Deep Recurrent Neural Network to Disaggregate Household Energy Consumption

by

Nguyen Viet Linh

Submitted to the Department of Electrical Engineering, Electronics, Computers and Systems
in partial fulfillment of the requirements for the degree of

Erasmus Mundus Master Course in Sustainable Transportation and Electrical Power Systems

at the

UNIVERSIDAD DE OVIEDO

September 2017

Author . . . .

Certified by . . . .
PABLO ARBOLEYA ARBOLEYA
Associate Professor
Thesis Supervisor

Deep Recurrent Neural Network to Disaggregate Household Energy Consumption

by

Nguyen Viet Linh

Submitted to the Department of Electrical Engineering, Electronics, Computers and Systems
on September 11, 2017, in partial fulfillment of the
requirements for the degree of
Erasmus Mundus Master Course in Sustainable Transportation and Electrical Power Systems

Abstract

Non-intrusive Load Monitoring (NILM) is a technique which accepts the total consumption in a house as an input and computes the estimated demand of individual appliances in that house. All NILM needs is one single meter to record the total power of a house. With NILM, the users can assess the information of appliances without the intrusion of measurement devices. Such information could help users adapt their energy-usage habit for better saving and facilitate the grid management. This motivates our effort to research the NILM topic, which focuses on disaggregation algorithm in this thesis. The main goals of this thesis are to build a web interface for the energy usage feedback, to enhance the disaggregation algorithms as well as to investigate the non-intrusive technique in fault study. The first part of this work reviews the NILM background and states whether NILM is helpful to alleviate the climate change. The second part describes various sources of dataset and selects one for visualizing in the web interface. This is the first attempt to build an online energy feedback platform for the end-users. Our third effort is to advance the concept for disaggregation algorithm, which is the deep recurrent neural network (DRNN). The results showed the better performance achieved by DRNN, compared with the classic optimization. DRNN has also done a fair job to test unseen appliances. Finally, the application of non-intrusive technique to fault study has been initially studied. Our discussion has found the feasibility of non-intrusive fault detection with satisfactory outcomes.

Thesis Supervisor: PABLO ARBOLEYA ARBOLEYA
Title: Associate Professor

Acknowledgments

First, I would like to thank my family for their financial and moral support, as well as for mastering Messenger/Skype to allow me to keep in touch while I have been in Europe.

I would like to express my appreciation gratitude to Dr. Arboleya for his unconditional support and tireless advice for my work, without which this thesis would have not been possible.

I would also like to extend my appreciation to my program coordinators, Dr. Pablo Garcia Fernandez and Dr. Jorge Garcia for their contribution and ultimate support for my thesis and all along my master program.

I would never forget to pay gratitude to my professors from ISEC Coimbra and University of Nottingham for providing me basics and advance knowledge to develop this research.

I would also thank for Dr. Hsueh-Hsien Chang for co-working with me in the topic of non-intrusive fault monitoring.

Last but not least, some friends I know during the program (too long to shortlist here, but they surely know), without whom this thesis could be done earlier. Completing the program feels like an adventure, and I am pleased we could enjoy together. “A tu manera, descomplicado En una bici que te lleve a todos lados” (Carlos, Shakira *et. al.* 2016)

Glossary

NILM	Non-intrusive Load Monitoring.
DAQ	Data Acquisition.
DWT	Discrete Wavelet Transform.
GA	Genetic Algorithm.
ANN	Artificial Neural Network.
BP	Backpropagation.
FNN	Feedforward Neural Network.
MFNN	Multilayer Feedforward Neural Network.
RNN	Recurrent Neural Network.
DRNN	Deep Recurrent Neural Network.
k-NN	k-Nearest Neighbors.
CO	Combinatorial Optimization.
REDD	Reference Energy Disaggregation Dataset.
ECO	Electricity Consumption and Occupancy.
UK-DALE	UK Domestic Appliance-Level Electricity.
ECApA	Energy Correctly Assigned per Appliance
SLGF	Single-line-to-ground Fault.
LLGF	Line-to-line Fault.
DLGF	Double-line-to-ground Fault.
WMRA	Wavelet Multi-resolution Analysis.
ESE	Electrical Service Entry.
MDMS	Meter Data Management System.
NIFM	Non-intrusive Fault Monitoring.
UL	Underwriters Laboratories Inc.
AC	Alternative Current.
DWT	Discrete Wavelet Transform.
FC	Fault Criterion.
SM	Statistical Moment.
SK	Skewness and Kurtosis.
IM	Induction Machine.
SVM	Support Vector Machine.
HEAL	Heuristic and Evolutionary Algorithms Laboratory.
AI	Artificial Intelligence.
EMTP	Electromagnetic Transient Program.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Non-intrusive Load Monitoring

Intrusive power monitoring has been used as the traditional approach in order to measure the power demand of appliances for several years. To record the energy consumption of each appliance, sensors are required to be installed at each load point. This method, however, incurs the huge cost for installation and maintenance. The large number of installed sensors also distorts the energy waveforms since those sensors consume an amount of energy.

To alleviate the above-mentioned issues, non-intrusive load monitoring (NILM) was initially introduced by G. W. Hart [1]. This method is an innovation to estimate the demand of each home appliance from a single sensor or meter which only records the aggregated power. This future load monitoring system can help users manage their energy-spending habits, or detect faulty/aging appliances with the minimum number of instruments. Therefore, the NILM platform demonstrates a future potential for the next generation of electrical networks. Fig. 1-1 depicts the differences between intrusive and non-intrusive monitoring.

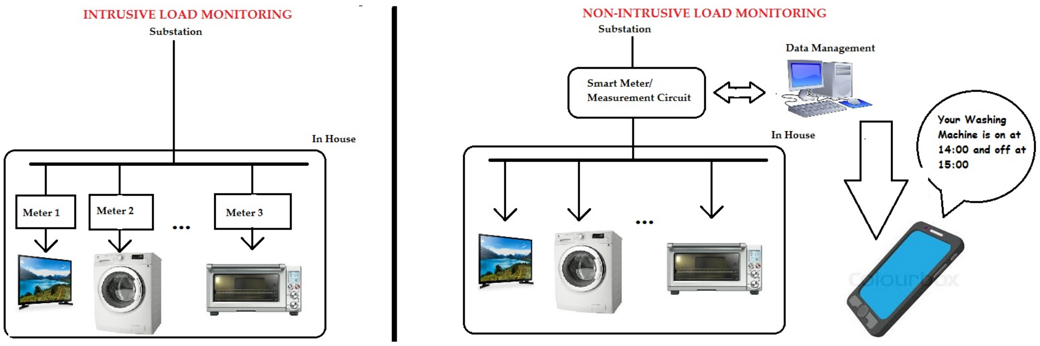

INTRUSIVE LOAD MONITORING
Substation
In House
Meter 1
Meter 2
...
Meter 3
NON-INTRUSIVE LOAD MONITORING
Substation
Smart Meter/ Measurement Circuit
Data Management
In House
...
Your Washing Machine is on at 14:00 and off at 15:00

Figure 1-1: Intrusive vs. Non-intrusive load monitoring.

1.2 NILM in business

Established in 2008, PlotWatt[1] is one of the pioneers conducting the NILM business. The company has cooperated with utilities and chain restaurants to provide their energy disaggregation services. As customers of PlotWatt, the users have to set up their own existing meter, and the real-time consumption data are compiled in the PlotWatt server. Then, the feedback of usage is daily updated via the web application and user's account inbox.

Similar to PlotWatt, Bigdely[2], a California-based startup, offers the services for energy data mining at the appliance level with smart meters. Although founded three years later than PlotWatt, Bidgely has successfully accomplished the disaggregation technology for widespread usages. At the utility level, Bigdely has recently developed the solution for demand side management (DSM). The software-as-services tool of Bidgely allows the utility companies to understand and quickly inform their end-users to adjust the energy use during the peak-load periods. Those solutions are done with only one single meter at the electrical service entry.

There are also several other companies who transact the same business (Smappee[3] or Neurio[4]). Their general approach is to interpret the prior information of similar

[1] PlotWatt Homepage: https://www.plotwatt.com/
[2] Bidgely Homepage: https://www.bidgely.com/
[3] Smappee Homepage: http://www.smappee.com/us/home
[4] Neurio Homepage: https://www.neur.io/

houses, and then discover the correlation between energy usage habits and weather condition as well as household features.

1.3 NILM Challenges

The biggest challenge of NILM is to ensure the high accuracy with the low sampling rate of the data acquisition (DAQ) meter (0.1 - 100 Hz), near real time. The high-frequency sampling DAQ, nevertheless, is too expensive for the common use. With the limited sampling rate, several appliances which have low-level consumption could not be recognized.

Furthermore, real-time feedback is computationally burdensome and easily incorrect due to the transient starting of appliances. As far as the author is aware, there is no study comparing the online feedback to the offline disaggregation. In addition, the energy information should be sufficient for accurate identification. This, as a result, leads to the computational complexity and the burden on the hardware technology. At least, those hardware technologies should be required to obtain the NILM data:

- Power sockets or sensors which measure the energy consumption.
- A communication network to transfer data from appliances to Internet.

Also, the efficiency of algorithms varies with different case studies. Each disaggregation method has its own advantages and disadvantages depending on the datasets. Therefore, several algorithms should be simultaneously implemented in the web server to enhance the disaggregation performance. This complicates the website database and it is still uncertain that how many algorithms should be involved. Those disaggregation methods are discussed in Chapter 2.

1.4 Objectives and Contributions

The aim of this work is to improve the accuracy of the disaggregation technology for the low-sampling frequency energy data. The ultimate goal is to give consumers feedback on efficient energy use and partially mitigate the climate change.

To illustrate, using alternative energy resources and demand management are the common solutions to the global warming problem. Investigating to renewable resources is expensive and time-consuming; while controlling demand is more effective and simple. In some areas where the electricity price is time-varying, the NILM application can give the user feedback if he/she should turn on the washing machine at the moment or two hours later for efficiency saving.

Furthermore, if consumers can access the information of detailed logs when appliances are in use, those logs can be used to check for faulty/aged loads. The NILM application can also tell you whether your refrigerator becomes outdated since the defected refrigerator tends to defrost (on/off) more frequently and consume more kWh than the normal one. Nonetheless, the demand management requires the strong users' awareness and consideration in their power consumption associated with electricity bills.

At the higher level, the capture of energy disaggregation can help the grid operators predict the future demand based on the consumers habit, as mentioned above. Sometimes, extra generation and transmission capacity should be expanded to satisfy the peak even though the average daily consumption does not significantly change. As a consequence, the network can oversupply the loads during the day on which the demand curve is flat. Based on the NILM forecasting, the network operator can manage the demand side response (DSR) by load shifting and hence lowering the operation cost while the average consumption remains. Another example where NILM can facilitate DSR is that the NILM website sends the messages to clients during the hot summer day: "We are having the transmission congestion right now. Please turn off your air conditioner for 15 minutes to lower the current peak". With the NILM platform, utilities and consumers are always available in communication instead of the traditional calling center services. This requires more consumer engagement which is one of the main characteristics of the smart grid.

1.5 Thesis Outline

The remainder of this thesis is organized as follows:

- Chapter 2 reviews NILM technology in academic papers.
- Chapter 3 describes a various range of public dataset for NILM research and the visualization in a web page.
- Chapter 4 explains the disaggregation technique proposed in this thesis.
- Chapter 5 discusses the results of the proposed method.
- Chapter 6 mentions application of energy disaggregation to fault study.
- Chapter 7 concludes the thesis work and recommends the future scope.

1.6 Publication

A part of this thesis was published in *Energies*, in cooperation with Dr. Hsueh-Hsien Chang from JinWen University of Science and Technology [2]. In that paper, we examined the feasibility of non-intrusive fault diagnosis in a distribution network.

1. Chang, H.-H.; Linh, N.V. Statistical Feature Extraction for Fault Locations in Nonintrusive Fault Detection of Low Voltage Distribution Systems. *Energies* **2017**, *10*, 611.

Chapter 2

Literature Review

This chapter reviews the state-of-the-art techniques applied to NILM. Two commonly studied techniques are feature extraction and disaggregation algorithm.

2.1 NILM Technical Background

Technically, a basic NILM framework consists of three important stages: data acquisition (DAQ), feature extraction, and pattern recognition. Fig. 2-1 describes those basic steps for a NILM platform.

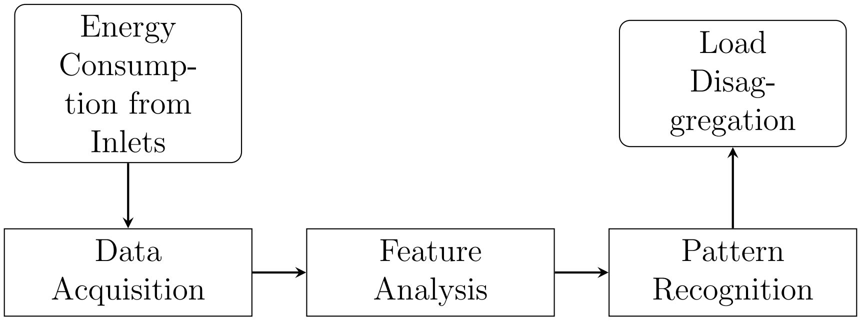


Figure 2-1: A schematic workflow of a NILM system.

Firstly, the energy data are acquired from the power inlets via a DAQ card.

Then, those datasets are mined and analyzed based on the feature extraction technique. The objective of this step is to reduce the dimensionality of the original dataset and/or smooth the raw waveforms. Machine learning technique finally compares those extracted features with various appliance patterns in the database.

For 25 years, NILM has been the interesting topic for computer and electrical engineering scientists. Fig. 2-2 shows a massive rise in the number of NILM research papers since 2010 [3]. Laughman *et al*. [4] as well as Zeifman and Roth [5] reviewed the background and content of the NILM literature. .

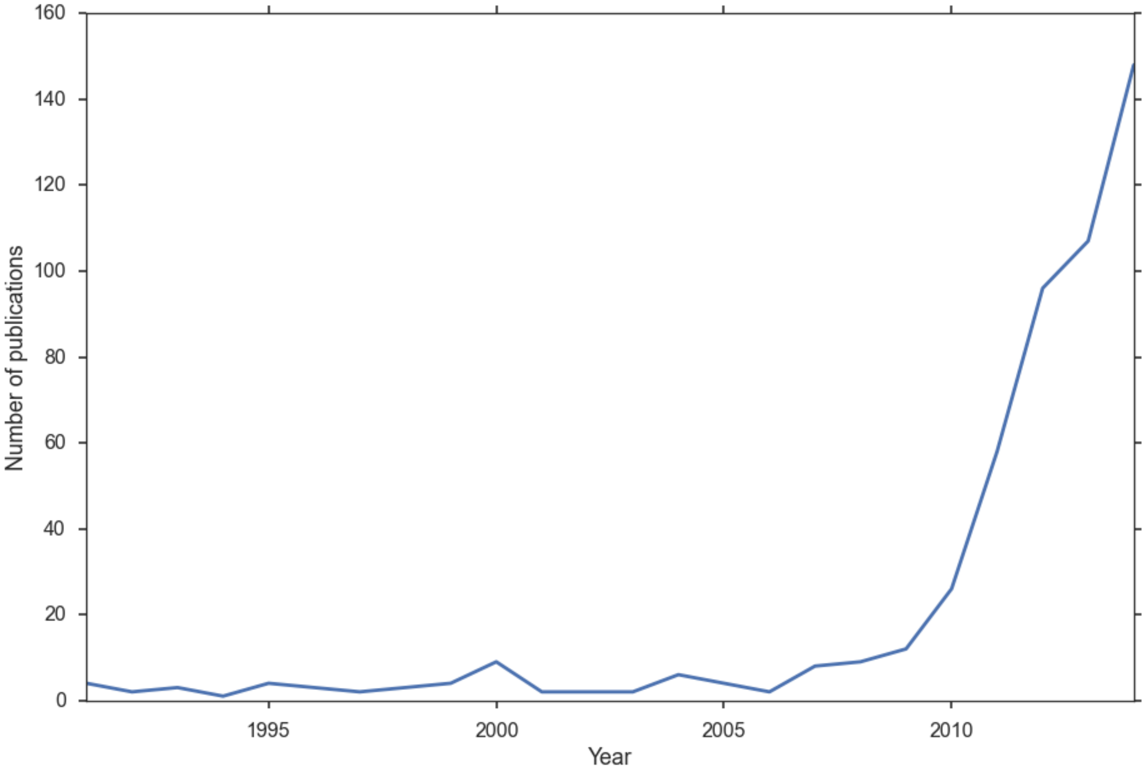


Figure 2-2: Number of published papers in NILM per year [3] (With permission)

2.2 Feature Extraction Techniques

In term of feature extraction, the commonly used signatures can be grouped into steady-state features and transient features.

2.2.1 Steady-State Features

The steady-state features involve active power (P), reactive power (Q), harmonics, voltage-current (V-I) trajectory and admittance waveform (AW). The AW, however, is not widely considered due to the limitation of zero division. The original paper of G. W. Hart [1] identified the home appliances based on the P-Q plane. In [6], the authors also used the P-Q-based method for the extraction. Nevertheless, this approach may not detect different appliances with the same consumption. Akbar and Khan [7] employed the current harmonic features, but several linear loads may not be detected due to the low level of harmonics. Recently, the V-I trajectory has been the innovation of feature extraction in NILM, as discussed in [8]. In this scheme, the instantaneous voltage and current are respectively measured and normalized based on their peaks. Then, the V-I plot traces different shapes corresponding to different appliances. This V-I trajectory-based extraction requires the intensive computation due to the two-dimensional matrix of V-I and complicated wave-shapes. Also, the users and companies may discourage the over-investment in smart meters to retain the instantaneous information of both V and I.

In general, the main drawback of using steady-state features is the insufficient information of the load behavior. Different appliances may have similar patterns in the steady-state operations, which may lead to the mistaken identification.

2.2.2 Transient Features

The typical transient features are starting current or switching transient waveform. Several academic papers have used transient features in NILM study to overcome the shortcomings of steady-state features [9–12]. In [9, 12], discrete wavelet transform (DWT) is proposed for extracting the transient features. Tsai *et al.* [10] deployed the energizing and de-energizing features in association with an artificial immune algorithm (AIA) and Fisher criterion. The energizing events are also processed by the S-transform to obtain the principal variables in the complex domain [11].

Generally, the use of transient features requires the implementation of signal processing such as DWT or S-transform at the high sampling rate to capture the transient effects. The main disadvantage of using transient features with those transforms is that the sampling frequency should be higher than 1 kHz to extract the distinction [13]. This is somewhat expensive in terms of the hardware cost for load-monitoring purposes.

2.2.3 Event Extraction

When appliances switch their on/off states, they tend to leave their different fingerprints. Hence, the simple edge detection can classify the distinction. This event-based method has been efficient at identifying the transient features. For example, Chang *et al.* [9] employed the turn-on transient energy with DWT at 30-kHz sampling frequency, which seems too expensive for hardware investment. At low frequency, the event features are simply the step changes from zero power to the steady demand as well as the switching duration. Nevertheless, this power change does not release complete information for event recognition. The multi-state appliances could be incorrectly detected by this method. In order to address this issue, some authors proposed the graph-signal processing and pattern matching technique to cluster the edge events [14, 15].

2.2.4 Hybrid Techniques

Apart from the steady state and transient behaviors, several hybrid techniques of feature extraction are also developed. Guedes *et al.* [16] used the high-order statistics (HOS) to extract the information from the current waveform. Genetic algorithms (GAs) were then applied to select the significant features. Those HOS features can retrieve the important information of electrical signals, so they possess the capability of discrimination in the feature space. Bouhouras *et al*. [17] described the concept of spectra distribution analysis for individual load activation. In that work, the raw signals were transformed into the corresponding fast Fourier Transform (FFT) spectra. Afterwards, the complete spectrum was divided into a number of non-overlapping frequency bands. The spectral band energy and entropies were finally calculated for feature extraction. Both [16, 17] reported the high performance in load disaggregation. Nevertheless, those outcomes were achieved at high sampling frequency, 15.4 kHz and 10.24 kHz, respectively.

2.3 Pattern Recognition

In the pattern recognition stage, machine learning, referred as the event-based tool, plays a crucial role in classification. Several papers, e.g., [18,19], have mentioned the use of artificial neural networks (ANNs) to facilitate the load recognition and harmonic sources for NILM systems. In [19], the authors used ANN to deal with harmonic issues, but the results do not involve the various operational modes of loads under different voltage sources. In NILM systems, most studies mainly focus on ANN with back-propagation (BP-ANN) for load recognition algorithms [10,16,18–20]. With simple implementation, k-Nearest Neighbors (k-NN) is also a commonly used recognizer. Saitoh *et al*. [20] implemented k-NN and support vector machine (SVM) to recognize household appliances. In that work, 1-NN was the most effective classifier compared with SVM. Tsai *et al*. [10] also drew a comparison between BP-ANN and k-NN. Both methods gave almost the same performance, and k-NN is preferable for its simplicity.

A hybrid recognizer that combines a supervised Self-Organizing Feature Map with Bayesian classification is also described in [21]. This method, however, encounters the difficulty of multi-state transitions in home appliances.

2.4 NILM in Other Frames

In this section, other frames for disaggregation algorithms are discussed. There are two common approaches: Hidden Markov model (sequential representation) and optimization-frame.

2.4.1 Hidden Markov Model

A hidden Markov model (HMM) is a sequential representation of data in which the states are unobserved [22]. The general idea is to use the states of appliances as the hidden variables, and the demand of each state is referred as the observation. To define the transition among states per appliance, a stochastic matrix is created. The extended version of HMM is the factorial hidden Markov model (FHMM) where a series of HMM chains is used [23]. The instantaneous observation is the combined observations from each individual chain at each time instant. In the NILM application, each chain describes the individual appliance in a house. Some NILM researchers have succeeded in applying HMM to load recognition [24,25]. In [26], multiple conditional factorial hidden Markov model (MCFHMM) algorithm is adopted for classification. Conditional FHMM is a modified version of FHMM where the additional conditions, e.g. time on a day or appliance dependency, are integrated. Although the identification accuracy of the unknown appliances exceeds 80%, MCFHMM may not correctly predict unknown linear loads.

2.4.2 Optimization Frame

A possible alternatives for NILM algorithm is combinatorial optimization (CO) or integer programming (IP). Some notable papers include Suzuki *et al*. [27] and Bhotto *et al*. [28]. In practice, the implementation of the CO-based method has not been widespread due to the following reasons:

1. The exact power states of every appliance should be pre-defined.
2. The total number of appliances in a house should be known.
3. Many electrical devices draw spikes or notches.
4. This method is unable to describe the physical characteristics of home appliances.

2.4.3 Deep Learning Approach in NILM

Deep Learning and Artificial Intelligence has been the growing research topic for image classification and speech recognition. In the classic machine learning, one-single stage of manual feature extraction is needed. The deep learning, on the other hand, has multiple stages of nonlinear feature mapping. This architecture gives the better quality of feature extraction, enhancing the performance. Therefore, deep learning also plays a potential role in NILM research. Kelly and Knottenbelt [29] proposed the convolutional neural network combined with auto-encoded inputs for NILM, achieving the high accuracy in disaggregation.

Chapter 3

Datasets for NILM Research

Various public sources of NILM datasets have been released since 2011. This facilitates the NILM research as the researchers could evaluate their work under the same conditions. In this chapter, three available public datasets are described, and one selected dataset is then visualized in the web page.

3.1 Public Datasets

3.1.1 Reference Energy Disaggregation Dataset

The Reference Energy Disaggregation Dataset (REDD) [30,31] is originally published by Massachusetts Institute of Technology (MIT). The goal is to create a benchmark data for evaluating the disaggregation algorithms. The dataset covers six houses in Massachusetts, USA, and consists of both main-circuit and individual-circuit levels. At the household level, the main circuits contain the 15-kHz current and voltage information. At the circuit-level data, each individual circuits meter the low-frequency (3-4 seconds) power readings at the corresponding UTC timestamp. According to

GoogleScholar, the REDD dataset was cited by 519 papers[1]. A brief summary of the dataset can be presented in Table 3.1.

Table 3.1: Description of REDD

House	Number of circuits	List of Appliances
1	20	Oven, Refrigerator, Dishwasher, Kitchen Outlets, Lighting, Washer Dryer, Microwave, Bathroom GFI, Electric Heat, Stove, Lighting
2	11	Kitchen Outlets, Lighting, Stove, Microwave, Washer Dryer, Refrigerator, Dishwasher, Disposal
3	22	Unmetered, Lighting, Electronics, Refrigerator, Dishwasher, Disposal, Furnace, Washer Dryer, Smoke Alarms, Microwave, Bathroom GFI, Kitchen Outlets, Lighting
4	20	Kitchen Outlets, Unmetered, Furnace, Lighting, Air Conditioning, Miscellaneous, Smoke Alarms, Stove, Bathroom GFI, Dishwasher, Washer Dryer
5	26	Unmetered, Furnace, Disposal, Electronics, Outdoor Outlets, Subpanel, Refrigerator, Dishwasher, Kitchen Outlets, Lighting, Washer Dryer, Microwave, Bathroom GFI, Electric Heat
6	17	Kitchen Outlets, Electronics, Bathroom GFI, Unmetered, Electric Heat, Air Conditioning, Lighting, Stove, Washer Dryer, Refrigerator, Dishwasher

3.1.2 Electricity Consumption and Occupancy Dataset

Electricity Consumption and Occupancy (ECO) dataset was collected in six houses in Switzerland during eight months [32,33]. For a household, 1-Hz aggregated

[1] https://scholar.google.es/scholar?um=1&ie=UTF-8&lr&cites=6707676167079398272

current, voltage and phase shift per phase are available. Also, 1-Hz data metered from selected individual appliances. The occupancy information was also provided. The ECO dataset was cited 59 times[2].

3.1.3 UK Domestic Appliance-Level Electricity Dataset

UK Domestic Appliance-Level Electricity (UK-DALE) dataset is the first UK open source from domestic appliance electricity data [34]. UK-DALE was collected from five households during 655 days per house. The whole-house meters records the reading at 16 kHz; while individual appliances were measured every six seconds. At the time of writing, UK-DALE was referred in 49 published papers[3].

3.1.4 Summary

Table 3.2 summarizes of those public datasets. They have been released as the result of the particular attention to the NILM research topic since 2010, as mentioned in Section 2.1. Since REDD is widely used for benchmarking the NILM algorithms, it is the selected option in this thesis.

Table 3.2: Comparison of three public household datasets

Dataset	**REDD**	**ECO**	**UK-DALE**
Countries	United States	Switzerland	United Kingdom
Duration	3-19 days	8 months	655 days
Number of Households	6	6	5
Ground-truth Frequency	3 seconds	1 second	6 seconds
Terms of Use	Registration	Public	Public
Number of citations	519	59	43

[2] https://scholar.google.es/scholar?um=1&ie=UTF-8&lr&cites=6717214458962576488

[3] https://scholar.google.es/scholar?um=1&ie=UTF-8&lr&cites=5486688338434564807

3.2 REDD Visualization

The major objective of visualization is to create a NILM web server where the inputs are the total consumption of a household, and the outputs are the disaggregated estimation. Based on this, an NILM application in phones could be developed for users to on-site manage their energy usage. In this scope, a simple web interface is initially built for visualizing the raw REDD dataset. The thesis only focuses on the low-frequency (3-s) REDD data as they are more realistic scenario for the current technology of smart meters. In the future work, different disaggregation algorithms should be implemented to automatically detect home appliances.

Fig. 3-1 shows the area-normalized histogram of the household-level meters in 6 houses. This normalization will return the unit area under the probability density function. As can be seen, the distributions vary according to different household characteristics. This means that the disaggregation model can predict different outcomes corresponding to different houses. Fig. 3-2 visualizes the demand for a typical day of house 1. It is quite obvious that the data are noisy, which may affect the model performance.

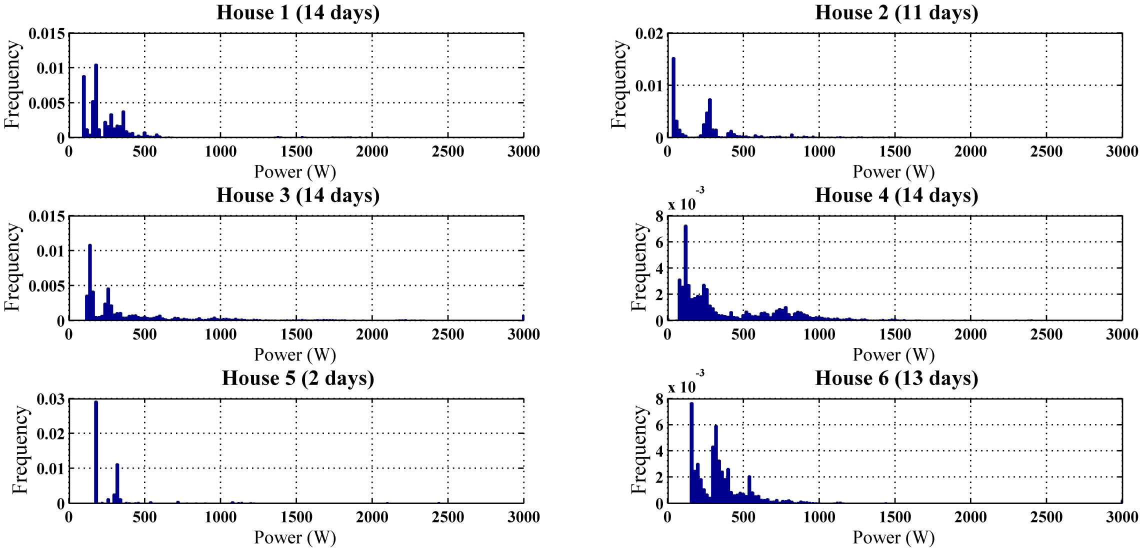
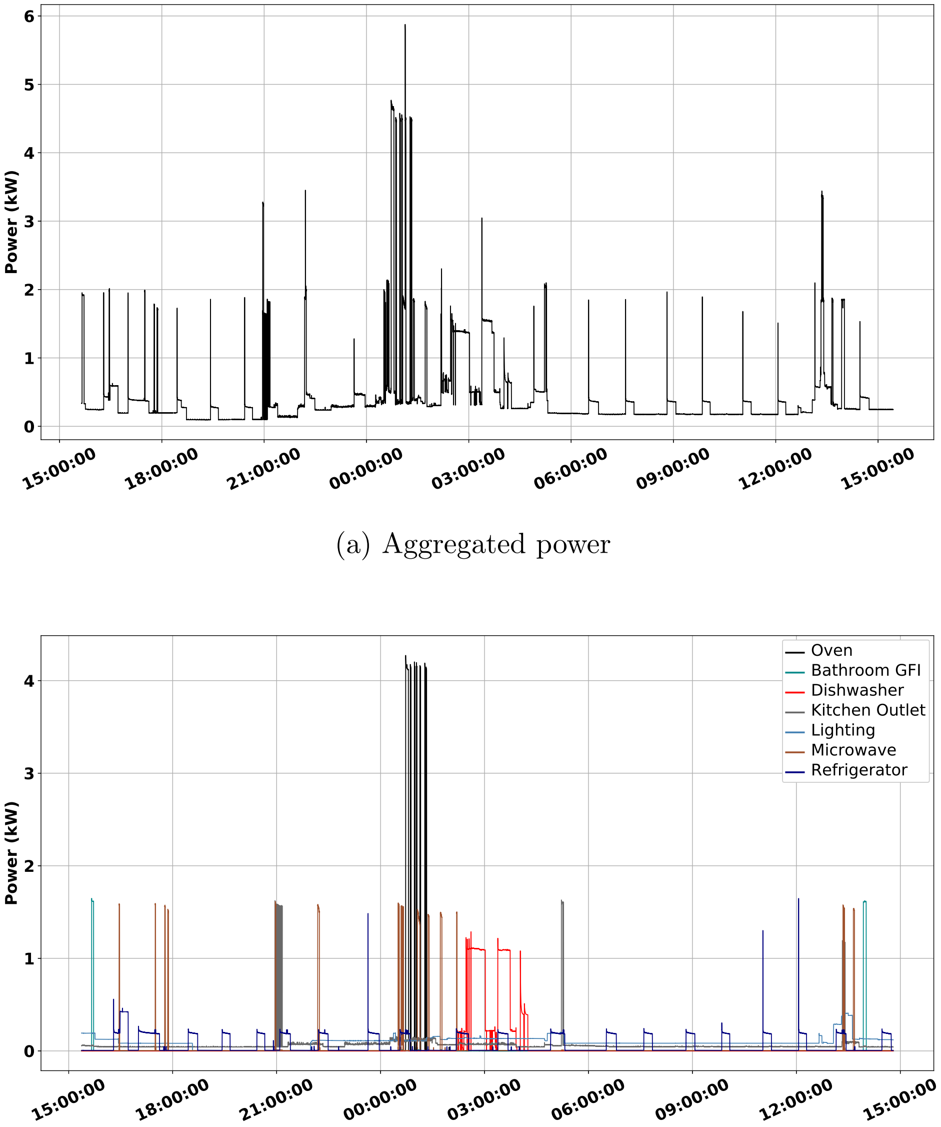


Figure 3-1: Area-normalized histograms of the main meters in 6 houses of REDD



(a) Aggregated power

(b) Ground-truth of some individual appliances.

Figure 3-2: Consumption of house 1 from 18th- 19th April, 2011.

3.2.1 Visualization Tools

The REDD visualization is carried out by Javascript in Microsoft Visual Studio Code[4] (Ver. 1.10.1). For the initial basic visualization, the classic line graph is selected to represent the time-series consumption. Those open-source Javascript libraries and

[4] Microsoft Visual Studio Code. Homepage: https://code.visualstudio.com/

toolkits are included for this visualization:

- Data-Driven Documents (d3)[5] (Ver. 4.10.0).
- JQuery[6] (Ver. 3.2.1).
- Dygraph[7] (Ver. 2.0.0)
- Jszip[8] (Ver. 3.1.3).
- Papaparse[9] (Ver. 4.3).
- BootstrapCDN[10] (Ver. 3.3.7)
- Font Awesome[11] (Ver. 4.7)

The detail descriptions for those toolkits and libraries are provided in their homepages. Since the data visualization is not the main scope of this thesis, this section will give a brief summary of the REDD visualization process.

Firstly, the data from channels were preprocessed and broken down into different appliances. For instance, the preprocessed data of house 1 are described as follows:

- Oven: Channel 3 and 4.
- Refrigerator: Channel 5.
- Dishwasher: Channel 6.
- Kitchen Outlets: Channel 7, 8, 15 and 16.
- Lighting: Channel 9, 17 and 18.
- Washer Dryer: Channel 10, 19 and 20.
- Microwave: Channel 11.
- Bathroom GFI: Channel 12.
- Electric Heat: Channel 13.
- Stove: Channel 14.

[5] D3 Library. Homepage: https://d3js.org/

[6] JQuery Library. Homepage: https://jquery.com/

[7] Dygraph Library. Homepage:http://dygraphs.com/index.html

[8] Jszip Library in Github. https://stuk.github.io/jszip/

[9] Papaparse Library. Homepage: http://papaparse.com/

[10] BootstrapCDN delivery network. Homepage: https://www.bootstrapcdn.com/

[11] Font Awesome Toolkit. Homepage: http://fontawesome.io/

This pre-processing is scripted in MATLAB[12] (Ver. 2014a). For the appliances measured by multiple channels (e.g. Oven, Washer Dryer ...), their powers are the sum of all channel readings. The output files of the pre-processing step for visualization are in the format:

- Individuals: “Name Appliance_house_ID.csv” (ID = 1 ... 6).
- Main: “Main_house_ID.csv”

Those files are stored in one folder corresponding to one house, so there are totally six folders. Such folders are compressed into zip files “House_ID.zip”. The Javascript code for this visualization can be saved in the html file “Test_ID.html”. There are six html files corresponding to six houses for the basic visualization.

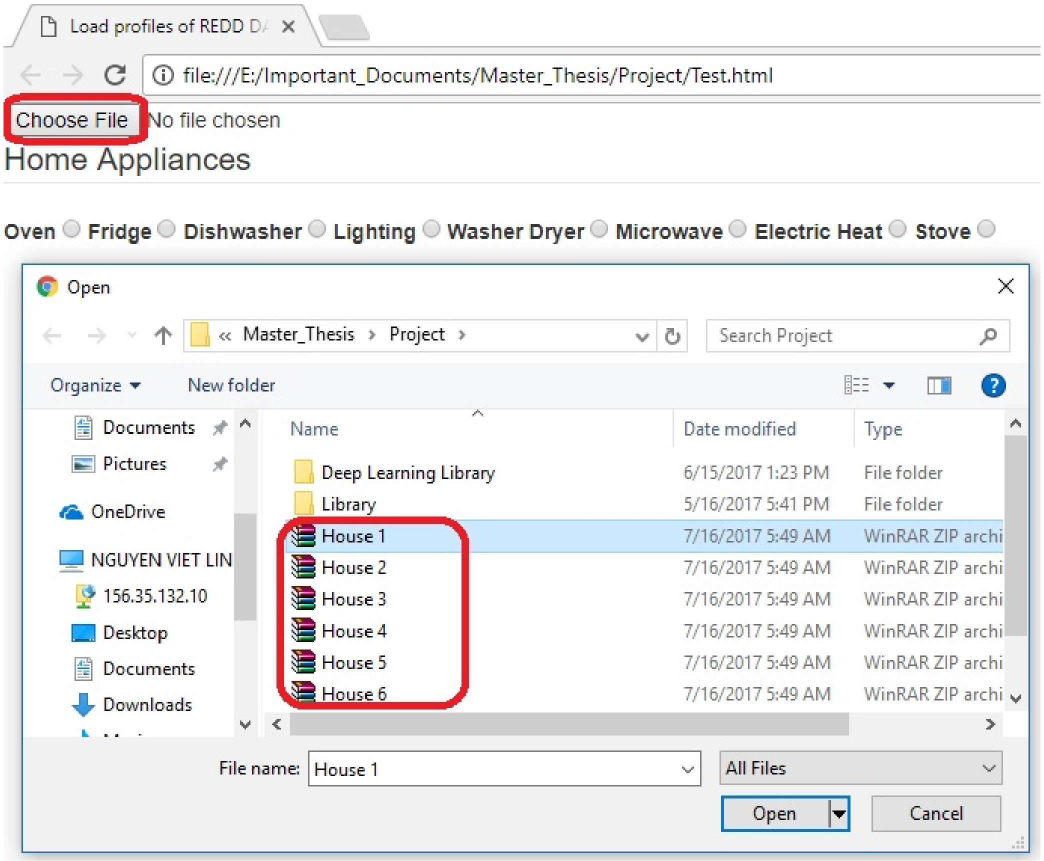
3.2.2 Webpage View of REDD Dataset

The screenshot of the web interface can be described in Fig. 3-3. When the users click “Choose File” to select the household. A list of households in the zip format can be shown.

The recording of the main circuit (1 Hz) will be displayed. As can be seen in Fig. 3-4, the aggregated power was recorded from 18/04/2011 to 24/05/2011. The x-axis is the UTC timestamp, and the y-axis is the power in W. When the mouse is positioned on any data point, its cursor is shown in the top right of the graph. This includes the details of time and day as well as the power in W. Also, the graph can be zoomed out for higher resolution, as shown in Fig. 3-5

In house 1, there are some days on which the main meters are failed to record the power: 28 - 30/04; 03 - 06/05; 07 - 11/05; and 13 - 22/05. In order to view the individual consumption, the user can click on the appliance list, e.g. dishwasher. The records of the dishwasher can be shown below the main circuit (see Fig. 3-6). As can

[12] MathWorks. Homepage: https://es.mathworks.com/



file:///E:/Important_Documents/Master_Thesis/Project/Test.html

Name	Date modified	Type
Deep Learning Library	6/15/2017 1:23 PM	File folder
Library	5/16/2017 5:41 PM	File folder
House 1	7/16/2017 5:49 AM	WinRAR ZIP archi
House 2	7/16/2017 5:49 AM	WinRAR ZIP archi
House 3	7/16/2017 5:49 AM	WinRAR ZIP archi
House 4	7/16/2017 5:49 AM	WinRAR ZIP archi
House 5	7/16/2017 5:49 AM	WinRAR ZIP archi
House 6	7/16/2017 5:49 AM	WinRAR ZIP archi

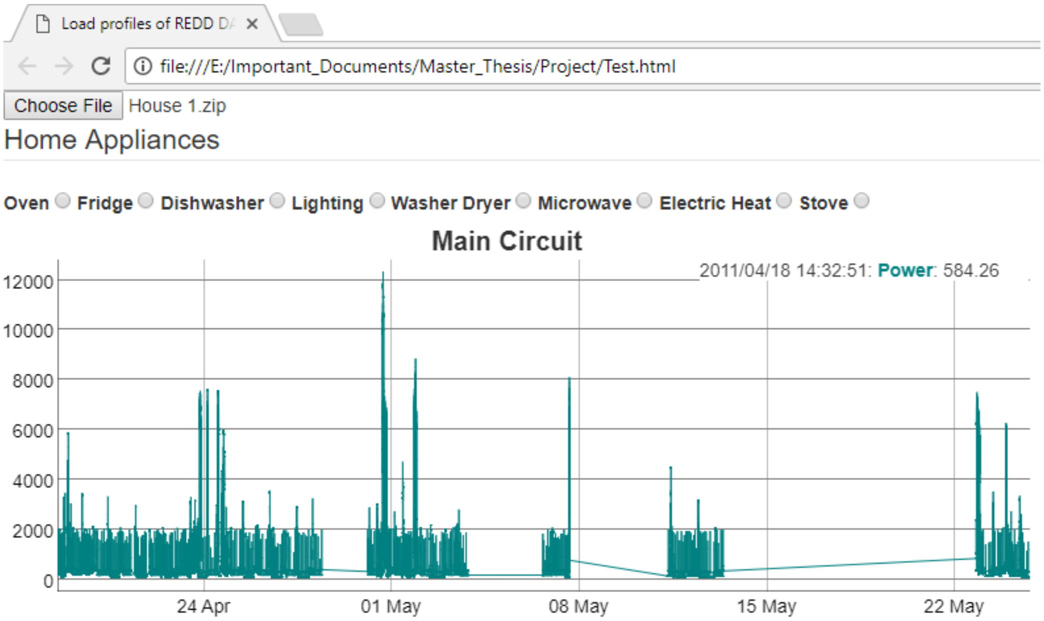
File name: House 1

Figure 3-3: Screenshot of the webpage interface for visualization house 1.

seen in Fig. 3-6, there is the non-alignment between the records of dishwasher and the main circuit. From 08/05/2011 to 10/05/2011, the household-level meter could not measure the main reading; while the dishwasher information was successfully captured during this day. This incorrect alignment could lead to the difficulty in comparing the disaggregation estimates with the ground-truth levels.

3.2.3 Selected Appliances

In REDD dataset, some appliances only consume a small portion of power (up to 30 W); while some could not be metered or in operation. Therefore, their information is not useful and considered to be neglected. This could be filtered by a median filter. Lists of selected appliances per house can be shown:



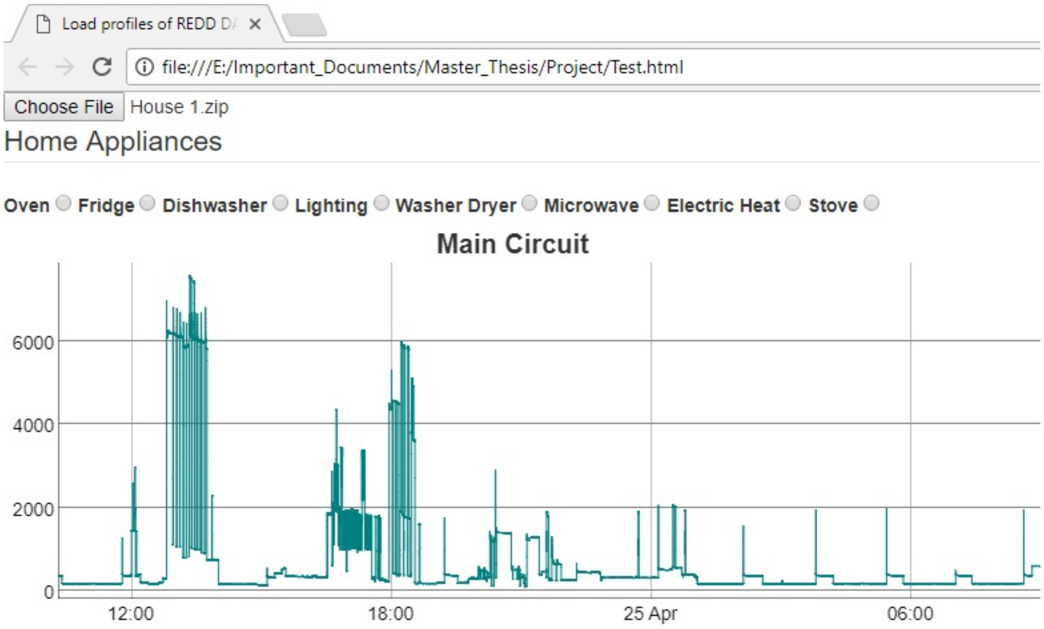
file:///E:/Important_Documents/Master_Thesis/Project/Test.html

Choose File House 1.zip

Load profiles of REDD D
Home Appliances
Oven ○ Fridge ○ Dishwasher ○ Lighting ○ Washer Dryer ○ Microwave ○ Electric Heat ○ Stove ○
Main Circuit
2011/04/18 14:32:51: Power: 584.26
12000
10000
8000
6000
4000
2000
0
24 Apr
01 May
08 May
15 May
22 May

Figure 3-4: The records of house-1 aggregated power.

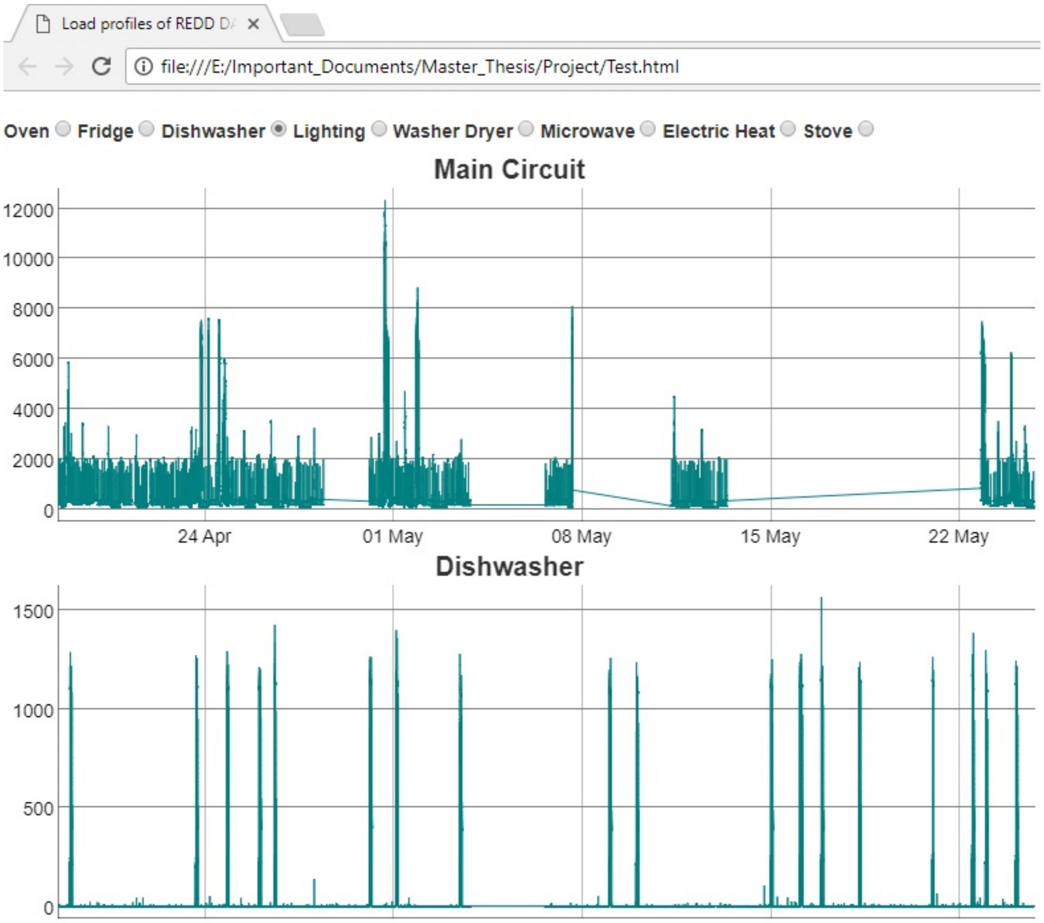
- House 1: Oven, Bathroom GFI, Dishwasher, Refrigerator, Kitchen Outlets, Lighting, Microwave, Washer Dryer.
- House 2: Dishwasher, Refrigerator, Kitchen Outlets, Lighting, Microwave, Stove.
- House 3: Bathroom GFI, Dishwasher, Electronics, Refrigerator, Furnace, Kitchen Outlets, Lighting, Microwave, Washer Dryer.
- House 4: Dishwasher, Furnace, Kitchen Outlets, Lighting, Stove, Washer Dryer.
- House 5: Bathroom GFI, Dishwasher, Electronics, Refrigerator, Furnace, Lighting, Subpanel.
- House 6: Air Conditioner, Electric Heat, Electronics, Refrigerator, Kitchen Outlets, Lighting, Stove, Unmeters (Unknown Outlets).


file:///E:/Important_Documents/Master_Thesis/Project/Test.html

Choose File House 1.zip

Load profiles of REDD D
Home Appliances
Oven ○ Fridge ○ Dishwasher ○ Lighting ○ Washer Dryer ○ Microwave ○ Electric Heat ○ Stove ○
Main Circuit
6000
4000
2000
0
12:00
18:00
25 Apr
06:00

Figure 3-5: The extracted records of house-1 aggregated power (zoomed from 24 - 25/04).



Load profiles of REDD D/

file:///E:/Important_Documents/Master_Thesis/Project/Test.html

Oven ○ Fridge ○ Dishwasher ● Lighting ○ Washer Dryer ○ Microwave ○ Electric Heat ○ Stove ○

Figure 3-6: The power records of the main circuit and dishwasher of house 1.

Chapter 4

Proposed Methods

This chapter explains two disaggregation algorithms: Combinatorial Optimization (CO) and Deep Recurrent Neural Network (DRNN) for the REDD dataset case study. In order to evaluate the performance of the proposed methods, two metrics: assigned energy and classification are used. The proposed methods can be generally depicted in the Fig. 4-1 below.

4.1 Data Preprocessing

As mention in Section 3.2.2, one of the main problems of REDD dataset is that the sampling frequency of the main circuit (1 s) differs from the sampling rate of appliance-level meters (3 s). Hence, the identification of the ground truths in the main circuit is not a simple task. To alleviate this issue, the aggregated power should be re-calculated, which is equal to the sum of all appliance-level meters.

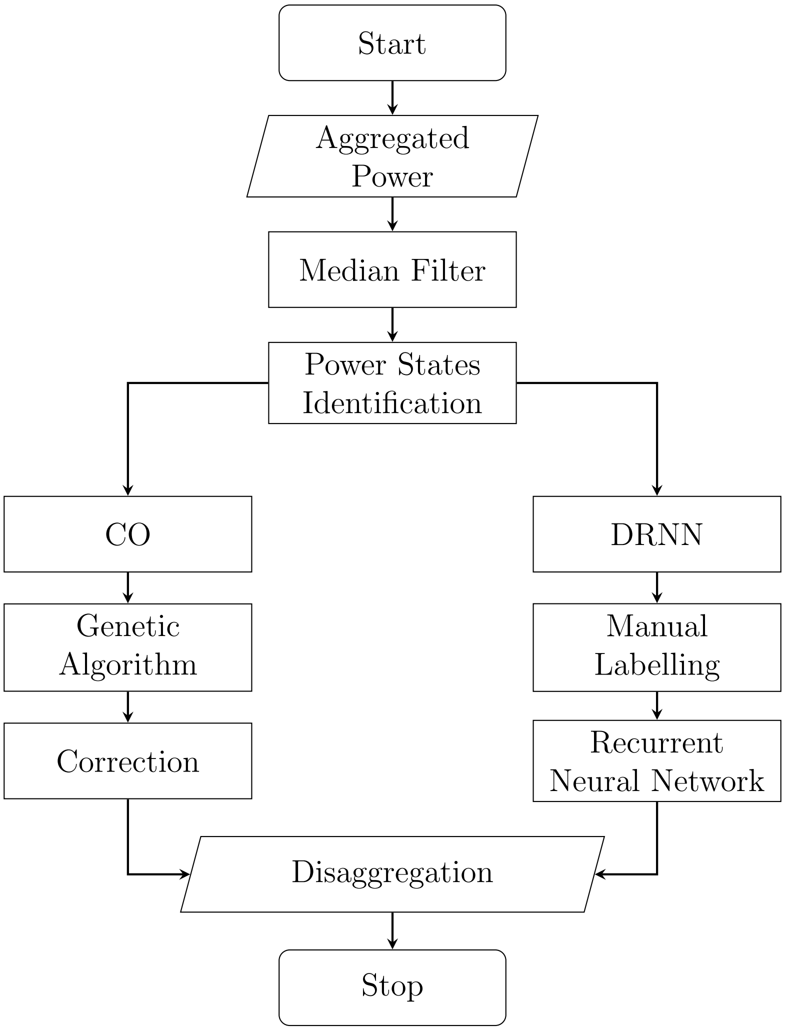


Figure 4-1: Flowchart of the proposed algorithms.

4.1.1 Median Filtering

Secondly, the simple median filter should be applied to remove the noise in aggregated power (e.g. short-time spikes, notches). The filter order applied in this case is 30. To illustrate, a filtering window, which has the width of 30 samples, slides over the entire signal. At k^{th} sample, $P_{filtered}(k)$ is the median of $[P_{k-15}, P_{k-14},..., P_{k+15}]$. This could be done by the following simple MATLAB script:

```
P_filtered=medfilt1(P,30);
% P: Raw waveform
% P_filtered: Filtered output
% 30: Filter order
```

Fig. 4-2 depicts the raw and the filtered aggregated power in house 1 during 14 days. The mean average error is:

$$MAE = \frac{\sum_{k=1}^{n} |P_k - P_{filtered,k}|}{n} = 14.9867 \text{ W}$$

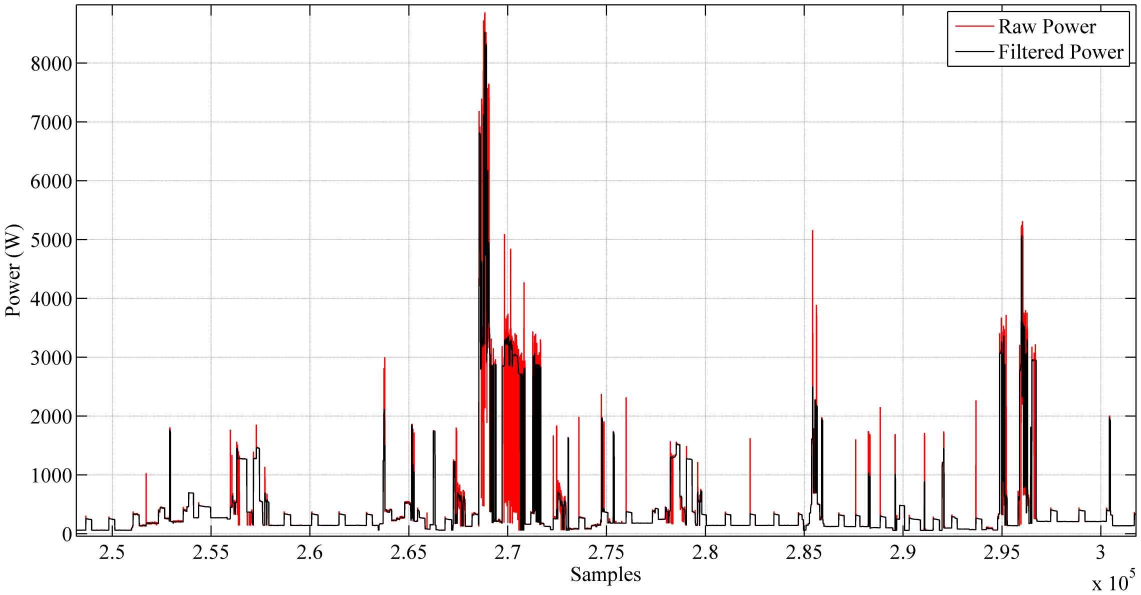


Figure 4-2: Raw and the filtered aggregated power in house 1 during 14 days (Zoomed at the sample 250000$^{\text{th}}$ to 300000$^{\text{th}}$).

4.1.2 Power States Clustering

This preprocessing stage plays a crucial role in thresholding the different states of appliances. This creates parameter matrices for CO in Section 4.2 and a binary on/off vector for the manual labeling of supervised learning in Section 4.3.

k-means clustering is the method for partitioning data into k clusters from n observations [35]. Those clusters are initially defined by the centroids. Then, each data point (observation) is assigned to the nearest centroid. Those centroids are updated every iterations by regulating the average of the observation in the previous loop. The loops continue until the centroid points are fixed. Mathematically, the aim of k-means algorithm is to minimize this function:

$$\sum_{p=1}^{q}\sum_{j=1}^{i}||X_j^{(p)} - Y_p||^2 \tag{4.1}$$

where $X_j^{(p)}$ is the j^{th} data point of the p^{th} cluster in which Y_p is its centroid. Totally, there are q clusters, and i data points per cluster. The k-means algorithm can be used to identify of threshold of each power state. An area-normalized histogram of an appliance is used to determine the number of states. Fig. 4-3 describes the histogram of the selected appliances in house 1. As can be seen, there is no considerable difference among the power states of kitchen outlets. The clustering for this appliance could not be a simple task. The dishwasher in house 1 could be clustered in three states, including the off state. The power-state threshold of this dishwasher can be identified by this simple MATLAB script:

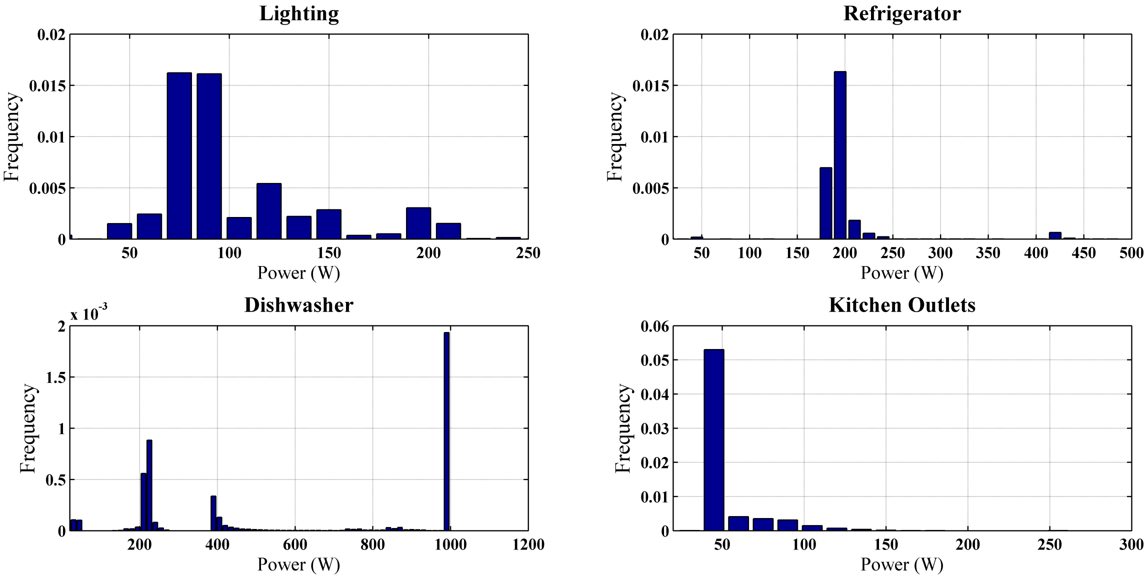


Figure 4-3: Area-normalized histogram of selected appliances in house 1

```
[idx,C]=kmeans(Dishwasher,3,'Distance','cityblock','Replicates',10);
```

```
%3 is number of states (including OFF state)
%idx: cluster indices in each observation
```

C returns the values of centroids 0 (off-state), 221 W (1st ON) and 1102 W (2nd ON). This can be visualized in Fig. 4-4. The power states (non-OFF) of appliances in house 1 can be summarized in Table 4.1. The power states of appliances in the remaining houses are listed in Appendix A.

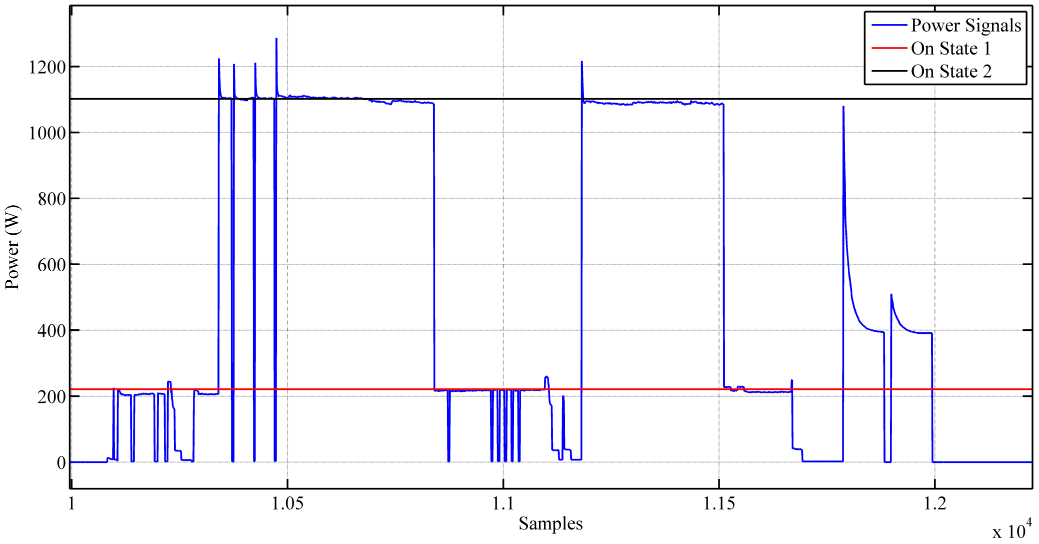


Figure 4-4: Three power states of the dishwasher in house 1.

Table 4.1: Non-off power states in W of house-1 appliances

Appliances	**State 1**	**State 2**	**State 3**	**State 4**
Oven	4117.5	-	-	-
Washer Dryer	2657	-	-	-
Microwave	1525	-	-	-
Lighting	45	64	82	-
Kitchen Outlets	22	64	1047	1522
Refrigerator	191	-	-	-
Dishwasher	221	1102	-	-
Bathroom GFI	1596	-	-	-

4.2 Combinatorial Optimization

The optimization-based algorithm of disaggregation can be described as follows. The time-series aggregated power of a house is recorded $P = \{P_1,\ P_2,\ ...,\ P_t\}$ where P_t is the kW reading at t^{th} time instant. Assume that the house has m appliances. The state of k^{th} appliance can be described by a binary vector $X = \{x_1,\ x_2,\ ...,\ x_m\}$. When the k^{th} appliance is on $x_k = 1$; otherwise, $x_k = 0$. The individual consumption of each appliance can be expressed in a vector $p = \{p_1,\ p_2,\ ...,\ p_m\}$. At every t^{th} instant, the combined demand should be equal the sum of all individual demands, as can be expressed:

$$P_t = \sum_{k=1}^{m} p_k x_k(t) + w_t \tag{4.2}$$

where w_t is the error.

In the case of supervised learning, the rating power of each appliance should be known. Then, for every iteration of t, we should try to search a variable vector X_t such that w_t is minimized, i.e.

$$\min_{X_t} \left| P_t - \sum_{k=1}^{m} p_k x_k(t) \right| \tag{4.3}$$

Assume that the m^{th} appliance has l states. Since any appliance must be in only one state at the t^{th} time instant, the constraint for multi-state appliance can be described:

$$x_{m,1}(t) + x_{m,2}(t) + ... + x_{m,l}(t) \leq 1 \tag{4.4}$$

Based on Table 4.1, there are totally 14 states if house 1 is considered. This corresponds to a matrix size 14 × 1, $X = \{x_1, x_2, ..., x_{14}\}$. The variables can be assigned as follows:

- Oven: x_1.
- Washer Dryer: x_2.

- Microwave: x_3.
- Lighting: x_4, x_5, x_6 (3 states).
- Kitchen Outlets: x_7, x_8, x_9, x_{10} (4 states).
- Refrigerator: x_{11}.
- Dishwasher: x_{12}, x_{13} (2 states).
- Bathroom GFI: x_{14}.

It is noticeable that the second state of the dishwasher is active if and only if its first state is previously active, i.e.

$$x_{12}(t) + x_{13}(t) \leq 1. \tag{4.5}$$

Then, Eq. 4.3 can be explicitly reformulated in MATLAB environment for the house-1 case, as follows:

```
%Create objective function f
f=@(x)abs(P(i) - ... %Aggregated Power at instant i-th
(4117.5*x(1)... %Oven
+2657*x(2)... %Washer Dryer
+1525*x(3)... %Microwave
+45*x(4)+64*x(5)+82*x(6)...  %Lighting
+22*x(7)+64*x(8)+1047*x(9)+1522*x(10)...   %Kitchen Outlets
+191*x(11)... %Refrigerator
+221*x(12)+1102*x(13)...    %Dishwasher
+1596*x(14))  %Bathroom GFI
```

4.2.1 Genetic Algorithms

Genetic Algorithm (GA) is the evolutionary computation for the metaheuristic optimization [36]. GA is inspired from the natural evolution in biology. Firstly, a population of individuals is randomly generated, which is the first generation. The fitness function of each chromosome is calculated. At the end of this process, GA selects a few number of individuals that return the optimal values of the fitness function or objective function. Those individuals are then used to breed their offsprings in the next iteration. The process repeats until either the optimal solution is found or the terminating condition is fulfilled. The basic GA elements include:

- Number of generations.
- Number of populations per generation.
- Operator parameters: crossover, mutation.

In the binary bits, the crossover operator swaps a sub-section of the chromosomes; while the mutation will change the value of an arbitrary bit. The number of bits that are being crossed (crossover point) or mutated (mutation point) depends on the setting parameters in algorithm. Usually, the probability of crossover is high (e.g. 0.8); while the mutation rate is relatively low (e.g. 0.1). Those operators can be simply illustrated in Fig. 4-5. The selections of population and generation values are

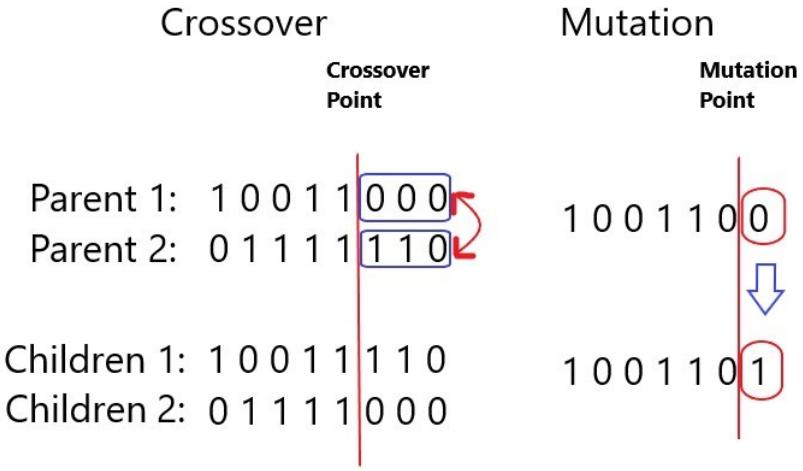

Crossover
Crossover Point
Parent 1: 1 0 0 1 1 0 0 0
Parent 2: 0 1 1 1 1 1 1 0
Children 1: 1 0 0 1 1 1 1 0
Children 2: 0 1 1 1 1 0 0 0
Mutation
Mutation Point
1 0 0 1 1 0 0
1 0 0 1 1 0 1

Figure 4-5: An example of GA operators.

primarily tuned during experiments. The small values may undercover the solution space; while the big values lead to computational inefficiency. In this work, the generation and population values are 100 and 200, respectively. The process in each iteration will stop if the average change in the fitness value is less than the threshold

TolFun = 1e-6. GA can be scripted in MATLAB environment as follows:

```
options=gaoptimset(’TolFun’,1e-6,’Generations’,100,’PopulationSize’,200);
x=ga(f,14,A,b,[],[],lb,ub,[],IntCon,options);
% 14: Number of variables
% A,b: inequality matrix constraints: A*x <= b
% A = [0 0 0 0 0 0 0 0 0 0 0 1 1 0]:
% lb: Matrix of 14x1 Lower Boundary for variables: 0 in this case
% up: Matrix of 14x1 Upper Boundary for variables: 1 in this case
% IntCon: Matrix indicates which variables are integer (IntCon = 1:14)
```

The inequality constraint $Ax \leq b$ is Eq. 4.3. A(12:13) = 1 (for dishwasher) while other elements are zero. b = 1.

4.2.2 Correction Rule

The result of GA optimization could lead to short-time transition where one appliance can be in its state for only few samples. Generally, it is not possible, so a correction rule should be applied to filter out the short-time transition event. This correction rule is formulated as follows [28]:

Algorithm 1 Correction Rule

```
$X \leftarrow$ *A binary matrix as a result of GA optimization*
**if** median($X_t$,$X_{t-1}$,...,$X_{t-M}$) = 0 and $X_{t-M} = 1$ **then**
    $X_{t-M} = 0$
**else if** median($X_t$,$X_{t-1}$,...,$X_{t-M}$) = 1 and $X_{t-M} = 0$ **then**
    $X_{t-M} = 1$
**end if**
```

This rule implies that one appliance will remain in its state at least within M samples. The appliance state at time $t - M$ is adjusted based on its current state t. In this work, $M = 5$.

4.3 Deep Neural Networks

In this section, the deep recurrent neural network (DRNN) approach for REDD dataset is described. After median filtered, the aggregated real power enters into the neural network layers for classification. Two common deep neural network architectures below are described:

- Multilayer Feed-Forward Neural Network.
- Recurrent Neural Network.

4.3.1 Background

Artificial Neural Networks (ANN) has been the effective technique to extract and recognize the patterns that are too complicated to be detected by humans or the other tools. An ANN architecture consists of an input layer, hidden layers, and one output layer. Basically, the information from the inputs is forwarded, through hidden layers, to the outputs. The backward correction, from outputs to inputs, is used to update the weights.

Forward Direction

If the input is the vector X, the weight matrix W is then applied to connect x with a set of neuron nodes. The output of the neuron layer is the weighted sum $W \times X$. This quantity is then transfered through the activation function F, and $Y = F(W \times X)$ is the final output of ANN. The process can be simply visualized in Fig. 4-6. Several common activation functions include:

- linear: $Y = F(X) = X$
- tanh: $Y = F(X) = tanh(X) = \frac{e^X - e^{-X}}{e^X + e^{-X}}$
- logsig: $Y = F(X) = logsig(X) = \frac{1}{1+e^{-X}}$

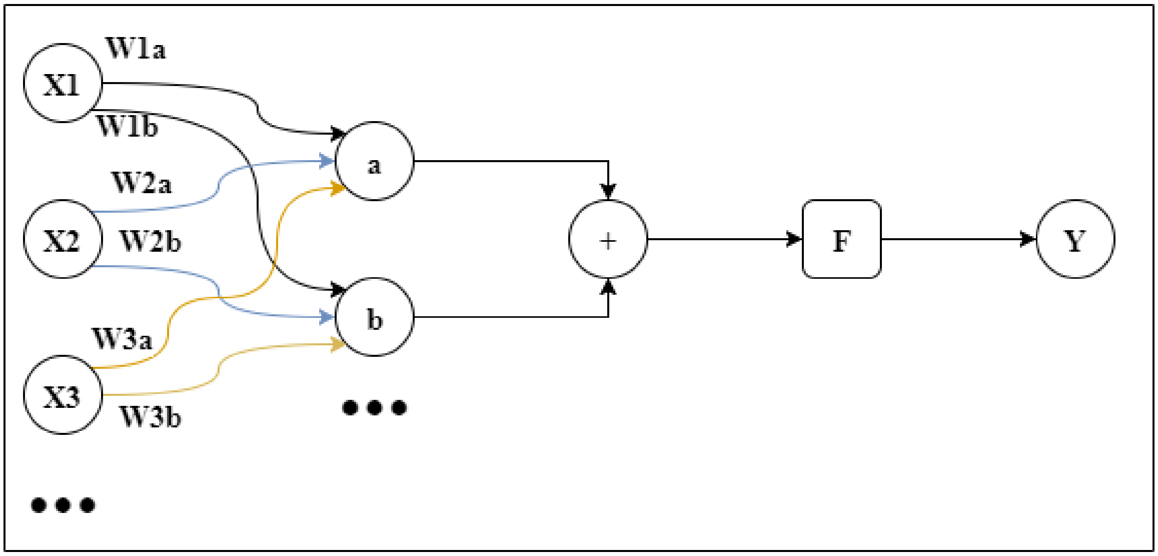


Figure 4-6: Forward pass of an ANN architecture.

- tansig: $Y = F(X) = tansig(X) = \frac{2}{1+e^{-2X}} - 1$

The activation function used in this work is tansig from the input to hidden layers and pure linear transfer from the hidden to output layer. The primary reason is that tansig is the simple nonlinear function which produces the output in the normalized range [0;1]. This function has the simple and continuous derivatives in which backpropagation can take place. In contrast, the linear function seems ineffective to detect features in a complicated dataset.

Backward Direction

Once the output Y of the forward is obtained, the error e between Y and the target vector T is computed. Then, W is adjusted in order to minimize this error term. In the training of ANN, backpropagation (BP) principle is commonly applied for correcting this error term. In this correction, the backward error is propagated to adjust the interconnecting weights W. To illustrate, the slope of the gradient descent is calculated by the ratio of the change in error and the change in weight. This ratio is a learning rate constant for the first-order gradient, or steepest descent gradient. The general backpropagation rule can be explained in Fig. 4-7 and Eq. 4.6. In general, if a chaining network is present, the term $\frac{\partial e}{\partial W}$ is the cascaded transfer function of the chain.

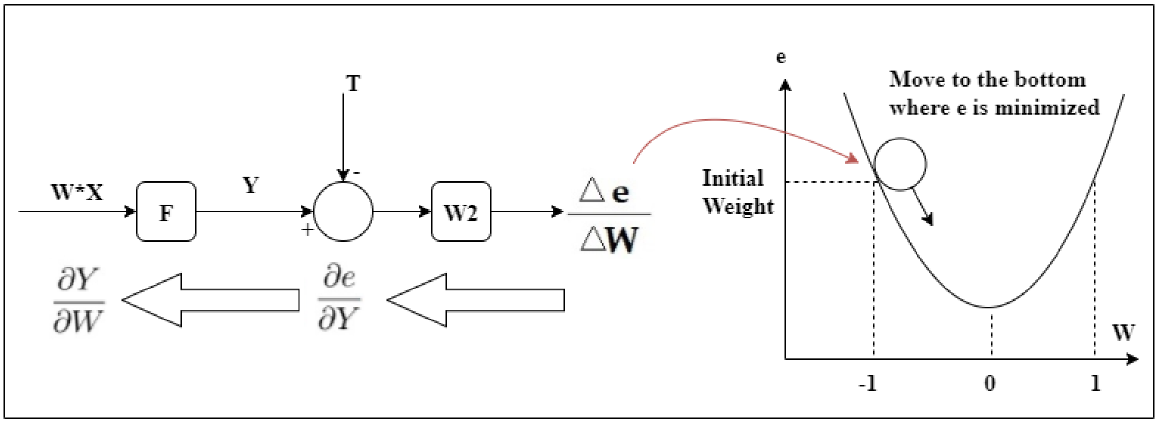

T
W*X
F
Y
-
+
W2
$\frac{\triangle \mathbf{e}}{\triangle \mathbf{W}}$
$\frac{\partial Y}{\partial W}$
$\frac{\partial e}{\partial Y}$
e
Move to the bottom
where e is minimized
Initial
Weight
W
-1
0
1

Figure 4-7: Backpropagation principle.

$$\mathbf{e} = (Y - T) \times W2 \Rightarrow \frac{\partial \mathbf{e}}{\partial Y} = W2 = \frac{\partial e}{\partial (WX)} \Rightarrow \frac{\partial \mathbf{e}}{\partial W} = X \times W2 = \frac{\partial Y}{\partial W}\frac{\partial \mathbf{e}}{\partial Y} \qquad (4.6)$$

Nevertheless, the steepest descent computes the first-order gradient, which can slow the convergence and increase training time for the complex dataset. To avoid this issue, the error minimization has been adapted to least-square minimization where fitness function is mean squared error (MSE) defined by Eq. 4.7:

$$MSE(l) = \frac{1}{2}\sum_j e_{j,l}^2 = \frac{1}{2}\sum_j (T_j(l) - Y_j(l))^2 \qquad (4.7)$$

where $T_j(l)$ and $Y_j(l)$ are the desired output and the network output for the l^{th} iteration at node j, respectively.

The network nodes (neurons) are activated until the requirements of MSE or epoch numbers are fulfilled. There are several algorithms being used to minimize the least-square $MSE(l)$. This work has implemented the common Levenberg-Marquardt algorithm ('trainlm' in MATLAB) for backpropagation correction as it is fast and highly recommended for supervised learning[1]. The term of classification in these case

[1] trainlm: https://es.mathworks.com/help/nnet/ref/trainlm.html

studies denotes a mapping from the input feature vectors (measured ratings) to the sets of assigned labels which indicate the home appliances.

4.3.2 Multilayer FeedForward Neural Network

Feedforward neural network (FNN) is the simplest architecture of ANN. In this network, the connection is completely straightforward and open-loop. Multilayer feedforward neural network (MFNN) has been commonly used for nonlinear classification. MFNN consists of one input layer, one output layer and at least one hidden layer. Fig. 4-8 depicts a simple MFNN structure. Multilayer Network is usually referred as Deep Network.

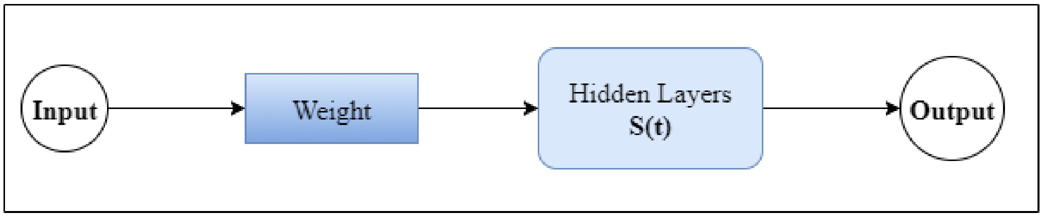


Figure 4-8: A simple MFNN architecture.

4.3.3 Recurrent Neural Network

Unlike FNN, recurrent neural network (RNN) propagates the information not only forward but also backward in each layer. A recurrent connection with a delay means the sequences of inputs can be processed. For example, a delay of 2 means that the values of the time series at time $t-1$ at $t-2$ are stored in the internal memory. At time t, they are simultaneously loaded together with the t^{th} value for processing. In other word, the previous outputs of RNN at $t-1$ and $t-2$ affects the decision at t. Fig. 4-9 illustrates this example structure.

The advantage of RNN, compared with FNN, could be simply described by the following example. In the text translation, FNN always translate the original meaning

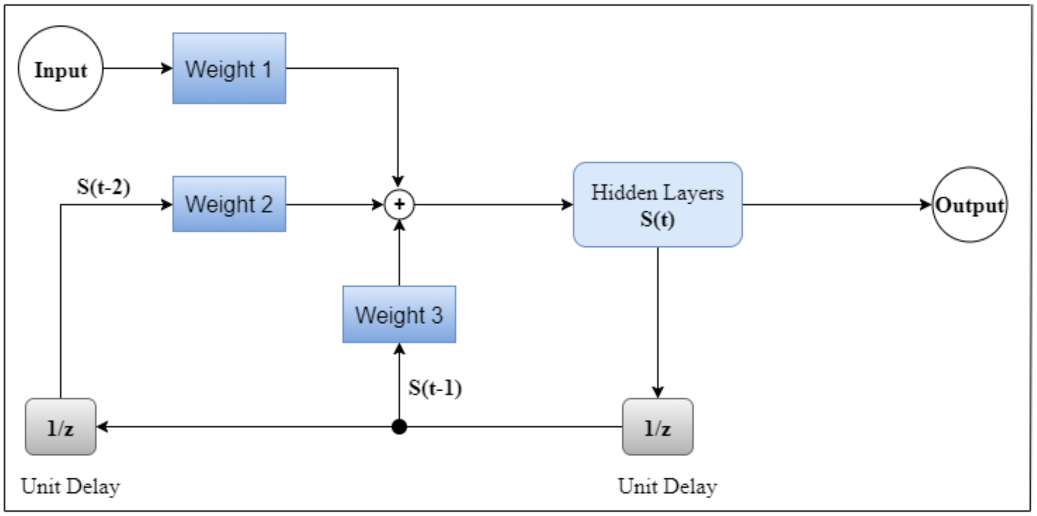


Figure 4-9: A simple RNN architecture with two delays.

of “sentence”, which does not depend on contexts. Meanwhile, the RNN model could differently translate “sentence” if it was preceded by “life”, meaning “punishment”. With the capability of recognizing the time-series patterns, RNN seems the natural fit to energy disaggregation. An example of RNN subroutine applied in house 1 could be scripted in MATLAB, as follows:

```
%P_agg: Aggregate Input
Train=1:2:length(P_agg); %Training set
Test=2:2:length(P_agg); %Test set
Xtrain=P_agg(Train); %Train data
testX=P_agg(Test); %Test data
Ttrain=T(1:end,Train); %Train target
testT=T(1:end,Test); %Test target
%-------------------------------------------------------------
%------------RNN initialized-------------------------
N=5; %Number of delays,
%No. of hidden layers: 10
net = layrecnet(1:N,[10 10],’trainlm’); %Levenberg-Marquardt BP
%-------------------------------------
net.trainParam.epochs = 200;
%---------------Start training RNN-------------
```

```
[net,tr,Y,E] = train(net,trainP,trainT);
%Y: the training output
%E: the training error
%net: neural network architecture
%---------------Test net response of new input-------------
%The network returns the new output (testY)
%corresponding to new input (testX)
testY=sim(net,testX);
%--------------------------------------------
```

4.4 Metrics

4.4.1 Disaggregation Metrics

The disaggregation metrics are followed by the formula of proportion of total energy correctly assigned over a period T [29, 31].

$$\text{Total energy correctly assigned} = 1 - \frac{\sum_{t=1}^{T}\sum_{i=1}^{n}|P_i^* - P_i|}{2\sum_{t=1}^{T}\bar{P}_t}\ (\%) \tag{4.8}$$

$$\text{Energy correctly assigned per appliance (ECApA)} = 1 - \frac{\sum_{t=1}^{T}|P_i^* - P_i|}{2\sum_{t=1}^{T}P_i}\ (\%) \tag{4.9}$$

$$\text{Relative error in total energy} = \frac{|E^* - E|}{\max(E^*, E)}\ (\%) \tag{4.10}$$

where:

- P_i^*: Estimated power of appliance i^{th} at t.
- P_i: Actual power of appliance i^{th} at t.
- $\bar{P}_t$: Aggregated actual power at t.
- E^*: Total predicted energy.

- E: Total actual energy.

4.4.2 Classification Metrics

For any classification problem, a confusion matrix is the most useful tool to evaluate the machine learning algorithm. The general confusion matrix can be described in Table 4.2.

Table 4.2: A general confusion matrix

Total Population	**Ground-truth Positive**	**Ground-truth Negative**
Predicted Positive	True Positive	False Positive
Predicted Negative	True Negative	False Negative

Then, the classification metrics could be followed:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.11}$$

$$\text{False-positive-rate (FPR)} = \frac{FP}{FP + TN} \tag{4.12}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.13}$$

$$\text{F1} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4.14}$$

$$\text{Accuracy} = \frac{TP + TN}{P + N} \tag{4.15}$$

A notice of Eq. 4.9 is that the ECApA can be negative values if the models predict more positives than the actual ground-truth, i.e. the model estimates more energy compared with the ground-truth of one appliance.

In order to identify positive and negative labels, the discrimination threshold should be defined. This threshold can be determined by receiver operating characteristics (ROC) curve. ROC curve shows the relationship between Recall versus FPR at various threshold settings. If the area under ROC curve (AUC) is close to 1, the classification is close to the perfection ($0.5 \leqslant \mathrm{AUC} \leqslant 1$). Therefore, the threshold can be tuned to obtain the optimal value where AUC is maximized (optimal operating point). Such information of each appliance can be detailed in Appendix B. A simple MATLAB script which returns the optimal discrimination threshold can be described as follows:

```
%tpr: Recall or True Positive Rate
%fpr:False Positive Rate
%auc: Area Under ROC Curve
%Opt: Optimal operating point of the ROC curve
%Y:Estimated (Network Output)
%T:Target (True Label)
[fpr,tpr,thre,auc,opt]=perfcurve(T,Y,’1’);%’1’ is positive
a=find(tpr==opt(2)); %Return the index where AUC is maximized
thd=thre(a(1)); %Optimal Discrimination Threshold
%Define positive and negative of estimated Y
Y(Y>=thd)=1; %Postive
Y(Y<thd)=0; %Negative
```

The main reason why different metrics should be used is that one metric could not provide all information for users. Some customers prefer the information of the daily estimated energy where disaggregation metrics are useful. On the other hand, some users want to know which appliances are currently active where the value of F1 could be more informative.

Chapter 5

Experimental Results

In the first scenario, this chapter shows the disaggregation results from two methods in Chapter 4. In the second scenario, the RNN model makes an attempt to disaggregate the common appliances in unseen houses. This means the training is carried out in one house and the model is tested in the other houses. At the time of writing, house 1, 2, and 4 are considered for testing RNN. This chapter compares the disaggregation result between CO and RNN in those three houses.

5.1 Combinatorial Optimization (CO)

Fig. 5-1(a), 5-2(a), and 5-3(a) shows the disaggregation performance of combinatorial optimization. It is obvious that CO model could well disaggregate refrigerators and lightings. One remark is that such appliances are regular in operation. In particular, the operation of refrigerators is periodic.

On the other hand, the disaggregation of dishwashers is unsuccessful in most cases. The high accuracy scores in dishwasher recognition are due to their high portions of true negative. This means that dishwashers are off for a long time. To be more precise, the CO model is totally unable to detect the ON state of the house-1 dishwasher, where its F1 score returns null.

For the multi-state noisy loads, *i.e.* kitchen outlets, the optimization outputs tend to return higher relative error than the other appliances do. This is due to the difficulty in clustering their exact power states as their state differences are insignificant. As a consequence, the CO model could not well recognize those loads. It is noticeable that kitchen outlets always consume some powers from the household networks.

The loads which consume high power like bathroom GFI, washer dryer, or microwave could be fairly disaggregated with acceptable errors and average F1 score. This is understandable as their non-off power states are quite distinctive. However, we could not state that the disaggregation with CO is successful if the average total energy correctly assigned is only around 45%. For more information, the estimated demand of some appliances and disaggregation of other houses are detailed in Appendix C.

5.2 Recurrent Neural Network (RNN)

5.2.1 Test in Seen Houses

Fig. 5-1(b), 5-2(b), and 5-3(b) presents the classification of RNN. The training is implemented in one house, and the test is done in the same house but in different period from the training set. For the same house, RNN obviously outperforms CO on every appliance on F1, accuracy, total energy correctly assigned and relative error. In particular, RNN is able to detect the on-state of dishwashers while CO is not. Nonetheless, RNN still suffers from inaccurate classification if the kitchen outlets are considered. It is also unexpected that Bathroom GFI is unsuccessfully disaggregated by RNN. Some examples of RNN outputs can be visualized, and RNN well re-constructs the appliance signatures from the aggregated power. This clearly explains why RNN is the potential candidate for disaggregating household appliances. Several examples of RNN outputs can be illustrated from Fig. 5-4, 5-5 and 5-6.

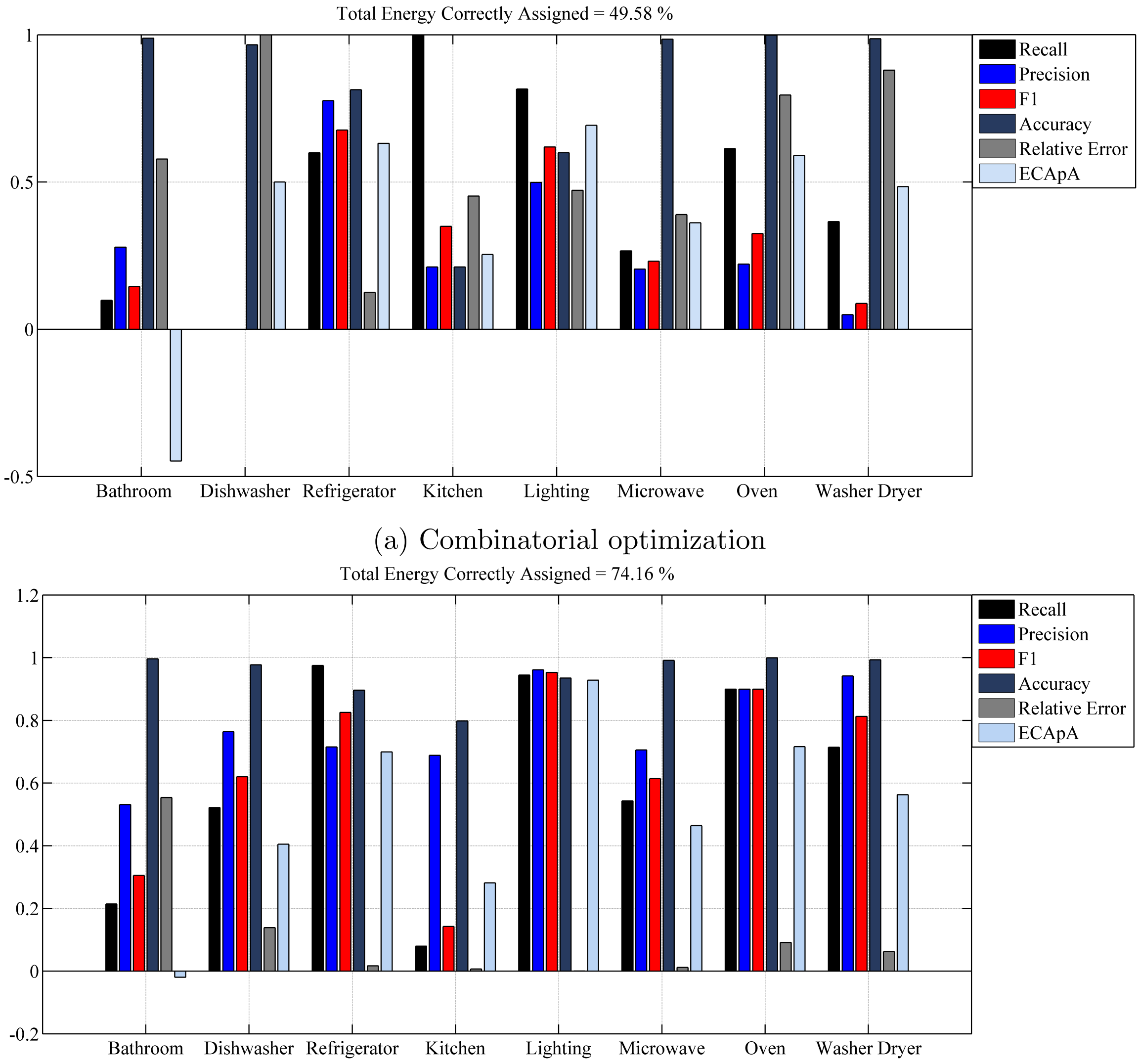
5.2.2 Test in Unseen Houses

In the next circumstance, two common appliances among 6 houses: dishwasher, and refrigerator are selected for the experiment. The data selection for this scenario can be listed in Table 5.1. The purpose is to consider how good the generalization of RNN model is, for the unseen data. Without the prior information of power states, CO could not follow this scenario.

Table 5.1: Data selection to test unseen appliances

Appliances	**Training(3 days/House)**	**Test**
Dishwasher	House 1, 2, 3, 4	House 5
Refrigerator	House 1, 2, 3, 6	House 5

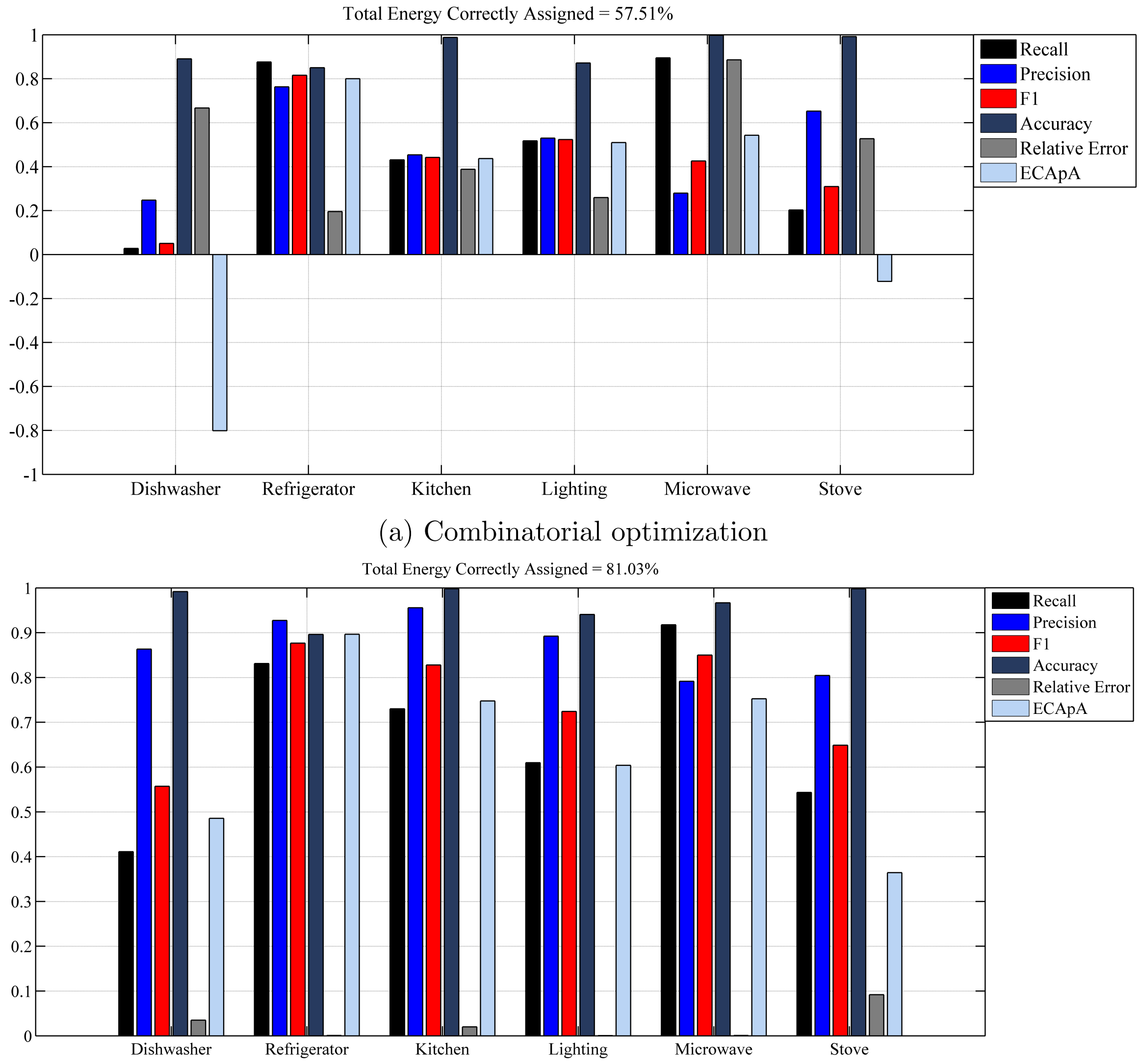
Fig. 5-7 describes the disaggregation performance in this case study. It is noted that the RNN has not succeeded in generalizing the dishwasher where its metrics are too low. Three days per house is possibly insufficient for the RNN model to learn the information in this case. Another remark is that the power in house 5 are recorded within only two days. If the RNN model aggregates the prediction over a longer duration (more test dataset), the errors tend to be reduced.



(a) Combinatorial optimization

(b) Recurrent neural network

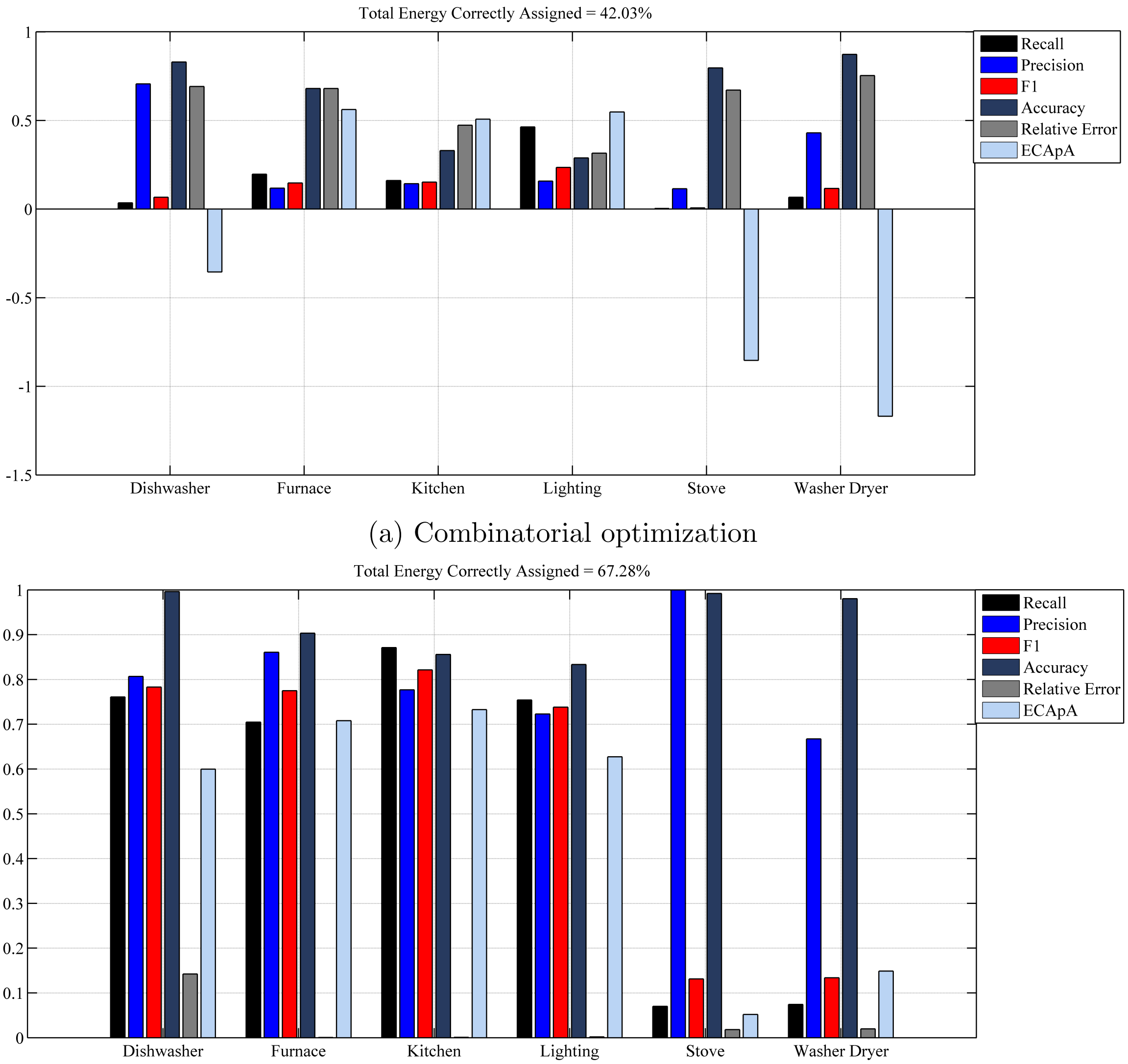
Figure 5-1: Disaggregation result of house 1



(a) Combinatorial optimization

(b) Recurrent neural network

Figure 5-2: Disaggregation result of house 2



(a) Combinatorial optimization

(b) Recurrent neural network

Figure 5-3: Disaggregation result of house 4

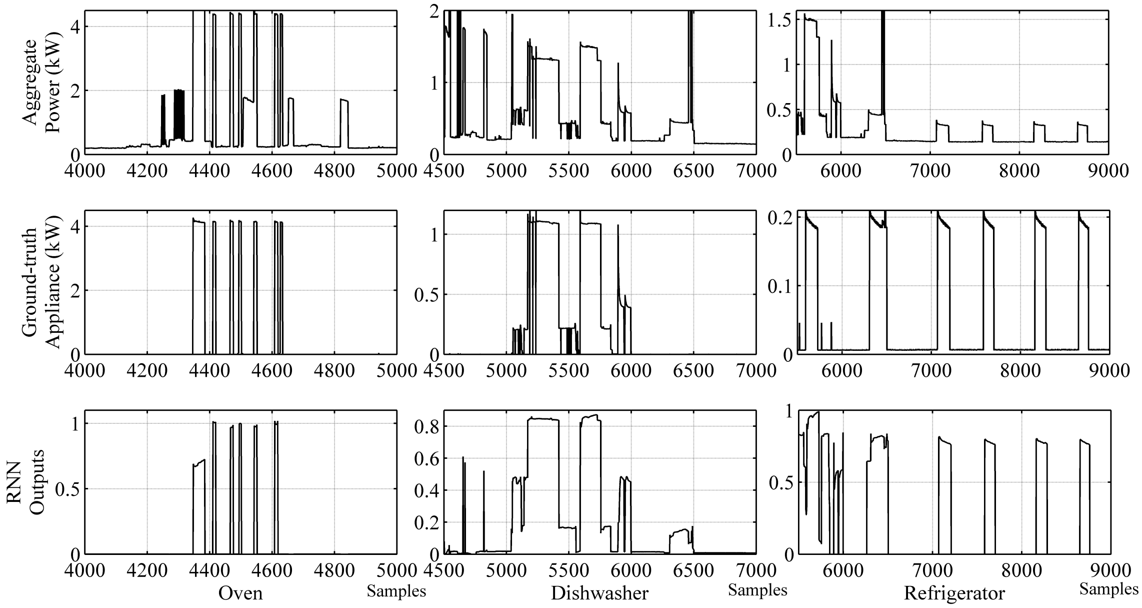


Figure 5-4: Examples of RNN outputs in House 1 on 18/04/2011

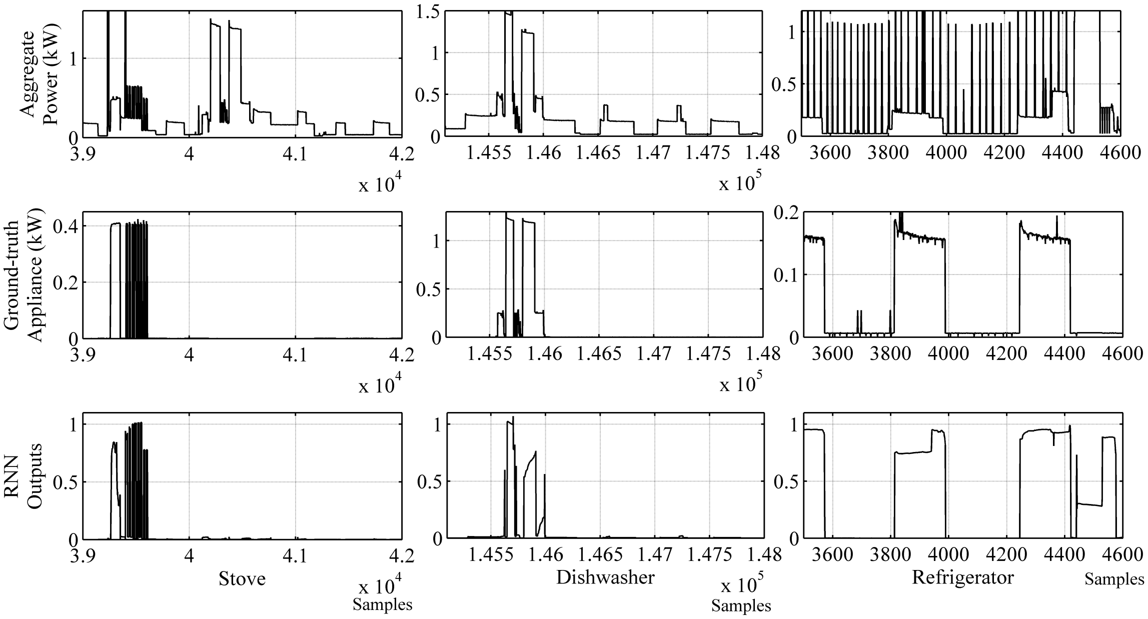


Figure 5-5: Examples of RNN outputs in house 2 on 18/04/2011 (Stove, Refrigerator) and 02/05/2011 (Dishwasher)

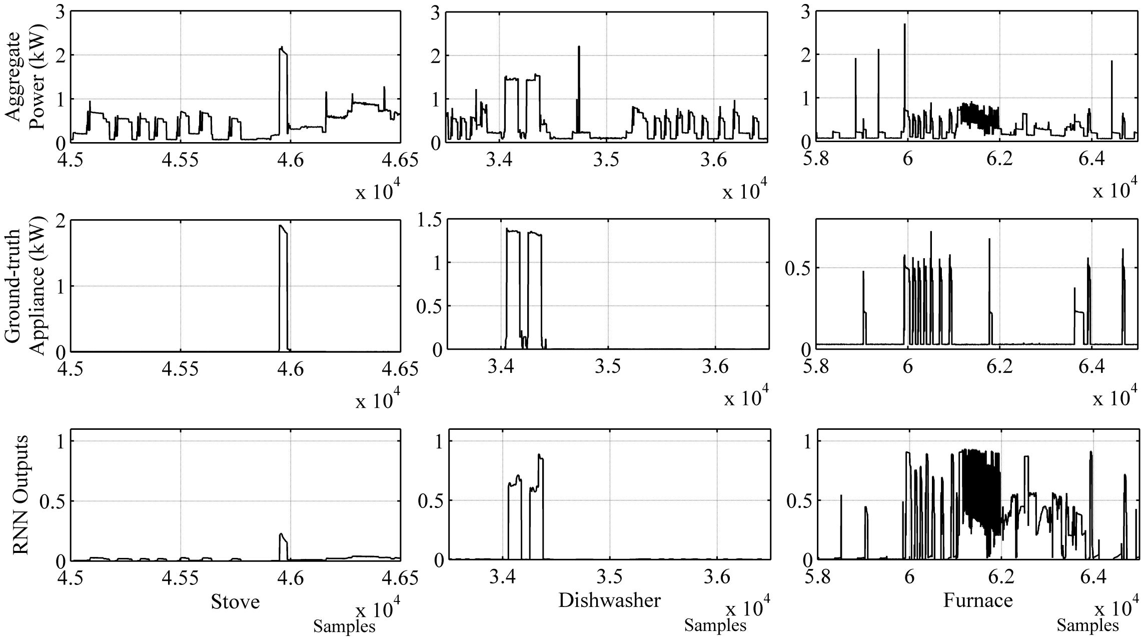
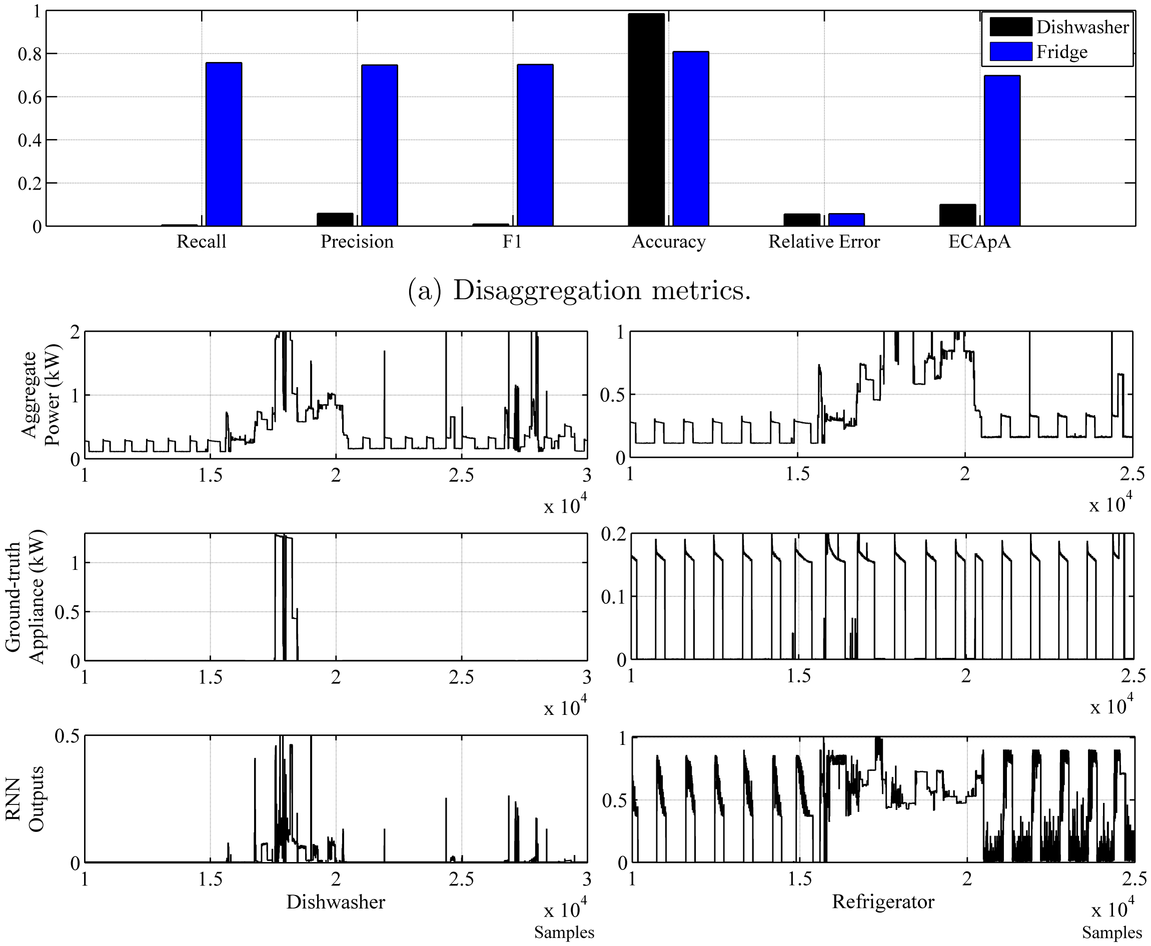


Figure 5-6: Examples of RNN outputs in house 4 on 20/04/2011 (Dishwasher), 21/04/2011 (Stove), and 22/04/2011 (Furnace)



(a) Disaggregation metrics.

(b) RNN outputs on 19/04/2011.

Figure 5-7: Results of RNN in unseen dishwasher and refrigerator in house 5.

Chapter 6

Energy Disaggregation Application to Fault Study

In this chapter which is based on [2], the energy disaggregation technique has been applied to locating the single-line-to-ground fault in distribution network. The aim is to examine the feasibility of using the non-intrusive technique in the fault study. As the most frequent phenomenon in fault study, the single-line-to-ground fault is studied in this chapter.

6.1 Problem Definition

In power system protection analyses, power system faults are one of the most important research topics. These faults which are mainly due to short-circuit phenomena, can drastically affect the operations of power systems. Faults in power systems are divided into three-phase balanced and un-balanced faults. Different types of unbalanced faults are single-line-to-ground fault (SLGF), line-to-line fault (LLF), and double-line-to-ground fault (DLGF). Among the faults of power systems, SLGF is the most frequent type. Due to the serious consequences of fault damage in power systems, many researchers have devoted their attention to this research topic. Chunju *et al*. [37] employed a wavelet fuzzy neural network (WFNN) to extract the fault characteristics from the SLGF signals in an industrial distribution power system. However, this method does not contain the high frequency information of pre-fault currents and only considers SLGF on phase A of Line 1. In fact, the equivalent capacitance and mutual inductance will change when the fault location changes, the charging and discharging currents will be also changed. Hence the high frequency currents, related with the distance from the relay point to the fault location, are different for different fault locations. Ekici *et al*. [38] proposed a SLGF location method simulated in a single 380kV and 360 km long power transmission line. In that paper, the wavelet energy and entropy criterion of the wavelet packet transform (WPT) coefficients are deployed for every faulty current and voltage signal to extract features. Those features are later used for training and testing of ANNs. However, the technique will be unsuitable for a more complicated power system. Reddy *et al*. [39] used the wavelet multi-resolution analysis (WMRA) technique to diagnose the faulty conditions. An adaptive neuro-fuzzy inference system (ANFIS) and ANN in conjunction with GPS are applied to locate the faults when faults occur randomly during the real-time smart grid operation of transmission lines. This paper focuses on the amplitudes of the second- and third-order harmonics generated during fault current occurrence to track the fault location. Among different coefficients pertaining to different decomposition levels, only the summation of the fifth-level detailed coefficients (d5) is considered for the sampling rate of 6 kHz. Borghetti *et al*. [40] built specific mother wavelets inferred from the recorded fault-originated transient waveforms to improve the wavelet analysis for distribution networks. In that paper, the network topology and the traveling wave speeds of the various propagation modes are assumed to be known. However, as concluded in [40], the orthogonalization of transient-inferred mother wavelet is expected to improve the algorithm accuracy by means of proper integration of time-domain fault location approaches. Bezerra Costa [41] has presented a wavelet-based methodology for real-time detection of fault-induced transients in transmission lines, where the wavelet coefficient energy takes into account the border effects of the sliding windows. As a result, the performance of the proposed energy analysis is not affected by the choice of the mother wavelet and presents no time delay in real-time fault detection.

In the trend of energy disaggregation, fault identification, localization and classification have become the new challenge in non-intrusive monitoring system. Even with only one set of voltage and current sensors in ESE, the system must still ensure the effectiveness of protection. Faults should be detected and cleared as fast as possible. As far as the authors are aware, fault identification using non-intrusive monitoring mechanisms in power systems is still in the primary development stage. To enhance this recognition accuracy for a more complicated low voltage distribution system by using non-intrusive fault monitoring (NIFM) techniques, a new fault identification approach in load buses and distribution lines for SLGF is the topic of this paper. Fig. 6-1 illustrates the block diagram for the SLGF diagnoses in NIFM systems. First, the WMRA technique is utilized to detect SLGF occurrence and select the effective transient signals which are the post-fault events. Second, the statistical moment transformation and normalization are employed for feature extraction. In general, the characteristics of any random data can be described by the statistical moments (SMs) [42, 43]. These moments are widely deployed in signal processing, e.g. blind decomposition [44], noise and signal detection [45, 46]. Since these statistical moments can represent the uniqueness of the raw power signatures, the features extracted in this stage could be the good candidates to solve the problem of the fault disaggregation.

In the final step for comparisons, different artificial intelligence (AI) algorithms—BP-ANN, k-NN, and SVM—are used to diagnose the SLGF of distribution lines or load buses for various power signatures in a NIFM system. Among those classifiers, SVM is proved to be the most effective learning tool for these case studies of this work. By using the proposed methods (SK and SVM), the number of transient signal features can be reduced without losing its fidelity. The accuracy rate of the proposed methods is tested in a model system and simulated by using the Electromagnetic

Transient Program (EMTP) software. These results show that the proposed methods can analyze the signals efficiently and effectively, thus enhancing the performance of the fault identification in the NIFM system.

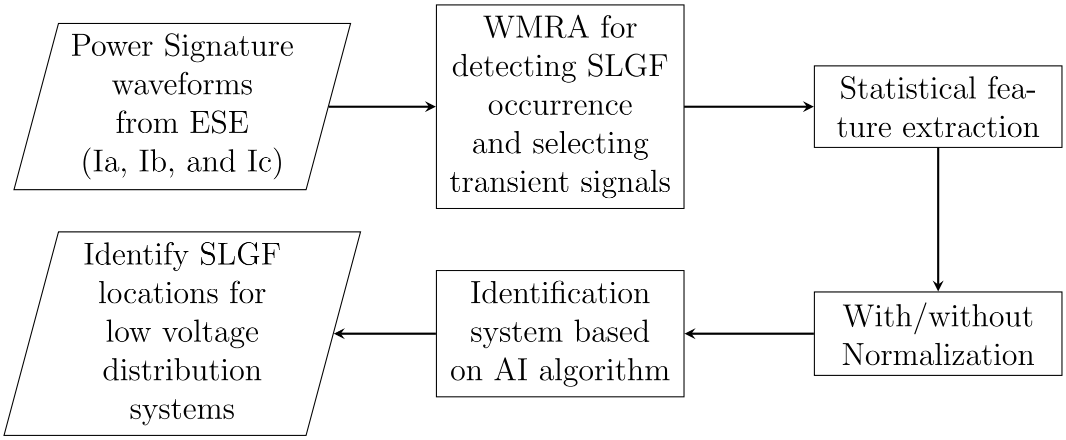

Power Signature waveforms from ESE (Ia, Ib, and Ic)
WMRA for detecting SLGF occurrence and selecting transient signals
Statistical feature extraction
With/without Normalization
Identification system based on AI algorithm
Identify SLGF locations for low voltage distribution systems

Figure 6-1: Block diagram of a non-intrusive power fault identification system (Adapted from [2]). ESE: electrical service entry; SLGF: single-line-to-ground fault; and WMRA: wavelet multi-resolution analysis.

6.2 Data Acquisition

Fig. 6-2 shows a typical NIFM system for the experimental case studies in this paper. Based on the statistical feature extraction, the identification algorithm in the NIFM system recognizes the distribution-line faults and load-bus faults of five 220-V loads measured on a three-phase 480-V common bus These loads include a 45-hp induction motor, a 55-hp induction motor, a 30-hp induction motor, a 20-hp induction motor driven by line frequency variable voltage drives, and a load bank supplied by a six-pulse thyristor rectifier for AC power. In this scheme, the smart meter/NIFM manages and measures the operations and power demands of each load. Then, this information of each fault event is sent to the meter data management system (MDMS) via wireless sensor communication. Through the Internet or Web systems, the client terminal can access these load data and fault events.

Experimental data sets were generated by simulating the data on the current waveforms at the ESE. Each final sample consists of $(T \times 60 \times 256)$ data samples obtained from 256 sampling points for one cycle over a period of T in a 60 Hz power system. Each power signature example includes a voltage variation from 3% to +3% at 0.5% intervals. This yields 13 samples of each power signature for each bus and $N \times 13 \times 3$ raw data for all power signatures (I_a, I_b, and I_c) where N is the number of buses in a three-phase power system network. The full input data set comprises a $(N \times 13 \times 3) \times (T \times 60 \times 256)$ matrix as the experimental data sets. Basically, the fuse or circuit breaker must interrupt the fault quickly (generally less than 4 ms) in order to provide the maximum protection for equipment and personnel [47]. Underwriters Laboratories Inc. (UL, Northbrook, IL, USA) defines breaker current limitation as a breaker that interrupts and isolates a fault in less than 1/2 of an AC cycle. 1/2 a cycle is completed in 8.3 ms [48]. Fig. 6-3 shows the current and voltage waveforms simulated on the ESE when the line 2 of the case study in Fig. 6-2 has SLGF on the phase A.

6.3 Fault Detection

First, the wavelet transform (WT) technique of multi-resolution analyses is utilized to detect the time of SLGF occurrence in power systems. The post-fault signals are preprocessed before inputting the data into AI identification algorithms. The WT is a powerful tool for studying transients by characterizing band-pass filters [49, 50]. Therefore, it is suitable to detect the occurrence of faults in power systems. WT decomposes power signatures into different scales of WT coefficients (WTCs). WT can be grouped into continuous WT (CWT), discrete WT (DWT), stationary WT (SWT) and wavelet packet analysis. The CWT is expressed as:

$$C_{a,b}(t) = \int_{-\infty}^{\infty} \psi_{a,b}(t) x(t) dt \tag{6.1}$$

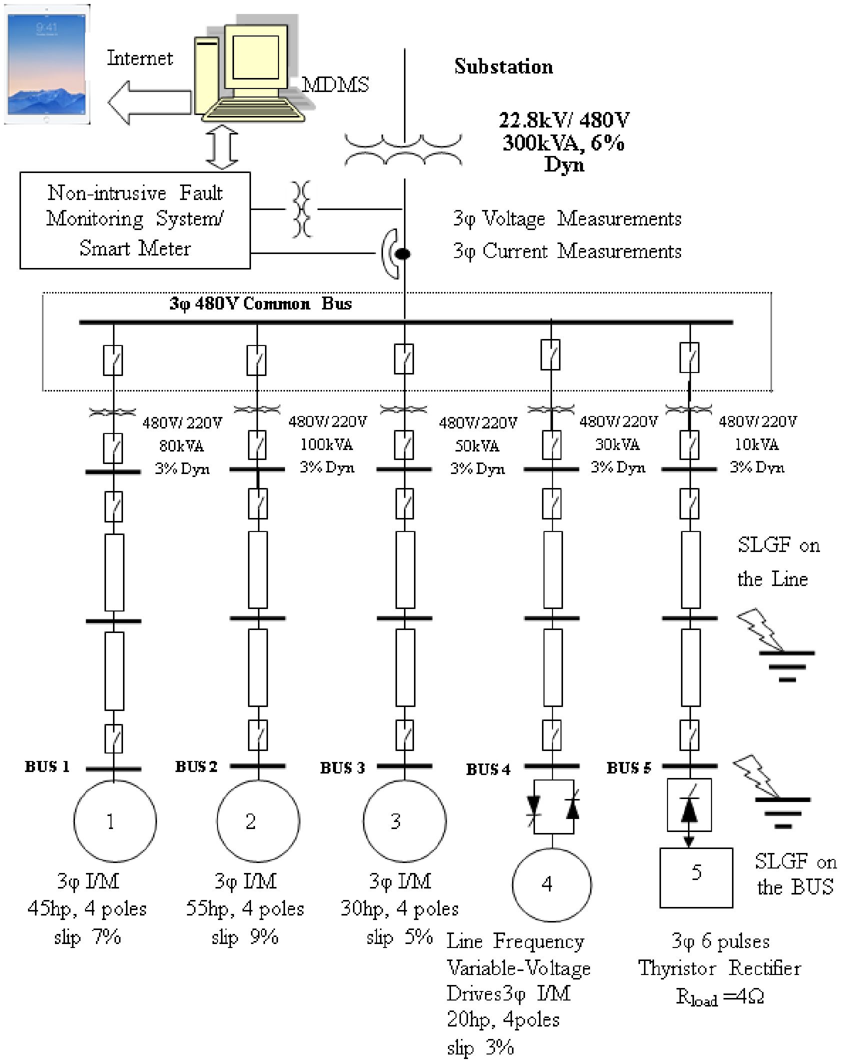


Figure 6-2: Power fault identification system for a NIFM system [2]. MDMS: meter data management system.

where $x(t)$ is the original signal and $\psi_{a,b}(t)$ is the daughter wavelet. The daughter wavelet is defined by:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right);\ a \in R^{+},\ b \in R \qquad (6.2)$$

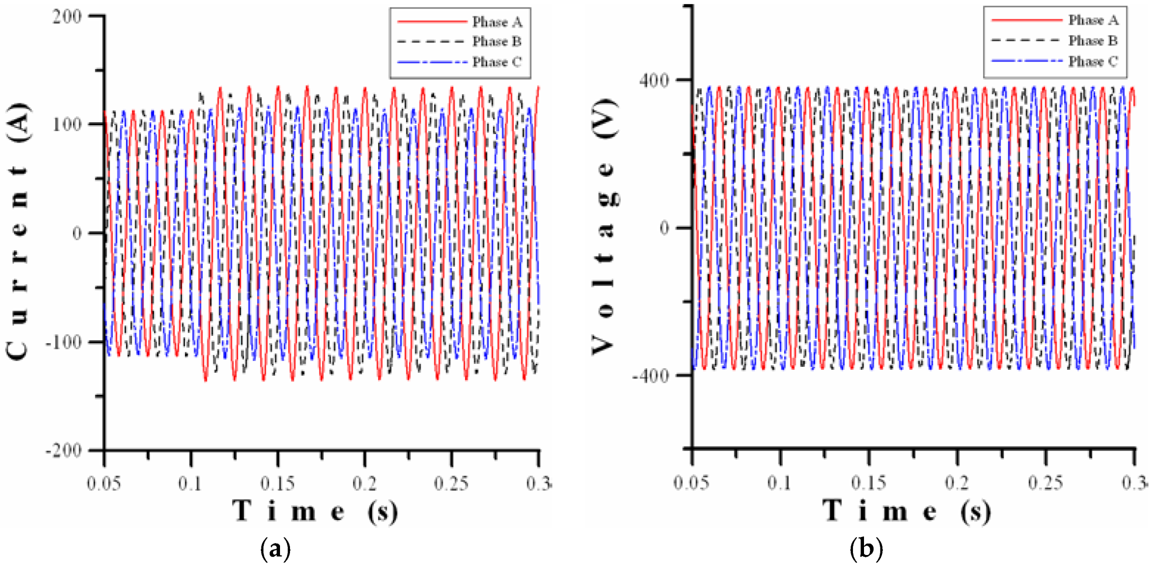


Figure 6-3: Current and voltage waveforms of SLGF on the phase A for the distribution line 2 of the case study [2]. (a) Current waveforms; and (b) voltage waveforms

where $\psi(t)$ is the chosen mother wavelet, a is the scaling factor, and b is the shift factor. In the discrete case, the scaling and shift factors are represented as follows:

$$\begin{aligned} a &= a_0^m \\ b &= nb_0a_0^m \end{aligned} \tag{6.3}$$

Then, the DWT is obtained as:

$$D_{m,n} = {a_0}^{-\frac{m}{2}} \sum_k x[k]\psi[{a_0}^{-m}k - nb_0];\ a_0 > 1;\ b_0 \neq 0 \tag{6.4}$$

where m is a scaling index; n is a sampling time point, for n = 1, 2, ..., N; and N is the number of sampling points; $m, n, b_0 \in Z$, Z is a set of integers, and k is an operating index; a_0 is selected as a spacing factor. The dyadic grid is perhaps the simplest and most efficient discretization for practical purposes and lends itself to the construction of an orthonormal wavelet basis.

The WMRA technique is an existing and well-known method in fault detection applications [37–39, 41, 51–53]. The timefrequency location feature of the WT makes it suitable to be used in fault detection method based on high and fundamental frequency components. This research uses the MRA technique to decompose the current signals into some approximation and detail levels of resolution. By analyzing certain levels of approximation and details, the stages of fault detection are processed. Based on [52], the occurrence (T_{fault}) of SLGF is detected.

Fig. 6-4 shows the fault-detection procedure, where IF is a counter that expresses the sample number under SLGF. SUM_d1 is the sum value of the detailed output (d1 coefficients) for a one-cycle period and is represented as an absolute value. Fault criterion (FC) is the signal magnitude threshold as the lower limit of SUM_d1, while Ns is the sample number that signifies the duration time for which a transient event has to persist continuously. When SUM_d1 is greater than or equal to FC, the value of IF is incremented and as soon as it attains the level Ns. This indicates an internal fault, and a trip signal is initiated. As shown in Fig. 6-4, the sampling rate is 15.360 kHz; i.e., 256 samples/cycle at 60 Hz. The summated values associated the three phases are compared with a preset threshold level FC. The whole process is based on a moving window approach where the one-cycle window is moved continuously by one sample. In these cases of the paper, T_{fault} is 0.1 s. The optimal settings for FC and Ns are 0.085 and 128, respectively.

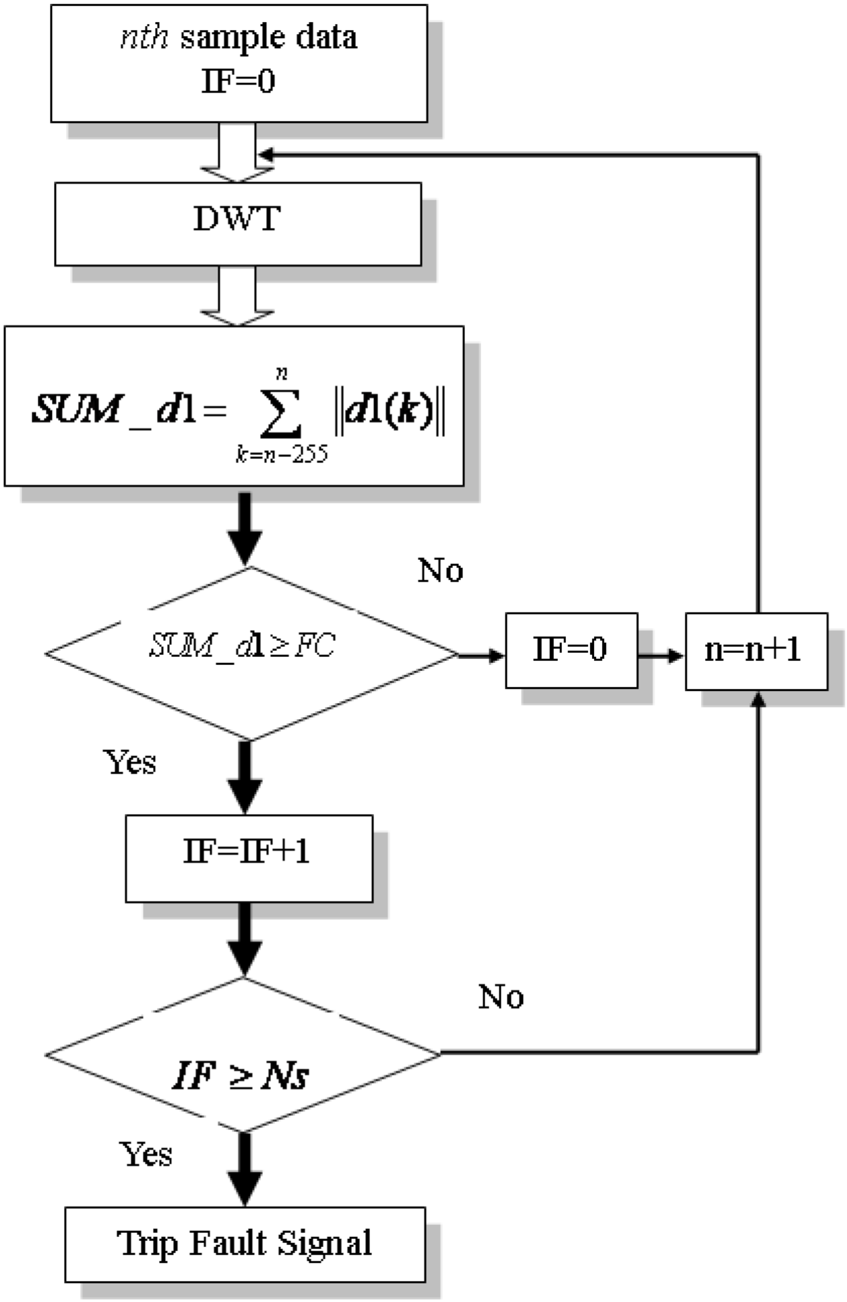


Figure 6-4: Block diagram of fault detection [2].

6.4 Proposed Methods

6.4.1 Statistical Moments

It is generally not possible to directly identify SLGF in power systems based upon these complicated raw waveforms. Thus, some transformations of the recorded SLGF time-series are implemented to extract the time-invariant features. These are statistical moments (SMs). Those transformations can also reduce the vulnerability and the variation of the datasets. In the faults, the energy (mean square value) in the fault signals is expected to increase. The features of nth-order SMs are generated from the raw fault data, which is defined by Eq. 6.5. The SMs used here are the four (first to fourth order) moments, stored in a matrix of size (65 × 4) for each power signature. Specially, the first and second order moments are the mean and the variance of the fault signals, respectively:

$$f^{(n)} = \frac{1}{N}\sum_{i=1}^{N} x_i^n \tag{6.5}$$

for $n = 1$, 2, 3, and 4.

6.4.2 Z-Scores

Normalization plays a crucial rule in preprocessing, as proven by Bishop [54]. This normalization works well for populations that are normally distributed. To normalize, the input variable of statistical moments f_{old} is converted into zero mean and unit variance, the new input variable of statistical moments f_{new} is called Z-scores or standard score normalization. The formula is illustrated by Eq. 6.6:

$$f_{new}^{(n)} = \frac{f_{old}^{(n)} - \mu^{(n)}}{\sigma^{(n)}} \tag{6.6}$$

for n = 1, 2, 3, and 4, where μ and σ are the mean value and the standard deviation for the feature vector of the statistical moments f_{old}, respectively.

6.4.3 Skewness and Kurtosis

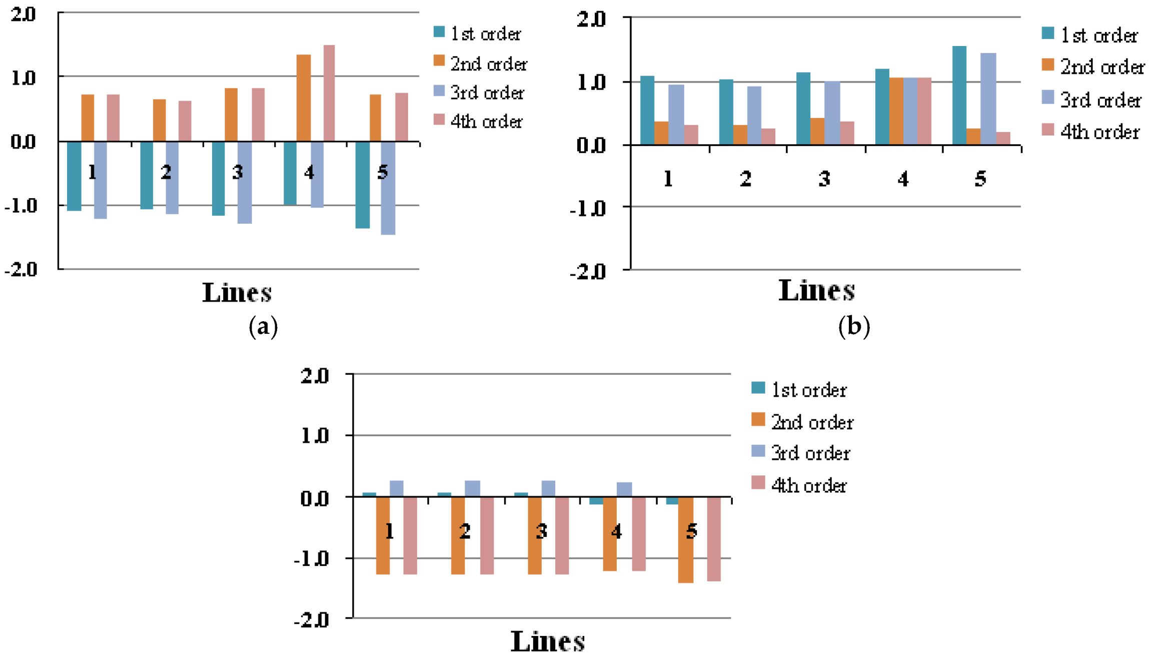
Another method of normalization is skewness and kurtosis (SK) or standardized central moments, using standard deviation as a measure scale. The nth-order standardized central moments are defined by Eq. 6.7:

$$f_{new}^{(n)} = \frac{1}{N\sigma^{(n)}} \sum_{i=1}^{N} \left(f_{old}^{(n)} - \mu^{(n)}\right)^{n} \tag{6.7}$$

for n = 1, 2, 3, and 4.

Eq. (9) shows that the 1st and 2nd standardized central moments are constants, with values of 0 and 1, respectively. This implies that these constants are not necessary for inputs of AI algorithms. In other words, skewness (normalized 3rd central moment) and kurtosis (normalized 4th central moment) are the sufficient features for fault identification. These moments are reduced into a matrix of size (65 × 2) for each power signature. To examine the effects of this normalization, the results of SLGF identification were compared between normalized and un-normalized statistical moments.

By examining the current distributions of energy from orders for SLGF on different five lines for different proposed methods, i.e., SMs, Z-scores, and SK, in Fig. 6-5, Fig. 6-6, and Fig. 6-7, respectively, some characteristics can be found. Firstly, the distributions of energy among Line 1, Line 2 and Line 3 are always in similar phases. There are only some minor differences in term of their magnitudes. This is understandable as Load 1, Load 2, and Load 3 are the same type of loads which are induction machines (IMs). Load 4, however, is the IM with the accompaniment of the line frequency variable-voltage drives; while Load 5 is the load bank supplied by a six-pulse thyristor rectifier for ac power. For this reason, Line 4 or Line 5 forms significantly different patterns compared with the remaining four lines, and can be easily discriminated. To distinguish the faults which have the similar patterns, it is necessary to combine with the powerful tool of classification algorithms even though these statistical methods can effectively extract features and reduce the size of datasets.



(a)

(b)

(c)

Figure 6-5: Current distributions of each order for the SLGF on different five lines by using the SMs [2]. (**a**) Phase A; (**b**) Phase B; and (**c**) Phase C.

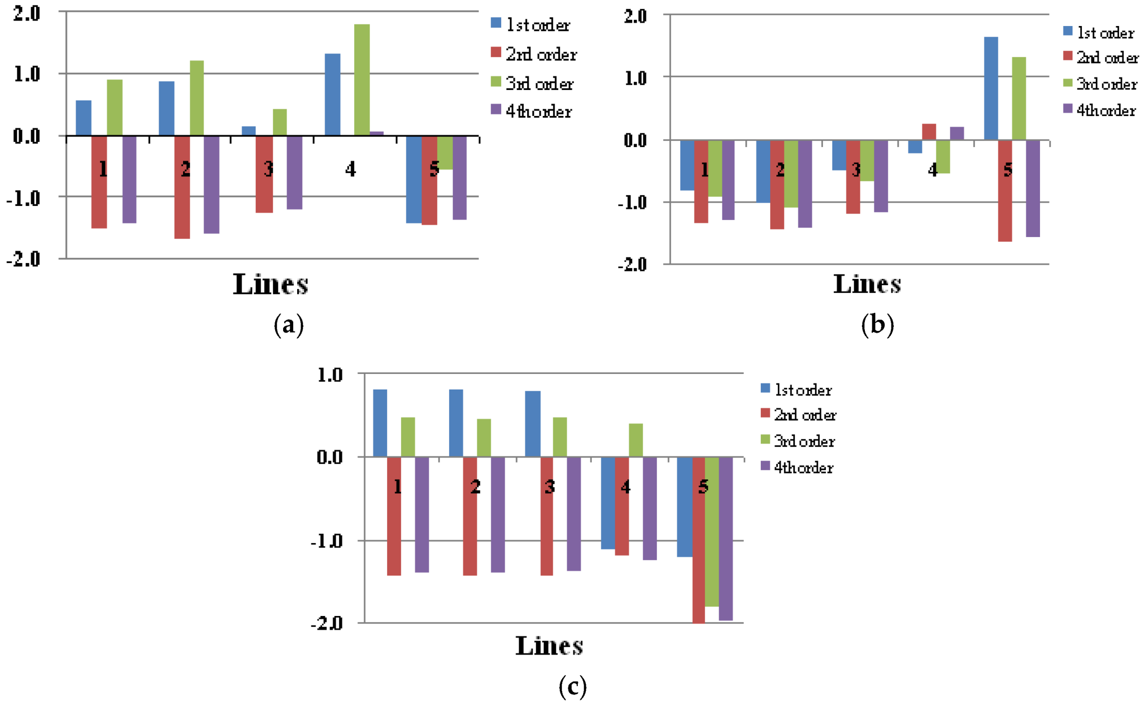


Figure 6-6: Current distributions of each order for the SLGF on different five lines by using the Z-scores [2]. (**a**) Phase A; (**b**) Phase B; and (**c**) Phase C.

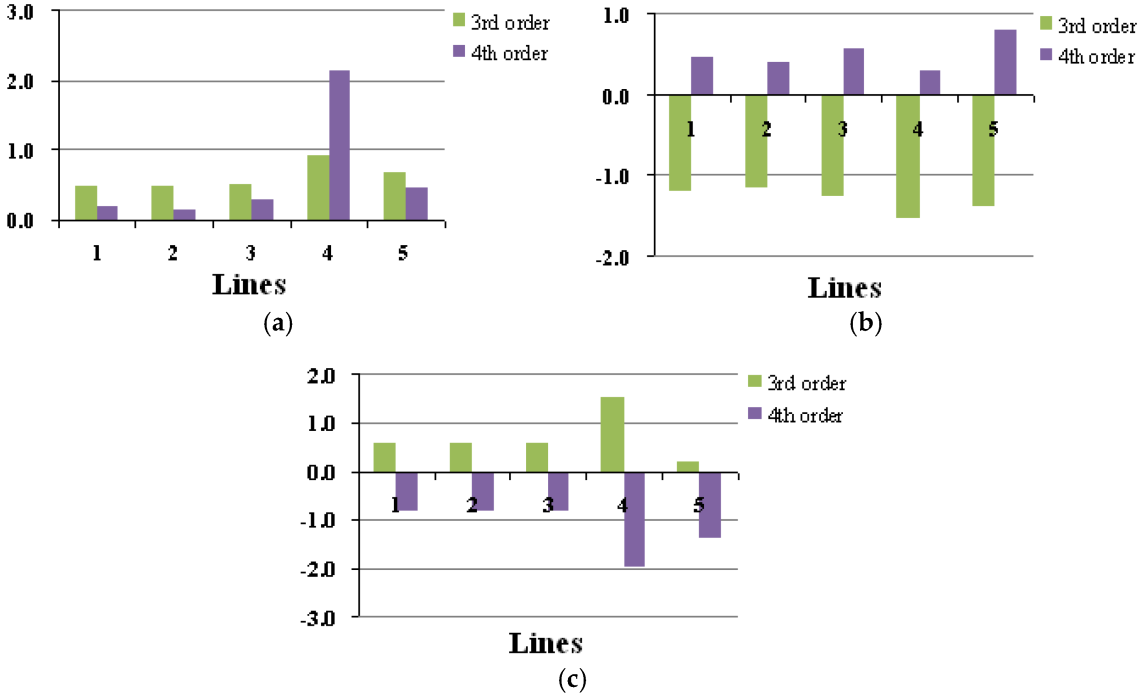


Figure 6-7: Current distributions of each order for the SLGF on different five lines by using the SK [2]. (**a**) Phase A; (**b**) Phase B; and (**c**) Phase C.

6.4.4 SVMs

Support vector machines (SVMs) are used as intelligent tools for identifying faulty lines and buses that is finding the location with respect to the ESE. They were first introduced by Vapnik [55] on the basis of structural risk minimization principle. The aim of SVM is to create a line of hyper-plane among different datasets for identification. The hyper-plane is defined by subsets of the training data called support vectors. Support vectors can create complex boundaries and maximize the margin separation through the quadratic minimizations. The SVM is able to solve both linear and nonlinear identification problems.

The implementation of SVM multi-class classification is probably the one-against-all (one-vs-rest) method. It constructs k SVM models where k is the number of classes. The mth SVM is trained with all of the examples in the mth class with positive labels, and all other examples with negative labels. Thus, given l training data (x_1, y_1), ... , ($\mathrm{x}_{l,}$ y_l), where $\mathrm{x}_i \in R^n$, i = 1, ... , l, $y_i \in \{1, \ldots, k\}$ is the class of x_i. The hyper-plane of the mth SVM is optimized to solve the following problem. The first term in the objective function represents the model complexity and the second term represents the model accuracy (i.e., classification error in the training data):

$$\begin{gathered}\min_{w^m, b^m, \xi^m} \tfrac{1}{2} \left(w^m\right)^T \omega^m + \mathrm{C}\sum_{i=1}^{l} \xi_i^m \\ \left(w^m\right)^T \phi(x_i) + b^m \geq 1 - \xi_i^m, \text{ if } y_i = m, \\ \left(w^m\right)^T \phi(x_i) + b^m \leq -1 + \xi_i^m, \text{ if } y_i \neq m, \\ \xi_i^m \geq 0, \; i = 1, \ldots, l\end{gathered} \tag{6.8}$$

where x_i is the ith training data which are mapped to a higher dimensional space by the function ϕ and C is the penalty parameter. y_i is the class label value which is either +1 or −1.

Then the largest value of the decision function is:

$$\arg\max_{m=1,\ldots,k}((w^m)^T\phi(x)+b^m) \tag{6.9}$$

The basic concept of SVM is to search for a balance between the regularization term $\frac{1}{2}\left(w^m\right)^T\omega^m$ and the training errors. Practically the Lagrangian dual problem of (10) is solved. Hence k l-variable quadratic programming problems are solved as follows:

$$\begin{aligned}&\max f(\alpha_1\ldots\alpha_n)=\sum_i\alpha_i-\tfrac{1}{2}\sum_i\sum_j\alpha_i\alpha_j \mathrm{y}_i y_j k\left(x_i x_j\right)\\&\text{s.t. o}\leq\alpha_i\leq C\forall i,\sum_i\alpha_i y_i=0\end{aligned} \tag{12}$$

The merit of SVM which is the inner product in the feature space by using a kernel function $k(x_i,x_j)$ is that it ties to make the training data linear-separable in the high dimension feature space, thus achieve nonlinear-separable in the input space. Typical choices of kernel function include the follows:

(1) Polynomial kernel: $k(x_i,x_j)=(\gamma(x_i^T\cdot x_j)+r)^d$;

(2) Gaussian kernel: $k(x_i,x_j)=\exp\left(-\gamma||x_i-x_j||^2\right)$;

(3) Sigmoid kernel: $k(x_i,x_j)=\tanh(\gamma(x_i^T\cdot x_j)+r)$.

where d is the exponent of polynomial function; $\gamma>0$, $d>0$ and r are kernel parameters.

It is obvious that Gaussian kernel function is the simplest model with the fewest hyper-parameters. Hence, Gaussian kernel function is reasonably used in this paper. To obtain a good performance, some parameters in SVM have to be chosen carefully. These parameters include:

- The regularisation parameter C, which determines the trade-off between minimizing the training error and minimizing model complexity. The higher the value of C is, the narrower the separating margin of hyperplane is. In contrast, the hyperplane could easily misclassify more feature points if C is too low.

- Parameter γ of the kernel function that implicitly defines the nonlinear mapping from input space to some high-dimensional feature space. The higher value of γ increases the complexity of the classification model, easily leading to overfitting. On the other hand, the lower γ may cause the under fitting error of classification.

In this paper, those parameters are tuned and configured during experiments. The Gaussian kernel function has been implemented with varying values for $C \in [0, 500]$, $\gamma \in [0, 1]$. The optimized values are 200 and 0.4 for C and γ, respectively.

In the fault location of this paper, the full input dataset can respectively form a matrix with a size of (65 × 4) and (65 × 2) for each fault current. From the same input datasets, a matrix with the same size is additionally created for all three fault currents in the case of faulty phase detection. The data in those matrices are randomly divided into two equal sets. One set of the data is for training and the other is for testing. To confirm the inferential power of SVM, the input dataset is created from the raw signature waveforms by different proposed methods with SMs, Z-scores, or SK for comparisons.

The proposed protection scheme is depicted as in Fig. 6-8. The one-vs-rest algorithm is applied for this binary classification. To illustrate, one class is set to be positive while the rests are negative, and the line/phase suffered from SLGF can be distinguished from the other lines/phases. Hence, this problem is divided into five sub-problems of binary classification. For example, two bits 0 and 1 respectively represent the negative and positive class. In the case study for the fault location, a set of five distribution lines represents a five-category attribute in the dataset. Those five classes indicate which line has a SLGF. A SLGF at Line 1, 2, 3, 4, or 5 can be respectively assigned as (1, 0, 0, 0, 0), (0, 1, 0, 0, 0), (0, 0, 1, 0, 0), (0, 0, 0, 1, 0), or (0, 0, 0, 0, 1). In the case study for faulty phase detection, phase A, B, and C can be set as (1, 0, 0), (0, 1, 0), and (0, 0, 1), respectively.

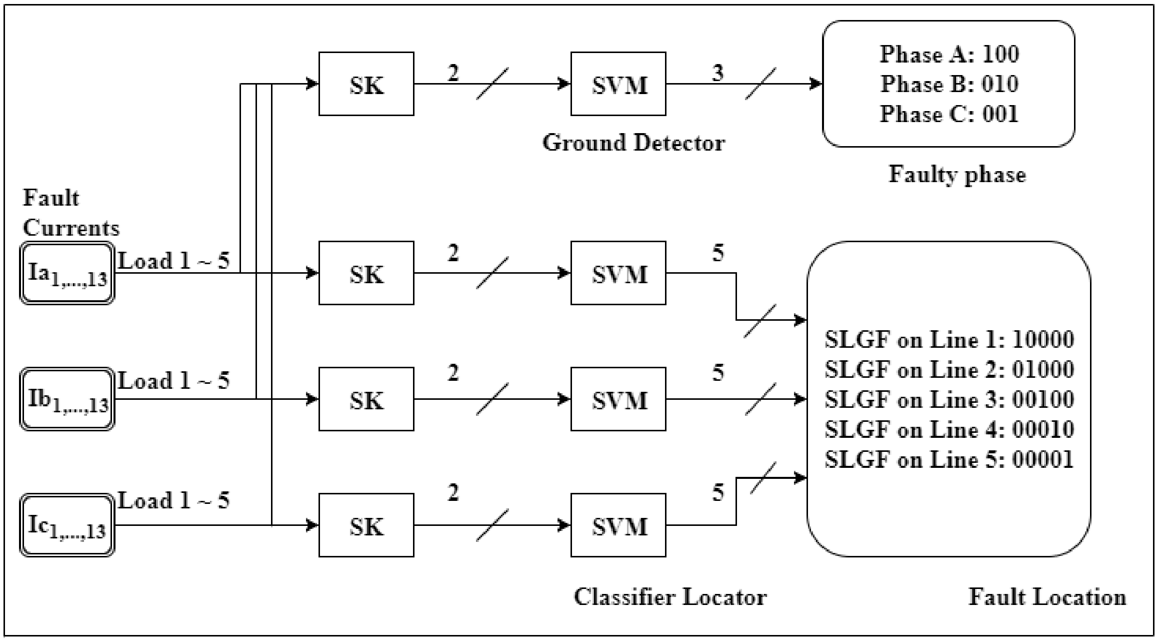


Figure 6-8: Protection scheme for proposed methods. SK: Skewness and Kurtosis; and SVM: support vector machine.(Adapted from [2])

6.5 Experimental Results

The inputs of AI identification algorithms are three different power signatures, currents of three phases (I_{a}, I_{b}, I_{c}) pre-processed by the proposed extraction methods. Besides SVM, two other popular algorithms BP-ANN and k-NN ($k = 1$) are also implemented for comparison. Those identification algorithms are carried out by HeuristicLab (Ver. 3.3, Heuristic and Evolutionary Algorithms Laboratory (HEAL), Hagenberg, Austria) [56]. The programs were run to identify SLGF events for two case studies on a personal computer equipped with a 2.2-GHz Intel Core i3-2330M central processing unit. One is distribution-line faults; the other is load-bus faults. Each entry in the AI algorithms represents 100 different trials. The identification results are compared between the AI algorithms and power signatures for the proposed methods. A simulated NIFM system as shown in 6-2 has five different loads on different buses. Those loads include a 45-hp induction motor, a 55-hp induction motor, a 30-hp induction motor, a 20-hp induction motor driven by variable-voltage drives, and a bank of loads supplied by a six-pulse thyristor rectifier for AC power.

6.5.1 Case Study 1, Distribution-Line Faults

In case study 1, the AI algorithms in the NIFM system identify SLGF on phase A at five different 220-V distribution lines from different time based upon the proposed methods, as shown in Fig. 6-2. The fault SLGF occurs on a specific single distribution line of the five distribution lines, the other distribution lines operate normally. There are essentially five 220-V distribution lines in the model system of Fig. 6-2 because the distance between the 480 $V/220\ V$ distribution transformer and the load bus is long.

Each line is 3 km long and there are two sections per line each section being 1.5 km in length. This allows the user to apply faults at the section junctions. The lines are modelled using a constant parameter line model in EMTP program. The line conductor is a 1 AWG with an 8.4 millimeter diameter and a dc resistance of 0.5426 Ω/km at 25 $^{\circ}C$. The line parameters are calculated at 60 Hz with an earth resistivity of 100 Ω-m. In the case study of distribution-line faults where each distribution line has SLGF on phase A, the results for different AI algorithms and three extraction methods including SMs, Z-scores, and SKs are shown in Fig. 6-9 In general, SVM obtains the most impressive results compared with the others, regardless whether any extraction method is used. On the other hand, k-NN ($k = 1$) has the unsatisfactory outcome where all average values of the test accuracy are below 72.22% in the case of SMs and Z-scores. Also, BP-ANN which does not exceed 84.35% in the average test accuracy could not well perform in SMs and Z-scores. Furthermore, the Z-scores normalization (SK) significantly improves the capability of disaggregation in 1-NN and SVM, but BP-ANN does not gain any benefit of this normalization. The approach of using the SK has taken a further result in the recognition accuracy. Most of results achieve visible improvement. Those achievements show that almost all cases obtain 100% in recognition accuracy. There is an exception in the case of BP-ANN.

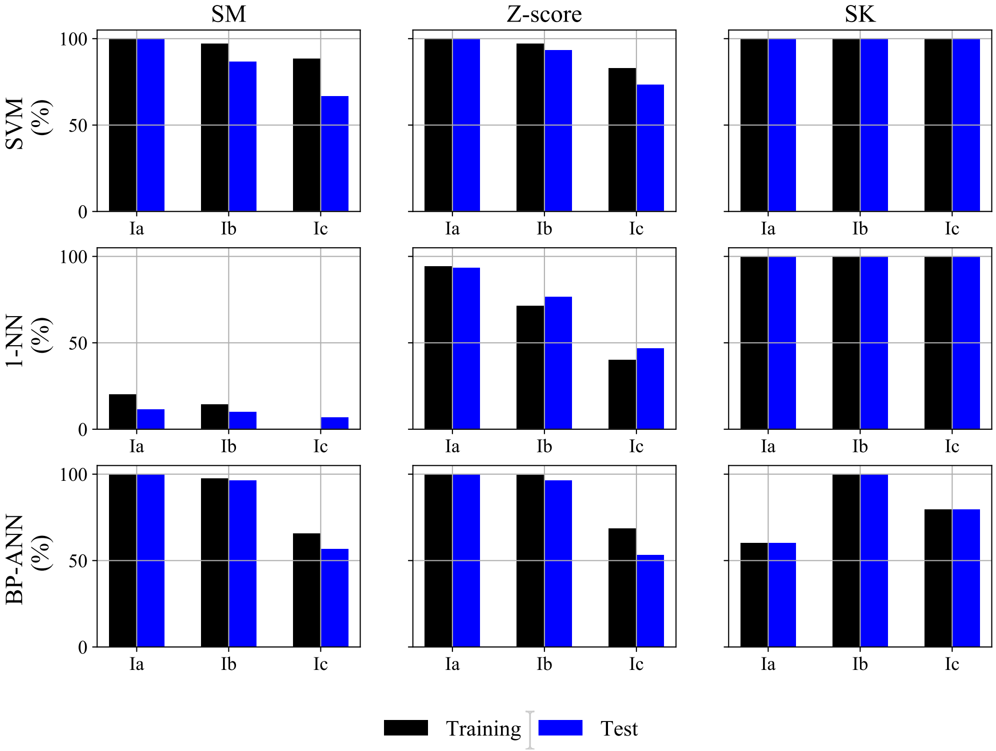


Figure 6-9: Recognition Accuracy (in %) of SLGF on the distribution lines (Adapted from [2]).

Another remark, from Table 6.1, is that the average simulation time if BP-ANN is used BP-ANN requires much more execution time compared with SVM and 1-NN. On average, the BP-ANN model needs around 5 seconds for 100 trials. Meanwhile, the execution times of the SVM and 1-NN models are approximately 0.3 and 0.1 seconds, respectively. In term of phases, the inputs from phase A obviously achieve the best outcome than those of phase B and C in the case of SMs and Z-scores when distribution-line faults occur on phase A. This can be shown in Table 6.2.

Table 6.1: Execution time of 100 trials (in second) for different algorithms in line fault detection (Adapted from [2]).

	SMs			**Z-score**			**SK**		
Phase Current	I_a	I_b	I_c	I_a	I_b	I_c	I_a	I_b	I_c
SVM	0.2813	0.3125	0.2969	0.2857	0.2592	0.5238	0.2344	0.1343	0.0782
1-NN	0.1111	0.1071	0.1070	0.1172	0.0993	0.1135	0.0625	0.0781	0.0781
BP-ANN	5.6908	5.3066	0.1200	4.8694	5.7919	3.1896	7.0159	8.2034	3.5001

Table 6.2: Confusion matrix for faulty phase detection using SK and SVM in case study 1 (Adapted from [2])

Total Population	**Ground-truth**		
Predicted	**Phase A**	**Phase B**	**Phase C**
Training phase A	33	0	0
Training phase B	0	33	0
Training phase C	0	0	33
Test phase A	32	0	0
Test phase B	0	32	0
Test phase C	0	0	32

6.5.2 Case Study 2, Load-Bus Faults

In case study 2, the AI algorithms in the NIFM system identify SLGF on phase A at five different 220-V load buses based upon the proposed methods, as shown in Fig. 6-2. The fault SLGF occurs on a specific one load bus of the five load buses, while the other load buses are operating normally. There are essentially five different loads on different 220-V load buses from five feeders in the model system of Fig. 6-2. The loads include a 45-hp induction motor, a 55-hp induction motor, a 30-hp induction motor, a 20-hp induction motor driven by variable-voltage drives, and a bank of loads supplied by a six-pulse thyristor rectifier for AC power.

Fig. 6-10 shows the results of SLGF location for different AI algorithms and three statistical extraction methods are obtained when SLGF occurs on phase A of each load bus. Obviously, SVM remains the best classifier compared with the others, regardless of whether any extraction method is used. In k-NN ($k = 1$), the average values of test accuracy are below 50% in the case of SMs and Z-scores. On average, BP-ANN does not surpass 78.78% in the test recognition accuracy when the inputs are from SMs and Z-scores. BP-ANN also requires more computer resources compared with the two others, as shown in Table 6.3. As the good features, SK maximizes the capability of disaggregation in all algorithms.

In term of phases for this case study, the inputs of phase A do not achieve better outcome than those of phase B and C in the case of BP-ANN and 1-NN when load-bus faults occur on phase A, regardless whether any extraction method is used. Table 6.4 shows the performance of using SK and SVM applied for faulty phase detection in the case of load-bus faults, which again verifies the high accuracy 95.95% in training and 90.625% in test. This means that SKs can extract the distinction among phases even when SLGF occurs. To summarize case studies 1 and 2, the currents monitored in phase A gives better distinctive features than those measured in phase B and C when SVM is used and SLGF occurs on phase A, regardless whether any extraction method is used. Among classifiers for these case studies, SVM gives the best results. In this proposal, SK does not only reduce the number of inputs for AI classifiers, but it also provides the distinctive features for phase and line disaggregation.

Table 6.3: Execution time of 100 trials (in second) for different algorithms in bus fault detection (Adapted from [2]).

	SMs			**Z-score**			**SK**		
Phase Current	I_a	I_b	I_c	I_a	I_b	I_c	I_a	I_b	I_c
SVM	0.3281	0.1563	0.2656	0.2969	0.2500	0.3438	0.0625	0.1406	0.1406
1-NN	0.2031	0.2344	0.1094	0.09377	0.2188	0.1719	0.0625	0.0313	0.0625
BP-ANN	8.7192	4.5627	5.0471	10.609	4.7033	4.4845	5.5159	6.9378	4.1564

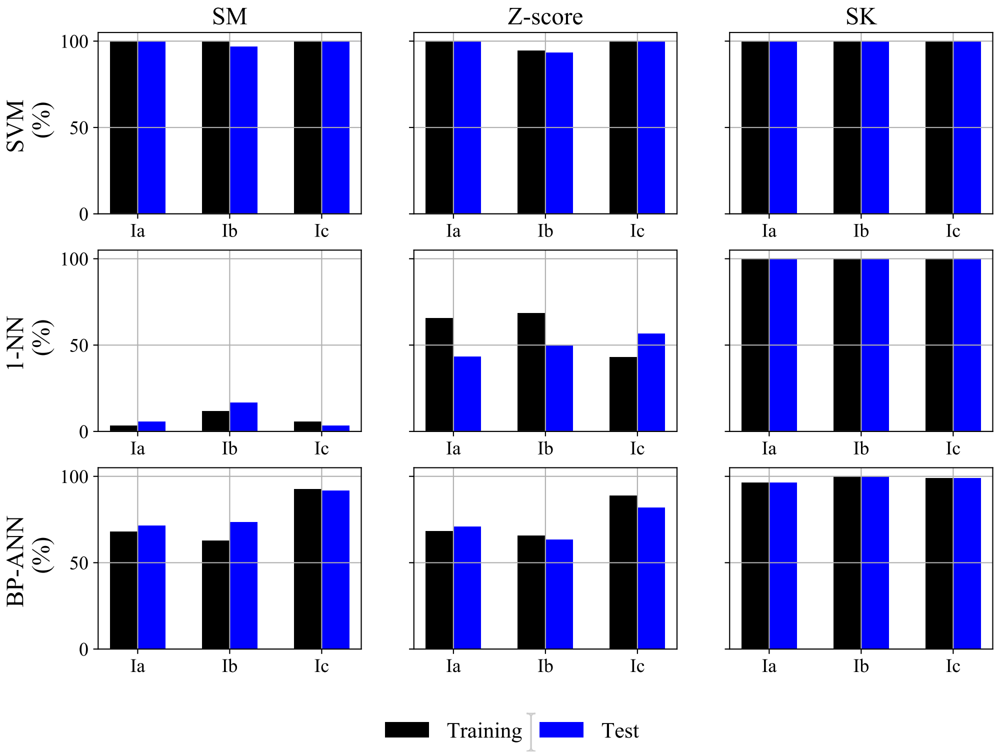


Figure 6-10: Recognition Accuracy of SLGF on the buses (Adapted from [2]).

Table 6.4: Confusion matrix for faulty phase detection using SK and SVM in case study 2 (Adapted from [2])

Total Population	**Ground-truth**		
Predicted	**Phase A**	**Phase B**	**Phase C**
Training phase A	29	0	0
Training phase B	4	33	0
Training phase C	0	0	33
Test phase A	23	0	0
Test phase B	9	32	0
Test phase C	0	0	32

6.6 Discussion and Summary

To verify the superiority of the proposed method (SK) in this section, the proposed method can distinguish between the short-time overload current or inrush current caused by the motor starting and overcurrent caused by the SLGF. Fig. 6-11 shows the current and voltage waveforms simulated on the ESE when the load 2 of the case study in Fig. 6-2 has a motor starting. It is obvious by comparing with Fig. 6-3 and 6-11 that the current waveforms of motor starting are different from those of SLGF. When the motor is starting and the SLGF occurs at a different time, Table 6.5 shows the results of identification for distinguishing between the short-time overload current caused by the motor starting and overcurrent caused by the SLGF on phase A. These currents which are monitored on the ESE are Ia, Ib, and Ic for phase A, B, and C, respectively. The total average value of the recognition accuracy is 100% for both training and test.

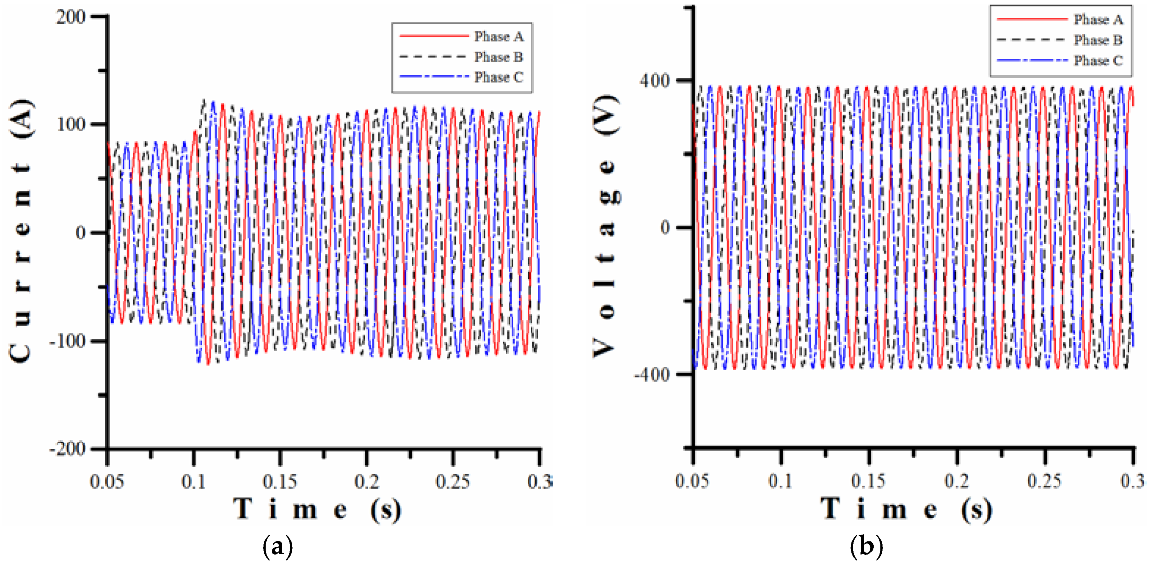


Figure 6-11: Current and voltage waveforms of motor starting for the load bus 2 of the case study [2]. (**a**) Current waveforms; and (**b**) voltage waveforms.

Table 6.5: Results of recognition accuracy (in %) in training and test between SLGF on the line 2 and motor starting current on the load 2 with the SK [2]

Recognition Accuracy (%)	**SVM**	**BP-ANN**	**1-NN**
I_a	100	100	100
I_b	100	100	100
I_c	100	100	100

This chapter has presented the implementation of energy disaggregation in the fault location application in power system, which proposes the use of the statistical feature extraction. The statistical variables of the fault current data have been computed during one cycle since the SLGF occurrence is detected by DWT. In comparison with the previous work on NILM, one major advantage of this concept is that the extraction of transient feature is less computationally burdensome since the statistical variables only need one simple calculation within one cycle. This concept of feature extraction can effectively reduce the dimensions of data inputs for machine learning processing. To verify the validity of the proposed method, three popular AI algorithms used in NIFM are compared in this paper. If the proposed methods are followed, SVM is the best candidate for the disaggregation when SLGF occurs. The evolution of the outcomes in SVM from SMs to SK also clarifies that a proper feature extraction does not need a complicated model for disaggregation. In other words, the better the feature extraction is, the less the machine classifiers learn. Moreover, the effect of good feature extraction, i.e., SKs, for improving the disaggregation is also demonstrated.

Those statistical features characterize the distinction among the locations of the fault current signals as well as each phase. The experimental results also show that the monitored current of phase A is the best when SLGF occurs on the phase A by using the proposed methods.

The case studies include some of the most challenging scenarios for a NIFM system such as system voltage variations and SLGF on phase A. The proposed methods (SK and SVM) demonstrate the enhanced capability for fault location and faulty phase detection, compared with other existing statistical methods such as Z-scores and SMs. So far, the dataset just considers only the variations of transformer taps for the first attempt in the new approach of NIFM. Further work should be carried out to confirm the feasibility of the non-intrusive technique in fault study. For example, the other fault types, *e.g.*, DLGF, DLF, and balanced faults, should be included. The variations of fault resistances and inception angles should be also considered.

Chapter 7

Conclusions, Limitations and Future Work

This thesis has presented the application of non-intrusive technique in load monitoring for the household use and in fault study for the protection purpose. We have mentioned the NILM dataset, disaggregation methods, as well as the technical feasibility of the non-intrusive fault detection. This chapter summarizes the contribution of this thesis, reflects the limitations, and provides directions for future research.

7.1 Conclusions

Firstly, the general problems of NILM have been addressed in Chapter 1. In order to achieve a practical solution, two core requirements should be matched. The solution is able to perform the accurate disaggregation with the low-sampling frequency smart meter data. Also, the solution should require less off-line training stage from household data.

In Chapter 2, the state-of-the-art work in NILM research has been reviewed.

We classified the literature review into feature analysis techniques and disaggregation algorithms. Nevertheless, most research papers state the use of high-sampling frequency data or require the complex training process.

Next, Chapter 3 has described some common datasets released since 2011. The REDD dataset was selected as it both contains the sub-metered and main-circuit data from different houses. It has been also the most used dataset for NILM research at the time of writing. One challenge of REDD dataset is that the sampling frequency between the sub-meters and main-circuit meters are not the same. As a result, the consumption of one appliance in the sub-meter plots is somehow unaligned with its consumption in the main-circuit plots. This can create the significant challenge to evaluate the ground-truth from the aggregated power. One possible solution is to sum all sub-metered data to approximately acquire the aggregated data which has the same sampling frequency with the appliance meters. The REDD dataset is then visualized in the web interface, which is the first step to initiate an online NILM platform.

Chapter 4 and 5 have proposed two disaggregation methods and their result in this thesis. One is eventless-based combinatorial optimization (CO) and the other is event-based recurrent neural network (RNN). The outcomes conclude that CO mostly falls behind RNN in those case studies, and RNN could work well for the unseen case studies. However, some improvement is considerably needed to enhance the RNN performance for the multi-state appliances.

Chapter 6 has presented the application of energy disaggregation technique to fault location. The chapter concludes the possible scenarios for that topic.

In summary, the contribution in this thesis can be outlined below:

- The application of deep recurrent neural network to NILM is demonstrated. This generally shows the better outcomes than the benchmark algorithms.
- A simple web interface which can receive the aggregated consumption and ex-

tract the disaggregation has been initially built.

- The energy disaggregation technique applied to fault study has been initially examined.

7.2 Limitations

1. **Ground-truth data collection for supervised learning**

   Usually, the end-users are asked to turn on only one appliance during a specific duration in order to obtain the ground-truth data of this appliance, for the training purpose. This, however, seems inconvenient in the users' point of view as their energy usage habit mostly depend on multiple appliances at the same time. Another way is to install each sub-meter at each appliance in house, but the cost of those sub-meters has to be covered.

2. **Failure to disaggregate multi-state appliances in CO**

   The CO model has been unable to disaggregate the dishwasher. This is the fundamental issue of non-unique solutions in optimization frame as one power state of an appliance can be equal to the sum of power states from other appliances. Additionally, as CO is eventless-based method, it could not describe the physical characteristics of multi-state appliances.

3. **Difficulty in identifying noisy outlets**

   On average, only 40% of the consumption from kitchen outlets in both CO and RNN are correctly assigned, which cannot be considered a success. One reason for this could be the noisy data from the kitchen outlets. It is not an easy task to exactly determine the power states in kitchen as kitchen is the place where a various number of appliances are in operation.

4. **Local minima from *k*-means**

   The local minima from *k*-means could provide the wrong power states of multi-state appliances, which can lead to the inaccurate disaggregation in the next phases. One simple solution is to manually re-run the k-means for multiple times.

5. **Non-alignment between the main-circuit meters and sub-meters data**

   Apparently, summing all ground-truth sub-meters to obtain the aggregated values is just the approximation. The results could be largely different if the main-circuit meters are taken into account. This, nonetheless, requires the same sampling frequency between household-level meters (aggregated power) and appliance-level meters (ground-truth) in the dataset.

6. **Long training of ANN**

   In ANN, training process is computationally burdensome, which requires several days up to months to achieve good performance. There is the possibility that training has to be re-done if the end-users buy or replace new appliances. This could frequently occur depending on their usage habit and how well the NILM models can generalize the unseen houses.

   Due to the limitation of our computer systems, only a small days per house can be used for training. The more the training data are provided, the better the network outputs are produced.

7. **Necessity to re-train and improve RNN for the unseen dishwasher**

   Clearly, the unseen dishwasher is unsuccessful to be disaggregated when all metrics are too low-grade. However, to improve those outcomes requires more training set, which leads to the investment in computer hardware.

8. **Empirical hyper-parameter optimization**

   Hyper-parameters, for example, indicate the number of population and generation in CO as well as the layer number, the neuron number and the delay number in RNN. The tuning of those hyper-parameters is mostly based on experience and the characteristic of dataset. Also, optimizing and adjusting those hyper-parameters in a complex model are time-consuming.

9. **Unready real-time feedback**

   Overall, RNN requires the powerful computer hardware. Generally, the current technology of embedded systems could not possibly put the complex RNNs into real-world applications. At this moment, one possible solution is to download the aggregated power data that the customers provided. RNN model, which has been trained in advance, could off-line process such data in MATLAB. Then, the results can be uploaded to the website and sent to the customers for feedbacks. It probably takes one or two hours for this process.

10. **Undefinable number of appliances in a house and power states**

    The purpose of optimization is to search for possible power combinations of individual appliances to match the aggregated power. This requires the exact number of appliances and their corresponding power states. Hence, this method may not be applied to estimate the unseen houses as their information is not available. Supervised RNN, however, can estimate power from the unseen houses if its model is well generalized.

11. **Difficulty in comparing different benchmark NILM algorithms**

    The NILM algorithms depend on the characteristics of dataset or metrics. As such, it is generally unable to give the comparison among algorithms.

12. **Small-scale dataset of NIFM**

    Chapter 6 mentioned the NIFM with the simulation dataset. Nevertheless, such dataset is still a tiny portion, which only considers the voltage variations. Also, harmonics and noise have not yet been investigated.

13. **Unready NIFM**

    Even though the results show the good accuracy in fault location, there are still a major challenge of this topic. The non-intrusive fault detection should be in real-time; whilst the fault system monitoring should be implemented by a high-frequency sampling DAQ. At this moment, the real-time computing is even an enormous difficulty for the load monitoring purpose. NIFM, at this stage, could be only suitable for the academic research. Further work should be carried out to assess the realistic prospect of non-intrusive fault study.

7.3 Future Work

Primarily, the future work will concentrate on developing the cloud computing system to improve energy efficiency in buildings. Energy feedback via NILM services is one of the key topics in that development. The NILM web should be able to monitor and classify the end-users' behavior as well as recommend the customers to adapt their usage habit or tariffs for better savings. The users can access the NILM application via their mobiles. An initial web interface has been built through this thesis work. In the near future, the web interface for different disaggregation models, *e.g.* ANN, CO, should be implemented to achieve the high performance as we believe that one disaggregation algorithm is insufficient.

In such an implementation, it might be useful to construct the general models for one appliance. For example, the dishwashers in house 1 and house 2 have different power consumptions. Therefore, it is not effective to build every dishwasher model for every house. ANN has been the useful tool to deploy this scheme. Also, computer hardware should be investigated for better training ANN to improve the recognition accuracy.

In addition, extensive manual training phase is one of the major challenge of supervised learning as the information of household appliances is not always available. Also, there are two common ways to obtain the ground-truth data. The first is to install all sub-meters of all appliances in a house, which incurs significant costs. This, again, has been the intrusive training phase. The second way is to ask the consumers to turn on only one appliance for a specific duration. However, they might dislike this as their habit will be affected. The future work should be able to propose an unsupervised training NILM which does not require much information of the household and extensive off-line training. Even though k-means could be used as the simple unsupervised method for NILM, it requires to know the number of appliances in a house, *i.e* k should be defined.

Furthermore, the parallel disaggregation of gas, water, and smart meters should be considered. Some appliances, *e.g.* dishwasher or washing machine, require both water and electricity. With more information from another utility, the disaggregation could be facilitated, and the better achievement could be possible. It is interesting to research new techniques which accept the inputs from multiple utilities and return the disaggregation estimates.

Finally, it is recommended to have the benchmark dataset for Spanish households. The reason is the household characteristics vary enormously from country to country. For example, dishwashers are uncommonly used in Spain. Accordingly, the methods which succeeds in disaggregating REDD may not be applicable to other datasets in other countries.

Appendix A

A.1 Power States of Appliances in Remaining Houses

Table A.1: Non-off power states in W of house-2 appliances

Appliances	**State 1**	**State 2**	**State 3**
Dishwasher	249	1196	-
Refrigerator	162	-	-
Kitchen Outlets	775	1059	-
Lighting	106	156	289
Microwave	1859	-	-
Stove	407	-	-

Table A.2: Non-off power states in W of house-3 appliances

Appliances	**State 1**	**State 2**	**State 3**	**State 4**	**State 5**
Bathroom GFI	1282.5	1196	-	-	-
Dishwasher	209	735	-	-	-
Electronics	75	98	131	172	-
Refrigerator	117	-	-	-	-
Furnace	704	-	-	-	-
Kitchen Outlets	50	127.5	370	905	-
Lighting	30	130	180	242	355
Microwave	1725	-	-	-	-
Stove	4760	-	-	-	-

Table A.3: Non-off power states in W of house-4 appliances

Appliances	**State 1**	**State 2**	**State 3**	**State 4**	**State 5**
Dishwasher	130	1300	-	-	-
Furnace	104	501	572	645	-
Kitchen Outlets	30	145	166	-	-
Lighting	30	54	116	152	226
Stove	104	400	-	-	-
Washer Dryer	265	-	-	-	-

Table A.4: Non-off power states in W of house-5 appliances

Appliances	**State 1**	**State 2**	**State 3**
Bathroom GFI	1608	1196	-
Dishwasher	384	1265	-
Electronics	22.5	52.5	-
Refrigerator	157	-	-
Furnace	157	-	-
Subpanel	62.5	80	97.5

Table A.5: Non-off power states in W of house-6 appliances

Appliances	**State 1**	**State 2**	**State 3**
Air Conditioning	407	2363	-
Electric Heat	67	-	-
Electronics	159	538	712
Refrigerator	131	-	-
Kitchen Outlets	50	1295	-
Lighting	69	140	-
Stove	4760	-	-
Unknown Outlets (Unmetered)	47	112	197

Appendix B

B.1 Discrimination Threshold of Each Appliances in RNN Outputs for Seen Houses

Appliance ID	House ID	Optimal Discrimination Threshold (ODT)	Area Under ROC Curve at ODT
Bathroom	1	0.3117	0.6067
Dishwasher	1	0.4382	0.7579
Refrigerator	1	0.4528	0.9223
Kitchen outlets	1	0.4881	0.5349
Lighting	1	0.5408	0.9288
Microwave	1	0.5418	0.7703
Oven	1	0.5755	0.9498
Washer Dryer	1	0.6847	0.8568
Dishwasher	2	0.4924	0.7052
Refrigerator	2	0.6812	0.8896
Kitchen outlets	2	0.8385	0.8649
Lighting	2	0.5208	0.7994
Microwave	2	0.4283	0.945
Stove	2	0.2937	0.7715

Appliance ID	**House ID**	**Optimal Discrimination Threshold (ODT)**	**Area Under ROC Curve at ODT**
Dishwasher	4	0.5007	0.8796
Furnace	4	0.5516	0.8347
Kitchen outlets	4	0.517	0.8587
Lighting	4	0.4269	0.8117
Stove	4	0.5946	0.5351
Washer Dryer	4	0.5502	0.5368

Appendix C

C.1 Estimated Demand of Some Appliances Using Combinatorial Optimization

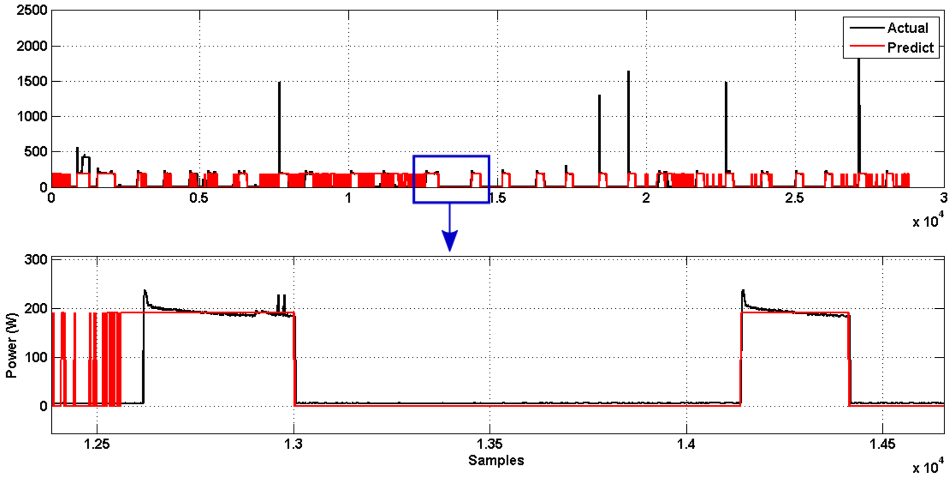


Figure C-1: Estimated demand of refrigerator in House 1 from 18/04/2011 13:00 to 19/04/2011 19:00.

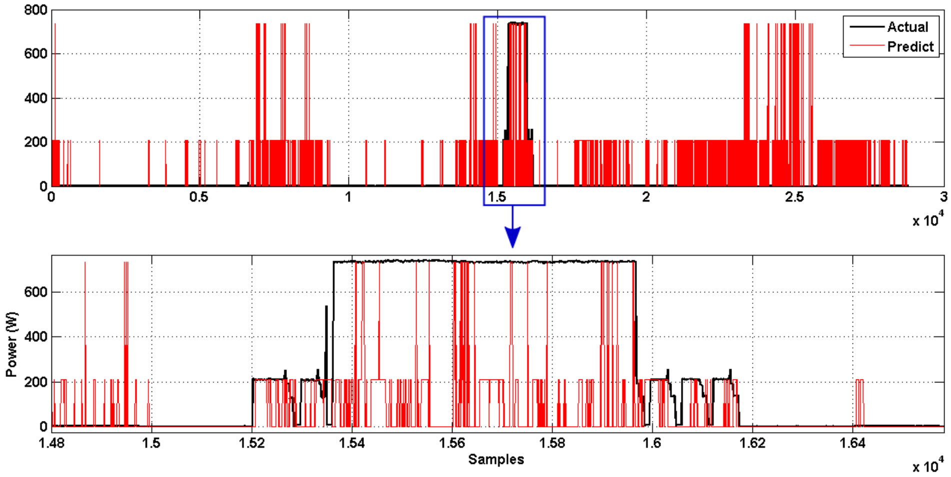


Figure C-2: Estimated demand of dishwasher in House 3 from 16/04/2011 5:11 to 18/04/2011 6:53.

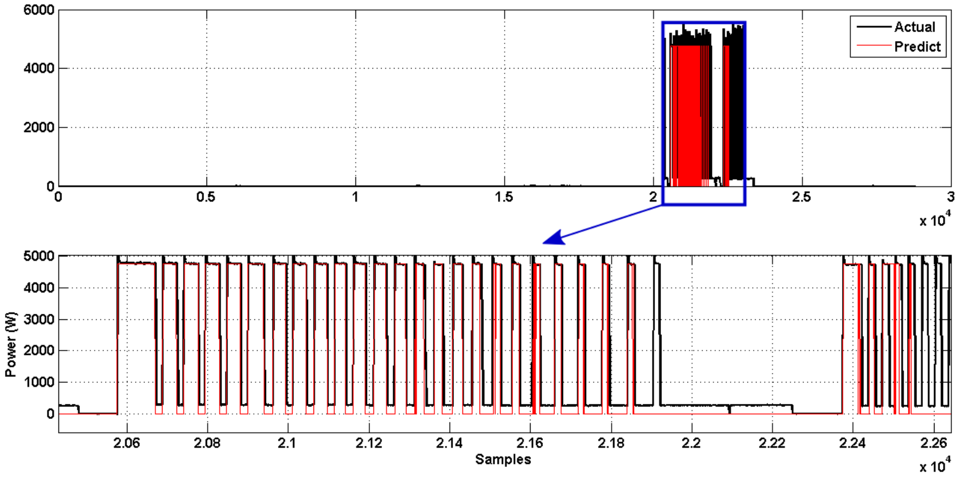


Figure C-3: Estimated demand of washer dryer in House 3 from 20/04/2011 4:26 to 20/04/2011 13:02.

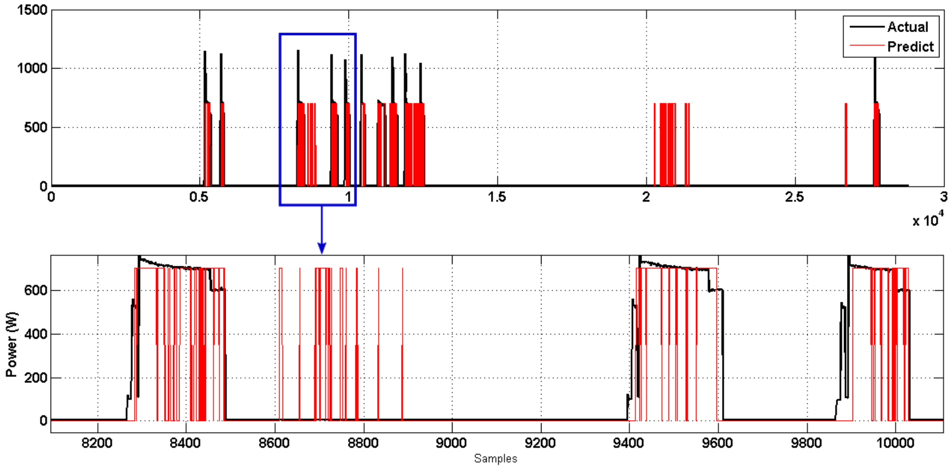


Figure C-4: Estimated demand of furnace in House 3 from 21/04/2011 14:06 to 21/04/2011 19:21.

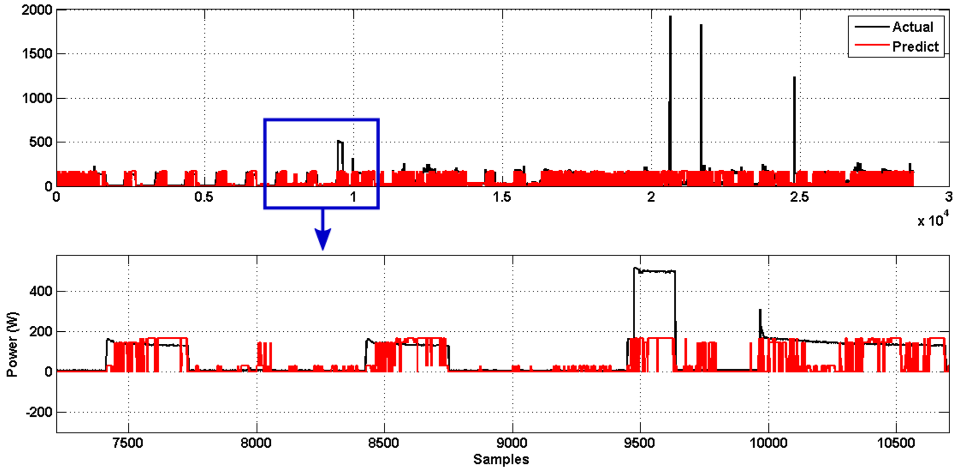


Figure C-5: Estimated demand of kitchen outlets in House 4 from 18/04/2011 17:35 to 18/04/2011 22:54.

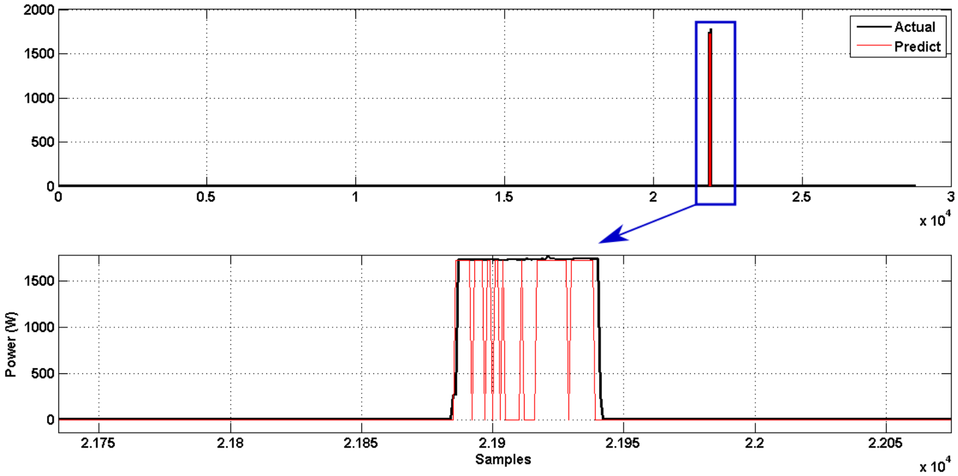


Figure C-6: Estimated demand of stove in House 6 from 24/05/2011 13:25 to 24/05/2011 18:14.

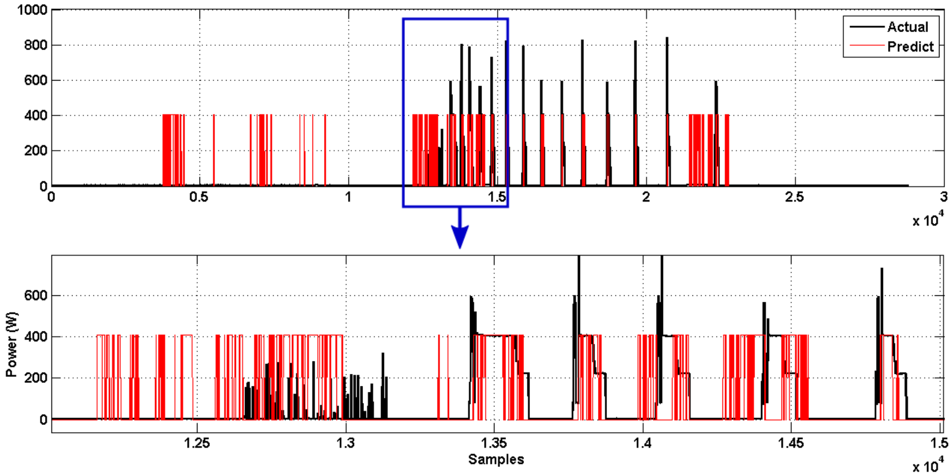


Figure C-7: Estimated demand of air conditioner in House 6 from 24/05/2011 13:25 to 24/05/2011 18:14.

C.2 Disaggregation Results of House 3, 5 and 6 Using Combinatorial Optimization

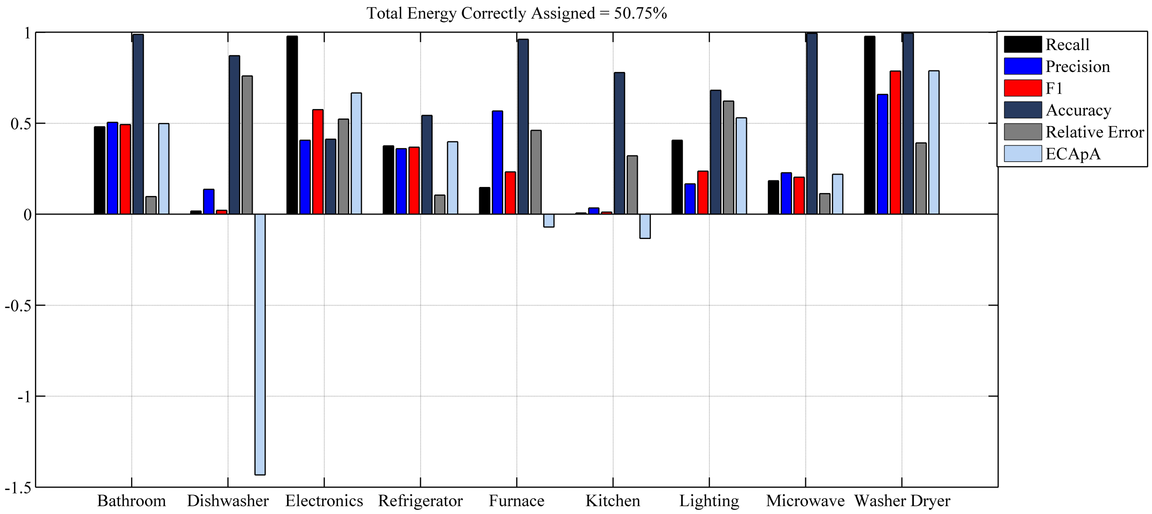


Figure C-8: House 3.

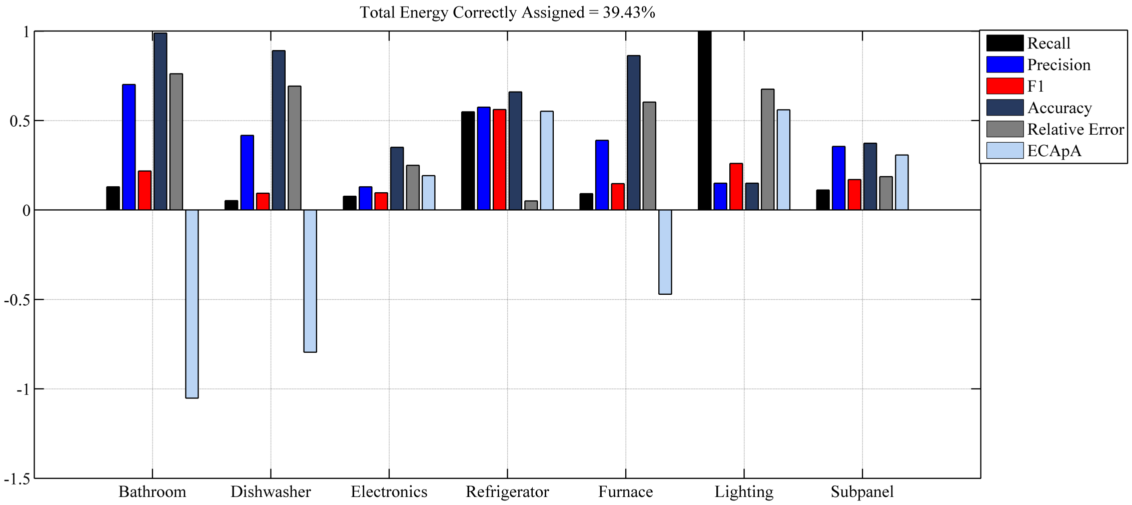


Figure C-9: House 5.

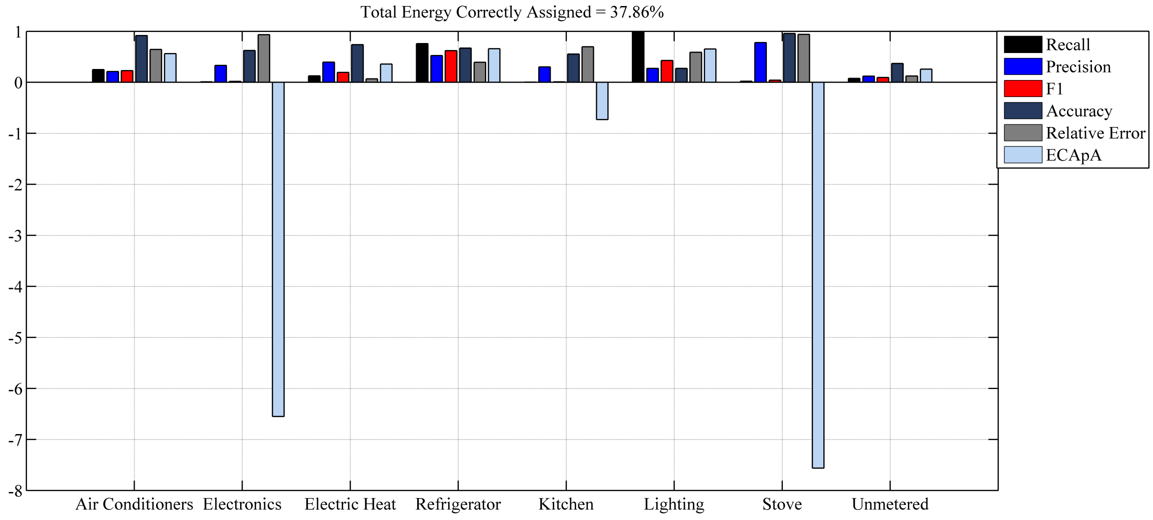


Figure C-10: House 6.

Bibliography

[1] G. W. Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891, Dec 1992. [Cited on pages 1 and 8.]

[2] Hsueh-Hsien Chang and Nguyen Viet Linh. Statistical feature extraction for fault locations in nonintrusive fault detection of low voltage distribution systems. *Energies*, 10(5):611, 2017. [Cited on pages 5, 50, 53, 55, 56, 58, 61, 62, 66, 68, 69, 70, 71, 72, and 73.]

[3] Parson. Overview of the nilm field. http://blog.oliverparson.co.uk/2015/03/overview-of-nilm-field.html, 2015. Accessed: 2017-07-13. [Cited on page 7.]

[4] C. Laughman, Kwangduk Lee, R. Cox, S. Shaw, S. Leeb, L. Norford, and P. Armstrong. Power signature analysis. *IEEE Power and Energy Magazine*, 1(2):56–63, Mar 2003. [Cited on page 7.]

[5] M. Zeifman and K. Roth. Nonintrusive appliance load monitoring: Review and outlook. *IEEE Transactions on Consumer Electronics*, 57(1):76–84, February 2011. [Cited on page 7.]

[6] A. I. Cole and A. Albicki. Data extraction for effective non-intrusive identification of residential power loads. In *IMTC/98 Conference Proceedings. IEEE Instrumentation and Measurement Technology Conference. Where Instrumentation is Going (Cat. No.98CH36222)*, volume 2, pages 812–815 vol.2, May 1998. [Cited on page 8.]

[7] M. Akbar and Z. A. Khan. Modified nonintrusive appliance load monitoring for nonlinear devices. In *2007 IEEE International Multitopic Conference*, pages 1–5, Dec 2007. [Cited on page 8.]

[8] T. Hassan, F. Javed, and N. Arshad. An empirical investigation of v-i trajectory based load signatures for non-intrusive load monitoring. *IEEE Transactions on Smart Grid*, 5(2):870–878, March 2014. [Cited on page 8.]

[9] H. H. Chang, K. L. Chen, Y. P. Tsai, and W. J. Lee. A new measurement method for power signatures of nonintrusive demand monitoring and load identification. *IEEE Transactions on Industry Applications*, 48(2):764–771, March 2012. [Cited on pages 8 and 9.]

[10] Men-Shen Tsai and Yu-Hsiu Lin. Modern development of an adaptive non-intrusive appliance load monitoring system in electricity energy conservation. *Applied Energy*, 96:55 – 73, 2012. Smart Grids. [Cited on pages 8 and 10.]

[11] Y. H. Lin and M. S. Tsai. Development of an improved time-frequency analysis-based nonintrusive load monitor for load demand identification. *IEEE Transactions on Instrumentation and Measurement*, 63(6):1470–1483, June 2014. [Cited on pages 8 and 9.]

[12] J. M. Gillis, S. M. Alshareef, and W. G. Morsi. Nonintrusive load monitoring using wavelet design and machine learning. *IEEE Transactions on Smart Grid*, 7(1):320–328, Jan 2016. [Cited on page 8.]

[13] Ahmed Zoha, Alexander Gluhak, Muhammad Ali Imran, and Sutharshan Rajasegarar. Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey. *Sensors*, 12(12):16838–16866, 2012. [Cited on page 9.]

[14] B. Zhao, L. Stankovic, and V. Stankovic. On a training-less solution for non-intrusive appliance load monitoring using graph signal processing. *IEEE Access*, 4:1784–1799, 2016. [Cited on page 9.]

[15] 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)*, pages 81–87, Dec 2014. [Cited on page 9.]

[16] Juan D.S. Guedes, Danton D. Ferreira, and Bruno H.G. Barbosa. A non-intrusive approach to classify electrical appliances based on higher-order statistics and genetic algorithm: a smart grid perspective. *Electric Power Systems Research*, 140:65 – 69, 2016. [Cited on pages 9 and 10.]

[17] Aggelos S. Bouhouras, Apostolos N. Milioudis, and Dimitris P. Labridis. Development of distinct load signatures for higher efficiency of nilm algorithms. *Electric Power Systems Research*, 117:163 – 171, 2014. [Cited on page 10.]

[18] J. G. Roos, I. E. 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)*, pages 1115–1118 vol.3, May 1994. [Cited on page 10.]

[19] D. Srinivasan, W. S. Ng, and A. C. Liew. Neural-network-based signature recognition for harmonic source identification. *IEEE Transactions on Power Delivery*, 21(1):398–405, Jan 2006. [Cited on page 10.]

[20] Takeshi SAITOH, Tomoyuki OSAKI, Ryosuke KONISHI, and Kazunori SUGAHARA. Current sensor based home appliance and state of appliance recognition. *SICE Journal of Control, Measurement, and System Integration*, 3(2):86–93, 2010. [Cited on page 10.]

[21] L. Du, J. A. Restrepo, Y. Yang, R. G. Harley, and T. G. Habetler. Nonintrusive, self-organizing, and probabilistic classification and identification of plugged-in electric loads. *IEEE Transactions on Smart Grid*, 4(3):1371–1380, Sept 2013. [Cited on page 10.]

[22] Leonard E. Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *Ann. Math. Statist.*, 37(6):1554–1563, 12 1966. [Cited on page 11.]

[23] Zoubin Ghahramani and Michael I. Jordan. Factorial hidden markov models. *Machine Learning*, 29(2):245–273, Nov 1997. [Cited on page 11.]

[24] SIAM, editor. *Unsupervised Disaggregation of Low Frequency Power Measurements*, 2011. [Cited on page 11.]

[25] Oliver Parson, Siddhartha Ghosh, Mark Weal, and Alex Rogers. Non-intrusive load monitoring using prior models of general appliance types. In *Proceedings of theTwenty-Sixth Conference on Artificial Intelligence (AAAI-12)*, pages 356–362, July 2012. [Cited on page 11.]

[26] Kofi Afrifa Agyeman, Sekyung Han, and Soohee Han. Real-time recognition non-intrusive electrical appliance monitoring algorithm for a residential building energy management system. *Energies*, 8(9):9029–9048, 2015. [Cited on page 11.]

[27] Kosuke Suzuki, Shinkichi Inagaki, Tatsuya Suzuki, Hisahide Nakamura, and Koichi Ito. Nonintrusive appliance load monitoring based on integer programming. In *2008 SICE Annual Conference*, pages 2742–2747, Aug 2008. [Cited on page 11.]

[28] M. Z. A. Bhotto, S. Makonin, and I. V. Baji. Load disaggregation based on aided linear integer programming. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 64(7):792–796, July 2017. [Cited on pages 11 and 32.]

[29] Jack Kelly and William J. Knottenbelt. Neural NILM: deep neural networks applied to energy disaggregation. *CoRR*, abs/1507.06594, 2015. [Cited on pages 12 and 38.]

[30] Reference Energy Disaggregation Dataset. Dataset available in. `http://redd.csail.mit.edu/`, 2011. Accessed: 2017-07-13. [Cited on page 13.]

[31] J. Zico Kolter and Matthew J. Johnson. REDD: A Public Data Set for Energy Disaggregation Research. In *SustKDD Workshop on Data Mining Applications in Sustainability*, 2011. [Cited on pages 13 and 38.]

[32] Christian Beckel, Wilhelm Kleiminger, Romano Cicchetti, Thorsten Staake, and Silvia Santini. The eco data set and the performance of non-intrusive load monitoring algorithms. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, BuildSys '14, pages 80–89, New York, NY, USA, 2014. ACM. [Cited on page 14.]

[33] Electricity Consumption and Occupancy Dataset. Dataset available in. http://data-archive.ethz.ch/delivery/DeliveryManagerServlet?dps_pid=IE594964, 2014. Accessed: 2017-07-15. [Cited on page 14.]

[34] Jack Kelly and William Knottenbelt. The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. *Scientific Data*, 2(150007), 2015. [Cited on page 15.]

[35] S. Lloyd. Least squares quantization in pcm. *IEEE Trans. Inf. Theor.*, 28(2):129–137, September 2006. [Cited on page 26.]

[36] Melanie Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, USA, 1998. [Cited on page 30.]

[37] Fan Chunju, K.K. Li, W.L. Chan, Yu Weiyong, and Zhang Zhaoning. Application of wavelet fuzzy neural network in locating single line to ground fault (slg) in distribution lines. *International Journal of Electrical Power & Energy Systems*, 29(6):497 – 503, 2007. [Cited on pages 51 and 56.]

[38] Sami Ekici, Selcuk Yildirim, and Mustafa Poyraz. Energy and entropy-based feature extraction for locating fault on transmission lines by using neural network and wavelet packet decomposition. *Expert Systems with Applications*, 34(4):2937 – 2944, 2008. [Cited on pages 51 and 56.]

[39] M. J. B. Reddy, D. V. Rajesh, P. Gopakumar, and D. K. Mohanta. Smart fault location for smart grid operation using rtus and computational intelligence techniques. *IEEE Systems Journal*, 8(4):1260–1271, Dec 2014. [Cited on pages 51 and 56.]

[40] A. Borghetti, M. Bosetti, M. Di Silvestro, C. A. Nucci, and M. Paolone. Continuous-wavelet transform for fault location in distribution power networks: Definition of mother wavelets inferred from fault originated transients. *IEEE Transactions on Power Systems*, 23(2):380–388, May 2008. [Cited on page 51.]

[41] F. B. Costa. Fault-induced transient detection based on real-time analysis of the wavelet coefficient energy. *IEEE Transactions on Power Delivery*, 29(1):140–153, Feb 2014. [Cited on pages 51 and 56.]

[42] Sylvain Ehrenfeld and Sebastian B Littauer. *Introduction to statistical methods*. McGraw-Hill, 1964. [Cited on page 52.]

[43] Cornelius Mack. *Essentials of statistics for scientists and technologists*. Springer Science & Business Media, 2012. [Cited on page 52.]

[44] J.A. Cadzow and X.K. Li. Blind deconvolution. *Digital Signal Processing*, 5(1):3 – 20, 1995. [Cited on page 52.]

[45] R. Dwyer. Use of the kurtosis statistic in the frequency domain as an aid in detecting random signals. *IEEE Journal of Oceanic Engineering*, 9(2):85–92, Apr 1984. [Cited on page 52.]

[46] A Tesei and CS Regazzoni. The asymmetric generalized gaussian function: A new hos-based model for generic noise pdfs. In *Statistical Signal and Array Processing, 1996. Proceedings., 8th IEEE Signal Processing Workshop on (Cat. No. 96TB10004*, pages 210–213. IEEE, 1996. [Cited on page 52.]

[47] OptiFuse. Rev a 01/2010. available online:. Accessed on: 26 July 2017. [Cited on page 54.]

[48] ABB Inc. Miniature circuit breaker-application guide. techreport, ABB Inc., Wichita Falls, TX, USA,, 2009. [Cited on page 54.]

[49] Yves Meyer. Wavelets: Algorithms and applications. *SIAM, Philadelphia*, 1993. [Cited on page 54.]

[50] D. C. Robertson, O. I. Camps, J. S. Mayer, and W. B. Gish. Wavelets and electromagnetic power system transients. *IEEE Transactions on Power Delivery*, 11(2):1050–1058, Apr 1996. [Cited on page 54.]

[51] M Da Silva, M Oleskovicz, and DV Coury. A hybrid fault locator for three-terminal lines based on wavelet transforms. *Electric Power Systems Research*, 78(11):1980–1988, 2008. [Cited on page 56.]

[52] C. H. Kim and R. K. Aggarwal. Closure on "a novel fault detection technique of high impedance arcing faults in transmission lines using the wavelet transform". *IEEE Transactions on Power Delivery*, 18(4):1596–1597, Oct 2003. [Cited on pages 56 and 57.]

[53] KM Silva, Benemar A Souza, and Nubia SD Brito. Fault detection and classification in transmission lines based on wavelet transform and ann. *IEEE Transactions on Power Delivery*, 21(4):2058–2063, 2006. [Cited on page 56.]

[54] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995. [Cited on page 59.]

[55] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013. [Cited on page 63.]

[56] Stephen Wagner. *Heuristic Optimization Software Systems - Modeling of Heuristic Optimization Algorithms in the HeuristicLab Software Environment*. Phd thesis, Johannes Kepler University, Linz, Austria, 2009. [Cited on page 66.]