NON INTRUSIVE LOAD MONITORING USING ADDITIVE TIME SERIES MODELING VIA FINITE MIXTURE MODELS AGGREGATION

Soudabeh Tabarsaii

A thesis
in
The Department
of
Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Applied Science (Electrical and Computer Engineering)
Concordia University
Montréal, Québec, Canada

November 2021

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: **Soudabeh Tabarsaii**

Entitled: **Non Intrusive Load Monitoring Using Additive Time Series Modeling Via Finite Mixture Models Aggregation**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Electrical and Computer Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

______________________________ Chair
Dr. Hassan Rivaz

______________________________ Examiner
Dr. Hassan Rivaz

______________________________ Examiner
Dr. Mohammed Ouf

______________________________ Supervisor
Dr. Nizar Bouguila

______________________________ Co-supervisor
Dr. Ursula Eicker , Dr. Manar Amayri

Approved by: ______________________________
Dr. Jun Cai
Graduate Program Director

____________ 20 ______ ______________________________
Dr. Mourad Debbabi
, Dean of Gina Cody School of Engineering and Computer Science

Abstract

Non Intrusive Load Monitoring Using Additive Time Series Modeling Via Finite Mixture Models Aggregation

Soudabeh Tabarsaii

Due to an exponential rise in energy consumption, it is imperative that buildings adopt sustainable energy consumption systems. A number of studies have shown that this can be achieved by providing real-time feedback on the energy consumption of each appliance to residents. It is possible to accomplish this through non-intrusive load monitoring (NILM) that disaggregates electricity consumption of individual appliances from the total energy consumption of a household. Research on NILM typically trains the inference model for a single house which cannot be generalized to other houses. In this Master thesis, a novel approach is proposed to tackle mentioned issue.This thesis proposes to use two finite mixture models namely generalized Gaussian mixture and Gamma mixture, to create a generalizable electrical signature model for each appliance type by training over labelled data and create various combinations of appliances together. By using this strategy, a model can be used on unseen houses, without extensive training on the new house.

The issue of different measurement resolutions in the NILM area is also a considerable challenge. As a rule of thumb, state-of-the-art methods are studied using high-frequency data, which is rarely applicable in real-world situations due to smart meters' limited precision. To address this issue, the model is evaluated on three different datasets with different timestamps, AMPds, REDD and IRISE datasets. To increase the aggregation level and compare with RNN and FHMM as two well-known methods in NILM, an extension that we called DNN-Mixtures, is proposed. The results show that the proposed model can compete with state of art techniques. For evaluation, accuracy, precision, recall and F-score metrics are used.

To courageous girls around the world,
who go beyond the injustices, suffering, and all unfair barriers to become masters of their own lives.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we begin with the energy saving and load monitoring facts and motivations behind the thesis (Section 1.1). In Section 1.2, we go through the Non-intrusive load monitoring problem and its related challenges. Then, we briefly explain the objective and the organization of the thesis in Sections 1.3 and 1.4, respectively.

1.1 Energy Management

The rapid advancement of artificial intelligence (AI), internet of things (IoT), smart meters, and smart grids have a significant impact on smart sustainable cities. Sustainable smart cities strive to use available resources responsibly, conserve the environment, and improve the well-being of societies. Developing smart sustainable cities entails energy management, which involves efficient use of energy resources in order to achieve sustainability and self-sufficiency. Energy management includes monitoring and controlling electrical usage for optimizing energy utilization and consequently saving energy or reducing energy consumption.

Increasing energy demands has posed a challenge to energy conservation in recent years. A significant increase in energy consumption will cause energy crisis, climate change, and affect the economy of the countries [1]. Buildings are one of the key areas that contribute to energy consumption in which energy demand is increasing steadily over time. Buildings can be made more energy efficient either by construct/design or optimize their energy use. The latter is feasible by using advanced energy monitoring systems and consuming energy in a responsible manner. Energy waste can be reduced by monitoring the energy consumption of buildings and reporting it back to consumers for taking the appropriate action. In this regard, computer-aided approaches have recently come into the spotlight. More specifically, the growing awareness of data within companies has led to the development of data science solutions, which encompass a variety of techniques. Building energy management is now being tackled using data science to solve problems such as prediction of energy demand, evaluation of building operations, equipment status, and failures in order to optimize operating and maintenance costs, and monitoring energy consumption patterns.

1.1.1 Load Monitoring

Energy monitoring is one of significant aspects of energy management; as such there is a need to monitor the power consumption of the premises before planning some of the technical policies to minimize the energy consumption. Generally, load monitoring is a process of detecting and acquiring the load measurement in a power system, with the help of smart measurement devices. Smart measurement devices, installed by energy suppliers, provide the basis for sophisticated disaggregation algorithms and possibly recommended systems. Using such a system, consumers could be notified when appliances require maintenance and receive appliance-specific feedback. Besides, a guideline can be provided to consumers to inspire positive behavior changes by monitoring their load and providing actionable feedback.

The load monitoring approach can be either intrusive or non-intrusive, depending on the approach used for the appliance: If the load monitoring involves installing measuring devices at each load of interest, it is referred to as intrusive load monitoring (ILM). The numerous sensors in the ILM system make it expensive, complex to install, and difficult to maintain. Since the system relies on the functionality of all submeters for accurate monitoring if any submeter fails, the accuracy of the monitoring system will be affected.

On the other hand, disaggregating individual appliance consumption from the total electricity data, without extra per-appliance measurements is referred to as non-intrusive load monitoring (NILM). In other words, NILM does not require intrusion into the individual appliances; rather, it is the process of acquiring and disaggregating overall electricity consumption at the utility service entry, which offers a simple and

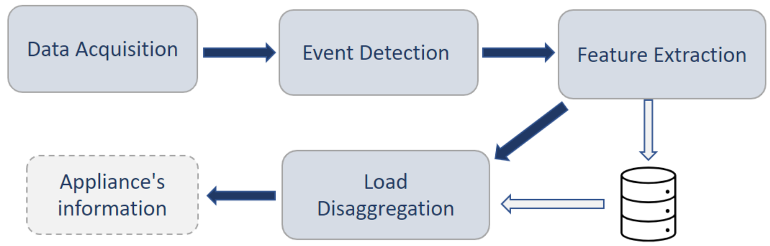


Figure 1: NILM framework block-diagram

cost-effective technique of monitoring.

1.2 NILM framework

Within the NLM framework, the first step is collecting the electrical data from the smart meter connected to the electrical panel of a building main supply. The NILM algorithm is based on the type of electrical data that the smart meter measures and its sampling rate. After collecting the data, a way to detect an event in the data has to be developed. These events represents the appliance operation states. After detecting an event, several electrical features are extracted from the data. With help of these features extracted from the appliance signature, system recognizes appliances in aggregated data. Using the features and labeled data, the disaggregation task is completed. In the following, the rudimentary concepts of each part are discussed.

1.2.1 Event Detection

The change of an appliance consumption creates a variation in the signal curve called event. The role of the event detection process in a NILM system is to detect the times when state transition actions appear from the aggregated measurements. The state transition actions usually involve appliance turn-on, turn-off, speed adjustments, and function/mode changes. For accurate load identification and valid power consumption estimation, an accurate event detection approach is a crucial prerequisite. Event detection is complicated since homes have different appliance types. In the following different appliance types and different event detection approaches are briefly discussed.

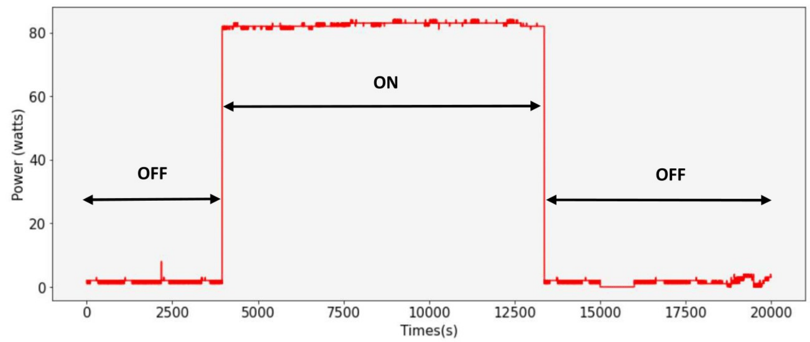


Figure 2: Light power consumption as an ON/OFF appliance

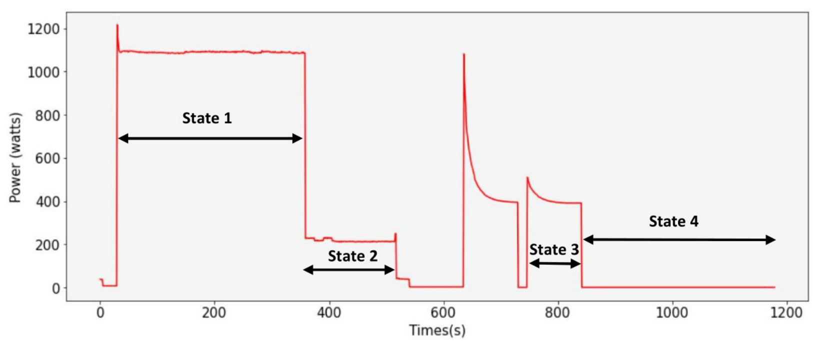


Figure 3: Dishwasher consumption as an Finite state appliance

Appliance type:

Each appliance has a different number of operational states and a different energy consumption pattern. Appliances can be divided into four different categories according to their operational characteristics, as follows:

- **ON/OFF appliances**: This class of appliances only have two states of operations ON and OFF, such as kettles, lamps , toasters, etc (Fig. 2).

- **Finite state appliances** : Finite state appliances are multi-state type with limited number of operating states. Some examples are dishwashers and washing machines. These types of devices are more complicated to model. For instance, a dish washer has several cycles that start and stop such as a water pump that drains the water between wash cycles, and a heating element that heats water (during wash cycles), dries the dishes, and can keep plates warm. These cycle could be more complicated with different temperature settings and other operation cycles (Fig. 3).

- **Continuous Variable Devices**: This category is problematic for NILM algorithm. The devices in this category don't have any fixed state and change their electrical characteristics based on their operating conditions. Some examples of appliances in this category are light dimmers and hand drills.

- **Always-ON**: This category of devices in NILM algorithms are those which are permanently ON and therefore include no events. Continuously operating appliances do exhibit small reading fluctuations, but these fluctuations are too small to be incorporated into event detection, rendering them problematic for NILM algorithms. They are low powered devices like security alarms and smoke detectors.

The current NILM approaches can be classified either as state-based or event-based based on how they detect events.

Event-based approaches:

Power monitored for the home is constantly changing (rising and falling, steps). These moves (if notable enough) can indicate that an event has occurred. This is typically called edge detection that some NILM algorithms use. These approaches are limited by fixed or adaptive threshold of changes and can be affected by the measurement noise, similarity among steady-state signatures and very low sampling rate.

state-based approaches:

These category of event detectors views edges passing as a probability distribution. In other words, appliances that changes states provide different edge measurements which have a probability distribution that match to that appliance. State-based NILMs that uses probabilistic approaches are usually based in HMM and its variants. These approaches require expert knowledge to set a-prior value for each appliance state via long periods of training. In addition, they have high computational complexity which is an important feature for some applications since it may exceed the beneficiaries.

1.2.2 Feature extraction:

A reliable NILM algorithm requires specific features or signatures that characterize appliance behavior. Each appliance type exhibits its own unique energy consumption pattern known as an appliance signature. These signatures are often used to specify operations from the aggregated load measurements. For appliances identification, the NILM uses two main classes of signatures: transient features and steady-state features

- **Transient features** : A short-term fluctuation of power or current before reaching a steady state. Using these features, appliance state transitions can be uniquely defined by identifying characteristics like size, shape, duration, and harmonics. For capturing all operation cycles, they require high sampling rates and longer monitoring times. This, in turn, requires costly hardware to be installed in households since smart meters report only low-frequency power.

- **Steady state features** : Relate to more sustained changes in power characteristics when a device is switched on. Some of these features are: active power, reactive power, current, and voltage wave forms, among others. The extraction of steady-state signature does not demand high-end metering devices and are the most frequently used features at low frequency in the literature.

1.2.3 Learning and Inference

In this step, the extracted signature is used to classify the state of an appliance and calculate its corresponding power consumption. Learning algorithms are employed to determine model parameters, while inference algorithms are employed to infer appliance states from observed aggregate power data and calculate their corresponding power consumption. A complete review of different methods is presented in next chapter.

1.3 Contributions of the Thesis

In this thesis we present a new approach based on finite mixture models for appliance recognition in NILM operational framework. We developed the approach based on the [2] while using Generalized Gaussian mixture and Gamma mixture instead of Gaussian mixture since it has shown relatively better performance than Gaussian mixture on signal processing tasks [3], [4], [5]. As a new perspective, we evaluate the model on unseen houses to show how the novel technique can tackle unsupervised NILM. Moreover, to upgrade the aggregation level, we proposed the cascade of CNN and the developed models and compare the performance with two widely used methods in NILM. Although these methods do not offer the ultimate and the most accurate solution, they open doors to more robust opportunities. More details will be presented through the next chapters.

1.4 Organization of the Thesis

The thesis is organized as follows : In Chapter 2, we present an overview of different methodologies in NILM including classifications and unsupervised methods. Then, finite mixture model and its application in NILM is described. In Chapter 3 we give a comprehensive description of finite mixture models, and two adopted ones in this thesis and their aggregation procedure. Following that a detailed explanation of the proposed method. Chapter 4 is where we evaluate the methods we talked about in previous chapters, in three different validation on three different datasets. Finally, we conclude the thesis with use-case, conclusion, and future work in Chapter 5.

Chapter 2

Background and Literature Review

This chapter begins with a brief introduction to related work in NILM (Section 2.1) followed by a quick historical glance at different methodologies adopted in this field in Sections 2.2 and 2.3. In Section 2.4 and 2.5 important state of art works in NILM area are covered. Finally in Section 2.6 the finite-mixture based approach is described. The chapter is concluded with a summary in Section 2.7.

2.1 Related work

Originally, Hart [6] proposed the Non-intrusive Load Monitor (NILM) as a paradigm aimed at providing information about energy consumption from individual appliances by analyzing single-point electrical data. The proposed simple linear model has been extended by a number of studies to incorporate directly sampled quantities to increase the resolving power of the P-Q space [7]

Since their publication, NILM research has continued to progress mainly through the adoption of more accurate disaggregation algorithms. Some of the early works are as follows : Fourier harmonic disaggregation techniques is aimed at decomposing and separating fine-grained devices such as low-load complex appliances in homes, offices, and industries [8]. Steady-state monitoring techniques are effectively employed in some environments, such as homes and small businesses [9]. In contrast, large industrial facilities and companies require more complex approaches due to the high amount load balancing and power factor correction [10].

Some surveys on NILM [11],[12] give a good overview of techniques proposed in this field. According to these surveys, NILM systems can be grouped into two families according to how they identify loads. The first one is supervised method which is based on labeled data and the latter is unsupervised methods. In the following sections we review the both categories.

2.2 Classification Methods

Generally, supervised learning is based on training a model in a first stage then using it to recognize the appliance load. In such cases, sub-metered appliances data or labeled observations must be gathered from the target building. The process of collecting these data is time-consuming, costly, and limits the scalability of NILM systems [13].

One category of the supervised methods concerns optimization approaches. They consider NILM problem as an optimization task. In this process a comparison between extracted features and the load features within a database is performed to find the closest possible match and seek for the best combination of devices included in the database. Some examples of this methods are Integer programming [14], genetic algorithms [15].

Graph-based signal processing methods are another category based on load waveform analyzing. These approaches do not require feature detection to carry out the disaggregation task since the study mostly focuses on the smoothness of a discrete signal using a graph.[16], [17]

Several machine learning classifiers are also placed in this class of NILM approaches. Support vector Machine (SVM) is a popular supervised training algorithm that uses optimal line separation between classes . Optimal line separation is finding a line that performs classification with the maximum margin where none of the training data falls within. SVM usually performs significantly better than other methods on classification tasks as long as the dataset is small.

SVMs have been employed in a number of NILM algorithms. By using pairwise SVMs with linear kernels, Figueiredo et al. [18] were able to disaggregate loads with very high accuracy. They used pairwise classification and all-against-one classifier to provide multiclassification in which the first one achieved high accuracy and the latter was less accurate. other examples are [19], [20], [21].

Another machine learning technique used in NILM area is K-nearest neighbour (KNN). The KNN classifier is an instance based learning method where the classification function is approximated by a majority vote of neighbors using a distance metric. K is the number of neighbors which can be calculated using cross-validation. The whole classification is based on the optimal choice of K. If K is small then the results might be sensitive to noise and if its large it may cause a decrease in accuracy. Berges et al. (2009) used 1-NN with Fourier regression coefficients to classify unseen transient signatures [22] and later improved the work using Euclidean distance measurements between signature vectors [23]. In [24] , authors used K-NN recognizer based on the information of transient features to identify the type and status of an appliance.

Many other supervised machine learning classifications used in NILM are overviewed in [11], [12].

2.3 Unsupervised methods

The second category consists of unsupervised methods. Unlike supervised NILM, unsupervised NILM techniques do not require pre-training and thus are suitable for real time NILM application. According to [11] unsupervised methods in NILM can be divided into three sub-groups. First approaches are based on unlabelled training data to build appliance models or create appliances databases. Appliance models can be generated either manually [25] or automatically [26] during training. This sub-category is usually based on hierarchical clustering or Hidden Markov Models (HMMs). One of the early work is [27] , where author presents a method based on conditional factorial HMM (FHMM) and Hidden semi-Markov models which were not able to disaggregate base load and refrigerator and were prone to convergence to local minima. More recent approaches are [26],[28],[29] and [30] based on FHMM, differential FHMM, additive FHMM and Hierarchical Dirichlet process and Hidden Semi-Markov Model factorial structure , respectively. Moreover, in [31] and [32] in order to improve performance of standard FHMM, context information and interactions chains are used, respectively. All these state-based approaches require user expert knowledge to set a prior value for each appliance, and require a training set to build the state models. The performance of these methods depend on how accurate generated models approximate appliance true usage. To the best of our knowledge, these methods have not been generalized across different unseen houses.

The second sub-group of approaches use labeled data from measured data of seen houses and apply the acquired model on an unseen house. Some deep learning methods can be placed within this subset of approaches [33]. However, as any other deep learning approach, it requires a large training set which leads to high computational complexity. In [34] KNN is suggested for similar houses, based on the assumption that if two houses have a similar aggregate consumption during different seasons, it is likely that they will also have a similar consumption at appliance level. The number of studies conducted in this sub-category are rare, cannot generalize well across different geographical locations and are sensitive to outliers (such as homes consuming very high amount of energy). In this case, candidates for energy feedback may have wrong outcomes due to the fact that 'neighboring' homes are not indicative of their energy consumption [34].

The last sub-group does not require training such as [35], in which authors presents unsupervised event detector based on a Fisher kernel discriminant analysis of the current harmonics features. They were able to detect classical abrupt steady state changes and active sessions like pulses and variable load intervals. In [36] an unsupervised approach based on clustering and matching pursuit is proposed. Both active and reactive power were analyzed but the method performs poorly for appliance loads below 400W and claimed that the results might improve only if additional features are included. [37] presents an unsupervised load disaggregation approach that is capable of identifying device operations based on changes in power characteristics when devices are switched on/off or switched to a different power state. The proposed algorithm does not require a priori information about the system and proactively adjusts to new or removed devices and the labeling of the power states and appliances was made empirically. Each detected appliance is mapped to a known power state of the ground truth or to the "unknown" appliance state container which is not previously identified by observer. Performance gradually improves requiring some time to reach high accuracy. Following that, [38] proposed the event based method forming patterns of appliance signatures on-the-fly using clustering, and based on pattern matching to label data.
These methods are interesting to investigate but not accurate enough compared to

other methods [11].

In the following we review some of the state-of-art classification methods in this area.

2.4 Deep Learning

The term "deep learning" describes a machine learning methodology that relies heavily on the study of the human brain in conjunction with statistics and applications of mathematics. Deep learning models are flexible, as they may be used to solve a variety of problems. The use of deep architectures has been around for years, but have not shown much success in the past. Their success appeared after the creation of new optimization techniques and architectures and a large amount of data available in many areas, which are essential to training deep networks. Deep learning models are beneficial for their automatic feature learning. They need a great amount of prior knowledge. The prior is mainly comes from the fact that the models are tuned in relation to a particular problem which is incorporated with the aid of humans.

Different deep learning structures, including Recurrent Neural Network (RNN) [39][40][41], Convolutional Neural Network (CNN) [42], [43], [40], [44] Autoencoder [45] and a combination of deep learning and HMM [42] have been deployed to develop NILM models. For instance, as one of the first work, Kelly and Knottenbelt [45] compared the disaggregation performance of the traditional machine learning methods (e.g., FHMMs) with the deep learning methods such as Autoencoders and LSTM networks and the results show that the deep learning methods outperform traditional methods. In the following parts, some of these structures are investigated.

RNNs are characterized by a "memory" that incorporates information from previous inputs to influence the current input and output. Even though it has huge potential, there are some noticeable issues with RNN such as vanishing and exploding gradients and difficulty in learning long sequences.

Long short-term memory (LSTM) is a famous type of recurrent neural network. It was first published in 1997 by Hochreiter and Schmidhuber [46] and has been used widely with great success in a wide range of fields such as speech recognition, time-series data forecasting, etc. An LSTM uses a similar control strategy to a recurrent neural network. Basically, it processes data transmitting information as it propagates forward. The main difference is within the internal cells that include forget gate, which allows LSTM to keep or forget the irrelevant information. The block diagram of LSTM is depicted in Fig. 4 . Following this structure, later time steps can make use of information from earlier time steps, which bring the long-term memory feature to the cell.

The ability to use long-term dependency of the time series signal brought the idea of applicability of LSTM in the NILM area like other aforementioned fields. For instance, [39] demonstrated the feasibility of LSTM for an eventless power disaggregation on low frequency data, in [40] bidirectional LSTM with parallel CNN layers for feature extraction is investigated to empirically study the disaggregation and [41] studied the developed model, context aware bidirectional LSTM on different datasets.

The convolutional neural network (CNN) is a type of feed-forward neural network that is inspired by the visual system of mammals, due to its ability to recognize complex patterns with high accuracy. Therefore, the majority of its early applications were in image recognition.

The advantage of using CNNs for sequence classification is that they can learn directly from the raw time series data, without the need for domain expertise to design input features. Ideally, the model will be able to learn an internal representation of the time series data and achieve capabilities comparable to models that use engineered features. Therefore, CNNs can be used to filter out specific power variation sequence of an appliance from an aggregated signal based on its specific short sequence features.

Different CNNs have been used with state- and event-based features, classification- and regression-based approaches, sequence to sequence and sequence to point prediction and in some tasks with some other tools like HMM and RNN [42]. For instance in [43] , sequence to sequence matching was done by looking at a window of aggregated input signal with a window of respective energy signal of the target appliance with an equal length as output. Also in [40], Zhang et al. proposed a regression sequence to sequence CNN is used to estimate the transient energy demand of a single appliance for a given aggregated energy signal.

In the sequence to point CNN structure, the whole prediction of each signal is reflected in one point form the sequence of target appliance for example the [44] predict energy consumption of appliances at the midpoint of the output window which outperforms sequence to sequence approach used in [45]. Other CNN structures like [47]

which explored a multi-channel 1D CNN are also proposed in the NILM area.
There are also works using a combination of DNNs. For example, by combining CNNs with variational autoencoders, Sirojan et al. [42] showed that their method outperforms the one proposed in [44]
Deep learning, in general can offer enormous potential, however, the largest challenges are determining which architectures to use, how to train it appropriately, and maintaining reliability across a large amount of datasets.

2.5 Hidden Markov Model

In view of the time-series nature of electrical data and the fact that different appliances can be in multiple operational states, HMM presents an elegant learning mechanism for purpose of disaggregation. The Factorial Hidden Markov Model (FHMM) is a combination of multiple single HMMs evolving in time separately, however the output of the model Y_t at any time t is dependent on the current states of all the HMM's. Essentially, each appliance has its own HMM that evolves in parallel to the others.

Within the context of NILM, observed variables are power consumption and hidden variables are considered as appliances working states (ON,OFF, standby).The model can be characterized as follows : Considering the observable sequences $Y = \{y_1, y_2, ..., y_T\}$, then $S = \{S^{(1)}, S^{(2)}, ..., S^{(M)}\}$ represents the set of hidden state sequences where $S^{(i)} = \{S_1^{(i)}, S_2^{(i)}, ..., S_T^{(i)}\}$ is the hidden state sequence of the chain i. The transition matrix $A = a_{i,j}, 1 \leq i, j \geq M$ represents state transition from $S^{(i)}$ to $S^{(j)}$ in which $a_{i,j} = P(x_{t+1} = S^{(j)}|x_t = S^{(i)})$, where $a_{i,j} > 0$ and $\sum_{j=0}^{M} a_{i,j} = 1$. Emission matrix, $B = P(y_t|S^{(j)})$, represents a symbol in an actual state. In the appliance model, it shows the possible power values in each state of an appliance.The initial state probability distribution $\pi = \pi_i$ denotes the probability of each state in initial time of $t = 1$. Finally, the set of parameters for the model would be $\lambda = \{\pi, B, A\}$. The structure of FHMM is demonstrated in Fig. 5. Generally, the task of FHMM consists of two main parts of inference and learning. The inference problem consists of computing the probabilities of the hidden variables given the observations. To solve this part Viterbi algorithm can be used, but mostly in the case of an FHMM, approximation methods such as Gibbs sampling is applied. The learning task consists of learning the parameters for a given structure. The parameters of a factorial HMM can be achieved via the Expectation Maximization (EM) 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).

Many FHMM based methods are used in energy disaggregation field. In [27] KIM presented an unsupervised technique for energy disaggregation using a combination of four FHMM: Regular FHMM in which independent hidden state chains evolving in parallel and the observation is a joint function of these chains, Factorial Hidden Semi-Markov Model (FHSMM), in which state are explicitly assigned to occupancy duration periods that could fit to exponential or Gamma distributions, Conditional FHMM (CFHMM), in which states changes are constrained to fulfill some conditions based on current state and finally, Conditional FHSMM (CFHSMM) which combines both FHSMM and CFHMM models. Expectation Maximization is used to learn parameters and Maximum Likelihood Estimation is adopted to infer load states. However, the proposed technique is limited to a few number of appliances.

In [28] by focusing on two complementary models, the additive and the difference FHMM, a novel inference method is proposed which does not suffer from local optima. Nevertheless the performance is weak for kitchen outlet. In another approach [26] used different HMM method in which authors introduced training process in which prior knowledge of the generic appliance types are tuned to specific appliance instances using only aggregate data from the home. Moreover, an extension of viterbi algorithm is used to infer the model. This approach mostly performs better on cycled appliances like fridge, but for more complicated appliances it is difficult. An unsupervised FHMM model is presented in [48]. Authors proposed non-parametric model using low-frequency real power feature and they used the combination of slice sampling and Gibbs sampling to infer the model in which process of detecting appliances and disaggregation is done simultaneously. Due to the limitation of inference for larger disaggregation problems, this approach is prone to local optima.

HMM-based disaggregation approaches have been widely used, yet it requires expert knowledge in setting apriori values for each appliance state. Thus, their performance is dependent on how well the generated models approximate appliance usage.

In addition, HMM-based approaches have better performance for controlled multi-state appliances like refrigerator, but their performance degrades for uncontrolled multi-state and variable appliances. An example is a lighting circuit with a number of independent on/off lights. The overall lighting circuit is a multi-state load but with random state transitions and random duration at a particular power consumption state.[49]

2.6 Finite Mixture Models

Finite mixture models (FMM) offers a flexible approach to describe probability distributions. Probability distributions are the basis of model-based statistical data analysis. Typically, they describe the generative model underlying the data. Finite mixture models have been widely used, mainly for clustering, in many applications from different domains such as pattern recognition, data mining, computer vision, and machine learning. The interested reader is referred to [50] and references therein for more details and applications of mixture models. A very famous one is Gaussian mixture model which has been widely used in many applications [51], [44], [52]. Recently, GMMs have received significant attention in additive time series disaggregation. Finite mixture models are usually applied for representing complex density functions. Taking advantage of the additive properties of the Gaussian that meet the additive properties of the features of additive time series, Gaussian mixture models have been used recently for load identification from aggregated data. For instance, in [2] a method based on GMMs has been applied for the first time for appliance recognition. A more comprehensive application of this approach can be found in [53] in which generalizable electrical signature models have been created by training over labelled data from different appliances and distinct combinations of them by merging the generated models. They used the ACS-F2 dataset [54] containing electrical characteristics acquired from different brands of domestic appliances. Their goal was to derive a generalizable. model of different brands and identify an unseen brand of the appliance from the aggregated data of a residential building. Though the idea of generating NILM models using a generative method was novel and interesting, the results were not competitive enough with RNN.

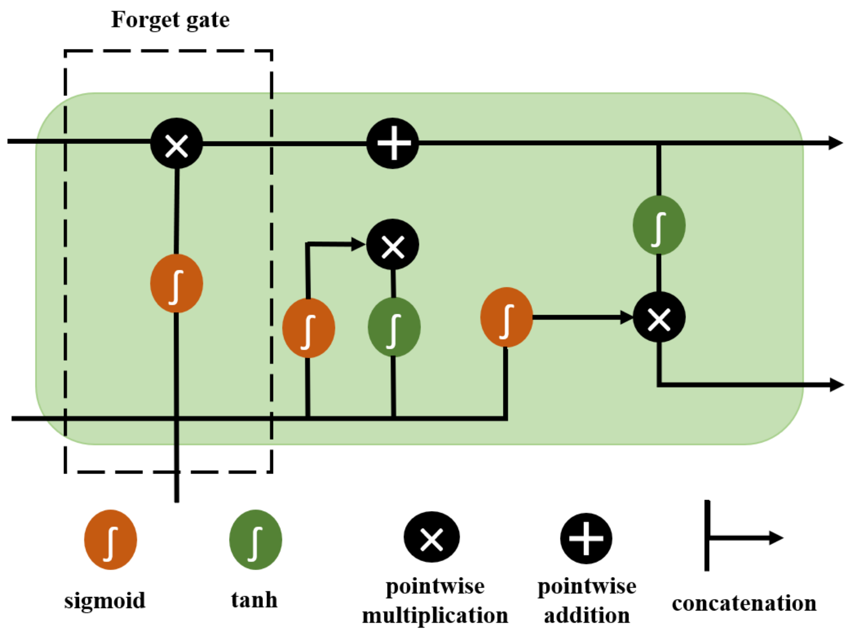


Figure 4: LSTM cell block

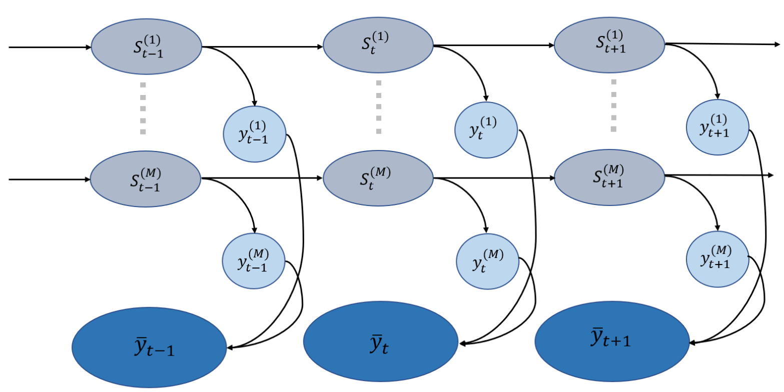


Figure 5: Graphical representation of FHMM

2.7 Summary

In this chapter, we reviewed a wide range of NILM methods, from conventional supervised methods to different categories of unsupervised methods. These methods mostly either rely on a large amount of data to train or suffers from poor accuracy. Based on the idea of [2], in the next chapter we propose a generative disaggregation model based on statistical features of appliance consumption. The objective is to mitigate the need for sophisticated plug- or circuit-level measurement in every new house under study. Furthermore, we explore the performance of generalized Gaussian mixture and Gamma mixture compared to GMMs as a more powerful technique to model data statistically.

Chapter 3

Methods and Algorithms

In this chapter, we present an overview about the finite mixture models in Section 3.1. Then, in Section 3.2, we provide further details on the NILM methodology used.. Finally, we conclude the chapter and summarize the findings in Section 3.3.

3.1 Finite Mixture Model

In the case of statistical data modeling, the assumption that each sample follows a unimodal distribution is too restrictive and may appear less intuitive. Mixture models allow to represent arbitrarily complex PDFs which makes them ideal for complex data modeling. Let x be a random variable following a K-component finite mixture of distributions, then its PDF can be written as following [50]

$$p(x|\theta) = \sum_{i=1}^{K} w_i p(x|\theta_i) \tag{1}$$

where $w_1, \ldots, w_K$ are the mixing probabilities which are positive and sum to one. Each θ_i is the set of parameters defining the ith component, and $\theta = \{\theta_1, \ldots, \theta_K, w_1, \ldots, w_K\}$ is the complete set of parameters specifying the mixture model.

A Gaussian mixture model is a common example of this, which consists of a mixture model comprising of two or more Gaussian distributions. The Gaussian distribution is given as follows:

$$\mathcal{N}(x|m, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\dfrac{(x-m)^2}{\sigma^2}} \tag{2}$$

in which m is mean, and σ^2 is the variance. In this case the mixture model is expressed as follows :

$$p(x|\theta) = \sum_{i=1}^{K} w_i \mathcal{N}(x|m_i, \sigma_i) \tag{3}$$

Within the mixture model, each individual Gaussian density is regarded as a component of the mixture and has specific values for its mean and variance. Given a set of N independent and identically distributed samples $\mathcal{X} = \{x^{(1)}, ..., x^{(N)}\}$, the log-likelihood corresponding to a K-component mixture is

$$\log p(\mathcal{X}|\theta) = \log \prod_{j=1}^{N} p(x^j|\theta) = \sum_{j=1}^{N} \log \sum_{i=1}^{K} w_i p(x^j|\theta_i) \tag{4}$$

To estimate θ, the maximum likelihood (ML) estimation approach is usually adopted as shown in equation (5) below:

$$\theta_{ML} = \arg\max_{\theta} \log p(\mathcal{X}|\theta) \tag{5}$$

The usual choice to solve the above-mentioned ML equation is the expectation-maximization (EM) algorithm [55].

EM algorithm is an iterative algorithm which finds local maximum of $\log p(\mathcal{X}|\theta)$. In the EM framework for finite mixture estimation, the observed data $\mathcal{X}$ are considered as being incomplete. If we consider that each x^j is derived from one of the component distributions of the mixture model, it will be associated with z_i^j equal to one or zero according to whether it is derived from ith component or not. Considering $\mathcal{X}$ and $\mathcal{Z} = \{z_i^j\}_{i=1,...,K}^{j=1,...,N}$ together as complete data, the complete formulation of log-likelihood would be:

$$\log\ p(\mathcal{X}, \mathcal{Z}|\ \theta) = \sum_{j=1}^{N} \sum_{i=1}^{K} z_i^j \log\ \ w_i p(x^j|\ \theta_i) \tag{6}$$

3.1.1 Expectation-Maximization algorithm

The Expectation-Maximization algorithm is a method for finding the maximum likelihood estimate in models containing latent variables. By beginning with initial parameter estimates and then iterating through an Expectation Step and a Maximization Step, it finds the solution that converges on a stable value for the parameters. In the

Expectation step, the parameter estimates are assumed to be fixed, and the latent variables' expected values are computed. The Maximization step takes the expected values of the latent variables and finds updated values for the previous parameter estimates that maximize the likelihood function.
For instance, in case of Gaussian mixture, The Expectation step assumes that the values of all the 3 parameters for the Gaussians in the model are fixed and then computes the probability that each given data point is derived from the individual Gaussians in the model. The probability of a data point coming from a distribution is referred to as the responsibility of the distribution to a particular point. Based on the calculated responsibility values, the Maximization step attempts to maximize the likelihood function across all model parameters by assuming that these responsibilities are fixed.

Expectation step

The responsibilities are posterior probabilities for a given component within the model and can be calculated as follows:

$$\begin{aligned} r_i^j = p(z_i^j = 1 \mid x^j,\ \theta_i) = \frac{p(z_i^j = 1).p(x^j|z_i^j = 1)}{\sum_{i=1}^K p(z_i^j = 1)p(x^j|z_i^j = 1)} &= \frac{w_i p(x^j \mid \theta_i)}{\sum_{i=1}^K w_i p(x^j \mid \theta_i)} \\ &= \frac{w_i \mathcal{N}(x^j \mid m_i, \sigma_i)}{\sum_{i=1}^K w_i \mathcal{N}(x^j \mid m_i, \sigma_i)} \end{aligned} \tag{7}$$

$\sum_{i=1}^K w_i \mathcal{N}(x^j \mid m_i, \sigma_i)$ is the normaliser term across all components.
Posterior probability, in Bayes theorem, is the updated probability after taking into consideration the new information. It can subsequently become a prior for a new updated posterior probability as new information arrives iterations. Here, the responsibility (or the posterior probability) of a component of the model to a data point is the normalized probability of a given data belonging to a specific Gaussian within the mixture model, then weighted by the estimated mixture proportions w_i. If we consider the w_i as the prior information then (7) is the posterior probability for a specific distribution given the observed data.

Maximization step

The responsibilities can be summed and normalised to estimate the contribution of the individual Gaussians to the observed data:

$$w_i = \frac{1}{N} \sum_i r_i \tag{8}$$

The responsibilities of each data point to the different distributions in the model can also be used to estimate the parameters, which in case of Gaussian are mean m and standard deviation σ:

$$m_i = \frac{\sum_j r^j x^j}{\sum_j r^j} \tag{9}$$

$$\sigma_i = \frac{\sum_j r^j (x - m^j)(x - m^j)}{\sum_j r^j} \tag{10}$$

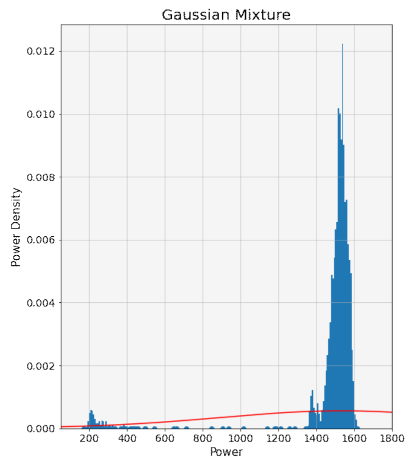
3.1.2 Choice of distribution

Mixture models consider the data to be composed of multiple components drawn from a probability distribution. The data is modeled based on this assumption and the important part of designing is the choice of distribution. In recent years, Gaussian distributions have been widely applied in different fields [51], [44], [52].

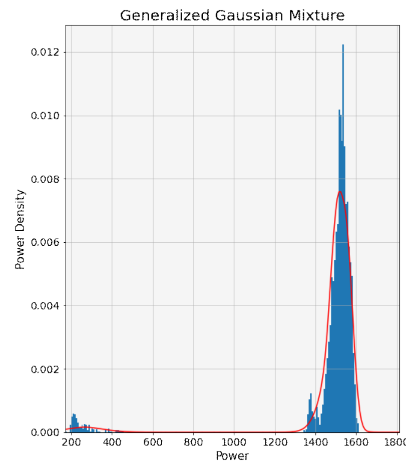
Despite the wide use of Gaussian mixture model because of its simplicity, many real-world applications cannot consider the Gaussian assumption which fails to fit the shape of the data. Despite its usefulness the Gaussian distribution is not flexible and robust enough. Indeed, it has a very rigid shape (i.e. symmetric bell shaped) and it is known to be very sensitive to outliers.

In order to overcome the limitations of the Gaussian mixture, the generalized Gaussian mixture (Generalized GMM) has been proposed as a more flexible model and applied to a variety of real life applications such as texture discrimination and retrieval [3], object detection [4], and image and video processing [5]. The generalized Gaussian mixture has an extra shape parameter as compared to the Gaussian. The shape parameter controls the tails of the distribution and changing its values makes it peaked or flat.

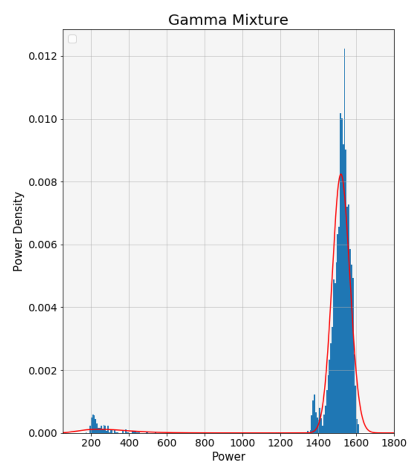
The generalized Gaussian distribution offers more flexibility, yet like the Gaussian, it is always symmetric. To consider data skewness, the Gamma distribution could be deployed as an interesting flexible choice for the mixture components [56]. Finite Gamma mixture models have been applied in several challenging tasks such as target recognition [57], and analyzing high resolution data like SAR images [58].



(a)



(b)



(c)

Figure 6: PDF of Microwave power consumption in house 1 from REDD dataset [59] with three different mixture models. (a) Approximation of density by Gaussian mixture with four components, (b) Approximation of density by generalized Gaussian mixture with three components, (c) Approximation of density by Gamma mixture with two components.

To demonstrate the difference of aforementioned mixture models, we modeled the PDF of Microwave power consumption with all of them (Figure6). According to the results, generalized GMM and Gamma mixture could approximate the PDF with only three and two components, respectively; while GMM could not approximate the PDF even with 5 components. This clearly shows the superior ability, generalization capability, and flexibility of the generalized GMM and Gamma mixture as compared to the GMM.

In the following we explain each of generalized Gaussian and Gamma distributions with details and explain the aggregation process of each, to explore the applicability of them in disaggregation process.

3.1.3 Generalized Gaussian

As previously mentioned in section 3.1.2, generalized Gaussian distribution has one more parameter than Gaussian called shape parameter. The formulation of the probability density is as follows [5]:

$$p_i(x|m_i, \sigma_i, \alpha_i) = \frac{\alpha_i \gamma_i}{\Gamma(1/\alpha_i)} e^{-\gamma_i^{\alpha_i}[|(x-m_i)|]^{\alpha_i}} \tag{11}$$

where m_i, σ_i and α_i are mean, standard deviation, and shape parameter, respectively; and $\gamma_i = \frac{1}{\sigma_i} \frac{\Gamma(3/\sigma_i)}{\Gamma(1/\sigma_i)}^{1/2}$. The shape parameter α_i is the rate of exponential decay of the PDF and controls the deviation from the normality of the distribution. By considering $\alpha_i = 2$ the density reduces to the Normal distribution and by choosing $\alpha = 1$ it becomes Laplacian distribution (See Fig. 7).

Generalized Gaussian Mixture

To train generalized Gaussian mixture, EM algorithm mentioned in section 3.1.1 is adopted as follows :(i) finding clusters with K-Means algorithm in the first step, and using produced mixing weights, centroids, calculated variance, and a shape equal to 2 (i.e. we start by supposing that it is a Gaussian mixture), as initial values of the mixture parameters θ, (ii) estimating the mixture parameters with EM algorithm.
The E-step (7) becomes :

$$r_i^{j\,t} = \frac{w_i p(x^j|m_i, \sigma_i, \alpha_i)}{\sum_{i=1}^{K} w_i p(x^j|m_i, \sigma_i, \alpha_i)} \tag{12}$$

And the M-step leads to :

$$\hat{w}_i^{(t+1)} = \frac{1}{N}\sum_{j=1}^{N} r_i^{j\,t} \tag{13}$$

$$\hat{m}_i^{(t+1)} = \frac{\sum_{j=1}^{N} r_i^{j\,t} x^j}{\sum_{j=1}^{N} r_i^{j\,t}} \tag{14}$$

$$\hat{\sigma_i^2}^{(t+1)} = \frac{\sum_{j=1}^{N} r_i^{j\,t}(x^j - m_i^t)^2}{\sum_{j=1}^{N} r_i^{j\,t}} \tag{15}$$

For the shape parameter, Newton-Raphson method is used:

$$\hat{\alpha}_i^{(t+1)} = \alpha_i^t - \frac{\partial^2 \log[p(\mathcal{X}|\theta_i^t)]}{\partial^2 \alpha_i}^{-1} \frac{\partial \log[p(\mathcal{X}|\theta_i^t)]}{\partial \alpha_i} \tag{16}$$

More details about the learning of generalized Gaussian model can be found in [5].

Generalized Gaussian Aggregation

In[58], it is showed that the PDF of the sum of two generalized Gaussian random variables can be approximated as generalized Gaussian distributed. Consider $X \sim GGD(m_1, \sigma_1, \alpha_1)$ and $Y \sim GGD(m_2, \sigma_2, \alpha_2)$ as two independent variables and Z as their sum. Same as in the Gaussian assumption, mean and variance of Z can be inferred as $m = m_1 + m_2$ and $\sigma^2 = \sigma_1^2 + \sigma_1^2$. However, the main difficulty is estimating the shape parameter of the resulting generalized Gaussian distribution. In [60], three methods are introduced to estimate the shape factor.

The first method is based on Kurtosis. The Kurtosis of the sum is known, thus the shape factor can be calculated by equating the two Kurtosises together. By considering $\delta = \frac{\sigma_1}{\sigma_2}$ and γ as the shape parameter of Z, it is possible to show that [60]:

$$\frac{\Gamma(\frac{1}{\gamma})\Gamma(\frac{5}{\gamma})}{\Gamma(\frac{3}{\gamma})^2} = \frac{\delta}{(1+\delta)^2}(\delta\frac{\delta_1(\frac{1}{\alpha_1})\delta_1(\frac{5}{\alpha_1})}{(\frac{3}{\alpha_1})^2} + \frac{\delta_2(\frac{1}{\alpha_2})\delta_2(\frac{5}{\alpha_2})}{\delta(\frac{3}{\alpha_2})^2} + 6) \tag{17}$$

In the second approach, the γ is adjusted to give the best possible approximation to the PDF at the tail by minimizing the error between the exact PDF and the approximated PDF :

$$\gamma_{Tail} =_{\gamma>0} \int_{n\sigma}^{\infty} (f_{approx}(z) - f_Z(z)^2 dz \tag{18}$$

where n is set to specify the region of the tail of the distribution. The third method is also based on minimizing the error between cumulative distribution (CDF) of Z and the approximated CDF.

$$\gamma_{Tail} =_{\gamma>0} \int_{0}^{\infty} (F_{approx}(z) - F_Z(z)^2 dz \tag{19}$$

The previous two optimization problems contain integrals of Foxs H-functions of one and two variables that are not solvable in closed form [60]. Therefore, in this thesis the shape parameter of the generalized Gaussian representing the combination of two appliances data is approximated using (17). The schematic aggregation of two generalized Gaussian random variables is displayed in Figure8.b.

3.1.4 Gamma distribution

Another method to construct the PDF of a random variable without making Gaussianity assumption is Gamma Mixture. The Gamma distribution has two parameters namely a shape parameter α_i and a scale parameter β_i:

$$p_i(x;\alpha_i,\beta_i) = \frac{x^{\alpha_i-1}e^{\frac{-x}{\beta_i}}}{\Gamma(\alpha_i)\beta_i^{\alpha_i}} \quad x>0 \;\; \alpha_i,\beta_i>0 \tag{20}$$

where $\Gamma(.)$ is the Gamma function, when $\alpha_i > 1$ the distribution is bell-shaped, indicating low rate heterogeneity and when $\alpha_i < 1$ the distribution is very skewed and L-shaped, which designates high level of rate variation. This feature makes the distribution more appropriate for accommodating different semi-bounded data sets with variety of rate levels (Figure9).

Gamma Mixture

The Gamma mixture model is obtained by considering the Gamma PDF [57] in (1) which gives a parameter vector $\theta_i = (\alpha_i, \beta_i)$ for each mixture component. The Gamma mixture model has been widely applied in the literature and the estimation of its parameters via the EM framework has been developed and discussed in many works (see, for instance, [57], [61]–[63]).

Gamma Mixture Aggregation

For the summation property, the common approach is based on Method-of-Moment matching. In this approach, Z, the sum of M variables following M Gamma distributions with parameters (α_i, β_i) is considered as a Gamma random variable with the same first and second moments [64]:

$$E(Z) \; = \; \sum_{i=1}^{M} \alpha_i \beta_i \tag{21}$$

$$\sigma^2 = Var(Z) \; = \; \sum_{i=1}^{M} \alpha_i \beta_i^2 \tag{22}$$

which leads to the following parameters of the Gamma distribution that Z follows:

$$\alpha = \frac{(\sum_{i=1}^{M} \alpha_i \beta_i)^2}{\sum_{i=1}^{M} \alpha_i \beta_i^2}, \;\; \beta = \frac{\sum_{i=1}^{M} \alpha_i \beta_i}{\sum_{i=1}^{M} \alpha_i \beta_i^2} \tag{23}$$

According to [64], this approximation is valid when the shape parameters are not too small and scale parameters of two variables differ by no more than a factor 10. The schematic aggregation of two Gamma random variables is depicted in Figure8.c.

3.2 Methodology

The proposed disaggregation framework is based on two parts, namely, training and testing, where we deploy the aggregation properties developed in the previous section. In addition, we propose a cascade approach as an extended model in order to investigate and analyze our approach with respect to other comparable methods.

3.2.1 Training Phase

In this part of the framework, the distribution of each appliance signal is estimated by a mixture model using labeled data. The main challenge here is the automatic selection of the number of mixture components which is highly related to number of operations. If we look at Figure10, for instance, the ON state of dishwasher consists of three main power operations which corresponds to the optimal number of components. In this thesis, we have used the Bayesian Information Criterion (BIC) for model selection. Choosing the optimal number of mixture's components allows to avoid under- or over-fitting problems. Afterwards, similar to [53] we created all possible combinations of aggregated data with acquired parameters and use aforementioned approaches in previous section to estimate the aggregated PDF.

The aggregated data can be derived from any combination of the appliances running at the same time. We create all possible combinations of appliances models. In the first scenario, each appliance model is approximated by generalized Gaussian mixture and combined through the merging procedure as summarized in algorithm I. As explained in section 3.1.3, weight, mean, variance and shape parameter of each combination of appliances are obtained. The same methodology is followed using the Gamma mixture with its own parameters as detailed in the previous section using mainly (23) (algorithm II). In both scenarios, a dictionary of model parameters through labelled data is achieved for test step and saved as a separated database to be deployed for testing phase. All the models are learned only for the ON state sequences of the appliances. In order to classify GND signal (OFF states), we manually include a Gaussian model. In other words, a Gaussian with zero mean and variance of one, which coincides with a generalized Gaussian with shape parameter of two, and with a gamma with small shape parameter and scale parameter of one.

Algorithm 1: Merge Two Generalized Gaussian Models

Input : Two generalized Gaussian model parameters $\lambda_1(\mu_1, \sigma_1^2, \omega_1, \alpha_1)$,$\lambda_2(\mu_2, \sigma_2^2, \omega_2, \alpha_2)$

Output: Merged model $\lambda(\mu, \sigma^2, \omega, \alpha)$

$M \leftarrow$ *total number of combination of components.*;

for $m := 1$ *to* M **do**

  $n_1 \leftarrow$ *number of components in* $M_1(\lambda_1)$;

  $n_2 \leftarrow$ *number of components in* $M_2(\lambda_2)$;

  $k \leftarrow 0$ **for** $k_1 := 1$ *to* n_1 **do**

    for $k_2 := 1$ *to* n_2 **do**

      $m_{m,k} \leftarrow m_{m,k_1} + m_{m,k_2}$

      $\sigma_{m,k} \leftarrow \sigma^2_{m,k_1} + \sigma^2_{m,k_2}$

      $\omega_{m,k} \leftarrow \omega_{m,k_1} . \omega_{m,k_2}$

      $\alpha \leftarrow$ *Derived from* (17)

      $k \leftarrow k + 1$

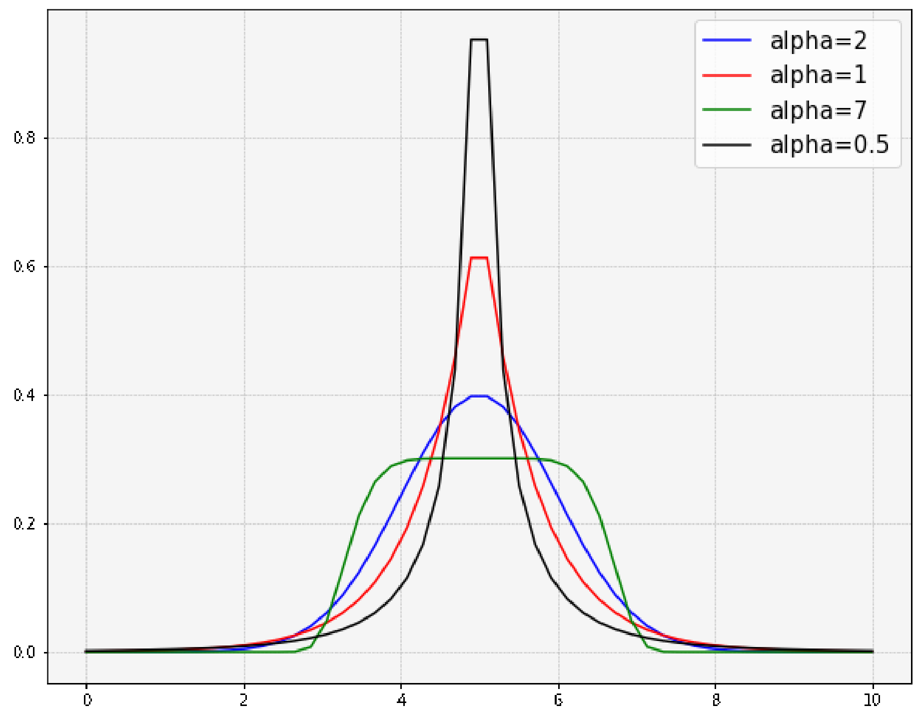


Figure 7: The probability density function of the generalized Gaussian distribution for different shape parameters where $m_1 = 5$ and $\sigma_1 = 1$.

Algorithm 2: Merge Two Gamma Models

Input : Two Gamma model parameters $\lambda_1(\alpha_1, \beta_1)$,$\lambda_2(\alpha_2, \beta_2)$

Output: Merged model $\lambda(\alpha, \beta)$

$M \leftarrow$ *total number of combination of components*;

for $m := 1$ *to* M **do**

  $n_1 \leftarrow$ *number of components in* $M_1(\lambda_1)$;

  $n_2 \leftarrow$ *number of components in* $M_2(\lambda_2)$;

  $k \leftarrow 0$ **for** $k_1 := 1$ *to* n_1 **do**

    for $k_2 := 1$ *to* n_2 **do**

      $\omega_{m,k} \leftarrow \omega_{m,k_1}.\omega_{m,k_2}$

      $\alpha \leftarrow$ *Derived from* (23)

      $\beta \leftarrow$ *Derived from* (23)

      $k \leftarrow k + 1$

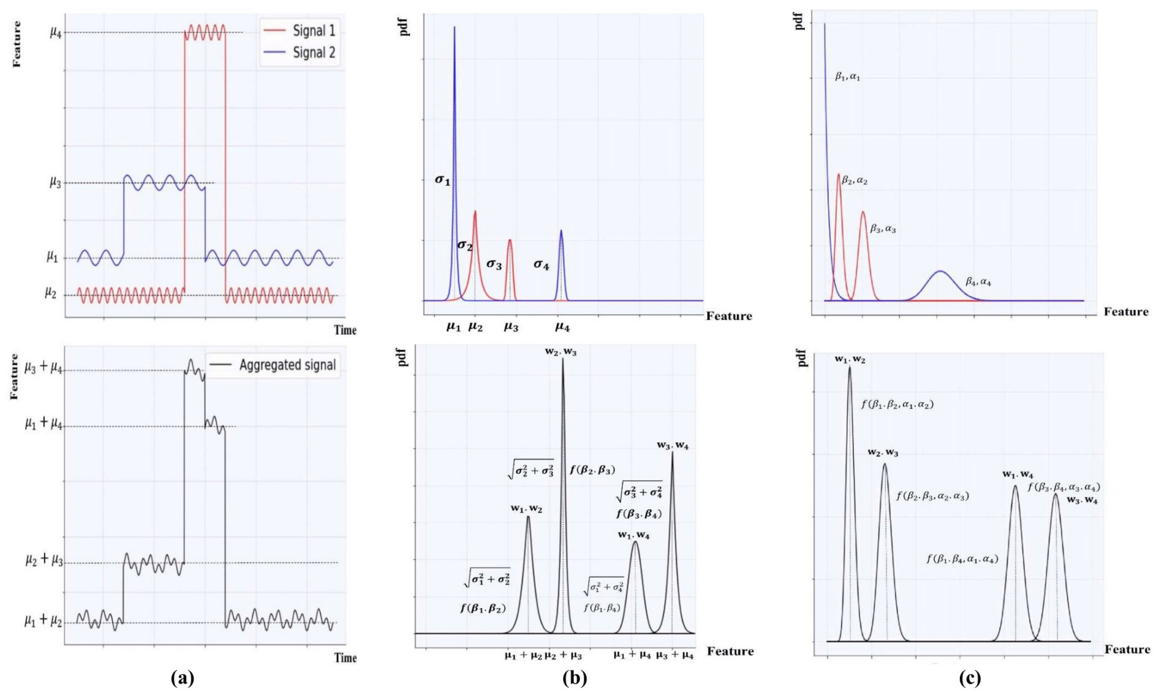


Figure 8: Synthetic example of the model merging for the classification of additive time series, (a) in the case of the generalized Gaussian mixture, (b) in the case of the Gamma mixture.

3.2.2 Testing phase

In testing stage, the dictionaries (based on generalized Gaussian parameters and Gamma parameters) obtained in training phase are used to predict the aggregated data. More specifically, each data point in the test set is evaluated using available combination features in the dictionary to predict which appliance combination generated the data point. To do so, the E-step from the EM algorithm is adopted to calculate the responsibility. The responsibility represents the probability of the n^{th}data point being generated from the k^{th} appliance combination:

$$r_n^k = \frac{w_k p(x_n \mid \theta_k)}{\sum_{i=1}^{M} w_i p(x_n \mid \theta_i)} \tag{24}$$

θ_k is the parameters of k^{th} combination model in dictionary and M is the total number of appliances combinations. Using the responsibility vector, we label the point with the combinations that maximizes its probability. In other words, to compare the probability of one data point belonging to one of the combinations, Bayes rule is used

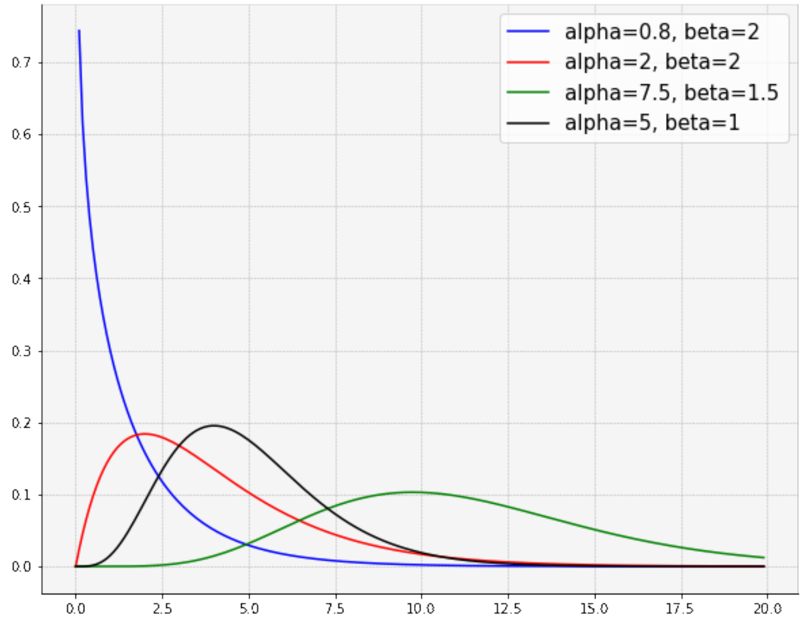


Figure 9: The probability density function of the Gamma distribution with different scale and shape parameters.

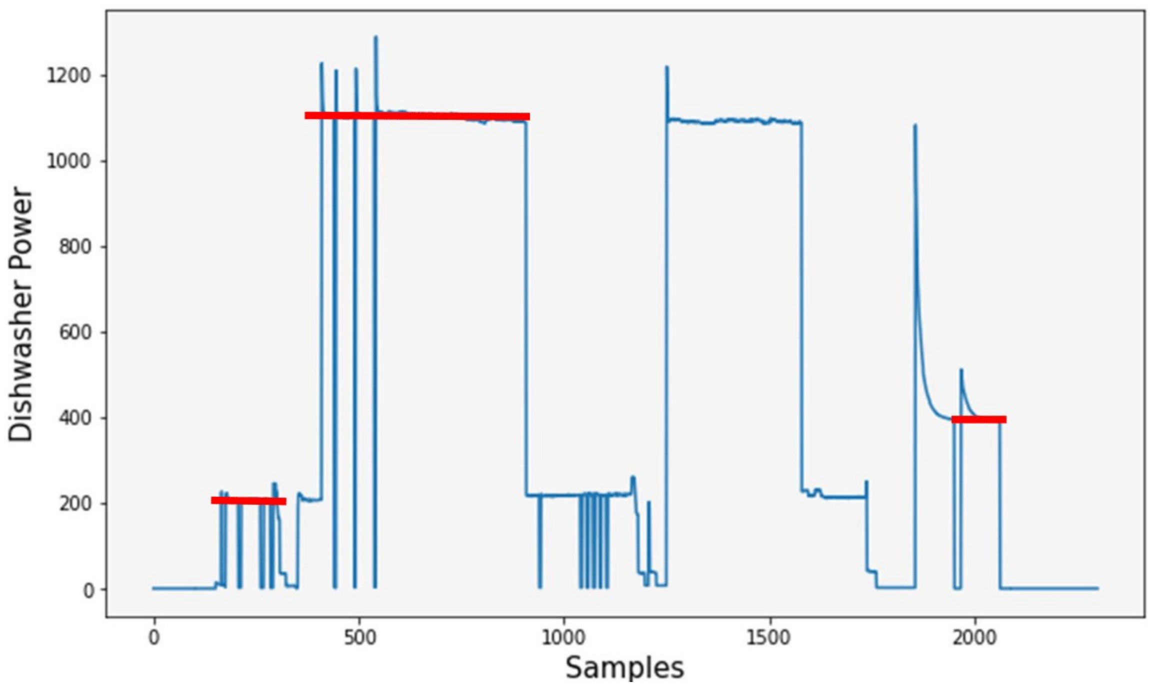


Figure 10: Three main states of power consumption of Dishwasher in house1 from REDD data set.

as an inference methodology:

$$\hat{X}_n^k = \arg\max_k r_n^k \tag{25}$$

Thus, for the data point x_n the label of $\hat{X}_n$ is chosen from all appliance labels.

3.2.3 Cascade Model

In this stage, a cascade DNN-Mixture model is introduced. The cascade model is more practical than the models developed in previous sections. This is due to the fact that although those models are accurate and flexible, their aggregation level is limited to two. By proposing this cascade model, we have the chance to detect the appliance state in more practical cases. The DNN is trained using a subset of the data in the training step. Meanwhile, the mixture models are created, followed by creating a dictionary of features combinations.

In the test section, the trained DNN is applied to the test input to achieve a preliminary version of appliance consumption from the total consumption. The final step is the same as section 3.2.2 where the appliance recognition is completed using the stored dictionary and posterior inference (24) . The block diagram of the hybrid cascade model is shown in Fig.11

The DNN that has been used is a convolutional neural network with two hidden layers consisting 8 Rectified linear hidden units and one Dense layer consisting 64 Rectified linear hidden units. We used one Dropout layer in order to prevent overfitting in this part of our hybrid model.

3.3 Summary

In this chapter we discussed finite mixture models and the superiority of generalized Gaussian mixture and Gamma mixture in statistical data modeling over Gaussian Mixture. The summation property of each mixture is analyzed to deploy the models in disaggregation task. Then we discussed the proposed model based on finite mixture models in NILM. The cascade model was introduced as a final step so that the model can be applied to a higher aggregation level.

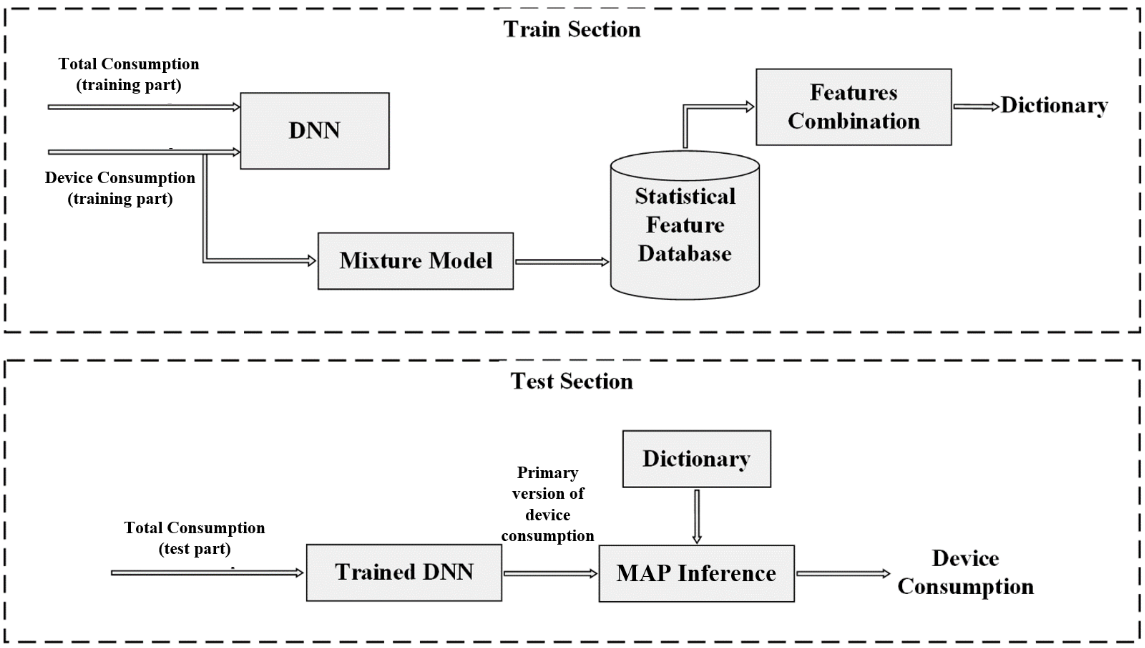


Figure 11: Cascade DNN-Mixture model.

Chapter 4

Experiments and Results

We start this chapter by explaining the evaluation metrics used in our experimental setup in Section 4.1. For the sake of verification, we applied experiments on three different datasets which are fully described in Section 4.2. Finally we explain the experimental setup comprehensively and discuss the results in 4.3.

4.1 Evaluation Metrics

In this setup, we use the following performance metrics: Accuracy, Precision, Recall, and F-score. These metrics are calculated from the resulting confusion matrices that provide the True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). TP represents the number of correct claims that the appliance was ON, FP shows the number of incorrect claims that a device was ON, FN denotes the number of times that appliance incorrectly claimed as being OFF while it was ON, and TN shows the number of correct claims that appliance was OFF. Accuracy is the total number of correct predictions (claims) with reference to the whole prediction. The main drawback of accuracy is that it cannot be descriptive for all kinds of appliances. For instance, in cases where the device is OFF most of the time, the whole prediction of the model could be zero, which results in a high accuracy metric. However, this is deceptive since the model did not predict at all. Due to this, we used other metrics in addition to accuracy in this study. Precision denotes the proportion of true ON identifications with reference to the total number of ON identifications made by the model, and Recall indicates the proportion of true ON predictions with reference to the total number of ON occurrences in the dataset. The F-score measure is the harmonic mean of Precision and Recall, which captures the salient features of both concerns and is widely used in many studies. F-score proves to be helpful where both Precision and Recall alone cannot provide sufficient descriptions of states in a dataset. For example, in some cases, the predictions of some models result in high Recall and low Precision or vice versa. This results in skewed identifications that the device is ON (correctly and/or incorrectly); F-score gives better information about the model performance in such situations.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{26}$$

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN} \tag{27}$$

$$Fscore = \frac{2.P.R}{P + R} \tag{28}$$

All of the above-mentioned metrics have been used in the following sections to elucidate and highlight the difference in capabilities of the implemented models.

4.2 Data sets

In the literature, different datasets have been used. For example, the dataset used in [2] and [53] contains two features: active and reactive power. Since, at present, in a large number of smart meters, the most common feature measured and recorded is the active power, the study has focused on active power alone. Three distinct datasets were chosen in our experiments based on their appropriateness and to cover as many scenarios as possible. The first one is the Almanac of Minutely Power dataset (AMPds) [65] which consists of measurements of consumption profiles of a single home in British Columbia, Canada for a period of one year and six months (from April 1, 2012 to March 31, 2013), at 1 minute sampling frequency. The measurements are circuit-based but the dataset included information on some specific loads from which we chose the Washer-Dryer, Water Heater Unit, Dishwasher and Fridge. The second dataset is the well known REDD dataset [59]. The REDD dataset consists of aggregate and circuit-level power profiles of six US households. The sampling frequency is 3 seconds, which is higher than usual for conventional smart meters in residential applications. The third dataset used in this work is from the data collected as part of a European project called Residential Monitoring to Decrease Energy Use and Carbon Emissions in Europe (REMODECE) [1]. This is a comparatively older database of residential consumptions, from Western, Central and Eastern European Countries. The part of the dataset used within this study is a sub-set called IRISE, dealing only with houses which were part of the project and located in France. It consists of the recordings of aggregated power for almost all electric appliances in the house at a sampling time of 10 minutes over a year. We included this dataset in order to examine the performance of the models on lower resolution measurements. We trained the models on 3 critical devices: Fridge, Water heater, and white appliances. Within this study, the devices classified as "White Appliances" are gamut cover, Dryer and Washing machine, which are not utilized heavily during the measurement period. In addition to that, due to the larger time-steps between measurements in this dataset, the data is quite sparse to consider them individually. As a result, these "white appliance" profiles have been aggregated together for the purpose of this study.

In this study, we have categorized houses in the data sets into two groups of "seen houses" and "unseen houses" for further evaluation. A "seen house" is defined as a house where the disaggregated data profiles are already available. In contrast, an "unseen house" is a house where its data-points are not labeled, and we use the aggregated data profiles for testing the model. The single house from AMPd (which is the only house in the dataset), house 1 from the REDD dataset, and house 28 from the IRISE dataset are used as the "seen houses" scenario for the cascade model evaluation. For the "unseen houses" scenario, we considered the following. For the REDD dataset, we used house 1 for training and house 3 for the testing, we selected these two houses because of the similarity of appliances. For the IRISE dataset, three houses (28, 36 and 48) were taken for training separately, while house 38 and house 14 were used for testing purpose.

4.3 Experiments and Results

One of the existing challenges in the NILM area is the lack of measured data for all the appliances in houses, where the only existing data is the total consumption of

[1] `https://remodece.isr.uc.pt/`

the house. To overcome this issue, developing robust models for disaggregating data, which can be applied to other houses, is necessary. As mentioned in section 2.3, one group of unsupervised approaches in NILM use labeled data from seen houses to build appliance models and apply the model to an unseen house to disaggregate the total measurement. Thus, in order to evaluate the applicability and reliability of proposed models on random unseen houses, we created a dictionary based on the data from specific houses in turn, and examined the models on unseen houses in the dataset by using the developed dictionary.

To examine the performance of the proposed models properly, we divided this section into three sub-sections to compare the obtained results of proposed mixture models with GMM on specific houses for all three datasets and to test our cascade model based on the methodology explained in 3.2.3 while comparing it with sate of art approaches.

4.3.1 Experiment 1 : Seen houses

In first experiment we compare the two proposed Finite Mixture Models with GMM model on the one house from each datasets. The single house from AMPd (which is the only house in the dataset) , house 1 from the REDD dataset, and house 28 from the IRISE dataset are used as the "seen houses" scenario.

Discussion

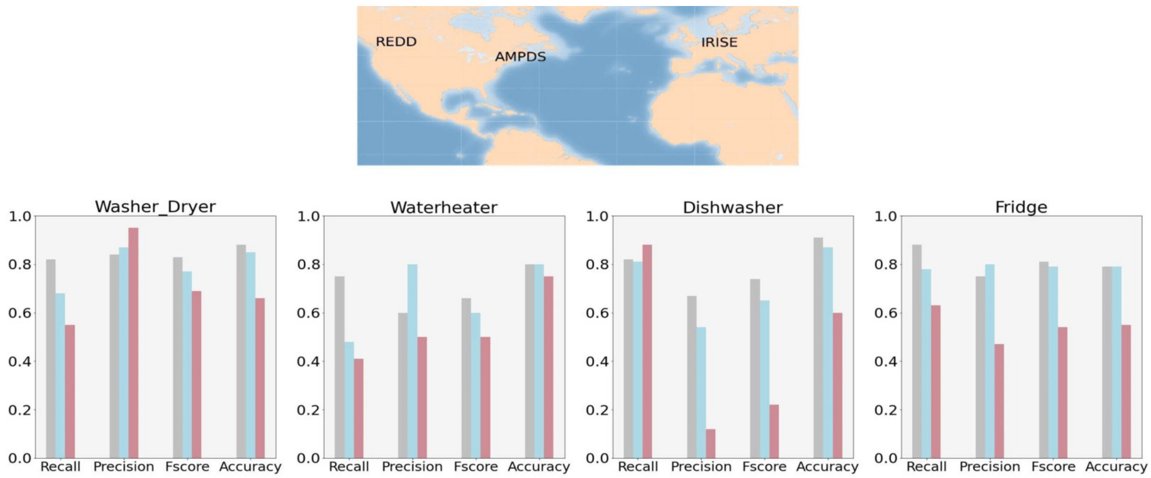
Figure12a demonstrates the results of selected appliances from the AMPds dataset. Upon looking at the F-score, one can see that the values for generalized GMM and Gamma models are 1.2-3.36 and 1.11-2.94 times higher than that of GMM. Similarly, in terms of accuracy, the values for Generalized GMM and Gamma models are 1.1-1.5 and 1.11-1.25 times higher than that of GMM, respectively. Based on this, we can see that Gamma and Generalized GMM models generally perform better than GMM for the AMPds dataset.

In the case of the Dishwasher, Generalized GMM significantly outperformed the other two approaches. The GMM does produces better results for the dishwasher when looking solely at the Recall metric, but suffers when it comes to Precision. Similarly, the GMM also performs better for Washer-Dryer when one focuses solely on Precision, but gives unsatisfactory results for Recall. This highlights the importance of studying the F-score along with the other metrics when it comes to assessing the performance of different models. This also highlights the unbalanced performance of the GMM. GMM seems to have a tendency to incorrectly predict that the device is ON most of the time (for the AMPds dataset). Both Generalized GMM and Gamma models do not exhibit this flaw since they are more accurate in approximating the density of data.

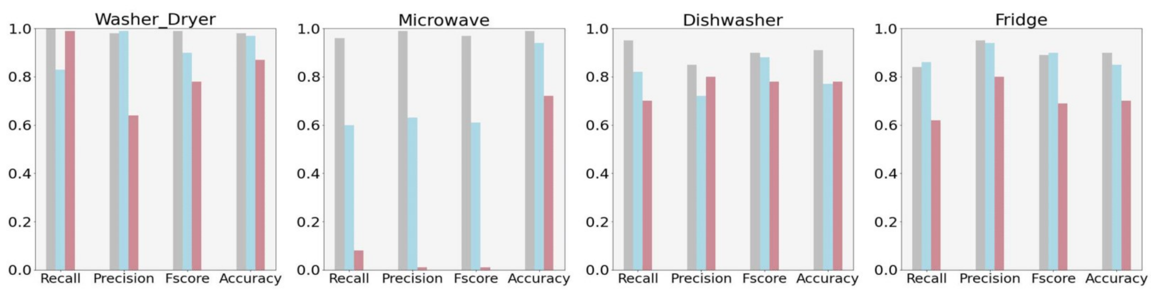
Analysis of the REDD dataset on the same lines as that done on the AMPds dataset reveals that for the REDD dataset, the Generalized GMM and Gamma model outperform GMM. This is especially true for the case of the Microwave. This was also elucidated clearly in the introduction using Figure6. The figure revealed the ability of Generalized GMM and Gamma mixture to more accurately approximate the densities for the microwave. On the other hand, GMM could not approximate the density which highlights the inability of GMM in adhering closely to the PDF of data and this inability leads to inaccurate results in the following steps. It must also be noted that for all appliances and across all metrics, Generalized GMM performs better with values higher than 0.8.

Moreover, comparison of Figure12a and Figure12b reveals that both Generalized GMM and Gamma model perform well and prove their flexibility when utilized for similar appliances such as Dishwasher, Washer-Dryer and Fridge for different datasets and from two distinct geographical locations. This leads us to conclude that the proposed modelling approaches are replicable across datasets and thus, may be applied across different time periods and regions.

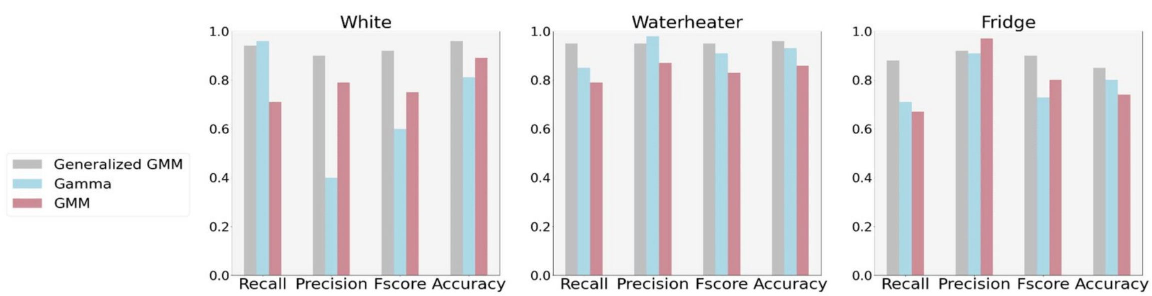
When it comes to the IRISE dataset results (shown in Figure12c), Generalized Gaussian Model outperforms the GMM on all devices, but Gamma model could not compete with other models. As mentioned in section 3.1.4, the approximation of the distribution of the sum of Gamma RDVs is valid when the scale parameters of two signals differ by no more than a factor of 10. For the categories in the IRISE dataset (especially for Fridge and White Appliances), this condition is seldom met. To explain in more detail, the scale parameters of the clusters for White appliances are $\beta_1 = \{16, 25, 20\}$ and for the Fridge are $\beta_2 = \{0.04, 0.2, 0.025\}$. The difference between β_1 and β_2 is off by more than a factor of 100, rather than the 10 preferred for good approximation for Gamma models. This results in sub-par Gamma mixture model performance for the IRISE dataset.



(a) Ampds House



(b) House 1 in REDD



(c) House 28 in IRISE

Figure 12: Results on seen houses from three datasets: a) AMPds, b) REDD and c) IRISE.

Experiment 2 : Unseen houses

In this sub-section, we explored the transferability of the introduced models on different houses in IRISE and REDD datasets. The workflow in this section is as follows:

- Three separate dictionaries based on houses 28, 36, and 48 are built in training phase.
- The dictionaries are applied in turn to a test set comprised of merged consumption data of house 14 and 38.

Discussion

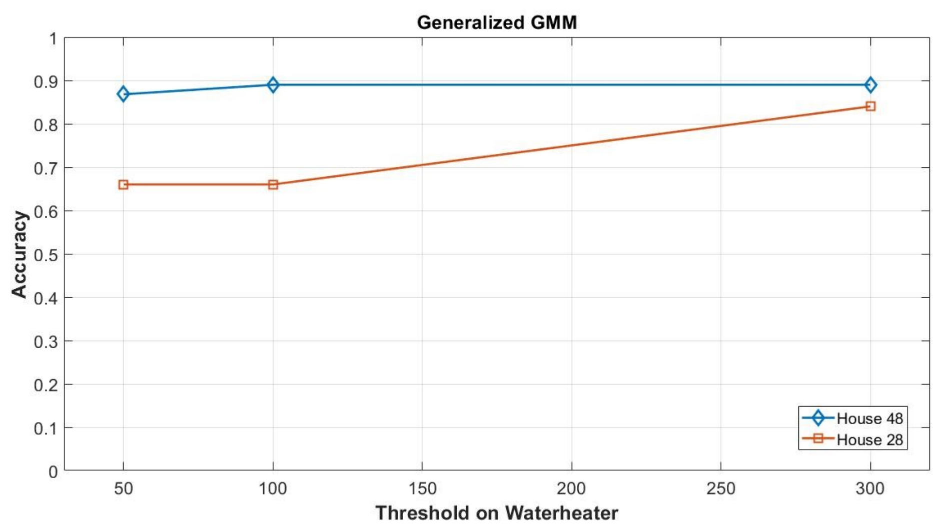
Table 1 shows the results of the above workflow. The results indicate that both generalized Gaussian mixture (GGM) and Gamma Mixture models show a very good performance in case of both houses 36 and 48. In these houses, the F-score of Waterheater and Fridge are minimum (0.74), while for the white appliances the value of F-score is over 0.5 which is acceptable for these types of appliances. On the other hand, the derived results from house 28 are below expectations especially for Waterheater. This different performance is due to the considered threshold as the ON state in the power consumption graph of appliance. For instance, to see the impact of threshold variation on generalized GMM model performance, the F-score and accuracy of the model for Waterheaters in houses 48 and 28 are compared in Figure13. According to this figure, both accuracy and F-score were almost consistence with regard to decreasing the threshold for house 48, while for house 28, it drops drastically. Specially for the F-score, with regard to Figure13 when we change the threshold of ON state from 300 watts to 100 watts F-score drops from 0.7 to 0.2. This reveals that the model based on the dictionary built on house 48 is more consistent to the decreasing threshold, while for house 28, it drops drastically. The main reason is the difference between Water heater power density.

To investigate the reason for weak results based on the dictionary from house 28 in more detail, in Figure14, the density of Water heater's consumption in this house is compared with the two unseen houses, 14 and 38, and one labeled house, House 48. There is a significant peak at 500 watts in Water heater power density in house 28, while in other houses, this peak occurs around 400 watts. This distinction denotes the

Table 1: Result of two proposed models on IRISE unseen house 38 and house 14 with dictionary based on three different houses

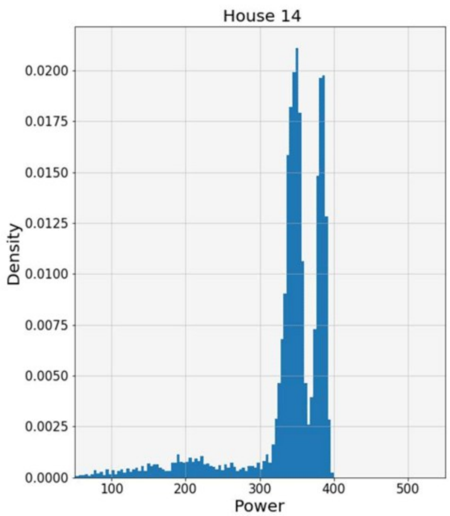
Dictionary based on house 48				
Generalized GMM	Recall	Precision	Fscore	Accuracy
Waterheater	0.71	0.90	0.80	0.86
Fridge	0.91	0.80	0.85	0.80
White	0.85	0.59	0.70	0.86
Gamma Mixture Model	Recall	Precision	Fscore	Accuracy
Waterheater	0.75	0.88	0.81	0.85
Fridge	0.76	0.82	0.79	0.74
White	0.92	0.47	0.62	0.79
Dictionary based on house 36				
Generalized GMM	Recall	Precision	Fscore	Accuracy
Waterheater	0.96	0.9	0.93	0.9
Fridge	0.88	0.93	0.9	0.88
White	0.7	0.58	0.69	0.86
Gamma Mixture Model	Recall	Precision	Fscore	Accuracy
Waterheater	0.62	0.9	0.73	0.71
Fridge	0.73	0.74	0.74	0.67
White	0.9	0.45	0.55	0.72
Dictionary based on house 28				
Generalized GMM	Recall	Precision	Fscore	Accuracy
Water heater	0.12	0.87	0.21	0.66
Fridge	0.87	0.75	0.8	0.73
White	0.91	0.34	0.49	0.67
Gamma Mixture Model	Recall	Precision	Fscore	Accuracy
Waterheater	0.27	0.82	0.4	0.62
Fridge	0.75	0.75	0.75	0.69
White	0.94	0.28	0.43	0.58

(a) F-score

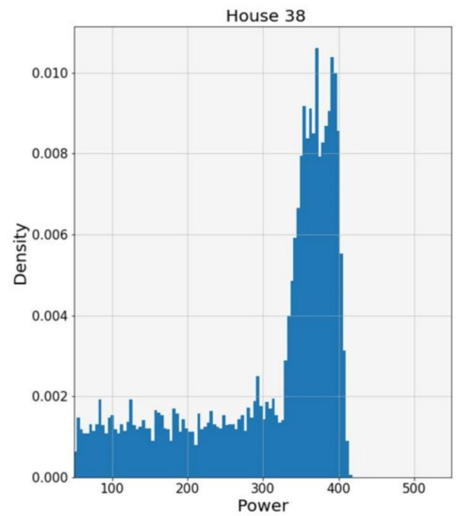


(b) Accuracy

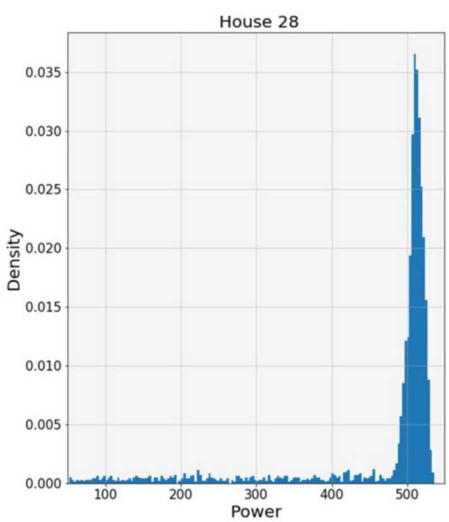
Figure 13: Performance of Generalized GMM model on test dataset with changing the threshold of ON state of Waterheater in two dictionaries based on house 48 and house 28



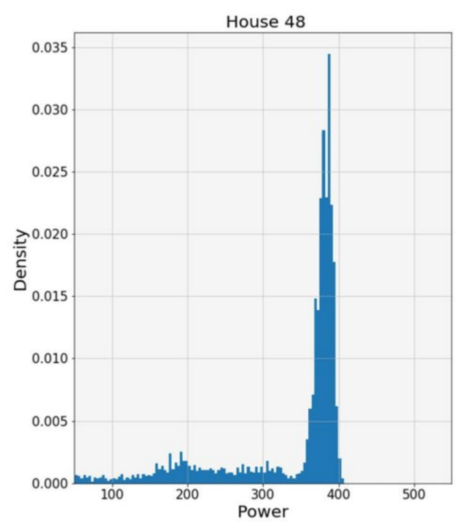
(a) Density of Waterheater in house 14



(b) Density of Waterheater in house 38



(c) Density of Waterheater in house 28



(d) Density of Waterheater in house 48

Figure 14: Density of Waterheaters' consumptions. There is a peak around 500 watts for house 28.

Table 2: Result on REDD unseen house3

Generalized GMM	Recall	Precision	Fscore	Accuracy
Dishwasher	0.99	0.78	0.81	0.97
Fridge	0.93	0.98	0.95	0.92
Microwave	0.93	0.91	0.92	0.99
Gamma Mixture Model	Recall	Precision	Fscore	Accuracy
Dishwasher	0.99	0.95	0.99	0.99
Fridge	0.97	0.99	0.97	0.98
Microwave	0.96	0.99	0.96	0.99

difference of the Water heaters, which may come from the different brands or the age of the device. Also, this difference could be due to variation in occupants' behavior. This fact explains the importance of similarity of devices between the tested house and the trained one used to build the dictionary. However, by increasing the threshold of ON state for Waterheater, the models based on house 28 can provide a promising results.

Within the same work follow, we trained the models on house 1 and tested the models on house 3 from REDD dataset. The results of the experiment are summarized in Table 2. Almost all metrics in both models are higher than 0.8 for all appliances. Consequently, tables I and II prove the applicability and reliability of both developed models for any unlabeled house in the same dataset. Therefore, proposed models have the potential to provide an extensible framework for building energy management.
To illustrate the extensibility of the proposed models better, a framework for implementing NILM with smart meter infrastructure could be introduced as follows: The NILM service provider maintains a comprehensive database of features (the so-called dictionary in previous sections) where the typical load profiles of many major common electric appliances and their combinations are centrally stored. The whole framework works based on detecting similarities in the appliances consumption of any household with those of stored in the database. This work is done through sending a feedback from an installed application by the owner to the NILM service provider. Next, the feedback is analyzed to find the most similar case among the available combinations in the database. Finally, the results of the analysis are sent back to the customer

along some suggestions about bow they can manage their consumption better.

4.3.2 Experiment 3 : Cascade models

In this part, the developed cascade models in section 3.2.3 were applied to compare its performance with other methods.

Table 3: Result on house1 AMPd dataset.

Dishwasher	Recall	Precision	Fscore	Accuracy
DNN-GGM	0.6	0.9	0.73	0.98
DNN-Gamma Mixture	0.54	0.94	0.7	0.98
RNN	0.72	0.87	0.79	0.9
FHMM	0.02	0.89	0.05	0.015
Fridge	Recall	Precision	Fscore	Accuracy
DNN-GGM	0.85	0.89	0.87	0.91
DNN-Gamma Mixture	0.85	0.89	0.87	0.91
RNN	0.9	0.83	0.89	0.96
FHMM	0.37	0.52	0.36	0.37
Washer-Dryer	Recall	Precision	Fscore	Accuracy
DNN-GGM	0.5	0.94	0.63	0.98
DNN-Gamma Mixture	0.64	0.75	0.7	0.88
RNN	0.9	0.88	0.8	0.96
FHMM	0.17	0.7	0.16	0.18
Waterheater	Recall	Precision	Fscore	Accuracy
DNN-GGM	0.43	0.73	0.54	0.91
DNN-Gamma Mixture	0.48	0.66	0.56	0.9
RNN	0.19	0.86	0.31	0.9
FHMM	0.21	0.65	0.28	0.16

Discussion

In Table 3 , Table 4 and Table 5 results of a single house in AMPds, REDD and Irise dataset are shown. Both cascade methods are showing almost identical results for all cases. The results in Table 3 and Table 4 elucidates that the outcomes of

Table 4: Result on house1 REDD dataset

Dishwasher	Recall	Precision	Fscore	Accuracy
DNN-GGM	0.89	0.96	0.95	0.92
DNN-Gamma Mixture	0.97	0.68	0.86	0.8
RNN	0.92	0.62	0.74	0.96
FHMM	0.38	0.42	0.43	0.34
Fridge	Recall	Precision	Fscore	Accuracy
DNN-GGM	0.91	0.86	0.88	0.94
DNN-Gamma Mixture	0.9	0.87	0.9	0.91
RNN	1	0.9	0.9	0.9
FHMM	0.73	0.23	0.35	0.36
Washer-Dryer	Recall	Precision	Fscore	Accuracy
DNN-GGM	0.67	0.96	0.79	0.91
DNN-Gamma Mixture	0.54	0.97	0.7	0.88
RNN	0.8	0.9	0.8	0.9
FHMM	0.088	0.059	0.07	0.8
Microwave	Recall	Precision	Fscore	Accuracy
DNN-GGM	0.86	0.77	0.82	0.9
DNN-Gamma Mixture	0.9	0.77	0.83	0.91
RNN	0.9	1	0.9	0.9
FHMM	0.002	0.0014	0.0018	0.96

Table 5: Result on IRISE house 28

Waterheater	Recall	Precision	Fscore	Accuracy
DNN-GGM	0.7	0.98	0.8	0.97
DNN-Gamma Mixture	0.75	0.98	0.85	0.97
RNN	0.89	0.96	0.92	0.9
FHMM	0.5	0.51	0.5	0.83
Fridge	Recall	Precision	Fscore	Accuracy
DNN-GGM	0.79	0.58	0.68	0.77
DNN-Gamma Mixture	0.74	0.6	0.66	0.78
RNN	0.69	0.76	0.73	0.8
FHMM	0.49	0.49	0.48	0.48
White	Recall	Precision	Fscore	Accuracy
DNN-GGM	0.6	0.69	0.64	0.9
DNN-Gamma Mixture	0.63	0.65	0.64	0.88
RNN	0.71	0.8	0.7	0.9
FHMM	0.49	0.49	0.49	0.95

the cascade models are very close to those of RNN, indicating their capability to compete with RNN. They also outperform the FHMM model for all appliances. In tables 3 and 4, it is shown that the developed cascade models reach an F-score higher than 0.7 for almost all devices except the Water Heater in AMPds. However, even for those instances, the cascade models generally outperform the RNN. Similarly, table 5 shows that the cascade models developed are capable of generating results comparable to RNN and better than FHMM. Almost all metrics for the proposed models are higher than 0.6. The results shows that the proposed cascade model can perform quite well even in case of White appliances which are the most controversial class of loads studied within this paper. All in all, it must be mentioned that although in most cases the proposed cascade model did not outperform the RNN, since they are Generative models, they provide more intuitive insights about the data, while RNN is discriminative and only classifies the data based on the features it learns every time. Besides, training RNN is usually a tough task which requires a considerable amount of time and it is prone to overfitting. Moreover, in case of long sequence, it becomes time consuming which is a very important issue in load disaggregation. In practice, the feeding sequence of data to the network is crucial in training, and it should include the complete cycle of the appliance consumption which is very long in some datasets. For instance, this sequence size for fridge in REDD dataset is 2400, hence it is extremely time consuming to train a RNN.

Chapter 5

Use-case discussion and Conclusion

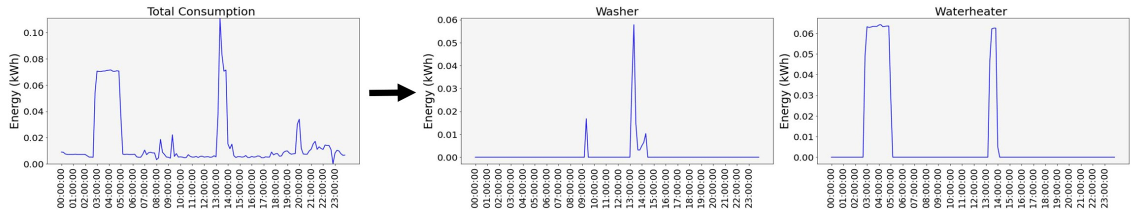
5.1 Disaggregation Use-case discussion

Monitoring systems are essential for determining the working conditions of a wide range of appliances used in homes industry, and commerce. In addition to reducing energy consumption, load monitoring techniques aim to eliminate unnecessary energy consumption by applying different approaches, such as appropriate timing of appliance use, optimization of their usages, and eliminating unnecessary activities. These purposes can be achieved by showing the inhabitants of a house the consumption of each appliance in the sum of the total billing to detect malfunction or excesses [66]. Disaggregating load can give consumers a more complete picture of how much energy they are using within their homes and where potential savings can be found. One of the end-user beneficiaries is providing useful information for consumers about what appliances they use and when, along with the associated energy usage and costs by itemized bills. Using this information they become aware of their top used appliances. Sending some actionable tips through messages can be effective for saving energy, and subsequently saving money. For example, advising to shift some high used appliances from peak hours period to off-peak hours.

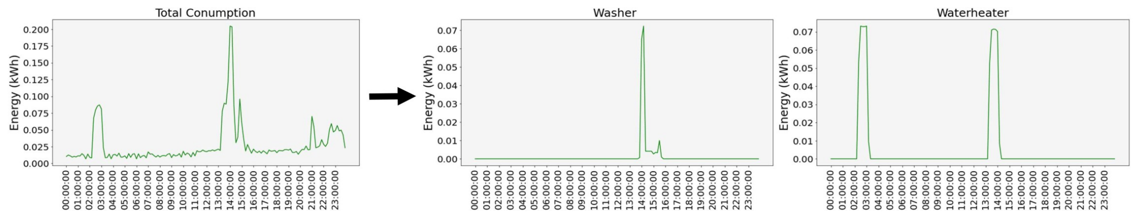
To illustrate the point, in figure 1.b the profile of house 34 from irise dataset is shown. Two high consumed appliances were ON at 14 pm which according to [1] is peak-hour since its between 6am-10pm.

Sending messages like : "You are using your washing machine at peak times.

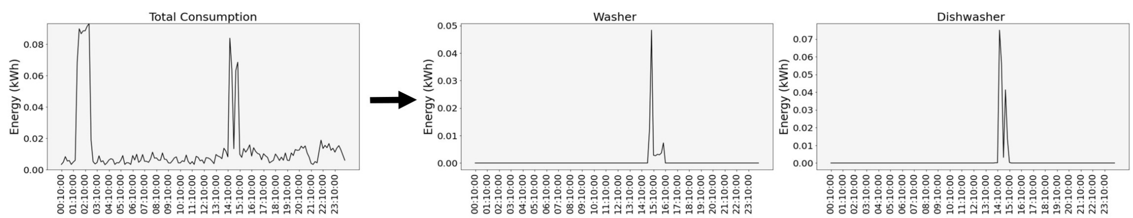
[1]https://www.french-property.com/guides/france/utilities/electricity/tariff



(a) House 14



(b) House 34



(c) House 28

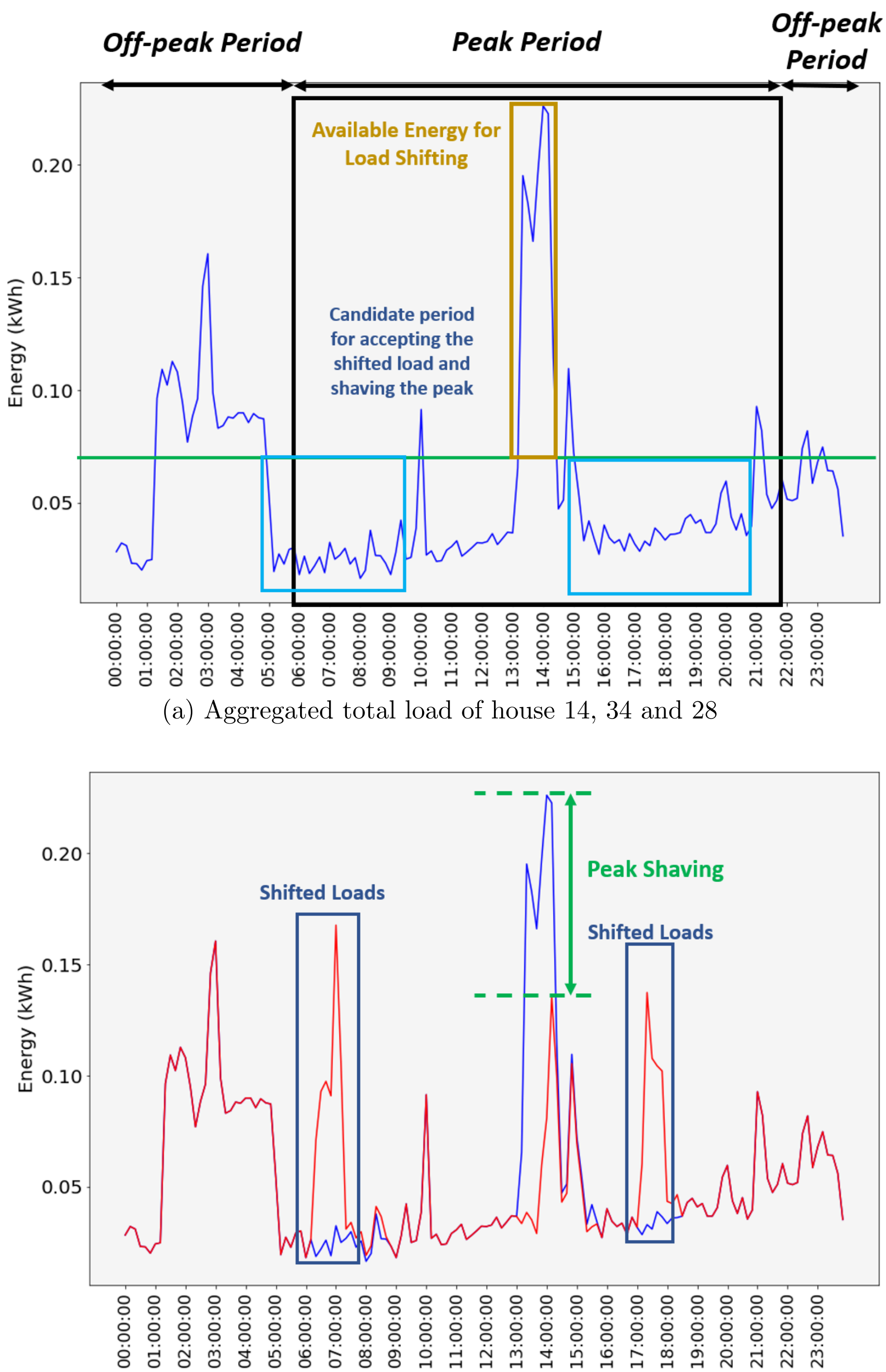
Figure 15: Total consumption of each house and the top used appliances in related time

Switch after 10pm to reduce your bill" would be helpful.

Furthermore, this strategy could be applied to large-scale clusters of houses to ease grid stress. For instance in Figure15, three different houses in same day have consumption spikes around 14pm which leads to a considerable peak in this time in grid. In this case, after finding which appliances caused the spikes for each house, suggesting the consumer shifts their use to lower stress hours in the grid, the overall profile would be more efficient (Figure16).

5.2 Conclusion

In this thesis, we proposed two dictionary-based probabilistic methods for appliance recognition in NILM operational framework. As an extension of GMM, we used generalized Gaussian mixture and applied the approximation of its summation property



(a) Aggregated total load of house 14, 34 and 28

(b) Minimizing the peak by shifting the Washer and Waterheater loads in house 14 to off-peak hours

Figure 16: Aggregated consumption of three houses at the same time (a) before , (b) after shifting spikes

in additive time series. Furthermore, we investigated the approximation of the PDF of power consumption of appliances using Gamma mixtures. Generally, the whole process of disaggregation consists of: 1) building dictionary by features of distribution acquired through different combinations and 2) predicting ON states of devices with the help of constructed dictionaries in aggregated data. Three datasets are chosen in the experiment part from different geographical locations and timestamps to examine the proposed models properly. Results demonstrated that our models outperform the GMM model, as we expected. To increase the aggregation level and compare with other methods, an extension that we called DNN-Mixtures, is proposed. The results show that the proposed model can compete with states of art techniques. Furthermore, another new aspect, the unseen scenario, is considered in this research. In two experiments on two datasets, the models are applied on the unseen houses with dictionary based on different houses, and following the analysis, it can be concluded that good results can support the idea of unsupervised approach in NILM.

5.3 Future Work

The work presented in this thesis can be improved in various ways. Some suggestions for future avenues of research are as follows:

- The proposed model is sensitive to the threshold of ON state in the devices. This threshold can be decreased and hence the model can be more resistant in the future work.
- By introducing prior information about general appliance types and occupancy information, the proposed approach can be integrated with a location-aware energy disaggregation framework.
- In section 3.1.4 it was mentioned that the summation methodology used for Gamma approximation is based on two conditions. The first one is that shape parameter should not be too small which is met for the datasets. The second, is that scale parameters of two variables differ by no more than a factor 10 which is not met in all appliance's combination and lead to less reliable results.

By using better methodology for sum of two Gamma distributed variables, the results be improved.

- The current method assumes that number of appliances in target location is supposed to be known which is not realistic. In practice, however, new appliances may be added to the house and the number of appliances may change. To formulate the problem, the data collected by the smart meter can be assumed to be consecutively sent to the learning system in batches with a fixed size and the dictionary can be updated through new data added to system.

- In this study, only active power was considered; by incorporating features such as reactive power and other steady-state features, dictionaries would become more comprehensive

Bibliography

[1] R. Gopinath, M. Kumar, C. P. C. Joshua, and K. Srinivas, "Energy management using non-intrusive load monitoring techniques-state-of-the-art and future research directions," *Sustainable Cities and Society*, p. 102 411, 2020.

[2] A. Ridi, C. Gisler, and J. Hennebert, "Aggregation procedure of gaussian mixture models for additive features," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, IEEE, 2016, pp. 2544–2549.

[3] M. S. Allili, "Wavelet modeling using finite mixtures of generalized gaussian distributions: Application to texture discrimination and retrieval," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1452–1464, 2011.

[4] T. Elguebaly and N. Bouguila, "Finite asymmetric generalized gaussian mixture models learning for infrared object detection," *Computer Vision and Image Understanding*, vol. 117, no. 12, pp. 1659–1671, 2013.

[5] M. S. Allili, N. Bouguila, and D. Ziou, "Finite general gaussian mixture modeling and application to image and video foreground segmentation," *Journal of Electronic Imaging*, vol. 17, no. 1, p. 013 005, 2008.

[6] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[7] J. Roos, I. Lane, E. C. Botha, and G. P. Hancke, "Using neural networks for non-intrusive monitoring of industrial electrical loads," in *Conference Proceedings. 10th Anniversary. IMTC/94. Advanced Technologies in I & M. 1994 IEEE Instrumentation and Measurement Technolgy Conference (Cat. No. 94CH3424-9)*, IEEE, 1994, pp. 1115–1118.

[8] K. D. Lee, "Electric load information system based on non-intrusive power monitoring," Ph.D. dissertation, Massachusetts Institute of Technology, 2003.

[9] C. Laughman, D. Lee, R. Cox, S. Shaw, S. Leeb, L. Norford, and P. Armstrong, "Advanced nonintrusive monitoring of electric loads," *IEEE Power and Energy Magazine*, pp. 56–63, 2003.

[10] S. Shaw, L. Norford, D. Luo, and S. Leeb, "Detection and diagnosis of hvac faults via electrical load monitoring," *HVAC&R Research*, vol. 8, no. 1, pp. 13–40, 2002.

[11] A. Faustine, N. H. Mvungi, S. Kaijage, and K. Michael, "A survey on non-intrusive load monitoring methodies and techniques for energy disaggregation problem," *arXiv preprint arXiv:1703.00785*, 2017.

[12] M. Sun, F. M. Nakoty, Q. Liu, X. Liu, Y. Yang, and T. Shen, "Non-intrusive load monitoring system framework and load disaggregation algorithms: A survey," in *2019 International Conference on Advanced Mechatronic Systems (ICAMechS)*, IEEE, 2019, pp. 284–288.

[13] K. S. Barsim and B. Yang, "Toward a semi-supervised non-intrusive load monitoring system for event-based energy disaggregation," in *2015 IEEE global conference on signal and information processing (GlobalSIP)*, IEEE, 2015, pp. 58–62.

[14] M. Z. A. Bhotto, S. Makonin, and I. V. Bajić, "Load disaggregation based on aided linear integer programming," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 7, pp. 792–796, 2016.

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

[16] V. Stankovic, J. Liao, and L. Stankovic, "A graph-based signal processing approach for low-rate energy disaggregation," in *2014 IEEE symposium on computational intelligence for engineering solutions (CIES)*, IEEE, 2014, pp. 81–87.

[17] K. He, L. Stankovic, J. Liao, and V. Stankovic, "Non-intrusive load disaggregation using graph signal processing," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 1739–1747, 2016.

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

[19] H. Altrabalsi, V. Stankovic, J. Liao, and L. Stankovic, "Low-complexity energy disaggregation using appliance load modelling," *Aims Energy*, vol. 4, no. 1, pp. 884–905, 2016.

[20] S. K. Barker, "Model-driven analytics of energy meter data in smart homes," 2014.

[21] J. Kolter, S. Batra, and A. Ng, "Energy disaggregation via discriminative sparse coding," *Advances in neural information processing systems*, vol. 23, pp. 1153–1161, 2010.

[22] M. Berges, E. Goldman, H. S. Matthews, and L. Soibelman, "Learning systems for electric consumption of buildings," in *Computing in Civil Engineering (2009)*, 2009, pp. 1–10.

[23] M. E. Berges, E. Goldman, H. S. Matthews, and L. Soibelman, "Enhancing electricity audits in residential buildings with nonintrusive load monitoring," *Journal of industrial ecology*, vol. 14, no. 5, pp. 844–858, 2010.

[24] M.-S. Tsai and Y.-H. Lin, "Modern development of an adaptive non-intrusive appliance load monitoring system in electricity energy conservation," *Applied Energy*, vol. 96, pp. 55–73, 2012.

[25] S. Makonin, F. Popowich, I. V. Bajić, B. Gill, and L. Bartram, "Exploiting hmm sparsity to perform online real-time nonintrusive load monitoring," *IEEE Transactions on smart grid*, vol. 7, no. 6, pp. 2575–2585, 2015.

[26] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Non-intrusive load monitoring using prior models of general appliance types," in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[27] H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han, "Unsupervised disaggregation of low frequency power measurements," in *Proceedings of the 2011 SIAM international conference on data mining*, SIAM, 2011, pp. 747–758.

[28] J. Z. Kolter and T. Jaakkola, “Approximate inference in additive factorial hmms with application to energy disaggregation,” in *Artificial intelligence and statistics*, PMLR, 2012, pp. 1472–1482.

[29] M. J. Johnson and A. S. Willsky, “Bayesian nonparametric hidden semi-markov models,” 2013.

[30] M. Zhong, N. Goddard, and C. Sutton, “Signal aggregate constraints in additive factorial hmms, with application to energy disaggregation,” *Advances in Neural Information Processing Systems*, vol. 27, pp. 3590–3598, 2014.

[31] F. Paradiso, F. Paganelli, D. Giuli, and S. Capobianco, “Context-based energy disaggregation in smart homes,” *Future Internet*, vol. 8, no. 1, p. 4, 2016.

[32] M. Aiad and P. H. Lee, “Unsupervised approach for load disaggregation with devices interactions,” *Energy and Buildings*, vol. 116, pp. 96–103, 2016.

[33] D. Kelly, *Disaggregation of domestic smart meter energy data*, `https://spiral.imperial.ac.uk/handle/10044/1/49452`, 2016.

[34] N. Batra, A. Singh, and K. Whitehouse, “Neighbourhood nilm: A big-data approach to household energy disaggregation,” *arXiv preprint arXiv:1511.02900*, 2015.

[35] B. Wild, K. S. Barsim, and B. Yang, “A new unsupervised event detector for non-intrusive load monitoring,” in *2015 IEEE global conference on signal and information processing (GlobalSIP)*, IEEE, 2015, pp. 73–77.

[36] H. Gonçalves, A. Ocneanu, M. Bergés, and R. Fan, “Unsupervised disaggregation of appliances using aggregated consumption data,” in *The 1st KDD workshop on data mining applications in sustainability (SustKDD)*, 2011.

[37] D. Egarter and W. Elmenreich, “Autonomous load disaggregation approach based on active power measurements,” in *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, IEEE, 2015, pp. 293–298.

[38] B. Zhao, L. Stankovic, and V. Stankovic, “On a training-less solution for non-intrusive appliance load monitoring using graph signal processing,” *IEEE Access*, vol. 4, pp. 1784–1799, 2016.

[39] L. Mauch and B. Yang, "A new approach for supervised power disaggregation by using a deep recurrent lstm network," in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, IEEE, 2015, pp. 63–67.

[40] W. He and Y. Chai, "An empirical study on energy disaggregation via deep learning," in *2016 2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE 2016)*, Atlantis Press, 2016, pp. 338–342.

[41] M. Kaselimi, N. Doulamis, A. Voulodimos, E. Protopapadakis, and A. Doulamis, "Context aware energy disaggregation using adaptive bidirectional lstm models," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3054–3067, 2020.

[42] T. Sirojan, B. T. Phung, and E. Ambikairajah, "Deep neural network based energy disaggregation," in *2018 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*, IEEE, 2018, pp. 73–77.

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

[44] C. Zhang, M. Zhong, Z. Wang, N. Goddard, and C. Sutton, "Sequence-to-point learning with neural networks for non-intrusive load monitoring," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[45] J. Kelly and W. Knottenbelt, "Neural nilm: Deep neural networks applied to energy disaggregation," in *Proceedings of the 2nd ACM international conference on embedded systems for energy-efficient built environments*, 2015, pp. 55–64.

[46] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[47] M. Kaselimi, E. Protopapadakis, A. Voulodimos, N. Doulamis, and A. Doulamis, "Multi-channel recurrent convolutional neural networks for energy disaggregation," *IEEE Access*, vol. 7, pp. 81 047–81 056, 2019.

[48] R. Jia, Y. Gao, and C. J. Spanos, "A fully unsupervised non-intrusive load monitoring framework," in *2015 IEEE international conference on smart grid communications (SmartGridComm)*, IEEE, 2015, pp. 872–878.

[49] L. Mauch, K. S. Barsim, and B. Yang, "How well can hmm model load signals," in *Proceeding of the 3rd international workshop on non-intrusive load monitoring (NILM 2016)*, 2016.

[50] N. Bouguila and W. Fan, *Mixture Models and Applications*. Springer, 2020.

[51] X. Zhuang, Y. Huang, K. Palaniappan, and Y. Zhao, "Gaussian mixture density modeling, decomposition, and applications," *IEEE Transactions on Image Processing*, vol. 5, no. 9, pp. 1293–1302, 1996.

[52] R. Singh, B. C. Pal, and R. A. Jabr, "Statistical representation of distribution system loads using gaussian mixture model," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 29–37, 2009.

[53] T. Rastogi, *Load identification from aggregated data using generative modeling*, `http://www.diva-portal.org/smash/get/diva2:1304677/FULLTEXT01.pdf`, 2019.

[54] A. Ridi, C. Gisler, and J. Hennebert, "Acs-f2—a new database of appliance consumption signatures," in *2014 6th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, IEEE, 2014, pp. 145–150.

[55] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[56] D. K. Dey, L. Kuo, and S. K. Sahu, "A bayesian predictive approach to determining the number of components in a mixture distribution," *Statistics and Computing*, vol. 5, no. 4, pp. 297–305, 1995.

[57] A. R. Webb, "Gamma mixture models for target recognition," *Pattern Recognition*, vol. 33, no. 12, pp. 2045–2054, 2000.

[58] Q. Zhao, H.-w. Li, and Y.-t. Shen, "On the sum of generalized gaussian random signals," in *Proceedings 7th International Conference on Signal Processing, 2004. Proceedings. ICSP'04*, IEEE, vol. 1, 2004, pp. 50–53.

[59] J. Z. Kolter and M. J. Johnson, "Redd: A public data set for energy disaggregation research," in *Workshop on data mining applications in sustainability (SIGKDD), San Diego, CA*, vol. 25, 2011, pp. 59–62.

[60] H. Soury and M.-S. Alouini, “New results on the sum of two generalized gaussian random variables,” in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, IEEE, 2015, pp. 1017–1021.

[61] D. Ziou and N. Bouguila, “Unsupervised learning of a finite gamma mixture using MML: application to SAR image analysis,” in *17th International Conference on Pattern Recognition, ICPR 2004, Cambridge, UK, August 23-26, 2004*, IEEE Computer Society, 2004, pp. 68–71.

[62] F. R. Al-Osaimi and N. Bouguila, “A finite gamma mixture model-based discriminative learning frameworks,” in *14th IEEE International Conference on Machine Learning and Applications, ICMLA 2015, Miami, FL, USA, December 9-11, 2015*, T. Li, L. A. Kurgan, V. Palade, R. Goebel, A. Holzinger, K. Verspoor, and M. A. Wani, Eds., IEEE, 2015, pp. 819–824.

[63] N. Bouguila, K. Almakadmeh, and S. Boutemedjet, “A finite mixture model for simultaneous high-dimensional clustering, localized feature selection and outlier rejection,” *Expert Syst. Appl.*, vol. 39, no. 7, pp. 6641–6656, 2012.

[64] T. Stewart, L. Strijbosch, H. Moors, and P. v. Batenburg, *A simple approximation to the convolution of gamma distributions*, `http://dx.doi.org/10.2139/ssrn.900109`, 2007.

[65] S. Makonin, F. Popowich, L. Bartram, B. Gill, and I. V. Bajić, “Ampds: A public dataset for load disaggregation and eco-feedback research,” in *2013 IEEE electrical power & energy conference*, IEEE, 2013, pp. 1–6.

[66] S. Darby *et al.*, “The effectiveness of feedback on energy consumption,” *A Review for DEFRA of the Literature on Metering, Billing and direct Displays*, vol. 486, no. 2006, p. 26, 2006.