature extraction and transformation, and for pattern analysis and classification.* Artificial Neural Networks are often referred as deep when they have more than 1 or 2 hidden layers.

3.2.1 Stochastic gradient descent (SGD)

Most deep learning training algorithms involve optimization of some sort. The most widely used is the gradient based optimization, which belongs to the first order type.

Optimization is the task of either minimizing some function $f(x)$ by altering x: $f(x)$ is called *objective function*, but in the case when it has to be minimized, it is also call the *cost function*, *loss function*, or *error function*. The aim of the optimization is reached doing small change ϵ in the input x, to obtain the corresponding change in the output $f(x)$:

$$f(x + \epsilon) \approx f(x) + \epsilon f'(x). \tag{3.59}$$

This formulation is based on the calculation of the derivative $f'(x)$. The *gradient descent* is the technique based on the reduction of $f(x)$ by moving x in small steps with the opposite sign of the derivative. The aim is to find the minimum of the cost function: when $f'(x) = 0$, the derivative provides no information about which direction to move, therefore this point is defined as stationary points. A local minimum is a point where $f(x)$ is lower than at all neighbouring and it is no longer possible to decrease $f(x)$ by making infinitesimal steps. The absolute lowest value of $f(x)$ is a *global minimum*.

For the concept of minimization to make sense, there must still be only one (scalar) output. For functions that have multiple inputs $f : \mathbb{R}^n \to \mathbb{R}$, the concept of *partial derivatives* is introduced. The gradient $\nabla_{\mathbf{x}} f(\mathbf{x})$ is the vector containing all the partial derivatives.

The method of *steepest descent* or *gradient descent* states that decrease f by moving in the direction of the negative gradient.

$$\mathbf{x'} = \mathbf{x} - \epsilon \nabla_{\mathbf{x}} f(\mathbf{x}), \tag{3.60}$$

where ϵ is the *learning rate*, a positive scalar determining the size of the step.

Large training sets are necessary for good generalization, but large training sets are also more computationally expensive. The cost function decomposes as a sum over training example of per-example loss function: i.e., the negative conditional log-likelihood of the training data is defined as:

$$J(\theta) = \mathbb{E}(L(\mathbf{x}, y, \theta)) = \frac{1}{m} \sum_{i=1}^{m} L(\mathbf{x}^{(i)}, y^{(i)}, \theta), \tag{3.61}$$

where L is the per-example loss $L(\mathbf{x}, y, \theta) = -\log p(y|\mathbf{x}; \theta)$. The gradient descent requires computing:

$$\nabla_\theta J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta L(\mathbf{x}^{(i)}, y^{(i)}, \theta). \tag{3.62}$$

The computational cost of this operation is proportional to the number of example m, therefore as the training set size grows the time to take a single

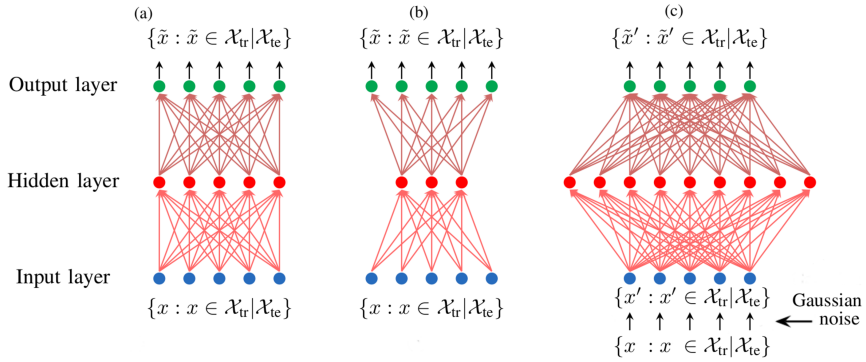


Figure 3.14: The different types of Autoencoders.

gradient step becomes prohibitively long.

Stochastic gradient descent (SGD) is an extension of the gradient descent algorithm: the insight is that the gradient is an expectation estimated using a small set of samples. On each step of the algorithm, a sample of example $\mathbb{B} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m')}\}$, called *minibatch*, is drawn uniformly from the training set. The minibatch size m' is typically chosen to be a relatively small number of examples. The estimate of the gradient is: $\mathbf{g} = \frac{1}{m'} \nabla_\theta \sum_{i=1}^{m'} L(\mathbf{x}^{(i)}, y^{(i)}, \theta)$ using examples from the minibatch $\mathbb{B}$. The SGD algorithm then follows the estimated gradient downhill:

$$\theta \leftarrow \theta - \epsilon\, \mathbf{g} \tag{3.63}$$

where ϵ is the learning rate.

3.2.2 Autoencoder

An Autoencoder is a kind of neural network typically consisting of only one hidden layer, trained to set the target values to be equal to the inputs.

$$\tilde{x} = f(W_2 h(x) + b_2) \tag{3.64}$$

Given an input set of examples $\mathcal{X}$, autoencoder training consists in finding parameters $\theta = \{W_1, W_2, b_1, b_2\}$ that minimize the Reconstruction Error:

$$\mathcal{J}(\theta) = \sum_{x \in \mathcal{X}} \|x - \tilde{x}\|^2 \tag{3.65}$$

Defining M the number of hidden units, and N the number of input units, output units, features size:

- (a): $M = N \rightarrow$ Basic Autoencoder (AE);

- (b): $M < N \rightarrow$ Compression Autoencoder (CAE);
- (c): $M > N$ and Gaussian Noise $\rightarrow$ Denoising Autoencoder (DAE);

Basic Autoencoder

A basic AE – a kind of neural network typically consisting of only one hidden layer –, sets the target values to be equal to the input. It is used to find common data representation from the input [80, 81]. Formally, in response to an input example $x \in \mathbf{R}^n$, the hidden representation $h(x) \in \mathbf{R}^m$ is

$$h(x) = f(W_1 x + b_1), \tag{3.66}$$

where $f(z)$ is a non-linear activation function, typically a logistic sigmoid function $f(z) = 1/(1+\exp(-z))$ applied component-wisely, $W_1 \in \mathbf{R}^{m \times n}$ is a weight matrix, and $b_1 \in \mathbf{R}^m$ is a bias vector.

The network output maps the hidden representation h back to a reconstruction $\tilde{x} \in \mathbf{R}^n$:

$$\tilde{x} = f(W_2 h(x) + b_2), \tag{3.67}$$

where $W_2 \in \mathbf{R}^{n \times m}$ is a weight matrix, and $b_2 \in \mathbf{R}^n$ is a bias vector.

Given an input set of examples $\mathcal{X}$, AE training consists in finding parameters $\theta = \{W_1, W_2, b_1, b_2\}$ that minimise the reconstruction error, which corresponds to minimising the following objective function:

$$\mathcal{J}(\theta) = \sum_{x \in \mathcal{X}} \|x - \tilde{x}\|^2 . \tag{3.68}$$

The minimisation is usually realised by stochastic gradient descent as in the training of neural networks. The structure of the AE is given in Figure 3.14a.

Compression Autoencoder

In the case of having the number of hidden units m smaller than the number of input units n, the network is forced to learn a compressed representation of the input. For example, if some of the input features are correlated, then this compression autoencoder (CAE) is able to learn those correlations and reconstruct the input data from a compressed representation. The structure of the CAE is given in Figure 3.14b.

De-noising Autoencoder

The de-noising AE (DAE) [82] forces the hidden layer to retrieve more robust features and prevent it from simply learning the identity. In such a configuration the AE is trained to reconstruct the original input from a corrupted

version of it. Formally, the initial input x is corrupted by means of additive isotropic Gaussian noise in order to obtain: $x'|x \sim N(x, \sigma^2 I)$. The corrupted input x' is then mapped, as with the AE, to a hidden representation

$$h(x') = f(W_1' x' + b_1'), \tag{3.69}$$

from which the original signal is reconstructed as follows:

$$\tilde{x}' = f(W_2' x + b_2'). \tag{3.70}$$

The parameters $\theta' = \{W_1', W_2', b_1', b_2'\}$ are trained to minimise the average reconstruction error over the training set, to have $\tilde{x}'$ reach as close as possible to the uncorrupted input x, which corresponds to minimising the objective function in Equation 3.65. The structure of the de-noising autoencoder is shown in Figure 3.14c.

Chapter 4

HMM based approach

Approaches based on hidden Markov models (HMMs) have been devoted particular attention in the last years. AFAMAP (Additive Factorial Approximate Maximum a Posteriori) has been introduced in [22] to reduce the computational burden of FHMM. The algorithm bases its operation on additive and difference FHMM, and it constraints the posterior probability to require only one HMM change state at any given time.

Each appliance is modelled as a bivariate HMM, i.e., an HMM whose emitted symbols are represented by active-reactive power pairs. More in details, each HMM is represented by the following parameters [76]:

- the number of states $m \in \mathbb{Z}_+$;
- the hidden states $x \in \{1, 2, \ldots, m\}$;
- the symbols emitted $\boldsymbol{\mu}_j \in \mathbb{R}^n$, where $j = 1, \ldots, s$;
- the symbol emission probability matrix $\boldsymbol{M}^{s \times m}$;
- the state transition probability matrix $\boldsymbol{P} \in [0, 1]^{m \times m}$;
- the starting state probability vector $\boldsymbol{\phi} \in [0, 1]^m$.

In the algorithm, it is assumed that each state of the HMM corresponds to a working state of the appliance, i.e., $x \in \{\mathrm{ON}_1, \mathrm{ON}_2, \ldots, \mathrm{OFF}\}$, so that the number of states m is equal to the number of symbols s and $\boldsymbol{M} \equiv \boldsymbol{I}^{m \times m}$ (*degenerate HMM*). In the proposed approach, $n = 2$ and for the sake of clarity in the remainder of this section it will be omitted since the individual active and reactive power components will be made explicit. For example, each symbol is defined as $\boldsymbol{\mu}_j = [\mu_{a,j}\, \mu_{r,j}]^T$, where the subscripts a and r distinguish the active and reactive components. For each appliance, the quantities to be estimated are the number of states m, the values of $\boldsymbol{\mu}_j$ for each state, the state transition probability matrix $\boldsymbol{P}$, and the starting state probability vector $\boldsymbol{\phi}$. Estimation of m and of $\boldsymbol{\mu}_j$ will be addressed in Subsection 4.1.1.

Regarding the state transition probability matrix $\boldsymbol{P}$, each entry P_{ij} represents the probability of transitioning from state i to state j. Thus, P_{ij} can be estimated with a Maximum Likelihood criterion by calculating the number of times state i transitions to state j and normalising by the total number of transitions from state i. Formally:

$$P_{ij} = \frac{C_{ij}}{\sum_{j'=1}^{m} C_{ij'}}, \tag{4.1}$$

where C_{ij} is the number of transitions from state i to state j. Typically, the greatest values in the matrix are located in the diagonal, meaning that the probability of remaining in the same state is higher compared to the probability of transitioning to another state. As might be expected, the greatest value of the transition matrix is the self-transition probability of the OFF state, since the activation of an appliance occurs after a long time in which it is turned off. In addition, the OFF state corresponds to the initial state, since the footprint starts just before the turning on instant, thus $\boldsymbol{\phi} = [0\,0\cdots 0\,1]^T$.

An example of a four states appliance model is shown in Figure 4.1, where the arc between two states is the probability of transition P_{ij}, while the arc starting e closing on the same state represent the probability P_{ii} of permanence in each state.

Since the pause interval between two footprint is not recorded, the user has to establish the time interval between two appliance activations, e.g., the typical time of use in the daytime or the number of activations per day of the appliance, in order to calculate the OFF interval and to use this value for the calculation of the transition probability related to the OFF state.

A probability value close to zero denotes that the transition is very unlikely. In practice, it is recommended to avoid such low probability values, since evaluating it in the log domain as usually done in the disaggregation algorithm would result in numeric problems. As so, it is recommended to fix the value to a little quantity, e.g.,$\simeq 10^{-5}$.

4.1 Additive Factorial Approximate Maximum A-Posteriori (AFAMAP)

FHMMs have been introduced in [77] as an extension of HMMs to model time series that depend on multiple hidden processes. Starting from the work of Kim and colleagues [18], FHMMs have been largely employed for NILM and several approaches have been proposed in the literature [83, 84, 19, 25, 28, 23]. Among them, AFAMAP [22] represents an effective algorithm able to achieve high performance with a reasonable computational cost.

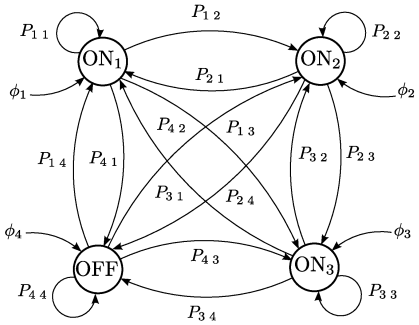


Figure 4.1: An example of a four states HMM.

AFAMAP has been proposed in [22] as an efficient disaggregation algorithm based on FHMMs. In this algorithm, an additional model which relies on the same HMMs composing the Additive FHMM (AFHMM) is introduced. It is based on a differential version of the aggregated signal, resulting in a Differential FHMM (DFHMM). The inference on the set of states of multiple HMMs can be computed through the Maximum A Posteriori (MAP) algorithm and a relaxation towards real values is taken into account, leading to a convex Quadratic Programming (QP) optimisation problem. The disaggregation process is performed by analysing the aggregated power divided in non-overlapping frames.

The reference work [22] describes an unsupervised approach to data disaggregation: in fact, an unsupervised procedure aimed to the extraction of the device load signature is paired with the disaggregation algorithm, referred to as AFAMAP (Additive Factorial Approximate Maximum a Posteriori). In this work, the aim is to investigate and to improve the disaggregation algorithm. Differently to the reference work, however, a supervised approach is used to create the HMMs, based on the circuit level power consumption signature. The signal can be obtained, clearly, from the aggregated data under the condition that the appliances run one at a time [16].

The theoretical approach towards disaggregation is based on the work of Kolter and Jaakkola [22]. In this work the system is modelled relying on Additive Factorial Hidden Markov Model (AFHMM), for which the value of each aggregated power sample corresponds to a combination of working states of the appliances into the system.

Also, in this approach, the assumption that at most one HMM may change its state at any given time is made, which holds true if the sampling time is reasonably short. In this case, the transition on the aggregate power, when moving from a sample to the next, corresponds to the state change of a particular HMM.

Because of that, the differential signal, built from the aggregated power, can be modelled as the result of a Differential Factorial Hidden Markov Model (DFHMM), which relies on the same HMM models composing the AFHMM.

By combining the two models, the inference on the set of states of multiple HMMs can be computed through the Maximum A Posteriori (MAP) technique, which take the form of a Mixed Integer Quadratic Programming (MIQP) optimization problem.

One of the shortcomings of this approach is the non-convex nature of the problem, because of the integer nature of the variables: in this case, a relaxation towards real values is taken into account, leading to a convex Quadratic Programming (QP) optimization problem. Thus, the Additive Factorial Approximate MAP (AFAMAP) approach is obtained.

In a real case scenario, the modelled output may not match with the observed aggregated signal, due to electrical noises, very small loads, or leakages. In that case, the issue is addressed by defining a robust mixture component in both AFHMM and DFHMM, named z_t and Δz_t, respectively.

When a *denoised* scenario [85] is considered, i.e., all the contributions to the aggregated energy demand are known, the robust mixture component is missing. When a *noised* scenario is considered, the robust mixture component is not used, and all the contributions are modelled as an additional appliance represented by the RoW model, which will be introduced in Subsection 4.1.2. This approach provides further advantages, since appliances with lower power consumption values risk to be modelled with working states associated to similar consumption values. This can lead the algorithm to an erroneous assignment of the disaggregation output between similar models. Furthermore, the authors in [22] demonstrated that the disaggregation performance degrades as the number of appliances increases. Thus, representing several appliances with a single model eases the disaggregation task.

In the reference work [22], the parameter n defines the problem dimensionality: in this work, it is assumed $n = 1$, because the algorithm uses only the active power data to characterize the observed aggregated signal.

Specifically, the parameters of the problem follow:

- $N \in \mathbb{Z}_+$ is the number of HMMs in the system;
- $\overline{y}_\tau \in \mathbb{R}$ is the observed aggregated output (in denoised environments $\overline{y}_\tau = \sum_{i=1}^{N} y_\tau^{(i)}$, where $y_\tau^{(i)}$ corresponds to the true appliance output);
- $\sigma^2 \in \mathbb{R}$ is the observation variance.

The differential signal is referred to as $\Delta\overline{y}_{b\tau} = \overline{y}_\tau - \overline{y}_{\tau-1}$.

For the i-th HMM the parameters are:

- $m_i \in \mathbb{Z}_+$ is the number of states;

- $x_\tau^{(i)} \in \{1, \ldots, m_i\}$ is the HMM state at time instant τ ($x_\tau^{(i)} \equiv m_i$ corresponds to the OFF state);
- $\mu_j^{(i)} \in \mathbb{R}$ is the j-th state mean value;
- $\boldsymbol{\phi_b^{(i)}} \in [0,1]^{m_i}$ is the initial states distribution;
- $\boldsymbol{P_b^{(i)}} \in [0,1]^{m_i \times m_i}$ is the transition matrix.

The aggregated signal $\overline{y}_\tau$ is analysed using a windowing technique, where $\tau \in w_f = [(f-1)T + 1, \ldots, fT]$ for $f = 1, 2, \ldots, F$. The window w_f is the timebase for the algorithm and, for convenience, a new temporal variable is introduced by defining the relation $t = \tau - (f-1)T$, for $t = 1, 2, \ldots, T$, with $T \in \mathbb{Z}_+$. After the analysis of all the F windows, the disaggregated signals $\hat{y}_t^{(i)}$ are recomposed using the inverse relation $\tau = t + (f-1)T$.

In the optimization problem, the variables are defined as:

$$\mathcal{Q} = \left\{ \boldsymbol{Q(x_t^{(i)})} \in \mathbb{R}^{m_i}, \boldsymbol{Q(x_{t-1}^{(i)}, x_t^{(i)})} \in \mathbb{R}^{m_i \times m_i} \right\},$$

for which the $Q(x_t^{(i)})_j$ variable is the indicator of the state assumed at time instant t, while the $Q(x_{t-1}^{(i)}, x_t^{(i)})_{j,k}$ variable is the indicator of the state transition from previous to actual time instant, for the i-th HMM.

The AFAMAP algorithm is shown in Figure 4.2.

In (4.2) the error terms are defined as:

$$E_t^{(a)} = \left(\overline{y}_t - \sum_{i=1}^{N} \sum_{j=1}^{m_i} \left\{ \mu_j^{(i)} Q(x_t^{(i)})_j \right\} \right)^2, \tag{4.4}$$

$$E_t^{(bc)} = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ k=1 \\ k \neq j}}^{m_i} \left\{ \left(\Delta \overline{y}_{bt} - \Delta \mu_{k,j}^{(i)} \right)^2 Q(x_{t-1}^{(i)}, x_t^{(i)})_{j,k} \right\}, \tag{4.5}$$

$$E_t^{(bnc)} = D\left(\frac{\Delta \overline{y}_{bt}}{\sigma_2}, \lambda \right) \left(1 - \sum_{i=1}^{N} \sum_{\substack{j=1 \\ k=1 \\ k \neq j}}^{m_i} Q(x_{t-1}^{(i)}, x_t^{(i)})_{j,k} \right). \tag{4.6}$$

The QP optimization problem is defined in the form:

Minimize

$$\frac{1}{2} \boldsymbol{x}^T \boldsymbol{H} \boldsymbol{x} + \boldsymbol{f}^T \boldsymbol{x}, \tag{4.7}$$

subjet to the constraint:

$$\boldsymbol{A_{eq}} \boldsymbol{x} = \boldsymbol{b_{eq}}, \tag{4.8}$$

$$\boldsymbol{lb} \leq \boldsymbol{x} \leq \boldsymbol{ub}. \tag{4.9}$$

Input: $\overline{y}_{1:T}$ aggregated signal; $\left\{\boldsymbol{\mu}^{(1:N)}, \boldsymbol{P_b}^{(1:N)}, \boldsymbol{\phi_b}^{(1:N)}\right\}$ parameters for N HMMs; ${\sigma_1}^2, {\sigma_2}^2, \lambda$ covariance and regularization parameters.

Minimize over $\{\mathcal{Q} \in \mathcal{L} \cap \mathcal{O}\}$

$$\begin{aligned} &\frac{1}{2{\sigma_1}^2}\sum_{t=1}^{T} E_t^{(a)} + \frac{1}{2{\sigma_2}^2}\sum_{t=2}^{T} E_t^{(bc)} + \frac{1}{2}\sum_{t=2}^{T} E_t^{(bnc)} + \\ &+ \sum_{t=2}^{T}\sum_{i=1}^{N}\sum_{\substack{j=1\\k=1}}^{m_i} \left\{ Q(x_{t-1}^{(i)}, x_t^{(i)})_{j,k} \left(-\log P_{b_{k,j}}^{(i)} \right) \right\} + \\ &+ \sum_{i=1}^{N}\sum_{j=1}^{m_i} \left\{ Q(x_1^{(i)})_j (-\log \phi_{b}{}_j^{(i)}) \right\} \end{aligned} \tag{4.2}$$

Output : $\hat{y}_{1:T}^{(1:N)}$, predicted individual HMM output

$$\hat{y}_t^{(i)} = \sum_{j=1}^{m_i} \mu_j^{(i)} Q(x_t^{(i)})_j \tag{4.3}$$

Figure 4.2: The AFAMAP algorithm.

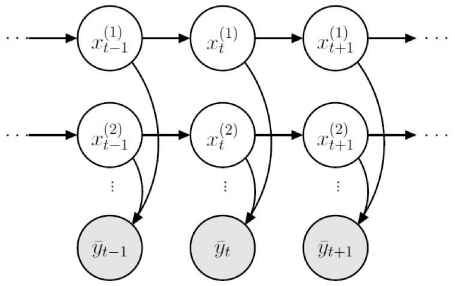


Figure 4.3: Additive FHMM model.

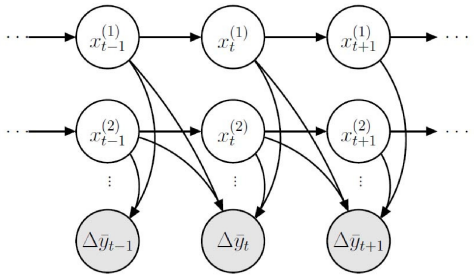


Figure 4.4: Differential FHMM model.

The variables of the problem are represented by the vector $\boldsymbol{x}$, which is composed of several subsets, based on the time instant t and the appliance index (i):

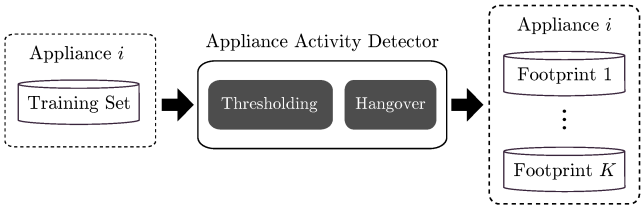
$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{\Theta_1} \\ \vdots \\ \boldsymbol{\Theta_T} \end{bmatrix} \quad , \quad \boldsymbol{\Theta_t} = \begin{bmatrix} \boldsymbol{\Psi_t^{(1)}} \\ \vdots \\ \boldsymbol{\Psi_t^{(N)}} \end{bmatrix} \quad , \quad \boldsymbol{\Psi_t^{(i)}} = \begin{bmatrix} \boldsymbol{\xi_t^{(i)}} \\ \boldsymbol{\beta_t^{(i)}} \end{bmatrix},$$

$$\boldsymbol{\xi_t^{(i)}} = \begin{bmatrix} Q(x_t^{(i)})_1 \\ \vdots \\ Q(x_t^{(i)})_{m_i} \end{bmatrix} \quad , \quad \boldsymbol{\beta_t^{(i)}} = \begin{bmatrix} Q(x_{t-1}^{(i)}, x_t^{(i)})_{1,1} \\ \vdots \\ Q(x_{t-1}^{(i)}, x_t^{(i)})_{1,m_i} \\ \vdots \\ Q(x_{t-1}^{(i)}, x_t^{(i)})_{m_i,1} \\ \vdots \\ Q(x_{t-1}^{(i)}, x_t^{(i)})_{m_i,m_i} \end{bmatrix},$$

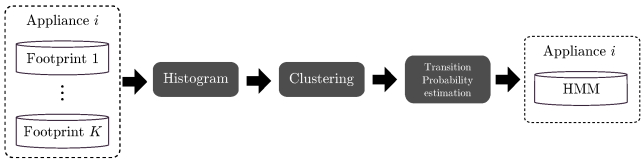
where the variables for the state are represented in $\boldsymbol{\xi_t^{(i)}}$, and the variables for the backward transition in $\boldsymbol{\beta_t^{(i)}}$.

The parameters of the problem fill up the elements of $\boldsymbol{H}$ and $\boldsymbol{f}$, according to the structure of the $\boldsymbol{x}$ vector, whereas $\boldsymbol{A_{eq}}$ and $\boldsymbol{b_{eq}}$ are used to represent the consistent constraints between the state and the transition variables. The vectors $\boldsymbol{lb}$ and $\boldsymbol{ub}$ define the lower and upper boundaries of the solution: because of the nature of the variables [22], the lower boundary is equal to 0, whereas the upper boundary to 1, for all the elements in $\boldsymbol{x}$.

In $\boldsymbol{Aeq}$ the constraint about $Q(x_{t-1}^{(i)}, x_t^{(i)})$ with $t = 1$ has to be removed since there is no information about $Q(x_t)$ at the previous time instant, thus falling back to the constraint $0 \cdot Q(x_{t-1}^{(i)}, x_t^{(i)}) = 0$.



(a) Footprint extraction block scheme.



(b) Model training block scheme.

Figure 4.5: Diagram of the footprint extraction procedure (a) and of the training phase of the appliance models (b).

4.1.1 Appliance modelling

The working states power level estimation consists in obtaining representative power level distributions related to each appliance state, i.e., the values of the emitted symbols $\boldsymbol{\mu}_j$. In a realistic scenario, this is obtained by using a set of examples of an appliance typical consumption cycle. This information can be extracted by observing the aggregate power signal, under the assumption that only one appliance at time is operating [16].

In particular, this stage involves the extraction of a *footprint* of the appliance, i.e., the active and reactive power signals comprised between the power on (transition from the OFF state to an ON state) and the power off (transition from an ON state to the OFF state). This is performed by firstly identifying these instants by means of an *Appliance Activity Detector* (AAD). Basically, it consists in detecting when the active power level signal exceeds a certain threshold or not (typical values are in the order of 20 W). Isolated occurrences of power levels below the threshold are managed by employing a *hangover* technique: it is a counter, which decreases its value for each sample the signal is below the threshold. If the signal returns over the threshold before the end of the counter, the footprint is considered continued. The typical value is 5-10 minutes. The diagram of the footprint extraction stage is shown in Figure 4.5a.

The power value and the temporal information of the OFF state cannot be obtained by analysing the signal extracted with the AAD. The value is reasonably assumed 0 W and 0 VAR for the active and reactive power signals, respectively. The temporal information, i.e., the typical interval intercurring between the OFF state and a ON state, has to be specified a-priori for each

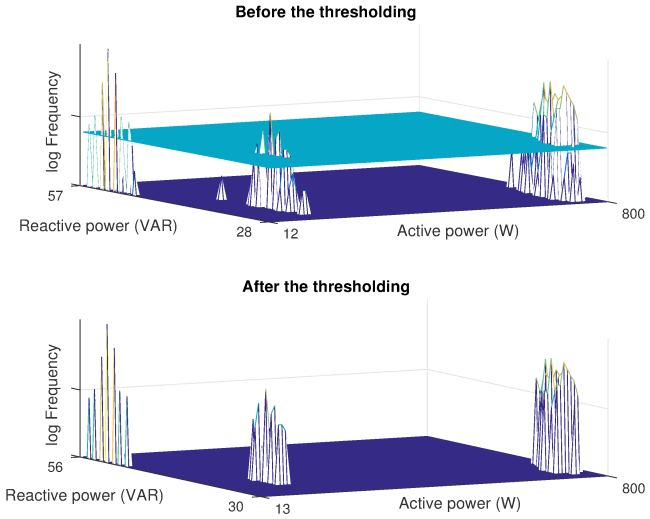


Figure 4.6: An example of a two-dimensional histogram of the active and reactive power signals related to the dishwasher in the dataset AMPds.

appliance based on the typical usages (e.g., once in an hour, three times in a day, etc.).

Different uses of the appliance in its life cycle from the user lead to the need of model representation of every combination of usage, under the assumption that the working state of the appliance are predetermined and not varying from different usage: reasonably, the working cycle of a washing machine are always the same (e.g., pre-washing, water heating, washing, rinsing and spinning), indifferently from the order of execution thus the number of working state is predetermined for every appliance.

Complex appliances (e.g., washing machines, dishwashers) are characterised by several working cycles and the extraction of a single footprint might not be completely representative of its operation. This motivates the need to acquire several footprints for each appliance. Furthermore, even though only one footprint is sufficient to explore all the working states of an appliance, multiple footprints allow to employ more data for the power level extraction phase, particularly useful for those power levels characterised by a short duration.

The estimation of the power level associated to a state of the HMM relies on the appliance consumption data, which is not composed of discrete values of consumption, but it presents a continuous variability in the values. In order to find the averaged values of the signal, within the period of permanence in the same working state, a clustering procedure is adopted, and the k-means [86]

has been selected as the algorithm.

Since the OFF state information is not present in the data, the number of clusters is set to $(m-1)$. After identifying the clusters, the power levels associated to each HMM state are represented by their centroids.

The clustering operation is not directly performed on the footprints extracted with the AAD. Indeed, after extracting the footprint, a bivariate histogram composed of 100 bins per kW and per kVAR is used to analyse the probability distribution of the active and reactive power signals. The number of bins is empirically chosen after analysing some footprints of the training set in order to obtain a sufficiently detailed histogram able to provide a good trade-off between variance and bias of the density estimate. Additionally, power levels with a low number of occurrences are excluded from the successive processing. More in details, bins having a number of occurrences below the threshold are considered of lower relevance, thus the related observations are discarded. This technique allows to obtain the number of working states m, which is determined by observing the number of clusters obtained in the final bivariate histograms. An example is shown in Figure 4.6, where the histogram before and after the thresholding operation is shown. It refers to the dishwasher consumption in the AMPds dataset. Additionally, it reduces a limitation of the clustering algorithm: k-means does not employ the information on the samples distribution in the cluster, since it selects the centroid which satisfies the rule of convergence over all data. Discarding bins with low occurrences forces k-means to select the centroids with higher probability and to discard local clusters with lower probability, that could result in erroneous centroids. Furthermore, it allows to discriminate close clusters which can be confused as a single one: indeed, transients between near clusters produce samples comprised between the cluster with higher occurrences, which merge the two clusters in a single one. The diagram of the clustering and of the model training stage is shown in Figure 4.5b.

In general, clusters present different characteristics depending on the magnitude of their centroid. Typically, the ones characterised by high values (e.g., 3000 W) are highly variable, since they depend on the appliance usage by the user, e.g., the water temperature chosen in the washing machine or the rinsing cycle of the dishwasher affect the maximum power consumption. On the other hand, clusters characterised by low power value (e.g., 300 W) have lower variability, since deviation from the centroid is mainly caused by intermediate working stages of the appliance, and they do not depend on the usage.

Figure 4.7 shows an example of the inference procedure conducted on the active power signal only, denoted as P_a, and on the joint active-reactive power signals, denoted as (P_a, P_r). The signals are related to the washing machine in the AMPds dataset. In particular, Figure 4.7a shows the active power signal

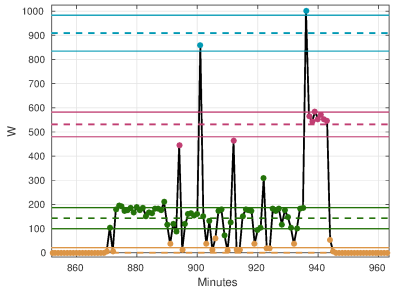
and the cluster membership of each sample when k-means operates on the P_a signal only. Figure 4.7b and Figure 4.7c show respectively the same active power signal and the reactive power signal, but the cluster membership is related to the outcome of k-means operating on the joint (P_a, P_r). Figure 4.25b shows at the bottom the 1-D P_a line with the clusters obtained when k-means operates on the P_a signal only and at the top the (P_a, P_r) plane with the clusters obtained when k-means operates on the joint (P_a, P_r) signals. In the figure, each cluster is depicted as an interval or as an ellipse whose size is twice the standard deviation of the cluster centered at its centroid. The number of clusters is different between the active power and the active and reactive power cases: in the first case 4 cluster can be identified, whereas the addition of the reactive power allows to distinguish 5 clusters. As shown in the figure, 2 clusters share the same value of active power, but differ in the reactive component. Using the reactive power, thus, allows to have a better representation of the working states of the appliance, therefore reducing the admissible combination of working states in the aggregated data.

Figure 4.8 shows an example of the inference procedure conducted on the active power signal. The signal is related to the washing machine in the AMPds dataset. In particular, Figure 4.8a shows the active power signal and the cluster membership of each sample. In the figure, each cluster is depicted as an interval whose size is twice the standard deviation of the cluster centred at its centroid. The related HMM is represented in Figure 4.8b.

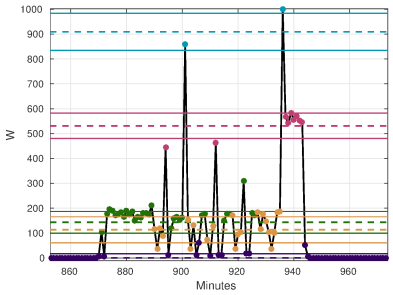
The HMM is a representation method based on the Finite State Machine (FSM), in which the transitions between the states are regulated by a probability matrix, proportional to the time of permanence in the states and the number of times the model pass from a state to another one. Figure 4.9 shows an example of HMM with 4 states.

The transition probability matrix $\boldsymbol{P}$ is obtained using the *Baum-Welch* [76] training algorithm: in the specific case when $\boldsymbol{M} \equiv \boldsymbol{I}^{m \times m}$, only the number of states m composing the HMM and the observed sequence of symbols have to be specified to the algorithm. Each HMM state is assigned to a power consumption state, therefore to a cluster resulting from the procedure described in Subsection 4.1.1.

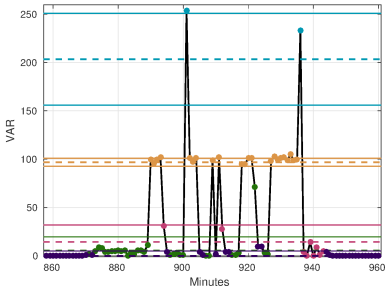
Table 4.1 shows the transition probability matrix related to the washing machine footprint showed in Figure 4.25b. The highest values in the matrix are the ones located on the diagonal, which represent the probability of remaining in the same state, respect to the transition to another one: indeed, for the state where the permanence time is low, this value is lower than the one of the state where the permanence time is higher. The highest value is the one related to the OFF state, because the activation of the appliance occurs after a long time in which it is turned off.



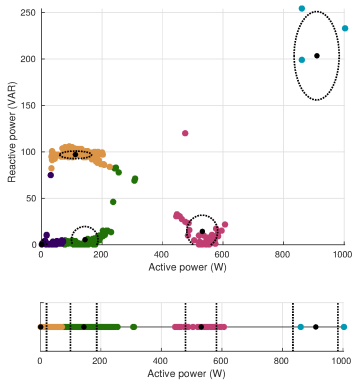
(a) Footprint (P_a) and cluster membership of each sample with k-means operating on P_a.



(b) Footprint (P_a) and related clusters with k-means operating on (P_a, P_r).



(c) Footprint (P_r) and related clusters with k-means operating on (P_a, P_r).



(d) Clusters in the (P_a, P_r) plane (above) and the P_a line (below).

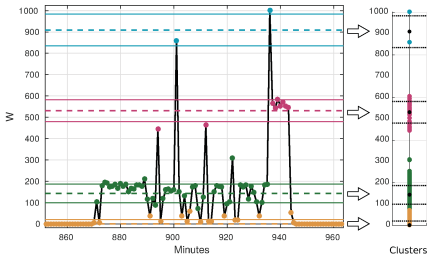
Figure 4.7: Washing machine footprint and clusters in the dataset AMPds.

Since the pause interval between two footprint is not recorded, the user has to establish the time interval between two appliance activations, e.g., the typical time of use in the daytime or the number of activations per day of the appliance, in order to calculate the OFF interval and to use this value for the calculation of the transition probability related to the OFF state.

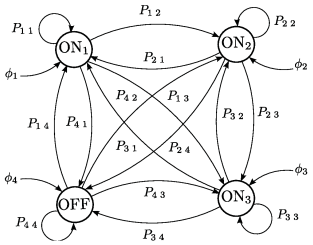
The probability value which tends to zero denotes that the transition is unlikely. In the practice, it is recommended to avoid zero probability value, because it is evaluated in log scale in the AFAMAP algorithm, and it tends to infinity. It is recommended to fix the value to a little quantity, e.g., $\simeq 10^{-5}$.

4.1.2 Rest-of-the-World model

In a real case scenario, a noise contribution can be observed on the aggregated signal, due to electrical noises in the system, very small loads, leakages. This



(a) Footprint and cluster membership of each sample.



(b) The 4 states HMM.

Figure 4.8: Washing machine footprint and HMM in the dataset AMPds.

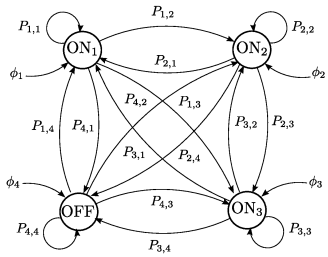


Figure 4.9: A 4 states HMM.

contribution can be considered as a source of power consumption, additionally to the appliances which the system tries to disaggregate, therefore it can be modelled with a HMM, as described in Subsection 4.1.1, leading to a *noise* model or *Rest-of-the-World* (RoW) model. The number of working states is a parameter which could depends on the application scenario, therefore it has to be explored in the experimental phase, nevertheless it would be grater than the number of states defined for the appliances, since it represents a set of multiple load working at the same time. The data used for training this model can be extracted by observing the aggregate power signal, when all the appliances of interest are switched off and all the remaining equipments in the house are working.

Referring to equation (2.1), the training signal used to create the RoW model is the residual power consumption from the aggregated data, excluded the appliances power consumption:

$$e(t) = y(t) - \sum_{i=1}^{N} y_i(t). \tag{4.10}$$

In the case where the dataset comprises always-on appliances, since no operating cycle or footprint is defined in this case the RoW model does not include

Table 4.1: An example of the HMM transition probability matrix.

		Destination state			
		ON_1	ON_2	ON_3	OFF
Start state	ON_1	0.832	0.085	0.081	0.002
	ON_2	0.080	0.690	0.202	0.028
	ON_3	0.012	0.028	0.916	0.045
	OFF	3.1e-05	2.7e-05	0.002	0.998

the OFF working state, as showed in the Figure 4.10.

The consumption values in the working states of the RoW model are extracted algorithmically using the k-means, even if there are no evident consumption values clusters, determined by any working state.

4.2 Algorithm improvements

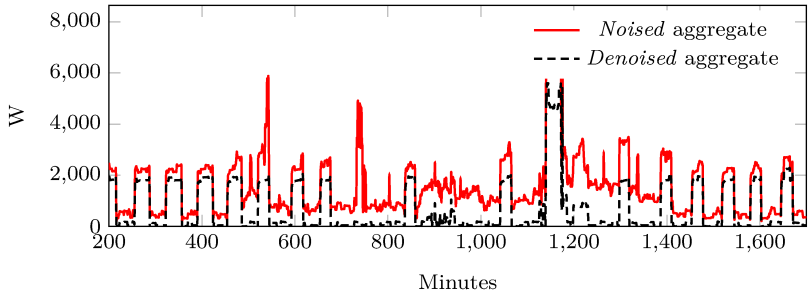
In the reference approach, the DFHMMs are obtained as the difference, in term of power consumption, between the current and the previous sample (referred to as *backward transition*), so that a change in the state of an HMM can be evaluated against the change in the aggregated power consumption. Similarly, an additional evaluation, based on the next against the current sample (referred to as *forward transition*), is carried out. Furthermore, a smarter employment of the solver boundaries is evaluated, starting from a more accurate analysis of the aggregated power or using heterogeneous information, as the reactive power consumption of the electrical system.

Since the AFAMAP algorithm operates offline, it is possible to further extend the model by taking into account the transition from the current to the next state. The original DFHMM [22] is computed by looking backward from the current sample to the previous one, and thus it can be addressed to as Backward DFHMM. The new differential FHMM is computed by looking forward, as showed in Figure 4.11, and thus is referred to as Forward FHMM.

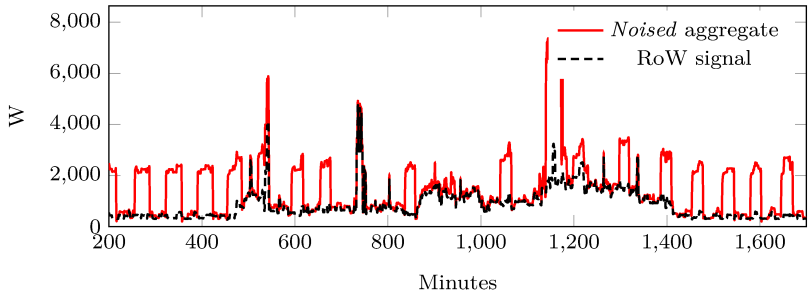
The formulation of the new model, also, differs from the original one, only in the index order. The new variables define the problem, as follow:

$$\mathcal{Q} = \left\{ \boldsymbol{Q}(\boldsymbol{x_t^{(i)}}) \in \mathbb{R}^{m_i}, \boldsymbol{Q}(\boldsymbol{x_{t+1}^{(i)}}, \boldsymbol{x_t^{(i)}}) \in \mathbb{R}^{m_i \times m_i} \right\},$$

where the variables are indicators of the transition from the next to the current state: $Q(x_t^{(i)})_j = 1 \Leftrightarrow x_t^{(i)} = j$, and $Q(x_{t+1}^{(i)}, x_t^{(i)})_{j,k} = 1 \Leftrightarrow x_{t+1}^{(i)} = j, x_t^{(i)} = k$. The consistent constraints between the state variables and transition variables



(a) Noised aggregated power vs denoised aggregated power.



(b) Noised aggregated power vs RoW signal.

Figure 4.10: The denoised aggregated power and the RoW signal, compared to the main aggregated power, in the AMPds.

need to be satisfied:

$$\mathcal{L} = \left\{ \mathcal{Q} : \begin{array}{c} \sum_{j=1}^{m_i} Q(x_t^{(i)})_j = 1 \\ \sum_{k=1}^{m_i} Q(x_{t+1}^{(i)}, x_t^{(i)})_{j,k} = Q(x_{t+1}^{(i)})_j \\ \sum_{k=1}^{m_i} Q(x_{t+1}^{(i)}, x_t^{(i)})_{k,j} = Q(x_t^{(i)})_j \\ 0 \leq Q(x_t^{(i)})_j, Q(x_{t+1}^{(i)}, x_t^{(i)})_{k,j} \leq 1 \end{array} \right\} . \tag{4.11}$$

Therefore, the new cost function is derived for the Forward DFHMM, based

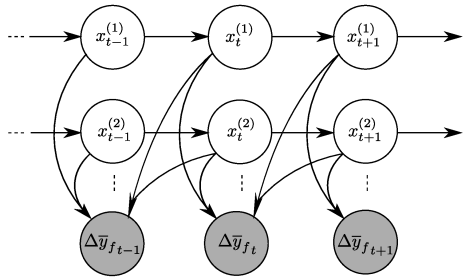


Figure 4.11: The Forward Differential FHMM.

on the forward differential aggregated signal $\Delta\overline{y}_{f_t} = \overline{y}_t - \overline{y}_{t+1}$, as follow:

$$\begin{aligned}
&\frac{1}{2{\sigma_3}^2}\sum_{t=1}^{T-1} E_t^{(fc)} + \frac{1}{2}\sum_{t=1}^{T-1} E_t^{(fnc)} + \\
&+\sum_{t=1}^{T-1}\sum_{i=1}^{N}\sum_{\substack{j=1\\k=1}}^{m_i}\left\{Q(x_{t+1}^{(i)}, x_t^{(i)})_{j,k}\left(-\log P_{f\,k,j}^{(i)}\right)\right\} + \\
&+\sum_{i=1}^{N}\sum_{j=1}^{m_i}\left\{Q(x_T^{(i)})_j(-\log {\phi_f}_j^{(i)})\right\},
\end{aligned} \tag{4.12}$$

where the error terms in (4.12) are defined as:

$$E_t^{(fc)} = \sum_{i=1}^{N}\sum_{\substack{j=1\\k=1\\k\neq j}}^{m_i}\left\{\left(\Delta\overline{y}_{f_t} - \Delta\mu_{k,j}^{(i)}\right)^2 Q(x_{t+1}^{(i)}, x_t^{(i)})_{j,k}\right\}, \tag{4.13}$$

$$E_t^{(fnc)} = D\left(\frac{\Delta\overline{y}_{f_t}}{\sigma_3}, \lambda\right)\left(1 - \sum_{i=1}^{N}\sum_{\substack{j=1\\k=1\\k\neq j}}^{m_i} Q(x_{t+1}^{(i)}, x_t^{(i)})_{j,k}\right). \tag{4.14}$$

The transition matrix $\boldsymbol{P_f^{(i)}}$ represents the probability of state change from the next to the current time instant: this parameter is equivalent to the typical representation of the transition matrix (i.e., the probability of state change from the previous time instant to the actual) evaluated after flipping the signal, thus it can be derived by using the available algorithm for HMM training. The parameter $\boldsymbol{\phi_f^{(i)}}$ represents the final state distribution, that is the initial state distribution starting from the end of the signal.

Since the duality in the forward and backward representation of the AFHMM (i.e., it is derived from the same observed signal, but in opposite directions), the problem definition using only one of the two versions of the DFHMM leads to

the already known performance. Considering simultaneously both versions of DFHMM may lead to performance improvements: for this reason the forward differential function (4.12) is added to the reference formulation (4.25), thus leading to a new optimization problem.

The variable vector $\boldsymbol{x}$ in the QP problem accounts for the new terms, following the same structure introduced in Section 4.1:

$$\boldsymbol{\Psi_t^{(i)}} = \begin{bmatrix} \boldsymbol{\xi_t^{(i)}} \\ \boldsymbol{\beta_t^{(i)}} \\ \boldsymbol{\phi_t^{(i)}} \end{bmatrix}, \qquad \boldsymbol{\phi_t^{(i)}} = \begin{bmatrix} Q(x_{t+1}^{(i)}, x_t^{(i)})_{1,1} \\ \vdots \\ Q(x_{t+1}^{(i)}, x_t^{(i)})_{1,m_i} \\ \vdots \\ Q(x_{t+1}^{(i)}, x_t^{(i)})_{m_i,1} \\ \vdots \\ Q(x_{t+1}^{(i)}, x_t^{(i)})_{m_i,m_i} \end{bmatrix},$$

where the new term $\boldsymbol{\phi_t^{(i)}}$ represents the variables for the forward transition.

The introduction of the new variables leads to an alteration of the problem constraints, represented by the parameters $\boldsymbol{A_{eq}}$ and $\boldsymbol{b_{eq}}$, and the variable boundaries $\boldsymbol{lb}$ and $\boldsymbol{ub}$. In $\boldsymbol{Aeq}$ the constraint about $Q(x_{t+1}^{(i)}, x_t^{(i)})$ with $t = T$ has to be removed since there is no information about $Q(x_t)$ at the following time instant, thus falling back to the constraint $0 \cdot Q(x_{t+1}^{(i)}, x_t^{(i)}) = 0$.

In order to solve the optimization problem, different solutions, which satisfy the constraints, need to be evaluated before the solver finds the optimal one. As such, the values of $\boldsymbol{x}$ that are not compatible with the given set of samples can be discarded, to restrict the search domain and improve the search efficiency.

On purpose, the lower and upper boundaries of the variable $\boldsymbol{x}$ are selected beforehand in order to prevent that the solver investigates those combinations of states that do not match the value of the aggregated power consumption. The selection method is similar to the one proposed in [87].

If several runs of a single appliance are evaluated, although the same working states are identified, the signature tends to differ from a run to the other. For this reason, the appliance power consumption can be modelled as a stochastic process and, therefore, the output value $y_t^{(i)}$, relative to a working state $x_t^{(i)}$ of an appliance, can be modelled as a gaussian variable, described by a mean value and a variance value:

$$y_t^{(i)} | x_t^{(i)} \sim \mathcal{N}\left(\mu_{x_t^{(i)}}^{(i)}, {\sigma_{x_t^{(i)}}^{(i)}}^2\right). \tag{4.15}$$

Regard to this, the power signal is replaced by a simplified model that presents a constant power consumption, corresponding to the mean value of

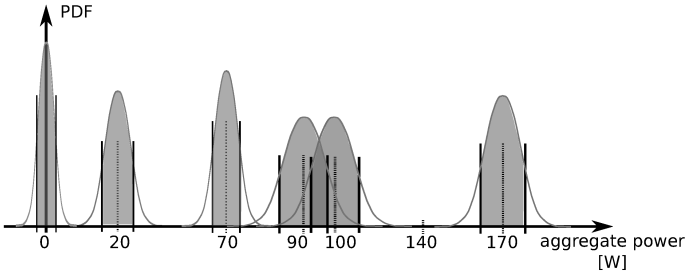


Figure 4.12: A sketch of the different probability density functions (PDF) for each aggregated power value produced by the combination of all appliances states power levels.

the working state power value, with a superimposed noisy contribution, described by the variance value in the working state.

Since the aggregated data $\overline{y}_t$ is assumed to correspond with the sum of the power consumption of each appliance, it can be modelled as a gaussian variable, described by a mean value and a variance value equivalent to the sum of the corresponding values of each appliance, under the assumption of statistical independence between the appliances:

$$\overline{y}_t | x_t^{(1:N)} \sim \mathcal{N}\left(\sum_{i=1}^{N} \mu_{x_t^{(i)}}^{(i)}, \sum_{i=1}^{N} {\sigma_{x_t^{(i)}}^{(i)}}^2 \right). \tag{4.16}$$

This simplified model results in a number of admissible combinations of working states equal to $\prod_{i=1}^{N} m_i$. It allows to evaluate which combination of working states fit the power value for each sample of the aggregated data, thus discarding the incompatible ones. The effectiveness interval for each combination is centred in mean value, and its width is twice the value of the standard deviation. For some combinations, which have similar mean value or great variance, the effectiveness intervals result overlapped: for those cases, if the power value falls in this region, both the combinations are considered valid.

Based on this observation, it is possible to manipulate the boundaries of the QP problem domain. For instance, if 2 HMMs are considered, $M1$ and $M2$, whose power levels are, $M1 = \{70, 0\}$ and $M2 = \{100, 20, 0\}$ respectively, the different combined power levels are $\{0, 20, 70, 90, 100, 170\}$, each one with its own variance value. This example is represented in Figure 4.12. Considering a few different values of aggregated power, e.g., $\overline{y}_t = \{20, 95, 140\}$, it can be observed that $\overline{y}_t = 20$ is obtained as the combination $(x_t^{(1)} = 2, x_t^{(2)} = 2)$,

therefore the allowed constraints are defined as:

$$\begin{bmatrix}0\\1\end{bmatrix} \leq \boldsymbol{\xi_t^{(1)}} \leq \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix} \leq \boldsymbol{\xi_t^{(2)}} \leq \begin{bmatrix}0\\1\\0\end{bmatrix}.$$

If $\overline{y}_t = 95$, the value falls in an overlapped interval, belonging to the combinations $(x_t^{(1)} = 2, x_t^{(2)} = 1)$ and $(x_t^{(1)} = 1, x_t^{(2)} = 2)$, thus, the allowed constraints are defined as:

$$\begin{bmatrix}0\\0\end{bmatrix} \leq \boldsymbol{\xi_t^{(1)}} \leq \begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}0\\0\\0\end{bmatrix} \leq \boldsymbol{\xi_t^{(2)}} \leq \begin{bmatrix}1\\1\\0\end{bmatrix}.$$

Whereas, if $\overline{y}_t = 140$, no combination is corresponding, thus the boundaries remain as default.

Clearly, the same process can be applied to bound the $\boldsymbol{\beta_t^{(i)}}$ and $\boldsymbol{\phi_t^{(i)}}$. In regard to this, however, since transitions are related to the steady states, the evaluation of the steady states is enough to bound both kinds of variables.

Even though disaggregation is aimed for the aggregated power consumption, in most cases the focus is centred on the active power alone. Nonetheless, given the generality of the AFAMAP algorithm, targeting the reactive aggregated power is also possible. In regard to this, in the present work, the application of the AFAMAP algorithm to the aggregated reactive power has been investigated as well, based on the fact that reactive power is a common trait of the power signature of a residential appliances subset.

In the current scenario, the disaggregation of the reactive power samples is carried out, in order to collect additional information about the activity states of the appliances. This information, in turn, is used to further define the lower and the upper boundaries of the states in the active power disaggregation. Similarly to the active power case, the HMMs are modelled for each appliances starting from the signature in the reactive power and the AFAMAP algorithm is run by using the aggregated reactive power signal as input.

Following the basic knowledge in circuit theory, an electrical load with a reactive component (i.e., an appliance) which has a reactive power consumption greater than 0 is necessary turned on, therefore the boundaries of the problem in active power disaggregation are assigned as follows:

$$\begin{bmatrix}0\\0\\0\end{bmatrix} \leq \boldsymbol{\xi_t^{(i)}} \leq \begin{bmatrix}1\\1\\0\end{bmatrix}.$$

Although, when the reactive power consumption is 0, the active component

could be both null or greater than 0, depending on whether the appliance is turned off or only the load passive component is working. Therefore, the boundaries of the problem in active power disaggregation are setted as default.

4.2.1 Experimental setup

The dataset used for the experiments is the Almanac of Minutely Power dataset (AMPds) [58]: it contains recordings of consumption profiles belonging to a single home in Canada for a period of two years at 1 minute sampling rate. It provides active and reactive power at appliance level, unlike most of the dataset in which only the active power is provided at appliance level, as described in Section 2.3: this information is crucial to test the new approach based on the reactive power disaggregation as constraint. Analysing the contents of the dataset, it can be noticed that the usage of the appliances is homogeneous throughout the entire period, therefore the experiments are evaluated on 6 months of data, which can be considered representative of the entire dataset. To create the HMM models of the appliances, the training requires at least one signature per appliance, although multiple signatures lead to a more general model. In the proposed work, a subset of the data, spanning over 14 days, has been deemed sufficient to collect all the signatures required to train all the HMMs. The HMM are trained in accordance to the Baum-Welch algorithm, after determining the ground truth state over the time: those are obtained through a clustering procedure, in which every cluster represents a power consumption level of the appliance, thus a state of the HMM. This process is achieved using the k-means algorithm, in which the number of the cluster is imposed in a supervised manner, starting from the knowledge of the operating states of the appliance. The power level mean and the variance values are achieved by means of a gaussian variable fitting procedure over the samples belonging to each cluster. To satisfy the condition of *denoised* system, the aggregated data is synthetically composed by summing the appliance level power signals. The experiments are conducted by using the appliances at higher contribution, therefore 6 appliances have been chosen: dryer, washing machine, dishwasher, fridge, oven, and heat pump. The simulations are conducted in Matlab environment and the CPLEX solver is used to solve the QP problem. The value of starting probability $\boldsymbol{\phi_b^{(i)}}$ of the i-th HMM is imposed to assume the certainty for the OFF state for $f = 1$, whereas for the consecutive windows, $1 < f \leq F$, it is imposed to assume the value of the last sample $\boldsymbol{\xi_T^{(i)}}$ of the previous window, in order to ensure the contiguity of the solution on the window border. The value of the ending probability $\boldsymbol{\phi_f^{(i)}}$, instead, is uniformly imposed in every state, since no information from the consecutive window is available. Different experiments are conducted varying the size of the windows

between the values $T \in \{10, 30, 60, 90, 120\}$ minutes, and the effectiveness of the innovative aspect is evaluated: the introduction of the forward term in the cost function, the selection of the boundaries related to the aggregated power level and to the disaggregation output of the reactive power. The variance parameters are defined with ${\sigma_1}^2 = {\sigma_2}^2 = {\sigma_3}^2 = 0.01$ according to the variance of the experimental data and the regularization parameter $\lambda = 1$.

4.2.2 Results

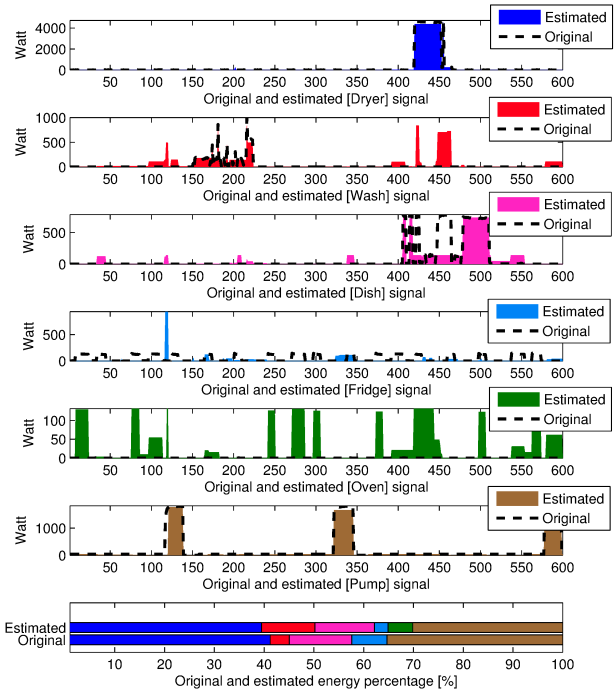
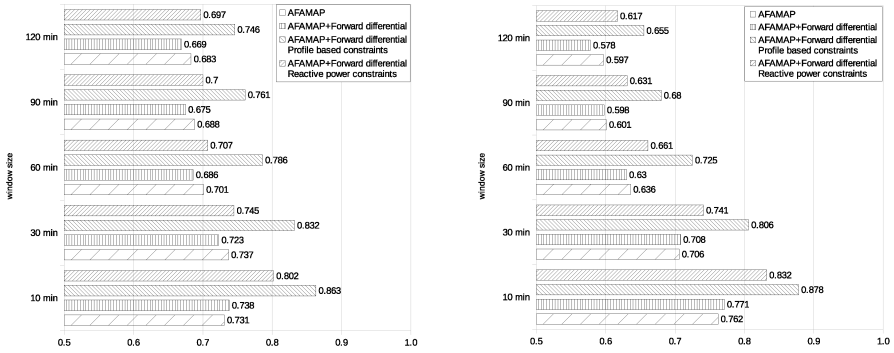


Figure 4.13: Appliances consumption: estimated AFAMAP disaggregation output against original signals.

The results of the experiments, based on the scenario described in Subsection 4.2.1, are presented in the current section.

In Figure 4.13, the AFAMAP disaggregated power consumption profiles of the appliances are compared against the corresponding true outputs, provided by the dataset: in the figure a time span of 10 hours, corresponding to 600 samples, is considered. At the bottom, the energy distribution over the same period, expressed among the appliances in terms of percent value, is compared between the reconstructed and the true appliances consumption.



(a) State based metric: F_{1S}

(b) Energy based metric: F_{1E}

Figure 4.14: Disaggregation performance on AMPds dataset using 6 appliances, with different algorithm configuration.

The signals reveal that the appliances which show an high steady power consumption are easily recognized, whereas the appliances with complex working cycles, or with several power levels, are more difficult to detect. Indeed, whenever several appliances present similar consumption levels, many combinations may satisfy the problem constraints, thus additional information is required to identify the active appliances. For instance, in Figure 4.13, the oven and the fridge are seldom recognised, whereas the detection of the dryer and the washing machine are partially more successful.

The evaluation of the algorithm performance is carried out by means of the metrics proposed in Section 2.4. Although the focus of the present work is the AFAMAP algorithm, the dataset being used and the proposed training method are different with respect to [22], therefore a direct comparison against the results proposed in the reference work is not possible. To overcome this shortcoming, the baseline has been created anew, by means of the AFAMAP algorithm, the AMPds dataset and the proposed training method.

The disaggregation results computed by means of the metrics are reported in Figure 4.14: in Figure 4.14a the state based metric is presented, whereas the energy based metric is proposed in Figure 4.14b. The results are shown for different values of the time window length. Clearly, since all the results exceed 0.5, the plots have been drawn from 0.5 onwards.

Both plots show that the best results are achieved using the shortest time window. On a different note, however, not every configuration improves in the same way.

Focusing on the state based metrics, it is possible to observe that the AFAMAP baseline shows a significant performance improvement with the decreasing of the window length, except when passing from the 30 to 10 minutes window

size. On the contrary, the forward differential model gives an improvements at the shorter window size, resulting in the best performance in the unbounded problem solution, with a F_{1S} of 0.738 and an improvement of 1% respect to the baseline.

Fixing the boundaries of the problem, in every simulation case, gives the benefit on the disaggregation results: the profile based method gives a considerable performance improvements with every window size, but the highest relative improvement can be noted at the smallest size, resulting to a F_{1S} of 0.863 and a relative improvement of 18%.

Alternatively, the boundaries can be setted based on the reactive power disaggregation feedback: the results, showed in Table 4.2, demonstrate that the reactive power reaches high performance in disaggregation. This is due to the high difference in the reactive components of each appliance, which involves a strong distinction in the creation of the HMM, therefore allowing an highly reliable disaggregation. The usage of this information results in a performance improvement for every window size, more considerable at the smallest size: in general, the usage of the reactive power feedback gives benefits to the disaggregation, with a F_{1S} of 0.802 and a relative improvement of 9.7%, therefore less than the profile based constraints.

Clearly, the same trends presented about the state based metrics still hold true when evaluating with the energy based metrics. The most notably difference between the two plot, in fact, is that the rate of improvement of the algorithms when decreasing the time window length: indeed, the forward differential model introduction results to a F_{1E} of 0.771 and a relative improvement of 1.2% respect to the baseline, whereas the profile based setting of the boundaries results to a F_{1E} of 0.878 with a relative improvement of 15.2% and the reactive power based method to a F_{1E} of 0.832 with an improvement of 9.2%.

The forward differential model seems to be beneficial only with the shortest time window: it may be a direct consequence of the problem formulation alteration. Indeed, the introduction of additional variables increases the size of the problem, therefore the computational burden, for which the solver demonstrates worst performance, as it happens for the baseline approach with larger window size.

Table 4.2: Disaggregation results on reactive power. The configuration used is: AFAMAP + Forward differential.

	window size				
Metric	10 min	30 min	60 min	90 min	120 min
State based: F_{1S}	0.922	0.877	0.869	0.867	0.865
Energy based: F_{1E}	0.935	0.883	0.877	0.875	0.874

Despite this, the improvements achieved adding the differential forward information to the model are restricted to the application scenario: since the algorithm operates on a per-sample basis, for each appliance behaviour two state changes unlikely happen across three contiguous samples, thus the forward difference cannot provide a substantial support to the inference of the actual working state.

The errors in the disaggregation phase are caused by the multiplicity of states combinations which can correspond to the same value of the aggregated data: for this reason the use of boundaries allows to exclude some solutions that are not eligible, therefore facilitates the solver to find the exact solution to the problem. Nevertheless, the variation over time of the power consumption associated to a specific appliance working state, causes an unwanted variability, i.e., a noise component, in the achieved solution.

4.3 Exploitation of the reactive power

In this section, a disaggregation algorithm based on FHMMs and active and reactive power measured at low sampling rates is proposed. The HMM models of the appliances and the proposed solution for obtaining their parameters from a training dataset are described. Load disaggregation is performed by proposing a reformulated version the Additive Factorial Approximate Maximum a Posteriori (AFAMAP) algorithm [22] that allows a straightforward extension to the bivariate case. The experimental evaluation has been conducted on the Almanac of Minutely Power dataset (AMPds) dataset [58] in noised and denoised scenarios, and the proposed solution has been compared to AFAMAP based on the active power only and to two variants of Hart's algorithm [16] both based on active and reactive power. The results show that in terms of F_1-Measure the proposed approach provides a significant performance improvement with respect to the comparative methods.

A part from [21], the aforementioned approaches employ the active power as the sole electrical parameter for NILM, despite some algorithmic frameworks have been formulated for operating on multidimensional feature vectors [22]. The use of reactive power has been employed since the very first work by Hart [16] and in more recent works based on the same principles [88, 89, 90, 91, 92] or on transient-state analysis [47, 48, 49, 50, 51]. However, up to the author's knowledge, the only work that employs both the active and the reactive power in the FHMM framework is the work by Zoha and colleagues [21].

Following a similar philosophy, a disaggregation algorithm based on FHMMs that uses both the active and reactive power is proposed. However, differently from [21], where the disaggregation algorithm is based on the structural variational approximation method and on the Viterbi algorithm, in the proposed

approach the active power is disaggregated by reformulating the AFAMAP algorithm for the bivariate case. As demonstrated in [22], this allows the introduction of a Differential FHMM (DFHMM) that improves the performance and reduces the computational cost. Thus, differently from [21], here the reactive power component is introduced also in the DFHMM. More in details, the proposed solution belongs to the family of supervised approaches based on steady state signals acquired from low frequency measurements. The reactive power is introduced in the FHMM framework by employing bivariate hidden Markov appliance models whose emitted symbols are represented by active and reactive power pairs. Differently from [21], the entire procedure for obtaining the bivariate HMM appliance models is described. The parameters are estimated by clustering the appliance disaggregate signals and the bivariate optimisation problem is solved by proposing an alternative formulation of AFAMAP [22] for disaggregating appliances consumption profiles. The proposed approach differs from the one presented in Section 4.2, since there the reactive power was employed alone in an initial disaggregation stage whose output served as a constraint for the subsequent disaggregation of the active power only. The proposed approach has been compared to the original AFAMAP algorithm [22], which employs the active power only, and to Hart's algorithm [16], which employs both the active and reactive power. Two implementations of Hart's algorithm have been developed for dealing with the occurrence multiple appliance combinations: in the first, the final combination is selected randomly. In the second, it is selected by choosing the most probable combination calculated on a training set. The experiments have been conducted on the Almanac of Minutely Power dataset (AMPds) [58], containing recordings of consumption profiles belonging to a single home for a period of two years at 1 minute sampling rate. Both the "noised" and the "denoised" scenarios have been addressed, and the results show that the proposed approach outperforms both AFAMAP and Hart's algorithm.

Finally, in [21] the experiments are conducted on low-power appliances only in a "denoised scenario", while here the "noised" is also considered.

In the following, the superscript (i) denotes terms related to HMM i, while subscripts a or r denote terms related to the active and reactive power component, respectively. The subscript $c \in \{a, r\}$ denotes a term related to the active or to the reactive power component. The parameters of the problem are the following:

- $N \in \mathbb{Z}_+$ is the number of HMMs in the system;
- $\overline{\boldsymbol{y}}(\tau) \in \mathbb{R}^n$ is the observed aggregate output, where $\tau = 1, 2, \ldots, \Upsilon$ is the sample index and Υ is the total number of samples;
- $\boldsymbol{\Sigma}_1 \in \mathbb{R}^{n \times n}$ is the observation covariance matrix related to the AFHMM;

- $\boldsymbol{\Sigma}_2 \in \mathbb{R}^{n\times n}$ is the observation covariance matrix related to the DFHMM;
- $\Delta\overline{\boldsymbol{y}}(\tau) = \overline{\boldsymbol{y}}(\tau) - \overline{\boldsymbol{y}}(\tau-1)$ is the differential signal.

As aforementioned, all the contribution to the aggregated power are considered, thus:

$$\overline{\boldsymbol{y}}(\tau) = \sum_{i=1}^{N} \boldsymbol{y}^{(i)}(\tau), \tag{4.17}$$

where $\boldsymbol{y}^{(i)}(\tau)$ corresponds to the ground truth consumption of the appliances and the *noise*. Recalling the notation of Chapter 4, the parameters of the i-th HMM at the sample index τ are:

- $m_i \in \mathbb{Z}_+$ is the number of states;
- $x^{(i)}(\tau) \in \{1,\dots,m_i\}$ is the HMM state at time instant τ, where m_i corresponds to the OFF state (if present);
- $\boldsymbol{\mu}_j^{(i)}$ is the emitted symbol in the j-th state, where $j = 1,2,\dots,m_i$;
- $\boldsymbol{\phi}^{(i)} \in [0,1]^{m_i}$ is the initial states probability distribution;
- $\boldsymbol{P}^{(i)} \in [0,1]^{m_i\times m_i}$ is the state transition probability matrix.

The aggregate signal $\overline{\boldsymbol{y}}(\tau)$ is analysed using non-overlapping frames of length T. Each frame $\overline{\boldsymbol{y}}_f(\tau)$, where $f = 1,2,\dots,F$, is defined as

$$\overline{\boldsymbol{y}}_f(\tau) = \begin{cases} \overline{\boldsymbol{y}}(\tau) & \text{if } \tau = (f-1)T+1,\dots,fT, \\ 0 & \text{otherwise.} \end{cases} \tag{4.18}$$

After the analysis of all the $F = \Upsilon/T$ frames, the disaggregated signals $\hat{\boldsymbol{y}}^{(i)}(\tau)$ are reconstructed as follows:

$$\hat{\boldsymbol{y}}^{(i)}(\tau) = \sum_{f=1}^{F} \hat{\boldsymbol{y}}_f^{(i)}(\tau). \tag{4.19}$$

In the following, the algorithm is formulated for a single frame of the signal and for convenience, a new temporal variable t is defined with the relation $t = \tau - (f-1)T$, for $t = 1,2,\dots,T$, with $T \in \mathbb{Z}_+$.

In [22], the parameter n defines the problem dimensionality: the authors use only the active power data to characterise the observed aggregated signal $\overline{y}_a(t)$, therefore they assumed $n = 1$. In this work, both the active and the reactive power are used for disaggregation, therefore $n = 2$ and the problem variables are decomposed in two components:

$$\overline{\boldsymbol{y}}_f(t) = \begin{bmatrix} \overline{y}_{a,f}(t) \\ \overline{y}_{r,f}(t) \end{bmatrix}, \quad \boldsymbol{\mu}_j^{(i)} = \begin{bmatrix} \mu_{a,j}^{(i)} \\ \mu_{r,j}^{(i)} \end{bmatrix}, \tag{4.20}$$

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} \sigma_{a,1}^2 & \sigma_{a,r,1} \\ \sigma_{r,a,1} & \sigma_{r,1}^2 \end{bmatrix}, \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} \sigma_{a,2}^2 & \sigma_{a,r,2} \\ \sigma_{r,a,2} & \sigma_{r,2}^2 \end{bmatrix}. \tag{4.21}$$

Since the statistical independence between the active and reactive power components is supposed, the covariance terms $\sigma_{a,r}$ and $\sigma_{r,a}$ are zero in both $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$, and the same problem formalisation as the $n = 1$ case can be used, introducing additional variables and constraining them each other. For the generic power component c, the variables in the optimisation problem are defined as follows:

$$\mathcal{Q}_c = \left\{ \boldsymbol{Q}_c(x^{(i)}(t)) \in \mathbb{R}^{m_i}, \boldsymbol{Q}_c(x^{(i)}(t-1), x^{(i)}(t)) \in \mathbb{R}^{m_i \times m_i} \right\}. \tag{4.22}$$

In the vector $\boldsymbol{Q}_c(x^{(i)}(t))$, the element $Q_c(x^{(i)}(t))_j$ indicates the state assumed at time instant t, while in the matrix $\boldsymbol{Q}_c(x^{(i)}(t-1), x^{(i)}(t))$
the element $Q_c(x^{(i)}(t-1), x^{(i)}(t))_{jk}$ indicates the state transition from previous to the current time instant.

This problem statement is a reformulated version of the algorithm proposed in [22]: since the original algorithm allows to operates with multivariate dimension, the variables associated to the state represent all the components. When only one dimension is considered, the variables $\mathcal{Q}_a$ is only associated at the active power level consumption. This problem statement instead, started from the univariate formulation, and the algorithm is extended to $n = 2$ by using twice the optimisation variables, thus introducing the $\mathcal{Q}_r$ variable set, and an additional minimisation function. Moreover, the supplementary variables need to be constrained to the original ones in order to assume the same value during the optimisation process, representing the bivariate resolution problem with a univariate problem formalisation:

$$\begin{cases} Q_a(x^{(i)}(t))_j - Q_r(x^{(i)}(t))_j = 0, \\ Q_a(x^{(i)}(t-1), x^{(i)}(t))_{jk} - Q_r(x^{(i)}(t-1), x^{(i)}(t))_{jk} = 0. \end{cases} \tag{4.23}$$

A numerically safer definition of the constraints can be defined using a tolerance α and inequalities:

$$\begin{cases} -\alpha \leq Q_a(x^{(i)}(t))_j - Q_r(x^{(i)}(t))_j \leq \alpha, \\ -\alpha \leq Q_a(x^{(i)}(t-1), x^{(i)}(t))_{jk} - Q_r(x^{(i)}(t-1), x^{(i)}(t))_{jk} \leq \alpha, \end{cases} \tag{4.24}$$

where $j, k = 1, \ldots, m_i$.

The final algorithm is shown in Algorithm 1. In eq. (4.25), the error terms

Algorithm 1 The proposed disaggregation algorithm.

1: **Input**:

- $\overline{\boldsymbol{y}}_f(t)$, for $t = 1, 2, \ldots, T$;
- $\left\{\boldsymbol{\mu}^{(i)}, \boldsymbol{P}^{(i)}, \boldsymbol{\phi}^{(i)}\right\}$, for $i = 1, 2, \ldots, N$;
- $\sigma^2_{c,1}$, $\sigma^2_{c,2}$;
- λ: regularisation parameter, described in [22].

2: **Minimise over** $\{\mathcal{Q}_c \in \mathcal{L}_c \cap \mathcal{O}_c\}$

$$\sum_{c\in\{a,r\}} \left\{ \frac{1}{2\sigma^2_{c,1}} \sum_{t=1}^{T} E'_c(t) + \frac{1}{2\sigma^2_{c,2}} \sum_{t=2}^{T} E''_c(t) + \frac{1}{2} \sum_{t=2}^{T} E'''_c(t) + \right.$$
$$+ \sum_{t=2}^{T} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ k=1}}^{m_i} \left\{ Q_c(x^{(i)}(t-1), x^{(i)}(t))_{jk} \left(-\log P^{(i)}_{kj} \right) \right\} +$$
$$\left. + \sum_{i=1}^{N} \sum_{j=1}^{m_i} \left\{ Q_c(x^{(i)}(1))_j (-\log \phi^{(i)}_j) \right\} \right\} \tag{4.25}$$

3: **Output**:

$$\hat{y}^{(i)}_{c,f}(t) = \sum_{j=1}^{m_i} \mu^{(i)}_{c,j} Q_c(x^{(i)}(t))_j \tag{4.26}$$

where $i = 1, 2, \ldots, N$ and $t = 1, 2, \ldots, T$.

are defined as:

$$E'_c(t) = \left(\overline{y}_{c,f}(t) - \sum_{i=1}^{N} \sum_{j=1}^{m_i} \mu^{(i)}_{c,j} Q_c(x^{(i)}(t))_j \right)^2, \tag{4.27}$$

$$E''_c(t) = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ k=1 \\ k\neq j}}^{m_i} \left\{ \left(\Delta\overline{y}_{c,f}(t) - \Delta\mu^{(i)}_{c,kj} \right)^2 Q_c(x^{(i)}(t-1), x^{(i)}(t))_{jk} \right\}, \tag{4.28}$$

$$E'''_c(t) = D\left(\frac{\Delta\overline{y}_{c,f}(t)}{\sigma_{c,2}}, \lambda \right) \left(1 - \sum_{i=1}^{N} \sum_{\substack{j=1 \\ k=1 \\ k\neq j}}^{m_i} Q_c(x^{(i)}(t-1), x^{(i)}(t))_{jk} \right). \tag{4.29}$$

The QP optimisation problem is defined as follows:

Minimise

$$\frac{1}{2} \boldsymbol{v}^T \boldsymbol{H} \boldsymbol{v} + \boldsymbol{f}^T \boldsymbol{v}, \tag{4.30}$$

subjet to the constraints:

$$\boldsymbol{A}_{eq}\,\boldsymbol{v} = \boldsymbol{b}_{eq}, \tag{4.31}$$

$$\boldsymbol{lb} \leq \boldsymbol{v} \leq \boldsymbol{ub}. \tag{4.32}$$

The variables of the problem are represented by the vector $\boldsymbol{v} = [\boldsymbol{v}_a\, \boldsymbol{v}_r]^T$ whose components are defined as follows:

$$\boldsymbol{v}_c = \begin{bmatrix} \boldsymbol{\Theta}(1) \\ \vdots \\ \boldsymbol{\Theta}(T) \end{bmatrix},\ \boldsymbol{\Theta}(t) = \begin{bmatrix} \boldsymbol{\Psi}^{(1)}(t) \\ \vdots \\ \boldsymbol{\Psi}^{(N)}(t) \end{bmatrix},\ \boldsymbol{\Psi}^{(i)}(t) = \begin{bmatrix} \boldsymbol{\xi}^{(i)}(t) \\ \boldsymbol{\beta}^{(i)}(t) \end{bmatrix}, \tag{4.33}$$

$$\boldsymbol{\xi}^{(i)}(t) = \begin{bmatrix} Q_c(x^{(i)}(t))_1 \\ \vdots \\ Q_c(x^{(i)}(t))_{m_i} \end{bmatrix}, \quad \boldsymbol{\beta}^{(i)}(t) = \begin{bmatrix} Q_c(x^{(i)}(t-1), x^{(i)}(t))_{1\,1} \\ \vdots \\ Q_c(x^{(i)}(t-1), x^{(i)}(t))_{1\,m_i} \\ \vdots \\ Q_c(x^{(i)}(t-1), x^{(i)}(t))_{m_i\,1} \\ \vdots \\ Q_c(x^{(i)}(t-1), x^{(i)}(t))_{m_i\,m_i} \end{bmatrix}, \tag{4.34}$$

where the variables for the state are represented in $\boldsymbol{\xi}^{(i)}(t)$, and the variables for the transition in $\boldsymbol{\beta}^{(i)}(t)$.

The parameters of the problem, e.g., the HMMs parameters and the aggregated power signal, compose the elements of $\boldsymbol{H}$ and $\boldsymbol{f}$, according to the structure of the $\boldsymbol{v}$ vector. In a QP problem, the coefficient of the quadratic terms in the cost function are defined in $\boldsymbol{H}$, as a symmetric matrix. In the proposed approach, since the independence between the active and reactive power is assumed, there are no joint quadratic terms, therefore $\boldsymbol{H}$ is structured as follow:

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_a & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{H}_r \end{bmatrix}. \tag{4.35}$$

Differently, the coefficients of the linear terms are expressed in $\boldsymbol{f} = [\boldsymbol{f}_a\ \boldsymbol{f}_r]^T$. Whereas $\boldsymbol{A}_{eq}$ and $\boldsymbol{b}_{eq}$ are used to represent the consistent constraints between the state and the transition variables. The vectors $\boldsymbol{lb}$ and $\boldsymbol{ub}$ define the lower and upper boundaries of the solution: because of the nature of the variables [22], the lower boundary is equal to 0, whereas the upper boundary to 1, for all the elements in $\boldsymbol{v}$.

Additional constraints to QP problem need to be considered, in order to impose the inequality constraints between the optimisation variables. Duplicating

the constraints of eq. (4.24):

$$\begin{cases} -\alpha \leq Q_a(x^{(i)}(t))_j - Q_r(x^{(i)}(t))_j, \\ Q_a(x^{(i)}(t))_j - Q_r(x^{(i)}(t))_j \leq \alpha, \end{cases} \tag{4.36}$$

$$\begin{cases} -\alpha \leq Q_a(x^{(i)}(t-1), x^{(i)}(t))_{jk} - Q_r(x^{(i)}(t-1), x^{(i)}(t))_{jk}, \\ Q_a(x^{(i)}(t-1), x^{(i)}(t))_{jk} - Q_r(x^{(i)}(t-1), x^{(i)}(t))_{jk} \leq \alpha, \end{cases} \tag{4.37}$$

results in the following optimisation constraint:

$$\boldsymbol{A}_{ineq}\, \boldsymbol{v} \leq \boldsymbol{b}_{ineq}. \tag{4.38}$$

This is needed only for the joint active-reactive problem, since, solving only for the active power, the related unique variable is not constrained to other variables. Indeed, in eq. (4.25) only the the active power terms need to be considered. Further details on the terms $\boldsymbol{H}$, $\boldsymbol{f}$, $\boldsymbol{A}_{eq}$, $\boldsymbol{b}_{eq}$, $\boldsymbol{lb}$, $\boldsymbol{ub}$, $\boldsymbol{A}_{ineq}$, and $\boldsymbol{b}_{ineq}$ are provided in 4.3.1.

As aforementioned, the aggregate signal is analysed in frames of length T. In the first frame, the value of starting probability vector $\boldsymbol{\phi}^{(i)} = [0\ 0\ \cdots\ 0\ 1]$, i.e., the appliance is initially assumed in the OFF state. In the subsequent frames, the value of $\boldsymbol{\phi}^{(i)}$ depends on the last state assumed in the previous frame in order to ensure the contiguity of the solution at the border. Thus, if the last state assumed in the previous frame is j, the corresponding element of $\boldsymbol{\phi}^{(i)}$ is set to 1, while the others are set to 0. This information is represented by the value of the solution $\boldsymbol{\xi}^{(i)}(t)$ in the last sample $t = T$.

4.3.1 AFAMAP formulation

This subsection provides further details on the algorithm formulation presented in Section 4.3. In particular, the following terms of the QP problem are described: $\boldsymbol{H}$, $\boldsymbol{f}$, $\boldsymbol{A}_{eq}$, $\boldsymbol{b}_{eq}$, $\boldsymbol{lb}$, $\boldsymbol{ub}$, $\boldsymbol{A}_{ineq}$, and $\boldsymbol{b}_{ineq}$.
The matrix $\boldsymbol{H}$ is structured as follows:

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_a & \mathbf{0} \\ \mathbf{0} & \boldsymbol{H}_r \end{bmatrix}, \tag{4.39}$$

where $\boldsymbol{H}_c \in \{\boldsymbol{H}_a, \boldsymbol{H}_r\}$ is given by:

$$\boldsymbol{H}_c = \begin{bmatrix} \frac{1}{\sigma^2_{c,1}} \boldsymbol{H}_{\boldsymbol{\Theta}(1)} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \frac{1}{\sigma^2_{c,1}} \boldsymbol{H}_{\boldsymbol{\Theta}(T)} \end{bmatrix}, \quad \boldsymbol{H}_{\boldsymbol{\Theta}(t)} = \begin{bmatrix} \boldsymbol{H}_{\boldsymbol{\Psi}(1\,1)(t)} & \cdots & \boldsymbol{H}_{\boldsymbol{\Psi}(1\,N)(t)} \\ \vdots & \ddots & \vdots \\ \boldsymbol{H}_{\boldsymbol{\Psi}(N\,1)(t)} & \cdots & \boldsymbol{H}_{\boldsymbol{\Psi}(N\,N)(t)} \end{bmatrix}, \tag{4.40}$$

and

$$\boldsymbol{H}_{\boldsymbol{\Psi}^{(i\,j)}(t)} = \begin{bmatrix} \boldsymbol{H}_{\boldsymbol{\xi}^{(i\,j)}(t)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \boldsymbol{H}_{\boldsymbol{\xi}^{(i\,j)}(t)} = \begin{bmatrix} \mu_{c,1}^{(i)}\,\mu_{c,1}^{(j)} & \cdots & \mu_{c,1}^{(i)}\,\mu_{c,m_j}^{(j)} \\ \vdots & \ddots & \vdots \\ \mu_{c,m_i}^{(i)}\,\mu_{c,1}^{(j)} & \cdots & \mu_{c,m_i}^{(i)}\,\mu_{c,m_j}^{(j)} \end{bmatrix}. \tag{4.41}$$

Regarding the vector $\boldsymbol{f}$, in Section 4.3 it has been defined as follows:

$$\boldsymbol{f} = \begin{bmatrix} \boldsymbol{f}_a & \boldsymbol{f}_r \end{bmatrix}^T, \tag{4.42}$$

where $\boldsymbol{f}_c \in \{\boldsymbol{f}_a, \boldsymbol{f}_r\}$ is given by the sum of five terms:

$$\boldsymbol{f}_c = -\boldsymbol{f}_{c,1} - \frac{1}{\sigma_{c,1}^2}\boldsymbol{f}_{c,2} - \boldsymbol{f}_{c,3} + \frac{1}{2}\frac{1}{\sigma_{c,2}^2}\boldsymbol{f}_{c,4} - \frac{1}{2}\boldsymbol{f}_{c,5}, \tag{4.43}$$

where

$$\boldsymbol{f}_{c,1} = \begin{bmatrix} \boldsymbol{f}_{1,\boldsymbol{\Theta}(1)} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}, \ \boldsymbol{f}_{1,\boldsymbol{\Theta}(1)} = \begin{bmatrix} \boldsymbol{f}_{1,\boldsymbol{\Psi}^{(1)}(1)} \\ \vdots \\ \boldsymbol{f}_{1,\boldsymbol{\Psi}^{(N)}(1)} \end{bmatrix}, \ \boldsymbol{f}_{1,\boldsymbol{\Psi}^{(i)}(1)} = \begin{bmatrix} \boldsymbol{f}_{1,\boldsymbol{\xi}^{(i)}(1)} \\ \mathbf{0} \end{bmatrix}, \boldsymbol{f}_{1,\boldsymbol{\xi}^{(i)}(1)} = \begin{bmatrix} \log \phi_1^{(i)} \\ \vdots \\ \log \phi_{m_i}^{(i)} \end{bmatrix}, \tag{4.44}$$

$$\boldsymbol{f}_{c,2} = \begin{bmatrix} \boldsymbol{f}_{2,\boldsymbol{\Theta}(1)} \\ \vdots \\ \boldsymbol{f}_{2,\boldsymbol{\Theta}(T)} \end{bmatrix}, \ \boldsymbol{f}_{2,\boldsymbol{\Theta}(t)} = \begin{bmatrix} \boldsymbol{f}_{2,\boldsymbol{\Psi}^{(1)}(t)} \\ \vdots \\ \boldsymbol{f}_{2,\boldsymbol{\Psi}^{(N)}(t)} \end{bmatrix}, \ \boldsymbol{f}_{2,\boldsymbol{\Psi}^{(i)}(t)} = \begin{bmatrix} \boldsymbol{f}_{2,\boldsymbol{\xi}^{(i)}(t)} \\ \mathbf{0} \end{bmatrix}, \boldsymbol{f}_{2,\boldsymbol{\xi}^{(i)}(t)} = \begin{bmatrix} \overline{y}_{c,f}(t)\,\mu_{c,1}^{(i)} \\ \vdots \\ \overline{y}_{c,f}(t)\,\mu_{c,m_i}^{(i)} \end{bmatrix}, \tag{4.45}$$

$$\boldsymbol{f}_{c,3} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{f}_{3,\boldsymbol{\Theta}(2)} \\ \vdots \\ \boldsymbol{f}_{3,\boldsymbol{\Theta}(T)} \end{bmatrix}, \ \boldsymbol{f}_{3,\boldsymbol{\Theta}(t)} = \begin{bmatrix} \boldsymbol{f}_{3,\boldsymbol{\Psi}^{(1)}(t)} \\ \vdots \\ \boldsymbol{f}_{3,\boldsymbol{\Psi}^{(N)}(t)} \end{bmatrix}, \ \boldsymbol{f}_{3,\boldsymbol{\Psi}^{(i)}(t)} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{f}_{3,\boldsymbol{\beta}^{(i)}(t)} \end{bmatrix}, \boldsymbol{f}_{3,\boldsymbol{\beta}^{(i)}(t)} = \begin{bmatrix} \log P_{1\,1}^{(i)} \\ \vdots \\ \log P_{m_i\,1}^{(i)} \\ \vdots \\ \log P_{1\,m_i}^{(i)} \\ \vdots \\ \log P_{m_i\,m_i}^{(i)} \end{bmatrix}, \tag{4.46}$$

$$\boldsymbol{f}_{c,4} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{f}_{4,\boldsymbol{\Theta}(2)} \\ \vdots \\ \boldsymbol{f}_{4,\boldsymbol{\Theta}(T)} \end{bmatrix}, \ \boldsymbol{f}_{4,\boldsymbol{\Theta}(t)} = \begin{bmatrix} \boldsymbol{f}_{4,\boldsymbol{\Psi}^{(1)}(t)} \\ \vdots \\ \boldsymbol{f}_{4,\boldsymbol{\Psi}^{(N)}(t)} \end{bmatrix}, \ \boldsymbol{f}_{4,\boldsymbol{\Psi}^{(i)}(t)} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{f}_{4,\boldsymbol{\beta}^{(i)}(t)} \end{bmatrix}, \tag{4.47}$$

$$\boldsymbol{f}_{4,\boldsymbol{\beta}^{(i)}(t)} = \begin{bmatrix} k^{(i)}_{1\,1}(t) \\ \vdots \\ k^{(i)}_{m_i\,1}(t) \\ \vdots \\ k^{(i)}_{1\,m_i}(t) \\ \vdots \\ k^{(i)}_{m_i\,m_i}(t) \end{bmatrix}, \tag{4.48}$$

$$\boldsymbol{k}^{(i)}(t) = \begin{bmatrix} 0 & \cdots & \left(\Delta\overline{y}_{c,f}(t) - \left(\mu^{(i)}_{c,1} - \mu^{(i)}_{c,m_i}\right)\right)^2 \\ \vdots & \ddots & \vdots \\ \left(\Delta\overline{y}_{c,f}(t) - \left(\mu^{(i)}_{c,m_i} - \mu^{(i)}_{c,1}\right)\right)^2 & \cdots & 0 \end{bmatrix} \tag{4.49}$$

$$\boldsymbol{f}_{c,5} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{f}_{5,\boldsymbol{\Theta}(2)} \\ \vdots \\ \boldsymbol{f}_{5,\boldsymbol{\Theta}(T)} \end{bmatrix}, \ \boldsymbol{f}_{5,\boldsymbol{\Theta}(t)} = \begin{bmatrix} \boldsymbol{f}_{5,\boldsymbol{\Psi}^{(1)}(t)} \\ \vdots \\ \boldsymbol{f}_{5,\boldsymbol{\Psi}^{(N)}(t)} \end{bmatrix}, \ \boldsymbol{f}_{5,\boldsymbol{\Psi}^{(i)}(t)} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{f}_{5,\boldsymbol{\beta}^{(i)}(t)} \end{bmatrix}, \tag{4.50}$$

$$\boldsymbol{f}_{5,\boldsymbol{\beta}^{(i)}(t)} = \begin{bmatrix} d^{(i)}_{1\,1}(t) \\ \vdots \\ d^{(i)}_{m_i\,1}(t) \\ \vdots \\ d^{(i)}_{1\,m_i}(t) \\ \vdots \\ d^{(i)}_{m_i\,m_i}(t) \end{bmatrix}, \boldsymbol{d}^{(i)}(t) = \begin{bmatrix} 0 & \cdots & D\left(\frac{\Delta\overline{y}_{c,f}(t)}{\sigma_{c,2}}, \lambda\right) \\ \vdots & \ddots & \vdots \\ D\left(\frac{\Delta\overline{y}_{c,f}(t)}{\sigma_{c,2}}, \lambda\right) & \cdots & 0 \end{bmatrix}, \tag{4.51}$$

where:

$$D(y, \lambda) = \min\left\{\frac{1}{2}y^2, \max\left\{\lambda|y| - \frac{\lambda^2}{2}, \frac{\lambda^2}{2}\right\}\right\}. \tag{4.52}$$

The matrix $\boldsymbol{A}_{eq}$ is defined as follows:

$$\boldsymbol{A}_{eq} = \begin{bmatrix} \boldsymbol{A}_{eq,a} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{A}_{eq,r} \end{bmatrix}, \boldsymbol{A}_{eq,c} = \begin{bmatrix} \boldsymbol{A}_{eq,\boldsymbol{\Theta}(1)} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \boldsymbol{A}_{eq,\boldsymbol{\Theta}(T)} \end{bmatrix}, \tag{4.53}$$

$$\boldsymbol{A}_{eq,\boldsymbol{\Theta}(t)} = \begin{bmatrix} \boldsymbol{A}_{eq,\boldsymbol{\Psi}^{(1)}_1(t)} & \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{A}_{eq,\boldsymbol{\Psi}^{(1)}_2(t)} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{A}_{eq,\boldsymbol{\Psi}^{(N)}_1(t)} & \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{A}_{eq,\boldsymbol{\Psi}^{(N)}_2(t)} \end{bmatrix}, \tag{4.54}$$

$$\boldsymbol{A}_{eq,\boldsymbol{\Psi}^{(i)}_1(t)} = \begin{bmatrix} \boldsymbol{A}_{eq,\boldsymbol{\xi}^{(i)}_1(t)} \\ \boldsymbol{A}_{eq,\boldsymbol{\beta}^{(i)}_{1b}(t)} \\ \boldsymbol{A}_{eq,\boldsymbol{\beta}^{(i)}_{1f}(t)} \end{bmatrix}, \boldsymbol{A}_{eq,\boldsymbol{\xi}^{(i)}_1(t)} = \begin{bmatrix} 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix}, \tag{4.55}$$

$$\boldsymbol{A}_{eq,\boldsymbol{\beta}_{1b}^{(i)}(t)} = \begin{bmatrix} -1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & -1 & 0 & \cdots & 0 \end{bmatrix}, \boldsymbol{A}_{eq,\boldsymbol{\beta}_{1f}^{(i)}(t)} = \begin{bmatrix} 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{4.56}$$

$$\boldsymbol{A}_{eq,\boldsymbol{\Psi}_{2}^{(i)}(t)} = \begin{bmatrix} \boldsymbol{A}_{eq,\boldsymbol{\xi}_{2}^{(i)}(t)} \\ \boldsymbol{A}_{eq,\boldsymbol{\beta}_{2b}^{(i)}(t)} \\ \boldsymbol{A}_{eq,\boldsymbol{\beta}_{2f}^{(i)}(t)} \end{bmatrix}, \boldsymbol{A}_{eq,\boldsymbol{\xi}_{2}^{(i)}(t)} = \begin{bmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 \end{bmatrix}, \tag{4.57}$$

$$\boldsymbol{A}_{eq,\boldsymbol{\beta}_{2b}^{(i)}(t)} = \begin{bmatrix} 0 & \cdots & 0 & 1 & \cdots & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 1 & \cdots & 1 \end{bmatrix}, \tag{4.58}$$

$$\boldsymbol{A}_{eq,\boldsymbol{\beta}_{2f}^{(i)}(t)} = \begin{bmatrix} -1 & \cdots & 0 & 1 & \cdots & 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & -1 & 0 & \cdots & 1 & \cdots & 0 & \cdots & 1 \end{bmatrix}, \tag{4.59}$$

The vector $\boldsymbol{b}_{eq}$ has the following form:

$$\boldsymbol{b}_{eq} = \begin{bmatrix} \boldsymbol{b}_{eq,a} & \boldsymbol{b}_{eq,r} \end{bmatrix}^T, \tag{4.60}$$

$$\boldsymbol{b}_{eq,c} = \begin{bmatrix} \boldsymbol{b}_{eq,\boldsymbol{\Theta}(1)} \\ \vdots \\ \boldsymbol{b}_{eq,\boldsymbol{\Theta}(T)} \end{bmatrix}, \boldsymbol{b}_{eq,\boldsymbol{\Theta}(t)} = \begin{bmatrix} \boldsymbol{b}_{eq,\boldsymbol{\Psi}^{(1)}(t)} \\ \vdots \\ \boldsymbol{b}_{eq,\boldsymbol{\Psi}^{(N)}(t)} \end{bmatrix}, \boldsymbol{b}_{eq,\boldsymbol{\Psi}^{(i)}(t)} = \begin{bmatrix} \boldsymbol{b}_{eq,\boldsymbol{\xi}^{(i)}(t)} \\ \boldsymbol{b}_{eq,\boldsymbol{\beta}^{(i)}(t)} \end{bmatrix}, \tag{4.61}$$

$$\boldsymbol{b}_{eq,\boldsymbol{\xi}^{(i)}(t)} = \begin{bmatrix} 1 \end{bmatrix}, \quad \boldsymbol{b}_{eq,\boldsymbol{\beta}^{(i)}(t)} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{4.62}$$

$$\boldsymbol{lb} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}, \ \boldsymbol{ub} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \tag{4.63}$$

The matrix $\boldsymbol{A}_{ineq}$ is given by:

$$\boldsymbol{A}_{ineq} = \begin{bmatrix} \boldsymbol{A}_{ineq,\boldsymbol{\Theta}(1)} \\ \vdots \\ \boldsymbol{A}_{ineq,\boldsymbol{\Theta}(T)} \end{bmatrix}, \quad \boldsymbol{A}_{ineq,\boldsymbol{\Theta}(t)} = \begin{bmatrix} \boldsymbol{A}_{ineq,\boldsymbol{\Psi}^{(1)}(t)} \\ \vdots \\ \boldsymbol{A}_{ineq,\boldsymbol{\Psi}^{(N)}(t)} \end{bmatrix}, \tag{4.64}$$

$$\boldsymbol{A}_{ineq,\boldsymbol{\Psi}^{(i)}(t)} = \begin{bmatrix} \boldsymbol{A}_{ineq,\boldsymbol{\Psi}^{(i)}(t),a} & \boldsymbol{A}_{ineq,\boldsymbol{\Psi}^{(i)}(t),r} \end{bmatrix}, \quad \boldsymbol{A}_{ineq,\boldsymbol{\Psi}^{(i)}(t),r} = -\boldsymbol{A}_{ineq,\boldsymbol{\Psi}^{(i)}(t),a} \tag{4.65}$$

$$\boldsymbol{A}_{ineq,\boldsymbol{\Psi}^{(i)}(t),a} = \begin{bmatrix} 0 & \cdots & 0 & 1 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & -1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & & & \\ 0 & \cdots & 0 & 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & -1 & 0 & \cdots & 0 \end{bmatrix} \tag{4.66}$$

Finally, the vector $\boldsymbol{b}_{ineq}$ is given by:

$$\boldsymbol{b}_{ineq} = \begin{bmatrix} \boldsymbol{b}_{ineq,\boldsymbol{\Theta}(1)} \\ \vdots \\ \boldsymbol{b}_{ineq,\boldsymbol{\Theta}(T)} \end{bmatrix}, \; \boldsymbol{b}_{ineq,\boldsymbol{\Theta}(t)} = \begin{bmatrix} \boldsymbol{b}_{ineq,\boldsymbol{\Psi}^{(1)}(t)} \\ \vdots \\ \boldsymbol{b}_{ineq,\boldsymbol{\Psi}^{(N)}(t)} \end{bmatrix}, \; \boldsymbol{b}_{ineq,\boldsymbol{\Psi}^{(i)}(t)} = \begin{bmatrix} \alpha \\ \vdots \\ \alpha \end{bmatrix} \tag{4.67}$$

As described in Section 4.3, the dimensionality of the variables vector and, accordingly, of each elements of the QP problem is defined as follows:

- $\boldsymbol{v}_c$: l-dimensional vector;
- $\boldsymbol{H}_c$: $[l \times l]$ symmetric matrix;
- $\boldsymbol{f}_c$: l-dimensional vector;
- $\boldsymbol{A}_{eq,c}$: $[m \times l]$ matrix;
- $\boldsymbol{b}_{eq,c}$: m-dimensional vector;
- $\boldsymbol{lb}, \boldsymbol{ub}$: $2\,l$ -dimensional vector;
- $\boldsymbol{A}_{ineq}$: $[2\,l \times 2\,l]$ matrix;
- $\boldsymbol{b}_{ineq}$: $2\,l$-dimensional vector;

where $l = T \cdot \sum_{i=1}^{N}(m_i + m_i^2)$ and $m = T \cdot \sum_{i=1}^{N}(1 + 2m_i)$.

4.3.2 Experimental setup

The proposed approach has been compared with the algorithm presented by Hart in [16], since it employs both the active and the reactive power to model the appliance working behaviour and it employs those electrical parameters for disaggregation. This section provides an overview of its basic operating principles as well as additional details on its implementation. In addition, the algorithm originally presented in [16] has been improved for handling the occurrence of multiple solutions by means of a MAP technique.

Hart's algorithm models each appliance as a Finite State Machine (FSM). Each FSM is represented by the following parameters:

- the number of states $m \in \mathbb{Z}_+$;
- the finite states $x \in \{1, 2, \ldots, m\}$;
- the symbols emitted $\boldsymbol{\mu}_j \in \mathbb{R}^n$, where $j = 1, \ldots, m$;
- state transition matrix $\boldsymbol{T} \in \{0, 1\}^{m \times m}$.

As in the proposed approach, each state of the FSM corresponds to a working state of the appliance and $n = 2$, i.e., the symbol emitted in the j-th state is defined as $\boldsymbol{\mu}_j = [\mu_{a,j}\, \mu_{r,j}]^T$. A tolerance parameter $\boldsymbol{\beta}_j = [\beta_{a,j}\, \beta_{r,j}]^T$ is associated to the emitted symbol in the j-th state, in order to define the effectiveness interval for the emitted symbol. The interval width is $2\,\boldsymbol{\beta}_j$ and it is centred in $\boldsymbol{\mu}_j$. For each appliance, the quantities to be estimated are the number of states m, the values of $\boldsymbol{\mu}_j$ and $\boldsymbol{\beta}_j$ for each state, and the state transition matrix $\boldsymbol{T}$.

In order to model the power consumption of an appliance as a stochastic process, under the assumption of multiple independent causes to the circuital power dissipation, the central limit theorem might be invoked. Therefore, the power consumption $\boldsymbol{y}^{(i)}(t)$ of the i-th appliance at time instant t, related to the working state $x^{(i)}(t)$, can be modelled as a bivariate Gaussian variable, described by a mean vector $\boldsymbol{\mu}_{x^{(i)}(t)}$ and a covariance matrix $\boldsymbol{\Sigma}_{x^{(i)}(t)}$:

$$\boldsymbol{y}^{(i)}(t)|x^{(i)}(t) \sim \mathcal{N}\left(\boldsymbol{\mu}_{x^{(i)}(t)}, \boldsymbol{\Sigma}_{x^{(i)}(t)}\right). \tag{4.68}$$

Following this approach, the consumption signal is replaced by a simplified model that represents a constant power consumption, corresponding to the mean value of the working state power value, with a superimposed noisy contribution, described by the variance value in the working state. Under the assumption of statistical independence between the active and reactive power components, the covariance matrix $\boldsymbol{\Sigma}_{x^{(i)}(t)}$ is diagonal:

$$\boldsymbol{\Sigma}_{x^{(i)}(t)} = \begin{bmatrix} \sigma^2_{a,x^{(i)}(t)} & 0 \\ 0 & \sigma^2_{r,x^{(i)}(t)} \end{bmatrix}, \tag{4.69}$$

where $\sigma^2_{a,x^{(i)}(t)}$ and $\sigma^2_{r,x^{(i)}(t)}$ represent respectively the variance of the active and reactive power in the cluster. The inference procedure is carried out independently for the two components. Therefore, at each state,

$$y^{(i)}_c(t)|x^{(i)}(t) \sim \mathcal{N}\left(\mu_{c,x^{(i)}(t)}, \sigma^2_{c,x^{(i)}(t)}\right). \tag{4.70}$$

The number of states m_i is defined in the clustering phase, described in Subsection 4.1.1, assuming that each cluster corresponds to a state in the FSM model: the estimation of the mean and the variance values for each component is performed with the Maximum Likelihood criterion on the clusters data. Each component of the tolerance parameter $\beta_{c,j}$, associated to the respective component of the emitted symbol $\mu_{c,j}$, is set equal to the standard deviation $\sigma_{c,j}$ of the Gaussian distribution.

Regarding the state transition matrix $\boldsymbol{T}$, each entry T_{ij} represents the admissibility of the transition from state i to state j, using the value $T_{ij} = 1$ if the transition is allowed and $T_{ij} = 0$ otherwise. This value is inferred from the

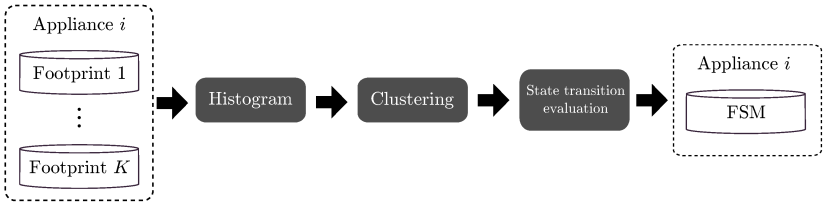


Figure 4.15: Block diagram of the clustering and of the model training stages of Hart's algorithm.

ground truth state evolution of each appliance consumption. Since this model does not represent the evolution in time of a signal, the permanence in the state is not represented, therefore the variable T_{ii} is set to 1. The diagram of the clustering and of the model training stage is shown in Figure 4.15.

Since the aggregated data $\overline{y}_c(t)$ is assumed to correspond with the sum of the power consumption of each appliance, it can be modelled as a Gaussian variable, described by a mean value and a variance value equivalent to the sum of the corresponding values of each appliance, under the assumption of statistical independence among the appliances:

$$\overline{y}_c(t)|x^{(1:N)}(t) \sim \mathcal{N}\left(\sum_{i=1}^{N} \mu_{c,x^{(i)}(t)}, \sum_{i=1}^{N} \sigma^2_{c,x^{(i)}(t)}\right). \tag{4.71}$$

This variable represents the Probability Density Function (PDF) of the working states combinations and it allows to evaluate which combination of working states fit the power value for each sample of the aggregated data. The number of admissible combinations of working states is equal to $\prod_{i=1}^{N} m_i$.

Following the same rule defined for each appliance symbol, the effectiveness interval for each combination is centred in mean value, and its width is twice the value of the standard deviation. For some combinations, which have similar mean value or great variance, the effectiveness intervals are overlapped: for those cases, if the power value falls in this region, both the combinations are considered valid.

The aggregate power data is analysed sample by sample: for each value, the effectiveness intervals in which the sample falls are selected. The related state combination might be admissible or not, depending on the previous state combination selected. Therefore, for each FSM, from the knowledge of the previous state selected, the admissible transition are evaluated through the transition matrix T_{ij}: the FSMs which do not make any variation in the state from the previous combination are not evaluated, then if the transition is not admissible for at least one FSM, the selected combination is discarded. The starting combination is evaluated on the first sample, without the evaluation

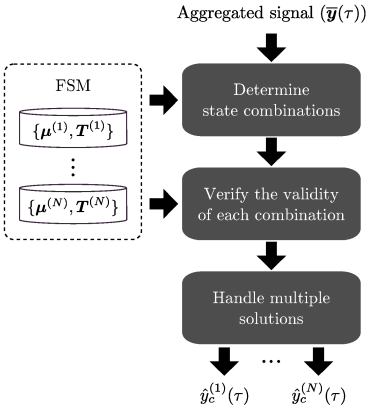


Figure 4.16: Diagram of the load disaggregation phase.

on the transition from any previous state. If no combination is admissible, the previous state in maintained for each FSM. If the aggregated data sample does not fall within any combination interval, the previous state is maintained for each FSM.

In this way, the time series of the state evolution is reconstructed for each FSM. The disaggregation consists in using the related power level consumption assigned to each state of the FSM, thus reconstructing the power consumption profile for each appliance. The general scheme of the disaggregation phase is shown in Figure 4.16.

In order to deal with the noise presence in the aggregated data, an FSM version of the *noise* model defined in Subsection 4.1.2 is considered, additionally to the FSM models representing the appliances.

In order to make a fair comparison of algorithms, representing an appliance, both kinds of model have the same number of states, values of power consumption and standard deviation of the gaussian variable. The values are resumed in Table 5.3.

In [16], the author did not describe the technique adopted for dealing with the occurrence of multiple solutions during the disaggregation phase. Two different approaches for dealing with the problem are adopted. The first consists in supposing that each combination of appliances is equally probable, thus the ambiguity is solved by choosing a random combination sampled from a uniform distribution. This algorithm will be denoted as "Hart" in the remainder of this dissertation.

A second approach, consists in adopting a MAP technique [58]: the *posterior probability* of each combination is calculated from the training data, and it is multiplied to the Gaussian PDF, resulting in the *posterior PDF*. The value of the posterior PDF in the aggregate data sample is denoted as the *posterior*

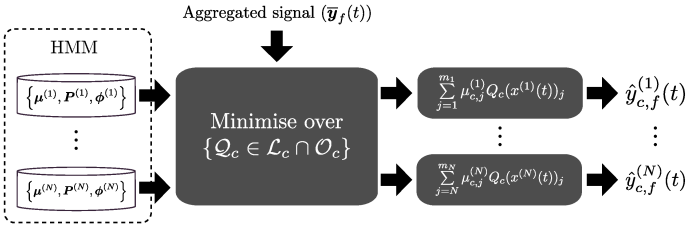


Figure 4.17: Diagram of the load disaggregation phase.

likelihood. The combination with the higher posterior likelihood value is then chosen as the most probable combination. This alternative of Hart's algorithm will be denoted as "Hart w/ MAP" in the remainder of this dissertation.

The general scheme of the disaggregation phase is shown in Figure 4.17. The algorithm is based on the work proposed by Kolter and Jaakkola [22], where the problem is modelled in the Additive Factorial Hidden Markov Model (AFHMM) framework.

Basically, this consists in modelling the value of each aggregated power sample as a combination of working states of the appliances. In [22], an assumption is made that at most one HMM changes its state at any given time, which holds true if the sampling time is reasonably short. In this case, a transition on the aggregate power, when moving from a sample to the next, corresponds to a state change of a particular HMM. As a consequence, a differential signal can be modelled as the result of a Differential Factorial Hidden Markov Model (DFHMM), which relies on the same HMM models composing the AFHMM. The DFHMM models the observation output as the difference between the states combination of the HMMs in two consecutive time instants. By combining the additive and differential models, the inference on the set of states of multiple HMMs can be computed through the Maximum A Posteriori (MAP) algorithm, which takes the form of a Mixed Integer Quadratic Programming (MIQP) optimisation problem. One of the shortcomings of this approach is the non-convex nature of the problem, due to the integer nature of the variables: therefore, a relaxation towards real values is taken into account, which allows the solution to assume any value in the range [0, 1], instead of the binary solution, leading to a convex Quadratic Programming (QP) optimisation problem.

In a real case scenario, the modelled output may not match with the observed aggregated signal, due to electrical noises, very small loads, or leakages. In that case, the issue is addressed by defining a robust mixture component both in the AFHMM and in the DFHMM. This component is missing in this dissertation, since all the contributions to the aggregated power are modelled. Indeed, each appliance and the *noise* is represented by its HMM.

Table 4.3: Number of states m_i related to each class of appliance.

Problem dimensionality	**Dryer**	**Washing machine**	**Dishwasher**	**Fridge**	**Electric oven**	**Heat pump**
Univariate	3	4	3	2	3	3
Bivariate	3	5	4	2	4	3

The dataset used for the experiments is the Almanac of Minutely Power dataset (AMPds) [58]: it contains recordings of consumption profiles belonging to a single home in Canada for a period of two years, at 1 minute sampling rate. Additionally to the aggregated power consumption, it provides active and reactive power at appliance level, unlike most of the dataset, in which the appliances consumption is described by the only active power, as showed in Section 2.3: this information is crucial in order to create the appliance models and test the new approach.

The experiments are conducted by using the six appliances which contribute the most to the power consumption: dryer, washing machine, dishwasher, fridge, electric oven, and heat pump. Regarding the significance of the reactive components of the appliances taken into consideration, the following values have been extracted from the datasets: (128.25 W, 7.96 VAR) for the fridge, (4545.91 W, 413.75 VAR) and (248.11 W, 408.94 VAR) for the dryer, (909.11 W, 203.44 VAR), (531.10 W, 14.37 VAR),(146.80 W, 3.60 VAR) and (137.54 W, 96.47 VAR) for the washing machine, (753.07 W, 33.31 VAR), (137.96 W, 35.86 VAR) and (14.42 W, 52.55 VAR) for the dishwasher, (3187.67 W, 136.63 VAR),(125.68 W, 121.67 VAR) and (89.54 W, 50.62 VAR) for the electric oven, (1798.83 W, 320.95 VAR) and (37.23 W, 17.03 VAR) for the heat pump. As shown by these values, the appliances evaluated in the experiments have a significant contribution of reactive power that make them suitable for evaluating the performance of the proposed approach. Analysing the contents of the dataset, the usage of the appliances proves to be homogeneous throughout the entire period, therefore the experiments are evaluated on 6 months of data, which can be considered representative of the entire dataset. A subset of the data, spanning over 14 days, has been considered sufficient to collect all the signatures required to train all the HMMs. This represents the training set in the Figure 4.5b.

Two different scenario are defined in this work, according to [85]. The *noised* scenario employs the aggregated power consumption in the dataset as the aggregated signal, therefore it includes the noise term. In this case, the training data used to create the *noise model* are obtained subtracting the ground truth consumption signals, related to the appliances of interest, from the aggregated power. Whereas, in the *denoised* scenario the aggregated data are synthetically composed by summing the ground truth appliance power signals in the dataset,

determining the absence of the noise term.

The proposed approach and Hart's algorithm are able to disaggregate both the active and the reactive power, however the performance metrics have been calculated on the active power only in order to compare it with the univariate formulation of AFAMAP. Furthermore, the active power is the physical quantity directly related to the cost in the bill, therefore it is the most relevant component to be analysed.

The frame size is set to $T = 60$ minutes, which is an interval sufficiently large to include a complete activation for the most of appliances under study. This value is considered within the *windowing* operation in the Figure 4.17, where the f-th frame is considered in the disaggregation. For the ones which have a longer activation, this value allows to include a complete operating sub cycle, for which the HMM is still representative. The variance parameters are set to $\sigma^2_{c,1} = \sigma^2_{c,2} = 0.01$ according to the variance of the experimental data, and the regularisation parameter is set to $\lambda = 1$.

The algorithm has been implemented in Matlab and the CPLEX[1] solver has been used to solve the QP problem. The amount of time required to disaggregate a frame of 60 minutes on a personal computer equipped with an Intel i7 CPU running at 3.3 GHz and 32 GB of RAM is about 30 s. The performance is compared to the univariate formulation of AFAMAP and to Hart's algorithm presented in Subsection 4.3.2. The tolerance parameter is set $\alpha = 10^{-6}$.

Table 5.3 presents the number of states, defined a-priori for each class of appliance. , with the power level values resulting from the clustering procedure, and the standard deviation related to each cluster. For appliances with similar consumption value in active power, different values of reactive power are associated: this phenomenon allows to reduce the number of state combination in the aggregate power, when passing from the univariate to the bivariate approach, improving the disaggregation performance.

The number of states in the *noise* model has been varied in the range {4, 6, 8, 10}, both in the univariate and bivariate approaches, in order to find the most performing model.

4.3.3 Results

In this section, the results of the experiments related to the *denoised* scenario will be shown. Since the aggregated power signal depends on which and how many appliances are considered, the experiments have been conducted by varying the number of appliances, in order to evaluate the disaggregation performance for different problem complexities. In particular, different test sets,

[1] `http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`

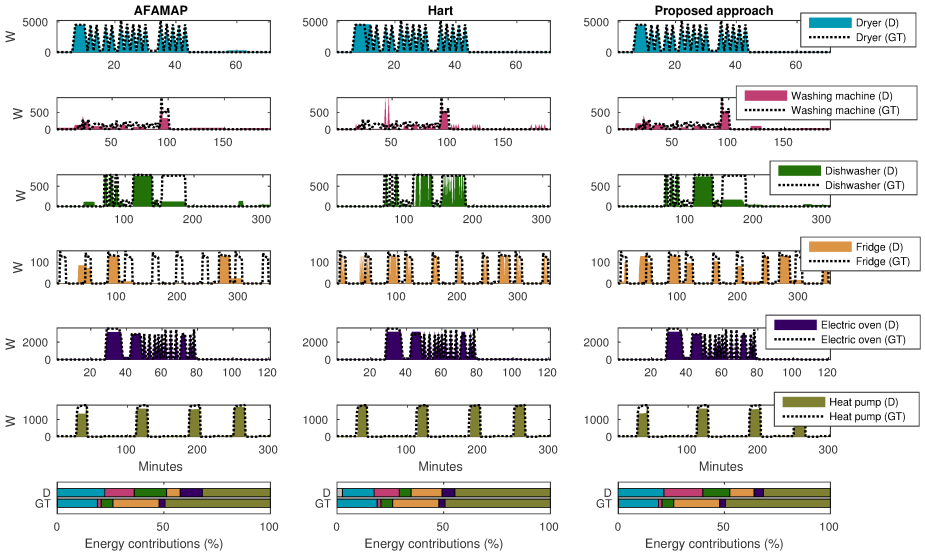
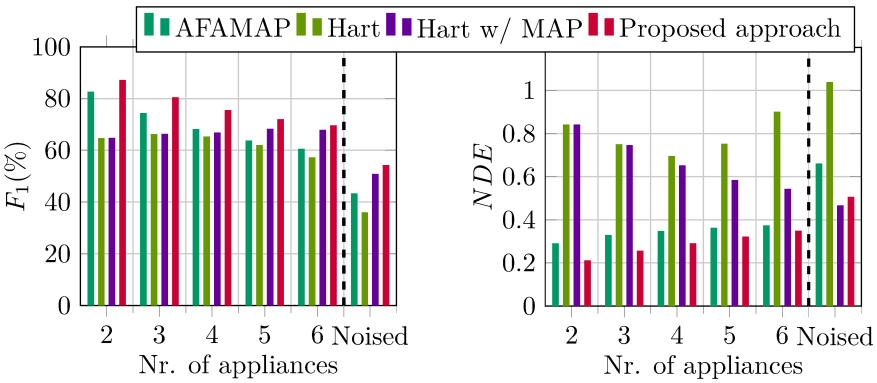


Figure 4.18: Algorithms comparison: AFAMAP vs Hart vs proposed approach. For each algorithm, the disaggregation output (D) is compared against the ground truth (GT) signals.

each composed of every combination of N appliances have been created. For each test set, the total number of experiments is $\binom{6}{N}$, with $N = 2, \ldots, 5$ and the final metrics are calculated averaging between the single experiments overall performance. Before calculating the final F_1-Measure, the Precision and Recall are averaged between the experiments. Differently, the final NDE is the average between the single experiment value.

In Figure 4.18, the disaggregated appliances active power (D) are compared to the corresponding ground truth (GT): in the figure, for each appliance, an adequate time span is considered, in order to evaluate the performance on a single or multiple activations. The bottom of the figure shows the comparison of the appliance contribution to the total energy in the aggregated signal, between the disaggregation outputs and the ground truth consumptions. The left side of the figure shows the disaggregation profiles resulting from the univariate formulation of the AFAMAP algorithm, the central shows the active power component resulting from the Hart's algorithm, and the right side shows profiles related to the proposed approach.

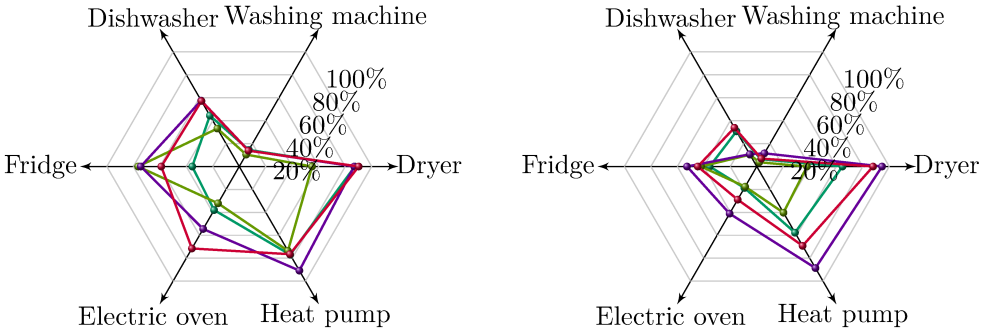
The overall disaggregation results are reported in Figure 4.19, where the F_1-Measure is reported in the Figure 4.19a and the NDE in the Figure 4.19b. The values are related to Table 4.5, where the absolute improvements of the



(a) Comparison of the disaggregation performance in terms of F_1-Measure for different number of appliances.

(b) Comparison of the disaggregation performance in terms of NDE for different number of appliances.

Figure 4.19: Disaggregation performance on AMPds dataset for all the addressed algorithms.



(a) Denoised scenario.

(b) Noised scenario.

Figure 4.20: Performance in terms of F_1-Measure (%) for the different appliances in the "6 appliances" case study: (a) denoised scenario, (b) noised scenario.

Table 4.4: Performance improvement in the "6 appliances" case study (denoised scenario).

Algorithm		Metric	Dryer	Washing machine	Dishwasher	Fridge	Electric oven	Heat pump
AFAMAP	(I)	F_1 (%)	87.3	14.5	44.4	35.5	38.0	76.9
Hart	(II)		54.9	10.4	33.1	76.4	32.1	73.4
Hart w/ MAP	(III)		86.8	14.0	57.5	74.4	54.5	90.8
Proposed approach	(IV)		90.2	13.8	57.1	58.9	71.5	76.5
Improvement	(IV)-(I)	$\Delta F_1/F_1$(%)	+3.3	-4.8	+28.6	+65.9	+88.2	-0.5
	(IV)-(II)		+64.3	+32.7	+72.5	-22.9	+122.7	+4.2
	(IV)-(III)		+3.9	-1.4	-0.7	-20.8	+31.2	-15.8
AFAMAP	(I)	NDE	0.215	2.279	0.685	0.878	0.478	0.388
Hart	(II)		0.798	4.383	1.282	0.670	1.725	0.739
Hart w/ MAP	(III)		0.481	3.003	0.768	0.685	1.026	0.411
Proposed approach	(IV)		0.229	2.384	0.446	0.735	0.286	0.377
Improvement	(IV)-(I)	ΔNDE	+0.014	+0.104	-0.239	-0.143	-0.192	-0.011
	(IV)-(II)		-0.569	-1.999	-0.836	+0.065	-1.439	-0.363
	(IV)-(III)		-0.252	-0.620	-0.322	+0.049	-0.740	-0.034

Table 4.5: Comparison of the disaggregation performance for different number of appliances (denoised scenario).

Algorithm		Metric	2 appl.	3 appl.	4 appl.	5 appl.	6 appl.
AFAMAP	(I)	F_1 (%)	82.4	74.2	68.0	63.5	60.4
Hart	(II)		64.5	66.0	65.1	61.8	57.0
Hart w/ MAP	(III)		64.6	66.1	66.6	68.1	67.7
Proposed approach	(IV)		**87.0**	**80.3**	**75.3**	**71.8**	**69.4**
Improvement	(IV) - (I)	$\Delta F_1/F_1$ (%)	+5.6	+8.2	+10.7	+13.1	+14.9
	(IV) - (II)		+34.9	+21.7	+15.7	+16.2	+21.8
	(IV) - (III)		+34.7	+21.5	+13.1	+5.4	+2.5
AFAMAP	(I)	NDE	0.288	0.327	0.346	0.360	0.371
Hart	(II)		0.839	0.748	0.693	0.750	0.899
Hart w/ MAP	(II)		0.840	0.744	0.650	0.582	0.541
Proposed approach	(IV)		**0.209**	**0.254**	**0.289**	**0.319**	**0.347**
Improvement	(IV) - (I)	ΔNDE	-0.079	-0.073	-0.057	-0.041	-0.024
	(IV) - (II)		-0.630	-0.494	-0.404	-0.431	-0.552
	(IV) - (III)		-0.631	-0.490	-0.361	-0.263	-0.194

proposed approach with respect to the AFAMAP and the Hart's algorithm are shown. The proposed approach reaches the best performance in each case study, with F_1-Measure of 87.0 and NDE equal to 0.209 in the 2 appliances case, and with F_1-Measure of 69.4 and NDE equal to 0.347 in the 6 appliances case, The proposed approach reaches the best performances in each case study, with F_1-Measure of 87.0 and NDE equal to 0.209 in the 2 appliances case, and with F_1-Measure of 69.4 and NDE equal to 0.347 in the 6 appliances case.

The radar chart in Figure 4.20 shows the F_1-Measure for each appliance the experiment including all the 6 appliances and the area of each coloured line is proportional to the F_1-Measure of the related algorithm averaged across the appliances. The values are related to Table 4.4, where the absolute improvements of the proposed approach with respect to the AFAMAP and the Hart's algorithm are shown.

As shown in the plots, the appliances presenting a high steady power con-

sumption are easily recognised, whereas the appliances with complex working cycles, or with several power levels, are more difficult to detect. For instance, the dryer, the electric oven, and the heat pump are successfully reconstructed, whereas the washing machine, the dishwasher and the fridge are partially erroneously reconstructed. Indeed, in the univariate formulation, whenever several appliances present similar consumption levels, many combinations may satisfy the problem constraints and the algorithm chooses an erroneous solution for disaggregation. Comparing the results with the proposed bivariate approach, the multiple combinations of the solution are reduced due to the component constraint to be satisfied by the algorithm, which leads to the correct solution and, consequently, to a better profile disaggregation of the active power component. For instance, although the appliances with higher power level maintain a successful disaggregation, the fridge and the dishwasher improve the correspondence with the ground truth signals. The washing machine partially improve the disaggregation performance in the activation period, whereas introduces some false energy assignation. The disaggregated profiles of Hart's method show that, for some appliances, the FSM is a modelling technique which allows a better representation for the appliances with sharply defined steady states, e.g., the fridge and the heat pump, but a worse representation for appliances with highly variable activity, e.g., the electric oven.

The more confident are the disaggregated profiles with respect to the ground truth signal, the better is the estimation of the energy consumption percentage distribution among the appliances: indeed, for the proposed approach, the consumption distribution have a better correspondence with the ground truth ones, with respect to the AFAMAP algorithm. For instance, the disaggregated profiles related to the fridge results to be more confident, which reflects on the increase of the energy assignation, whereas the dishwasher and the electric oven ones results to have a false energy assignation during the OFF period, corresponding to a decrease of the related energy contributions. Regarding the washing machine, some errors are introduced, therefore the energy assignation is erroneously increased. Regarding the dryer and the heat pump the energy contributions are maintained, because of the correspondence between the algorithms disaggregation performance. In the Hart's method, the improvements in the heat pump and the fridge are reflected on a better correspondence between the energy contributions, but the absence of the constraint between the aggregate power amount and the sum of the disaggregated profiles leads to an unassigned percentage of the total energy (represented as the *grey* portion).

Regarding the performance of the individual appliances, the major improvements with respect to AFAMAP are observed in the electric oven, the fridge and the dishwasher, with an relative increase of the F_1-Measure of $+88.2\,\%$, $+65.9\,\%$ and $+28.6\,\%$, and a variation in the NDE of -0.192, -0.143, -0.239

respectively. This is due to a more accurate correspondence between the disaggregated output and the ground truth, as already shown in the disaggregation output plots. On the contrary, the performance is almost unchanged for the washing machine, the dryer and the heat pump. With respect to the Hart's algorithm, the proposed approach shows an high improvement additionally for the dryer, with an absolute increase of F_1-Measure equal $+64.3\,\%$ and a variation in the NDE of -0.569, whereas, it shows a substantial loss for the fridge, with a decrease of F_1-Measure equal $-22.9\,\%$ and a variation in the NDE of $+0.065$. This demonstrates that the HMM modelling results more effective with a higher number of states. Since moving from the univariate to the bivariate model leads to a greater number of states, this also demonstrates the effectiveness of the proposed approach. Compared to the Hart's algorithm with the MAP stage, the performance on each appliance reduce their gain, particularly for the dishwasher and the dryer, with a decrease of F_1-Measure equal to $-0.7\,\%$ and an increase of $+3.9\,\%$ and a variation in the NDE of -0.322 and -0.252 up to the heat pump, where a loss of performance is shown, with an absolute increase in the F_1-Measure of $-15.8\,\%$ and a variation in the NDE of -0.034. The washing machine remains the appliance with the worst disaggregation performance: the reason is the model complexity, since it is the appliance with the highest number of states, both in the univariate and bivariate representation. Observing the radar chart, the area under the curve related to the proposed approach is increased with respect to AFAMAP and Hart's algorithm, resulting in an average performance improvement, whereas it is slightly higher with respect to the Hart's algorithm version with the MAP stage. The average performance of the system increases, resulting in a relative improvement of F_1-Measure equal to $+14.9\,\%$, $+21.8\,\%$ and $+2.5\,\%$, and a variation in the NDE of -0.024, -0.552, -0.194 with respect to AFAMAP, the Hart's algorithm and the version with MAP stage, respectively.

Concerning the experiments with for different number of appliances, the results shows that, lowering the number of appliances, the performance improve in the FHMM-based algorithms, while in the Hart's algorithm it reaches a peak with 4 appliance, after that the performance decrease. Regarding the Hart's algorithm version with the MAP stage, the performance decrease gradually with a lower number of appliance.

Compared to AFAMAP and to Hart's algorithm, the proposed approach provides a significant performance improvement also when the problem complexity is minimal, i.e., when the number of appliances is 2. The higher absolute increase from AFAMAP occurs with 6 appliances, whereas it decreases lowering the complexity of the problem: this demonstrates that the proposed approach resolves more ambiguities in the NILM solution when the number of combinations of working states is higher.

Table 4.6: Appliances performance improvement in the "6 appliances" case study (noised scenario).

Algorithm		**Metric**	**Dryer**	**Washing machine**	**Dishwasher**	**Fridge**	**Electric oven**	**Heat pump**	**Overall**
AFAMAP	(I)	F_1 (%)	64.6	6.3	30.7	35.6	18.7	57.7	43.1
Hart	(II)		37.9	3.7	10.3	42.1	17.6	40.3	35.8
Hart w/ MAP	(III)		94.6	11.6	10.7	52.7	41.1	88.6	50.7
Proposed approach	(IV)		87.8	6.9	33.8	44.6	28.9	69.2	**54.1**
Improvement	(IV)-(I)	$\Delta F_1/F_1$ (%)	+35.9	+9.5	+10.1	+25.3	+54.5	+19.9	+25.5
	(IV)-(II)		+131.7	+86.5	+228.2	+5.9	+64.2	+71.7	+51.1
	(IV)-(III)		-7.2	-40.5	+215.9	-15.4	-29.7	-21.9	+6.7
AFAMAP	(I)	NDE	0.305	4.395	0.888	0.909	0.939	0.787	0.659
Hart	(II)		0.882	5.960	1.714	0.982	1.593	0.929	1.037
Hart w/ MAP	(III)		0.254	1.965	1.110	0.942	0.974	0.432	**0.464**
Proposed approach	(IV)		0.272	4.055	0.829	0.861	0.930	0.467	0.504
Improvement	(IV)-(I)	ΔNDE	-0.033	-0.340	-0.058	-0.048	-0.010	-0.319	-0.155
	(IV)-(II)		-0.610	-1.905	-0.884	-0.121	-0.663	-0.462	-0.533
	(IV)-(III)		+0.019	+2.090	-0.280	-0.081	-0.044	+0.036	+0.040

Regardless the number of appliances, the performance of Hart's algorithm is lower compared to the proposed approach, because of the less descriptive capabilities of the FSM appliance model with respect to the HMM one. The comparative evaluation with the Hart's version with the MAP stage proves that, even if this approach exploits the information on the most probable solution in case of ambiguity, which is an ideal condition, the proposed approach reaches better performance. Furthermore, the proposed algorithm provides an optimum solution on a frame of T samples, which takes into account both the short-term and long-term dependencies of the signal. This differs in Hart's algorithm that finds the solution by processing the aggregate signal sample-by-sample. For this method, the performance decreases reducing the number of the appliances: a motivation behind this phenomenon can reside in the fact that the MAP stage of the Hart's algorithm chooses a solution with higher probability, but which results incorrect for the majority of the experiments, specially with few combinations.

In this section, the results of the experiments related to the *noised* scenario will be shown. Differently from the *denoised* scenario, the aggregated power signal does not vary with the appliances considered, therefore only the results with all the appliances will be shown. Regarding the number of states of the *noise model*, the experiments demonstrated that, for each approach, the best value is 4, except for the Hart's algorithm with the MAP stage, for which the best results are reached with 10 states. For the sake of conciseness, only the results for the best configuration will be reported in this section.

The overall disaggregation results are reported in Figure 4.19, on the last column, in order to make a comparative evaluation with the *denoised* scenario. The values are related to Table 4.6 on the *Overall* column, where the absolute improvements of the proposed approach with respect to the AFAMAP and the Hart's algorithm are shown. The proposed approach reaches the best overall

performances, with F_1-Measure of 54.1 and NDE equal to 0.504, despite of the Hart's algorithm version with the MAP stage shows an higher NDE value. This discordance will be motivated in the analysis. The radar chart in Figure 4.20b shows the F_1-Measure for each appliance. The values are related to Table 4.6.

Differently from the *denoised* scenario, the major improvement, with respect to AFAMAP, is observed for the dryer, with a F_1-Measure relative improvement of $+35.9\,\%$, and a variation in the NDE of -0.033, whereas the improvements are reduced for the remaining appliances. This proves the effectiveness of the transition from the univariate to the bivariate formulation of the problem, even in the presence of noise.

With respect to Hart's algorithm, the proposed approach shows a higher improvement for the dryer, the dishwasher, and the heat pump with an improvement of $+131.7\,\%$, $+228.2\,\%$, $+71.7\,\%$, and a variation in the NDE of -0.610, -0.884, -0.462. Differently, Hart's algorithm with the MAP stage achieves a higher F_1-Measure, and the relative difference of F_1-Measure for the heat pump, the electric oven and the dryer is $-19.4\,\%$, $-12.2\,\%$, $-6.8\,\%$, while in terms of NDE the difference is $+0.036$, -0.044, $+0.019$. This demonstrates that the HMM modelling leads to performance improvements with respect to the FSM modelling even in the presence of noise, but considering the MAP stage this improvements is substantially reduced. The washing machine is still the appliance with the worst disaggregation performance, following the trend of the *denoised* scenario. Observing the radar chart, the area under the curve related to the proposed approach is increased with respect to AFAMAP and Hart's algorithm, resulting in an average performance improvement, whereas it is comparable with respect to the Hart's algorithm version with the MAP stage, due to unbalancing between the appliances.

The average performance of the system increases, resulting in a F_1-Measure absolute improvement of $+25.5\,\%$, $+51.1\,\%$ and $+6.7\,\%$, and a variation in the NDE of -0.155, -0.533, $+0.040$ with respect to AFAMAP, the Hart's algorithm and the version with MAP stage, respectively.

Comparing those results to the *denoised* scenario ones, the overall performance is lower, due to the introduction of the noise contribution in the aggregated power, except for the Hart's algorithm with the MAP stage: despite the F_1-Measure shows a degradation of performance, the NDE decreases, meaning that this version of the algorithm maintains the trend showed with the increase of the number of appliances. In fact, the *noised* scenario can be defined as the *denoised* scenario using the *noise* model additionally to the appliances models, therefore the MAP stage introduces additional advantages, leading to a performance improvement. The MAP stage exploits additional information which are not introduced within the AFHMM, but represents an almost ideal FSM based case study.

4.4 Footprint extraction procedure

Among different NILM approaches, the supervised ones reach better performance [56, 53], that is the resulting disaggregated signals have a better correspondence with the true appliance energy consumption. Therefore, those methods results to be more reliable for the final user.

The supervised section in the NILM algorithms corresponds to the appliance modelling stage, as showed in Figure 4.21b, where the training phase is carried out. A model is created starting from the appliance level consumption (e.g., training set), in order to represent each appliance in a parametric way, and its parameters are used in the NILM algorithm in order to disaggregate the portion of the aggregated power consumption related to each appliance, as represented in Figure 4.21c.

The power consumption profile of an appliance can be depicted as the repeating of a working cycle, alternated by time intervals when the appliance is turned off. The repetition rate, related to the length of the *off-intervals*, depends on the user consumption habit.

Therefore, in order to analyze the consumption features of an appliance, it is sufficient to extract the working cycle in the appliance level consumption, defined as the *footprint*, and to exploit it as training set in the appliance modelling stage.

This stage of the supervised NILM chain is named *footprint extraction*, as showed in Figure 4.21a.

In literature, different approaches have been proposed to extract the appliance working cycle features from the aggregated data. An unsupervised method, based on spectral clustering, is proposed in [22]: the most different activation occurrences, which can be denoted in the aggregated power, are saved; then, they are grouped between the most similar, using the clustering technique. A bayesian approach is used in [19, 20]: a generic bayesian model for the appliance category is defined; then, it is fitted on the activation within the aggregated power, using a threshold schema on the likelihood function. Most of those approaches have limitations, concerning the aggregated power, where the appliance activation can be overlapped and it can cause trouble in the extraction phase.

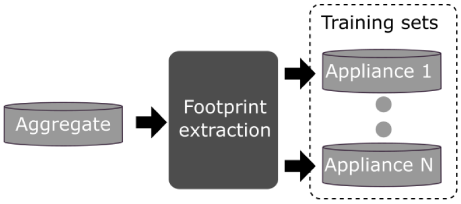
To overcome this, in a real scenario, the user interaction with the system can be considered, in order to improve the reliability of the footprint extraction: in those cases, the user needs a facilitated procedure to determinate the appliance activation instant and an easy way to interact with the energy monitoring system. Therefore, in this work a *user-aided* footprint extraction procedure is proposed.

The easiest way to extract the footprint from the aggregated power is to use

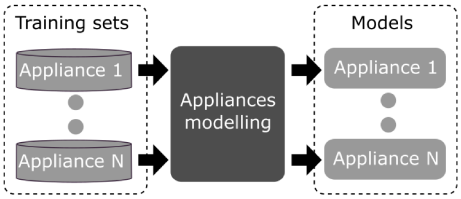
the appliance alone, turning off all the other devices in the electrical network, as described in [16]. This approach results to be the more reliable for the user, thus it is adopted in the presented work.

The appliance modelling stage employs the footprint, in order to represent the appliance consumption behavior: despite several works deal with model for the classification, such as SVM, k-NN [37] or deep neural networks [32], the Hidden Markov Model (HMM) is a widespread modelling technique [18, 23, 29], since it is able to represent the behavior of the appliance in working states and to regulate the transition with a probability value. This representation is close to the real appliance mode of operation, where each working state corresponds to a power consumption value.

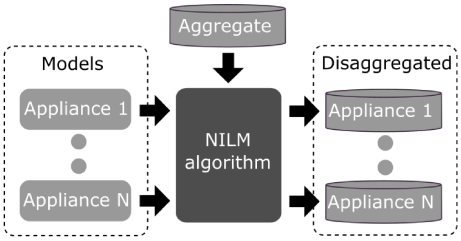
In this work, the disaggregation algorithm is based on HMM, in particular the AFAMAP (Additive Factorial Approximate Maximum a Posteriori) algorithm [22] is used.



(a) The footprint extraction stage.



(b) The appliance modelling stage.

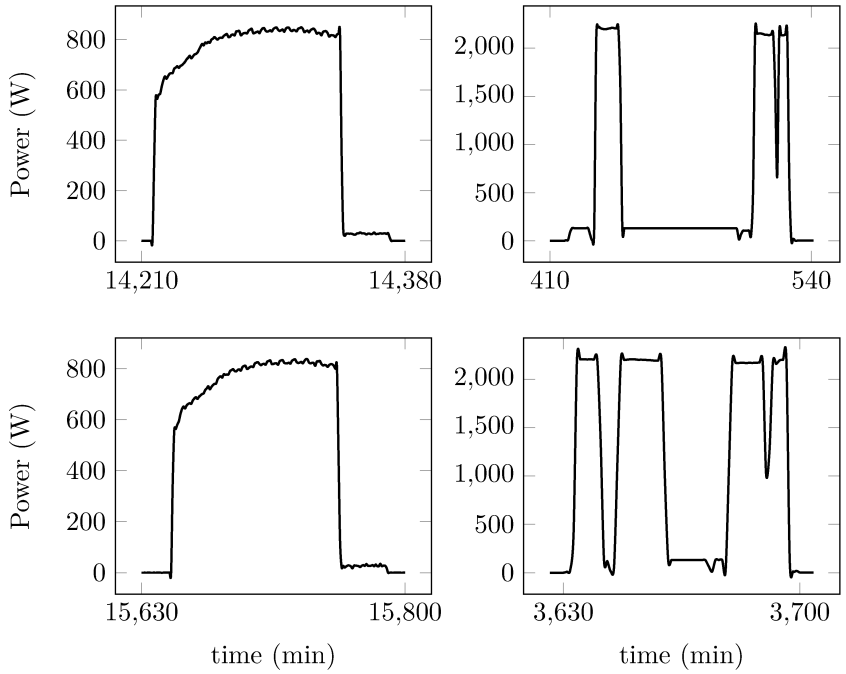


(c) The disaggregation algorithm stage.

Figure 4.21: The Supervised NILM chain.

The unavailability of the appliance level consumption, for extract the footprint, represents one of the main issue in the NILM supervised approach. In

real scenarios, only the aggregated power consumption is available to the user. Therefore, the footprint extraction stage aims to extract the appliance footprint from the aggregated power: this work aims to investigate the performance of a footprint extraction procedure based on the HMM and AFAMAP algorithm.



(a) Dryer, household 1.

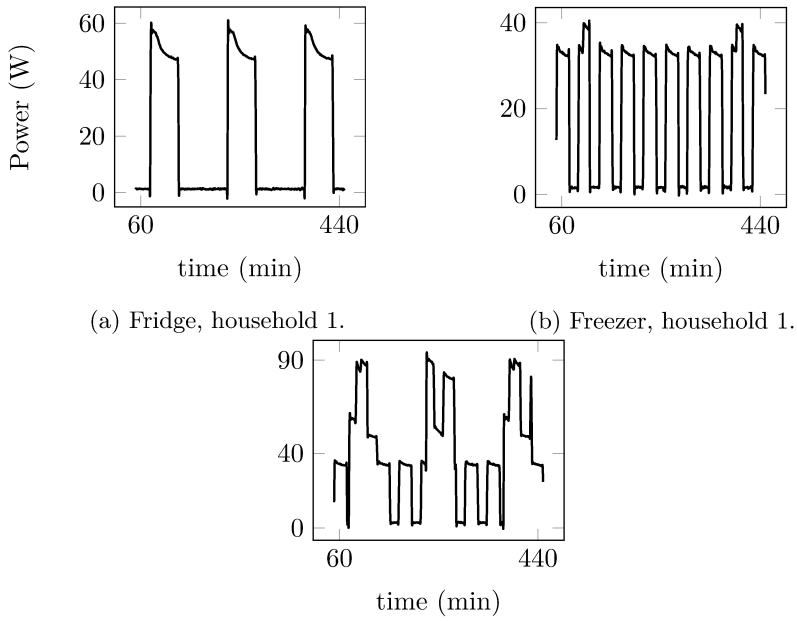
(b) Dishwasher, household 2.

Figure 4.22: Alike and different footprints for the same appliance, in ECO.

A working cycle of an appliance is the interval between the power on and the power off by the user. In this time interval, the appliance power consumption signal is defined as *footprint*. Some examples of footprint taken from the ECO dataset [57] are shown in Figure 4.22, that reports the power consumption traces recorded from the appliances located inside different Swiss households.

The usage of an appliance differs every time, especially in the case of equipments with different usage modes: e.g., the operating cycles of a washing machine can be set in a different way each time, or the operation of the dishwasher may vary according to the selected rinsing cycle. The different usage mode of the same appliance reflects on different footprint, as shown in Figure 4.22b: the power levels in the two footprint of the dishwasher are the same, but they appear in different orders, which demonstrate that the working state composing the appliance working cycle are unique, but they are employed in different orders, based on the user habits. Therefore, it is necessary to record different

occurrence of the appliance footprint, in order to explore the different user habits in the appliance usage.



(a) Fridge, household 1.

(b) Freezer, household 1.

(c) Fridge-freezer combination, household 1.

Figure 4.23: Power consumption of continuously turned on appliances, in ECO.

On other hand, this aspect is not significant for appliances with easier working principle, and a less complex circuit composition. In this case, the usage pattern of the appliance can not be different in times, thus the footprint appears to be similar in each occurrence, as shown in Figure 4.22a: the footprint of the dryer follows the same trend in time, which demonstrates the unique working cycle of the appliance and the unique way of usage by the user.

The footprint extraction is a necessary step in supervised NILM algorithms. In this context, the user exploits the aggregated power sensing system. An easy method to record the appliance footprint is to switch off all the appliances in the household and to turn on only the appliance of interest [16]. In this way, the aggregated power consumption corresponds to the appliance one.

The appliance switch on and off are detected by using a threshold schema on the active power consumption: when the value exceeds a threshold, the current is flowing in the circuit and the appliance is turned on, whereas when the value is below, the appliance is turned off. A threshold equal to the value of 50 W is a good choice for most datasets, nevertheless this value depends on the type of appliance and the activation power consumption. The samples

between those two events are saved as the power consumption data related to the footprint. Multiple usages of the same appliance define different occurrences of the footprint.

In a household not all appliances can be turned off, e.g., the fridge and the freezer have to be continuously powered in order to maintain the food inside in safe condition. As shown in Figure 4.23a and Figure 4.23b, their power consumption are continuous in time, with a periodic working cycle. In this scenario, the aggregated consumption presents a continuous component, resulting from the sum of the fridge and freezer consumption, as shown in Figure 4.23c. This signal can be modeled as the consumption of a unique model, representing the combination fridge-freezer as a composed appliance.

The presence of this component in the aggregated power does not allow to acquire a *clean* footprint of the appliance of interest, since all the appliances power signals are summed up on the aggregated power. Therefore, the footprint results to be *corrupted* and a procedure to clean it is needed.

In order to clean a corrupted footprint, a procedure to separate the fridge-freezer consumption from the appliance footprint one is needed.

The fridge-freezer contribution can be recorded on the aggregated power turning off all the other appliances in the household: in this way, the characterization of the fridge-freezer combination is not afflicted by noise or other appliances consumption, thus the extracted model results to be highly reliable and accurate.

The steps to be followed are the following:

1. the consumption of the fridge-freezer combination is recorded, in a adequate span of time to collect enough data for the modelling;

2. a corrupted version of the appliance of interest footprint is acquired;

3. the extraction procedure is applied to the recorded footprint, using the a priori knowledge of the fridge-freezer model and a generic model of the appliance.

The process of signal separation can be interpreted as a disaggregation problem with 2 sources: therefore, the same NILM algorithm, which is executed after the footprint extraction and the appliance modelling step, can be exploited for the footprint extraction step, as well. In order to obtain the disaggregated traces, the NILM algorithm requires both the model of the fridge-freezer combination and of the appliance of interest. The first one is available, whereas the appliance model is not available, because the footprint extraction step precedes the appliance modelling step. Therefore, it is necessary to provide a generic model, which represents the class related to the appliance of interest, and which is suitably fitted on the specific appliance features, e.g., a priori knowledge of

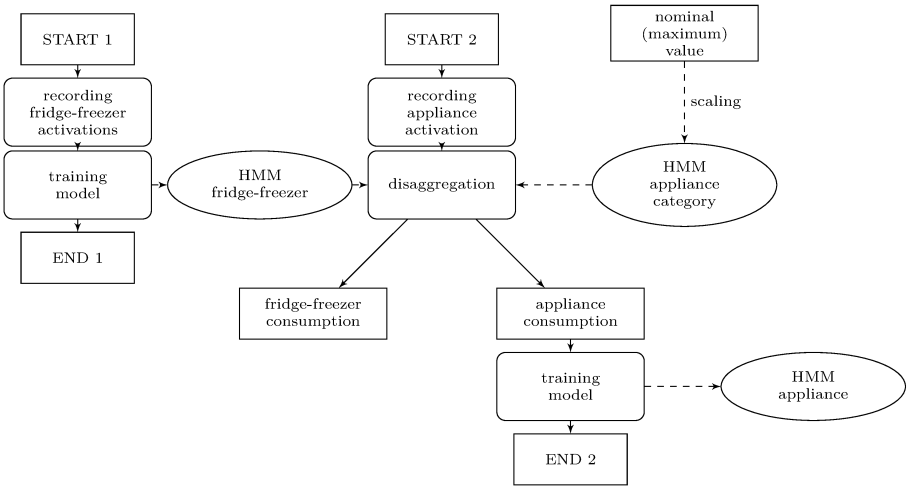


Figure 4.24: Footprint extraction algorithm flowchart.

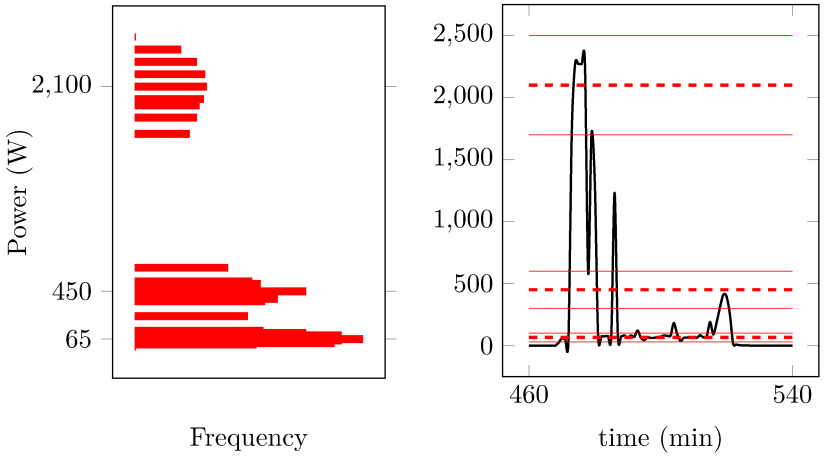
the maximum power consumption, in order to represent it as good as possible. This procedure introduces an uncertainty in the appliance modelling stage, which might be the cause of the error in the footprint extraction stage.

In this work, the NILM algorithm chosen for the disaggregation step is the *AFAMAP* proposed by *Kolter and Jaakkola* [22]: the algorithm requires the HMM of each appliance that contributes to the aggregated power signal.

From the analysis carried out in Section 4.4, the availability of the HMM of both the fridge-freezer combination and the appliance of interest is necessary. The first one is obtained from the corresponding consumption recorded, thus it is a model with high reliability: as showed in Figure 4.23c, it is a model with 4 working states, derived from the composition of the 2 working states of the fridge and the freezer. Whereas, for the appliance of interest, the model is not available, since it is derived after the footprint extraction step. Therefore, a generic HMM is exploited: it is obtained from a reference dataset, under the assumption that all the appliances of the same category acts in the same way, while passing from a working state to another, so that the transition probability matrix results the same for each appliance in the category. Furthermore, it is assumed that the number of the working states is the same for all the appliances of the same category, since the working cycle of the appliance type observed in the footprint: therefore, the number of states is defined a priori for the appliance type, such as described in Table 5.3.

For the appliances with a number of working states greater than 2, it is assumed that the consumption values are proportional each other: therefore, the consumption values in the model are scaled based on the nominal (maximum) value, which is given a priori to the algorithm.

In this way, the HMM represents the appliance as good as possible, omitting the approximation on the consumption values of the middle working state and the approximation on the transition probability matrix.



(a) Histogram of the power consumption values.

(b) Footprint and clusters associated to the working states.

Figure 4.25: Washing machine in ECO, household 1.

After the AFAMAP algorithm execution, two disaggregated consumption profiles are obtained: the appliance one corresponds to the extracted footprint. Starting from this, the HMM representing the appliance is created, which is used in the disaggregation algorithm to solve the NILM problem.

In order to reach a good generalization in the HMM creation, the availability of different appliance footprints is necessary, as described in Section 4.4: this process allows to mitigate the errors introduced in the footprint extraction phase. A suggested value of occurrences to record is in the order of 10.

In Figure 4.24 the flowchart of the footprint extraction algorithm is depicted.

Table 4.7: Number of working states defined for each category of appliance.

Appliance	**num. of states**
Fridge	2
Freezer	2
Dryer	3
Washing machine	4
Dishwasher	3
Oven	3

The diagram is composed of two sections: in the left one, the contribution of the fridge-freezer combination is recorded, from which the HMM is obtained; in the right one, the appliance activations are recorded, to obtain the footprint and the related HMM. This procedure is repeated for each appliance footprint recorded, which needs to be extracted.

4.4.1 Experimental setup

In order to execute the AFAMAP disaggregation algorithm, it is necessary to generate the HMM of each appliance from the extracted footprints.

As first step, the power consumption values associated to each working state need to be extracted. This is achieved via a clustering procedure.

Recording many footprints allow to reach a proper solution during the iterative procedure in the algorithm: indeed, this kinds of algorithm operates more effectively when a significant amount of data is available for each cluster to find.

The cluster centroid represents the power consumption value of the appliance in that working state: the inference of a gaussian variable on the data related to the same cluster is carried out. The resulting mean value corresponds to the centroid of the cluster and the variance determines the width of the cluster, as shown in Figure 4.25b.

The levels with high variability are susceptible of great variance in the consumption value, e.g., the state with higher consumption, while the levels with lower variability have a tighter interval, e.g., the OFF state, as shown in Figure 4.25. The variability in the levels is an information representative of the appliance category, as well as of the user usage habits.

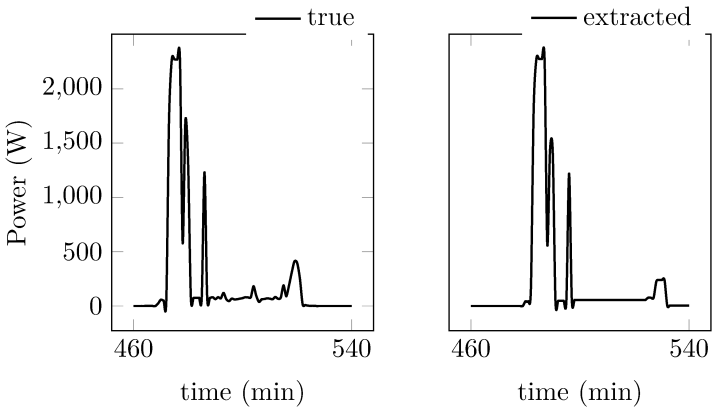
As final step, the HMM is created by using the well known training techniques.

The experiments have been conducted using different datasets: the first one for the generic model extraction, and the second one for testing the footprint extraction algorithm. The disaggregation experiments have been conducted on the same dataset, to evaluate the effectiveness of the footprint extraction algorithm, compared to the use of the true appliance level consumption, to create the appliance model.

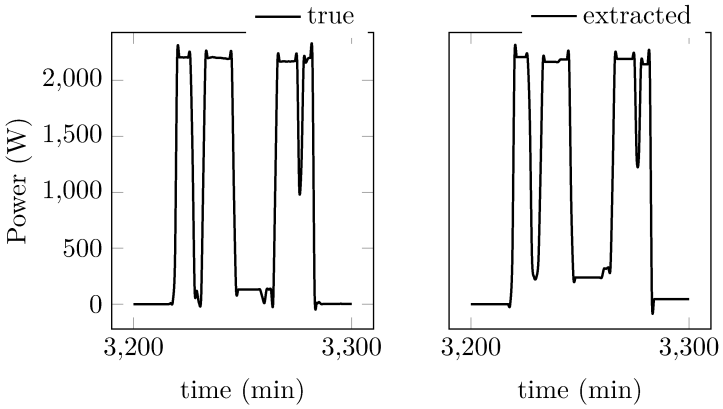
The general model has been extracted using the *AMPds* dataset [58]. The experiments on footprint extraction and disaggregation are conducted on the *ECO* dataset [57], considering the households 1 and 2, whose appliances are:

- household 1: dryer, washing machine;
- household 2: dishwasher, oven.

The experiments include the the fridge-freezer combination, present in each household.



(a) Washing machine in ECO, household 1.



(b) Dishwasher in ECO, household 2.

Figure 4.26: Comparison between the true and the extracted footprint for some appliances.

4.4.2 Results

Figure 4.26 shows two example of extracted footprints, compared to the original ones. In both cases, a good correspondence between the temporal trends can be noticed, which denotes that the model representing the fridge-freezer combination has a high reliability and it allows to extract the appliance footprint contribution in a suitable way. However, for several portions of the footprint, the correspondence with the power level is not correct: this might be due to the incorrect power levels of the general model, which are obtained from a

Table 4.8: Disaggregation performance in ECO, household 1.

	Metric	Fridge-freezer	Dryer	Washing machine	AAA	Footprint
State based	P	0.506	0.657	0.909	0.691	True
	R	0.568	0.821	0.948	0.779	
	$\boldsymbol{F_1}$	**0.536**	**0.730**	**0.928**	**0.732**	
	P	0.483	0.622	0.880	0.661	Extracted
	R	0.531	0.788	0.937	0.752	
	$\boldsymbol{F_1}$	**0.506**	**0.695**	**0.908**	**0.704**	
Energy based	P	0.955	0.488	0.849	0.764	True
	R	0.815	0.972	0.978	0.922	
	$\boldsymbol{F_1}$	**0.879**	**0.650**	**0.909**	**0.835**	
	P	0.953	0.422	0.809	0.728	Extracted
	R	0.790	0.976	0.982	0.916	
	$\boldsymbol{F_1}$	**0.864**	**0.589**	**0.887**	**0.811**	

scaling operation respect to the nominal consumption value. Indeed, the error is introduced in the middle power levels, while for the maximum power level the correspondence is exact. In the entire process, the uncertainty introduced from the disaggregation algorithm, used to separate the footprint from the consumption of the fridge-freezer combination, needs to be considered.

The experiments have been conducted on a portion of 30 days of the ECO dataset. To evaluate the effectiveness of the footprint extraction procedure, the disaggregation results have been evaluated using:

- the models created by using the appliance level consumption, available in the dataset (*true* footprint);
- the models created by using the *extracted* footprint, following the procedure described in Section 4.4.

The disaggregation results have been evaluated using the Precision (P) and Recall (R) metrics, defined in Section 2.4 in state and energy based sense. To compare the performance of the entire disaggregation system, the F-score (F_1) metric averaged across the appliances (AAA) has been used.

The parameters used in the AFAMAP algorithm were the same employed in Section 4.2. The disaggregation window parameter has been set $T = 60$ min.

The disaggregation results are showed in Table 4.8 and Table 4.9. For both metrics, the algorithms achieve good performance: the best results are reached in the household 2 experiment, with a F_1 of 0.898 in state based sense, and 0.956 in energy based sense. This is due to the relatively simple problem studied in those cases: a disaggregation problem with only 3 appliances, with highly distinguishable values of power consumption, reveals to be solvable with high accuracy. The experiments in Table 4.9 shows a better performance respect

Table 4.9: Disaggregation performance in ECO, household 2.

Metric		Fridge-freezer	Dishwasher	Oven	AAA	Footprint
State based	P	0.741	0.926	0.977	0.881	True
	R	0.781	0.980	0.984	0.915	
	$\boldsymbol{F_1}$	**0.760**	**0.952**	**0.980**	**0.898**	
	P	0.735	0.855	0.972	0.854	Extracted
	R	0.773	0.974	0.982	0.910	
	$\boldsymbol{F_1}$	**0.754**	**0.911**	**0.977**	**0.881**	
Energy based	P	0.983	0.873	0.973	0.943	True
	R	0.944	0.983	0.984	0.970	
	$\boldsymbol{F_1}$	**0.963**	**0.925**	**0.979**	**0.956**	
	P	0.981	0.816	0.975	0.924	Extracted
	R	0.939	0.982	0.988	0.970	
	$\boldsymbol{F_1}$	**0.960**	**0.891**	**0.982**	**0.946**	

to the Table 4.8 one: the reason is the appliances footprints and the resulting HMMs composition. Indeed, the second problem is composed of models with a lower number of states (e.g., 3 states for the dishwasher, 3 states for the oven, respect to the 3 states for the dryer and 4 states for the washing machine), thus the disaggregation problem results to be simpler in the resolution, and the overall performance reaches higher values. This trend was already introduced from the author of the disaggregation algorithm [22], who shows that the higher is the number of states related to the HMM, the higher is the complexity of the problem definition, and lower is the disaggregation performance due to the more difficult resolution. Regarding the first problem, the fridge-freezer combination has the consumption values close to the dryer ones, which leads to an ambiguity during the problem resolution and a lower performance for the total problem. In general, the appliance with the better performance is the one with the higher power consumption value: for the first problem the washing machine, for the second one the oven.

In both experiments the results corresponding to the true footprint show higher performance respect to the extracted footprints ones: it means that the footprint extraction procedure introduces an error in the appliance modelling stage, which results in a error during the disaggregation algorithm resolution. Nevertheless, the results of the extracted footprint experiments show performance with an admissible relative loss: for the household 1 experiment, the relative loss results of 3.83% in state based sense, and 2.87% in energy based sense, while for the household 2 experiment, it results of 1.89% in state based sense, and 1.05% in energy based sense .

In conclusion, the models obtained after the footprint extraction procedure show a good correspondence with the original ones, which means that the

footprint extraction is sufficiently reliable. Therefore, the footprint extraction algorithm introduced in this work provides a convenient procedure to the user for modelling the appliance at the cost of an acceptable loss in disaggregation performance.

Chapter 5

DNN based approach

The recent success of Deep Neural Networks (DNN) in several application scenarios [93, 94] drove the scientific community to employ this paradigm also for NILM. Kelly & Knottenbelt [32] compared three alternative DNNs: in the first, they employed a convolutional layer followed by long short-term memory (LSTM) layers [60] to estimate the disaggregated signal from the aggregate one. In the second, a denoising autoencoder composed of convolutional and fully connected layers is trained to provide a denoised signal from the aggregate one. The third network estimates the start time, the end time and the mean power demand of each appliance. The algorithms where evaluated on the UK-DALE dataset and showed superior performance with respect to the combinatorial optimisation and FHMM algorithms implemented in the Non-intrusive Load Monitoring Toolkit (NILMTK) [73].

5.1 Neural NILM

The work by Kelly and Knottenbelt [32] compared three different neural network architectures: in the first, they employed a convolutional layer followed by LSTM layers [60] to estimate the disaggregated signal from the aggregated one. In the second, a denoising autoencoder (dAE) composed of convolutional and fully connected layers is trained to provide a denoised signal from the aggregated one. The third network estimates the start time, the end time, and the mean power demand of each appliance. The algorithms were evaluated on the UK-DALE dataset and the results showed that the dAE approach outperforms the alternative neural networks architectures as well as the FHMM algorithm implemented in the Non-intrusive Load Monitoring Toolkit (NILMTK) [73].

5.2 Denoising AutoEncoder approach

The NILM task can formulated as a denoising problem by expressing the aggregated signal as the sum of the power consumption of the appliance of interest

and a noise component that incorporates all the remaining contributions. In particular, equation (2.1) can be reformulated as:

$$y(t) = y_j(t) + v_j(t), \tag{5.1}$$

for $j = 1, 2, \ldots, N$, where

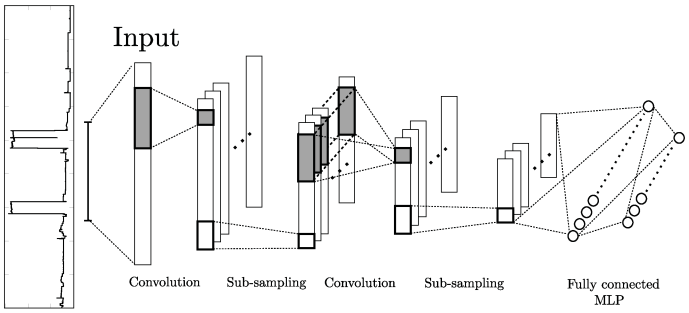
$$v_j(t) = \sum_{\substack{i=1 \\ i \neq j}}^{N} y_i(t) + e(t), \tag{5.2}$$

represents an overall noise term for the appliance j that comprises both the measurement noise and the contributions of the other appliances. Thus, for obtaining $y_j(t)$, it would be sufficient to remove the noise term $v_j(t)$ from the aggregate measurement $y(t)$.

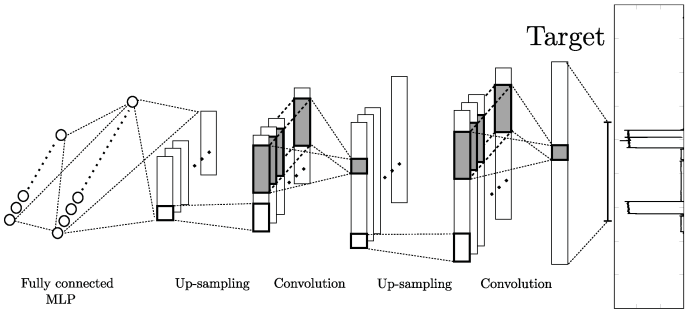
In [32] and similarly in [31], noise removal is performed by means of a dAE, i.e., a neural network that is trained to reconstruct a clean signal from its noisy version presented at the input. Denoising autoencoders have been originally formulated in the context of *representation learning* and as an unsupervised training method [95]. The same structure has been later employed to perform actual noise removal, such as in speech related tasks [96, 97]. An autoencoder can be seen as an encoder network followed by a decoder network. The encoder provides an internal representation of the input signal and the decoder transforms it back into the input signal domain. A common choice consists in creating a network with specular encoder and decoder topologies. In the context of NILM, for each appliance, an autoencoder is trained to reconstruct the ground truth $y_j(t)$ given the aggregated signal $y(t)$.

5.3 Algorithm improvements

In this section, several algorithmic and architecture improvements to the dAE approach for NILM are proposed and an exhaustive comparative evaluation with the AFAMAP (Additive Factorial Approximate Maximum a Posteriori) algorithm [22] is conducted. In particular, compared to [32] the dAE approach for load disaggregation is improved by conducting a detailed study on the topology of the network, and by introducing pooling and upsampling hidden layers, and the rectifier linear unit (ReLU) activation function [98] in the output layer. Additionally, the network output is recombined by using a median filter on the overlapped portions of the disaggregated signal. The second contribution is an exhaustive performance comparison between AFAMAP and the dAE approach. Indeed, FHMMs have been largely employed in the last years since they are an effective approach for load disaggregation, and AFAMAP, in particular, re-



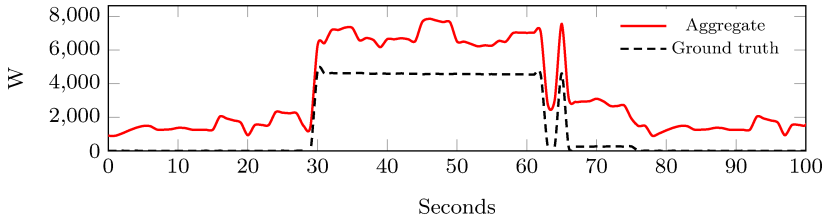
(a) Encoder network. The input signal is the aggregated power consumption.



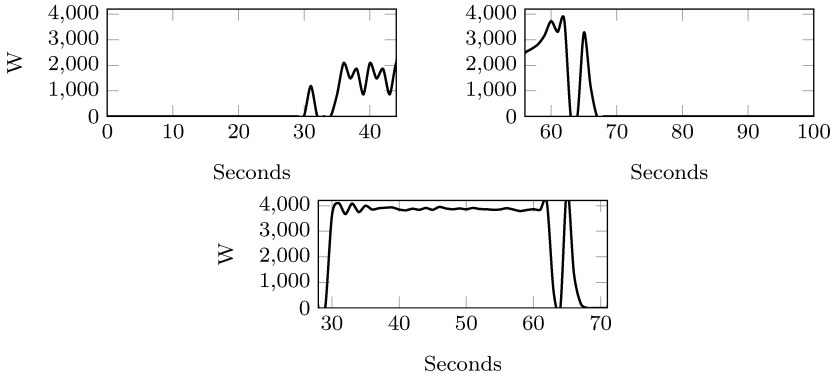
(b) Decoder network. The target signal is ground truth power consumption of each appliance.

Figure 5.1: Generic autoencoder architecture employed for disaggregation.

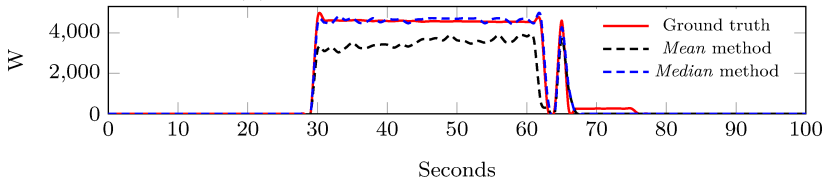
ceived noteworthy attention by the scientific community [99, 100], as described in Section 4.1. However, an exhaustive performance comparison between the two methods has not been yet conducted. Indeed, the authors of [32] compare their proposed approaches to the FHMM method implemented in NILMTK [73], but their comparison does not consider more advanced FHMM algorithms such as AFAMAP [22]. Additionally, their experiments consider only a noised scenario on a single dataset (UK-DALE). Here, the evaluation is performed on three datasets, UK-DALE [61], AMPds [58], and REDD [30] in different conditions: firstly, the algorithms are evaluated on denoised and noised scenarios. In the denoised scenario, the aggregated signal is the sum of the power profiles of the appliances that are disaggregated. In the noised scenario, the aggregated signal comprises also measurement noise and the contributions of unknown appliances. Successively, the algorithms generalisation capabilities are evaluated by performing disaggregation on the data acquired in a house not considered in the training phase (unseen scenario). The performance is evaluated by using both energy-based metrics and state-based metrics [73]: the first, evaluate the capability of the algorithm to estimate the actual power profile of the appliances, while the second the capability of estimating whether the appliance is



(a) A portion of aggregated data, analysed with sliding window technique.



(b) Output of the dAE for each window.



(c) Disaggregated traces comparison between *median* and *mean* recombining methods.

Figure 5.2: Network outputs recombined by using the mean operation and the median operation recombination on the overlapped portions.

in the "on" or "off" state. In order to perform the experiments in presence of noise, a Rest of the World (RoW) model has been introduced in the original AFAMAP [22] algorithm. This model represents all the appliances but the ones of interest and makes AFAMAP able to operate in a noised scenario. The obtained results show that on average the dAE approach outperforms AFAMAP in all the addressed experimental conditions.

The general network topology proposed here for NILM is shown in Figure 5.1: the encoder network (Figure 5.1a) is composed of one or more one-dimensional convolutional layers that process the input signal and produce a set of feature maps. Each convolutional layer is followed by a linear activation function, by a max pooling layer, and by additional convolutional and pooling layers. Fi-

nally, one or more fully connected layers followed by a ReLU [98] activation function close the encoder network. The max pooling operation returns the maximum value within a neighbourhood, and in image processing, it makes the obtained representation invariant to small translations of the input. In NILM, this translates into being more independent on the location of an activation inside an analysis window. Additionally, max pooling reduces the size of the feature maps and the number of units in the fully connected layers, thus reducing the number of training parameters. The ReLU activation function calculates the maximum between its input and zero, and in this case it prevents the occurrence of negative values of the disaggregated active power. The decoder (Figure 5.1b) is structured specularly to the encoder, with upsampling layers taking the place of max pooling layers. Compared to [32], several network topologies are explored, with multiple convolutional stages, max pooling and upsampling layers are introduced, and the ReLU activation function in the fully connected layers.

Table 5.1: Energy ratio (ER) for each house in the considered datasets.

Dataset	AMPds	UK-DALE				REDD		
		House 1	House 2	House 4	House 5	House 1	House 2	House 3
ER	0.731	0.680	0.564	0.867	0.833	0.634	0.463	0.613

Table 5.2: Definition of the training, validation and test sets for the considered datasets.

Dataset		**Train+Validation**	**Test**
AMPds		1 year, 6 months	6 months
UK-DALE	House 1	1 year, 8 months, 3 days	7 days
	House 2	4 months, 3 days	7 days
	House 4	6 months, 25 days	7 days
	House 5	2 months, 3 days	6 days
REDD	House 1	33 days	3 days
	House 2	12 days	2 days
	House 3	12 days	6 days

The dAE network is trained to minimise the mean squared error between its output and the activation of a single appliance. Training is performed by using the Stochastic Gradient Descent (SGD) algorithm with Nesterov momentum [101], and with the early-stopping criterion to prevent overfitting. The input data and the target are normalized in order to improve the learn efficiency.With respect to the reference work [32], several advancements have been introduced in the training phase. In particular, during the training phase, the initial value of the learning rate is decreased when the performance on a validation

set decreases. When this occurs, training is resumed from the epoch where the performance started decreasing. If the validation performance remains confined in a certain interval, typically when the learning process has reached the convergence or the learning rate has become too little, the early-stopping criterion is used. This is adopted in order to prevent overfitting.

In the disaggregation phase, the input signal $y(t)$ is analysed by using sliding windows whose lengths depend on the size of the appliance activations. Windows are partially overlapped and the output signal is recombined by using a median filter on the overlapped portions. This differs from what proposed in [32], where the authors recompose the overlapped portions by calculating their mean value. The problem with this solution is that when an activation is only partially comprised in the analysis window, the network tends to underestimate the value of the output signal. As the window slides, the estimate increases, but averaging the overlapped portions produces an overall underestimated signal. Differently, by using the median operation on the overlapped portions, this phenomenon is mitigated, since greater values are preserved. The overall operation is depicted in Figure 5.2.

Table 5.3: Number of states m related to each class of appliance.

Nr. of states	**Dryer**	**Washing machine**	**Dishwasher**	**Fridge**	**Electric oven**	**Heat pump**	**Kettle**	**Microwave**
m	3	4	3	2	3	3	2	2

The input signal is normalised following the same technique used in the training phase, while the disaggregated traces are denormalised after recombining outputs.

5.3.1 Experimental setup

Table 5.4: Window width (in samples) for the dAE architecture. The number of samples depends on the dataset sampling rate.

Dataset	**Dryer**	**Washing machine**	**Dishwasher**	**Fridge**	**Electric oven**	**Heat pump**	**Kettle**	**Microwave**
UK-DALE	-	1024	1536	512	-	-	128	288
AMPds	75	120	210	45	120	90	-	-
REDD	1536	-	2304	496	-	-	-	96

In order to conduct an exhaustive evaluation on different scenarios, three public datasets have been chosen. The Almanac of Minutely Power dataset (AMPds) [58] contains recordings of consumption profiles belonging to a single home in Canada for a period of two years, at 1 minute sampling period. The experiments are conducted by using six appliances: dryer, washing machine,

dishwasher, fridge, electric oven, and heat pump. The second dataset, UK-DALE [61], is composed of consumption profiles recorded in five houses in UK over two years, at 6 seconds sampling period. The houses consumptions are not equally distributed over this time period, e.g., house 3 contains only the kettle consumptions and some minor appliances recordings, thus it is not considered in the experiments. The five target appliances considered in all the experiments are: fridge, washing machine, dish washer, kettle and microwave. The third dataset, REDD [30], contains aggregate and circuit-level power profiles of several US households. The sampling period of the aggregate data is 1 s, while the one of the target profiles is 3 s, thus aggregate data was downsampled in order to match the sample period of the target profiles. The experiments are conducted by using four appliances: dryer, dishwasher, fridge, and microwave. In the seen scenario, the data from two houses is used both for training and testing. In the unseen scenario, the same data is used for training, while testing is performed on the data of a third house.

The chosen appliances represent the principal contributions to the peak of power consumption in the aggregated signal, which allows us to consider the *denoised* scenario as an approximation of the *noised* scenario in the traits of higher power consumption. On the other hand, the *noise* contribution, assigned to the RoW model, depends on the number of remaining appliances not modelled and on the total energy of the main aggregated signal, and this affects the disaggregation performance in the *noised* scenario. The *energy ratio* (ER), defined as:

$$\mathrm{ER} = \frac{E_{\mathrm{RoW}}}{E_{\mathrm{main}}} = \frac{\sum_{t=1}^{T} e(t)}{\sum_{t=1}^{T} y(t)}, \tag{5.3}$$

expresses the energy proportion between the RoW model and the total aggregated data, and the values for each house in the considered datasets is showed in Table 5.1.

The datasets are split in different portions for training and testing, and their dimensions depend on the availability of appliances activations within the dataset. Regarding the training procedure, within the period specified in Table 5.2, the first 20 % of activations are used to compose the validation set, while the remaining 80 % are used for the models training.

Regarding the ground truth consumption availability, two different scenarios can be defined. In the *seen* scenario, the disaggregation is computed on the same houses used to train the models, but in different period from the training data. In this scenario, both models, HMM and neural network, are created exploiting the same portion of training, in order to conduct a fair comparison between the methods. On the other hand, in the *unseen* scenario, the disaggregation is computed on the data related to a house not considered in the training phase. In this scenario, the ground truth consumptions related to each

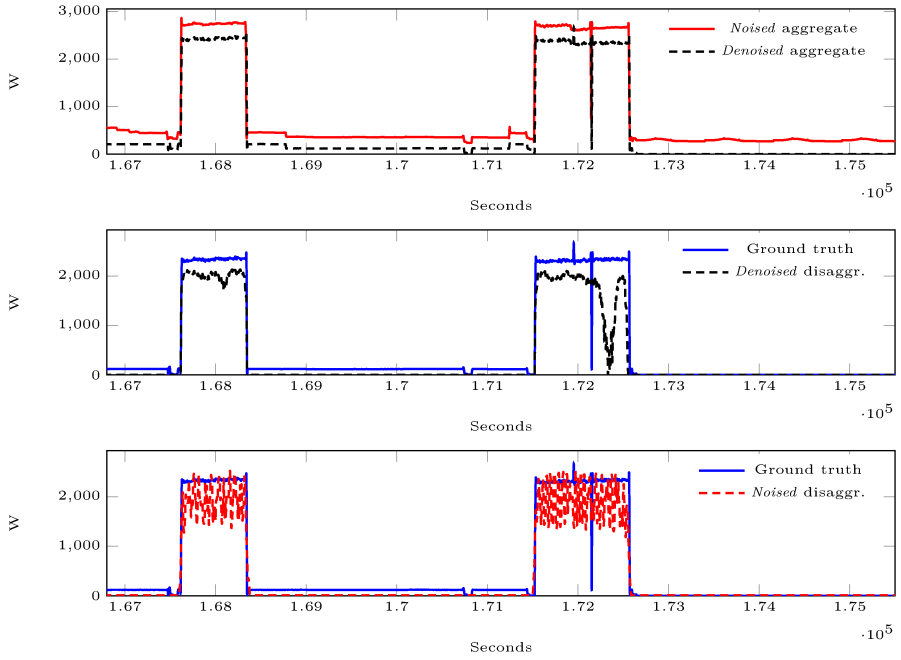


Figure 5.3: Disaggregated profiles in *denoised* and *noised* scenario in UK-DALE dataset, *seen* case study, related to the dishwasher in house 1.

appliance are not available in the house where the disaggregation in performed, therefore no training data can be considered to create the models. The generalisation property of the neural network allows to avoid a training procedure and to use the model trained on a set of data different from the test, whereas the footprints needs to be suitably extracted in order to train the HMM. One possible approach, in this sense, is represented by the user-aided footprint extraction algorithm, described in Section 4.4, that describes a procedure for the extraction of an approximated version of the appliance activations within the aggregated data when all the appliances are turned off, except the always-on in the house, i.e., the fridge and the freezer.

The experiments on the UK-DALE dataset have been performed as in [32], both for the *seen* and the *unseen* scenario.

The parameters related to the AFAMAP algorithm are defined as follows: the frame size is set to 60 minutes, which is an interval sufficiently large to include the whole activation for most of the appliances under study. For the ones with a longer activation, this frame size allows to include a complete operating sub cycle, for which the HMM is still representative. The variance parameters are set to $\sigma_1^2 = \sigma_2^2 = 0.01$ according to the variance of the experimental data, and the regularisation parameter is set to $\lambda = 1$. Table 5.3 presents the number of

states, defined a-priori for each class of appliance. In the *denoised* scenario no parameters optimisation has been conducted, whereas in the *noised* scenario, the number of the RoW states has been varied between the values $\{6, 8, 10\}$ for both datasets.

The algorithm has been implemented in Matlab, and the CPLEX[1] solver has been adopted to solve the QP problem. The experiments have been conducted on a working station equipped with an Intel i7 CPU at 3.3 GHz, and 32 GB RAM. The time required for an experiments depends on the number of samples and the number of states of the HMM models: because of the different sampling rate between the datasets, the experiments last from 1 hour for AMPds to 3 hours for UK-DALE, while the introduction of the RoW model increases the simulation time up to 2 hours for AMPds and 5 hours for UK-DALE.

The parameters related to the dAE approach are defined as follows: each network receives data in a mini-batch of 64 sequences, and a mean and variance normalization is computed on the input data. In order to guarantee the same normalization over the whole dataset, the mean and variance values are computed from a random sample of the training set. Whereas, on the target data a min-max normalization is performed using the maximum power consumption value of the related appliance. The training data is composed of 50 % of actual appliance related data, and 50 % of synthetic data obtained by randomly combining real appliance activations. The training sequences have been extracted by using NILMTK [73]: this toolkit provides the method for the power activation extraction from the ground truth power consumption related to each appliance from both datasets. The data analysing window of the dAE needs to be enough large to comprise an entire activation of the appliance, but not too much to include other contributions, especially for appliances with short-duration activation. The window width depends on the appliance type, as described in the Table 5.4:

As aforementioned, training has been performed by using the SGD algorithm with Nesterov momentum set 0.9. The maximum number of epochs has been set to 200 000, and the number of epochs for the variable step size technique has been set to 20 000. The initial value of the learning rate has been set to 0.1, with a decreasing factor equal to 10. The variable step size criterion has been applied on the F_1-Measure calculated on the validation set, and the relative tolerance for early stopping criterion has been set equal to 0.01. The neural network has been implemented by means of the Lasagne library[2], built on top of Theano [102]. All the network weights have been initialised randomly using Lasagne default initialisation, without any layerwise pre-training.

In [32], the network topology is composed of an input and an output con-

[1] `https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`
[2] `https://lasagne.readthedocs.io/en/latest/`

volutional layer with 8 kernels of size 4. The middle layers consists of 3 fully connected layers with ReLU activation functions, where the number of neurons in the central layer is equal to 128, whereas for the other layers the number depends on the length of the input sequence. In the disaggregation phase, a hop size of 16 samples has been considered. The performance of this work represents the baseline for this approach. An intensive parameters optimisation has been conducted regarding to the number of kernels (N), size of each kernel (S), and number of neurons in the central layer (H). The experiments have been conducted using each combination of parameters within the ranges: N={2, 4, 8, 16, 32, 64}, S={2, 4, 8, 16, 32, 64}, H={8, 16, 32, 64, 128, 256, 512, 1024, 2048}. Kernels larger than the input size have not been considered. The architecture that achieves the highest performance has been used as starting point of an additional campaign of experiment, for which the first convolutional layer has been preserved, and a second stage, including pooling and up-sampling layers, has been introduced. The parameters have been varied within the same ranges defined above.

Max pooling is calculated on a segment with sizes equal to 2 or 4 samples, and the overlapped portion is either equal to half of the window or not present. For this new architecture the experiments have been conducted with a full search of the optimal parameters. The disaggregation phase has been carried out with a sliding window technique over the aggregated signal, using overlapped window with hop size in the range $\{1, 2, 4, 8, \frac{1}{4}window, \frac{1}{2}window\}$, where *window* represents to the window width defined in Table 5.4.

The number of networks tested for each appliance in three datasets has been varied from 150 to 200, and this experimental campaign has been conducted on both *denoised* and *noised* scenario, in the *seen* and *unseen* conditions.

The experiments have been conducted on nVIDIA K80 GPUs. The training time varies depending on the network dimension and appliance type: because of the different sampling rates of the datasets, the experiments require from 2 to 10 hours depending on the size of the training set.

5.3.2 Results

Regarding the AFAMAP algorithm, in the *noised* scenario, preliminary experiments have demonstrated that the highest performance is obtained when the number of states of the RoW model is 6. For the sake of conciseness, only the results for that number of states are reported.

For the same reason, the results of the entire experimental campaign of the dAE algorithm will not be reported. For each scenario, the introduction of the second stage of CNN improves the performance with respect to the single CNN stage for the majority of appliances, as well as the effectiveness of the pooling

layer. The experiments demonstrated that a hop size with 1 and 2 samples results in the best performance.

For the AMPds and UK-DALE datasets, the dAE algorithm outperforms AFAMAP both in the *noised* and the *denoised* scenarios, as shown in Table 5.5, Table 5.6, Figure 5.5a, and Figure 5.5b. More in details, Figure 5.5 shows the radar charts related to the $F_1^{(E)}$ metric for each appliance, and the area inside a line gives an overall performance indicator of the related approach. On the AMPds dataset, in the *denoised* case study, the absolute improvement in terms of $F_1^{(E)}$ amounts to $+17.3\,\%$, while in the *noised* scenario the absolute improvements amounts to $+13.3\,\%$. The same trend can be observed by considering the other metrics. Compared to AFAMAP, NEP reduces by 2.012 in the *denoised* scenario, whereas it reduces by 3.819 in the *noised* scenario. State-based metrics show a similar trend, since, in the *denoised* case study, $F_1^{(S)}$ improves by $+24.7\,\%$, while and in the *noised* case study the absolute improvement is $+29.8\,\%$. Similar remarks apply to MCC. Analysing the performance of the individual appliances, the dAE algorithm outperforms AFAMAP for all the appliances in both the *denoised* and the *noised* scenario. In terms of $F_1^{(E)}$, the highest absolute improvement can be observed for the dishwasher ($+45.9\,\%$) in the *denoised* scenario, and for the oven in the *noised* scenario ($+48.4\,\%$). Considering the other metrics, the dAE algorithm outperforms AFAMAP for all the appliances in both scenarios, except for the fridge in the *noised* scenario, where AFAMAP achieves lower NEP and higher $F_1^{(S)}$. Indeed, for this appliance in the noised scenario, the performance improvement in terms of $F_1^{(E)}$ is modest compared to the other appliances.

Compared to AFAMAP, in the UK-DALE dataset the absolute improvement in terms of $F_1^{(E)}$ is $+4.4\,\%$ in the *denoised* case study, and to $+48.7\,\%$ in the *noised* scenario. The same trend can be observed by considering the other metrics: NEP reduces by 0.672 in the *denoised* scenario and by 11.564 in the *noised* scenario, while $F_1^{(S)}$ improves by $+11.7\,\%$ in the *denoised* case study and by $+36.51\,\%$ in the *noised* case study. MCC increases by 0.166 and by 0.466 respectively in the *denoised* and in the *noised* scenario. Analysing the performance of the individual appliances, the dAE algorithm achieves superior performance for all the appliances in the *denoised* scenario, except for the washing machine and the microwave, for which the $F_1^{(E)}$ is similar. In the *noised* scenario, the dAE algorithm outperforms AFAMAP for all the appliances, with the highest improvement equal to $+69.6\,\%$ for the kettle. The same trend can be observed considering the other metrics. In the *noised* scenario, the optimisation of the network parameters allows to outperform the dAE architecture presented in [32] for all the appliances, with the highest improvement of $F_1^{(E)}$ equal to $+26.1\,\%$ for the dishwasher. Considering the other metrics, the improvement follows the same trends, except for the washing machine evaluated

in terms of NEP, and the dishwasher evaluated in terms of $F_1^{(S)}$ and MCC.

Regarding the REDD dataset (Table 5.7), in the *denoised* scenario the performance difference of the dAE algorithm with respect to AFAMAP varies with the evaluation metric. In particular, in terms of $F_1^{(E)}$ and MCC, AFAMAP outperforms the dAE algorithm respectively by 6.5 % and 0.007. In terms of MCC, however, the relative improvement is limited, since it is equal to 0.95 %. In terms of NEP and $F_1^{(S)}$, the dAE approach outperforms AFAMAP as shown in the experiments with the UK-DALE and AMPds datasets. This behaviour can be explained by considering that in the *denoised* seen scenario the HMM models in AFAMAP are trained by using data of the same building used in the disaggregation phase, while the network in the dAE approach is trained by using multiple buildings, and testing is performing on one of those. This aspect is less relevant in the *noised* scenario, because in AFAMAP the RoW model introduces a high variability in the disaggregation solution. Indeed, in this scenario the dAE approach outperforms AFAMAP regardless the evaluation metric.

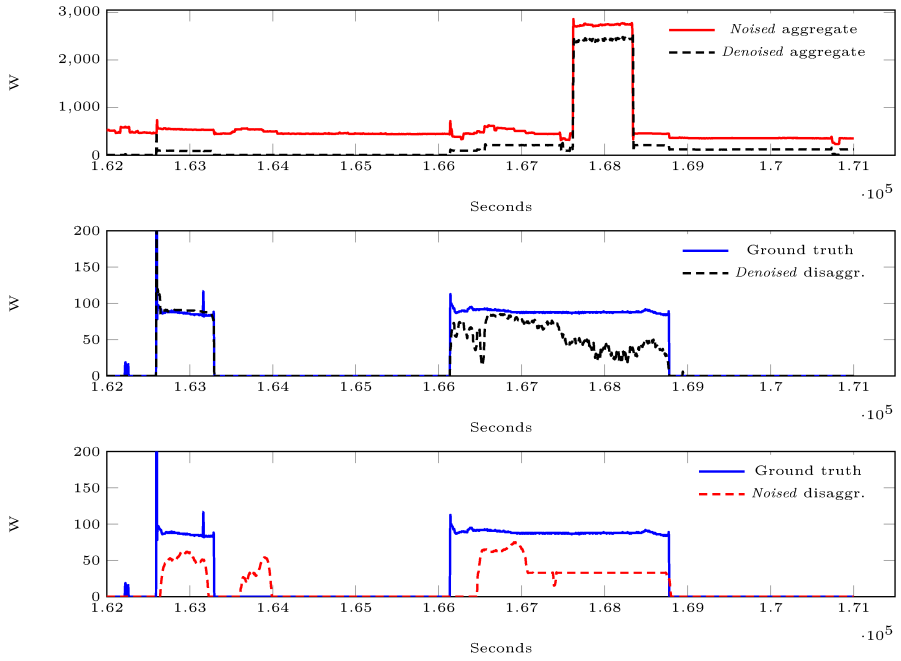


Figure 5.4: Disaggregated profiles in *denoised* and *noised* scenario in UK-DALE dataset, *seen* case study, related to the fridge in house 1.

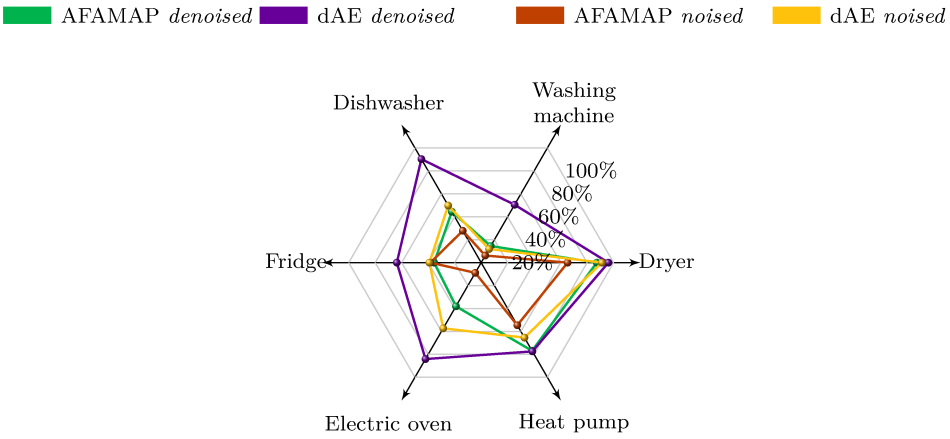
Generally, the dAE approach reaches higher disaggregation performance since it allows to reproduce complex activation profiles, which are learned during the training procedure and are associated to the aggregated profiles, even

Table 5.5: Disaggregation performance in the seen scenario (AMPds dataset). Numbers in bold indicate the best performing approach.

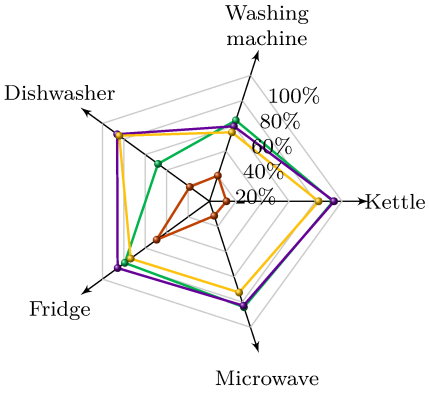
Scenario	Algorithm	Metric	Dryer	Washing machine	Dishwasher	Fridge	Electric oven	Heat pump	Overall
Denoised	AFAMAP [22]	$F_1^{(E)}$ (%)	87.3	14.5	44.4	35.5	38.1	76.9	60.4
		NEP	0.281	7.761	2.093	0.837	2.909	0.352	2.372
		$F_1^{(S)}$ (%)	60.7	7.4	11.9	36.0	5.0	86.2	50.3
		MCC	0.631	0.092	0.161	0.335	0.121	0.855	0.366
	dAE	$F_1^{(E)}$ (%)	96.1	50.5	90.3	63.7	84.1	77.4	**77.7**
		NEP	0.068	0.919	0.182	0.558	0.289	0.142	**0.360**
		$F_1^{(S)}$ (%)	76.0	54.8	76.8	75.6	53.4	93.4	**75.0**
		MCC	0.780	0.567	0.773	0.690	0.584	0.932	**0.721**
Noised	AFAMAP + RoW	$F_1^{(E)}$ (%)	65.3	6.2	27.8	38.3	8.9	54.6	40.8
		NEP	0.999	18.100	2.812	0.938	6.305	0.873	5.004
		$F_1^{(S)}$ (%)	16.0	7.6	10.7	43.3	2.0	55.5	30.8
		MCC	0.239	0.096	0.141	0.198	0.041	0.543	0.210
	dAE	$F_1^{(E)}$ (%)	91.2	11.9	49.8	39.1	57.3	65.4	**54.1**
		NEP	0.131	4.416	0.640	0.940	0.568	0.419	**1.185**
		$F_1^{(S)}$ (%)	76.8	10.8	58.2	33.1	45.9	79.8	**60.6**
		MCC	0.784	0.165	0.593	0.217	0.489	0.789	**0.506**

in the presence of the noise contribution. As shown in Table 5.5, Table 5.6 and Table 5.7, the highest performance is reached in the disaggregation of the appliances with higher peak power consumption, since it allows a better association between the target and the aggregated input sequence during the training phase. In the HMM based approach, each state of an appliance model represents one value of power consumption, which does not allow to represent highly variable or transient phenomena between the working states of the appliance. Additionally, in the AFAMAP algorithm the disaggregation solution is obtained by considering all the appliance models at the same time, while in the dAE approach each network operates independently from the others. This may cause a false energy assignment to an appliance, due to the need to satisfy the constraint that the sum of the reconstructed profiles corresponds to the aggregated power. In presence of noise, the performance degrades significantly, since the presence of the RoW, composed of a higher number of states compared to appliance models, increases the number of admissible solutions and, as a consequence, the chance of errors in the disaggregated profiles reconstruction. Moreover, in the AFAMAP algorithm there is no information on the total duration of the complete activation, since appliance models incorporate only the information on the working state transition and on the consumption values.

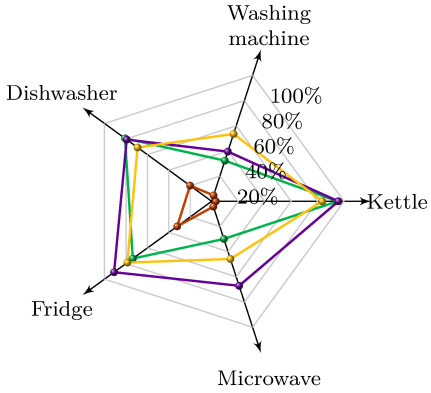
Further evaluations can be carried out by analysing the disaggregated profiles in *denoised* and *noised* scenario. Considering the UK-DALE experiments in *seen* scenario, the profiles related to the dishwasher in the house 1 are shown in Figure 5.3. The appliance activation is correctly detected by the dAE in both scenarios, without producing false positives in the disaggregated trace. In the *noised* scenario, the reconstructed profiles have a high uncertainty, caused



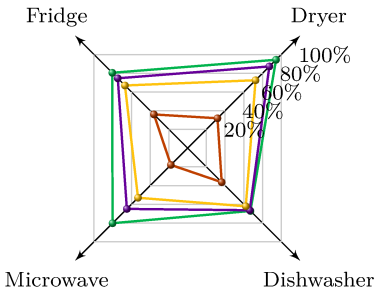
(a) Disaggregation performance on the AMPds dataset, *seen* scenario.



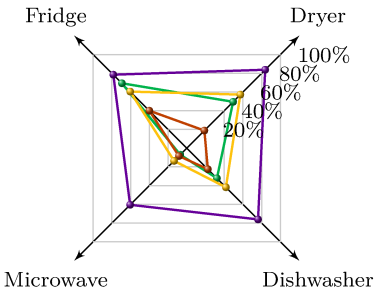
(b) Disaggregation performance on the UK-DALE dataset, *seen* scenario.



(c) Disaggregation performance on the UK-DALE dataset, *unseen* scenario.



(d) Disaggregation performance on the REDD dataset, *seen* scenario.



(e) Disaggregation performance on the REDD dataset, *unseen* scenario.

Figure 5.5: Performance for the different appliances for the all the addressed algorithms. The energy-based F_1-Measure (%) is represented.

Table 5.6: Disaggregation performance in the seen scenario (UK-DALE dataset). Numbers in bold indicate the best performing approach.

Scenario	Algorithm	Metric	Kettle	Washing machine	Dishwasher	Fridge	Microwave	Overall
Denoised	AFAMAP [22]	$F_1^{(E)}$ (%)	93.4	64.3	48.1	79.1	84.1	77.4
		NEP	0.435	14.090	1.322	0.358	1.038	3.449
		$F_1^{(S)}$ (%)	81.9	41.2	22.5	84.6	78.1	70.4
		MCC	0.797	0.451	0.287	0.781	0.788	0.621
	dAE	$F_1^{(E)}$ (%)	94.1	59.6	86.2	85.8	82.9	**81.8**
		NEP	0.087	13.087	0.220	0.207	0.287	**2.777**
		$F_1^{(S)}$ (%)	95.7	56.2	57.4	93.2	90.4	**82.1**
		MCC	0.957	0.559	0.620	0.896	0.903	**0.787**
Noised	AFAMAP + RoW	$F_1^{(E)}$ (%)	12.8	20.4	18.5	49.4	11.5	24.9
		NEP	1.754	53.063	1.752	0.865	4.193	12.325
		$F_1^{(S)}$ (%)	7.79	15.80	16.95	51.91	18.24	35.49
		MCC	0.150	0.145	0.179	0.324	0.177	0.195
	Kelly [32]	$F_1^{(E)}$ (%)	80.1	35.1	58.2	64.1	59.5	60.4
		NEP	0.522	1.384	0.707	0.609	0.923	0.829
		$F_1^{(S)}$ (%)	82.12	35.32	69.53	65.68	62.58	69.18
		MCC	0.821	0.372	0.706	0.575	0.626	0.620
	dAE	$F_1^{(E)}$ (%)	82.4	54.8	84.3	73.6	72.4	**73.6**
		NEP	0.393	2.135	0.278	0.472	0.524	**0.760**
		$F_1^{(S)}$ (%)	86.6	40.8	55.6	78.2	75.5	**72.0**
		MCC	0.866	0.425	0.583	0.683	0.751	**0.661**

by the presence of noise in the aggregated power, but the average energy in the activation has a good correspondence with the ground truth one, which demonstrates the low degradation of performance compared to the *denoised* scenario. The same experiment has been considered for the fridge, whose profiles are shown in Figure 5.4. The dAE algorithm recognises the appliance activation in the *denoised* scenario, with a less accurate profile reconstruction in the activation overlapped with other appliances respect to the isolated ones. Differently, the performance degrades in the *noised* scenario, with an incorrect activation detection and the production of some false positives, caused by the presence of noise in the aggregated signal.

As aforementioned, the *unseen* scenario is evaluated by using the UK-DALE and REDD datasets, due to the availability of recordings from several houses in both.

As in the *noised seen* scenario, preliminary experiments conducted by varying the number of states in the RoW model demonstrated that the highest $F_1^{(E)}$ is obtained with 6 states. Similarly, for the dAE algorithm the results of the entire experimental campaign will not be reported the sake of conciseness. For each scenario, the introduction of the second stage of CNN and of the pooling operation improves the performance with respect to the single CNN stage for the majority of the appliances. Regarding the hop size in the sliding window disaggregation phase, as in the *seen* scenario the highest performance is reached by using 1 and 2 samples.

Similarly to the *seen* scenario in the UK-DALE dataset, the baseline [32] performance for each appliance in the *noised* scenario is outperformed by means

Table 5.7: Disaggregation performance in the seen scenario (REDD dataset). Numbers in bold indicate the best performing approach.

Scenario	Algorithm	Metric	Dishwasher	Dryer	Fridge	Microwave	Overall
Denoised	AFAMAP [22]	$F_1^{(E)}$ (%)	67.1	94.4	80.5	80.2	**82.6**
		NEP	1.086	0.093	0.338	0.491	0.502
		$F_1^{(S)}$ (%)	50.12	97.59	89.79	66.86	78.85
		MCC	0.512	0.975	0.833	0.666	**0.746**
	dAE	$F_1^{(E)}$ (%)	66.3	87.3	74.6	64.9	76.1
		NEP	0.515	0.265	0.543	0.397	**0.430**
		$F_1^{(S)}$ (%)	71.7	92.7	80.5	69.6	**80.9**
		MCC	0.669	0.926	0.666	0.695	0.739
Noised	AFAMAP + RoW	$F_1^{(E)}$ (%)	36.4	31.9	36.0	17.9	35.4
		NEP	2.207	1.187	0.905	2.287	1.646
		$F_1^{(S)}$ (%)	32.9	57.6	39.6	16.2	46.0
		MCC	0.354	0.567	0.260	0.176	0.339
	dAE	$F_1^{(E)}$ (%)	62.1	72.5	66.9	53.0	**66.1**
		NEP	0.551	0.506	0.760	0.615	**0.608**
		$F_1^{(S)}$ (%)	64.0	81.8	70.9	61.6	**72.4**
		MCC	0.495	0.814	0.468	0.604	**0.595**

of the optimisation of the network parameters, with the highest absolute improvement of $F_1^{(E)}$ equal to $+30.2\,\%$ for the washing machine. The same trend can be observed for the other metrics, excepting for the $F_1^{(S)}$ and the MCC, where the dishwasher performance degrades.

For both datasets, the dAE algorithm outperforms AFAMAP in both scenarios, as shown and Table 5.9 and Table 5.8. In the UK-DALE dataset, the absolute improvement in terms of $F_1^{(E)}$ amounts to $+8.6\,\%$ in the *denoised* case study, whereas it increases to $+50.5\,\%$ in the *noised* scenario, demonstrating the superiority of the neural network based approach with respect to the HMM one, especially in presence of the noise contribution. The results evaluated with the other metrics confirm the same trend, with a reduction of NEP equal to 0.543 in the *denoised* case study and to 5.418 in the *noised* case study. Considering the state based metrics, the improvement evaluated with the $F_1^{(S)}$ amounts to $+12.52\,\%$ in the *denoised* scenario and $+53.10\,\%$ in the *noised*, as well as regarding the MCC with an absolute improvement of $+0.170$ in the *denoised* scenario and $+0.594$ in the *noised* scenario. As showed in Figure 5.5c, overall the dAE algorithm outperforms AFAMAP both in the *denoised* and in the *noised* scenarios. In particular, the dAE exhibits a noteworthy robustness against the presence of noise, while the $F_1^{(E)}$ of AFAMAP reduces significantly. Observing the results of each appliance, the highest absolute improvement is obtained for the kettle and it is equal to $+80.4\,\%$. In the *denoised* scenario, the dAE algorithm outperforms AFAMAP for all the appliances, with only exception of the dishwasher where the $F_1^{(E)}$ is $1.6\,\%$ lower. Considering the other metrics, in the *noised* scenario, the performance is improved for all the appliances, while in the *denoised* scenario the same trend can be observed,

Table 5.8: Disaggregation performance in the unseen scenario (REDD dataset). Numbers in bold indicate the best performing approach.

Scenario	Algorithm	Metric	Dishwasher	Dryer	Fridge	Microwave	Overall
Denoised	AFAMAP [22]	$F_1^{(E)}$ (%)	32.2	49.5	69.3	7.0	46.4
		NEP	3.336	0.811	0.491	4.754	2.348
		$F_1^{(S)}$ (%)	18.6	89.8	73.6	4.3	55.9
		MCC	0.282	0.901	0.650	0.056	0.472
	dAE	$F_1^{(E)}$ (%)	76.1	83.7	78.5	60.5	**76.6**
		NEP	0.348	0.292	0.426	0.470	**0.384**
		$F_1^{(S)}$ (%)	87.5	85.8	88.1	67.4	**84.2**
		MCC	0.877	0.860	0.805	0.711	**0.813**
Noised	AFAMAP + RoW	$F_1^{(E)}$ (%)	22.5	18.8	40.0	8.4	26.4
		NEP	3.803	1.521	0.946	3.728	2.500
		$F_1^{(S)}$ (%)	14.2	41.3	37.3	5.1	35.0
		MCC	0.228	0.399	0.180	0.083	0.222
	dAE	$F_1^{(E)}$ (%)	41.8	57.2	60.4	13.6	**47.6**
		NEP	0.756	0.955	1.053	1.752	**1.129**
		$F_1^{(S)}$ (%)	49.2	59.3	71.7	16.8	**54.6**
		MCC	0.543	0.617	0.497	0.166	**0.456**

except for the washing machine, which degrades its performance in terms of NEP, $F_1^{(S)}$ and MCC.

On the REDD dataset, the absolute improvement in terms of $F_1^{(E)}$ amounts to $+30.20\,\%$ in the *denoised* scenario and $+21.18\,\%$ in the *noised* scenario. The other metrics follow the same trends, with a reduction of NEP equal to 1.964 in the *denoised* case study and to 1.371 in the *noised* case study. Considering the state based metrics, the improvement evaluated with the $F_1^{(S)}$ amounts to $+28.3\,\%$ in the *denoised* scenario and $+19.60\,\%$ in the *noised*, as well as regarding the MCC with an absolute improvement of $+0.341$ in the *denoised* scenario and $+0.234$ in the *noised* scenario. In the REDD dataset, differently from the *seen* scenario described above, the dAE algorithm outperforms on each appliance in both scenario, with the highest improvements in terms of $F_1^{(E)}$ of $+53.51\,\%$ for the microwave, except for the dryer in the *denoised* scenario with the state based metrics. The radar chart represented in Figure 5.5e shows this improvement, and it represent the performance loss of both algorithm in the *noised* scenario with respect to the *denoised* scenario.

In the *unseen* scenario the generalisation property of the dAE approach allows to apply the model without the need of training, with a reasonable degradation of performance. Regarding the AFAMAP algorithm, the approximation introduced by the footprint extraction procedure causes a lack of correspondence between the HMM and the appliance working states consumptions, and this results in a higher performance degradation, particularly in presence of *noise* where RoW model is present. This demonstrates the effectiveness of the neural networks approaches in an *unseen* scenario, which is the most interesting condition, because it represents a real world application of the NILM service.

Table 5.9: Disaggregation performance in the unseen scenario (UK-DALE dataset). Numbers in bold indicate the best performing approach.

Scenario	Algorithm	Metric	Kettle	Washing machine	Dishwasher	Fridge	Microwave	Overall
Denoised	AFAMAP [22]	$F_1^{(E)}$ (%)	95.1	32.3	80.9	73.6	29.9	66.1
		NEP	0.114	2.089	0.457	0.449	3.311	1.284
		$F_1^{(S)}$ (%)	97.11	25.59	12.84	74.68	38.33	61.68
		MCC	0.971	0.353	0.177	0.690	0.440	0.526
	dAE	$F_1^{(E)}$ (%)	95.7	39.5	79.3	91.1	67.1	**74.7**
		NEP	0.056	2.406	0.371	0.195	0.675	**0.741**
		$F_1^{(S)}$ (%)	99.7	23.4	54.5	95.5	65.1	**74.2**
		MCC	0.997	0.286	0.604	0.931	0.664	**0.696**
Noised	AFAMAP + RoW	$F_1^{(E)}$ (%)	3.2	4.7	20.2	32.2	4.2	17.3
		NEP	3.087	6.559	2.078	1.021	18.413	6.231
		$F_1^{(S)}$ (%)	0	5.1	11.8	33.6	3.3	18.7
		MCC	-0.001	0.085	0.151	0.120	0.090	0.089
	Kelly [32]	$F_1^{(E)(\%)}$	79.1	23.3	39.2	65.1	20.6	50.8
		NEP	0.448	1.607	0.892	0.562	2.875	1.277
		$F_1^{(S)}$ (%)	93.9	26.5	55.9	77.8	30.9	66.8
		MCC	0.940	0.373	0.597	0.712	0.416	0.608
	dAE	$F_1^{(E)}$ (%)	83.6	53.5	69.2	78.7	45.8	**67.8**
		NEP	0.177	1.439	0.648	0.419	1.383	**0.813**
		$F_1^{(S)}$ (%)	95.6	67.5	50.9	82.8	45.4	**71.8**
		MCC	0.957	0.687	0.502	0.757	0.510	**0.683**

As described in the previous section, the state based metrics confirm that the dAE produces a more reliable activation detection, with respect to the HMM based approach, even in an unseen scenario.

5.4 Exploitation of the reactive power

The idea is based on the integration of the reactive power P_r in the structure of the dAE, described above in Section 5.3. Adding the information contained in the reactive power profile starts from the assumption that an increase in available information could improve the performance of the algorithm. Following this trend, the introduction of this variable within the network architecture needs the alteration of the network structure.

Since the focus of the disaggregation is represented by the reconstruction of the active power profile, the target of the network is maintained as the ground truth active power consumption of the dataset.

While, despite of using only the active power P_a, the input sequence in composed of two feature, i.e. the pair (P_a, P_r). This alteration reflect on the first CNN of the architecture, in which each kernel are instantiated with a depth equal to the number of channel, e.g. 2 on purpose. The features maps are produced by the convolution of the input sequence with the kernel, where the result from each channel is summed, in order to produce a feature map with unitary depth. Therefore, the unique alteration to the architecture is represented by the first CNN layer, whilst from the first pooling layer until the

output of the dAE the structure remains the same.

Since the output of the network is instantiated in order to reproduce only the active power profile, this network is not a proper *auto-encoder* structure, because of the mismatch between the input and target features size. Therefore, without losing in generalization during this dissertation, this architecture might be defined as *unbalanced auto-encoder*.

All the training procedure are maintained from the univariate dAE described in the Section 5.3: the network is trained on both activation and silence portion of the dataset, with the early-stopping procedure, in order to prevent the over fitting problem. The data augmentation is still performed, with the difference that both input features need to be considered, i.e. the synthetic composition of the aggregate signal is performed both for the active and reactive power, by exploiting the active and reactive power consumption related to each appliance activation in the ground truth trace.

5.4.1 Experimental setup

In order to conduct an exhaustive evaluation on different scenarios, two public datasets have been chosen: the Almanac of Minutely Power dataset (AMPds) [58] and the UK-DALE [61]. The choice of the target appliance is equal to what described in Subsection 5.3.1, whilst some variation need to be conduct, since the different availability of the P_r signal in the datasets.

Regarding the AMPds dataset, the scenario is not varied, since the availability of the reactive power both in the aggregate power signal and in the ground truth power traces, therefore it is possible to conduct the experiment in both *noised* and *denoised* scenarios, with the data augmentation in the training phase. Minor optimizations has been conducted with respect to the scenario described in Subsection 5.3.1, e.g. testing on additional hyper parameter values, therefore the reference only-P_a performance result slightly better, but this does not represent a relevant improvement.

Regarding the UK-DALE dataset, house 4 does not contains apparent power consumption, therefore it is excluded from the train buildings. Otherwise, the buildings assignation to compose the *seen* and *unseen* scenarios is not varied. Moreover, in each building, the reactive power consumption in the ground truth signals related to each appliance is not available, therefore the *denoised* scenario cannot be considered in the experimental phase. Moreover, for the same reasons, no data augmentation con be performed in the training phase. Therefore, the evaluation of the reference method described in Subsection 5.3.1 is revised, in order to proceed with a fair comparison.

The aggregate power consumption in the dataset has not the reactive power component, but the *apparent* power P_{app} is paired with the active power trace,

Table 5.10: Disaggregation performance in the seen scenario (AMPds dataset). Numbers in bold indicate the best performing approach.

Scenario	Algorithm	Metric	Dryer	Washing machine	Dishwasher	Fridge	Electric oven	Heat pump	Overall
Denoised	only P_a	$F_1^{(E)}$ (%)	96.1	50.5	90.3	63.7	84.1	77.4	77.7
	(P_a, P_r)		96.7	80.1	92.0	77.5	90.8	76.7	**86.2**
Noised	only P_a	$F_1^{(E)}$ (%)	94.8	11.7	57.3	39.0	64.9	67.6	57.4
	(P_a, P_r)		95.7	45.1	35.0	53.4	32.6	75.6	**61.4**

Table 5.11: Disaggregation performance in the *seen noised* scenario (UK-DALE dataset). Numbers in bold indicate the best performing approach.

Algorithm	Metric	Kettle	Washing machine	Dishwasher	Fridge	Microwave	Overall
only P_a	$F_1^{(E)}$ (%)	90.2	44.3	73.8	69.0	70.3	70.7
(P_a, P_r)		79.3	39.4	68.2	80.1	76.4	**71.5**

therefore using the inverse formula:

$$P_r = \sqrt{P_{app}^2 - P_a^2} \tag{5.4}$$

the aggregate reactive power trace is calculated.

Since the testing traces considered in the Subsection 5.3.1 are external to the UK-DALE dataset and they do not provide the reactive component, a variation to the portion of the train and test set need to be conducted. Indeed, the entire dataset, which was previously used for the training phase, is split in the new train (90 %) and test (10 %) sets. Within the train set, the activations for the validation phase has been taken in the portion of 10 %.

The parameter optimization campaign follows the same defined in Subsection 5.3.1, using an architecture with a double convolutional stage and the polling/upsampling stage, but a reduced range of values has been tested. The experiments have been conducted using each combination of parameters within the ranges: N={8, 32, 128}, S={4, 8, 16, 32}, H={256, 1024, 4096}. The number of kernels in the second stage is double the first stage ones. The pooling factor ids equal to {2, 4}, without overlapping. The experiments have been conducted on nVIDIA K80 GPUs.

5.4.2 Results

For the AMPds and UK-DALE datasets in the *seen* scenario, the (P_a, P_r) dAE algorithm outperforms the only P_a version both in the *noised* and the *denoised* scenarios, as shown in Table 5.10, Table 5.11, Figure 5.6a, and Figure 5.6b. More in details, Figure 5.6 shows the radar charts related to the $F_1^{(E)}$ metric for each appliance, and the area inside a line gives an overall performance indicator of the related approach. On the AMPds dataset, in the *denoised*

Table 5.12: Disaggregation performance in the *unseen noised* scenario (UK-DALE dataset). Numbers in bold indicate the best performing approach.

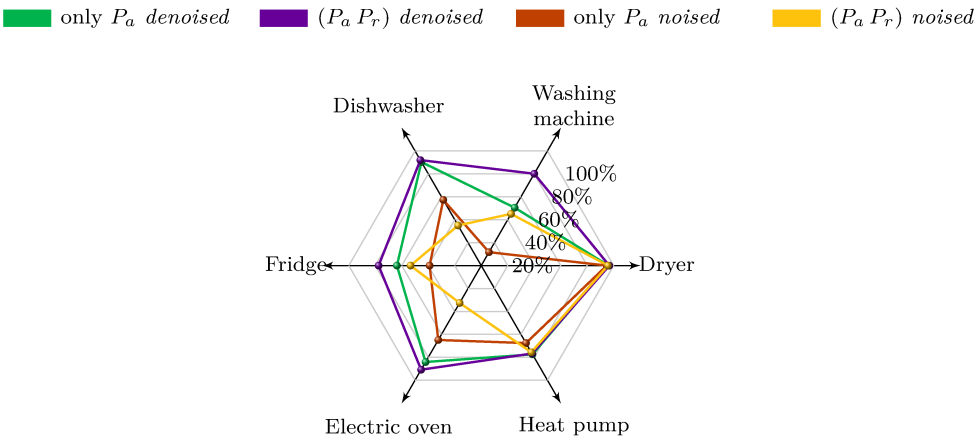
Algorithm	**Metric**	**Kettle**	**Washing machine**	**Dishwasher**	**Fridge**	**Microwave**	**Overall**
only P_a	$F_1^{(E)}$ (%)	85.7	25.8	51.8	72.9	10.3	**53.9**
(P_a, P_r)		68.4	25.9	56.5	82.6	14.0	52.5

case study, the absolute improvement in terms of $F_1^{(E)}$ amounts to $+8.5\,\%$, while in the *noised* scenario the absolute improvements amounts to $+4.0\,\%$. In general, the trend is maintained, with the *denoised* scenario reaches higher performance with respect to the *noised* scenario. Analysing the performance of the individual appliances, the (P_a, P_r) dAE algorithm outperforms the only P_a version for the most of the appliance, specially in the *denoised* scenario. In terms of $F_1^{(E)}$, the highest absolute improvement can be observed for the dishwasher, with $+29.6\,\%$ in the *denoised* scenario, and $+33.4\,\%$ in the *noised* scenario. In the *denoised* scenario, dishwasher and oven demonstrate an high degradation of the performance. On the UK-DALE dataset, in the *noised* scenario, the absolute improvement in terms of $F_1^{(E)}$ amounts to $+0.8\,\%$, since the improvement and the degradation of the performance in each appliance are close to be balanced.

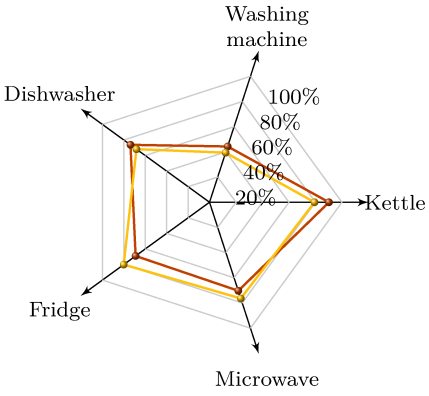
The motivation has to be researched in the generalization property of the network: the networks trained in the AMPds reaches higher improvements since the data augmentation has been performed, whereas on the UK-DALE has not been carried out.

For the UK-DALE datasets in the *unseen* scenario, the (P_a, P_r) dAE algorithm degrades the performance with respect to the only P_a version, for the most of the appliance, as shown and Table 5.12 and in the Figure 5.6c. The absolute difference in terms of $F_1^{(E)}$ amounts to $-1.4\,\%$, since the degradation on the most of appliances, whereas the improvements result to be quite absent in the remaining appliances.

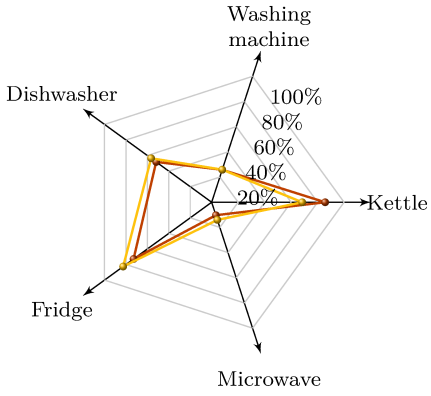
In the *unseen* scenario, the network in the UK-DALE dataset does not generalize since no data augmentation has been performed especially in presence of the noise contribution. Furthermore, the introduction of the P_r information causes a further degradation of the performance.



(a) Disaggregation performance on the AMPds dataset, *seen* scenario.



(b) Disaggregation performance on the UK-DALE dataset, *seen* scenario.



(c) Disaggregation performance on the UK-DALE dataset, *unseen* scenario.

Figure 5.6: Performance for the different appliances for the all the addressed algorithms. The energy-based F_1-Measure (%) is represented.

Chapter 6

Other contributions

6.1 Advanced Computational Intelligence for Smart Water and Gas Grid

In recent years, the *smart grid* paradigm has drawn the attention of a wide set of machine learning and computation intelligence approaches. Specifically, main efforts have been focused on energy fields, leaving aside water and, especially, natural gas fields. Even if many technology advancements have been recently achieved in metering and monitoring systems [103, 104], a large gap remains between energy achievements and water, and natural gas ones.

Great attention has been paid to problems concerning the resource monitoring aimed to avoid unnecessary waste, specifically for water and natural gas [105, 106].

Since the relevance of those issues, short-term predictions of water and natural gas consumption are performed exploiting state-of-the-art techniques, whereas automatic leakage detection in smart water grids is approached with unsupervised techniques.

6.1.1 Short/Medium-Term Load Forecasting

As highlighted by Fagiani *et al.* [107], a conspicuous lack of available datasets is present, and an appropriate comparison between the few developed approaches, for both forecasting or fault detection applications, is prevented by the employment of non-standard evaluation criteria.

For this reason, a comparative evaluation among the state-of-the-art forecasting techniques has been conducted, moving from the work presented by Fagiani *et al.* in [108].

Genetic Programming (GP), Support Vector Machine for Regression (SVR), Artificial Neural Networks (ANNs), Echo-State Networks (ESNs), Deep Belief Networks (DBNs), and Extreme Learning Machine (ELM) have been tested with the presented datasets. Differently from ANNs, that have been widely adopted for water forecasting purposes [109, 110, 111], only recently GP and

adaptive ELM have been applied for water prediction in [112] and [113], respectively. Unfortunately, as highlighted in [107], each study presents its results with a different set of evaluation criteria, and therefore, a proper comparison between them is not possible, as well as identify the best approach.

Evaluated approaches

In this section, the adopted ranges of the parameters are presented for each selected prediction technique.

Genetic Programming

The GP evaluations have been performed adopting the GPLAB Toolbox [114], and selecting the operators plus (+), minus (−), product (×), division (÷), power (x^n), sine (sin), and cosine (cos). Moreover, the population size has been set to 100, whereas 1000 generations have been evaluated. Finally, three depths of the nodes have been evaluated, 20, 15, and 10, in order to find a best individual.

Support Vector Machine for Regression

The Radial Basis Function (RBF) has been chosen as kernel function, and the search grid approach has been performed assuming the following values for C and γ:

$$C = \left\{2^{-5}, 2^{-3}, \ldots, 2^{15}\right\}, \ \gamma = \left\{2^{-15}, 2^{-12}, \ldots, 2^{3}\right\}.$$

Extreme Learning Machine

The Radial Basis Function (RBF) kernel has been evaluated for the kernel-based approach version [115].

Artificial Neural Network

In the experiments, one of the Radial Basis Function has been used, specifically the one defined as: $a = \exp(-n^2)$. Moreover, the tests have been executed using a network with one input layer, one hidden layer and one output layer, and the number of hidden nodes has been varied from 5 to 15.

Echo State Network

The experiments have been conducted evaluating different combinations of *reservoir* size, initial transient value, leaking rate (α), and regularization coefficient (β). Specifically, accordingly to Lukoševičius guide [116], the range of

each parameter has been set as follows:

$$\begin{aligned} \textit{reservoir size} &= \{2, 5, 10, 15, 20, 25, 30, 35, 40\}\,, \\ \textit{initial transient} &= \{0, 1, 2, \ldots, 15\}\,, \\ \alpha &= \left\{1, \frac{1}{2}, \frac{1}{3}, \ldots, \frac{1}{10}\right\}, \\ \beta &= \left\{1, 10^{-1}, 10^{-2}, \ldots, 10^{-15}, 0\right\}. \end{aligned}$$

Deep Belief Network

Following the recommendation in Hinton [117], the tests have been performed splitting the training data in "mini-batches" of 10 and 100 elements. Moreover, for the maximum number of iteration and the number of nodes have been set the following ranges:

$$\textit{Nodes} = \{2, 3, \ldots, 10\}\,, \qquad \textit{Max Iter} = \{100, 300, 500\}\,.$$

where *Nodes* refers to the neurons adopted in the hidden layer, whereas the number of output neurons is set to one, and the number of input neurons is set equal to the number of input data. Finally, the experiments have been conducted using the DBN MATLAB® code [118, 119].

Computer Simulations and Results

The forecasting experiments have been conducted adopting the publicly available datasets, AMPds [120] and DIFD. Both the datasets present water and natural gas consumption, but they belong to different level of the micro-grid scenario. The former refers to the domestic level, and allows the study of heterogeneous information having also power consumption and environmental temperature data. The latter provides consumption at building level, and is composed of four sub-sets, two for water and two for natural gas consumption. Unfortunately, the pair of subsets that provides water and gas recording from the same building, Abercrombie House, cannot be combined to study heterogeneity effect due to defects of the sets, such as missing data and different lengths. Concerning the AMPds data, in order to provide a exhaustive evaluation of the application of heterogeneous inputs, the prediction of the output resources, water and gas, have been computed for all the possible combinations of available data, thus, a total of 8 combinations as reported in Table 6.1. Furthermore, to provide a thorough evaluation of the seasonality effect on the prediction performance, Table 6.1 reports also the results achieved in Fagiani *et al.* [108], that refer to the consumption of the 1 year set. Moreover, the experiments have been performed for $1\,h$, $6\,h$, $12\,h$ and $24\,h$ of time resolution.

For each set, the approaches have been trained using 70% of the data, ran-

Table 6.1: Best results achieved for each technique applied to AMPds database released in 2013 and 2014, with a length of 1 year and 2 years, respectively. The column marked with "Comb." reports the resources combination that achieves the best result.

AMPds	1 *h*			6 *h*			12 *h*			24 *h*		
	NMSE	R^2	Comb.	NMSE	R^2	Comb.	NMSE	R^2	Comb.	NMSE	R^2	Comb.
	Natural Gas Prediction - 1 year											
ANN	0.866	0.314	WGET	0.276	0.723	GET	0.200	0.799	WGET	**0.176**	**0.823**	**G**
DBN	0.785	0.215	WGT	0.340	0.660	WGT	0.201	0.798	WGET	0.199	0.800	GT
ESN	0.740	0.260	GT	0.333	0.666	GET	0.213	0.786	WGET	0.231	0.767	GT
SVR	0.733	0.267	GET	0.269	0.731	WGET	**0.191**	**0.809**	**GT**	0.250	0.748	GT
ELM	**0.700**	**0.299**	**WGET**	**0.266**	**0.733**	**WGET**	0.194	0.805	WGET	0.241	0.757	GT
GP	0.798	0.202	WGT	0.328	0.671	GET	0.283	0.715	GT	0.299	0.698	WGT
	Natural Gas Prediction - 2 years											
ANN	**0.619**	**0.381**	**WGT**	0.182	0.818	GET	0.170	0.830	GET	**0.153**	**0.847**	**GT**
DBN	0.723	0.277	WGET	0.238	0.762	WGET	0.185	0.815	GET	0.188	0.812	GET
ESN	0.695	0.305	WGET	0.247	0.753	GET	0.176	0.823	GET	0.188	0.811	GT
SVR	0.644	0.356	WGET	0.182	0.818	WGET	**0.145**	**0.854**	**WGT**	0.178	0.821	GT
ELM	0.620	0.380	WGET	**0.180**	**0.820**	**WGET**	0.147	0.853	WGT	0.176	0.823	GT
GP	0.753	0.247	WGET	0.273	0.728	GET	0.186	0.813	GT	0.204	0.795	WGT
	Water Prediction - 1 year											
ANN	**0.745**	**0.253**	**WET**	0.343	0.656	WGT	**0.408**	**0.590**	**WE**	**0.345**	**0.652**	**W**
DBN	0.876	0.124	WGET	0.401	0.598	WGET	0.573	0.424	WT	0.415	0.581	WGET
ESN	0.815	0.184	WET	0.367	0.632	W	0.522	0.475	W	0.334	0.663	WGT
SVR	0.817	0.182	WGET	**0.299**	**0.700**	**WET**	0.484	0.513	WET	0.352	0.645	WGT
ELM	0.769	0.231	WGET	0.310	0.689	WGET	0.490	0.507	WT	0.351	0.646	WGET
GP	0.908	0.091	WET	0.384	0.615	WE	0.535	0.463	W	0.389	0.607	WGE
	Water Prediction - 2 years											
ANN	0.790	0.253	WET	0.319	0.680	W	**0.223**	**0.777**	**W**	**0.550**	**0.448**	**W**
DBN	0.867	0.133	WGET	0.407	0.592	WGET	0.317	0.682	WGET	0.607	0.390	WE
ESN	0.818	0.181	WGET	0.413	0.587	WGET	0.308	0.691	W	0.581	0.417	W
SVR	0.806	0.194	WGE	**0.316**	**0.683**	**WGET**	0.269	0.730	W	0.581	0.416	W
ELM	**0.765**	**0.235**	**WGET**	0.319	0.681	WGET	0.279	0.720	WET	0.575	0.422	W
GP	0.912	0.088	WGET	0.365	0.636	W	0.297	0.702	WGET	0.617	0.380	WGT
W = water			G = natural gas			E = electric power			T = temperature			

domly selected. Then, the evaluation criteria have been computed using the test set, composed of the remaining 30% of the data. For the stochastic approaches, such as ANN, ESN, DBN, ANN, and GP, each parameters combination has been evaluated 10 times. Only the best results have been reported in Table 6.1 and Table 6.2. Moreover, the tests have been conducted assuming 2, 3, and 5 time-lagged observation as input data. Therefore, for the neural network based approaches, the number of input neurons has been chosen accordingly.

The normalized mean square error (NMSE), the determination coefficient (R^2, commonly known as Nash-Sutcliffe efficiency coefficient [121]), the mean square error (MSE), the mean absolute percentage error (MAPE) and the relative root mean square error (RRMSE) have been adopted as evaluation criteria.

For the sake of the clarity, the only evaluation criteria reported in Table 6.1 and Table 6.2 are the NMSE and R^2. In addition, for the AMPds, Table 6.1, the column "Comb." reports the heterogeneous input data combination that achieves the best performance for the specific resolution. Details about the corresponding parameters are given in the text for points of interests. Whereas, for the DFID in Table 6.2, the column "Param." reports the value of the pa-

rameters for the reported performance. Specifically, for the ANN the values correspond to the lags number and the number of hidden neurons, respectively. Number of lags, number of nodes, maximum iterations number and "mini-batches" size, respectively, are reported for the DBN. In the ESN, the number of lags, the reservoir size, the initial transient, the leaking rate, and the regularization coefficient are reported. For SVR and ELM, number of lags, C and γ values are indicated, respectively. The number of lags and the maximum depth are reported for GP.

AMPds Dataset

Table 6.2: Best results achieved for each technique applied to DFID dataset. The column marked with "Param." reports the parameters combination that achieves the best result for the corresponding approach.

DFID	Overall GAS			Abercrombie House GAS			Abercrombie House WATER			Whitehall WATER		
	NMSE	R^2	Param.	NMSE	R^2	Param.	NMSE	R^2	Param.	NMSE	R^2	Param.
ANN	**0.266**	**0.734**	**5-13**	**0.275**	**0.725**	**5-13**	**0.676**	**0.324**	**5-13**	**0.121**	**0.879**	**5-6**
DBN	0.509	0.491	3-1-2-500	0.493	0.507	5-1-2-500	0.844	0.156	2-1-2-100	0.578	0.422	2-1-2-100
ESN	0.395	0.605	5-25-1-1	0.396	0.604	5-25-1-1	0.750	0.250	5-40-5-1	0.367	0.633	5-40-10-1
SVR	0.274	0.726	$5\text{-}2^{-1}\text{-}2^{3}$	0.298	0.702	$5\text{-}2^{-1}\text{-}2^{3}$	0.753	0.247	$5\text{-}2^{9}\text{-}2$	0.150	0.850	$5\text{-}2^{5}\text{-}2^{3}$
ELM	0.267	0.733	$5\text{-}2^{3}\text{-}2^{3}$	0.286	0.714	$5\text{-}2^{3}\text{-}2^{3}$	0.686	0.314	$5\text{-}2^{7}\text{-}2^{3}$	0.143	0.857	$5\text{-}2^{13}\text{-}2^{3}$
GP	0.457	0.543	5-20	0.535	0.465	5-20	0.891	0.108	5-10	0.343	0.657	5-20

The natural gas results are reported in the upper half of Table 6.1. The results show that the overall best performance is reached by the SVR approach with 2 years data at 12 h resolution. Whereas, for 1 year data, the best result is achieved at 24 h resolution with ANN, without using heterogeneous data. Moreover, the major improvements of the 2 year predictions with respect to 1 year ones are achieved for 6 h and 12 h resolution. For both sets, a best improvement in the results is performed passing from 1 h to 6 h of resolution. The results confirm that the heterogeneous data are essential information for the prediction, even if it seems reducing the data resolution decreases the number of heterogeneous components. Specifically, in the 1 year set, the best results at 1 h and 6 h resolution are obtained merging all the heterogeneous components, whereas for 12 h and 24 h resolution the best results are achieved with the pair gas-temperature and with the gas consumption only, respectively.
Finally, for the 2 years set, the reduction of the heterogeneous components does not appear as marked as in the 1 year set. Specifically, the best results at 1 h and 12 h resolution are achieved with three heterogeneous components (WGT). At 6 h resolution the best result is achieved with four components (WGET), and only two components (GT) are used as input for the best result at 24 h resolution.

The water predictions are reported in the lower half of Table 6.1. Differently from the natural gas, the best results achieved for the 2 years set at 1 h, 6 h and 24 h of resolution experienced a severe deterioration with respect to the 1

year ones. Despite this, the ANN obtained the overall best result with the 2 years dataset at $12\,h$ resolution. For 1 year data, the overall best performance is achieved by the SVR at $6\,h$ resolution with 5 lags and an heterogeneous input composed of water, energy, and temperature data (WET). For the 1 year set, a general reduction of the heterogeneous components is shown only for the results at $12\,h$ resolution. For the 2 year set, the heterogeneous data do not provide a clear improvement at the prediction, and a clear reduction in the number of heterogeneous components is shown when the resolution decreases. Specifically, at $12\,h$ and $24\,h$ resolution, the best performance are achieved with only water consumption information as input. Noteworthy, for 1 year consumption, with the exception of ESN, all the approaches achieve their best results at $6\,h$ resolution, then the performance show a slight degradation for 12 and $24\,h$ resolution.

Finally, for both water and gas prediction, ESN, DBN, and GP approach have confirmed to be unsuitable for short-term prediction in domestic environment. On the contrary, ANN, SVR, and ELM have shown to be effective for short-term forecasting on this dataset.

DFID Dataset

For all the DFID sub-sets the ANN approach achieves the best performance, as depicted in Table 6.2. The longest sub-sets are the gas ones, about 5 years of records, and they achieve close results. For all the sub-sets, the best performance is achieved using 5 lags, and both SVR and ELM confirmed their good behaviour for short-term prediction. As for the AMPds, ESN, DBN, and GP are unsuitable for this case scenario. Noteworthy, all the approaches show a marked performance deterioration for the Abercombrie House water sub-set.

6.1.2 Automatic Leakage Detection

Statistical approaches have been widely adopted in order to resolve the novelty detection issue in several fields [122, 123], but the detection of abnormal events has not been applied to the residential networks, yet.

The study presented in [106] has been further developed by exploiting additional information, in order to improve the system performance. Specifically, the extension aims to introduce both pressure and temporal information in addition to the flow features extracted from the water household consumption.

Gaussian Mixture Model (GMM), Hidden Markov Model (HMM), and One-Class Support Vector Machine (OC-SVM) have been adopted to characterize the normality model. To identify the leakages, log-likelihood has been used in the case of GMMs and HMMs, whereas the distance from the hyperplane has been used in the case of OC-SVM.

The Algorithm

The adopted approach, proposed in [106], for the leakages detection is composed of a preliminary stage of *normality model creation*, and of the actual *novelty detection* stage. In both stages, the input data, training and test set, respectively, are processed in the *features extraction* block in order to properly extract the selected features. In the *normality model creation* stage, these features are used to create the *normality models*, whereas in the *novelty detection* stage, the features are used to compute the log-likelihoods (GMM and HMM), or the distances to the hyperplane (OC-SVM), by assuming as reference models the normality ones. For the *novelty detection* stage, the final decision, leakage detected or not, is obtained by comparing the given log-likelihoods/distances against a set of pre-computed thresholds.

Features

The first part of the adopted features has been presented in [106]. This set has been expanded by introducing two subset of temporal features, namely the *frame position in temporal window* (FPW) and the *temporal window enumeration* (TWE). Each of these subsets is composed of three features, computed for different temporal windows: hourly, daily, and weekly, respectively.

About the temporal features, the hourly window starts at the first minutes of the hour and ends at the last minute of the same hour. The daily window goes from the first minute (higher resolution for the adopted dataset) after the midnight, 00:01, to the midnight of the next day, 00:00. Finally, the weekly window starts at the first minute after the midnight of Sunday, and ends at the midnight of Sunday.

The FPW features provides the frame position within the temporal window. Specifically, all the samples within a temporal window are enumerated, and for each frame the assigned feature corresponds to the value assigned to its first sample, thus it represents the starting position of the frame with respect to the overall sample set within the temporal window.

On the other hand, regarding the TWE features, the sequence of temporal windows are enumerated, and the feature identifies which temporal window the frame belongs to. About the hourly window the evaluated window sequence spans over a day; for the daily window the evaluated window sequence spans over a week; finally, for the weekly window the evaluated window sequence spans over a year. Specifically, for each frame, the assigned feature value corresponds to the position, within the sequence of temporal windows, of its first sample. Thus, according to the selected temporal window, it represents the starting position expressed in terms of hour of the day, day of the week, or week of the year.

Novelty Detection: Decision

The detection of a leakage is carried out by performing a frame-by-frame

decision. For each frame the log-likelihood, or distance to the separation hyperplane, computed by adopting the background model, and is compared against a set of thresholds.

Therefore, once established the set of thresholds for each approach, for each threshold a decision vector is produced. The vector is created out of the decision assumed for each frame: if the likelihood or distance value of the frame exceed the threshold, the frame is labelled as abnormal, otherwise it is marked as normal.

Experimental set-up

Simulated Network

Both pressure information and leakages have been computed/simulated by means of the EPANET tool [124]. The simulated circuit has been designed in order to represent a typical domestic network. The network has a length of $30\,m$, with an overall elevation of $3\,m$. The water supplier is simulated by means of a reservoir with an elevation of $30\,m$, that is defined in the EPANET Users The available data pieces, about water consumption from the AMPds, are used as consumption patterns in specific junctions or nodes.

The network assumes a $20\,mm$ diameter standard piping, and a roughness (C-factor) of 100. The output pressure and flow values are computed from the first junction of the network, connected to the reservoir, and from the corresponding connection pipe, respectively.

Leakages

In all the experiments the datasets have been split in frames with a constant length of 5 hours, and an overlap factor of 2/3. The evaluations based on the AMPds water set have been performed by assuming three sampling resolutions: 1, 10, and 30 minutes.

Each dataset has been split in training and test set, composed of 70% and 30% of the original set, respectively.

For each tested condition, 10 background models have been trained, and for each of them 10 random leakages have been alternately introduced in the simulated network. In this way, an overall of 100 random leakages have been induced, by producing random variations in their size, starting point, and duration. The length of the leakages has been randomly selected between 5 and 10 hours, that corresponds to 300 and 600 samples, 30 and 60, or 10 and 20, at 1, 10, and 30 minutes of resolution, respectively.

In this work, the *simulated* leakages have been modelled by means of an *emitter* in the simulated circuit. In this way, the leakage output flow (leakage size) depends on flow and pressure in the reference node. As done in Nasir *et al.* [125], the leakage size has been varied by manipulating the discharge coefficient of the emitter. According to the EPANET Users Manual, "the emitter models

the flow through a nozzle or orifice that discharges to the atmosphere. The flow rate through the emitter varies as a function of the pressure available at the node":

$$Q = C \cdot P^{g}, \tag{6.1}$$

where Q is the flow rate, P is the pressure, C the discharge coefficient, and g is the pressure exponent, usually equal to 0.5 for nozzles and sprinkler heads. For the simulated circuit both flow and pressure data are collected, measuring them from the input (first) pipe and node, respectively. A random discharge coefficient $C \in [0.024, 0.047]$ has been assumed, in order to evaluate leakages compliant with a real-case scenario [126].

In addition, for both GMM and HMM, different numbers of Gaussian components have been adopted, $N_g = \{2, 4, 8, 16, 32, 64, 128, 256\}$, and, furthermore in the case of HMM, the state number (excluding the start and end states) has been varied as well, among 1 and 4. About the OC-SVM, the kernel parameter γ has been given the following values: $\left\{2^{-15}, 2^{-13}, \ldots, 2, 2^{3}\right\}$.

Simulations and Results

Features selection

In order to reduce the number of features combinations, and, at the same time, to avoid erroneous selection while performing the SFS [127], after evaluating the features separately, the ones that achieved the three best results, are selected. Each of them is used to perform the second step of the SFS, where the selected feature is combined with the remaining features, one at a time, and evaluated. Once completed the second step for the winning features, only the overall best combination is selected to proceed with the next steps.

To finalize the dataset, each features matrix, obtained as combination of the available features for all the frames, has been normalized with *min-max* technique.

Each decision vector produced in the decision phase, has been evaluated in terms of *true detection rate* (TDR), and *false detection rate* (FDR).

The overall performance is expressed in terms of *Receiver Operating Characteristics* (ROC), obtained by combining the TDR and FDR results for all the threshold values for each feature combination and for each set of parameters.

Results discussion

In this section, the experiments performed with FPW and TWE features are reported in Table 6.3 and 6.4, respectively. When jointly addressing the features extracted from flow and pressure data, the flow based feature will include an additional **f**, the pressure based features will include a **p**.

Concerning the FPW features in Table 6.3, as for the previous experiment, all the best performance are achieved with HMM, followed by GMM, with

the exception of those based on pressure data at 30 minutes resolution where OC-SVM perform slightly better that HMM. Even if the pressure-based results show the lowest performance, with a very high standard deviation, they present a reverse trend with respect to flow-based ones. Decreasing the resolution produces lower results for the flow data, and better results, instead, for the pressure ones. Moreover, by combining the features extracted from flow and pressure, the trend exhibited by the flow is mitigated, and reducing the resolution produces slightly better results.

Table 6.3: Best results and corresponding features combination achieved for each resource and resolution with FPW temporal features. The "Param." column reports the number of Gaussians adopted for the GMM, the states and Gaussians number for the HMM, or the γ for the OC-SVM.

Res.	Features comb.	AUC (%)	STD	Model	Param.
Pressure Data					
1	dWPEC+WEEK+DAY+dMA+dENE	56.14	21.91	GMM	2
1	MA+WEEK+dWDE	**56.21**	24.83	HMM	4-4
1	dMA+DAY+WEEK+dENE	55.70	26.73	OC-SVM	2
10	dENE+DAY+WEEK+MA+HOUR+ENE	56.44	19.07	GMM	2
10	dWDE+WEEK+WDE	56.78	24.92	HMM	4-32
10	dDATA+WEEK	**57.28**	28.28	OC-SVM	2^{-1}
30	WPEC+WEEK+MA	56.33	19.74	GMM	4
30	WPEC+DAY+WEEK+ENE+WDE	**57.44**	25.49	HMM	3-8
30	WPEC+WEEK	57.25	28.29	OC-SVM	2^{-5}
Flow Data					
1	MA+ENE	80.88	10.56	GMM	128
1	MA+ENE+DAY	**86.14**	11.18	HMM	3-128
1	MA+WDE+ENE	60.44	19.06	OC-SVM	2^{-7}
10	MA+DAY+WEEK	70.11	15.08	GMM	256
10	WPEC+DATA+dWPEC+DAY	**78.56**	15.50	HMM	4-256
10	MA+DATA	65.55	27.26	OC-SVM	2^{-7}
30	MA+WEEK	69.98	15.46	GMM	256
30	DATA+DAY	**78.72**	12.78	HMM	4-256
30	DATA+dWDE	67.52	20.75	OC-SVM	2^{3}
Flow&Pressure Data					
1	MAf+ENEp+MAp+dMAp+dENEp	85.95	8.62	GMM	32
1	MAf+ENEf+MAp	**87.70**	9.54	HMM	3-256
1	WDEf+dWPECp	61.29	18.75	OC-SVM	2^{-15}
10	MAf+ENEp+MAp+dENEp+dMAp	86.25	8.55	GMM	32
10	MAf+ENEp+MAp	**87.64**	8.94	HMM	3-64
10	MAf+DATAf+dMAp	65.59	27.23	OC-SVM	2^{-7}
30	MAf+MAp+ENEp+dENEp	86.61	8.86	GMM	32
30	MAf+ENEp+MAp	**88.06**	9.35	HMM	3-64
30	DATAf+dWDEf	67.52	20.75	OC-SVM	2^{3}

About the experiment with TWE features, the achieved results are reported in Table 6.4, and the same trends shown for FPW features are confirmed. Regarding the pressure data, the exploitation of TWE features produces a significant improvement with respect to the FPW one for both GMM and HMM. On the other hand, when based on the flow data, the TWE features at 1 minute resolution do not produce better performance, instead at 10 and 30

minutes of resolution, the TWE temporal features allow to reach better results than the FPW ones. Finally, about the combination of flow and pressure data, even the TWE features do not produce positive contribution, achieving a result similar to the previous experiments.

Table 6.4: Best results and corresponding features combination achieved for each resource and resolution with TWE temporal features. The "Param." column reports the number of Gaussians adopted for the GMM, the states and Gaussians number for the HMM, or the γ for the OC-SVM.

Res.	Features comb.	AUC (%)	STD	Model	Param.
Pressure Data					
1	dENE+WEEK+dMA+MA+ DAY+HOUR	59.78	23.97	GMM	2
1	dENE+WEEK+dMA	**61.48**	24.32	HMM	3-32
1	dENE+DAY	56.88	27.58	OC-SVM	2^{-9}
10	dENE+WEEK+dWDE	59.27	20.61	GMM	2
10	dENE+WEEK	**61.09**	24.10	HMM	3-32
10	DATA+DAY	57.46	27.02	OC-SVM	2^{-3}
30	dDATA+WEEK+DAY+dMA	58.93	21.13	GMM	64
30	WPEC+WEEK+MA	**63.64**	21.88	HMM	4-16
30	dDATA+DAY	56.78	26.47	OC-SVM	2^{-13}
Flow Data					
1	MA+ENE	80.83	10.43	GMM	64
1	MA+ENE+WPEC	**85.54**	11.74	HMM	4-128
1	MA+WDE	60.43	19.02	OC-SVM	2^{-7}
10	MA+HOUR+dMA	69.82	16.48	GMM	256
10	DATA+WPEC+HOUR+ dWPEC+WEEK+MA	**79.40**	15.11	HMM	4-256
10	DATA+MA	65.55	27.26	OC-SVM	2^{-9}
30	DATA+HOUR+dMA+MA	71.94	14.44	GMM	256
30	DATA+HOUR+WEEK	**79.64**	14.85	HMM	4-256
30	DATA+dWDE	68.39	20.86	OC-SVM	2^{3}
Flow&Pressure Data					
1	MAf+MAp+ENEp+dMAp	86.04	8.80	GMM	32
1	MAf+ENEf+ENEp+MAp	**87.73**	9.34	HMM	3-64
1	MAf+WDEf	60.43	19.02	OC-SVM	2^{-7}
10	MAf+MAp+ENEp+dENEp+ dMAp	86.26	8.58	GMM	32
10	MAf+ENEp+MAp	**87.29**	9.36	HMM	3-128
10	DATAf+MAf	65.55	27.26	OC-SVM	2^{-9}
30	MAf+ENEp+MAp+dMAp	86.68	9.07	GMM	32
30	MAf+ENEp+MAp	**87.34**	8.90	HMM	1-128
30	DATAf+dWDEf	68.39	20.86	OC-SVM	2^{3}

6.2 Advanced Computational Intelligence for Audio application

In the recent years, the interest in intelligent environments and in particular in smart-homes has been constantly increasing. The idea is that the living environment should support people in their everyday life by combining the information coming from different sensors and processing it in a intelligent way [128]. The recent reports on population ageing in the most advanced countries are driving governments and the scientific community to focus on technologies

for providing assistance to people in their own homes, making an intelligent environment able to support people, prevent injuries, and detect emergencies. Particular attention has been devoted to solutions based on acoustic signals since they provide a convenient way to monitor people activities and they enable hands-free human-machine interfaces.

In this context, three different arguments are addressed in this dissertation: firstly, a solutions for detecting the falls of persons based on audio sensors is proposed; secondly, a solution for multi-room Voice Activity Detector (VAD) in domestic scenarios is presented; finally, a solution for emergency detection based on audio signals is described.

6.2.1 Human fall detection

The research community devoted particular attention to the solutions for detecting the falls of persons, since they represent the primary cause of injury-related death for the elders [129]. Approaches to the problem are based either on wearable sensors (e.g., accelerometers) or on ambient sensors (e.g., microphones, cameras, floor vibration sensors). Recently, several approaches appeared in the literature that are based exclusively on audio signals [130, 131]. The motivation is that microphones are perceived as less invasive compared to wearable sensors and cameras and they do not suffer from occlusions.

In this dissertation, a fall classification system based on a new type of acoustic sensor that operates similarly to stethoscopes is proposed. In this way, the microphone captures the acoustic waves transmitted through the floor and it mainly captures the sound of falling objects, resulting in a minor sensitivity to the environmental noise. In addition, it is able to capture the subtle signal components transmitted through the floor, which are absent in the signal transmitted through the air.

The fall signals are then processed to recognize from wich kind of fall they are produced: for this purpose, a multiclass classifier is implemented. The algorithm is based on Mel-Frequency Cepstral Coefficients as low-level acoustic features and Gaussian means supervectors as features for a Support Vector Machine classifier.

The acoustic sensor

The acoustic sensor, as described in the related patent[1], is composed by a resonance enclosure and a microphone located inside the enclosure, as depicted in Figure 6.1. The resonance enclosure is characterized by a membrane to guarantee the acoustic coupling with the floor, and a container (i.e., the inner container in Figure 6.1) in which the acoustic resonance phenomenon takes

[1]Patent pending AN2013A000056, 18/03/2013.

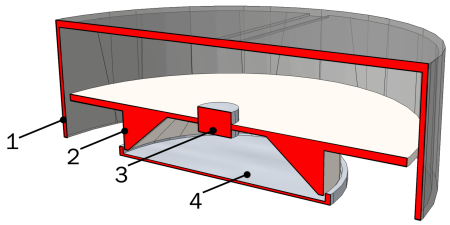


Figure 6.1: The floor acoustic sensor: conceptual scheme. 1 - The outer container. 2 - The inner container. 3 - The microphone slot. 4 - The membrane touching the floor.

place. The inner container can be covered by a layer of acoustic isolation material and it is enclosed by another container (i.e., the outer container in Figure 6.1): this is done to avoid that aerial acoustic waves could be captured by the microphone located inside the acoustic enclosure, allowing it to sense the sole sound waves transmitted through the floor.

The fall classification algorithm

A typical classification algorithm is composed of a first stage of features extraction, in order to best describe the audio event, and a classification stage, where the models of the different classes are stored and are exploited to take the final decision.

Feature extraction stage

The choice of Mel-Frequency Cepstral Coefficients (MFCC) as low-level acoustic features is typical in audio application, like speech and speaker recognition, because represents the perception of human hearing.

Cepstral Mean Normalization (CMN) is usually applied to MFCCs to increase the robustness against channel distortions. CMN consists in subtracting the mean of each cepstral coefficients calculated over the entire event.:

Classification stage

The classification method implemented is similar to the GMM-UBM/SVM paradigm of speaker recognition systems [132, 133], with a Universal Background Model (UBM) representing the entire acoustic space and modelled by means of a Gaussian Mixture Model (GMM). Training of the UBM is performed on a large corpus $\mathcal{U}$ of acoustic events. Considering then a set of N acoustic events corpora $\mathcal{V}_i$ $i = 1, 2, \ldots, N$, for each acoustic event of each corpus a Gaussian Mean Supervector (GMS) is obtained in the *Vector Mapping* block by adapting the UBM with Maximum A-Posteriori (MAP) algorithm [134] and concatenating the mean values. The final step of the training phase is the estimation of the SVM parameters. In this work, an SVM with a radial

basis function has been used.

Classification is performed by extracting the supervector from an input audio signal as in the training phase, and then determining the acoustic event class evaluating the SVM discriminant function. Since the number of classes is greater than two and SVMs are binary classifiers, the "one versus all" [135] technique has been adopted. LIBSVM [136] has been employed both in the training and testing phases of the SVM.

The fall events dataset

In order to evaluate the performance of the acoustic sensor and the classifier, it is necessary to create a set of audio events corresponding to falls of several objects in a variety of conditions.

In this section, the experimental setup and the dataset characteristics are discussed in details.

The dataset composition

For every class considered in this work a wide range of fall events have been recorded.

Considering different kinds of objects falls, the following objects have been considered, as specified in [137]: a volleyball, a metal basket, a book, a metal fork, a plastic chair.

Each object has been dropped at the following distances from the sensor: 1, 2, 4 e 6 meters. The human fall events were performed by 4 male people, who simulated 3 falls from the same distances. Replicating realistic situation is fundamental for the conduction of a valid experimental phase: indeed, it is necessary to drop objects from different angles in order to reproduce fall patterns with significant diversity.

Summarizing, for each combination of object and distance, 12 distinct fall events have been recorded, for a total amount of 288 events.

The experiments

This section presents the experimental results as well as the parameters values of the classification algorithm.

The audio signals of the dataset are downsampled to 16 kHz, and processed in frames 25 ms long and overlapped by 15 ms. In particular, in the MFCC extraction stage the FFT size is 512 and the number of mel filters is 29. Regarding the UBM training, the threshold value in the EM algorithm was set to 0.001. The same value has been employed in the MAP adaptation algorithm, but the maximum number of iterations was set to 5.

The experimental phase is composed of the validation phase, where the performance of the algorithm is analysed while the number of UBM components

	Ball	**Basket**	**Book**	**Chair**	**Fork**	**Falls**		**P**(%)	**R**(%)	$\mathbf{F_1}$(%)
Ball	48	0	0	0	0	0	$\Longrightarrow$	100.00	100.00	100.00
Basket	0	48	0	0	0	0	$\Longrightarrow$	100.00	100.00	100.00
Book	0	0	48	0	0	0	$\Longrightarrow$	94.12	100.00	96.98
Chair	0	0	1	46	1	0	$\Longrightarrow$	100.00	95.83	97.87
Fork	0	0	0	0	48	0	$\Longrightarrow$	100.00	100.00	100.00
Falls	0	0	2	0	0	46	$\Longrightarrow$	97.87	95.83	96.84
							AVG	98.67	98.61	98.61

Table 6.5: Classification performance for the developed sound database: the confusion matrix, presicion, recall and f-measure are reported.

and the SVM hyperparameters vary, and the test phase, where the performance of the algorithm is evaluated on a unseen dataset using the parameters determined in the validation phase.

The technique used for exploiting the totality of the data is the cross-validation "leave-one-distance-out", which consists in dividing the dataset in 4 folds, each containing falls recorded at a certain distance. Then, 3 folds at turn are employed as training and development sets and 1 as test set. The performance is evaluated in terms of *Precision* (P), *Recall* (R) and *F-measure* (F_1).

In the validation phase, the number of components of the UBM has been varied from 1 to 64, while the SVM C and γ parameters have been determined on a grid search using the values $\{2^{-5}, 2^{-3}, \ldots, 2^{15}\}$ and $\{2^{-15}, 2^{-13}, \ldots, 2^{3}\}$ respectively. The best performance was obtained by using an UBM with 16 mixtures, $C = \{2, 2, 2^{-1}, 2\}$ and $\gamma = \{2^{-9}, 2^{-9}, 2^{-7}, 2^{-7}\}$. The values of C and γ are four since they are specific for each test fold. Table 6.5 shows the confusion matrix and the classification performance during the test phase.

It is evident how the classification method is appropriate to discriminate among the different types of fall events. Those results demonstrate the effectiveness of the approach in the acoustic environment contemplated in the dataset.

6.2.2 Multi-room Voice Activity detection

The time boundaries knowledge of the human speech portions contained in an audio signal is fundamental in many audio signal processing applications: Automatic Speech Recognition (ASR), audio encoding systems, speech enhancement algorithms. The automatic detection of human speech time boundaries is known as Voice Activity Detection (VAD) or simply speech detection. In a clean signal, i.e., not corrupted by noise or with very high signal-to-noise ratio (SNR), the VAD task can be easily achieved, however, when the signal is corrupted by noise (e.g., babble noise), the speech detection is extremely

challenging.

However, a special attention and further efforts deserve the application of VADs in domestic and multi-room environments. In particular, this scenario requires the speech-event localisation in space in addition to its detection in time. The state-of-the-art approaches require many processing-stages to obtain the final decision and they are often based on a combination of multiple decisions (e.g., by means of majority voting techniques). The approach proposed in [138] is composed of two separate parts: the speech detection and the speaker localisation. Multi-microphone decision fusion within a Viterbi decoding framework is used in [139]. In particular, the sequence of speech/non-speech events are estimated by the Viterbi algorithm applied on a combined scores calculated by single-channel event GMMs. The performance achieved by the state-of-the-art multi-room VADs are not directly comparable to those presented in this work due to the mismatch existing among the used datasets, also in terms of amount of data.

In this dissertation, the parametrization of the mVAD is carried out by means of multi-stage procedure targeted to the selection of features, network sizes and audio channels. A mVAD is able to simultaneously detect the speech activity in multiple target rooms by using the microphone signals of these rooms as its inputs.

Deep Belief Network Multi-Room VAD

The feature extraction stage transforms input audio signals into real, informative and non-redundant values (i.e., features) which facilitate the subsequent learning task. These features are then fed to the classifier which is based on a Deep Belief Network (DBN). Each classifier output undergoes a thresholding operation and a final hangover scheme leads to the speech activities in each target room.

Feature Extraction

The feature extraction stage extracts six different features from the input signal sampled at 16 kHz: Envelope-Variance Measure, Pitch, Wavelet Coefficient (WC) and Linear Prediction Error (LPE), Mel-Frequency Cepstral Coefficient (MFCC), RelAtive SpecTrAl transform - Perceptual Linear Prediction (RASTA-PLP), Amplitude Modulation Spectrum (AMS). Different frame lengths are used whilst the frame sample rate is common and equal to 100 Hz. The other feature-related parameters are chosen by means of numerous specific evaluations.

A final *min-max* normalisation is performed to restrict the feature value range to [0,1].

DBN-DNN based Classifier

A deep belief network is a probabilistic generative model composed of several layers of stochastic, latent and typically binary variables also referred to as hidden units or feature detectors.

A DBN is obtained by stacking several simpler learning modules: Restricted Boltzmann Machines (RBM). The restricted type of Boltzmann Machine does not allow connections among units of the same layer. RBMs can be straightforwardly trained by means of the Contrastive Divergence (CD-k) algorithm.

RBMs are trained using the CD-1 algorithm layer-by-layer. Firstly, RBM_1 is pre-trained, then the hidden unit activation probabilities of RBM_1 became the visible units of RBM_2 and the pre-training algorithm is applied to RBM_2. This process proceeds iteratively for each layer in the network. Finally, for classification tasks, a further layer can be added on the top and the supervised training algorithm is applied to fine-tune the network weights. The DBN-mVAD uses a top discriminative layer with n units. The network outputs range between [0,1] and, in order to detect the speech activity, they are individually compared to n thresholds.

Hangover Scheme

A simple post-processing of each DBN decisions is needed in order to handle isolated speech detections and to mitigate the early non-speech classification introduced by the DBN outputs thresholding due to slow transitions from speech to non-speech. This technique is commonly named *hangover*. If there are at least two consecutive speech frames, the counter is set to a predefined value equal to 8. This value has been chosen after performing several experiments. On the contrary, for each non-speech frame, the counter is decreased by 1 and the actual non-speech frame is transformed to speech until the counter is not zero.

Other Neural Network based Classifiers

For comparison purpose, two additional types of neural network are used as fully data-driven classifier: the Multi-Layer Perceptron (MLP-mVAD) based and the Bidirectional Long Short-Term Memory Network (BLSTM-mVAD) based one.

Multi-Layer Perceptron

A well-known approach is the steepest descend with error back-propagation. One shortcoming of the MLP, which can significantly affect the training process, is the weights initialisation, which is usually accomplished by following a zero-mean Gaussian distribution. A second relevant drawback occurs when the network became deeper, i.e., many hidden layers. However, both issues, can be mitigated by exploiting a layer-by-layer unsupervised pre-training procedure, peculiar of a Deep Belief Network.

Bidirectional Long Short-Term Memory Network

A BLSTM-RNN is a recurrent neural network in which the usual non-linear neurons (i.e., sigmoid function) are replaced by the long short-term memory blocks. To train such a network, a back-propagation through time (BPTT) algorithm with stochastic gradient descend is used after a weight initialisation. The BLSTM-RNN used here performs a multi-class classification task. Each class corresponds to speech and non-speech in every target rooms, hence, the network has $K = 2^n$ output units. The n mVAD decisions are obtained by opportunely recombining and thresholding the K outputs values.

DIRHA Dataset

The dataset, provided by the DIRHA project [140], contains signals recorded in an apartment equipped with 40 microphones installed on the walls and the ceiling of each rooms. In every-day interactions, the majority of the speech events is expected to occur in the kitchen and living room, thus, these rooms are selected as our *target rooms*. The whole dataset is composed of two subsets called *Real* and *Simulated*.

In this work only the Simulated dataset is used because it contains more data and is characterised by higher noise source rate and a wider variety of background noises. In addition, overlapping events (i.e., speech and noises) may occur with respect to the Real dataset. The Simulated dataset is composed of 80 scenes in Italian language 60 seconds long. Each scene consists of localised acoustic and speech events on which different real background noise, having random dynamics, are superimposed. Events occur randomly in time and in space, constrained on the grid for which Room Impulse Response (RIR) measurements are available. The dataset is built by convolving the signals with the appropriate RIR.

Experiments

The analysis of proposed mVAD is conducted by means of a multi-stage strategy whose steps are the following: first feature selection, network size selection, second feature selection, microphone selection, third feature selection.

The first feature selection stage consist of 63 tests (i.e., all the combination of the feature sets described in Section 6.2.2) in which the network size is fixed to two hidden layers of 10 units each and the input feature set is varied. A second feature selection is explored in order to assess or consolidate the best feature set with the new best topology. Finally, different pairs of microphones are evaluated as network input and a third feature selection is applied on the best pair to finalise the analysis. The 1)-3) stages are ran using two specific microphones: one from the kitchen and one from the living room.

Table 6.6: Comparison of training algorithm parameters. BP stands for "backpropagation" and BPTT indicates the "backpropagation through time".

mVAD Type	**Training algorithm**	**Weight initialisation**	**Epochs**
MLP	BP	Gaussian distr. $\mu = 0, \sigma = 0.1$	Early Stopping
DBN	CD-1 + BP	DBN pre-training	200 + Early Stopping
BLSTM	BPTT	Gaussian distr. $\mu = 0, \sigma = 0.1$	Early Stopping

The experiments are conducted by means of the n-fold cross-validation technique to reduce the performance variance: the dataset is divided in n subsets having approximately the same amount of each of the class occurrences. Then, each of the subset is held out in turn for testing, training on the remaining part. Here, $n = 10$ and a validation set is also employed during the training, thus, 8-1-1 fold(s) respectively compose the training, validation and test sets.

The metrics used to evaluate the VAD performance are Precision (P), Recall (R) and F-measure (F). They are computed by summing two separated confusion matrices, i.e., one for each room. In particular, focusing on a room, a false positive is a frame erroneously detected as speech when the speech signal is absent in that room. For example, a noise or speech frame coming from the other room classified as speech. On the other side, if the frame is correctly classified as non-speech it represents a true negative. A true positive is a frame correctly detected as speech when the speech signal is present in the room, while it represents a false negative if it is wrongly classified as non-speech. Precision and Recall are further exploited to create the P-R curve (PRC) by varying each threshold from 0 to 1. The area under the PRC (AUC) is the metric used to identify the best performance achieved.

The training algorithm depends on the classifier type. Table 6.6 shows a comparison of the training algorithm parameters. In addition, the DBN pre-training is performed for 200 epochs, the momentum is set to 0.9 in BLSTM and MLP based mVAD, whilst it is equal to 0.3 and 0.8 for DBN-mVAD pre-training and fine-tuning, respectively. The learning rate is equal to 10^{-6}, 10^{-7} and $\frac{0.1}{\text{Training Set Length}}$ for BLSTM-, MLP- and DBN-mVAD, respectively. Finally, the early stopping criterion stops the training after 20 consecutive epochs with no error decrement. These values are chosen as a result of intensive experiments. Two GPU-based toolkits have been employed for the experiments: *currennt*[2] for BLSTM-mVAD and a custom version of GPUMLib[3] for DBN-

[2] `http://sourceforge.net/projects/currennt/`
[3] `http://gpumlib.sourceforge.net/`

m/MLP-mVAD.

Results

The results are shown in this section accordingly with the multi-stage strategy adopted.

First Feature Selection

The first analysis consists of varying the input feature set by keeping fixed the neural network size. In particular, two hidden layers of 10 units each are used for the three neural networks. For each combination of n features, where n ranges between 1 and 6, the precision/recall curve is built by varying 40 hangover thresholds.

The best feature set results: PiMfEv (i.e., Pitch, MFCC and EVM_wH). It is composed of 56 features/frame.

Network Size Selection

A further analysis is conducted by varying the neural network size. In particular, the best feature set for each mVAD is kept fix and used as input of more than 250 neural networks having different topology. However, only the topologies which achieved AUC $> 50\%$ are reported for the sake of conciseness. This selection begins evaluating several networks having one hidden layer of different sizes. Following, networks with two and three hidden layers of different sizes are progressively added and tested.

Figure 6.2 shows four graphs containing a series of PRCs related to the three mVADs. PRCs are collected by taking into account the first hidden layer size: the PRCs of networks having 10, 20 and 25, 30, and 40 units are respectively shown in the top-left, top-right, bottom-left and bottom-right plots. Figure 6.2 straightforwardly shows a significant gap among the PRCs of the DBN approach and the other two mVADs. Despite the relevant improvements of the two mVADs with respect to their AUC in Section 6.2.2, the highest performance is attained by the DBN-mVAD.

Second Feature Selection

In the light of the results achieved by the proposed approach with respect to the other approaches, the remaining analysis stages are conducted solely on the DBN-mVAD. In order to obtain more reliable results on the best performing feature set given the new topology (i.e., two hidden layers of 25,15 units) a second feature selection analysis is performed. The result is not reported for the sake of conciseness, however, it confirmed that the feature set selected in the first analysis, i.e., PiMfEv, gives the highest AUC.

Microphone Selection

The fourth stage of the proposed analysis aims to evaluate different microphones as input for the DBN-mVAD. A set of 7 more microphones is considered in addition to the two initially selected. The microphones composing the best

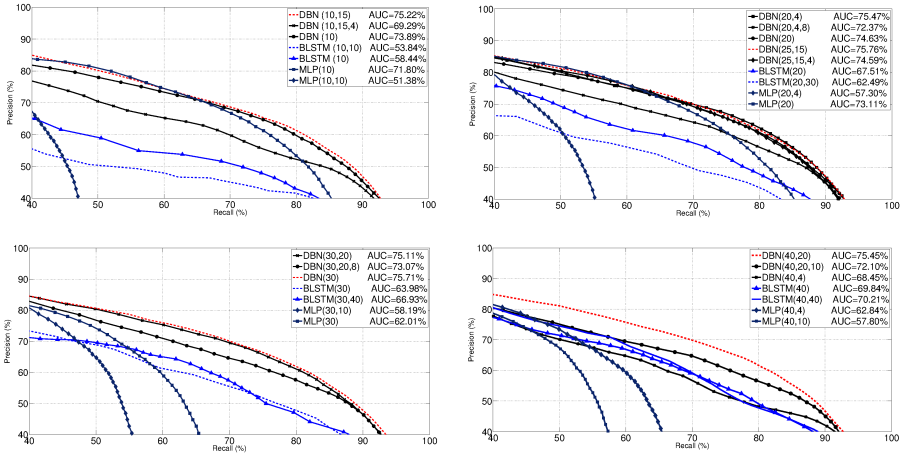


Figure 6.2: Comparison of Precision-Recall Curves (PRCs) of the three mVADs having different sizes of neural classifier. The curves are grouped by the network first layer size, s: the top-left graph shows PRCs of $s = 10$, the top-right PRCs are related to $s = 20$ and $s = 25$, the bottom-left graph contains PRCs of $s = 30$ units and the bottom-right of $s = 40$ units. The red dashed curve is the most performing in each plot.

pair are both installed on the ceiling and, although they are omnidirectional they seems to better capture the speech of their own rooms to the detriment of speech coming from other rooms.

Third Feature Selection

The last stage of analysis consists of a third feature selection using the network topology found in Section 6.2.2 and the pair of microphones resulting from the previous step. The results show that a further significant performance improvement can be achieved using the feature set composed of PiMfRaWcEv (i.e., Pitch, MFCC, RASTA-PLP, WC-LPE and EVM-wH, 176 features/frame). Hence, the DBN-classifier that achieves the best performance has 176 input units, two hidden layers of 25,15 units and one output layer with two units. Given the best PRC, the thresholds for each DBN outputs are chosen by maximising the F-measure. The statistical significance of the F-measure increment, i.e., +3.75%, has been assessed by the one-tailed z-test with the significance level $p < 0.0005$ [141].

6.2.3 Emergency event detection

The focus of this work is on assistive and monitoring applications for emergency detection.

In regard to audio-based systems, as demonstrated by the recent projects

and works [142, 143, 144], they significantly increased their popularity in the recent years. The motivation resides in their versatility, since audio signals allow capturing people's activity, monitoring the acoustic environment and they enable speech-based user interfaces. In addition, people perceive microphones as less invasive sensors respect to video cameras [145], and they do not risk to forget them as with wearable sensors.

The related contributions in this field are focused on the recognition of acoustic events. The system presented in [146, 145] is based on microphones placed on the ceiling and on floor lamps. The authors developed a framework for the detection and localization of acoustic events, comprising both speech (e.g., coughing) and non-speech sounds. Events classes are modelled by means of Gaussian Mixture Models and they are located with the general cross-correlation phase-transform (GCC-PHAT) algorithm. This information is then used for emergency detection. Similarly, the work by [147] addresses the classification of sound events using generative models and multidomain features.

This modality is activated for surveillance purposes, e.g., when the user is outside the house. The system now monitors the acoustic environment to detect events that deviate from normality.

In this dissertation, the novelty detector is based on the approach proposed in [148], which consists in extracting a set of features from the audio signal, and in modelling normal sounds by means of a statistical generative model. In regard to the normality model, Gaussian Mixture Models and Hidden Markov Models have been both considered in order to find the best performing technique for the application scenario. Differently from [148], in the recognition phase, the decision is performed on a chunk-based analysing.

Novelty detection

Novelty detection is the task of classifying *novel* data with respect to the data available during system training [149, 150, 151]. Typically, "abnormal" data are not exploitable to train the model owing to the difficulty or impossibility of collecting them, thus the novelty detection goal lies in the modelling of normality in order to detect abnormalities. Given this scenario, the whole available dataset is solely composed of "normal" data which denote the description of "normality" to be learnt.

The novelty detection phase aims to determine the class affiliation of unknown signals analysing them in *chunks*, i.e., group of frames. This means that the likelihood values of C adjacent frames computed using the trained model λ are combined together. The class affiliation of a chunk results from the comparison of the likelihood value and a predefined threshold ξ: if the likelihood value is below ξ, an unexpected sound event is detected, an alarm is generated, and

Table 6.7: Detection performance summary on the validation set for the GMM model for different sets of features. The best performing chunk size resulted for the day subset is $C = 128$, whilst for the night subset $C = 32$.

GMM	**Day Subset**		**Night Subset**	
	N_g	AUC (%)	N_g	AUC (%)
MFCC	16	79.64	256	99.04
MFCC+TEO+MPEG7	2	80.88	4	96.26
PNCC	128	**90.58**	2	**99.13**
PNCC+TEO+MPEG7	2	81.34	4	96.26

an automatic phone call towards a user defined phone number is established.

Features

The choice of an appropriate set of features is one of the crucial steps to well-describe the "normality" characteristics of input audio signals. Three different types of features are extracted frame-by-frame: Power Normalised Cepstral Coefficients, Critical Band-based Teager Energy Operator (TEO) Autocorrelation Envelope, MPEG-7 Features (Audio Waveform Type Descriptor, Audio Spectrum Flatness Descriptor, Audio Fundamental Frequency Descriptor). In the novelty detector, the 13 PNCCs static coefficients are augmented with their first derivative for a total of 26 coefficients per frame. The number of TEO-based coefficients per frame is 21, while the one of MPEG-7 features is 23. The final feature vector is thus composed of 70 coefficients per frame.

The normality model

The feature vectors extracted from input audio signal are used as input to two well-known statistical modeller in order to create a distribution of "normality" in real-life conditions: Gaussian Mixture Model (GMM) and Hidden Markov Model (HMM).

Gaussian Mixture Model

Chunk-based analysis The chunk analysis of input signal derives from the interpretation of the novelty event characteristics. Indeed, dangerous situations, home intrusions or abnormal events in general can not have very short durations (i.e., hundreds of ms), therefore, isolated detections of novelty or normal frames should be appropriately filtered out.

The chunk signal analysis deals with these frame-based decision shortcomings. In this way, the decision is based on a longer time interval and takes into account the temporal evolution of an event.

Thus, a chunk is formed by collecting a set of adjacent frames, for GMMs, or sequences for HMMs. Chunks are overlapped by 2/3 and their likelihood values are computed as the joint likelihood.

Experiments

The experiments conducted for tuning and evaluating the novelty detection system are hereby presented. The models parameters, i.e., the number of GMM components and the number of HMM states, have been determined on the validation set, a portion of the whole data set. Different combinations of the features presented in Section 6.2.3 have also been evaluated in the validation set. The overall best performing combination has then been evaluated on the test set of the corpus.

A3Novelty corpus

The novelty detection system has been evaluated on the A3Novelty corpus, which consist of more than 56 hours of recording acquired in a university laboratory. These recordings were performed during day and night hours, since their acoustic conditions differ significantly.

The abnormal event sounds are grouped in four categories: *Sirens*, composed of three different siren sounds; *Falls*, composed of two occurrences of a person or an object falling to the ground; *Breakage of objects*, noise produced by the breakage of an object due to the impact with the ground; *Screams*, consist in four different human scream. Both single person or a group of people are considered.

The A3Novelty corpus is composed of two types of recordings: *background* and *background with novelty*. The former contains only background sounds such as human speech, technical tools noise and environmental sounds. The latter, on the other hand, contains the artificially generated novelty events (16 during the day and 30 during the night recordings).

Experiment Setup

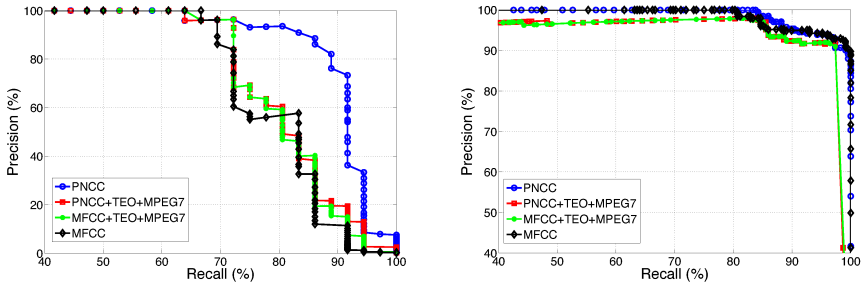
The experiments have been conducted on a excerpt of the A3Novelty corpus. The sampling rate has been reduced to 16 kHz, and processing has been performed in frames 30 ms long overlapped by 20 ms. The hop size is equivalent to 10 ms leading to a frame rate of 100 fps.

The day and night recordings of the A3Novelty corpus have been both divided in three subsets, namely: *training set*, *validation set* and *test set*. Both the day and night training sets are composed of 11 hours of background recordings. Validation sets have a total duration of 4 hours, with the day one containing 7 novelty events and the night one containing 9. The test sets are 4 hours long and contain 5 and 10 novelty occurrences respectively for day and night subset.

The validation sets have been employed for evaluating the performance of different feature sets, the normality model parameters and chunk sizes C. In particular, for each model, four different sets of features have been evaluated: MFCC, MFCC + TEO + MPEG7, PNCC, and PNCC + TEO + MPEG7. In regard to the models, a variety of GMMs each having an increasing number of Gaussians $N_g = \{2, 4, 8, 16, 32, 64, 128, 256, 512\}$ have been considered, while

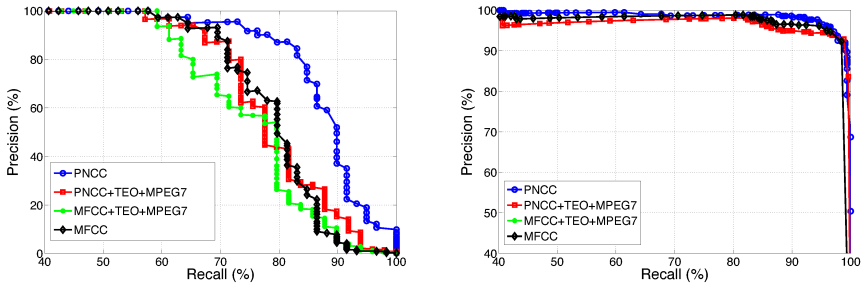
concerning HMMs, the number of hidden states varies from $N_s = 3$ to $N_s = 5$, and the number of Gaussians from $N_g = 2$ to $N_g = 256$.

The performance has been evaluated using Precision (P), Recall (R) and F-measure (F). A true positive or a true negative is respectively represented by a correctly classified novelty or normal chunk. P and R have been used to construct the precision/recall curve (PRC) obtained by varying the threshold ξ. Furthermore, the area under curve (AUC) of each PRC gives the overall system performance.



(a) Day validation subset. (b) Night validation subset.

Figure 6.3: PRC curves of GMMs using different features sets



(a) Day validation subset. (b) Night validation subset.

Figure 6.4: ROC curves of HMMs using different features sets.

Novelty detection results

Table 6.7, Table 6.8, Figure 6.3 and Figure 6.4 show the results obtained in the day and night validation sets. In particular, Table 6.7 reports the AUC obtained by best performing GMM for each set of features. Table 6.8, on the other hand, shows the AUC values obtained by the best performing HMM.

In Figure 6.3 and Figure 6.4, the best performing GMMs and HMMs PRCs are respectively shown. Figure 6.3a and Figure 6.4a are referred to the day subset and they highlight the higher AUC provided by PNCC-based feature

Table 6.8: Detection performance summary on the validation set for the HMM models for different sets of features. The best chunk size resulted for day subset is $C = 16$, whilst for the night subset $C = 4$ except for MFCC feature set where $C = 2$.

HMM	**Day Subset**			**Night Subset**		
	N_s	N_g	AUC (%)	N_s	N_g	AUC (%)
MFCC	4	16	79.03	4	32	98.20
MFCC+TEO+MPEG7	5	32	75.56	4	128	97.87
PNCC	4	32	**87.79**	4	4	**99.02**
PNCC+TEO+MPEG7	5	2	78.97	5	32	97.68

sets with respect to MFCC-based ones. This behaviour demonstrates the effectiveness of PNCC features in improving the robustness of the classification algorithm to background noise. Indeed, the day dataset is characterised by diverse background noises which leads to erroneous detections in MFCC-based models. The PRCs of the night subsets shown in Figure 6.3b and Figure 6.4b are almost overlapped, as it could be expected from the characteristics of night subset. Indeed, the normal background strongly differs from the novelty events leading to high P and R independently from the features set or model employed.

Comparing GMM and HMM models, Table 6.7 and Table 6.8 clearly show that the GMM ones achieve the overall best result using PNCCs only both in day and night dataset. GMM has been preferred due to its lower computation cost with respect to HMM. A further deduction derives from the comparison between the PNCC and the PNCC + TEO + MPEG7 curves: indeed the additional features do not improve the detection performance. In addition, to keep the computational cost as low as possible, the sole PNCCs have been chosen as features.

In the light of these considerations, the novelty detector has been parametrised as follows: the feature set is composed of PNCCs, the normality model for the day subset is represented by a GMM with 128 components, and the chunk size is equal to 128 vectors. The night model differs for the number of gaussians, 2, and the chunk size, which contains 32 vectors.

Chapter 7

Conclusions

In this dissertation, the Machine Learning approaches for Non-Intrusive Load Monitoring have been studied. Within all the technique explored by the scientific community, this work has been focused on the Hidden Markov Model based and the Deep Neural Network based, since their capability and promising performance, at the forefront of the improvements which could be introduced.

For the HMM based approaches, firstly the appliance modelling and all the related aspects have been introduced, therefore the AFAMAP algorithm and its method improvements has been described. Specifically, the variation on the formulation has been detailed for the exploitation of the reactive power. The algorithm has been tested on both denoised and noised scenario, by means of the usage of the Rest-of-the-world model. The last aspect discussed deal with a facilitate procedure for the footprint extraction related to a specific appliance from the aggregated data.

For the DNN based approaches, the dAE has been introduced and the optimization in the model training phase and in architecture has been described. In addition, the recombining technique in the disaggregation phase has been improved. The algorithm has been in tested in both denoised and noised scenario, for which a different optimization procedure has been conducted, as well as in the case of seen and unseen scenario. As last aspect, the exploitation of the reactive power has been considered in the network architecture, providing its own optimization procedure in all the considered scenario.

In addition, advancement in Computation Intelligence application to other field has been conducted: in the area of the smart water and gas grid, the approaches have concerned the consumption forecasting in the short/medium-term and the leakage detection in different grid scenario, whilst in the area of the audio application, the technique have been applied to solve the problems of human fall detection, multi-room voice activity detection and emergency event detection.

In the Chapter 2, an updated review of the State of the Art regarding the NILM algorithms is presented, together with an updated list of available datasets, which are typically used for parameter tuning and evaluation pur-

poses. For what concern the NILM methods addressed in this review, they were first divided into two main categories: load classification and source separation algorithms. This reflect the nature of the method for the disaggregation and the limits or the improvements which could be explored, despite the same problem statement. It is pointed out that most of the contributions make use of the sole active power signal, and only few methods use the reactive power (or the phase difference between the voltage/current phasors). Exploiting this information can be beneficial to obtain a performing disaggregation action, but, on the other hand, requires a specific hardware able to provide the needed measurements. Clearly, a direct comparison between all methods presented is not immediately possible, due to the difference in terms of performance criteria and involved datasets. Indeed, the metrics used in those works could vary, representing different aspects of the obtained results. In terms of performance, the most promising methods appeared to be the HMM models, which are widely used for their capability to represent the appliance consumption behaviour with a relatively easy training procedure. On the other hand, the DNN based method, following the recent success in various Computational Intelligence fields which allow the grow up of the interest in the application for other task, have been recently employed in NILM with promising performance and future improvements.

In the Chapter 3 the Machine Learning approaches, adopted in NILM methods, have been described.

The AFAMAP algorithm [22], resulting one of the most performing and computationally efficient among the HMM based approaches, has been described in the Chapter 4. The appliance models based on HMM have been introduced and the procedure for estimating their parameters has been described. This consists in the extraction of the footprint of the appliance by means of an Appliance Activity Detector and in the estimation of the power levels of each working state by clustering the appliance footprint with the k-means algorithm. The same procedure in used to compose the Rest-of-the-World model for the testing the FHMM algorithms in the noised scenario. AFAMAP is revised in order to improve its performance through a more exhaustive exploitation of the information pertaining the appliance activity. The proposed algorithm exploits both additive and differential FHMM to model the activity of the appliances. At each time step, the best combination of appliances working state is chosen to represent the actual aggregated consumption: as result of the optimisation process, a set of coefficients are returned to weight the appliances working state and compose the own disaggregated consumption. The revised algorithm, however, takes into account additional elements. In regards to the FHMM model, a forward differential model is paired to the reference backward differential FHMM, thus not only the transition from the previous state to the

current one is included, but also the transition from the current state to the next one. In addition, the use of solver boundaries is explored: firstly, the setting has been related to the admissible state combination of the aggregated power; alternatively, the reactive power disaggregation output has been used to select the boundaries, endorsing the heterogeneous data usage effectiveness. Later, active and reactive power have been introduced in Additive Factorial HMM for non-intrusive load monitoring. The disaggregation algorithm is in an alternative formulation of the AFAMAP developed in order to deal with the bivariate formulation of the problem. As results, the algorithm is able to output the disaggregated profiles in the active and reactive power components. The proposed approach has been compared to the univariate formulation of AFAMAP and to the algorithm presented by Hart in [16]. The latter is based on Finite State Machine appliance models and it employs both the active and reactive power. The algorithm has been improved for handling the occurrence of multiple solutions by means of a MAP technique. The experiments have been conducted on the AMPds [58] dataset, which provides the ground truth appliance consumption both in the active and reactive power components. The results showed that, in a denoised scenario, the proposed approach outperforms both the comparative methods, with an absolute F_1-Measure improvement of $+14.9\,\%$ and $+2.5\,\%$ in the 6 appliances case study. As last aspect, a footprint extraction procedure has been introduced as a solution for the appliance modelling in real NILM scenarios. Indeed, in order to create the appliance model and to use this in the disaggregation algorithm, the user needs to record the appliance consumption profile. A facilitated procedure is needed, in order to obtain a clean footprint from the aggregated power signal in real scenario: therefore, a user-aided footprint extraction procedure is defined. The solution introduced here relies on the availability of a general model for the appliance category to obtain the clean footprint. This is the starting point of the modelling stage: in this work the AFAMAP algorithm has been used. The resulting models have been tested in a disaggregation problem, and they have been compared with the same problem solved using the true appliance model, i.e., the models created using the actual footprint from the appliance level consumption. The results have showed a moderate performance reduction compared to the ideal case due to the footprint extraction stage. For those reasons, the footprint extraction procedure introduced in this work can be considered as an effective method for the user employment in a real NILM scenario.

In the Chapter 5, a DNN architecture based on the denoising autoencoder topology has been proposed. Compared to the work by Kelly and Knottenbelt [32] several improvements have been introduced. In the training phase, the variable step size has been adopted, with an early stopping criterion based on the performance metric calculated on the validation set. In the disaggregation

phase, the median filter has been applied to combine the overlapped portion of signal in the sliding window analysis of the aggregated power data. In order to achieve the best performance, for each network an optimisation of the network parameters has been conducted, starting from the reference architecture and introducing a second layer of CNN and a pooling stage to compress the size of the output. The proposed approach has been compared to the AFAMAP [22] algorithm. This algorithm has been adopted for the *noised* scenario with the introduction of RoW model. The experiments have been conducted on the AMPds [58], on the UK-DALE [61] and on the REDD [30] datasets, evaluating both the *denoised* and *noised* scenario. Furthermore, the availability of recordings from more than one building in the UK-DALE and in the REDD datasets allowed to evaluate the algorithms on an *unseen* scenario. The results showed that the proposed approach outperforms the comparative methods in the overall average between the appliance, both in *denoised* and *noised* scenario. Regarding the *unseen* scenario, the performance demonstrated that the generalisation property of the dAE allowed acceptable degradation of performance, respect to the AFAMAP algorithm, in which the footprint extraction stage introduced errors in the HMM modelling phase. Moreover, the dAE has been tested adding the reactive power consumption at the input of the network, increasing the information level of the algorithm. The experiments have been conducted on the AMPds [58] and on the UK-DALE [61], with the same configuration of the previous application. The results showed that the exploitation of the reactive power gives a performance improvement to the method, particularly for some appliance, whereas for other the performance decrease. The overall results showed that this method reached better performance in *seen* scenario, whilst it degraded in *unseen* scenario.

In the Chapter 6 advanced techniques has been applied to other research fields.

In the first part of this chapter, the interest is focused on the smart water and gas grid.

As first aspect, the study has been focused on short-term forecasting of water and natural gas consumption. The predictions have been performed using Artificial Neural Network, Echo State Network, Deep Belief Network, Support Vector Regression, Extreme Learning Machine, and Genetic Programming, on the AMPds and DFID datasets. Concerning the natural gas, the best result has been achieved by the SVR approach for $12\,h$ resolution. Moreover, the ELM and ANN have also confirmed to be suitable for shot-term prediction, whilst the water prediction shows that the overall best result has been achieved with the 2 years data by ANN at $12\,h$ resolution. The ANN has achieved the best results for all the DFID sub-sets, and ELM and SVR have performed close results.

As second aspect, temporal features and pressure information have been exploited to improve the performance of the automatic leakage detection approach, in smart water grid. The experiments have been performed for different data resolutions, 1, 10, and 30 minutes, extrapolated from the flow data available in the AMPds dataset and simulating realistic leakages using the EPANET tool. GMM, HMM, and OC-SVM have been adopted, under a comparative perspective, to model the normality background by assuming different parameters. For both GMM and HMM the overall best results are achieved by addressing both flow and pressure, for all time resolutions, and the GMM performance are close to the HMM ones. The introduction of pressure information allows to adopt an approach with lower computational burden, and to detect leakages even for low resolution (30 minutes), as well.

In the second part of the chapter, different Audio application has been considered.

As first aspect, an innovative (patent-pending) acoustic sensor for detection of sounds transmitted though the floor has been proposed. A fall classification algorithm for the discrimination between sounds produced by falls of distinct objects has been developed. The algorithm is a typical data-driven classifier, where MFCCs are used as features, Supervectors as feature descriptors and SVM as expert system. The algorithm was test on a dataset which includes six different fall events, and it allowed to achieve an overall F-measure (averaged among all events) superior than 98%, thus confirming the effectiveness of the approach.

As second aspect, a data-driven voice activity detection approach based on a deep belief network and applied on a multi-room domestic environment (DBN-mVAD) has been present. Regarding the experiments, an optimization procedure consisting of five stages is performed and results are compared to two mVADs based on a multi-layer perceptron and on a bidirectional long short-term memory neural network. Compared to the first stage of the DBN-mVAD, an absolute AUC increment of 10.41% has been achieved whilst, in terms of F-measure, the increment is equal to 8.56%.

As last aspect, a novelty detection algorithm has been presented, as a system able to discriminate between normal an abnormal acoustic events. The evaluation has been conducted on the A3Novelty signal corpora, consisting of several hours of recordings.Different combinations of features and normality model parameters have been evaluated, and the overall best solution in terms of performance/computational cost ratio resulted in the PNCC features together with a GMM statistical model. The experiments conducted on test sets resulted in an overall F-measure equal to 87.32% and 95.53% respectively for the day and night set, thus demonstrating the effectiveness of the approach.

7.1 Future Research Topics

Since the high interest regarding the consumption reduction and the recent improvement in the smart grid researches, the interest on improving those method will be certainly maintained, pointing out as good results as a distributed network of smart plug. For this reason, future works will be oriented on different aspect, related to each algorithm discussed above.

Regarding the AFAMAP algorithm, a more reliable appliance model will be considered in order to improve the representation of the working states, e.g., the usage of Gaussian Mixture Model (GMM) within the HMM allows the representation of a more suitable power level density distribution with respect to a simple Gaussian distribution. Furthermore, additional information about the working states duration will be introduced, allowing the discrimination of HMMs with similar transition probabilities but different time in the switching activity. This translates into a fully exploitation of the differential model. Additionally, an observation window of longer duration could be introduced in the differential model.

Regarding the appliance modelling stage, an unsupervised clustering technique will be introduced to automatically detect the number of power levels, e.g., regarding appliances which not belongs to the categories considered. Regarding the disaggregation and solver algorithms, binary variables will be introduced in the problem formalisation, leading to a Mixed Integer Quadratic Program (MIQP), in order to impose the variable to assume binary results and not integer values as in fuzzy logic, which can lead to ambiguous evaluation in the HMM state evolution. Finally, further experiments will be conducted in order to compare the proposed solution to other approaches recently presented in the literature [27, 28, 36].

Regarding the user-aided footprint extraction procedure, the separation of the model representing the fridge-freezer combination in the single component will be evaluated, since the AFAMAP algorithm shows a better working in the problem resolution using models with lower number of states. Moreover, more experiments will be performed using different datasets in literature, in which a more detailed study about the generalization performance can be carried out, specially for the generic model selection.

For the dAE approach, the introduction of a constraint between the neural model output will be considered, in order to assume the equality between the aggregated data and the sum of the profiles reconstructed, in the *denoised* scenario. In order to apply this constraint in the *noised* scenario, the introduction of the neural based RoW model will be required.

Regarding the exploitation of the reactive power in the dAE algorithm, future works will be focused on the investigation regarding the appliance which

degrades performance. Moreover, the reactive power on target will be inserted, in order to allow a fully balanced input-output architecture.

Improvements raise high interest in other application fields, too.

Regarding the smart water and gas grid, due to the lack of data, more data will be collected in order to create a suitable dataset of water and natural gas for heterogeneity evaluations, whereas, in the leakage detection application, investigation of neural network approaches for automatic novelty detection, such as autoencoder, will be performed and further evaluations will be performed by applying temporal clustering.

Regarding the audio application, the floor acoustic sensor will be optimized by means of a focused acoustic design, in order to make it suitably working in different operating contexts. From the fall classification/detection perspective, more experiments will be carried out to evaluate the validity of the proposed solution in presence of noisy environments and diverse floors. Moreover, regarding the multi-room VAD, the exploitation of a larger dataset specially for enhancing the pre-training will be carried out. Furthermore, the introduction of signal-processing techniques, such as beamforming, to reduce the signal intensity of the speech coming from non-target rooms, and the employment of more than two microphones as network inputs, will be considered. Finally, in the emergency event detection application, novelty detection can be improved introducing activity detection techniques at the likelihood level.

List of Publications

[1] R. Bonfigli, A. Felicetti, E. Principi, M. Fagiani, S. Squartini, and F. Piazza, "Denoising autoencoders for non-intrusive load monitoring: Improvements and comparative evaluation," *Energy and Buildings*, to appear.

[2] R. Bonfigli, E. Principi, M. Fagiani, M. Severini, S. Squartini, and F. Piazza, "Non-intrusive load monitoring by using active and reactive power in additive factorial hidden markov models," *Applied Energy*, vol. 208, no. Supplement C, pp. 1590 – 1607, 2017.

[3] patent, "Metodo per il monitoraggio non intrusivo del consumo di apparecchiature elettriche collegate ad una linea di alimentazione comune," Domanda numero: 102017000004554, patent pending.

[4] R. Bonfigli, E. Principi, S. Squartini, M. Fagiani, M. Severini, and F. Piazza, "User-aided Footprint Extraction for Appliance Modelling in Non-Intrusive Load Monitoring," in *Proc. of the IEEE Symposium Series on Computational Intelligence*, Athens, Greece, Dec. 6-9 2016, pp. 1–8.

[5] R. Bonfigli, M. Severini, S. Squartini, M. Fagiani, and F. Piazza, "Improving the performance of the AFAMAP algorithm for non-intrusive load monitoring," in *Proc. of the IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, Canada, 2016, pp. 303–310.

[6] M. Fagiani, S. Squartini, R. Bonfigli, M. Severini, and F. Piazza, "Exploiting temporal features and pressure data for automatic leakage detection in smart water grids," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, July 2016, pp. 295–302.

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

[8] M. Fagiani, S. Squartini, R. Bonfigli, and F. Piazza, "Short-term load forecasting for smart water and gas grids: A comparative evaluation," in *Environment and Electrical Engineering (EEEIC), 2015 IEEE 15th International Conference on*, June 2015, pp. 1198–1203.

[9] G. Ferroni, R. Bonfigli, E. Principi, S. Squartini, and F. Piazza, "A deep neural network approach for voice activity detection in multi-room domestic scenarios," in *2015 International Joint Conference on Neural Networks (IJCNN)*, July 2015, pp. 1–8.

[10] E. Principi, P. Olivetti, S. Squartini, R. Bonfigli, and F. Piazza, "A floor acoustic sensor for fall classification," in *Audio Engineering Society Convention 138*, May 2015.

[11] E. Principi, S. Squartini, R. Bonfigli, G. Ferroni, and F. Piazza, "An integrated system for voice command recognition and emergency detection based on audio signals," *Expert Systems with Applications*, vol. 42, no. 13, pp. 5668 – 5683, 2015.

[12] G. Ferroni, R. Bonfigli, E. Principi, S. Squartini, and F. Piazza, "Neural networks based methods for voice activity detection in a multi-room domestic environment," in *XIII AI*IA Symposium on Artificial Intelligence*, Dec 2014.

Bibliography

[1] Nipun Batra, Jack Kelly, Oliver Parson, Haimonti Dutta, William Knottenbelt, Alex Rogers, Amarjeet Singh, and Mani Srivastava, "NILMTK: An open source toolkit for non-intrusive load monitoring," in *Proc. of the 5th international conference on Future energy systems*. ACM, 2014, pp. 265–276.

[2] David Archer, *Global warming: understanding the forecast*, John Wiley & Sons, second edition, 2012.

[3] C. Rosenzweig, D. Karoly, M. Vicarelli, P. Neofotis, Q. Wu, G. Casassa, A. Menzel, T.L. Root, N. Estrella, B. Seguin, P. Tryjanowski, C. Liu, S. Rawlins, and A. Imeson, "Attributing physical and biological impacts to anthropogenic climate change," *Nature*, vol. 453, no. 7193, pp. 353–357, 2008.

[4] Naomi Oreskes, "The scientific consensus on climate change," *Science*, vol. 306, no. 5702, pp. 1686–1686, 2004.

[5] Hassan Farhangi, "The path of the smart grid," *IEEE Power Energy Mag.*, vol. 8, no. 1, pp. 18–28, 2010.

[6] K. Carrie Armel, Abhay Gupta, Gireesh Shrimali, and Adrian Albert, "Is disaggregation the holy grail of energy efficiency? The case of electricity," *Energy Policy*, vol. 52, pp. 213–234, 2013.

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

[8] S. Darby, "The effectiveness of feedback on energy consumption," Tech. Rep., University of Oxford, Oxford, UK, 2006.

[9] Karen Ehrhardt-Martinez, Kat A. Donnelly, and John A. Laitner, "Advanced metering initiatives and residential feedback programs: a meta-review for household electricity-saving opportunities," Tech. Rep. E105, American Council for an Energy-Efficient Economy Washington, DC, 2010.

[10] J. Laitner, K. Ehrhardt-Martinez, and V. McKinney, "Examining the scale of the behaviour energy efficiency continuum," in *American Council for an Energy Efficient Economy, European Council for an Energy Efficient Economy Conference*, Cote d'Azur, France, 2009, paper ID 1367.

[11] G.T. Gardner and P.C. Stern, "The short-list: the most effective actions us households can take to curb climate change," *Environment: Science and Policy for a Sustainable Environment*, vol. 50, no. 5, pp. 12–24, 2008.

[12] Simin Ahmadi-Karvigh, Burcin Becerik-Gerber, and Lucio Soibelman, "A framework for allocating personalized appliance-level disaggregated electricity consumption to daily activities," *Energy and Buildings*, vol. 111, pp. 337–350, 2016.

[13] I. Abubakar, S.N. Khalid, M.W. Mustafa, Hussain Shareef, and M. Mustapha, "Application of load monitoring in appliances' energy management – a review," *Renewable and Sustainable Energy Reviews*, vol. 67, pp. 235–245, 2017.

[14] Ahmed Zoha, Alexander Gluhak, Muhammad Ali Imran, and Sutharshan Rajasegarar, "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey," *Sensors*, vol. 12, no. 12, pp. 16838–16866, 2012.

[15] Z. Wang and G. Zheng, "Residential appliances identification and monitoring by a nonintrusive method," *IEEE Transactions on Smart Grid*, vol. 3, no. 1, pp. 80–92, March 2012.

[16] George William Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[17] Michael Zeifman and Kurt Roth, "Nonintrusive appliance load monitoring: Review and outlook," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 76–84, February 2011.

[18] Hyungsul Kim, Manish Marwah, Martin F. Arlitt, Geoff Lyon, and Jiawei Han, "Unsupervised disaggregation of low frequency power measurements," in *Proc. 11th SIAM Int. Conf. Data Mining*, Mesa, AZ, USA, 2011, pp. 747–758.

[19] Oliver Parson, Siddhartha Ghosh, Mark Weal, and Alex Rogers, "An unsupervised training method for non-intrusive appliance load monitoring," *Artificial Intelligence*, vol. 217, pp. 1–19, 2014.

[20] Oliver Parson, Mark Weal, and Alex Rogers, "A scalable non-intrusive load monitoring system for fridge-freezer energy efficiency estimation," in *Proc. of the 2nd Int. Workshop on Non-Intrusive Load Monitoring*, Austin, TX, USA, Jun. 3 2014.

[21] A. Zoha, A. Gluhak, M. Nati, and M.A. Imran, "Low-power appliance monitoring using Factorial Hidden Markov Models," in *Proc. of the 8th Int. Conf. on Intelligent Sensors, Sensor Networks and Information Processing: Sensing the Future (ISSNIP)*, Melbourne, VIC, Australia, 2013, vol. 1, pp. 527–532.

[22] J.Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial HMMs with application to energy disaggregation," *Journal of Machine Learning Research*, vol. 22, pp. 1472–1482, 2012.

[23] I. Valera, F. J. R. Ruiz, and F. Perez-Cruz, "Infinite factorial unbounded-state hidden markov model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1816–1828, Sept 2016.

[24] Yeqing Li, Zhongxing Peng, Junzhou Huang, Zhilin Zhang, and Jae Hyun Son, "Energy Disaggregation via Hierarchical Factorial HMM," in *Proc. of the 2nd Int. Workshop on Non-Intrusive Load Monitoring*, Austin, TX, USA, 2014.

[25] Mingjun Zhong, Nigel Goddard, and Charles Sutton, "Signal aggregate constraints in additive factorial HMMs with application to energy disaggregation," in *Advances in Neural Information Processing Systems*, 2014, pp. 3590–3598.

[26] Mingjun Zhong, Nigel Goddard, and Charles Sutton, "Interleaved factorial non-homogeneous hidden markov models for energy disaggregation," in *Proc. of Advances in Neural Information Processing System, Workshop on Machine Learning for Sustainability*, Lake Tahoe, Nevada, USA, 2014, pp. 1–5.

[27] A. Cominola, M. Giuliani, D. Piga, A. Castelletti, and A.E. Rizzoli, "A hybrid signature-based iterative disaggregation algorithm for non-intrusive load monitoring," *Applied Energy*, vol. 185 Part 1, pp. 331–344, 2017.

[28] Stephen Makonin, Fred Popowich, Ivan V. Bajić, Bob Gill, and Lyn Bartram, "Exploiting HMM Sparsity to Perform Online Real-Time Non-intrusive Load Monitoring," *IEEE Transactions on Smart Grid*, vol. 7, no. 6, pp. 2575–2584, 2016.

[29] Matthew J. Johnson and Alan S. Willsky, "Bayesian Nonparametric Hidden semi-Markov Models," *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 673–701, Feb. 2013.

[30] J Zico Kolter and Matthew J Johnson, "REDD: A public data set for energy disaggregation research," in *Proc. of the SustKDD Workshop on Data Mining Appliations in Sustainbility*, San Diego, CA, USA, 2011, pp. 1–6.

[31] Felan Carlo C. Garcia, Christine May C. Creayla, and Erees Queen B. Macabebe, "Development of an intelligent system for smart home energy disaggregation using stacked denoising autoencoders," in *Proc. of the Int. Symp. on Robotics and Intelligent Sensors (IRIS)*, Tokyo, Japan, Dec. 17-20 2016, pp. 248–255.

[32] Jack Kelly and William 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*, New York, NY, USA, 2015, BuildSys '15, pp. 55–64, ACM.

[33] L. Mauch and B. Yang, "A new approach for supervised power disaggregation by using a deep recurrent LSTM network," in *Proc. of GlobalSIP*, Orlando, FL, USA, 2015, pp. 63–67.

[34] L. Mauch and B. Yang, "A novel DNN-HMM-based approach for extracting single loads from aggregate power signals," in *Proc. of ICASSP*, Shanghai, China, 2016, pp. 2384–2388.

[35] Men-Shen Tsai and Yu-Hsiu Lin, "Modern development of an adaptive non-intrusive appliance load monitoring system in electricity energy conservation," *Applied Energy*, vol. 96, pp. 55–73, 2012.

[36] Bochao Zhao, Lina Stankovic, and Vladimir Stankovic, "On a Training-Less Solution for Non-Intrusive Appliance Load Monitoring Using Graph Signal Processing," *IEEE Access*, vol. 4, pp. 1784–1799, 2016.

[37] Marisa Figueiredo, Ana De Almeida, and Bernardete Ribeiro, "Home electrical signal disaggregation for non-intrusive load monitoring (NILM) systems," *Neurocomputing*, vol. 96, pp. 66–73, Nov. 2012.

[38] J. M. Gillis, S. M. Alshareef, and W. G. Morsi, "Nonintrusive load monitoring using wavelet design and machine learning," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 320–328, Jan 2016.

[39] Steven R Shaw, Steven B Leeb, Leslie K Norford, and Robert W Cox, "Nonintrusive load monitoring and diagnostics in power systems," *Instrumentation and Measurement, IEEE Transactions on*, vol. 57, no. 7, pp. 1445–1454, 2008.

[40] Jon Froehlich, Eric Larson, Sidhant Gupta, Gabe Cohn, Matthew Reynolds, and Shwetak Patel, "Disaggregated end-use energy sensing for the smart grid," *IEEE Pervasive Computing*, vol. 10, no. 1, pp. 28–39, 2011.

[41] Hesham K Alfares and Mohammad Nazeeruddin, "Electric load forecasting: literature survey and classification of methods," *International Journal of Systems Science*, vol. 33, no. 1, pp. 23–34, 2002.

[42] B Neenan, J Robinson, and RN Boisvert, "Residential electricity use feedback: A research synthesis and economic framework," *Electric Power Research Institute*, 2009.

[43] Marco Severini, Stefano Squartini, and Francesco Piazza, "Hybrid soft computing algorithmic framework for smart home energy management," *Soft Computing*, vol. 17, no. 11, pp. 1983–2005, 2013.

[44] M. Severini, S. Squartini, and F. Piazza, "Computational framework based on task and resource scheduling for micro grid design," in *Neural Networks (IJCNN), 2014 International Joint Conference on*, July 2014, pp. 1695–1702.

[45] Marisa B. Figueiredo, Bernardete Ribeiro, and Ana de Almeida, "Electrical signal source separation via nonnegative tensor factorization using on site measurements in a smart home.," *IEEE T. Instrumentation and Measurement*, pp. 364–373, 2014.

[46] Leen De Baets, Joeri Ruyssinck, Chris Develder, Tom Dhaene, and Dirk Deschrijver, "On the Bayesian optimization and robustness of event detection methods in NILM," *Energy and Buildings*, vol. 145, pp. 57–66, 2017.

[47] Leslie K. Norford and Steven B. Leeb, "Non-intrusive electrical load monitoring in commercial buildings based on steady-state and transient load-detection algorithms," *Energy and Buildings*, vol. 24, no. 1, pp. 51 – 64, 1996.

[48] Mario Berges, Ethan Goldman, H Scott Matthews, Lucio Soibelman, and Kyle Anderson, "User-centered nonintrusive electricity load monitoring for residential buildings," *Journal of Computing in Civil Engineering*, vol. 25, no. 6, pp. 471–480, 2011.

[49] Hsueh-Hsien Chang and Hong-Tzer Yang, "Applying a non-intrusive energy-management system to economic dispatch for a cogeneration system and power utility.," *Applied Energy*, vol. 86, no. 11, pp. 2335 – 2343, 2009.

[50] Hsueh-Hsien Chang, "Non-intrusive demand monitoring and load identification for energy management systems based on transient feature analyses," *Energies*, vol. 5, no. 11, pp. 4569–4589, 2012.

[51] Hsueh-Hsien Chang, Putu Wegadiputra Wiratha, and Nanming Chen, "A non-intrusive load monitoring system using an embedded system for applications to unbalanced residential distribution systems," *Energy Procedia*, vol. 61, pp. 146–150, 2014.

[52] Misbah Aiad and Peng Hin Lee, "Unsupervised approach for load disaggregation with devices interactions," *Energy and Buildings*, vol. 116, pp. 96–103, 2016.

[53] Yung Fei Wong, Y. Ahmet Sekercioglu, T. Drummond, and Voon Siong Wong, "Recent approaches to non-intrusive load monitoring techniques in residential settings," in *Proc. of the IEEE Symp. on Computational Intelligence Applications In Smart Grid (CIASG)*, Singapore, Singapore, 16-19 Apr. 2013, pp. 73–79.

[54] J. M. Alcalá, J. Ureña, Á. Hernández, and D. Gualda, "Sustainable homecare monitoring system by sensing electricity data," *IEEE Sensors Journal*, vol. 17, no. 23, pp. 7741–7749, Dec 2017.

[55] Seyed Mostafa Tabatabaei, Scott Dick, and Wilsun Xu, "Toward non-intrusive load monitoring via multi-label classification," *IEEE Transactions on Smart Grid*, vol. 8, no. 1, pp. 26–40, 2017.

[56] Robert S Butner, Douglas J Reid, M Hoffman, Gregory P Sullivan, and Jeremy Blanchard, *Non-Intrusive Load Monitoring Assessment: Literature Review and Laboratory Protocol*, Pacific Northwest National Laboratory, 2013.

[57] Christian Beckel, Wilhelm Kleiminger, Romano Cicchetti, Thorsten Staake, and Silvia Santini, "The ECO data set and the performance of non-intrusive load monitoring algorithms," in *Proceedings of the 1st ACM International Conference on Embedded Systems for Energy-Efficient Buildings (BuildSys 2014). Memphis, TN, USA*. November 2014, pp. 80–89, ACM.

[58] Stephen Makonin, Fred Popowich, Lyn Bartram, Bob Gill, and Ivan V. Bajic, "AMPds: A public dataset for load disaggregation and eco-feedback research," in *Proc. of the IEEE Electrical Power and Energy Conference (EPEC)*, Halifax, NS, Canada, 2013.

[59] F. Paradiso, F. Paganelli, D. Giuli, and S. Capobianco, "Context-based energy disaggregation in smart homes," *Future Internet*, vol. 8, no. 1, 2016.

[60] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[61] Jack Kelly and William Knottenbelt, "The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes," *Scientific data*, vol. 2, 2015.

[62] Huijuan Shao, Manish Marwah, and Naren Ramakrishnan, "A temporal motif mining approach to unsupervised energy disaggregation: Applications to residential and commercial buildings," in *Proc. of the Twenty-Seventh AAAI Conference on Artificial Intelligence*. 2013, pp. 1327–1333, AAAI Press.

[63] Kyle Anderson, Adrian Ocneanu, Diego Benitez, Derrick Carlson, Anthony Rowe, and Mario 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, August 2012.

[64] Sean Barker, Aditya Mishra, David Irwin, Emmanuel Cecchet, Prashant Shenoy, and Jeannie Albrecht, "Smart*: An open data set and tools for enabling research in sustainable homes," *SustKDD, August*, 2012.

[65] A. Reinhardt, P. Baumann, D. Burgstahler, M. Hollick, H. Chonov, M. Werner, and R. Steinmetz, "On the accuracy of appliance identification based on distributed load metering data.," in *Proceedings of the 2nd IFIP Conference on Sustainable Internet and ICT for Sustainability (SustainIT)*, October 2012.

[66] C Holcomb, "Pecan street inc.: A test-bed for NILM," in *International Workshop on Non-Intrusive Load Monitoring, Pittsburgh, PA, USA*, 2012.

[67] Jean-Paul Zimmermann, Matt Evans, Jonathan Griggs, Nicola King, Les Harding, Penelope Roberts, and Chris Evans, "Household electricity survey: A study of domestic electrical product usage," *Intertek Testing & Certification Ltd*, 2012.

[68] Nipun Batra, Manoj Gulati, Amarjeet Singh, and Mani 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*. ACM, 2013, pp. 1–8.

[69] Andrea Monacchi, Dominik Egarter, Wilfried Elmenreich, Salvatore D'Alessandro, and Andrea M. Tonello, "GREEND: an energy consumption dataset of households in italy and austria," *CoRR*, vol. abs/1405.3100, 2014.

[70] Nipun Batra, Oliver Parson, Mario Berges, Amarjeet Singh, and Alex Rogers, "A comparison of non-intrusive load monitoring methods for commercial and residential buildings," *CoRR*, vol. abs/1408.6595, 2014.

[71] Mehdi Maasoumy, B Sanandaji, Kameshwar Poolla, and Alberto Sangiovanni Vincentelli, "BERDS - berkeley energy disaggregation data set," in *Proc. of the Workshop on Big Learning at the Conference on Neural Information Processing Systems (NIPS 2013)*, 2014.

[72] Lucas Pereira, Filipe Quintal, Rodolfo Gonçalves, and Nuno Jardim Nunes, "SustData: A public dataset for ICT4S electric energy research," in *ICT for Sustainability 2014 (ICT4S-14)*. Atlantis Press, 2014.

[73] Jack Kelly, Nipun Batra, Oliver Parson, Haimonti Dutta, William Knottenbelt, Alex Rogers, Amarjeet Singh, and Mani Srivastava, "NILMTK v0.2: A non-intrusive load monitoring toolkit for large scale data sets: Demo abstract," in *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, New York, NY, USA, 2014, BuildSys '14, pp. 182–183, ACM.

[74] Brian W Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochimica et Biophysica Acta (BBA)-Protein Structure*, vol. 405, no. 2, pp. 442–451, 1975.

[75] Claudia Beleites, Reiner Salzer, and Valter Sergo, "Validation of soft classification models using partial class memberships: An extended concept of sensitivity and co. applied to grading of astrocytoma tissues," *Chemometrics and Intelligent Laboratory Systems*, vol. 122, pp. 12 – 22, 2013.

[76] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb 1989.

[77] Z. Ghahramani and M.I. Jordan, "Factorial Hidden Markov Models," *Machine Learning*, vol. 29, no. 2-3, pp. 245–273, 1997.

[78] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.

[79] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[80] Ian Goodfellow, Honglak Lee, Quoc V. Le, Andrew Saxe, and Andrew Y. Ng, "Measuring invariances in deep networks," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J.D. Lafferty, C.K.I. Williams, and A. Culotta, Eds., pp. 646–654. Curran Associates, Inc., 2009.

[81] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle, "Greedy layer-wise training of deep networks," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J.C. Platt, and T. Hoffman, Eds., pp. 153–160. MIT Press, 2007.

[82] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, Dec. 2010.

[83] Sundeep Pattem, "Unsupervised disaggregation for non-intrusive load monitoring," in *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*. IEEE, 2012, vol. 2, pp. 515–520.

[84] Oliver Parson, Siddharta Ghosh, Mark Weal, and Alex Rogers, "Non-intrusive load monitoring using prior models of general appliance types," in *Twenty-Sixth Conference on Artificial Intelligence (AAAI-12)*, 2012.

[85] Stephen Makonin and Fred Popowich, "Nonintrusive load monitoring (NILM) performance evaluation," *Energy Efficiency*, vol. 8, no. 4, pp. 809–814, 2014.

[86] John A Hartigan and Manchek A Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[87] Dominik Egarter, Manfred Pöchacker, and Wilfried Elmenreich, "Complexity of power draws for load disaggregation," *CoRR*, 2015.

[88] A.I. Cole and A. Albicki, "Algorithm for nonintrusive identification of residential appliances," in *Proc. of the IEEE Int. Symp. on Circuits and*

Systems (ISCAS), Monterey, CA, USA, 31 May - 3 Jun. 1998, pp. 338–341.

[89] C. Laughman, Kwangduk Lee, R. Cox, S. Shaw, S. Leeb, L. Norford, and P. Armstrong, "Power signature analysis," *Power and Energy Magazine, IEEE*, vol. 1, no. 2, pp. 56–63, Mar 2003.

[90] Alan Marchiori, Douglas Hakkarinen, Qi Han, and Lieko Earle, "Circuit-level load monitoring for household energy management," *IEEE Pervasive Computing*, vol. 10, no. 1, pp. 40–48, 2011.

[91] Markus Weiss, Adrian Helfenstein, Friedemann Mattern, and Thorsten Staake, "Leveraging smart meter data to recognize home appliances," in *Proc. of IEEE Int. Conf. on Pervasive Computing and Communications (PerCom)*, Lugano, Switzerland, 2012, pp. 190–197.

[92] Hugo Goncalves, Adrian Ocneanu, and Mario Berges, "Unsupervised disaggregation of appliances using aggregated consumption data," in *Proc. of the 1st KDD Workshop on Data Mining Applications in Sustainability (SustKDD)*, San Diego, CA, USA, 2011.

[93] Alex Krizhevsky, "Learning multiple layers of features from tiny images," M.S. thesis, University of Toronto, 2009.

[94] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[95] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol, "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion," *Journal of Machine Learning Research*, vol. 11, no. 3, pp. 3371–3408, 2010.

[96] Shoko Araki, Tomoki Hayashi, Marc Delcroix, Masakiyo Fujimoto, Kazuya Takeda, and Tomohiro Nakatani, "Exploring multi-channel features for denoising-autoencoder-based speech enhancement," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Brisbane, Australia, Apr. 19-24 2015, pp. 116–120.

[97] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori, "Speech Enhancement Based on Deep Denoising Autoencoder," in *Proc. of Interspeech*, Lyon, France, Aug. 25-29 2013, pp. 436–440.

[98] V. Nair and G.E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. of the 27th Int. Conf. on Machine Learning (ICML)*, Haifa, Israel, Jun. 21-24 2010, pp. 807–814.

[99] A. Gabaldon, R. Molina, A. Marín-Parra, S. Valero-Verdu, and C. Alvarez, "Residential end-uses disaggregation and demand response evaluation using integral transforms," *Journal of Modern Power Systems and Clean Energy*, vol. 5, no. 1, pp. 91–104, 2017.

[100] M. Zhong, N. Goddard, and C. Sutton, "Latent bayesian melding for integrating individual and population models," in *Proc. of Advances in Neural Information Processing Systems*, Montréal, Canada, Dec. 7-12 2015, pp. 3618–3626.

[101] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. of the 30th Int. Conf. on Machine Learning (ICML)*, Atlanta, USA, Jun. 16-21 2013, pp. 2176–2184.

[102] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016.

[103] S. Spinsante, M. Pizzichini, M. Mencarelli, S. Squartini, and E. Gambi, "Evaluation of the Wireless M-Bus Standard for Future Smart Water Grids," in *Wireless Communications and Mobile Computing Conference, 9th International*, 2013, pp. 1382 – 1387.

[104] S. Spinsante, S. Squartini, L. Gabrielli, M. Pizzichini, E. Gambi, and F. Piazza, "Wireless M-Bus Sensor Networks for Smart Water Grids: Analysis and Results," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 579271, pp. 16, 2014.

[105] M. Fagiani, S. Squartini, L. Gabrielli, S. Spinsante, and F. Piazza, "A Review of Datasets and Load Forecasting Techniques for Smart Natural Gas and Water Grids: Analysis and Experiments," *Neurocomputing*, vol. 170, pp. 448–465, 2015.

[106] M. Fagiani, S. Squartini, M. Severini, and F. Piazza, "A Novelty Detection Approach to Identify the Occurrence of Leakage in Smart Gas and Water Grids," in *Neural Networks (IJCNN), 2015 International Joint Conference on*, July 2015, pp. 1–8.

[107] M. Fagiani, S. Squartini, L. Gabrielli, M. Pizzichini, and S. Spinsante, "Computational Intelligence in Smart water and gas grids: An up-to-

date overview," in *Neural Networks (IJCNN), 2014 International Joint Conference on*, July 2014, pp. 921–926.

[108] Marco Fagiani, Stefano Squartini, Leonardo Gabrielli, Susanna Spinsante, and Francesco Piazza, "An Experimental Review of Databases and Load Forecasting Techniques for Smart Natural Gas and Water Grids," *Neurocomputing*, 2015, to appear.

[109] Junping Liu and Mingqi Chang, "Application of the Grey Theory and the Neural Network in Water Demand Forecast," in *Natural Computation (ICNC), 2010 Sixth International Conference on*, 2010, vol. 2, pp. 1070 – 1073.

[110] Ahmad Azari, Mojtaba Shariaty-Niassar, and Mahmoud Alborzi, "Shotterm and Medium-term Gas Demand Load Forecasting by Neural Networks," *Iranian Journal of Chemistry and Chemical Engineering*, vol. 31, no. 4, pp. 77 – 84, 2012.

[111] Xingtong Zhu and Bo Xu, "Urban Water Consumption Forecast Based on QPSO-RBF Neural Network," in *Computational Intelligence and Security, Eighth International Conference on*, 2012, pp. 233 – 236.

[112] Mohsen Nasseri, Ali Moeini, and Massoud Tabesh, "Forecasting Monthly Urban Water Demand using Extended Kalman Filter and Genetic Programming," *Expert Systems with Applications*, vol. 8, no. 6, pp. 7387 – 7395, 2011.

[113] Jinming Jia and Shengyue Hao, "Water Demand Forecasting Based on Adaptive Extreme Learning Machine," in *Advances in Intelligent Systems Research*, 2013, pp. 42 – 45.

[114] Sara Silva and Jonas Almeida, "Gplab - A Genetic Programming Toolbox for MATLAB," in *In Proc. of the Nordic MATLAB Conference (NMC-2003)*, 2005, pp. 273 – 278.

[115] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme Learning Machine: Theory and Applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[116] Mantas Lukoševičius, "A Practical Guide to Applying Echo State Networks," in *Neural Networks: Tricks of the Trade*, Grégoire Montavon, GenevièveB. Orr, and Klaus-Robert Müller, Eds., vol. 7700 of *Lecture Notes in Computer Science*, pp. 659 – 686. Springer Berlin Heidelberg, 2012.

[117] GeoffreyE. Hinton, "A Practical Guide to Training Restricted Boltzmann Machines," in *Neural Networks: Tricks of the Trade*, Grégoire Montavon, GenevièveB. Orr, and Klaus-Robert Müller, Eds., vol. 7700 of *Lecture Notes in Computer Science*, pp. 599 – 619. Springer Berlin Heidelberg, 2012.

[118] M. Tanaka and M. Okutomi, "A Novel Inference of a Restricted Boltzmann Machine," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*, Aug 2014, pp. 1526–1531.

[119] M Tanaka, "Deep Neural Network," 2013, MATLAB Central File Exchange.

[120] S. Makonin, F. Popowich, L. Bartram, B. Gill, and I.V. Bajic, "AMPds: A Public Dataset for Load Disaggregation and Eco-Feedback Research," in *Electrical Power Energy Conference (EPEC), 2013 IEEE*, Aug 2013, pp. 1–6.

[121] Neil D. Bennett, Barry F.W. Croke, Giorgio Guariso, Joseph H.A. Guillaume, Serena H. Hamilton, Anthony J. Jakeman, Stefano Marsili-Libelli, Lachlan T.H. Newham, John P. Norton, Charles Perrin, Suzanne A. Pierce, Barbara Robson, Ralf Seppelt, Alexey A. Voinov, Brian D. Fath, and Vazken Andreassian, "Characterising Performance of Environmental Models," *Environmental Modelling & Software*, vol. 40, no. 0, pp. 1 – 20, 2013.

[122] Markos Markou and Sameer Singh, "Novelty Detection: A Review-Part 1: Statistical Approaches," *Signal Processing*, vol. 83, no. 12, pp. 2481–2497, 2003.

[123] Marco A.F. Pimentel, David A. Clifton, Lei Clifton, and Lionel Tarassenko, "A Review of Novelty Detection," *Signal Processing*, vol. 99, pp. 215–249, 2014.

[124] L. A. Rossman, *The EPANET Water Quality Model*, vol. 2, Research Studies Press Ltd., Somerset, England, Coulbeck B. edition, 1993, Software available at `www.epa.gov/nrmrl/wswrd/dw/epanet.html`.

[125] M.T. Nasir, M. Mysorewala, L. Cheded, B. Siddiqui, and M. Sabih, "Measurement Error Sensitivity Analysis for Detecting and Locating Leak in Pipeline using ANN and SVM," in *Multi-Conference on Systems, Signals Devices (SSD), 2014 11th International*, Feb 2014, pp. 1–4.

[126] Tracy C. Britton, Rodney A. Stewart, and Kelvin R. O'Halloran, "Smart Metering: Enabler for Rapid and Effective Post Meter Leakage Identi-

fication and Water Loss Management," *Journal of Cleaner Production*, vol. 54, no. 0, pp. 166–176, 2013.

[127] Sergios Theodoridis and Konstantinos Koutroumbas, *Pattern Recognition, Fourth Edition*, Academic Press, Burlington, 4th edition, 2008.

[128] G. Acampora, D. J. Cook, P. Rashidi, and A. V. Vasilakos, "A survey on ambient intelligence in healthcare," *Proceedings of the IEEE*, vol. 101, no. 12, pp. 2470–2494, Dec. 2013.

[129] Muhammad Mubashir, Ling Shao, and Luke Seed, "A survey on fall detection: Principles and approaches," *Neurocomputing*, vol. 100, pp. 144–152, 2013.

[130] Yun Li, K C Ho, and Mihail Popescu, "Efficient source separation algorithms for acoustic fall detection using a Microsoft Kinect," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 3, pp. 745–755, 2014.

[131] Yun Li, KC Ho, and Mihail Popescu, "A microphone array system for automatic fall detection," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 5, pp. 1291–1301, 2012.

[132] Tomi Kinnunen and Haizhou Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12–40, 2010.

[133] Nicolas Obin, "Cries and whispers-classification of vocal effort in expressive speech," in *Proc. of Interspeech*, Portland, OR, USA, Sep. 9-13 2012, pp. 2234–2237.

[134] D. Reynolds and T. Quatieri, "Speaker verification using adapted gaussian mixture models," *Digital Signal Process.*, vol. 10, no. 1, pp. 19–40, 2000.

[135] C. Bishop, *Pattern Recognition and Machine Learning*, Springer Science+Business Media, LLC, New York, 2006.

[136] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM: a library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27, 2011.

[137] Majd Alwan, Prabhu Jude Rajendran, Steve Kell, David Mack, Siddharth Dalal, Matt Wolfe, and Robin Felder, "A smart and passive floor-vibration based fall detector for elderly," in *Proc. of Inf. Commun. Technol.*, 2006, vol. 1, pp. 1003–1007.

[138] Yuuki Tachioka, Tomohiro Narita, Shinji Watanabe, and Jonathan Le Roux, "Ensemble integration of calibrated speaker localization and statistical speech detection in domestic environments," in *Proc. of HSCMA, 2014*, Florence, Italy, May 12-14 2014, pp. 162–166.

[139] P Giannoulis, A Tsiami, I Rodomagoulakis, A Katsamanis, G Potamianos, and P Maragos, "The athena-RC system for speech activity detection and speaker localization in the dirha smart home," in *Proc. of HSCMA, 2014*, Florence, Italy, May 12-14 2014, pp. 167–171.

[140] Luca Cristoforetti, Mirco Ravanelli, Maurizio Omologo, Alessandro Sosi, Alberto Abad, Martin Hagmüller, and Petros Maragos, "The DIRHA simulated corpus," in *Proc. of LREC*, Reykjavik, Iceland, May 26-31 2014, vol. 5.

[141] Mark D Smucker, James Allan, and Ben Carterette, "A comparison of statistical significance tests for information retrieval evaluation," in *Proc. of ACM Conf. on Information and Knowledge Management*, Lisboa, Portugal, Nov. 6-9 2007, pp. 623–632.

[142] A. Brutti, L. Cristoforetti, M. Matassoni, P. Svaizer, and M. Omologo, "Controllare la casa con la voce: il progetto DIRHA," in *Proc. of AISV*, Venice, Italy, 2013, pp. 89–99.

[143] Michel Vacher, Benjamin Lecouteux, and François Portet, "Multichannel automatic recognition of voice command in a multi-room smart home: an experiment involving seniors and users with visual impairment," in *Proc. Interspeech*, Singapore, 2014, pp. 1008–1012.

[144] J. F. Gemmeke, B. Ons, N. Tessema, H. Van Hamme, J. van de Loo, G. De Pauw, W. Daelemans, J. Huyghe, J. Derboven, L. Vuegen, B. Van Den Broeck, P. Karsmakers, and B. Vanrumste, "Self-taught assistive vocal interfaces: An overview of the ALADIN project," in *Proc. of Interspeech*, Lyon, France, 2013, pp. 2039–2043.

[145] S. Goetze, J. Schroder, and S. Gerlach, "Acoustic monitoring and localization for social care," *Journal of Computing Science and Engineering*, vol. 6, no. 1, pp. 40–50, 2012.

[146] D. Hollosi, J. Schroder, S. Goetze, and J.-E. Appell, "Voice activity detection driven acoustic event classification for monitoring in smart homes," in *Proc. of the 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL)*, Rome, Italy, 2010, pp. 1–5.

[147] S. Ntalampiras, I. Potamitis, and N. Fakotakis, "A multidomain approach for automatic home environmental sound classification," in *Proc. of Interspeech*, Makuhari, Japan, 2010, pp. 2210–2213.

[148] S. Ntalampiras, I. Potamitis, and N. Fakotakis, "Probabilistic Novelty Detection for Acoustic Surveillance Under Real-World Conditions," *IEEE Trans. Multimed.*, vol. 13, no. 4, pp. 713–719, Aug. 2011.

[149] M. A. F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, Jun. 2014.

[150] M. Markou and S. Singh, "Novelty detection: a review – part 1: statistical approaches," *Signal processing*, vol. 83, no. 12, pp. 2481–2497, 2003.

[151] M. Markou and S. Singh, "Novelty detection: a review – part 2: neural network based approaches," *Signal processing*, vol. 83, no. 12, pp. 2499–2521, 2003.