

**TIME-FREQUENCY ANALYSIS TECHNIQUES FOR NON-INTRUSIVE LOAD
MONITORING**

by

Jessie Michael Gillis

B. Eng., University of Ontario Institute of Technology, Canada, 2015

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of

Master of Applied Science

in

The Faculty of Engineering and Applied Science

Electrical and Computer Engineering

University of Ontario Institute of Technology

October, 2016

© Jessie Gillis, 2016

Abstract

The role of existing revenue meters is to measure and record the energy consumed by the customer in order to report usage to the electric utilities for billing purposes. Existing smart meters use the automated meter reading (AMR) technology to transmit the customers' load profiles at hourly or sub-hourly intervals, while providing time-of-use pricing to customers to motivate them to better utilize their appliances by avoiding the hours of peak demand. However, smart meters have proven to be not intelligent enough to provide useful information to the customers with respect to the devices/appliances that they use on a daily basis. The concept of non-intrusive load monitoring, in which the homes electrical signals are measured at a single point (i.e., one node), has been proposed to enhance the functionality of existing smart meters. In non-intrusive load monitoring, the collected data at a single point, represented by the smart meter, is used to monitor the energy consumption of individual appliances/devices instead of installing dedicated sensors on each individual component, which has proven to be cost-ineffective and time consuming to install and maintain. The work in this thesis examines time-frequency analysis techniques and in particular the wavelet transform to extract the features contained within the electrical load signals. A novel approach that is based on wavelet design was utilized to generate a wavelet library which was used to match each load signal to a specific wavelet using Procrustes and covariance analysis. In order to automate the load identification process, two machine learning classifiers representing an eager learner and a lazy learner were used in this work. The proposed wavelet design concept has been verified experimentally, and the results of implementing the proposed load detection and classification approach shows significant improvement in the classification accuracy compared to other existing detection approaches reaching an overall accuracy of 98%.

Acknowledgements

I would like to express my sincere gratitude to my thesis supervisor Dr. Walid Morsi Ibrahim for the continued support and guidance throughout my time as an undergraduate and graduate student. In my time under his supervision, I have grown as an individual and an engineer and am confident that without his support I would not be the person I am today. I would also like to thank him, the University of Ontario Institute of Technology, and the Government of Ontario for their financial support in allowing me to be free to conduct research to better the lives of homeowners in helping them understand and reduce their energy consumption. Lastly, I would like to thank my parents for all the love and guidance they have given me.

“The people who are crazy enough to think they can change the world, are the ones who do.”

Steve Jobs

"If something is important enough, even if the odds are against you, you should still do it."

Elon Musk

Table of Contents

Abstract.....	ii
Acknowledgements.....	iii
Table of Contents.....	iv
List of Tables	vii
List of Figures.....	viii
Nomenclature.....	x
1. Introduction	1
1.1 Background	1
1.2 Problem Statement and Motivation.....	2
1.3 Contribution	5
1.4 Thesis Organization.....	6
2. Literature Review	8
2.1 Introduction	8
2.2 Previous Work on Non-Intrusive Load Monitoring.....	8
2.2.1 Time Domain Analysis	8
2.2.2 Frequency Domain Analysis.....	14
2.2.3 Time-Frequency Domain Analysis.....	15
2.3 Performance Metrics	18
2.3.1 Calculating the Performance of Each Method.....	22
2.4 Summary	24
3. Orthogonal Wavelet Design	25
3.1 Introduction	25
3.2 Discrete Wavelet Transform	25
3.3 Feature Extraction and Identification.....	29
3.4 Wavelet Design Procedures	29
3.4.1 Introduction.....	29
3.4.2 Wavelet Filter Properties	30
3.4.3 Design Fundamentals.....	31

3.4.4	Wavelet Clustering.....	36
3.4.5	Wavelet Naming	43
3.5	Procrustes Analysis	44
3.6	Covariance Analysis.....	45
3.7	Summary	46
4.	Machine Learning Classification for Non-Intrusive Load Monitoring	48
4.1	Introduction	48
4.2	Signal Acquisition.....	48
4.3	Cycle Length	50
4.4	Smart Matching Module.....	53
4.5	Machine Learning Classifiers.....	54
4.5.1	k-Nearest Neighbor.....	55
4.5.2	Decision Tree.....	56
4.6	k-fold Cross Validation	56
4.7	Monte Carlo.....	57
4.8	One Against the Rest.....	58
4.9	The Complete NILM Algorithm	58
4.10	Summary	60
5.	Results and Evaluation	62
5.1	Introduction	62
5.2	Test System Description.....	63
5.3	Dataset Generation	65
5.4	Daubechies Wavelets	68
5.5	MATLAB Simulation	70
5.6	Simulation Results	73
5.6.1	Length-6.....	73
5.6.2	Length-8.....	74
5.7	Discussion	74
5.7.1	Length-6.....	74
5.7.2	Length-8.....	75
5.8	Simulation Discussion.....	76
5.9	Experimental Setup	76

5.9.1	Hardware Specifications	77
5.9.2	Test Environment.....	87
5.10	Experimental Dataset Generation.....	88
5.10.1	The Experimental Dataset.....	91
5.11	Computation and Processing.....	93
5.11.1	Cycle Lengths	93
5.11.2	Monte Carlo method	94
5.12	Experimental Results.....	96
5.12.1	Length 6	96
5.12.2	Length 8	97
5.13	Discussion and Analysis.....	97
5.13.1	Length-6.....	98
5.13.2	Length-8.....	99
5.14	Summary	99
6.	Conclusions and Recommendations.....	101
6.1	Conclusion.....	101
6.2	Recommendations	104
6.3	Future Work	105
	References.....	106
	Appendix.....	110
	Hardware Component Specifications.....	110
	Phillips Incandescent Light Bulb.....	110
	Dell Studio 1535 Personal Computer	110
	Omega 1608FS-PLUS Data Acquisition Module	111
	Chroma Programmable AC Source Model# 61604.....	111
	Dranetz PX5 Power Quality Analyzer.....	112

List of Tables

Chapter 2

Table 2.1: NILM Methods based on Time-domain Analysis Techniques.....	13
Table 2.2: NILM Methods based on Frequency-domain Analysis Techniques	15
Table 2.3: NILM Methods based on Time-Frequency Domain Analysis Techniques	18
Table 2.4: Pairwise Comparison of Performance Metrics	20
Table 2.5: Measurement Type Scores.....	20
Table 2.6: Performance of Each Method	23

Chapter 3

Table 3.1: Number of Wavelets before and after clustering.....	43
--	----

Chapter 4

Table 4.1: Decomposition Levels for different Sampling Frequency.....	49
---	----

Chapter 5

Table 5.1: Switching Strategy to Generate Dataset	68
Table 5.2: Daubechies Wavelets used in the Results.....	69
Table 5.3: Simulation- K-NN for 1 Cycle of Length-6	73
Table 5.4: Simulation- DT for 1 Cycle of Length-6	74
Table 5.5: Simulation- K-NN for 1 Cycle of Length-8	74
Table 5.6: Simulation- DT for 1 Cycle of Length-8	74
Table 5.7: Programmable AC Source Power Quality Measurements.....	88
Table 5.8: Load Configuration Chart.....	90
Table 5.9: Length 6 – Classification Accuracy using K-NN	96
Table 5.10: Length 6 – Classification Accuracy for Supervised DT	97
Table 5.11: Length 8 – Classification Accuracy for Supervised K-NN	97
Table 5.12: Length 8 – Classification Accuracy for Supervised DT	97

List of Figures

Chapter 1

Fig. 1. 1: NILM Methods based on Time-domain Analysis Techniques.....	3
---	---

Chapter 2

Fig. 2.1: Example Electric Power Signal Breakdown [3]	9
Fig. 2.2: Example of V-I Trajectory [19].....	11
Fig. 2.3: Example of P-Q Mapping [3]	11
Fig. 2.4 Logarithmic Regression Line for a Performance Metric of Sampling Frequency ...	22

Chapter 3

Fig. 3.1: Multi-resolution Analysis Example.....	28
Fig. 3.2: Length-6 Golden Section Example for 3 Iterations.....	40
Fig. 3.3: Golden Section Converging onto Optimal k value.....	40
Fig. 3.4: Sum of Squares Error vs Number of Representative Wavelets.....	41
Fig. 3.5: Silhouette Coefficient vs Number of Representative Wavelets	41
Fig. 3.6: Sum of Squares Error vs Number of Representative Wavelets.....	42
Fig. 3.7: Silhouette Coefficient vs Number of Representative Wavelets	42
Fig. 3.8: Wavelet Naming Example [15]	44

Chapter 4

Fig. 4.1: Sequence Difference between Samples.....	50
Fig. 4.2: Experimental Set-up for Analysis of Current Waveform.....	51
Fig. 4.3: Switching Transient in the case of ILB	52
Fig. 4.4: Switching Transient of ILB after Sequence Subtraction.....	52
Fig. 4.5: Switching Transient of ILB after Sequence Subtraction for One Cycle	53
Fig. 4. 6: Depiction of wavelet smart matching module [15]	54
Fig. 4.7: The Complete NILM Algorithm	60

Chapter 5

Fig. 5.1: NILM Test System	65
Fig. 5.2: PSCAD models of the loads used in the test system.	67
Fig. 5.3: db3 Against Wavelet Design Set.....	70
Fig. 5.4: db4 Against Wavelet Design Set.....	70
Fig. 5.5: MATLAB Simulation Algorithm.....	72
Fig. 5.6: Trends of 1,000 Monte Carlo Trials	72
Fig. 5.7: Actual Experimental Setup.....	77
Fig. 5.8: Transient Voltage Signal in case of ILB	78

Fig. 5.9: Transient Current Signal in case of ILB.....	79
Fig. 5.10: Steady State Voltage Signal in case of ILB.....	79
Fig. 5.11: Steady State Current Signal in case of ILB.....	80
Fig. 5.12: Transient Voltage Signal in case of CFL	81
Fig. 5.13: Transient Current Signal in case of CFL.....	81
Fig. 5.14: Steady State Voltage Signal in case of CFL.....	82
Fig. 5.15: Steady State Current Signal in case of CFL	82
Fig. 5.16: Transient Voltage Signal in case of PC.....	83
Fig. 5.17: Transient Current Signal in case of PC	83
Fig. 5.18: Steady State Voltage Signal in case of PC	84
Fig. 5.19: Steady State Current Signal in case of PC.....	84
Fig. 5.20: Transient Voltage Signal in case of Battery Charger	85
Fig. 5.21: Transient Current Signal in case of Battery Charger.....	85
Fig. 5.22: Steady State Voltage Signal in case of Battery Charger	86
Fig. 5.23: Steady State Current Signal in case of Battery Charger.....	86
Fig. 5.24: Programmable AC Source Voltage Waveform	88
Fig. 5.25: 5 ILB Transients for Case 1 at 114V.....	92
Fig. 5.26: CFL Normal Transients.....	92
Fig. 5.27: CFL Abnormal Transient	93
Fig. 5.28: ILB 1 Cycle Transient Example.....	94
Fig. 5.29: Monte Carlo Accuracy Convergence over Iterations	95

Nomenclature

v	Wavelet Transform Scale Parameter
φ	Wavelet Transform Shift Parameter
db	Daubechies Wavelet Family
h_0	Low-pass Filter Coefficients
h_1	High-pass Filter Coefficients
ε_k	k-fold run error
ε_{cv}	k-fold cross validation error
Δ	Difference or Change in a quantity
δ	Euclidean distance proximity measure
θ	Phase Angle
λ	Dissimilarity Measure
τ	Window Size
$g[t]$	Window Function
$\hat{g}(t)$	Hilbert Transform of $g(t)$
$\text{sgn}(f)$	Signum Function
$V(t)$	Scaling Function
w	Performance Metric Weight
\bar{w}	k-fold accuracy measure
$W(t)$	Wavelet Function
$x[t]$	Discrete time representation of a signal x
x	Switching Transient
x_d	Switching Transient after Sequence Subtraction
$X[f]$	Frequency domain representation of $x[t]$
A2D	Analog to Digital
AC	Alternating Current
AMR	Automated Meter Reading
ANSI	American National Standard Institute
CA	Covariance Analysis
CFL	Compact Florescent Lightbulb
cov	Covariance
CSV	Comma Separated Values
CT	Current Transformer
DAQ	Data Acquisition Module
db	Daubechies
DFT	Discrete Fourier Transform
DT	Decision Tree
E	Energy
ED	Euclidean Distance
EMI	Electro-Magnetic Interference
EV	Electric Vehicle
f	Frequency
Fib_q	Fibonacci number of integer q

FT	Fourier Transform
GM	Gillis-Morsi
HMM	Hidden Markov Model
HT	Hilbert Transform
I	Current
ILB	Incandescent Light Bulb
K-NN	k- Nearest Neighbor
LED	Light Emitting Diode
MRA	Multi Resolution Analysis
N	Number of loads in the system
NN	Nearest Neighbor
NILM	Non-Intrusive Load Monitoring
OAR	One Against the Rest
P	Active Power
P_{tot}	Total number of positive cases
PA	Procrustes Analysis
PC	Personal Computer
PCA	Principal Component Analysis
PSCAD	Power System Computer Aided Design
PSO	Particle Swarm Optimization
Q	Reactive Power
REDD	Reference Energy Disaggregation Dataset
RMS	Root mean Square
Si	Silhouette Coefficient
s	Seconds
SMM	Smart Matching Module
SSE	Sum of Squares Error
STFT	Short Time Fourier Transform
SVM	Support Vector Machine
THD	Total Harmonic Distortion
TOU	Time of Use
TP	True Positive
USB	Universal Serial Bus
V	Voltage
VA	Volt Ampere
W	Watt
WT	Wavelet Transform

1. Introduction

1.1 Background

Over the past 10 years, the average price of electricity has steadily increased while the overall demand has decreased [1]. Smart meters have been installed in every home in Ontario, Canada to allow electric utilities to access energy data in 15 minute intervals, thus billing the customers based on the time at which they use energy. Time-of-Use (TOU) pricing is the method used in Ontario, Canada through which different prices for energy were identified based on the time of day [2]. The main limitation with existing smart meter technology is that no energy breakdown is provided to either the customer or the utility, only the aggregated energy usage per customer is measured. Therefore, residential customers have limited knowledge about the usage of their appliances and their efficiency. Furthermore, with this limited information on the energy usage of each customer, electric utilities are unable to fully take advantage of demand response programs since they have no information on the response of each appliance within each home to their signal requests. Due to these reasons, there is a dire need to identify individual load operation to ensure effective demand response and at the same time allow more customers to participate in utility programs. This can be accomplished by providing the breakdown of energy usage between the customers and utilities. Therefore, a homeowner's total energy usage needs to be disaggregated into each individual load operation so that one sensor can be used to gather data at one location. This approach does not intrude on the homeowner, but does provide the necessary information about load usage. This process is referred to as load disaggregation or Non-Intrusive Load Monitoring (NILM) which is formally defined as

methods of monitoring an electrical signal from a single point and using information contained within the signal to determine individual load operation [3].

1.2 Problem Statement and Motivation

Existing approaches for NILM in the literature are grouped into three main categories: time domain analysis, frequency domain analysis and time-frequency domain analysis, as depicted in Fig. 1.1. Time domain analysis-based approaches focus on using active power, reactive power, voltage, and current in the time domain at sampling rates greater than or equal to one sample per second. Frequency domain-based analysis approaches focus on the use of information contained in the frequency domain, such as harmonics which require sampling rates typically greater than or equal to one sample per second. Time-frequency domain approaches focus on capturing the switching transients of loads using either the voltage or current waveform at higher sampling rates (kHz to MHz range). These approaches use advanced signal processing techniques such as the windowed Fourier transform to perform a time and frequency analysis of a signal. Both time domain or frequency domain approaches are usually found to be inaccurate due to their low sampling rate and the limitations associated with the analysis techniques, while time-frequency approaches have been more effective due to their higher sampling rates and use of a wider range of signal processing techniques.

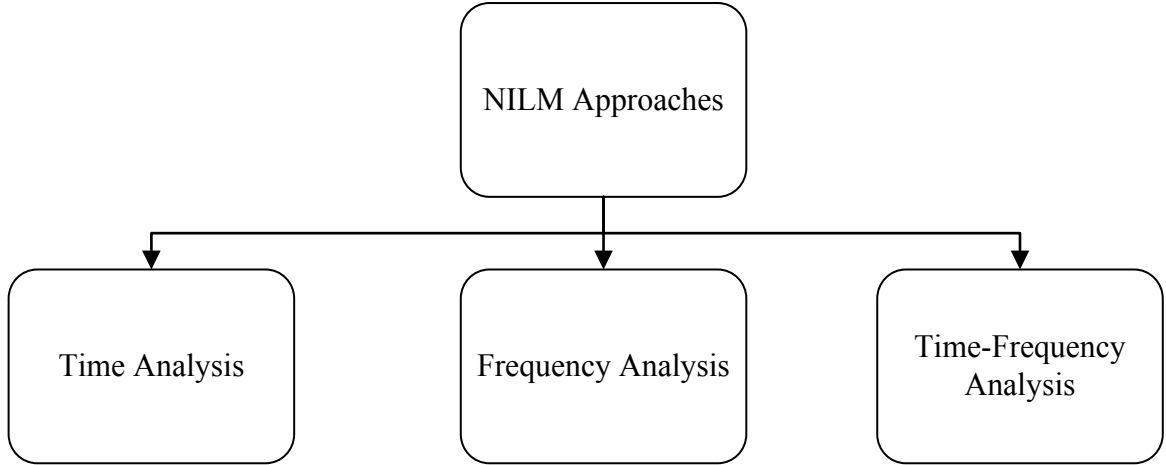


Fig. 1. 1: NILM Methods based on Time-domain Analysis Techniques

NILM approaches are commonly evaluated by comparing the classification accuracy of one method against the other. This seems elementary but becomes difficult in reality as the majority of approaches use various types of data with different sampling rates. Therefore, classification accuracy is one of the main factors to compare, but additional hardware requirements and the algorithms complexity are crucial to an approaches overall performance.

Time domain analysis techniques rely on using information contained within power quantities such as active and reactive power [4]. Their strength is in their simplicity and use of lower sampling rates [5]. However, these methods are easily susceptible to noise and power quality disturbances which usually make their overall classification accuracies dissatisfactory [6]. Furthermore, due to the use of lower sampling rates, load detection time can be minutes or hours making these approaches unfavorable in close to real-time applications [7].

Frequency domain analysis techniques rely on using only frequency information such as harmonics and changes in frequency [8]. These techniques are acceptable at

distinguishing loads that draw the same active power and perform well for large loads that have unique harmonic signatures [9]. These methods are not as susceptible to power quality disturbances, giving them advantages in some cases. However, frequency domain analysis techniques struggle for purely resistive loads with no harmonic signature and for small loads in which the harmonic profile is next to undetectable [9]. Frequency domain analysis techniques use only the information contained within the frequency, the time information is lost and cannot be used in the detection process.

Time-frequency domain analysis techniques rely on using the raw information contained within the voltage or current waveforms themselves [10]. By analyzing the switching transients of loads in both the time and frequency domain, features from these unique transients are processed to classify load operation. These methods strengths are in obtaining high classification accuracies although come at the cost of higher sampling rates and higher computational complexities [11]. Time-frequency transforms are considered effective signal processing techniques to provide both time and frequency information of the analysis signals. Most approaches rely on the Short-Time Fourier Transform (STFT) or the Wavelet Transform (WT) to extract features from the transient signal [12]. The STFT uses the sine and cosine functions as the base signals for the analysis with a fixed window size. In contrast, the wavelet transform uses a large library of wavelet functions having different characteristics with a variable size window length. The sine and cosine functions forming the basis of STFT limits its applicability especially during transients which usually involve irregular wave shapes. Sine and cosine are smooth and extend from $-\infty$ to $+\infty$ making them undesirable for transient analysis. On the contrary, the wavelet library contains functions with limited time support and have different wave shapes that

are suitable for transient analysis. Moreover, the variable window size allows for better time and frequency resolution in the analysis compared to the STFT, which suffers from a fixed window size since any attempt to increase the frequency resolution comes at the expense of decreasing the time resolution and vice-versa.

1.3 Contribution

The main contribution of this thesis is to study the advantages and disadvantages of the time-frequency analysis techniques used for NILM in order to develop a systematic analytic approach that will preserve the load signal features in both time and frequency. The proposed NILM approach aims to detect common household load operation in close to real-time in order to help customers make better choices on their energy use. The advanced technique must be robust against noise and interference from other loads, able to detect loads with multiple different states, and able to detect when a new load is introduced.

In this work, the NILM problem is addressed using the wavelet transform which provides time-frequency representation of the analysis signals, through the concept of wavelet design applied to the NILM problem. In wavelet design, each load signal is matched to a pre-designed wavelet that must satisfy the wavelet filters' properties. The wavelet transform is utilized to extract specific load features from the transient signal through multi-resolution analyses. Machine learning algorithms are utilized to classify load operation based on a training set of known load data and a testing set of unknown load data. The performance of this approach is evaluated considering voltage variations, frequencies deviations, and harmonics distortion to demonstrate the robustness of the approach for different electrical supply conditions. The work presented in this thesis has

been published in several peer-reviewed journals and conferences which can be found in [13]-[16].

1.4 Thesis Organization

This thesis includes six chapters. Chapter 1 explains the motivation of the NILM and the issues with existing smart meters. This presents the need for a method to perform load disaggregation without intruding on the homeowner, which is followed by the problem statement discussing the three different analysis techniques used in NILM with emphasis on their pros and cons. Finally, the contribution of this thesis is outlined.

Chapter 2 discusses the literature review of existing NILM approaches in terms of time, frequency, or time-frequency techniques, as well as the sampling rate required for each method, and the classification accuracy achieved. Furthermore, the time-frequency transforms used in NILM are reviewed with the pros and cons of each method.

Chapter 3 introduces the wavelet transform and the wavelet design concept used in this thesis. This chapter explains the systematic analytic procedures of the wavelet design concept including the mathematical background of the wavelet transform, and its application to extract the time-frequency features of the load signal when monitored at a single point. As a powerful signal processing tool, the wavelet transform is capable of handling non-stationary signals, which is the case when considering transient signatures. Additionally, the implementation of wavelet design on two different lengths of wavelet filters to generate new wavelets with different characteristics is performed.

Chapter 4 describes the methodology used to address the NILM problem using the wavelet transform. The wavelet library is described and the concept of wavelet-signal matching is introduced through the development of the smart matching scheme. Different

transient lengths are investigated in order to determine the most ideal signal length. Moreover, machine learning classifiers are presented with k-fold cross validation and Monte Carlo to calculate classification accuracy. Finally, a test system is designed in which the system can be modelled to create a useful dataset.

Chapter 5 discusses the simulation and experimental results with a detailed analysis. A thorough introduction of the test system is presented including methods of system modelling. The algorithm is implemented in the simulation environment while being tested rigorously by varying the voltage, frequency, and harmonics of the system over several trials. The results of these trials are presented and discussed by comparing classification accuracies. Additionally, the experimental work performed in this study is described with a breakdown of the different hardware components necessary to generate a dataset. The results of the experimental test have been presented, discussed, and compared to those obtained through simulation.

Finally, Chapter 6 presents the main conclusion and recommendations regarding the time-frequency analysis technique adopted in this thesis, which has proven to be able to perform better than any other existing technique. Future work is discussed, which includes observing results on a larger wavelet library for higher orders of wavelets, and a future investigation into machine learning's role in further improving the methods success.

2. Literature Review

2.1 Introduction

The literature review is intended to provide a review of existing analysis techniques applied to NILM. By reviewing previously published work from a variety of sources the reader should be able to understand existing approaches used in NILM, the advantages and the disadvantages of each, and the research gaps that exist. The performance of each approach will be evaluated based on comparing whether the method is a time, frequency or time-frequency analysis technique, the sampling rate of the method, and the types of measurements required. At that point, the differences in methods are presented and a comparison of classification accuracies is completed. Finally, a performance metric is utilized to compare the effectiveness of each method in successfully identifying loads in different situations.

2.2 Previous Work on Non-Intrusive Load Monitoring

In the literature, previous work can be grouped based on whether the technique is time domain, frequency domain or time-frequency domain. These three analysis techniques differ in many aspects; hence the subsequent sections are dedicated for such comparison.

2.2.1 Time Domain Analysis

By far the most common methods of NILM, time domain techniques rely on extracting features contained in the time domain and in particular the active power over a given time. Usually these methods use the Root Mean Square (RMS) active power measured in 1 second, 10 second, 1 minute, 5 minute, 15 minute or 1 hour timestamps. The primary idea for time domain based NILM methods was first proposed by George W. Hart in 1992 [3]. Hart proposed that appliance level operation can be extracted by observing the

active and reactive power signals of the entire home by detecting when an appliance was switched on and comparing that to a list of known appliances. An example of the power signals Hart was referring to can be seen in Fig. 2.1 [3]. For example, a load may be switched and the change in power is detected as 2 kW. This 2 kW change is compared against a list of known loads in the home and matched to the oven knowing the oven is the only 2 kW load. Of course, there are many problems with this approach such as how to detect an appliance has changed states, multistate appliances, the rate at which data is collected, prior knowledge of the loads power draw, and struggling with loads that consume the same power. However, this idea initiated NILM research and many researchers today still compare their methods to that of Hart.

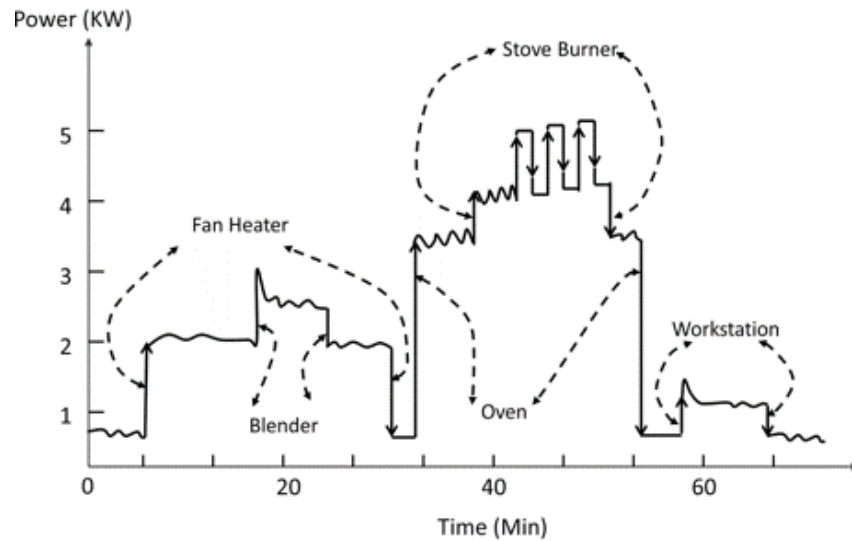


Fig. 2.1: Example Electric Power Signal Breakdown [3]

This section presents an overview of the different time domain methods that have been applied to the NILM problem. It can be observed in Table 2.1 that most time domain methods rely on fast (steady-state) sampling rates, previous knowledge of loads, and several electrical quantities. The first methods of NILM depend on examining the change in power as Hart had suggested. Mathematically, these methods can be described as

calculating active (P) and reactive (Q) power from voltage (V), current (I), and the angle between them (θ) as seen in Equation (2.1).

$$\begin{aligned} P &= VI\cos\theta \\ Q &= VI\sin\theta \end{aligned} \tag{2. 1}$$

The change in power can be calculated as the difference in power at two different time periods, as seen in Equation (2.2).

$$\begin{aligned} \Delta P &= P(t_2) - P(t_1) \\ \Delta Q &= Q(t_2) - Q(t_1) \end{aligned} \tag{2. 2}$$

Dong et al. [17], developed a windowing technique to analyze only small portions of a power signal for the change in power. By doing so, they reduced the computational complexity and only focused on a portion of the total signal. This also allowed their method to be much faster in classifying load operation than what had previously been proposed. Henao et al. [18], proposed the use of subtractive clustering as a method to detect load operation by using the change in active power. They started with the assumption that all of the different load powers summed up to the total power plus some error, which can be seen in Equation (2.3). The load operation was determined using machine learning and subtracting each known load from the total power of the home. The main limitation with this approach is the error increases when many loads are switched on at the same time.

$$P(k) = \sum_{i=1}^{N_p} p_i(k) + e_i(k) \tag{2. 3}$$

Where P is the total active power of all loads, p_i is the active power of appliance i , e is the estimated error for that appliance, and N_p is the number of loads in the home.

Another common method of NILM that was extensively discussed in the literature is appliance mapping. Appliance mapping can be described as capturing multiple different types of information and mapping them onto a physical space. The most common examples of these methods are V-I trajectory [19] or P-Q space [20]-[23]. Examples can be observed in Fig. 2.2 and 2.3 below.

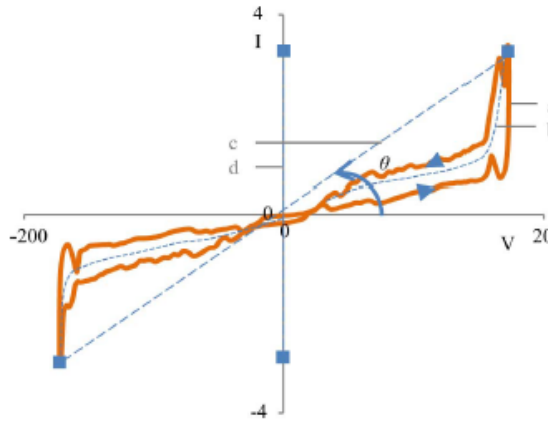


Fig. 2.2: Example of V-I Trajectory [19]

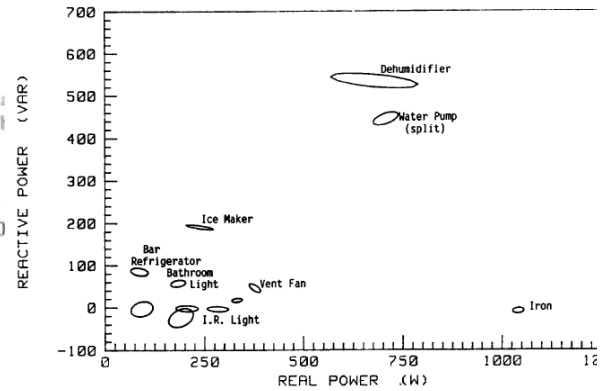


Fig. 2.3: Example of P-Q Mapping [3]

Combining this with the knowledge of different load types which may draw the same active power or current may be seen as a compelling way of determining appliance operation. The main pitfall of this method is that large resistive loads (e.g., electric oven, electric water heater, electric heat, electric dryer) all appear in the same area of the map due to their similarity in having a low reactive power component.

In addition to these methods, step changes associated with the change in power over a longer period of time have been examined. Most loads do not simply turn on momentarily, as what is assumed in the previous methods, but instead take time to start up. This causes these loads power signatures to appear to switch on in a series of steps. Other loads are multi-state, which have multiple different operations which can each appear to be a new load (e.g., washing machine has rinse, wash, spin). Bijker et al. [24] pursued this

method by using an optimization approach to discover step changes while considering the time between every step. The time and the active power magnitude are able to more accurately predict whether a load has changed states or whether a new load has been switched on, as well as what load is in operation. Le et al. [25] added to this approach by using the least absolute error, differences in state, and probability with time and the active power to determine load states in order to classify load operation.

More recently, time domain analysis techniques have shifted to Hidden Markov Models (HMM). In HMM, probability theory combined with unobserved states are utilized to build a mathematical model that can be used for applications where the state is not known, but the output associated with the state is known. This creates a useful method for NILM because it is simple to determine if a load is on (by looking at the power or current), but difficult to determine every load state for that time. An example could be if a home is consuming 1 kW, each load state is unknown, but what is known is that a load or combination of loads are on. HMM can be implemented to narrow down the number of possible different states by removing states that are not possible or that are improbable based on parameters such as time or energy consumption. Egarter et al. [26] use a HMM with Particle Filtering to detect appliance states, where Particle Filtering is a method of estimating states in nonlinear and non-Gaussian situations. Once these states have been estimated, Factorial HMM can be employed to create a model of a user's power consumption in order to perform load classification. Makonin et al. [27] define super states to create models using Factorial HMM, in order to combine several HMMs to determine load states, where Factorial HMM creates chains for each load of different states (on/off), which allows for combining HMMs for multi-state appliances. Viterbi's algorithm is

employed to keep dependencies between loads but is modified in their work for multi-state loads. Machine learning techniques are utilized to determine appliance states, as well as load operation.

Together, these analysis techniques represent the majority of different time domain approaches that exist in literature, which can be grouped into: applying the change in power, appliance mapping, observing step changes, and implementing Hidden Markov Models. Table 2.1 compares the approaches discussed earlier by considering the differences in measurements required, if a previous knowledge of loads is necessary, the sampling rate, and the overall classification accuracy. It is important to note that classification accuracies should not be compared directly due to differences in datasets, load types, sampling rates and the type of measurement required in each method.

Table 2.1: NILM Methods based on Time-domain Analysis Techniques

Type of Method	Reference	Measurements Required	Previous Knowledge of Loads	Sampling Time	Classification Accuracy
Change in Power	[3]	V, I, P, Q	Yes	1 second	N/A
	[17]	P, Q	Yes	1 second	96%
	[18]	P	Yes	1 second	86%
Appliance Mapping	[19]	V, I	Yes	1 second	99%
	[20]	P, Q	Yes	1 second	99%
	[21]	P, Q	Yes	1 second	96%
	[22]	P, Q, f	Yes	1 second	98%
	[23]	P, Q	Yes	0.1 seconds	80%
Step Changes	[24]	P	Yes	1 second	56%
	[25]	P	Yes	1 second	98%
Hidden Markov Model	[26]	P	Yes	1 second	90%
	[27]	I, P, Q	Yes	3 second	99%

2.2.2 Frequency Domain Analysis

Some existing NILM methods are based on a frequency domain analysis where harmonics and change in frequency are employed to predict load operation. Chang et al. [28] proposed an approach in which harmonics are extracted as a feature set, with Particle Swarm Optimization (PSO) implemented on the feature set repeatedly until a load is chosen. Consequently, a neural network was designed for classification of appliance operation. In [29], Electro-Magnetic Interference (EMI) signatures (also known as radio frequency interference) of loads are collected and analyzed using frequency domain analysis and Principal Component Analysis (PCA). In PCA, the feature vector (harmonics) is determined by performing a projection into PCA space followed by calculating the smallest eigenvalues for the eigenvectors, which is useful in reducing the dimensionality and determining load operation. Chong et al. [30] employed a lookup table to compare different harmonic magnitudes against a known list, which was developed for each load. In total, they investigated 15 different harmonic orders for three different loads in order to distinguish loads with a high degree of accuracy. Table 2.2 compares the frequency domain approaches by considering the differences in measurements required, if a previous knowledge of loads is necessary, the sampling rate, and the overall classification accuracy.

Table 2.2: NILM Methods based on Frequency-domain Analysis Techniques

Type of Method	Reference	Measurements Required	Previous Knowledge of Loads	Sampling Rate	Classification Accuracy
Optimization Approaches	[28]	V, I	Yes	6kHz	96%
Principal Component Null Space Analysis	[29]	EMI	Yes	2MHz	80%
Harmonic Comparison	[30]	V, I	Yes	1.8kHz	90%

2.2.3 Time-Frequency Domain Analysis

Time-frequency analysis techniques combine the benefits of both time and frequency domain techniques since they can preserve both time and frequency information. Researchers identified that a significant number of appliances cannot be detected in time or frequency domains due to similarities in load type and power consumption, but could easily be detected by analyzing the switching transients of these loads in a joint time-frequency domain. Switching transients are unique, and have been used in other power related applications such as predicting failure of a power system [31]-[32]. These switching transients occur for a few cycles, meaning they require fast sampling rates and are described as short signals which look similar to communication signals. This has led researchers to approach the NILM problem in a similar manner to communication signal processing. The STFT is a time-frequency method of performing feature extraction on a short signal by introducing a windowing function into the Discrete Fourier Transform (DFT).

$$X[f, \tau] = \sum_{n=-\infty}^{\infty} x[n]g[n - \tau]e^{-i2\pi fn} \quad (2.4)$$

The STFT can be observed in Equation (2.4), where $x[n]$ is a time domain representation of the target signal, $X[f]$ is the frequency domain representation of the target signal and $g[n]$ is the window function with a fixed window size of τ . Duarte et al. [33] proposed processing the voltage waveform using the STFT with a fast sampling rate achieving mediocre results due to the nature of the voltage switching transients. Leeb et al. [34]-[35] considered using both the voltage and current waveforms with a much lower sampling rate allowing them to achieve high classification accuracies. However, their work was not tested on a large number of different load types. Kong et al. [36] applied Cepstrum smoothing to the voltage waveform to remove noise in order to analyze the signal in the quefrency domain.

The Hilbert transform (HT) was introduced in [37] as another method of performing a time-frequency analysis of switching transients. The HT can be described as the convolution of the target signal with $1/\pi t$, which forces the integral in the convolution to diverge. The Cauchy principal value is applied in order to define the HT. In the HT, phase selectivity is employed in order to perform time-frequency analysis by shifting the positive and negative frequencies by 90° . Equation (2.5) shows the HT, where $\hat{g}(t)$ is the HT of $g(s)$.

$$\hat{g}(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{g(s)}{t - s} ds \quad (2.5)$$

The Fourier Transform (FT) of the HT can be calculated showing the relationship between the FT and HT, as seen in Equation (2.6). Where $j=\sqrt{-1}$, $\text{sgn}(f)$ is the signum function and $G(f)$ is the FT of $g(s)$.

$$\hat{G}(f) = -j\text{sgn}(f)G(f) \quad (2.6)$$

The Wavelet Transform (WT) is a time-frequency analysis representation that is suitable for nonstationary signals. The WT is unlike the STFT which uses the sine and cosine functions as the base signals for the analysis with a fixed window size, while, the WT uses a large library of wavelet functions having different characteristics with variable window length. This variable window size allows for better time-frequency resolution compared to the STFT, which suffers from a fixed window size where any attempt to increase the frequency resolution decreases the time resolution and vice-versa. Equation (2.7) demonstrates the WT where $x[t]$ is the target signal, $w[t]$ is the chosen wavelet and v and φ are the scale and shift parameters, respectively.

$$X[f] = \frac{1}{\sqrt{v}} \sum_{t=-\infty}^{\infty} x[t]w\left[\frac{t-\varphi}{v}\right] \quad (2.7)$$

In the literature, the current waveform is utilized with the WT due to the number of features contained in the current waveform when compared to the number of features found in the voltage waveform. Chang et al. [38]-[43], have presented extensive work in this area in which they have found accuracies greater than 90% for many load types while analyzing different features. Duarte et al. [33] compared their work in the STFT to the WT to demonstrate an increase in accuracy, while Mathis et al. [44] proposed technical labelling with the WT to improve accuracy.

These techniques use time-frequency analysis to demonstrate a strong representation of the time-frequency approaches that exist in the literature. Each of these different methods can be categorized into: the short time Fourier transform, the Hilbert transform, and the wavelet transform. Table 2.3 compares these techniques by analyzing the different measurements required, if a previous knowledge of loads is imperative, the sampling rate, and the overall classification accuracy. The reader should note that the majority of these methods utilize a sampling rate of 3-30kHz and the current waveform to perform NILM.

Table 2.3: NILM Methods based on Time-Frequency Domain Analysis Techniques

Type of Method	Reference	Measurements Required	Previous Knowledge of Loads	Sampling Rate	Classification Accuracy
Short Time Fourier Transform	[33]	V	Yes	2.605MHz	71%
	[34] [35]	V, I	Yes	7.7kHz	100%
	[36]	V	Yes	5kHz	96%
Hilbert Transform	[37]	I	Yes	1kHz	93%
Wavelet Transform	[33]	V	Yes	2.605MHz	80%
	[38]	I	Yes	15.36kHz	100%
	[39]	I	Yes	15.36kHz	96%
	[40]	I	Yes	7.68kHz	100%
	[41]	V, I	Yes	3.84kHz	88%
	[42]	I	Yes	15.36kHz	90%
	[43]	I	Yes	30.72kHz	96%
	[44]	V, I	Yes	5kHz	95%

2.3 Performance Metrics

The performance metric of each method is calculated based on the criteria described below. Each technique will be compared against the others in time, frequency and time-frequency analysis. Three metrics will be used to compare each method, the different electrical measurements required, the sampling rate and the corresponding classification

accuracy. For different electrical measurements, a higher score represents if fewer measurements are necessary as well as how simple those measurements are to collect. For different sampling rates, a higher score will be assigned to lower sampling rates as a lower sampling rate suggests less data is required leading to small storage size with less computational burden. Lastly, classification accuracy is the actual ability of each method to correctly predict load operation. All of these metrics represent a numeric quantity that can be compared in a scientific manner.

A tool known as pairwise comparison can be employed to weight and judge individual metrics so a final performance score can be calculated. Table 2.4 shows the pairwise comparison for the three NILM performance metrics. A weight for individual metrics is calculated using Equation (2.8) based on the importance of that metric relative to the other metrics. It can be observed that classification accuracy is highest ($w=0.5$), while the sampling rate and the measurements required are equal ($w=0.25$). This becomes clear as it is expected that classification accuracy be the most important since if the algorithm is not accurate the other metrics are insignificant.

$$w_i = \frac{\sum_{j=0}^L m_{i,j}}{\sum_{k=0}^L \sum_{j=0}^L m_{k,j}} \quad (2.8)$$

where w_i is the weight for row i , L is the number of metrics, and k and j correspond to the row and column, respectively.

Table 2.4: Pairwise Comparison of Performance Metrics

	Measurements Required	Sampling Rate	Classification Accuracy	Weighting
Measurements Needed	1	1	1/2	0.25
Sampling Rate	1	1	1/2	0.25
Classification Accuracy	2	2	1	0.5

As a result of the numeric comparison of the performance metrics, a numeric way of quantifying a score for each technique is required. Particular metrics need their own method of defining a scale. For the required measurements, the scale can be determined based on more quantities which represents a lower score in a linear fashion, where not all measurements are equal. This is caused by some metrics being a combination of others (i.e., Power is the product of voltage and current) while others are more difficult to measure. Therefore, each measurement will be given the scores presented in Table 2.5 where a combination of metrics will have a lower score.

Table 2.5: Measurement Type Scores

Metric	Score (%)
Voltage (V)	100
Current (I)	100
Voltage and Current	50
Active Power (P)	50
Reactive Power (Q)	50
Apparent Power (S)	50
Active, Reactive and Frequency	33
EMI	33

Moreover, a scoring system for the sampling rate is also required. The sampling rate is a difficult metric to evaluate due to the vast difference in the analysis types. In time-

domain analysis, RMS measurements are taken at certain time steps, in order to calculate RMS values using Nyquist sampling theorem. In other words, the minimum sampling rate required to calculate RMS measurements in theory must be at least 120Hz (60Hz×2). However, in reality, the sampling rate must be higher than 120Hz to account for sampling the waveform in poor places such as at the zero crossing. Therefore, the minimum sampling rate recommended for time domain analysis is 240Hz. Nevertheless, in time-frequency domain analysis the fastest sampling rate called for in the literature is 2.605MHz. Using this information, maximum and minimum scores are determined based on the slowest sampling rate of 240Hz and the highest sampling rate of 3MHz. Accordingly, 240Hz corresponds to a score of 100% while 3MHz corresponds to a score of 0%. Furthermore, a sampling rate of 50kHz can be selected as the middle value (50%), considering most common DAQ / A2D hardware samples up to 50kHz, where anything faster requires more expensive hardware. Knowing these three points, a scoring line can be created using a logarithmic regression line, which is chosen due to the fact that it quickly decreases to its mid-point of (50%), allowing it to be ideal in order to compare methods that have similar sampling rates in the 2-30kHz region. Logarithmic regression lines decrease at a slower pace for the remaining 50% so as to not significantly hinder methods with faster sampling rates. The equation of the logarithmic regression line can be calculated using the maximum (3MHz), minimum (240Hz) and midpoint (50kHz), which can be seen in Equation (2.9). The plot of the logarithmic regression line can be seen in Fig. 2.2.

$$y = -10.54 \times \ln(x) + 159.66 \quad (2.9)$$

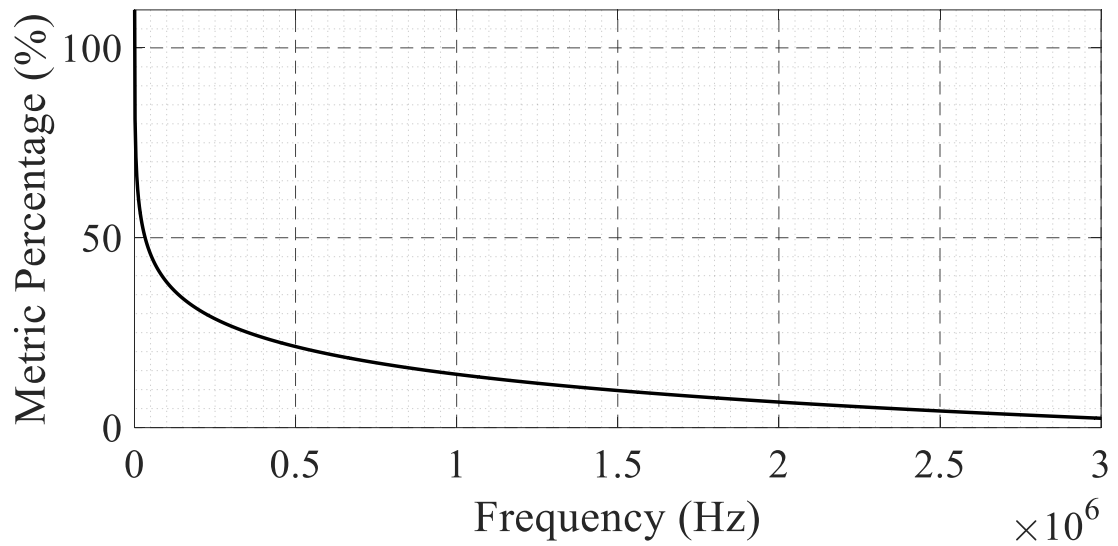


Fig. 2.4 Logarithmic Regression Line for a Performance Metric of Sampling Frequency

The last metric that needs to be determined is for classification accuracy. This metric can be taken directly as the percentage of correct classifications between 0 and 100%.

2.3.1 Calculating the Performance of Each Method

As a result of the performance evaluation system, each method can be compared against the three performance metrics discussed in the section above. Specifically, the measurements required, the sampling rate, and the classification accuracy. At this point, a final score can be calculated for individual techniques based on the weightings shown in Table 2.4. In Table 2.6, metrics are calculated for each method, and a final score is presented.

Table 2.6: Performance of Each Method

Analysis Technique	Method	Reference	Measurements Required (%)	Sampling Rate (%)	Classification Accuracy (%)	Final Score (%)
Time Domain Analysis	Change in Power	[3]	50	100	N/A	N/A
		[17]	50	100	96	85.5
		[18]	50	100	86	80.5
	Appliance Mapping	[19]	50	100	99	87
		[20]	50	100	99	87
		[21]	50	100	96	85.5
		[22]	33	100	98	82.25
		[23]	50	100	80	77.5
	Step Changes	[24]	50	100	56	65.5
		[25]	50	100	98	86.5
	Hidden Markov Model	[26]	50	100	90	82.5
		[27]	50	100	99	87
Frequency Domain Analysis	Optimization Approaches	[28]	50	68	96	77.7
	Principal Component Null Space Analysis	[29]	33	7	80	50
	Harmonic Comparison	[30]	50	81	90	77.75
Time-Frequency Domain Analysis	Short Time Fourier Transform	[33]	100	4	71	61.5
		[34] [35]	50	65	100	78.75
		[36]	100	70	96	90.5
	Hilbert Transform	[37]	100	87	93	93.25
	Wavelet Transform	[33]	100	4	80	66
		[38]	100	58	100	89.5
		[12]	100	58	96	87.5
		[40]	100	65	100	91.25
		[41]	50	73	88	74.75
		[42]	100	58	90	84.5
		[43]	100	51	96	85.75
		[44]	50	70	95	77.5

2.4 Summary

In this chapter, three different analysis techniques of performing NILM have been investigated: time domain analysis, frequency domain analysis, and time-frequency domain analysis. These techniques all differ in many ways and as such were investigated separately in this literature review. Each method of NILM was explained and sampling rates were compared to demonstrate the variety of different approaches and signal types that have attempted in literature. A performance metric was developed in order to compare each method. In Table 2.6, the performance metrics in bold represent the best attempt to date for the methods discussed in this literature review. In order to further research in the area of NILM, two different directions exist. In time domain analysis, HMM is in the area of interest around methods of mathematical modeling different states. In time-frequency domain analysis, the WT has produced intriguing results based on feature extraction to identify loads. Several unanswered questions still exist such as: the choice of wavelet to use in the WT, as well as the ideal length of wavelet filter.

The subsequent chapters in this thesis will answer these questions by developing a systematic approach that can be employed for different loads in many possible configurations under all load conditions.

3. Orthogonal Wavelet Design

3.1 Introduction

Time-frequency analysis techniques provide a time-frequency representation of the analysis signal which allows for the signal information to be preserved in both time and frequency. The wavelet transform as a time-frequency representation of the signal, has a large library of wavelet families that have different wave shapes which offers great flexibility in representing the waveform in the time and frequency domains. To date, little research has been conducted regarding the choice of the wavelet functions in the NILM application. This chapter will thoroughly describe the discrete wavelet transform, followed by a detailed explanation of how the energy of the coefficients can be calculated using multi resolution analyses for feature extraction. Additionally, the wavelet design concept will be introduced, including an explanation of the process of generating a family of new wavelets. Finally, two different methods of choosing a unique wavelet from the set of designed wavelets will be presented. These two methods are Procrustes and covariance analysis, both of which are common mathematical methods of comparing two different shapes.

3.2 Discrete Wavelet Transform

This section introduces the Fourier Transform (FT), the STFT, and the WT. The Discrete FT can be described mathematically in Equation (3.1).

$$X[f] = \frac{1}{N} \sum_{t=1}^{N-1} x[t] e^{-i2\pi ft/N} \quad (3.1)$$

In Equation (3.1), the target signal is represented as $x(t)$, the function to perform the FT is $e^{-i2\pi ft/N}$, and the frequency representation of the target signal is $X(f)$, where t is time, f is the frequency, and N is the length of the signal. The FT has several complications in signal processing, first, the window size is fixed since the FT is computed for the entire signal at once, which creates problems in the case of non-stationary signals. Moreover, the time information in the signal is lost since the transform provides only amplitude-frequency spectrum of the signal. Second, the basic functions used to compute the FT are fixed (i.e., sine or cosine), proving the solution space in which the transform exists is always the same. The first complication of having a fixed window size for the entire function can be partly corrected by inserting a fixed moving window into the transform. This change allows a fixed window size over smaller sections of the waveform, which is the STFT.

In Equation (2.4), the STFT is presented with the only adjustment being the insertion of the moving window function $g[t - \tau]$, where τ is the amount in which the window shifts at one time. The addition of the moving window in the time domain creates a new issue. As the moving window in time gets smaller, the frequency resolution gets higher but at a poor time resolution. This trade-off should be realized when using the STFT. In order to address the previous complication of a fixed window size, as well as the problem of using the basic sine and cosine functions, the wavelet transform was developed. The WT has a non-fixed window size which allows for more than one function to be used in the transform. To change the window size, two parameters have been used in the transform, these parameters are v and ϕ , which signify the time scale and shift parameters, respectively. In addition, the basic functions in the FT and STFT can be replaced by a function, $w(t)$ which can be described as the mother wavelet as seen in Equation (2.7). The mother wavelet $w(t)$,

is a compactly supported function for which the baby wavelets form an orthogonal basis. Baby wavelets are scaled and shifted waveforms of mother wavelets where baby wavelets follow two main conditions: wavelet coefficients must be normalized and wavelets must follow double shift orthogonality. These conditions will be explained in more detail in the wavelet design section of this chapter.

Additionally, the WT can be mathematically represented using Multi-Resolution Analysis (MRA). In MRA, the signal $x[t]$ is broken down into a sum of its high and low frequency components. These components form an orthonormal basis, and are linked to the low pass and high pass filters by Equation (3.2). For more information on MRA, the reader may refer to [45].

$$\begin{aligned} V(t) &= \sum_k h_0(k) \sqrt{2} V(2t - k) \\ W(t) &= \sum_k h_1(k) \sqrt{2} V(2t - k) \end{aligned} \tag{3.2}$$

Where h_0 is the low pass filter coefficients, h_1 is the high pass filter coefficients, V is the scaling function and W is the wavelet function.

MRA can be described as the signal passing through a series of filters which will split the signal into its low and high frequency components. This process is called decomposition, where the number of decomposition levels (filters) is related to the sampling frequency. For each decomposition level, there is an approximation of the signal and a number of details that correspond to the decomposition level. The signal can be synthesized from the approximation (low frequency) and the details (high frequency). The approximation and each detail represent a separate frequency range, which, therefore correspond to the series of filters. These coefficients have extracted unique parts of the

signal from that frequency range which make them useful in feature extraction. An example of three levels of decomposition for a target signal using the wavelet ‘db5’ can be seen in Fig. 3.1 where the MATLAB Wavelet Toolbox was utilized to complete the decomposition [46]:

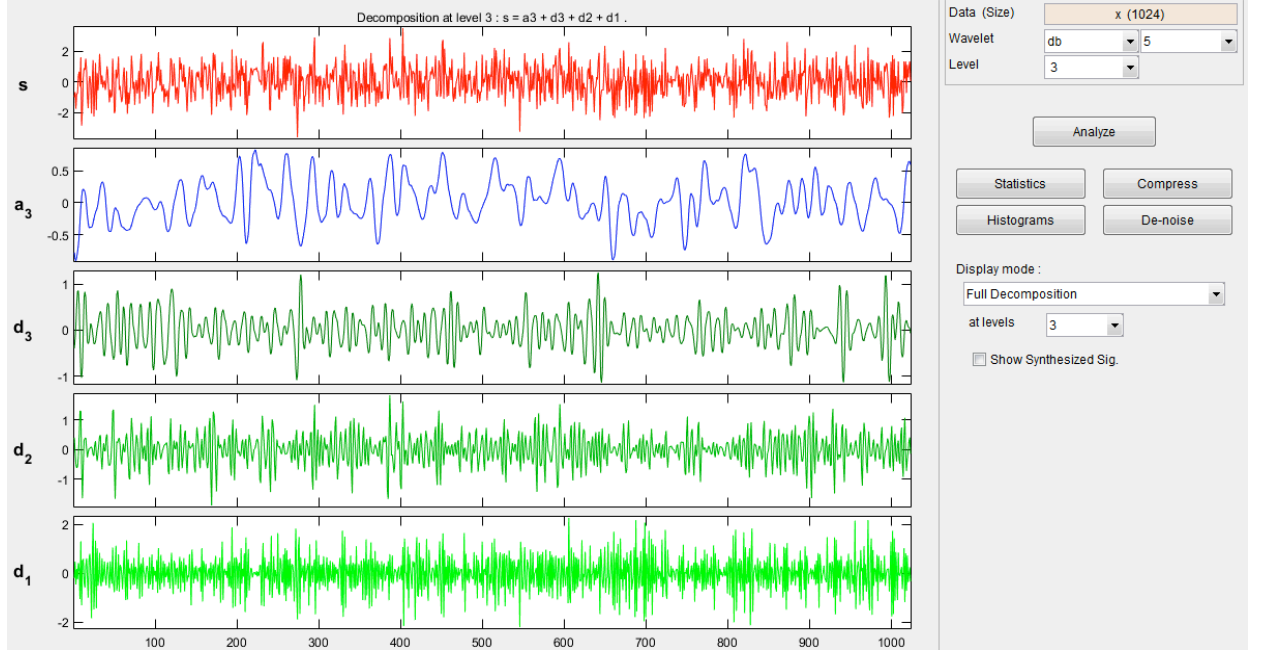


Fig. 3.1: Multi-resolution Analysis Example

Where s is the target signal, a_3 is the approximation for three levels of decomposition, and d_1 , d_2 and d_3 are the details for each level.

$$\begin{aligned} cA_d^j(l) &= \langle X, V_{j,l} \rangle \\ cD_d^j(l) &= \langle X, W_{j,l} \rangle \end{aligned} \quad (3.3)$$

Mathematically, this can be represented in Equation (3.3) where A is the approximation, D is the detail and j is the decomposition level.

3.3 Feature Extraction and Identification

In the wavelet transform, the details and approximation of the wavelet transform are unique to the signal being transformed which can be described as features. The energy of each of the coefficients can be calculated, as to not have a signal for each feature, but rather a single set of values that can be considered as the features for each waveform. To calculate the energy of the coefficients for the approximation and the details, Equation (3.4) is used.

$$\begin{aligned} E_{cA_d^j} &= \sum_l |cA_d^j(l)|^2 \\ E_{cD_d^j} &= \sum_l |cD_d^j(l)|^2 \end{aligned} \tag{3.4}$$

3.4 Wavelet Design Procedures

3.4.1 Introduction

In general, wavelets are time localized signals that share a special set of properties (e.g., normalization, double shift orthogonality) that allow them to be used to extract features from a target signal. Previous work on wavelets utilized a standard set of wavelets, due to their ease of use and inclusion in the MATLAB Wavelet Toolbox [46]. However, the small section of wavelets families represented in the MATLAB Wavelet Toolbox (i.e., Daubechies (db) family of wavelets which are compactly supported orthonormal wavelets), do not represent the full scope of wavelets that exist as well as what can be used in power system applications such as NILM. In fact, the wavelet families that exist in MATLAB are usually better suited to other signal processing applications such as compression, video or image processing. Therefore, this section will thoroughly describe the design constraints around wavelets, as well as create a new family of wavelets for the NILM application. Lastly, a library of wavelets will be created for several different lengths of wavelet filters.

3.4.2 Wavelet Filter Properties

In order to design wavelets, the properties of wavelets must first be examined. The two scale equations described in Equation (3.2) can be used as a starting point in this investigation. Specifically, by substituting $t-n$ for t into the scaling function $V(t)$, it becomes delayed by the integer n .

$$V(t-n) = \sum_k h_0(k) \sqrt{2} V(2(t-n)-k) = \sum_k h_0(k) \sqrt{2} V(2t-(2n+k)) \quad (3.5)$$

Next, $m=k+2n$ can be substituted into Equation (3.5).

$$V(t-n) = \sum_m h_0(m-2n) \sqrt{2} V(2t-m) = \sum_k h_0(k-2n) \sqrt{2} V(2t-k) \quad (3.6)$$

Similarly, the same procedure can be followed for the wavelet function $W(t)$.

$$W(t-n) = \sum_k h_1(k-2n) \sqrt{2} V(2t-k) \quad (3.7)$$

This leads to a compelling result, since the time shift of the scaling and wavelet function are orthogonal as seen in Equation (3.8), where $\delta(t)$ is the delta-dirac function.

$$\begin{aligned} \int_{-\infty}^{\infty} V(t)V(t-n)dt &= \delta(t) = \sum_k h_0(k)h_0(k-2n) \\ \int_{-\infty}^{\infty} W(t)W(t-n)dt &= \delta(t) = \sum_k h_1(k)h_1(k-2n) \\ \int_{-\infty}^{\infty} V(t)W(t-n)dt &= 0 = \sum_k h_0(k)h_1(k-2n) \end{aligned} \quad (3.8)$$

In addition, this can be utilized to determine the properties of the low pass and high pass filters.

Property 1: $n=0$

$$\sum_k h_0^2(k) = 1, \sum_h h_1^2(k) = 1 \quad (3.9)$$

Property 2: $n \neq 0$

$$\sum_k h_0(k)h_0(k-2n) = 0, \sum_k h_1(k)h_1(k-2n) = 0 \quad (3.10)$$

Property 3: $h_1(k)$ is an alternating flip of $h_0(k)$, therefore there is an odd integer M that:

$$h_1(k) = (-1)^k h_0(M-k) \quad (3.11)$$

Where $h_0(k)$ is low pass since:

$$\sum_k (-1)^k h_0(k) = 0 \quad (3.12)$$

These properties as described in Equations (3.9) – (3.12) form the minimum requirements necessary to design the low pass filter $h_0(k)$ and the high pass filter $h_1(k)$.

3.4.3 Design Fundamentals

3.4.3.1 Length-2 Filter

In the case of wavelet design for the length-2 filter, the two filter coefficients are $h_0(0) = \pm 1/\sqrt{2}$ and $h_0(1) = h_0(0)$. This leads to a unique solution using properties 1 and 3, but does not provide any degrees of freedom to conduct wavelet design. Therefore, wavelet design in the length-2 filter is not feasible, as such it will not be presented in this thesis.

3.4.3.2 Length-4 Filter

In the case of wavelet design for the length-4 filter, a parametric solution exists using properties 1 and 3, which only provides one degree of freedom to conduct wavelet design.

This significantly limits the number of wavelet design possibilities for the length-4 filter rendering it useless for NILM. The target goal of this wavelet design process is to match wavelets based on their shape to unique loads in order to perform NILM. But in the case of the length-4 filter, there is not enough variety in shape to perform a pattern analysis. Therefore, wavelet design in the length-4 filter will not be presented in this work.

3.4.3.3 Length-6 Filter

In the case of wavelet design for the length-6 filter, a solution with two degrees of freedom exists with filter coefficients $h_0(0)$, $h_0(1)$, $h_0(2)$, $h_0(3)$, $h_0(4)$, $h_0(5)$. When substituting these filter coefficients into the three properties described in Equations (3.9) - (3.12), the following equations for h_0 can be determined:

Property 1:

$$h_0^2(0) + h_0^2(1) + h_0^2(2) + h_0^2(3) + h_0^2(4) + h_0^2(5) = 1 \quad (3.13)$$

Property 2:

$$\begin{aligned} h_0(0)h_0(2) + h_0(1)h_0(3) + h_0(2)h_0(4) + h_0(3)h_0(5) &= 0 \\ h_0(0)h_0(4) + h_0(1)h_0(5) &= 0 \end{aligned} \quad (3.14)$$

Property 3:

$$h_0(0) - h_0(1) + h_0(2) - h_0(3) + h_0(4) - h_0(5) = 0 \quad (3.15)$$

These properties create four equations and six unknowns, thus allowing two degrees of freedom. A solution in terms of two parameters, a and b is calculated.

$$\begin{aligned}
h_o(0) &= \frac{[(1 + \cos(a) + \sin(a))(1 - \cos(b) - \sin(b)) + 2 \sin(b) \cos(a)]}{4\sqrt{2}} \\
h_o(1) &= \frac{[(1 - \cos(a) + \sin(a))(1 + \cos(b) - \sin(b)) - 2 \sin(b) \cos(a)]}{4\sqrt{2}} \\
h_o(2) &= \frac{[1 + \cos(a - b) + \sin(a - b)]}{2\sqrt{2}} \\
h_o(3) &= \frac{[1 + \cos(a - b) - \sin(a - b)]}{2\sqrt{2}} \\
h_o(4) &= \frac{1}{\sqrt{2}} - h_o(0) - h_o(2) \\
h_o(5) &= \frac{1}{\sqrt{2}} - h_o(1) - h_o(3)
\end{aligned} \tag{3.16}$$

Which can be simplified using trigonometric identities and setting $c=\pi/4 -a-b$ to:

$$\begin{aligned}
h_o(0) &= \cos(c) \times \cos(b) \times \cos(a) \\
h_o(1) &= \cos(c) \times \cos(b) \times \sin(a) \\
h_o(2) &= -\cos(c) \times \sin(b) \times \sin(a) - [\sin(c) \times \sin(b) \times \cos(a)] \\
h_o(3) &= -\cos(c) \times \sin(b) \times \sin(a) - [\sin(c) \times \sin(b) \times \sin(a)] \\
h_o(4) &= -\sin(c) \times \cos(b) \times \sin(a) \\
h_o(5) &= \sin(c) \times \cos(b) \times \cos(a)
\end{aligned} \tag{3.17}$$

The final step in the wavelet design process is to choose numeric values for a and b . The goal of this design process is to design a set of wavelets to be matched to load signals. Another important note is that a and b are only used inside sine and cosine functions.

Therefore, a and b can be varied between $-\pi$ and π to solve for all possible combinations of wavelet filter coefficients. In this manner, all of the different possibilities of wavelet shapes are incorporated into the design process. Accordingly, a resolution is required in order to determine a quantity to vary each parameter by. A resolution of $0.1 \times \pi$ is selected as it was observed experimentally that any further increase in resolution did not produce a significant change in wavelet shape. This computation was performed in MATLAB for each of the 21 steps ($-0.1 \times (-\pi \text{ to } \pi)$) for a and b , respectively, where 441 ($21^2=441$) different wavelets were designed.

3.4.3.4 Length-8 Filter

In the case of wavelet design for the length-8 filter, a solution with three degrees of freedom exists, where the filter coefficients are $h_0(0)$, $h_0(1)$, $h_0(2)$, $h_0(3)$, $h_0(4)$, $h_0(5)$, $h_0(6)$, $h_0(7)$. When substituting these filter coefficients into the three properties presented in Equations (3.9) - (3.12), the following equations for h_0 can be determined:

Property 1:

$$h_0^2(0) + h_0^2(1) + h_0^2(2) + h_0^2(3) + h_0^2(4) + h_0^2(5) + h_0^2(6) + h_0^2(7) = 1 \quad (3.18)$$

Property 2:

$$\begin{aligned} h_0(0)h_0(2) + h_0(1)h_0(3) + h_0(2)h_0(4) + h_0(3)h_0(5) + h_0(4)h_0(6) \\ + h_0(5)h_0(7) = 0 \\ h_0(0)h_0(4) + h_0(1)h_0(5) + h_0(2)h_0(6) + h_0(3)h_0(7) = 0 \\ h_0(0)h_0(6) + h_0(1)h_0(7) = 0 \end{aligned} \quad (3.19)$$

Property 3:

$$h_0(0) - h_0(1) + h_0(2) - h_0(3) + h_0(4) - h_0(5) + h_0(6) - h_0(7) = 0 \quad (3.20)$$

These properties create five equations and eight unknowns, thus allowing three degrees of freedom. A solution is determined in terms of three parameters, d , e and f , where $g = \pi/4 - d - e - f$.

$$\begin{aligned} h_0(0) &= \cos(g) \times \cos(f) \times \cos(e) \times \cos(d) \\ h_0(1) &= \cos(g) \times \cos(f) \times \cos(e) \times \sin(d) \\ h_0(2) &= -\cos(g) \times \cos(f) \times \sin(e) \times \sin(d) - [\cos(g) \times \sin(f) \times \sin(e) \times \cos(d)] \\ &\quad - [\sin(g) \times \sin(f) \times \cos(e) \times \cos(d)] \\ h_0(3) &= \cos(g) \times \cos(f) \times \sin(e) \times \cos(d) - [\cos(g) \times \sin(f) \times \sin(e) \times \sin(d)] \\ &\quad - [\sin(g) \times \sin(f) \times \cos(e) \times \sin(d)] \\ h_0(4) &= -\cos(g) \times \sin(f) \times \cos(e) \times \cos(d) - [\sin(g) \times \sin(f) \times \sin(e) \times \sin(d)] \\ &\quad - [\sin(g) \times \cos(f) \times \sin(e) \times \cos(d)] \\ h_0(5) &= \cos(g) \times \sin(f) \times \cos(e) \times \cos(d) - [\sin(g) \times \sin(f) \times \sin(e) \times \cos(d)] \\ &\quad - [\sin(g) \times \cos(f) \times \sin(e) \times \sin(d)] \\ h_0(6) &= -\sin(g) \times \cos(f) \times \cos(e) \times \sin(d) \\ h_0(7) &= \sin(g) \times \cos(f) \times \cos(e) \times \cos(d) \end{aligned} \quad (3.21)$$

Similarity to the length-6 filter design, the last step in the design process is to determine numeric values for d , e and f . Hence, d , e and f will be varied between $-\pi$ and π , to match the length-6 design process, which creates all of the different possibilities of

wavelets. Finally, a resolution of $0.1 \times \pi$ is implemented for each possible parameter as it was found experimentally that any additional increase in resolution does not produce a variation in change in shape. Overall, this computation was performed in MATLAB for each of the 21 steps of d , e and f , respectively where 9261 ($21^3=9261$) different wavelets were designed.

3.4.4 Wavelet Clustering

In section 3.4.3, a set of wavelets of length-6 and 8 were designed totaling 9,702 different wavelets. This large number of wavelets can cause computational problems due to the high computational complexity which requires powerful processing and large storage capacity. Therefore, a clustering technique can be developed to group wavelets with similar shapes, without removing wavelets that possess a unique shape. Unsupervised k- means clustering [47] is implemented based on the Euclidean distance of the wavelet against all other wavelets in the set. This process is utilized to group wavelets of the same filter length [47]. The Euclidean distance of two wavelets, W_1 and W_2 is calculated based off of Equation (3.22):

$$ED = \sqrt{\sum_L [W_1(l) - W_2(l)]^2} \quad (3.22)$$

3.4.4.1 K-means Clustering

K-means is a well-known method which can partition a dataset, x , into k clusters based on a chosen distance metric. In the first iteration, a random set of k starting points is chosen as the initial centroids, where each point in x is evaluated against the centroids using the distance metric seen in Equation (3.22). K-means sets each point into its closest cluster

which forms the preliminary clusters. During the subsequent iterations, the centroids are calculated using the mean of all current points in the cluster where each point in x is once again evaluated against the centroids. If a point x_i is found to be closer to a new cluster, it is moved towards that new cluster. This process is repeated until either k-means cannot improve the clusters centroid location, which signifies that a solution has been determined, or a maximum number of iterations is reached. In this work, 100 iterations have been selected as it was experimentally observed that after 100 iterations, the error of the clusters did not improve. A number of iterations had to be determined because k-means has the possibility of running for a prolonged period of time, where the possibility of finding a feasible solution might not be within the time constraints available.

Once this process was complete for all wavelet pairs, a vector of EDs representing the similarity between wavelet pairs was utilized to cluster the wavelets. In order to identify the most suitable number of clusters, the Sum of Squares Error (SSE) can be computed for all clusters N .

3.4.4.2 Sum of Square Error (SSE)

The SSE is the evaluation of a point's distance to its cluster centroid. Using this metric, it is viable to observe the variance of cluster points from the ideal cluster centroid. The formula for the SSE is observed in Equation (3.23).

$$SSE = \sum_{i=1}^N \sum_{\gamma \in Z} ED(\bar{\gamma}_i, \gamma)^2, \text{ where } \bar{\gamma}_i = \frac{1}{\mu_i} \sum_{\gamma \in Z} \gamma \quad (3.23)$$

Where Z_i is the i^{th} cluster, and $\bar{\gamma}_i$ is the centroid of cluster Z_i with μ_i data objects in the i^{th} cluster. When using clustering techniques, it is known that increasing the number of

clusters reduces the SSE. The Silhouette coefficient (S_i) is utilized to determine the optimal number of representative clusters / wavelets.

3.4.4.3 Silhouette Coefficient

The silhouette coefficient [47] is one of the metrics utilized to interpret the validity of clusters of data. Silhouette is an effective measure because it evaluates a point's distance to every other point in the same cluster, a_i , where the same point's distance is compared to points in a neighboring clusters, b_i . The equation for the silhouette coefficient can be seen in Equation (3.24).

$$S_i = \frac{(b_i - a_i)}{\max(a_i, b_i)} \quad (3.24)$$

A point is considered well fitted when S_i equals 1, and the whole set of clusters is considered well partitioned when the mean of all S_i equals 1.

3.4.4.4 Golden Section Search

Golden section search [48] is an optimization technique which employs using the golden ratio to find an optimal solution of a unimodal problem. Golden section search is a numerical optimization algorithm which searches within a boundary for a solution that is either a maximum or minimum. By evaluating a function at 38.2% and 61.8% from within its boundaries, it is possible to determine the direction the function is pointing toward [48]. By shifting the lower or upper boundary during each iteration, the boundary of the problem is restricted to a smaller area where the optimized solution exists.

3.4.4.4.1 Optimal Cluster Evaluation

In order to evaluate which k value is ideal in k -means, the boundary of the problem is restricted to $k = 1, 2, 3, \dots, k_{max}$, where k_{max} and q can be calculated using the equations in (3.25).

$$\begin{aligned} k_{max} &= \frac{n}{Fib_q} \\ q &= \frac{wavelet_length}{2} + 1 \end{aligned} \tag{3.25}$$

Where, n is the number of wavelets in the dataset and q is the index number for the required Fibonacci number [48].

As a result, the number of wavelets in the dataset gets exponentially larger when increasing the length of the wavelet coefficients. Therefore, a way to preprocess the range where the optimal solution exists is required. By using Fibonacci's exponential rate of increase, it is realistic to choose a number, based on the length of the wavelet, which restricts where the optimal k value resides. In Equation (3.26), an example of how to find this optimal value for length-6 is presented, in order to find the corresponding q value. Where Fib_4 is the 4th Fibonacci number.

$$\begin{aligned} q &= \frac{6}{2} + 1 = 4, \quad Fib_4 = 3 \\ k_{max} &= \frac{441}{3} = 147 \end{aligned} \tag{3.26}$$

Golden section search can be utilized to process k -means at the two points where the golden ratio resides. k -means is implemented with 10 replications to increase the

chances at which the precise solution at the specified k value is determined. The Silhouette coefficient is calculated for each golden section split, as such the boundary gets smaller based on the point that has the greater silhouette coefficient. This is completed for as long as necessary until golden section search converges onto one k value that generates a silhouette coefficient of 1. Below, in Fig. 3.2 and Fig. 3.3 is a brief example of applying this method to length-6 for 3 iterations to show how golden section converges onto the optimal k value, where W. is short for Wavelets:

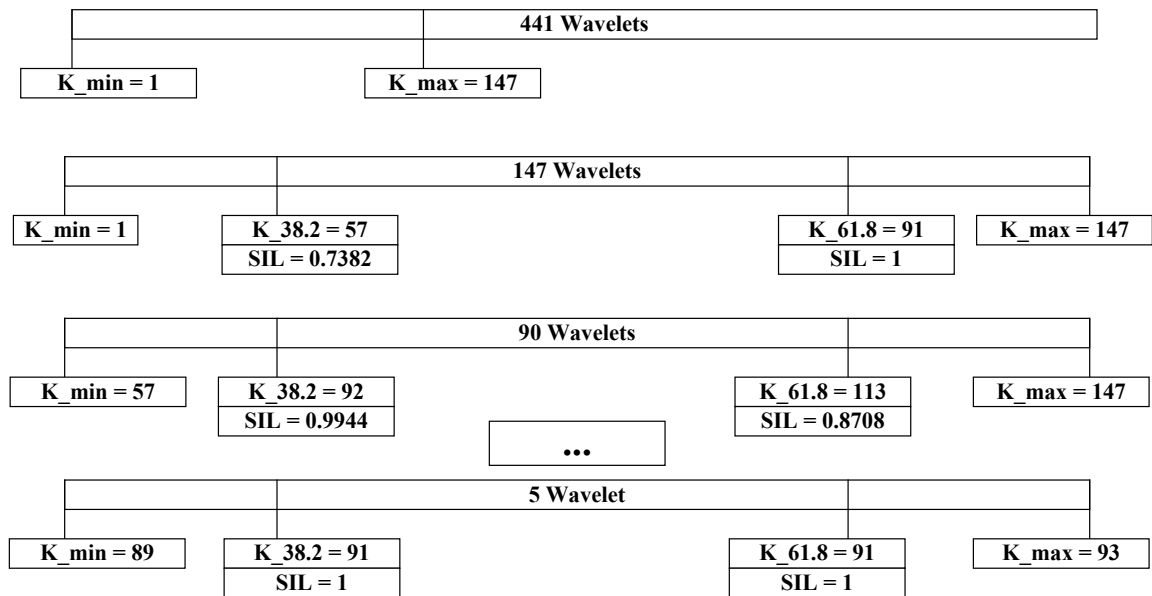


Fig. 3.2: Length-6 Golden Section Example for 3 Iterations

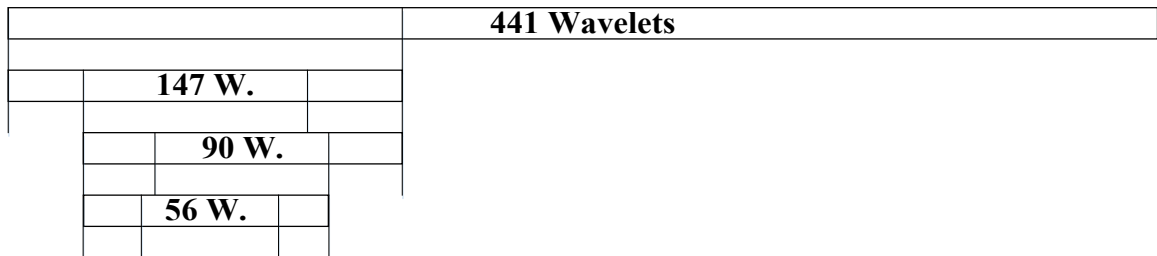


Fig. 3.3: Golden Section Converging onto Optimal k value

3.4.4.5 Length-6 Wavelet Clustering

In the case of length-6, the optimal number of clusters was determined to be 91 based on a Silhouette coefficient of 1 and an SSE of 4.825×10^{-29} . Fig. 3.4 and Fig. 3.5 present the SSE and Silhouette coefficient in the case of the length-6 filter.

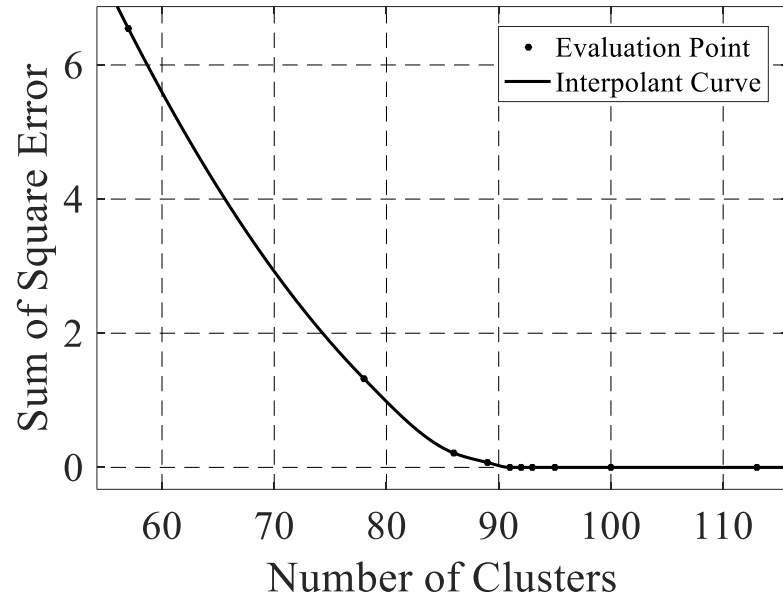


Fig. 3.4: Sum of Squares Error vs Number of Representative Wavelets

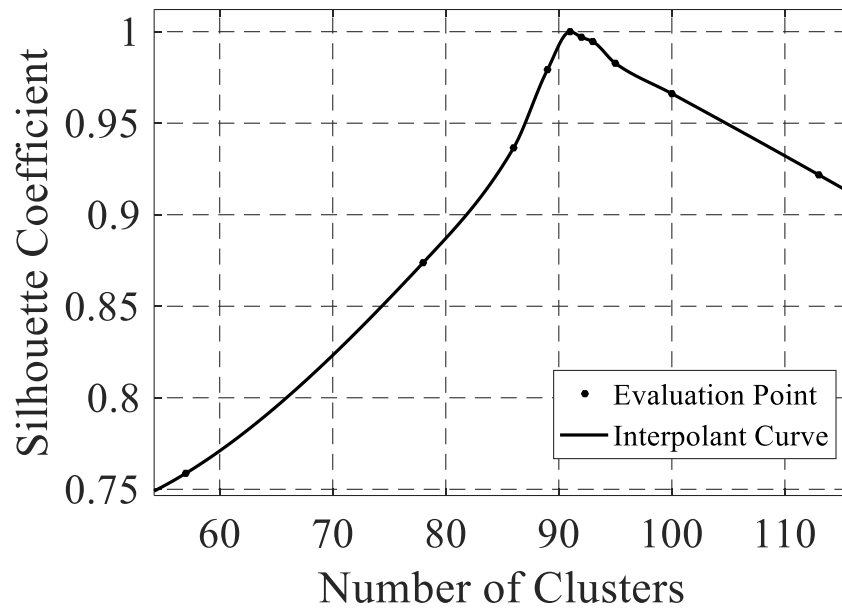


Fig. 3.5: Silhouette Coefficient vs Number of Representative Wavelets

3.4.4.6 Length 8 Wavelet Clustering

In the case of length-8, the optimal number of clusters was determined to be 820 based on a Silhouette coefficient of 1 and an SSE of 1.273×10^{-26} . Fig. 3.6 and Fig. 3.7 present the SSE and Silhouette coefficient in the case of the length-8 filter.

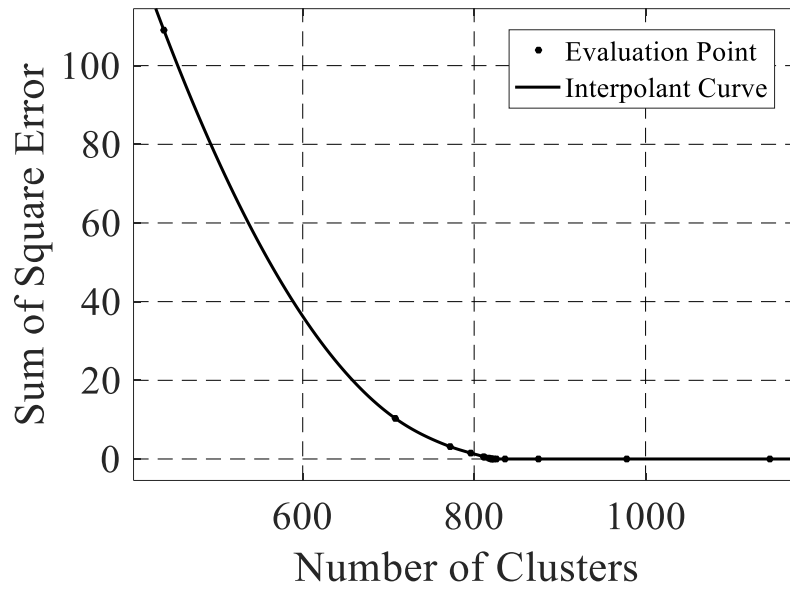


Fig. 3.6: Sum of Squares Error vs Number of Representative Wavelets

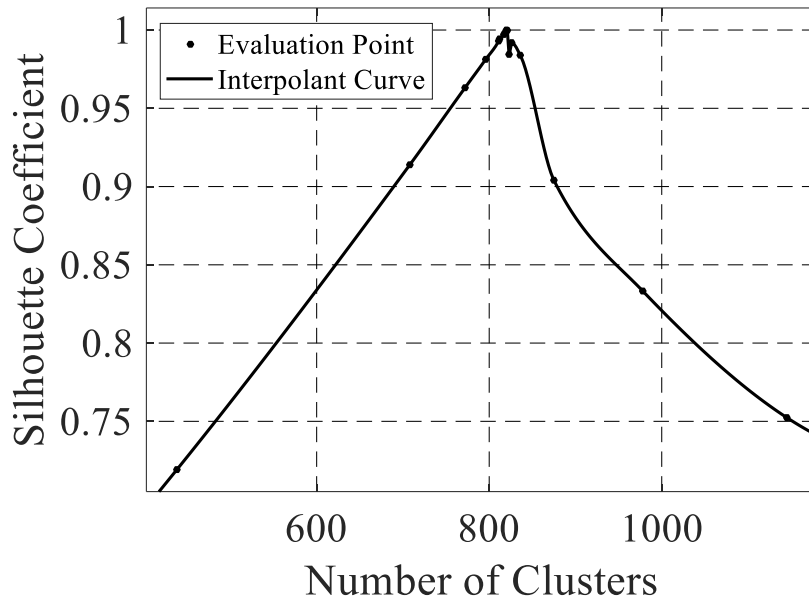


Fig. 3.7: Silhouette Coefficient vs Number of Representative Wavelets

3.4.4.7 Wavelet Clustering Summary

To conclude, k-means clustering was implemented with the Euclidean distance on each wavelet pair in order to cluster wavelets of the same length. This reduces the computational complexity associated with a large number of wavelets. The SSE and silhouette coefficient were used to find the optimal number of clusters for each filter length. Golden section search was implemented to limit the number of trials in order to find the optimal solution. The results of clustering can be seen in Table 3.1 below.

Table 3.1: Number of Wavelets before and after clustering

Filter Length	Before Clustering (Original Wavelets)	After Clustering (Representative Wavelets)
6	441	91
8	9,261	820

3.4.5 Wavelet Naming

After the designed wavelets have been clustered, a naming convention is necessary to keep track of wavelets of different length and version (either clustered or un-clustered). Typically, wavelets have a family name and a short name. The family name has been chosen as Gillis-Morsi, where the wavelet short name is ‘GM’ in accordance to the convention described in [15]. In order to denote the different lengths of filters, a “3” will be utilized for length-6, and a “4” will be utilized for length-8. This is common to work presented in literature with other wavelets that have been designed. For example, Daubechies wavelets (wavelet short name “db”) follow a similar convention. “db3” is Daubechies wavelet of length-6, while “db4” is Daubechies wavelet of length-8 [49]. Next, a number representing the clustering version is placed beside the length to determine whether the wavelet is part of the clustered or un-clustered set. A “0” represents the original

un-clustered set while a “1” denotes the clustered set. Lastly a number from 1 to the length of that set of wavelets is implemented as the third index. Each of these numbers will be separated by a period to make the name clear. An example of this naming convention is presented in Fig. 3.8. The “3” in the second block represents length-6, the “1” in the third block represents the clustered wavelet set and the “24” in the last block represents the 24th wavelet in that set. The complete name for this wavelet is written as GM3.1.24.

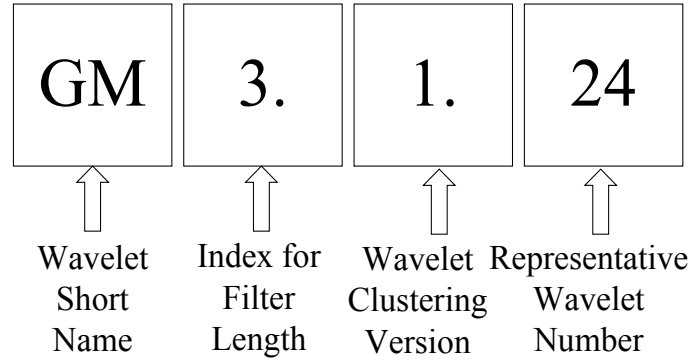


Fig. 3.8: Wavelet Naming Example [15]

3.5 Procrustes Analysis

In order to identify a specific wavelet among all of the representative wavelets in each filter length that best resembles the transient in the target signal (i.e., transient load pattern), a method which relies on shape comparison is necessary. Procrustes Analysis (PA) [50] is a shape comparison tool that can be utilized to identify the wavelet W that best resembles the transient of the load signal x_d . PA minimizes the difference between the comparison signal R and the transient of the load signal x_d by changing the transformation parameters α , ξ and β [50] as seen in Equation (3.27).

$$R = oW\xi + \beta$$

$$\min \left\{ \sum_{i=1}^N (X_{d,i} - R_i)^2 \right\} \quad (3.27)$$

The dissimilarity λ between the transient signal X_d and the comparison signal R is evaluated using the mean $\overline{X_d}$, which can be observed in Equation (3.28).

$$\lambda = \frac{\sum_{i=1}^N (X_{d,i} - R_i)^2}{\sum_{i=1}^N (X_{d,i} - \overline{X_d})^2} \quad (3.28)$$

The minimum dissimilarity between all wavelets and the transient signal X_d in the set is taken as the identified wavelet W^* .

3.6 Covariance Analysis

A second method to identify a specific wavelet among all of the representative wavelets in each filter length that best resembles the transient in the target signal is by utilizing Covariance Analysis (CA). In CA, the covariance of each representative wavelet sequence $W_R(l)$ and the sequence of the target signal $X_d(l)$ is computed, where the representative wavelet that provides the maximum covariance is considered the best matched wavelet $W_R^*(l)$ to the target signal $X_d(l)$.

$$\text{cov}(X_d, W_R) = \frac{1}{L-1} \sum_{l=1}^L (X_d(l) - \mu_{X_d})(W_R(l) - \mu_{W_R}(l)) \quad (3.29)$$

Where μ_{X_d} and μ_{W_R} are the mean of the sequence of the target signal X_d and the mean of the representative wavelet W_R , respectively. If the Covariance is normalized using the standard deviation of X_d and W_R , the values range from zero to one where zero indicates no linear relationship between the signals, while a covariance of one indicates a perfect linear relationship between the signals [47].

3.7 Summary

In this chapter, the STFT and WT were presented as different time-frequency analysis techniques utilized in feature extraction. It was observed that the FT does not preserve the time information making it a poor choice for this application. The STFT introduced a window function to the FT to preserve the time information, however the transform still used the sine and cosine functions and hence were found to be not suitable to analyze nonstationary signals. Moreover, both FT and STFT suffer from a fixed window size where there is usually a trade-off between time resolution and frequency resolution. The WT was presented as a method which not only overcame the issues associated with the FT and STFT but also allows for MRA by providing variable window sizes for different frequency ranges of a signal to be examined separately through the use of different decomposition levels. The wavelet choice with respect to the WT has also been discussed where wavelet design was introduced as a method to create a set of wavelets. In wavelet design, different properties of wavelets were examined in order to conduct the wavelet design process for length-6 and length-8 filters. By varying the parameters in the wavelet design equations from $-\pi$ to π , it was observed that a large number of wavelets (9,702), each with a unique shape, were designed. In total, 441 wavelets in length-6 and 9,260 wavelets in length-8 were designed. Due to this large number of wavelets, clustering was introduced as a method to decrease the number of wavelets down to 91 and 820, for length-6 and length-8, respectively. Next, wavelet selection methods were introduced to match each load transient to a wavelet utilizing Procrustes and covariance analysis. These two methods of selection offer two different ways to choose a wavelet for each load. In Procrustes analysis, the minimum dissimilarity was calculated while in covariance, the maximum covariance was

calculated. Both of these methods relied on comparing each wavelets shape to the load transient under study. The next chapter is dedicated to show the implementation of wavelet design and wavelet-signal matching in the NILM problem.

4. Machine Learning Classification for Non-Intrusive Load Monitoring

4.1 Introduction

The discrete wavelet transform with the wavelet design process including the results before and after clustering have been presented in Chapter 3. Procrustes and covariance analysis were presented in order to choose a specific wavelet based on the shape of the transient load signal. Machine learning classifiers will be introduced in this chapter, where both a lazy learner and an eager learner will be utilized in this work. Moreover, this chapter presents a complete NILM algorithm that uses wavelet design and machine learning classifiers. In addition, a system will be designed to test the effectiveness of the proposed method, by detailing the different components required to determine the performance of the NILM algorithm.

4.2 Signal Acquisition

In order to capture load transients, several parameters must be considered such as the sampling frequency and the waveform to be captured. The voltage signal may be used; however, the voltage waveform contains little information regarding the features in the transients. The voltage signal is prone to any abnormal change in the supply with very small changes to load switching. On the other hand, the current is more sensitive to changes in load operation compared to the voltage due to the fact that each load will draw a unique current. Therefore, the signatures of the current waveform will be investigated in this thesis. It was discussed in Chapter 3, that the sampling frequency directly affects the number of decomposition levels, which thus impacts the number of features to be extracted for identification. Each feature (decomposition level) can be described as a band pass filter

for a specific frequency range. Furthermore, Nyquist theorem states that for the chosen sampling rate, only half of that rate contributes to feature identification in order to ensure the sampled signal is alias-free (i.e., a signal that does not suffer from aliasing, or the effect that can cause several signals to become indistinguishable from each other). This effect is a significant problem in many signal processing applications, as usually aliasing removes the ability to separate the low frequency components of a signal [51]. Finally, in Chapter 2 it was observed that most time frequency analysis techniques apply a sampling frequency in the range of 1kHz to 2MHz where any higher frequency above 50kHz does not guarantee that economic hardware exists. Therefore, a sampling frequency greater than 1kHz and less than 50kHz is favored. Every decomposition level of the wavelet transform is related to frequency in Table 4.1 where the choice of the sampling frequency and the number of decomposition levels ensures that the frequency of the power system (60 Hz) is centered at the approximation level.

Table 4.1: Decomposition Levels for different Sampling Frequency

Decomposition Level	Frequency Band (Hz)	Nyquist Sampling Frequency
Approximation	0-120	240
Detail Level 1	120-240	480
Detail Level 2	240-480	960
Detail Level 3	480-960	1,920
Detail Level 4	960-1,920	3,840
Detail Level 5	1,920-3,840	7,680
Detail Level 6	3,840-7,680	15,360
Detail Level 7	7,680-15,360	30,720
Detail Level 8	15,360-30,720	61,440

It can be observed from Table 4.1 that at level 8, the sampling frequency required is higher than 50kHz and therefore level 8 is undesirable. The majority of power quality

analyzers evaluate loads at either 128 samples per cycle (7,680Hz) or 256 samples per cycle (15.36kHz) due to the reason the harmonic magnitude gets attenuated beyond these values. Therefore, a sampling frequency of 15.36kHz is selected, where 6 levels of decomposition exist in the wavelet transform.

In order to accurately capture the switching transient, all steady state information must be removed. The preferred method to remove steady state information is to perform a sequence (cycle) subtraction in which the previous cycle is subtracted from the next cycle. By performing this subtraction, only the transient information remains in the signal. An example of sequence subtraction can be examined in Fig. 4.1.

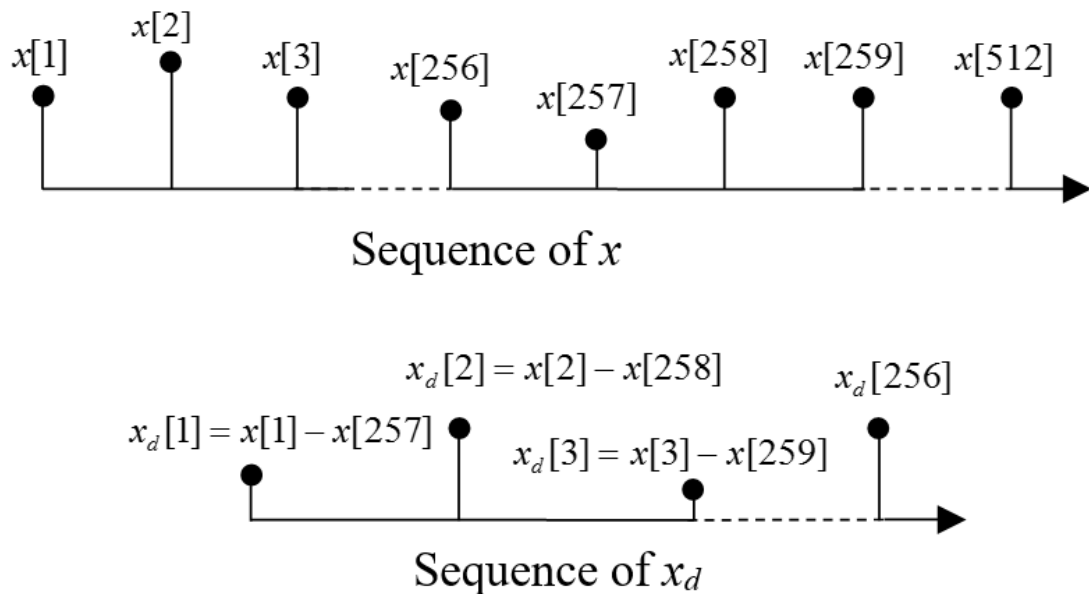


Fig. 4.1: Sequence Difference between Samples

4.3 Cycle Length

Following the choice of the sampling frequency as 15.36kHz, the next step is to identify the number of cycles of the signal which will be utilized for processing. In order to determine the proper number of cycles, a simple investigation into transients can be

conducted. In the smart grid research laboratory at UOIT, a Chroma 2kVA programmable AC source was set to create a perfect sine wave at 60Hz and 120 Volts. A Dranetz PX-5 power quality analyzer was used to sample the current waveform at 15.36kHz, where a 60 Watt ILB was switched on after two seconds of measurements. Fig. 4.2 represents the experimental setup utilized in capturing the electrical signals. A summary of specifications for each component can be found in the Appendix.

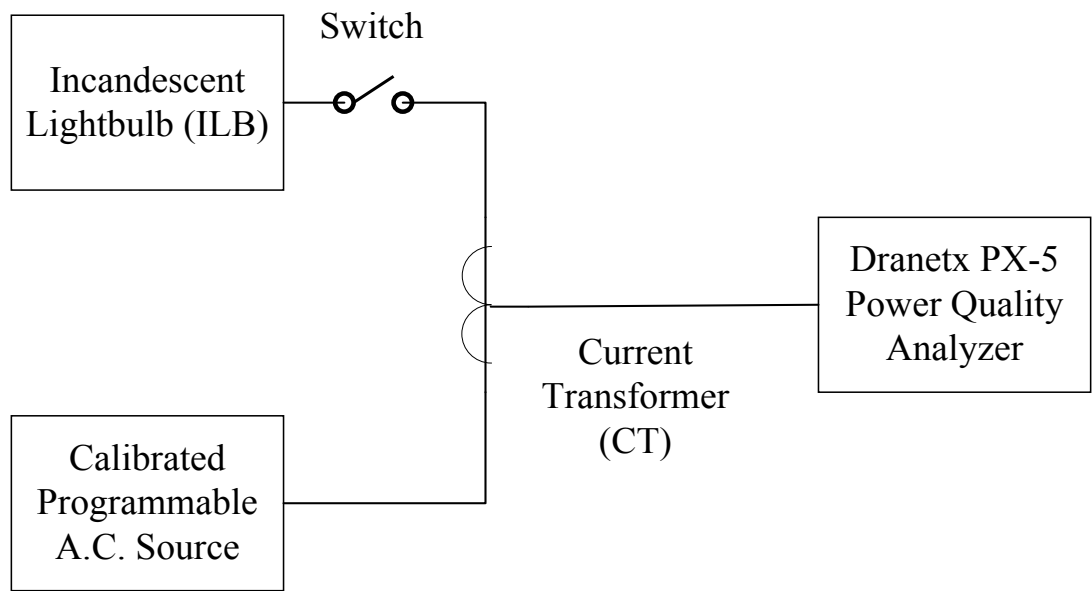


Fig. 4.2: Experimental Set-up for Analysis of Current Waveform

Visual inspection of Fig. 4.3 revealed that after nearly two cycles, the transient reaches steady state. In Fig. 4.4, the switching transient after applying the sequence subtraction described in section 4.2 is presented. This sequence subtraction removes all steady state information, leaving only the switching transient. Typically, when transients are analyzed, one complete cycle is studied from when switching occurs, which can be seen in Fig. 4.5. Therefore, one cycle is selected for this thesis.

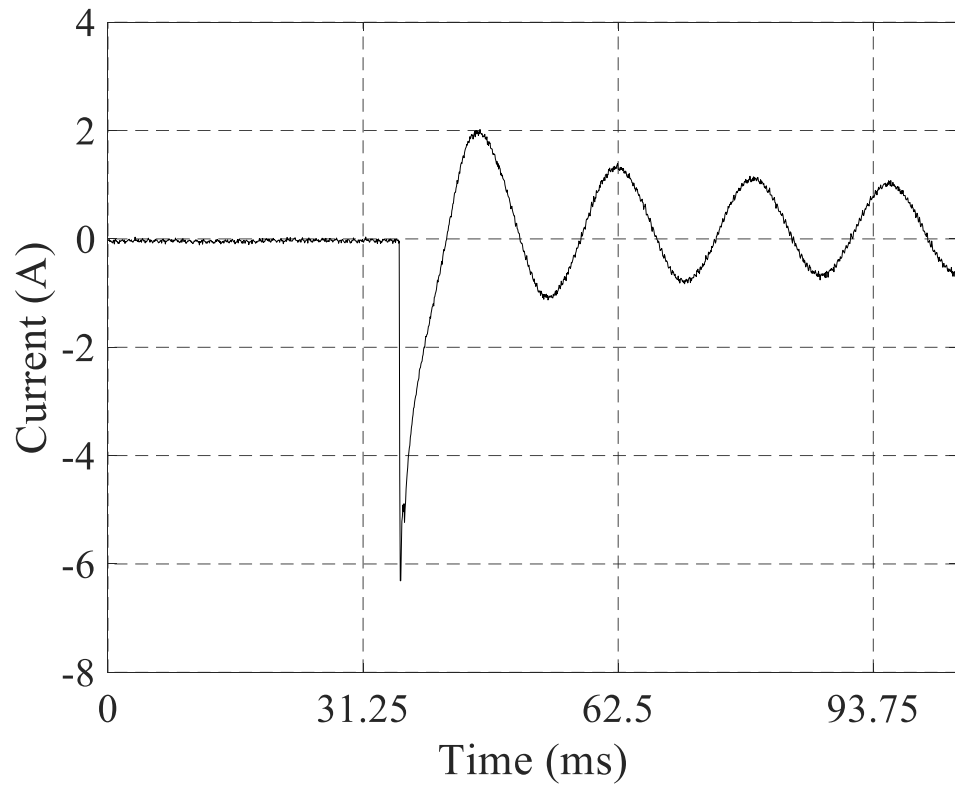


Fig. 4.3: Switching Transient in the case of ILB

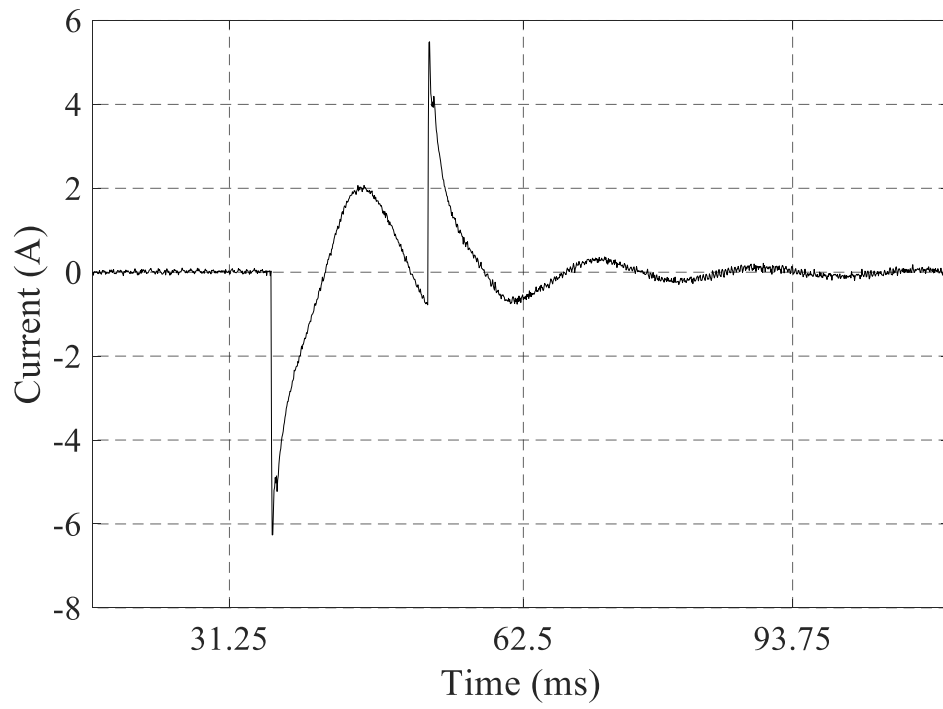


Fig. 4.4: Switching Transient of ILB after Sequence Subtraction

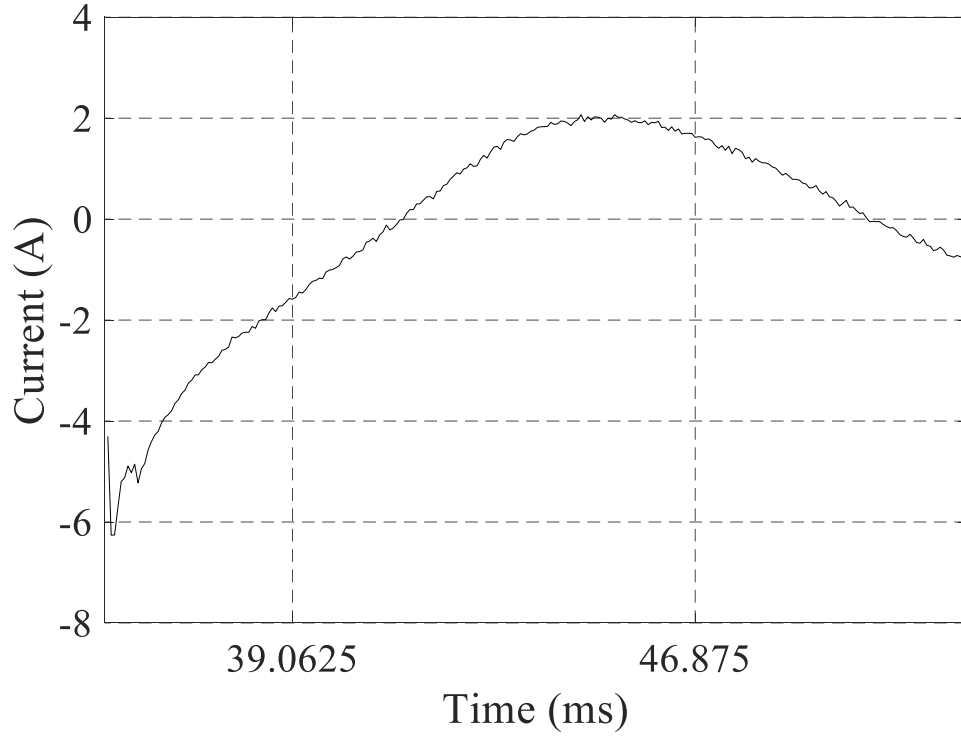


Fig. 4.5: Switching Transient of ILB after Sequence Subtraction for One Cycle

4.4 Smart Matching Module

Wavelet design was introduced in Chapter 3, followed by the computation of the sequence subtraction for the target signal $X_d(l)$, which was described in section 4.2. A systematic method to choose a representative wavelet based on the represented current waveform is required. The current waveform is matched to one of the representative wavelets, generated in the wavelet design section for the length-6 and length-8 filters. After applying the wavelet clustering procedures, Procrustes or covariance analysis can be utilized for each representative wavelet sequence $\psi_R(l)$ and the sequence of the target signal $X_d(l)$. The dissimilarity for Procrustes analysis in (3.28) or the covariance in covariance analysis in (3.29) is computed where the representative wavelet is selected based on the minimum dissimilarity or maximum covariance, which is considered the best

matched wavelet $\psi_R^*(l)$ to the target signal $X_d(l)$ for each respective method.

This process is repeated until each load signal (i.e., target signal) is matched to the best representative wavelet, from each list of wavelets in the wavelet library. This includes 91 wavelets from length-6 and 820 from length-8. This entire process can be achieved through the developed Smart Matching Module (SMM) which can be observed in Fig. 4.6.

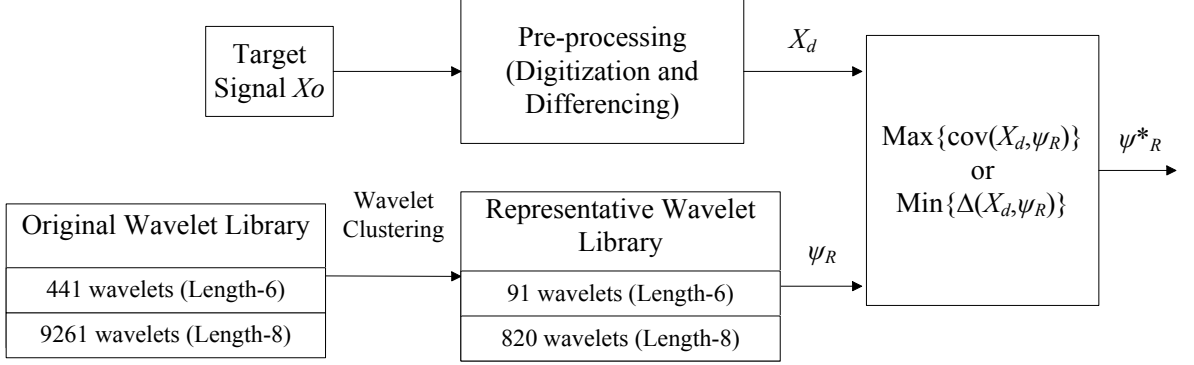


Fig. 4. 6: Depiction of wavelet smart matching module [15]

4.5 Machine Learning Classifiers

Pattern discovery in a dataset using machine learning [52]-[53] may be determined by employing a classifier such as Decision Tree (DT) [54] which attempts to learn the pattern from the hidden features (i.e., the energy of the wavelet coefficients in the NILM application). Other classification techniques such as support vector machine (SVM) and Bayesian classifiers might be utilized, however, since the aim of this work is to investigate the effectiveness of introducing the concept of wavelet design to NILM, it is desirable to use a deterministic classification technique. These techniques include DT or k-Nearest Neighbor (K-NN), instead of other non-deterministic techniques such as SVM which utilize optimization theory or Bayesian which relies on probability theory in the case of Bayesian classifiers. Typically, in machine learning, the dataset is divided into two subsets,

one dataset for training, while the other set is preserved for testing. In addition, the chosen machine learning method is trained with the information in the training set, followed by using this knowledge to classify the loads using the testing set. This classification process produces a classification accuracy which can be used to compare the effectiveness of this method against other approaches. In this study, two machine learning classifiers (DT as an eager learner and k-NN as a lazy learner) are employed to estimate class labels (i.e., loads) in a supervised way.

4.5.1 k-Nearest Neighbor

Unlike eager learners that perform load classification based on an induced classification model using all labeled data, lazy learners such as k-NN [55] delay the process of model development and perform classification by finding all training instances $(\tau, v) \in \theta$ that are relatively similar to the attribute test instance $\zeta = (\tau', v')$. The algorithm uses the Euclidean distance proximity measure δ to generate a list θ_ζ containing all nearest neighbours (NN) that are closest to the training instance ζ .

$$\delta(\tau, \tau') = \sqrt{\sum_{l=1}^L (\tau_l - \tau'_l)^2} \quad (4.1)$$

The class label of the test instance is predicted using a majority voting scheme which can be observed in Equation (4.2). This scheme relies on comparing the class label c with the class label v of the NN using the indicator function Ξ which returns one if true and zero otherwise:

$$v' = \arg \max_c \sum_{(\tau_l, v_l) \in \Theta_\zeta} \Xi(c = v') \quad (4.2)$$

The role of the indicator function Ξ is to compare the class label of the test record with the class labels of its neighbors. If the class label of the test record matches the class label of the record in the neighborhood, the indication function returns one, otherwise it returns zero. The class is predicted using the maximum argument (argmax) which indicates the correct class would be the class of the majority of records in the neighborhood.

4.5.2 Decision Tree

The load classification procedures in NILM applications using an eager learner such as DT usually requires all input data to be labeled. The input dataset is split into two subsets: one set \tilde{I} for training and another set \tilde{Y} for testing. The training dataset \tilde{I} is utilized in constructing the classification model through the DT induction algorithm [54]. The process of developing the classification tree works by determining the best split amid the features of the data set, while also calculating the impurities in the tree nodes using the Gini index, which can be seen in Equation (4.3) [54]:

$$Gini(\sigma) = 1 - \sum_{d=0}^{D-1} [f(c|\sigma)]^2 \quad (4.3)$$

Where D is the number of classes and $f(c|\sigma)$ is the fraction of records belonging to class d at a given node σ .

4.6 k-fold Cross Validation

This section presents the procedures involving the estimation of the classification accuracy using cross-validation. In k -fold cross-validation [56], the data is subdivided into

k equal-sized folds, where during each run one-fold is utilized for testing while the remaining k-1 folds are used for training. This process is repeated k times, where for each run the error ε_k is computed using the number of records incorrectly predicted $\bar{\gamma}$ and the total number of records Γ . The cross-validation error ε_{cv} can be computed by summing the errors for all k runs.

$$\varepsilon_k = \frac{\bar{\gamma}}{\Gamma} , \quad \varepsilon_{cv} = \sum_k \varepsilon_k \quad (4.4)$$

The accuracy $\bar{\omega}$, which is a measure of the performance of the classifier can be defined as:

$$\bar{\omega} = 1 - \varepsilon_{cv} \quad (4.5)$$

In order to test the confidence of the classifiers, it was necessary to assess their performances with different training and testing conditions. To facilitate this, k-fold cross validation can be used to split the dataset into two different partitions. An example for 2-fold cross validation is to split the dataset into two equal sets, where one is used for training while the other is used for testing. Therefore, the classifier uses half of the total dataset for training and is tested on the remaining half of the dataset. A 3-fold split takes 1/3 of the dataset as the testing set and the remaining 2/3 as the training set. The general formula is to take 1/K as the testing set and (K-1)/K as the training set [47]. In this work, 10-fold cross validation was implemented.

4.7 Monte Carlo

Monte Carlo was first introduced in 1949 as a numerical method to solve complex problems that involve stochastic parameters [57]. The training of DT (eager learner) and K-NN (lazy learner) along with cross-validation introduces uncertainty in the classification accuracy due to the stochastic nature of these algorithms. This work considers the Monte

Carlo method to probabilistically estimate the classification accuracy.

The training of both DT and K-NN classifiers and the cross-validation procedures to estimate the classification accuracy $\bar{\omega}$ are computed in each trial μ of Monte Carlo. After a large number of trials M , the mean classification accuracy $\bar{\omega}_{mean}$ is computed.

$$\bar{\omega}_{mean} = \sum_{\mu=1}^M \frac{\bar{\omega}^{(\mu)}}{M} \quad (4.6)$$

4.8 One Against the Rest

Typically, more than two loads need to be determined, inferring more than two classes exist. In order to apply binary classification (i.e., one class is positive and the other class is set to negative) [54] of more than two loads, the problem is decomposed into N binary classification problems (where N is the total number of loads). For each load (i.e., class), a binary problem is generated by considering all records that belong to one class as positive while the remaining records are considered negative [54]. For example, in the first binary classification problem, load 1 is considered the positive class while the remaining loads (2 to N) are considered the negative classes.

4.9 The Complete NILM Algorithm

The first step of this approach is to measure the current waveform, which can be performed using clamp on Current Transformers (CTs) placed around the main lines entering the home. Next, these CTs reduce the current magnitude so the A2D converter can sample the analog signal at the specified frequency of 15.36kHz, before sending the sampled data to a computer or microcontroller for processing. Fig. 4.8 presents a complete diagram of both the electrical and data signal flow. In the pre-processing stage, the first step is to compute the sequence subtraction followed by computing the wavelet transform

employing each specified wavelet chosen from the SMM. The energy of the coefficients is calculated using Equation (3.4) where 7 energy values (for each WT) are fed into a binary machine learning classifier that based on its training set, is able to identify the load class type. The classification accuracy can be computed for this algorithm using Equation (5.1).

By implementing the approach discussed in this thesis, several questions remain unanswered:

- 1- What is the best way to choose a representative wavelet for each load (Procrustes or covariance)?
- 2- What is the best machine learning classifier to apply to this work?
- 3- Does the choice of machine learning classifier depend on the wavelet choice?
- 4- Does this method work for many different load types such as: resistive loads, reactive loads, and switched mode power supplies?

In order to answer all of these questions, a test system is designed to evaluate the performance of the proposed approach. This test system will be used in both a simulation environment, followed by an experiment to verify the results of the simulation as detailed in the following chapter. This complete algorithm can be seen in Fig. 4.7.

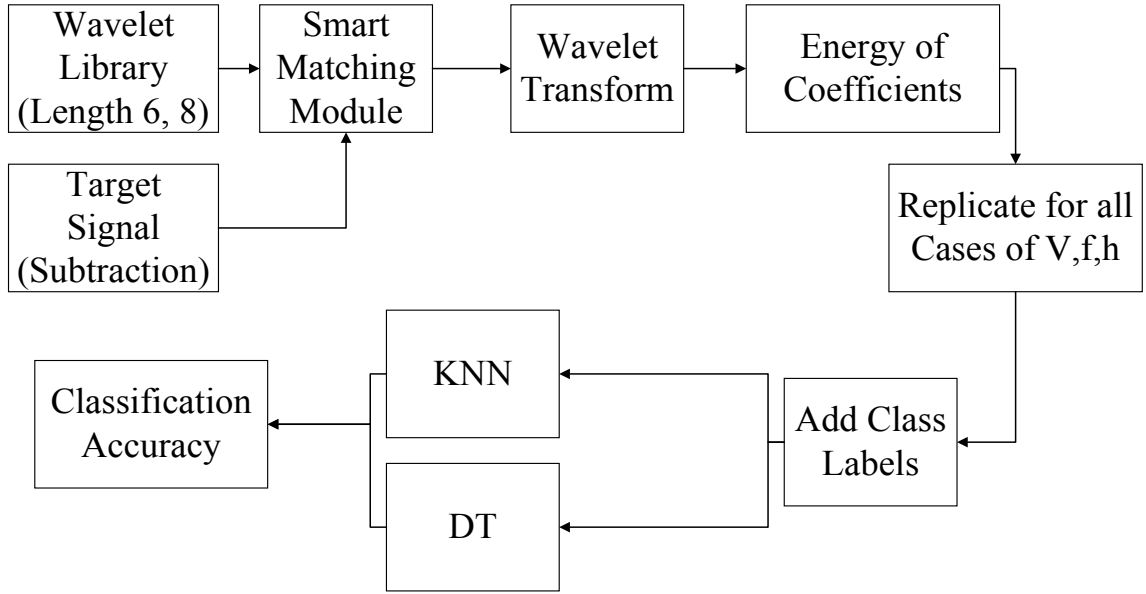


Fig. 4.7: The Complete NILM Algorithm

4.10 Summary

In this chapter, the complete NILM method is developed starting with looking at the process of capturing load transients. The sampling rate is chosen based off of the number of decomposition levels, followed by the sequence subtraction which is utilized to remove all steady state information. Furthermore, the number of cycles of the transient waveform needed to complete the analysis is determined, where one cycle starting at when switching occurs is chosen in this thesis. Next the SMM is introduced as a method to choose wavelets using Procrustes and covariance analysis. In addition, machine learning classification is introduced where K-NN and DT classifiers are described. The method of k-fold cross validation was presented as a means to assess the performance through computing the classification accuracy after applying Monte Carlo to address the stochastic nature of the cross validation. The OAR method is introduced by turning each load classification problem into a binary classification problem. Finally, the complete NILM algorithm is

developed in order to test the approach in both simulation and through an actual experiment as will be presented in Chapter 5.

5. Results and Evaluation

5.1 Introduction

This chapter will assess the performance of the proposed NILM algorithm developed in this thesis through simulation and experimentation. First, a test system is presented in which four loads can be utilized to evaluate the effectiveness of the algorithm. A dataset generated in a Power System Computer Aided Design (PSCAD) software is presented by building load models for the selected four loads. The voltage, frequency and harmonic distortion are varied to verify the algorithm is robust against power quality disturbances that occur on the electric power grid. In addition, these loads are switched on for each possible combination to create a series of different cases, where the current waveforms from each of these cases are sampled at 15.36kHz and saved into a set of comma separated values (CSV) files.

Next a simulation is constructed in MATLAB based on the results of the SMM. This simulation examines a series of different cases corresponding to different switching combinations of each load, which have been generated in PSCAD. Furthermore, the simulation calculates the energy of the wavelet coefficients for each case, followed by utilizing both K-NN and DT machine learning classifiers to calculate the classification accuracy. Finally, k-fold cross validation and Monte Carlo are applied to determine a proper classification accuracy over many trials based on the stochastic nature of the problem. The classification accuracies are examined using two different machine learning classifiers. These machine learning classifiers are DT and K-NN which represent an eager and lazy learner, respectively. Results are presented to illustrate the effectiveness of the approach through the simulation.

The performance of the wavelet-based approach is applied through an experimental endeavor. Initially, the experimental setup will be described to illustrate how the test system was constructed with lab equipment in order to match the same setup utilized in simulation. In addition, the hardware specifications for each component will be described, to make sure the hardware chosen correctly performs under transient and steady state conditions. Upon completion of the hardware testing, a dataset will be generated in a similar manner to the one created in the simulation. The only difference between this experimental set and the dataset generated in simulation is that in this set, switching occurs at a random point in the voltage waveform rather than always at the same point as was the case in simulation. Such phenomena is referred to as the zero crossing and will also be discussed in this chapter.

MATLAB is utilized to perform post processing on the dataset in a similar manner to that of the simulation. Both K-NN and DT machine learning classifiers will be implemented with k-fold cross validation and Monte Carlo to determine the actual classification accuracy. The classification accuracies will be examined for both machine learning classifiers, and the results are presented. These results will be compared to what was obtained in simulation validating both the experiment and simulation showing that this method outperforms the other NILM algorithms discussed in Chapter 2.

5.2 Test System Description

In order to address the questions raised in Chapter 4, a test system with the following hardware must be designed: different load types, constant AC source, CTs and A2D converter, switches, and a computer for processing.

There are many typical load types that exist in homes. Resistive loads like heaters, stoves or dryers are the primary load type, while others are inductive loads with motors, like air conditioners or washing machines. Each load in a home has a different power consumption pattern over a different period of time. In the literature review, it was discussed that large loads like an air conditioner, stove or dryer can be detected using time, frequency or time-frequency analysis techniques. However, smaller loads that exhibit a similar power consumption are difficult to detect and are usually switched on for large periods of a day. Some examples of these loads could be light bulbs, computers, cable boxes, audio receivers or cell phone chargers which are plugged in and remained turned on for an extended period of time. These low power / high energy loads are the loads of interest in this thesis. Loads representing resistive, non-resistive, and switched mode power supply loads need to be investigated for these small load examples. The simplest example of a resistive load is an ILB and therefore a 60 Watt ILB can be chosen. For a non-resistive load, a simple example that exists in almost all homes is a Compact Florescent Lightbulb (CFL), therefore a 13 Watt CFL is selected. The next load type is the switched mode power supply and there are several different types of loads that exhibit this behavior. A simple example that exists in most homes is a personal computer (PC), so a 100 Watt Laptop PC can be chosen. However, this 100 Watt PC cannot be used to represent all loads as some switched mode loads are much larger than 100 Watts while others are much smaller. Therefore, a general battery charger will be included in which the size of the battery can be modified to test the behavior of this load at higher and lower power levels. These 4 loads (ILB, CFL, PC, and Battery Charger) are representative of different small load types that exist in homes. A diagram of the complete test system can be observed in Fig. 5.1. In this

diagram, the four loads are on the left, followed by a switch for each load to turn the load on or off for testing. The calibrated AC power source provides power to the system, and a single CT is used to reduce the magnitude of current for measuring. The output of the CT is sampled using an A2D converter where these discrete samples are transferred into a computer for processing. This test system will be modeled in simulation and will be developed using real hardware.

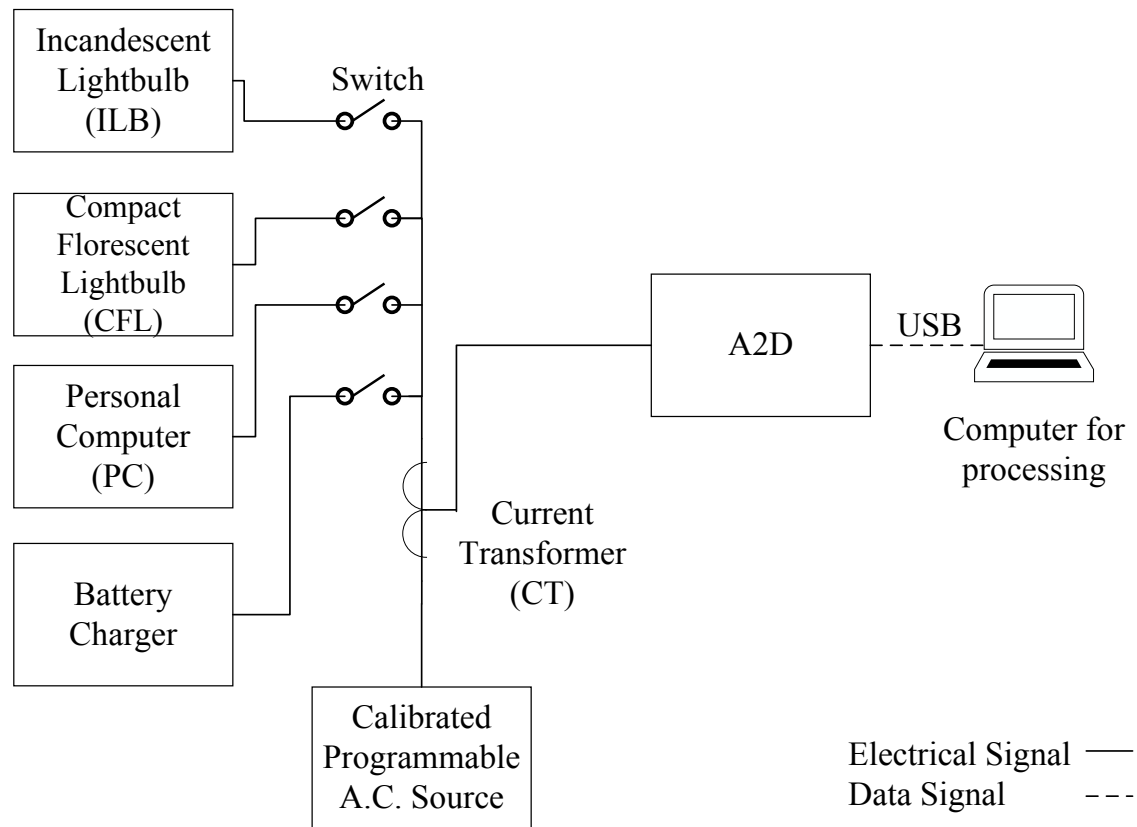


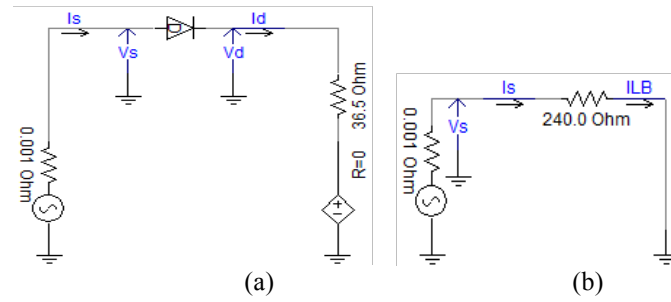
Fig. 5.1: NILM Test System

5.3 Dataset Generation

In order to test the proposed algorithm, a dataset is necessary for training and testing. Several commercially available datasets exist such as Pecan Street's Dataport [58], and Reference Energy Disaggregation Dataset (REDD) [59]. However, after examining all of

these datasets it was determined that none were suitable for testing this approach because most of these datasets only provide steady state information, while the ones that do provide transient information, only provide a short time interval which is not useful as an effective way to test the algorithm. With that in mind, it was necessary to create a dataset with load models of each load in order to switch them on and off. Moreover, this allows for variations in voltage, frequency, and harmonics, which give a more realistic analysis of the performance of the algorithm.

Each of the four load models were created in PSCAD [60]. The circuit models of each load developed in the PSCAD environment are depicted in Fig. 5.2. More information on how these load models were designed can be found in [61] and [62].



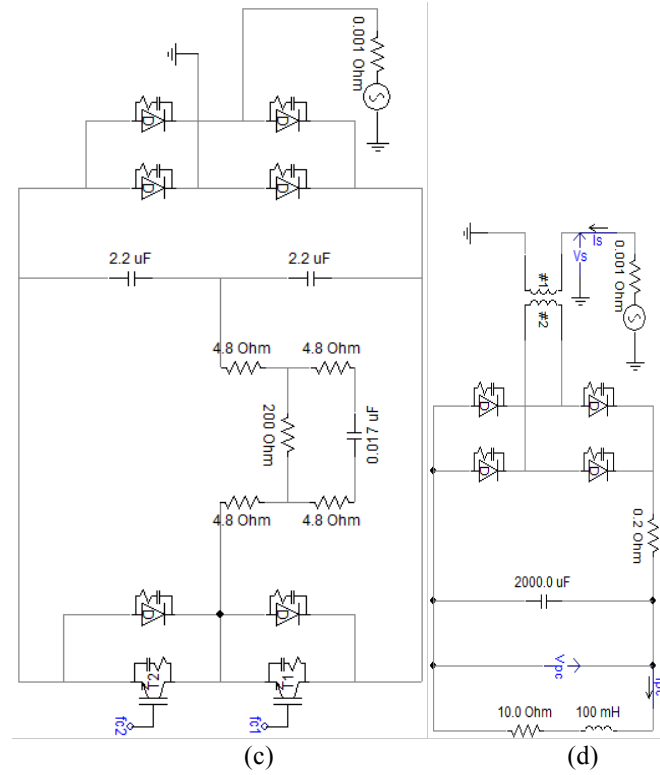


Fig. 5.2: PSCAD models of the loads used in the test system.

(a) Battery charger, (b) Incandescent light bulb, (c) Personal computer, (d) Compact fluorescent lamp









In PSCAD, the simulation time was fixed to ten seconds while the loads were switched on after three seconds. The current waveforms were sampled at 256 samples per cycle implying the time per cycle was 0.0167 s (i.e., 1/60 s). First, the process switches on each individual load after three seconds for all possible combinations of the remaining loads. For example, if the CFL is assumed as the load involved in switching, eight cases represent the remaining loads while taking all possible combinations as can be observed in Table 5.1, with the rest of the loads remaining in their previous states (on / off) while switching occurs. Considering there are four loads, 32 cases ($4 \text{ loads} \times 8 \text{ cases}$) are the result of all possible switching combinations. This procedure is repeated several times, each time applying one of three power quality disturbances. The voltage variation can be determined in accordance with the ANSI c84.1 [63] standard of voltage variation for Class


A, between 114 Volts and 126 Volts, while ANSI c37.106 [64] can be utilized for frequency variations, between 55 Hz to 65 Hz. The three types of power quality disturbances are therefore:

- 1- voltage magnitude variation ($\pm 5\%$ in 1% steps)
- 2- frequency variation (55 to 65 Hz in 1 Hz steps)
- 3- harmonic distortion by adding a fifth harmonic (1% to 5% in 1% steps)

These power quality disturbances bring the total number of cases to 864, which is the outcome of adding all of the possible cases: 11 voltage variation cases \times 32 switching cases = 352, and 11 frequency variation cases \times 32 switching cases = 352, and 5 harmonic distortion cases \times 32 switching cases = 160. The waveforms from each of these cases are saved in a CSV file for further processing in MATLAB.

Table 5.1: Switching Strategy to Generate Dataset

Case	Battery Charger	CFL*	PC	ILB
1	0		0	0
2	0		0	1
3	0		1	0
4	0		1	1
5	1		0	0
6	1		0	1
7	1		1	0
8	1		1	1

Where 1 represents a load switched on, 0 represents a load switched off, and  represents the load in question being switched.

5.4 Daubechies Wavelets

In order to accurately compare the results of Procrustes and covariance analysis, another wavelet must be chosen for comparison. In the majority of the literature presented

to date, Daubechies wavelets are utilized due to their strong abilities in feature extraction. Therefore, this thesis will use Daubechies wavelets as the base case of comparison. However, there are several different ways of observing Daubechies wavelets. In the MATLAB Wavelet toolbox, the true definition of Daubechies is implemented where the wavelet coefficients are calculated to a high degree of accuracy. This is not the best comparison against the wavelet design method due to the high resolution of Daubechies wavelets in the MATLAB toolbox, where the wavelet design process utilizes a lower resolution. Therefore, to more accurately compare the results of wavelet design, the wavelet representing Daubechies in the wavelet design set (at a lower resolution) to Daubechies can be used. These wavelets represent Daubechies to the resolution presented in this work. In the results section, the results from the lower resolution Daubechies wavelets are presented to give an accurate comparison against the wavelet design process utilizing the SMM. In Fig. 5.3 and Fig. 5.4, Daubechies wavelets in the MATLAB wavelet toolbox are compared against their equivalent in the wavelet design set. The names presenting the wavelets corresponding to Daubechies can be observed in Table 5.2.

Table 5.2: Daubechies Wavelets used in the Results

MATLAB Toolbox	Wavelet Design Set
db3	GM3.1.45
db4	GM4.1.399

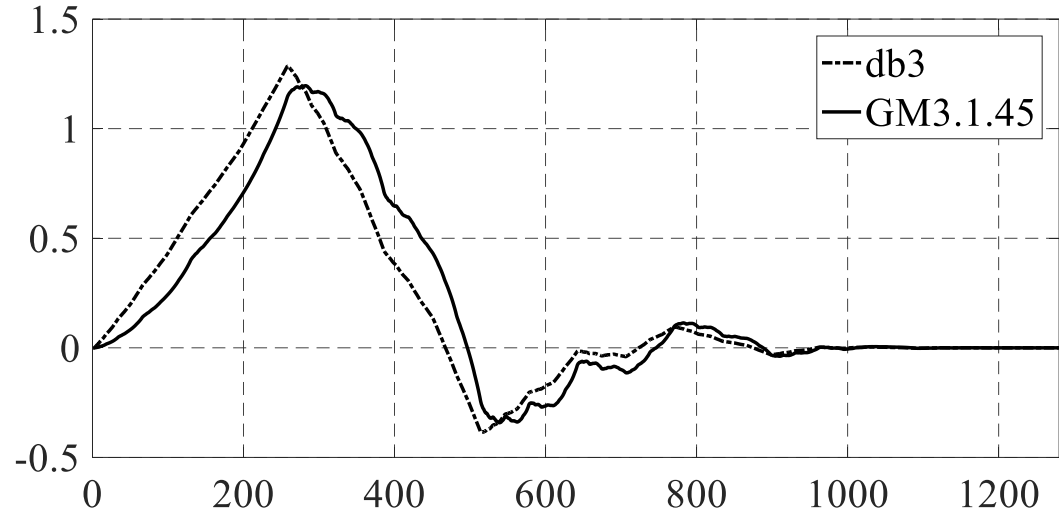


Fig. 5.3: db3 Against Wavelet Design Set

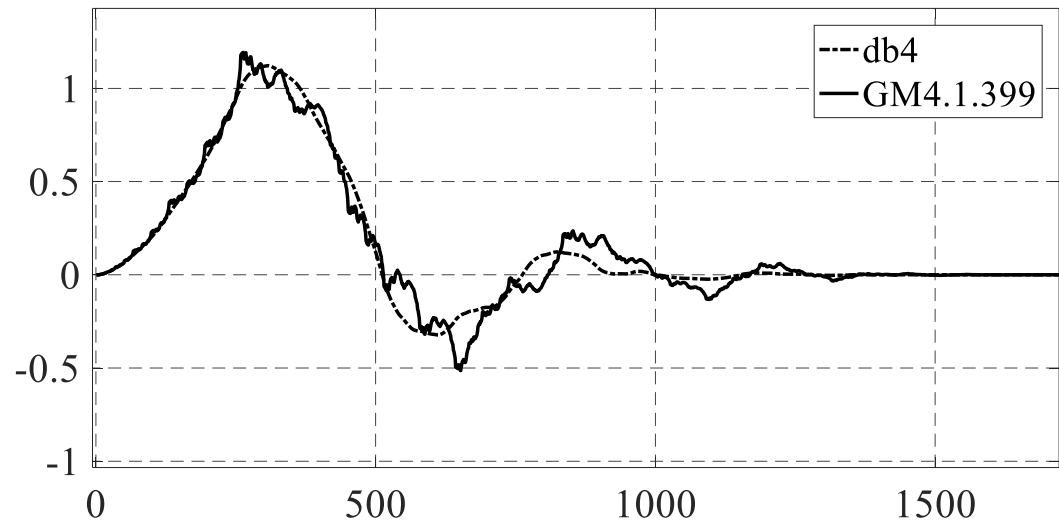


Fig. 5.4: db4 Against Wavelet Design Set

5.5 MATLAB Simulation

As a result of the dataset created in PSCAD, the algorithm can be implemented through simulation in MATLAB computational software with the Wavelet Toolbox [46]. Initially, the dataset is imported into MATLAB by translating each CSV file generated in PSCAD into the MATLAB workspace. Additionally, wavelet-signal matching is performed using the SMM where a wavelet is selected for each load. The selected wavelets

can be observed in Tables 5.3-5.6. The load signature recorded to perform the analysis is the waveform at 120 Volts and 60Hz with no harmonic distortion. This ensures the best wavelet is selected for the specific load, where no power quality disturbances modify the waveform. The SMM is applied for each length of wavelet filter (length-6 and length-8). Moreover, the measured currents are utilized to compute the sequence subtraction in order to find the change in current and hence the signal X_d . The WT is applied to compute the energy of the coefficients for the six decomposition levels, where the choice of six wavelet decomposition levels ensures that the power system frequency (i.e., 60 Hz) is centered at the approximation which is over the interval of 0 to 120 Hz. This dataset consists of 864 records \times 7 features (one approximation and six details) which represent the dataset to be employed for the classification using DT and K-NN. Class labels corresponding to each load are added by creating eight points in the dataset. The dataset including class labels are transferred into machine learning classifiers that perform classification. Supervised classification was performed, in which each classifier works independently to classify the load operation. Classification accuracies were computed and were compared against the GM wavelet most closely matching Daubechies wavelets of the same length. The complete steps of the simulation algorithm are depicted in Fig. 5.5.

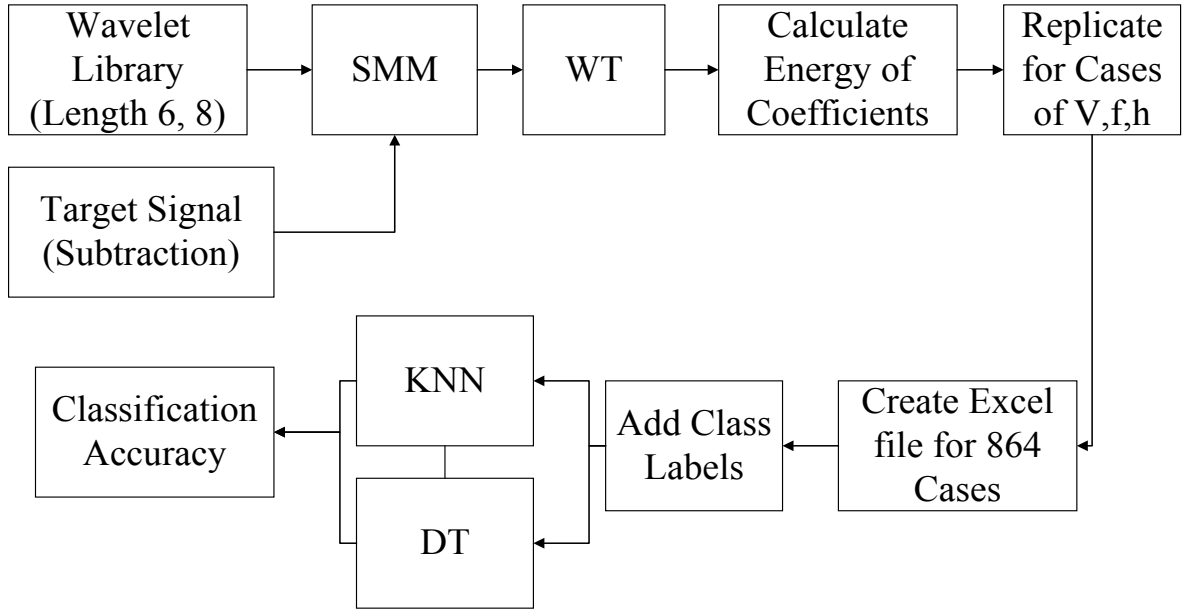


Fig. 5.5: MATLAB Simulation Algorithm

The total number of Monte Carlo trials is determined to be 1,000 as it has been seen that any increase does not provide much change in the results. Fig. 5.4 below depicts the classification accuracy for each load over 1,000 trials. The reader may observe that the accuracy does not vary after about 300 trials.

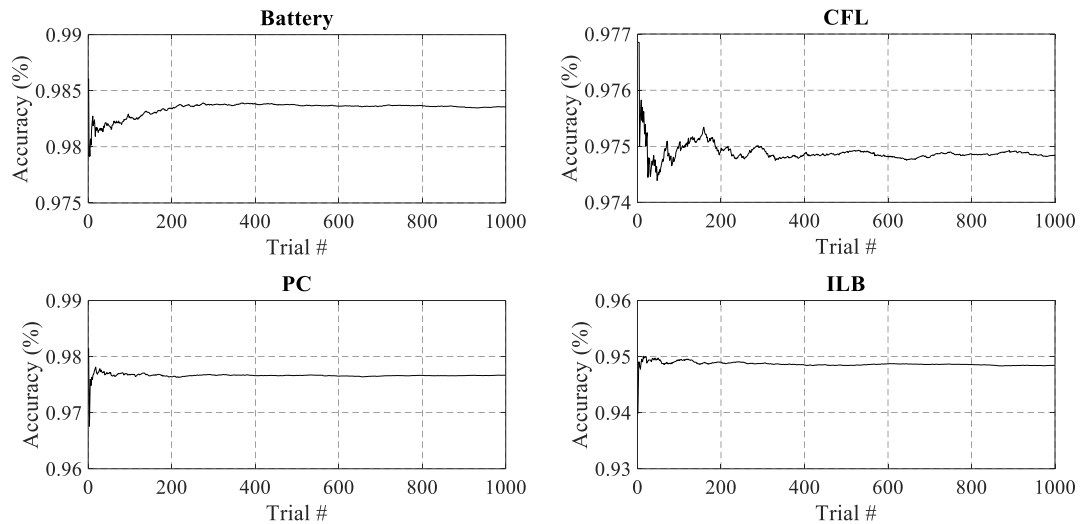


Fig. 5.6: Trends of 1,000 Monte Carlo Trials

5.6 Simulation Results

In this section, the results are presented for supervised learning, where K-NN and DT will be presented separately, to allow for a comparison in machine learning techniques. In addition, the results of length-6 and length-8 filters are compared, where Daubechies wavelet results are set side to side with the results of Procrustes and covariance analysis utilizing the SMM. In each of these tables, the classification accuracy is calculated using Equation (5.1). This classification accuracy represents the true positive classification accuracy as the purpose of this work is to show the effectiveness of the approach. If any other classification accuracy was presented, the results would be skewed due to the method in which the OAR technique handles negative classification.

$$ACC\% = \frac{TP}{P_{tot}} \quad (5.1)$$

Where TP is the number of true positive classifications, and P_{tot} is the total number of positives cases.

5.6.1 Length-6

Table 5.3: Simulation- K-NN for 1 Cycle of Length-6

Load	Daubechies Wavelet	Accuracy (%)	Procrustes SMM	Accuracy (%)	Covariance SMM	Accuracy (%)
Battery	GM3.1.45	96.63	GM3.1.53	98.04	GM3.1.53	96.64
CFL	GM3.1.45	86.33	GM3.1.58	86.10	GM3.1.58	86.33
PC	GM3.1.45	96.36	GM3.1.45	96.39	GM3.1.45	96.37
ILB	GM3.1.45	89.18	GM3.1.53	86.98	GM3.1.58	86.08
Total		92.12		91.88		91.35

Table 5.4: Simulation- DT for 1 Cycle of Length-6

Load	Daubechies Wavelet	Accuracy (%)	Procrustes SMM	Accuracy (%)	Covariance SMM	Accuracy (%)
Battery	GM3.1.45	97.81	GM3.1.53	98.19	GM3.1.53	97.76
CFL	GM3.1.45	94.10	GM3.1.58	94.90	GM3.1.58	94.03
PC	GM3.1.45	98.27	GM3.1.45	98.23	GM3.1.45	98.26
ILB	GM3.1.45	93.18	GM3.1.53	92.74	GM3.1.58	95.87
Total		95.84		96.01		96.48

5.6.2 Length-8

Table 5.5: Simulation- K-NN for 1 Cycle of Length-8

Load	Daubechies Wavelet	Accuracy (%)	Procrustes SMM	Accuracy (%)	Covariance SMM	Accuracy (%)
Battery	GM4.1.399	99.03	GM4.1.268	97.96	GM4.1.268	97.95
CFL	GM4.1.399	87.49	GM4.1.531	84.55	GM4.1.531	84.52
PC	GM4.1.399	95.48	GM4.1.471	95.51	GM4.1.471	95.50
ILB	GM4.1.399	90.25	GM4.1.543	90.80	GM4.1.653	90.71
Total		93.06		92.21		92.17

Table 5.6: Simulation- DT for 1 Cycle of Length-8

Load	Daubechies Wavelet	Accuracy (%)	Procrustes SMM	Accuracy (%)	Covariance SMM	Accuracy (%)
Battery	GM4.1.399	98.58	GM4.1.268	96.97	GM4.1.268	96.98
CFL	GM4.1.399	96.32	GM4.1.531	98.57	GM4.1.531	98.67
PC	GM4.1.399	96.88	GM4.1.471	96.95	GM4.1.471	96.93
ILB	GM4.1.399	94.38	GM4.1.543	98.70	GM4.1.653	98.71
Total		96.54		97.80		97.82

5.7 Discussion

5.7.1 Length-6

In the results listed in Tables 5.3 and 5.4, several different trends and conclusions can be observed. First, in length-6, the results presented demonstrate Daubechies wavelets

receive an accuracy of 92% and 95.8% for K-NN and DT, respectively. The accuracy for K-NN is about 0.3-0.75% higher than both of the wavelets selected using the SMM in both Procrustes and covariance. However, in the case of DT, the wavelets selected in the SMM outperform Daubechies case by 0.15-0.75%. This is due the fact that in the case of the length-6 filters, there are only 91 wavelets in the set to choose from. This results in significant duplication in wavelet choice (3 of the 4 loads) for both Procrustes and covariance utilizing the SMM. This duplication of wavelets does not allow for a different set of features to be extracted for each load. Investigating further, the CFL and ILB misclassify more than the battery charger and PC, by approximately 10% for K-NN and 3% for DT, which shows some wavelets are matched favorably while others are not. Overall, the most important observation in the case of the length-6 filter is that for K-NN, Daubechies wavelet outperforms the SMM, while in DT the SMM outperforms Daubechies wavelet.

5.7.2 Length-8

In length-8 a pattern starts to emerge, in which the SMM utilizing Procrustes and covariance produce a better result (1.3% higher) than Daubechies case in DT, while the opposite is true for K-NN (0.8% lower). This is an important observation considering it demonstrates that the selection in wavelet as well as the choice of machine learning method has a direct effect on the classification accuracy. Moreover, in the length-8 filter, each load has a specific wavelet which supports a reason for improving the number of correct classifications. In length-8 there are 820 wavelets to choose from when compared to 91 in length-6, which allowed both Procrustes and covariance to select superior wavelets for feature extraction in this application. Finally, for the length-8 filter it is observed that

covariance produces a similar result to Procrustes as the wavelets selected are identical for all loads except the ILB, which implies the result that covariance and Procrustes analysis are almost identical in choosing wavelets for NILM.

5.8 Simulation Discussion

The goal of this chapter is to assess the performance of the SMM utilizing Procrustes and covariance analysis in identifying load operation when compared to Daubechies case presented in the literature. The differences in K-NN and DT utilizing supervised learning will be discussed for varying filter lengths. Specifically, Procrustes and covariance with the SMM produce similar results, making it difficult to select one method as superior to the other since there are some cases where covariance outperforms Procrustes (e.g. in the case of length-8 with DT, 96.48% for covariance and 96.01% for Procrustes), while in other cases the opposite is true (e.g. in the case of length-6 with K-NN, 91.88% for Procrustes and 91.35% for covariance). It is important to note that Procrustes and covariance select the same wavelet for every load except the ILB. Finally, by observing the different machine learning classifiers, DT outperforms K-NN by an average of 3% in part because DT is an eager learner that builds a large tree for each load, where K-NN is a lazy learner that utilizes the Euclidean distance to determine load operation.

5.9 Experimental Setup

To verify the results of the previous simulations, an experimental set up was developed to test the proposed algorithm where the same four-load system has been developed with hardware to test the algorithm. Fig. 5.1 shows a depiction of the set-up where each load is plugged in at a single point, the power is provided from a constant AC source, and the current at a single point is measured using a current transformer. The output

of the current transformer is a voltage across a resistor which is measured by a data acquisition module (DAQ) represented by an A2D converter. In accordance with the algorithm, the sampling rate is set to 256 samples per cycle or 15.36kHz. Moreover, the DAQ transfers the data into a computer via USB where a C-language program performs the algorithm, which allows for classification to be computed in real time, as what would be expected in a home. A schematic and a picture of the actual experimental setup can be observed in Fig. 5.1 and Fig. 5.7, respectively.

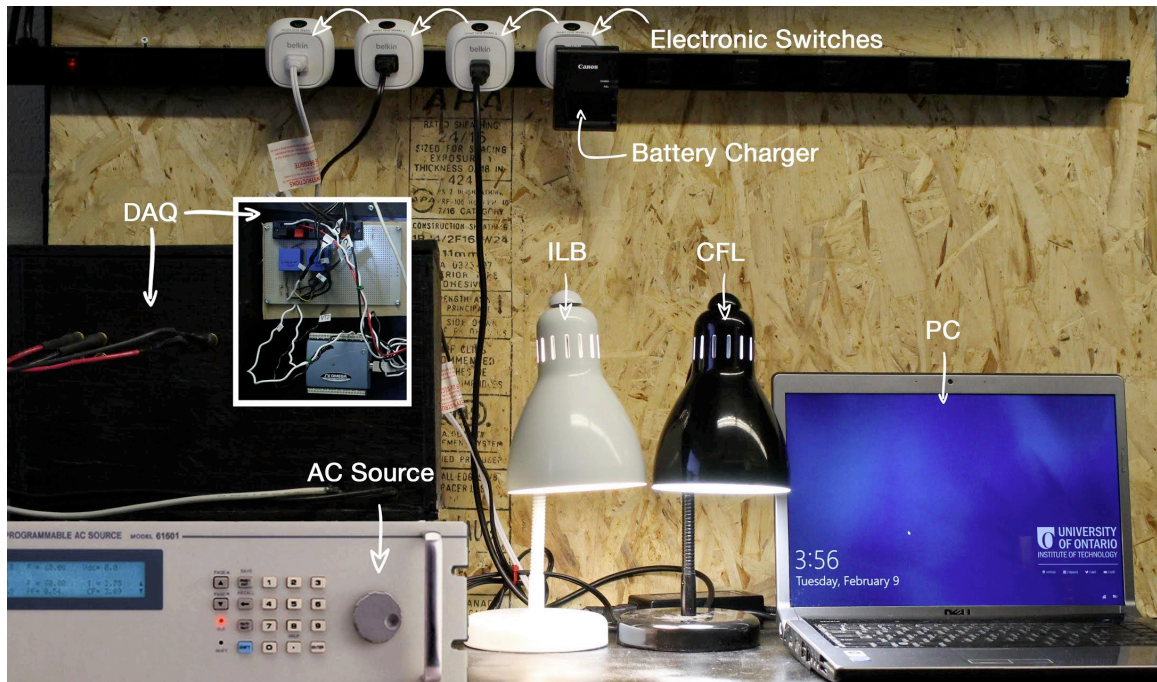


Fig. 5.7: Actual Experimental Setup

5.9.1 Hardware Specifications

This section will thoroughly describe each hardware component utilized in the experiment, where specifications for each component have been summarized in the Appendix. Data presented in this section was recorded using the Dranetz PX5 power quality analyzer. The PX5 is a commercial grade power quality analyzer capable of

sampling four different currents and three different voltages simultaneously at 15.36kHz. It is able to capture steady state and transient events as well as harmonics, inter-harmonics and other power quality indices. More specifications on the PX5 are summarized in the Appendix.

5.9.1.1 ILB

In simulation, a 60 Watt ILB was modelled in PSCAD as a pure resistive load. In order to verify the simulation, a Philips Duramax 60 Watt lightbulb was selected [65]. This lightbulb operates at 120 Volts and in steady state conditions draws 0.5 Amps of current, with a rated light output of 830 lumens. A summary of the specification sheet for the ILB is listed in the Appendix. Figures 5.8-5.11 present the captured electric signals in both transient and steady state.

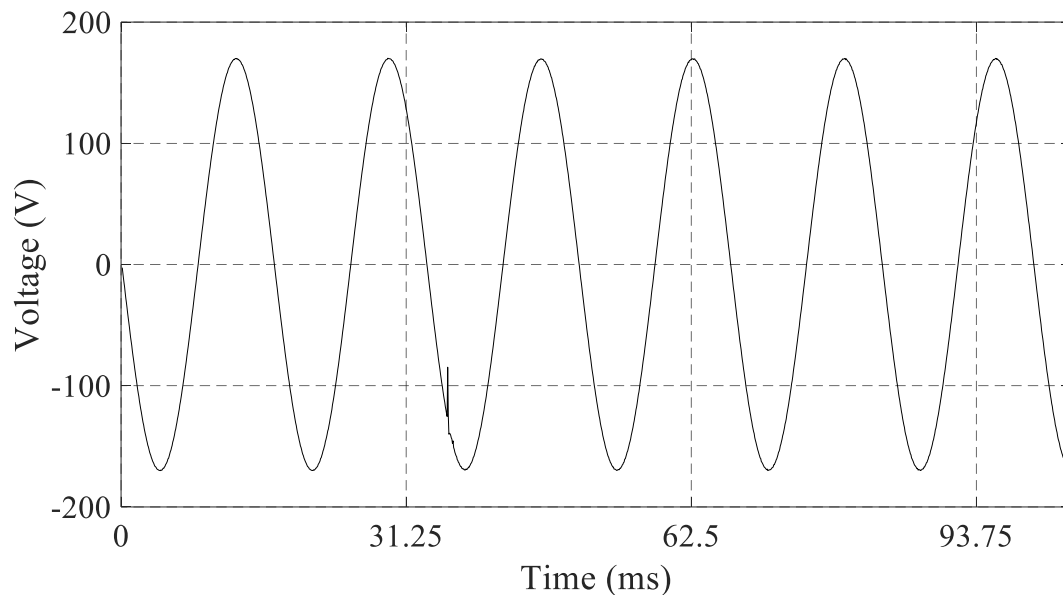


Fig. 5.8: Transient Voltage Signal in case of ILB

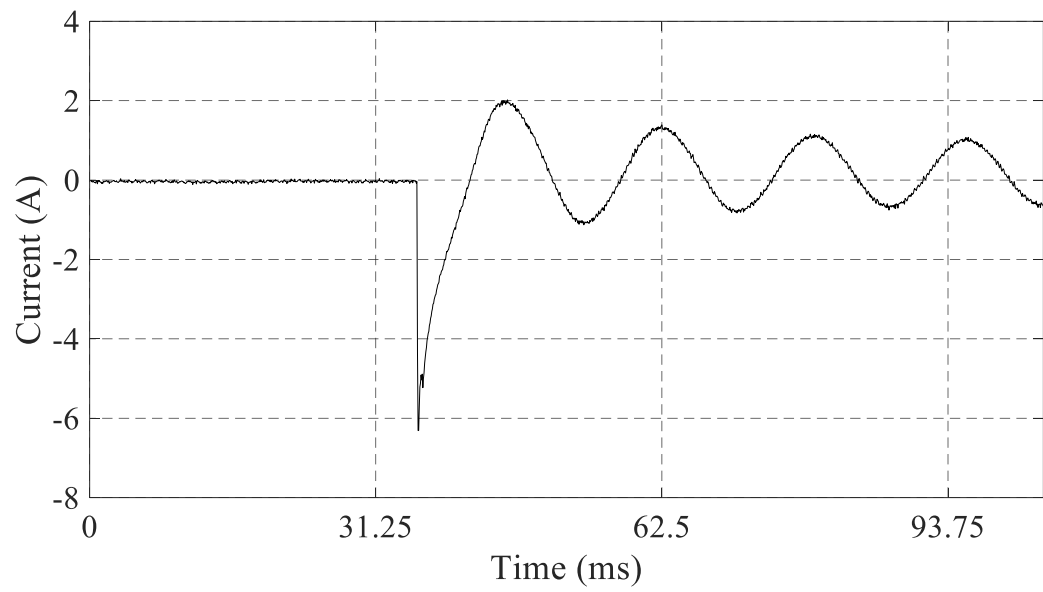


Fig. 5.9: Transient Current Signal in case of ILB

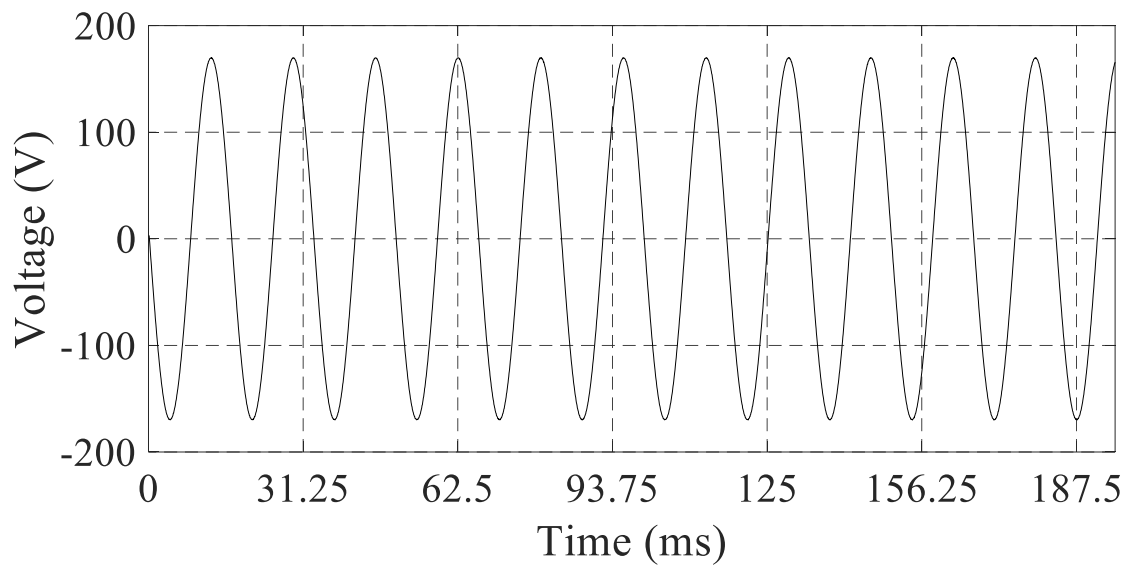


Fig. 5.10: Steady State Voltage Signal in case of ILB

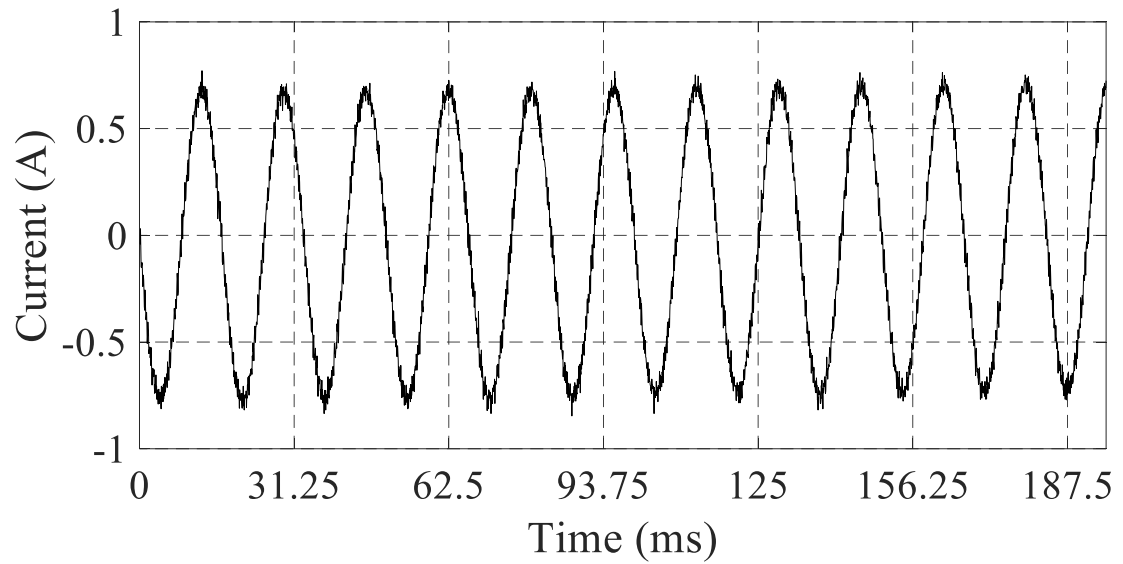


Fig. 5.11: Steady State Current Signal in case of ILB

5.9.1.2 CFL

In simulation, a 13 Watt CFL was modelled in PSCAD. In order to verify the simulation, a Globe 13 Watt lightbulb was selected and operates at 120 Volts where under steady state conditions draws 0.21 Amps of current with a rated light output of 900 lumens. Figures 5.12-5.15 demonstrate the gathered electric signals in both transient and steady state for the CFL.

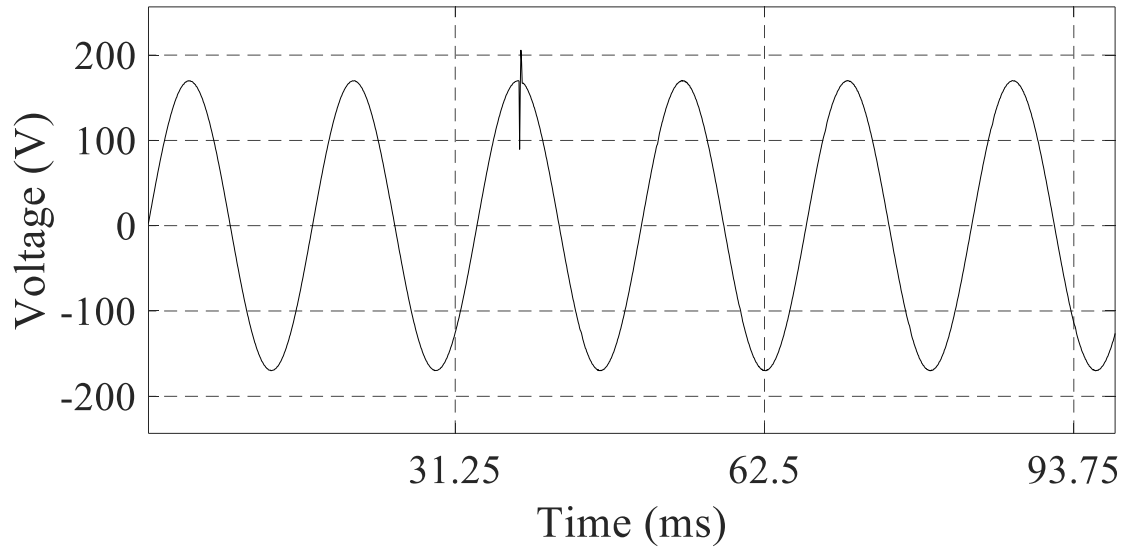


Fig. 5.12: Transient Voltage Signal in case of CFL

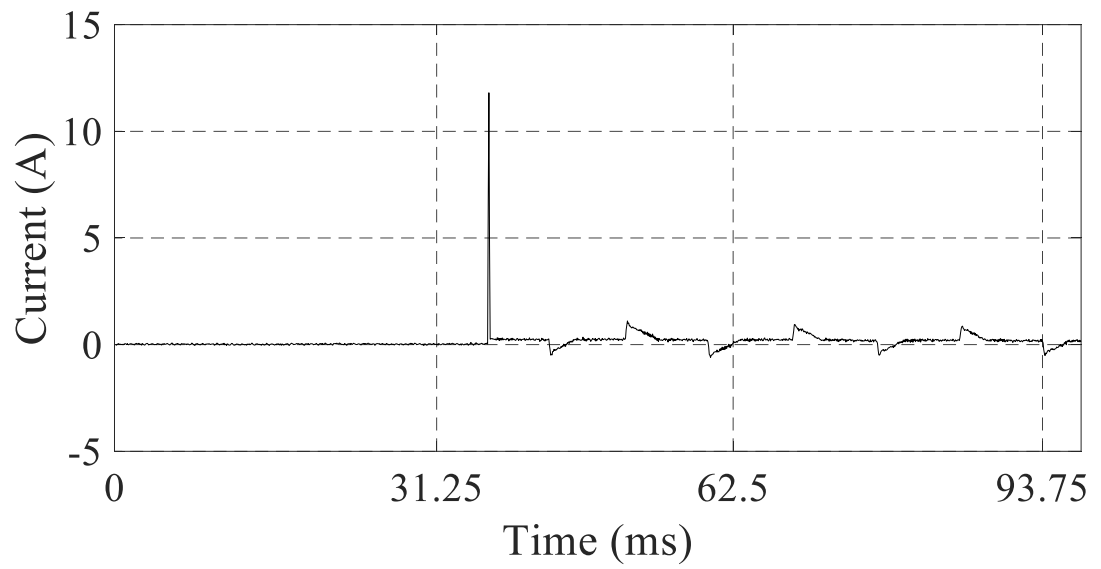


Fig. 5.13: Transient Current Signal in case of CFL

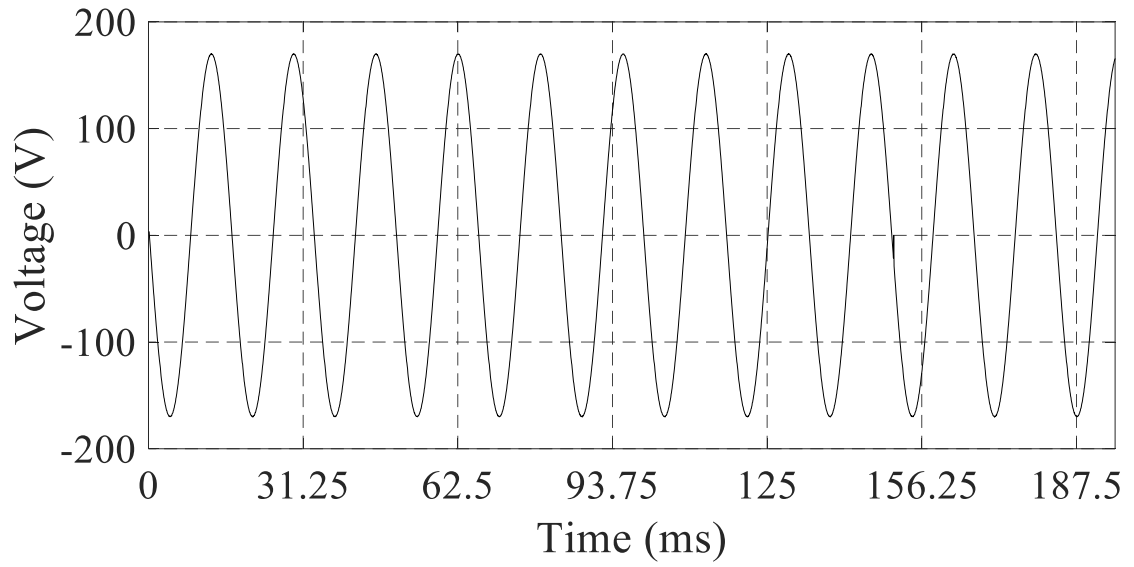


Fig. 5.14: Steady State Voltage Signal in case of CFL

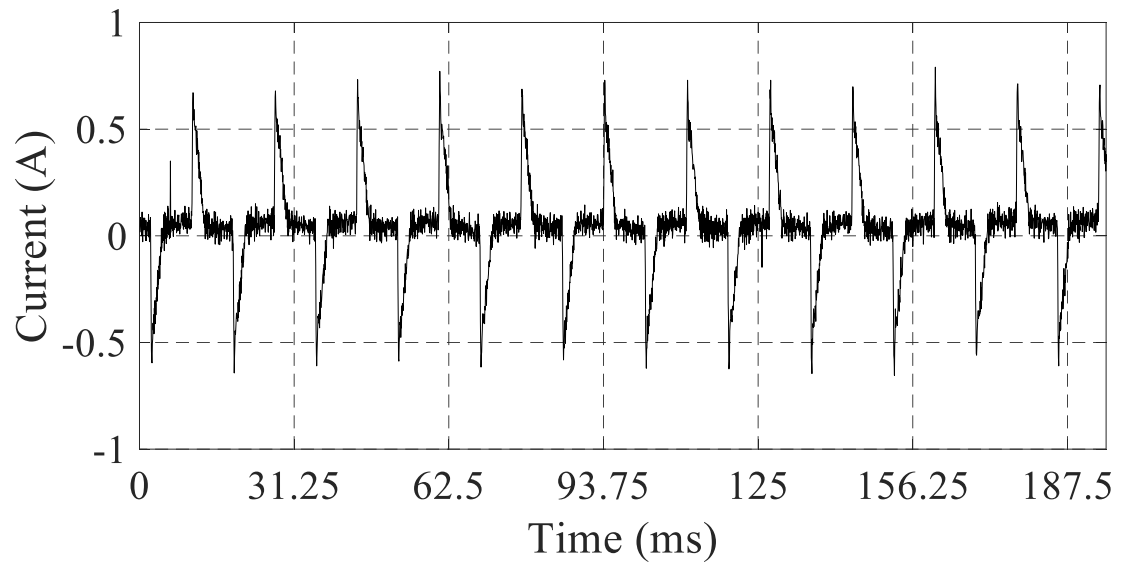


Fig. 5.15: Steady State Current Signal in case of CFL

5.9.1.3 PC

In simulation, a 100 Watt laptop was modelled in PSCAD. In order to verify the simulation, a Dell Studio 1535 was selected [66] and operates at 120 Volts in steady state conditions while drawing 1 Amp of current. A summary of the specification sheet can be

observed in the Appendix. Figures 5.16-5.19 demonstrate the gathered electric signals in both transient and steady state that have been collected for the PC.

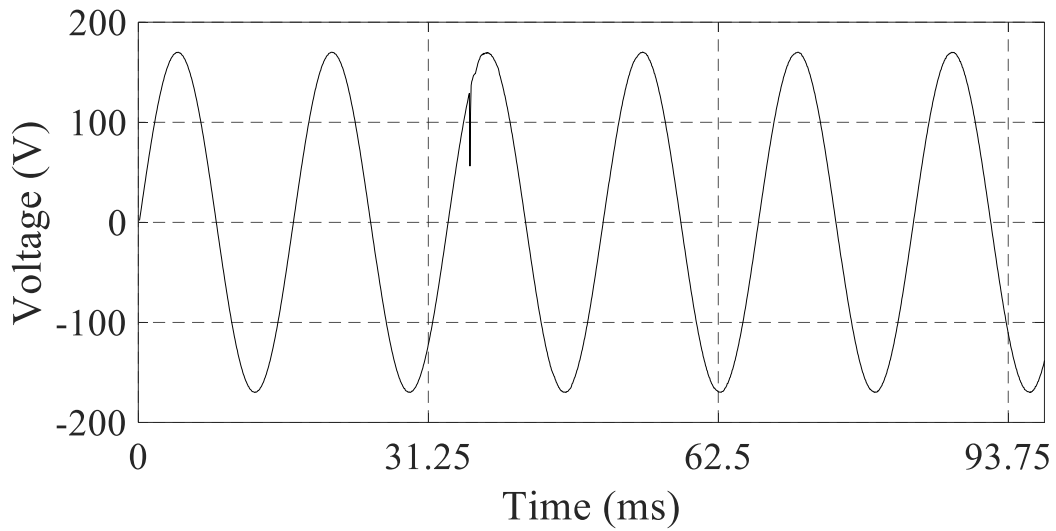


Fig. 5.16: Transient Voltage Signal in case of PC

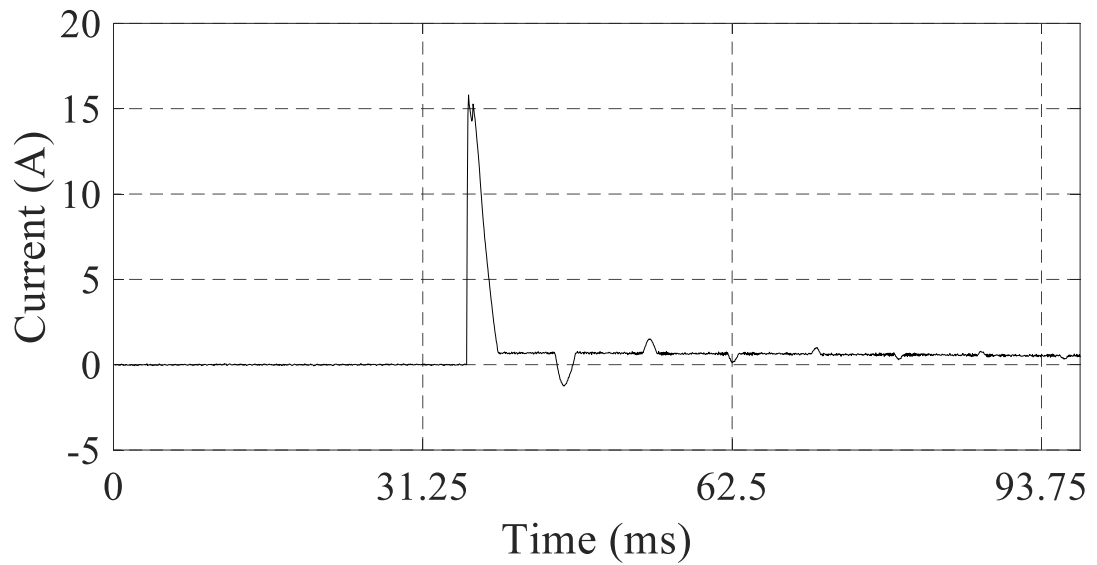


Fig. 5.17: Transient Current Signal in case of PC

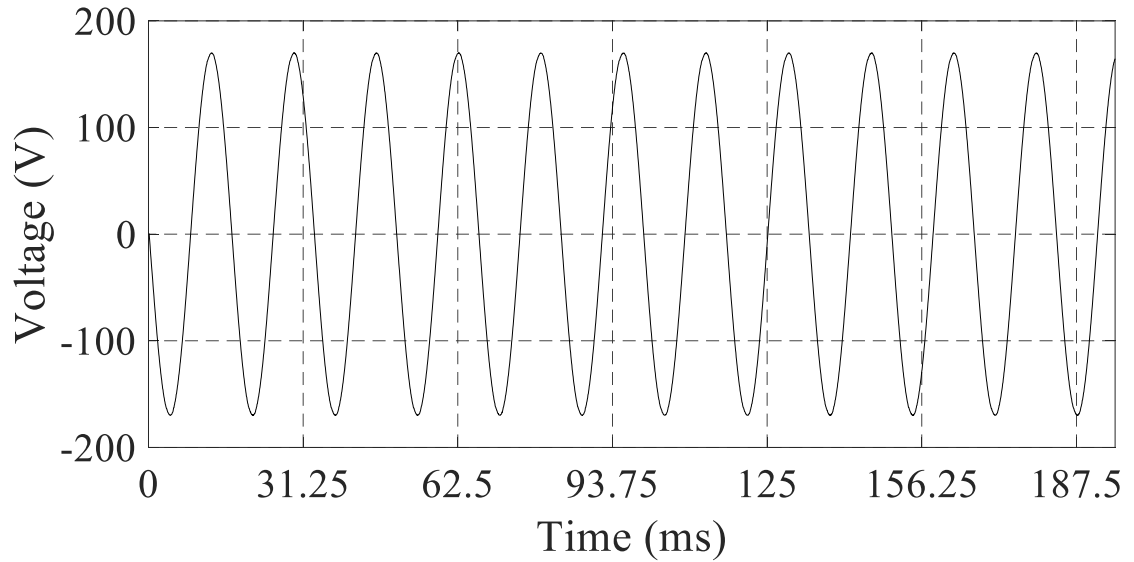


Fig. 5.18: Steady State Voltage Signal in case of PC

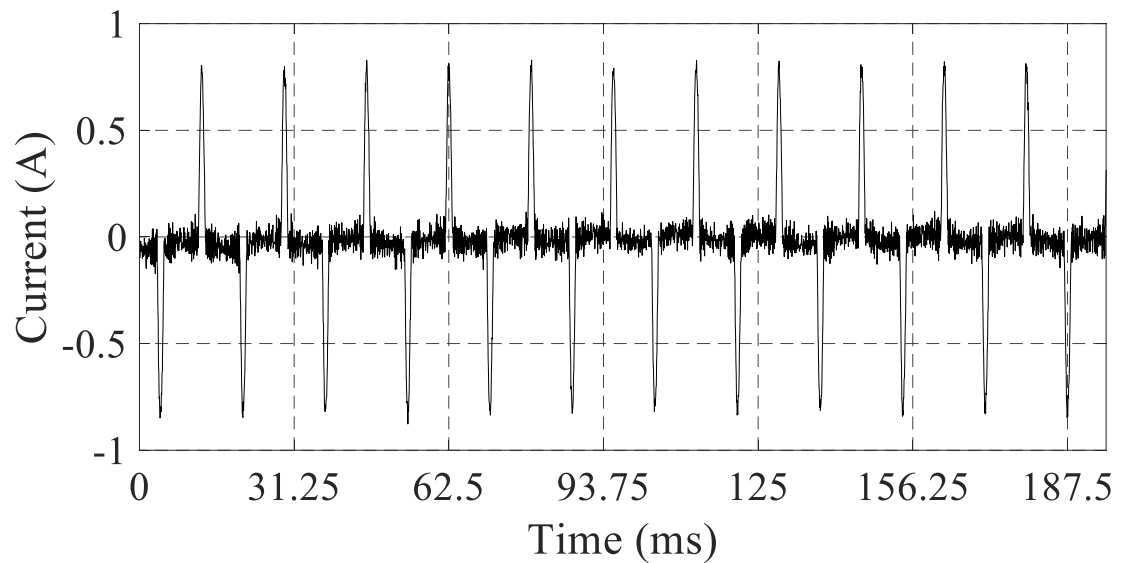


Fig. 5.19: Steady State Current Signal in case of PC

5.9.1.4 Battery Charger

In simulation, a 12 Volt battery charger was modelled in PSCAD. In order to verify the simulation a Canon Camera Battery Charger was selected as it generates a similar transient to a larger battery, but does not draw as high of a current. This battery charger operates at 120 Volts and in steady state conditions draws 0.12 Amps of current. The

battery chosen is a 6.36 Watt hour Lithium Ion battery. Figures 5.20-5.23 demonstrate the captured electric signals in both transient and steady state for the battery charger.

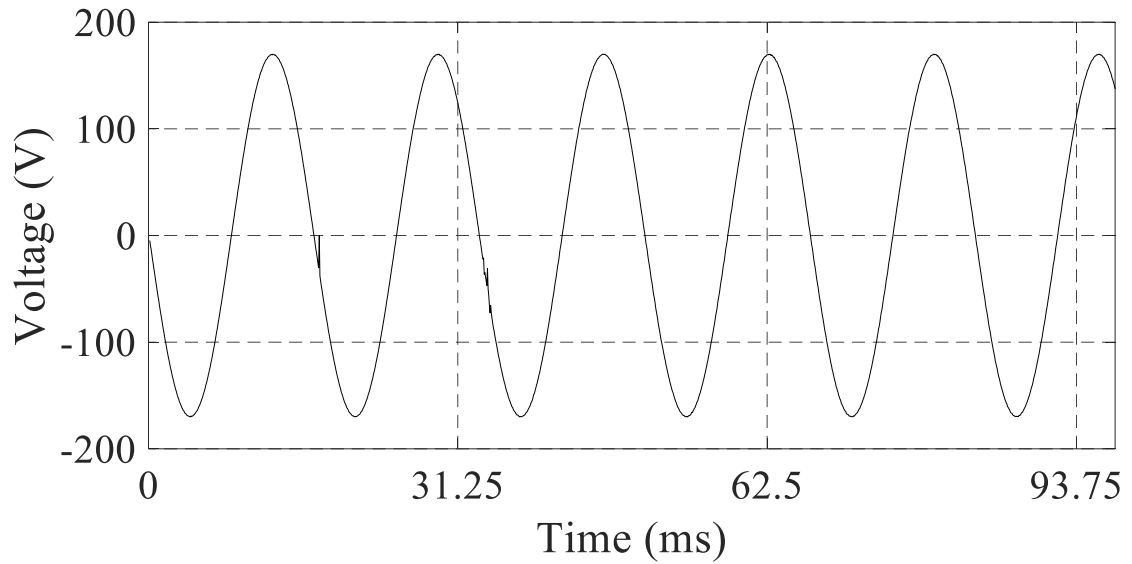


Fig. 5.20: Transient Voltage Signal in case of Battery Charger

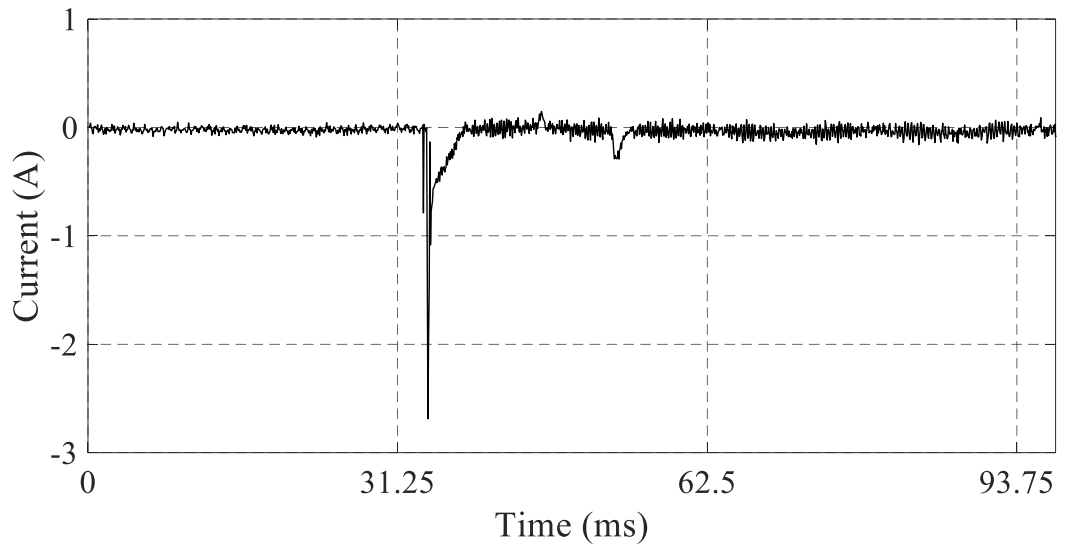


Fig. 5.21: Transient Current Signal in case of Battery Charger

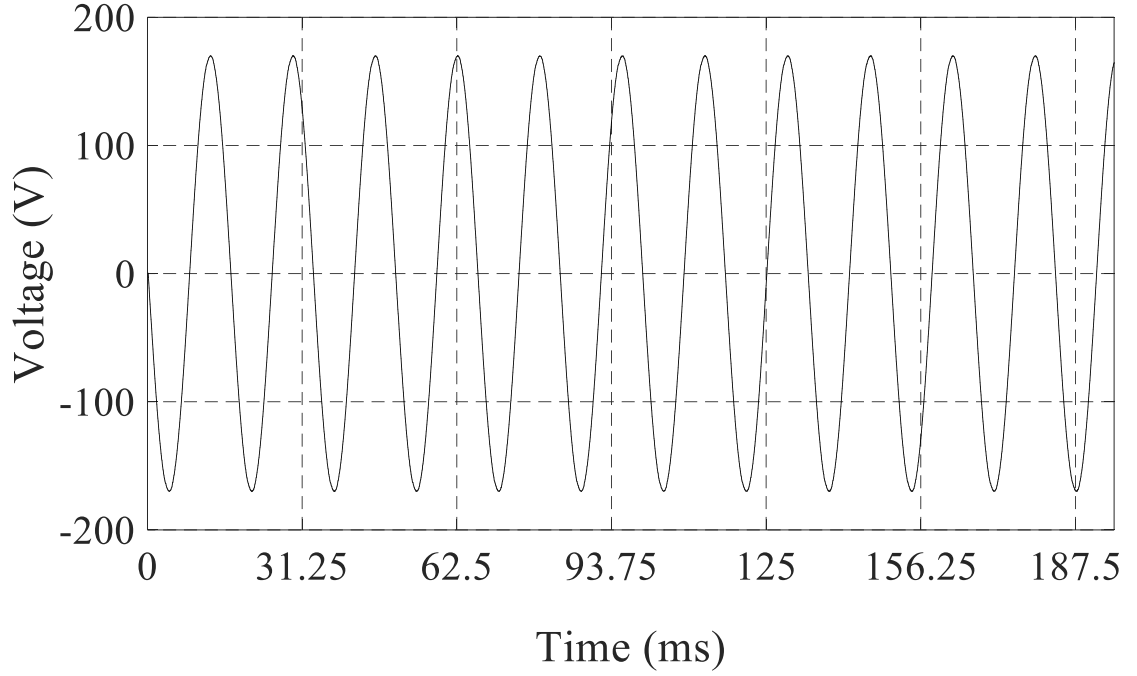


Fig. 5.22: Steady State Voltage Signal in case of Battery Charger

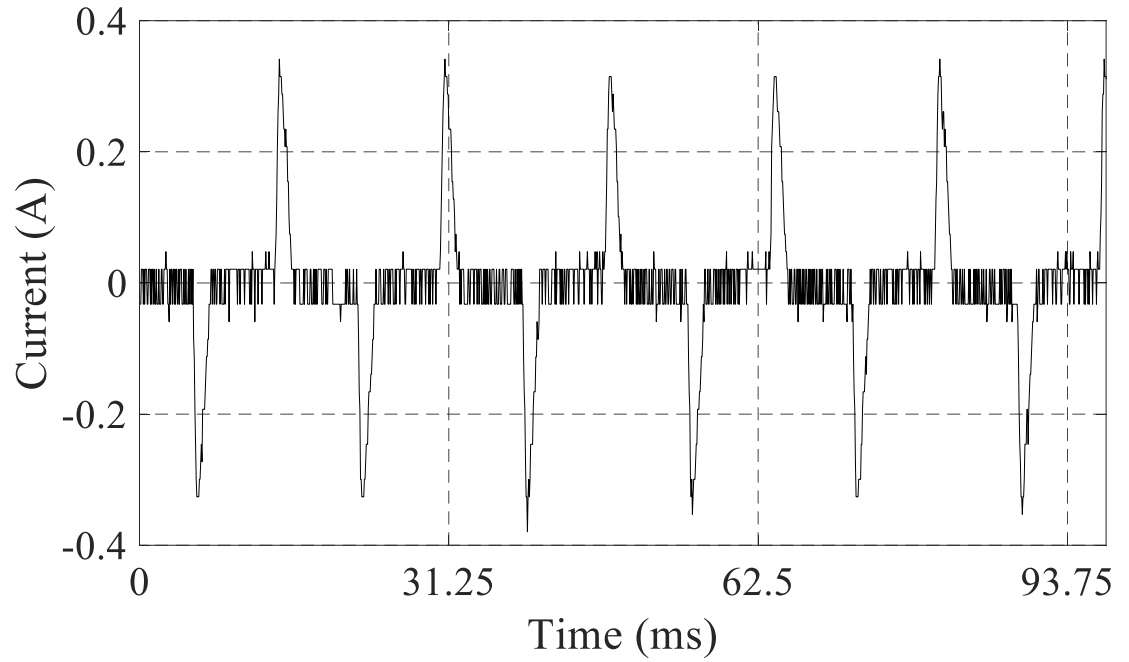


Fig. 5.23: Steady State Current Signal in case of Battery Charger

5.9.1.5 Current Transformer

The CTs selected in this thesis are Siemens CTs [67], which are rated for a five Amp pickup current with a step-down turn's ratio of 175.

5.9.1.6 Data Acquisition Module

The DAQ selected in this thesis is an Omega 1608FS-PLUS [68]. A summary of the specification sheet can be observed in the Appendix, but the basic specifications are that it can sample up to eight channels at a maximum speed of 50 kHz per channel or four channels at 100 kHz while delivering 16-bit accuracy. Since the DAQ allows for USB output, any USB compatible device can be enabled to work. Currently a laptop running Windows 8 is being utilized but in the future it is possible to implement a microcontroller or FPGA in order to implement the algorithm.

5.9.1.7 Programmable AC Power Source

The programmable AC source chosen is a Chroma 61604 [69]. This is a 2000 Volt-Ampere source with an accuracy of $\pm 0.2\%$ and a THD of less than 0.3%. It is capable of varying frequency between 0 and 1,000 Hz while varying voltage between 0 and 300 Volts AC. A summary of the specification sheet can be found in the Appendix.

5.9.1.8 Electronic Circuit Breakers

The electronic circuit breakers selected in this work are Belkin WeMo Insight Smart Home Monitoring devices [70]. These switches are controlled using Wi-Fi and operate in a similar manner to a wall outlet meaning they are able to pass 120 Volts at up to 15 Amps.

5.9.2 Test Environment

The dataset was generated using a Programmable AC Power Source following the same procedures explained in subsection 5.3. The specifications for the AC Source have been described above and are listed in the Appendix, however, measurements were captured in the lab to verify the specification sheet and the measurement equipment. As can be seen from Table 5.7, most of the measurements that were captured are within the

proper range as presented in the specification sheet. A voltage snapshot can be observed in Fig. 5.24.

Table 5.7: Programmable AC Source Power Quality Measurements

Criteria	Specification Sheet	Actual Measurement
Voltage at 120V	119.76-120.24	119.7-120.5
Frequency at 60Hz	59.91-60.09	59.96-59.98
THD-V in %	0.3	0.20-0.31

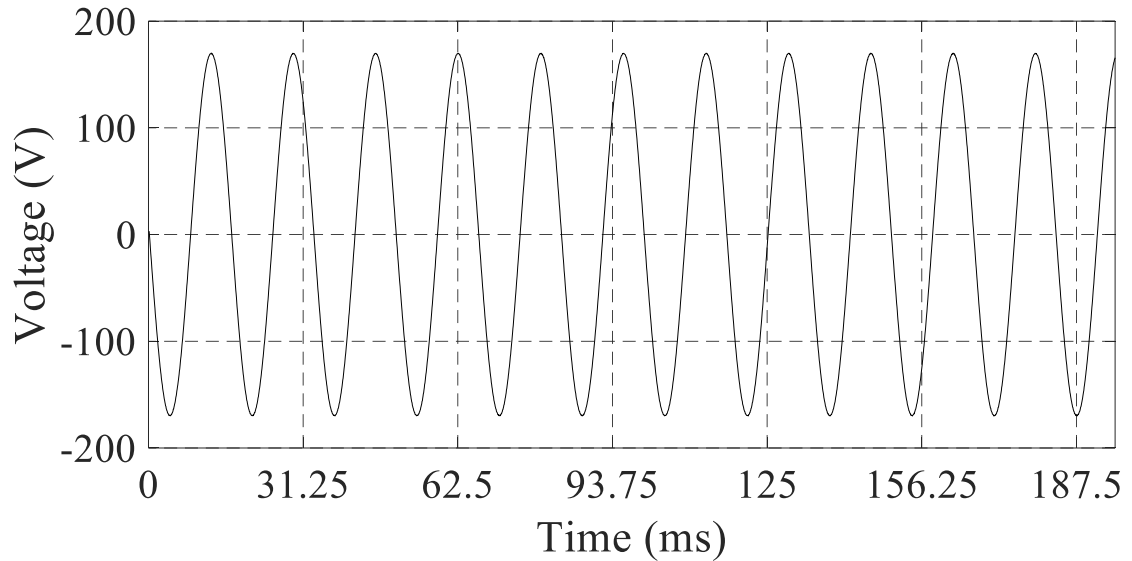


Fig. 5.24: Programmable AC Source Voltage Waveform

5.10 Experimental Dataset Generation






















As a result of the hardware testing, an experimental dataset can be generated to test the proposed technique where each transient will be captured five times to ensure consistency. Upon completion of each possible case (32 cases in total) being switched on five times, the voltage is varied between 114 Volts and 126 Volts in accordance with the ANSI c84.1 standard of voltage variation for Class A [63]. A step change of 2 Volts in each configuration is utilized to train the classifier, where the number of cases is 32 which represents the result of all switching combinations. The voltage is varied between 114 Volts and 126 Volts in 2-Volt steps providing seven levels of voltage, which is applied to the












original 32 cases bringing the total number of cases to $32 \text{ cases} \times 7 \text{ voltage levels} = 224$ cases. These cases were replicated five times which brings the overall number of cases to be $224 \text{ cases} \times 5 \text{ times} = 1,120$ cases. Overall, 1,120 different transients are collected to guarantee a comprehensive dataset is utilized in testing. Two cycles of each transient were captured starting half a cycle before triggering to one and a half cycles after switching. This ensures the entire transient is captured in order to analyze the desired part of the waveform. At the end of the dataset, an integer was added corresponding to the load that was triggered to create that transient. A numeric '1' refers to the ILB, '2' for the CFL, '3' for the PC, and '4' for the Battery Charger. This resulted in a dataset of 1,120 rows and 513 columns ($256 \text{ samples per cycle} \times 2 \text{ cycles} + 1$).

In order to generate this dataset an automated approach was required. A program was written in Python and C that alternates through each case to turn on and off the electronic switches while saving each transient to a CSV file before proceeding to the next case. Each voltage level was manually set, followed by the program being executed for that set of 160 cases where each load is switched to the proper case, followed by the load to be tested being switched on and the transient being saved. Each load would remain in an on state for five seconds which was chosen to give enough time for the load to reach steady state. Next, the loads were switched back off for 30 seconds to make sure the transient expired. The five second and 30 second times were determined experimentally, as it was observed that about one second was necessary for each load to completely reach steady state. Five seconds was selected to ensure that the loads always reached steady state. It was also observed that if the loads were switched within 20 seconds of the previous switching event, the transient may become distorted. Therefore, 30 seconds was preferred to make sure the

transients appeared in a consistent manner. The overall length of one complete run of (32 cases \times 5 of each case = 160 cases) took 93 minutes (35 seconds \times 160 case) to complete. This was repeated seven times to incorporate the voltage magnitude variations resulting in the entire time of gathering data to be about 11 hours if performed continuously. Table 5.8 lists each of the 32 different cases that exist for switching each of the four loads.

Table 5.8: Load Configuration Chart

Case Number	ILB	CFL	PC	Battery Charger
1		0	0	0
2		0	0	1
3		0	1	0
4		0	1	1
5		1	0	0
6		1	0	1
7		1	1	0
8		1	1	1
9	0		0	0
10	0		0	1
11	0		1	0
12	0		1	1
13	1		0	0
14	1		0	1
15	1		1	0
16	1		1	1
17	0	0		0
18	0	0		1
19	0	1		0
20	0	1		1
21	1	0		0

22	1	0		1
23	1	1		0
24	1	1		1
25	0	0	0	
26	0	0	1	
27	0	1	0	
28	0	1	1	
29	1	0	0	
30	1	0	1	
31	1	1	0	
32	1	1	1	

5.10.1 The Experimental Dataset

After completing the above generation of the dataset, some notes about the dataset can be mentioned, such as every transient in the dataset is different. This is an expected and almost obvious result; however, it can be observed by comparing five transients for the same case at the same voltage. It is important to recognize that all of the transients have the same shape but differ in some small ways. Fig. 5.25 shows five different transients of an ILB all taken at the same voltage level and same case.

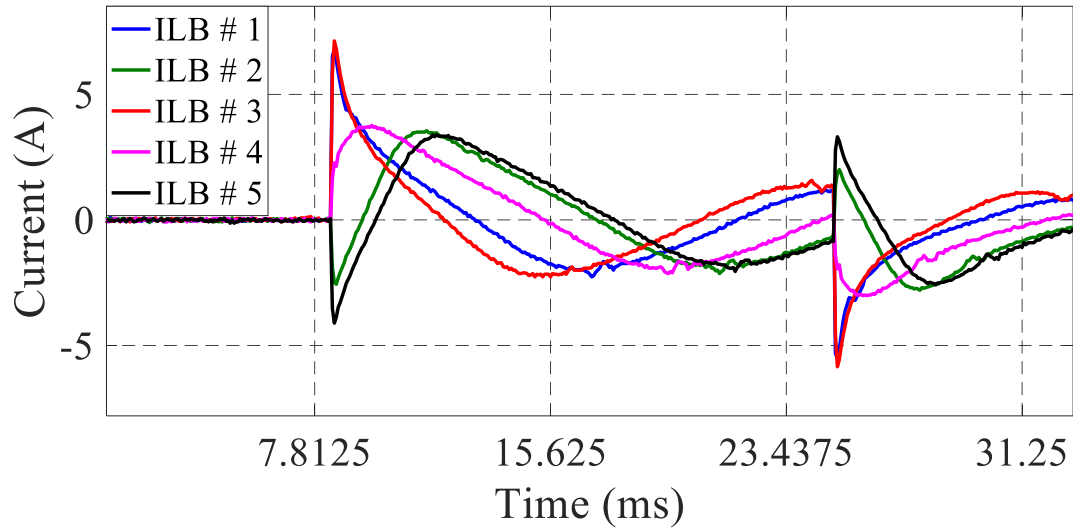


Fig. 5.25: 5 ILB Transients for Case 1 at 114V

One final note is that the transient is dependent on when the load is switched in the voltage cycle. If the voltage is near zero when the load is switched, the transient will be almost zero. This can be observed in Fig. 5.26 and Fig. 5.27. Fig. 6.26 presents the expected transient with the location of where switching occurred in the voltage cycle. Fig. 5.27 displays the current transient recorded when switched at the voltage zero crossing, which is significantly different from the transient that was expected. These transients are included in the dataset as they have an equal possibility of occurring in real life to the dataset. This also provides an explanation into why five switching's were completed for each case.

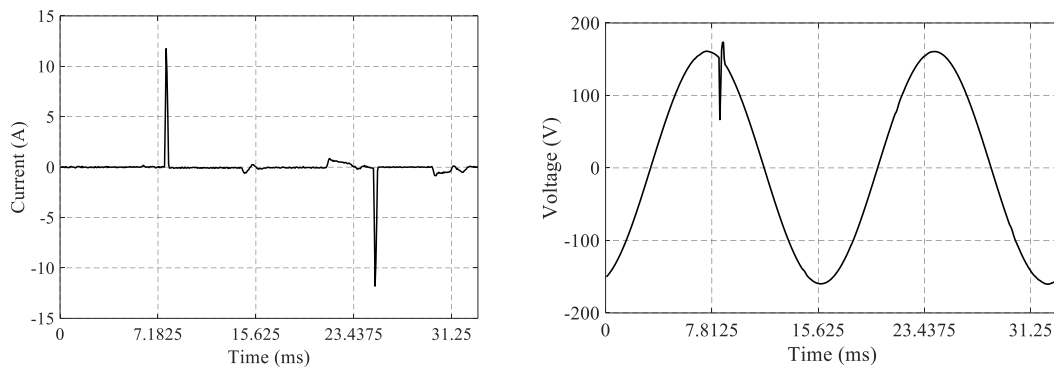


Fig. 5.26: CFL Normal Transients

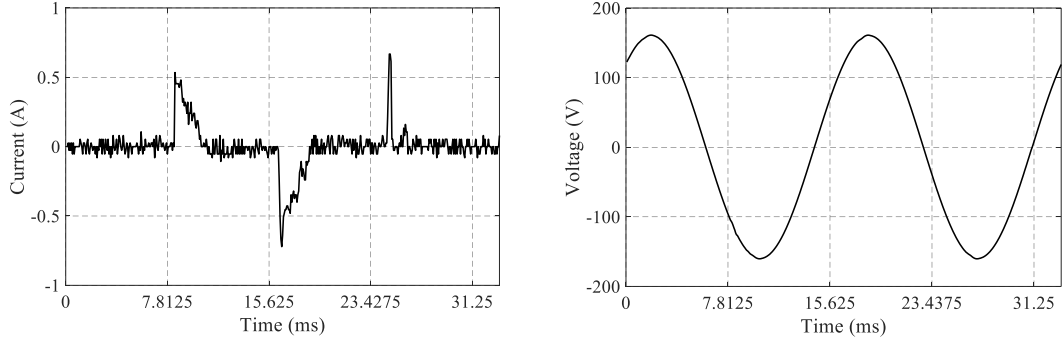


Fig. 5.27: CFL Abnormal Transient

5.11 Computation and Processing

In this section, the classification accuracy is computed after pre-processing the transient signals obtained from the experimental set-up. Monte Carlo is then applied to assess the performance of the machine learning classifiers.

5.11.1 Cycle Lengths

The recorded signals were captured as two cycles in length, half a cycle pre-trigger plus one and a half cycles post-trigger. This allows for testing the algorithm with exactly one cycle of information, consisting of only the 256 samples where the transient existed after switching. This infers no signal is taken regarding pre-switching or after one cycle in processing, where Fig. 5.28 demonstrates one cycle of an ILB transient.

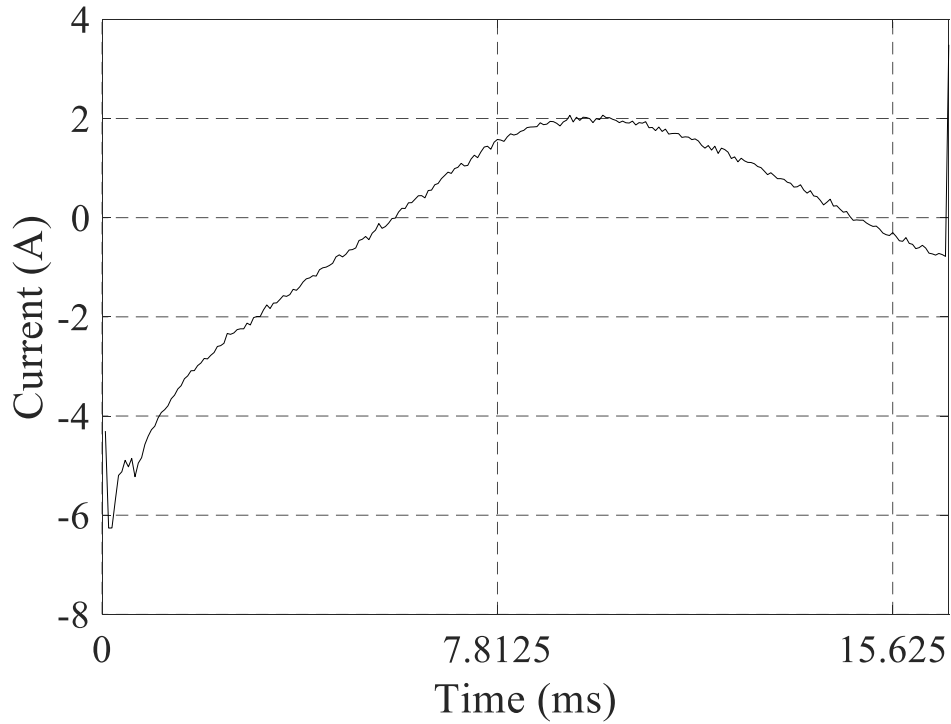


Fig. 5.28: ILB 1 Cycle Transient Example

5.11.2 Monte Carlo method

5.11.2.1 Preprocessing

Before running the Monte Carlo method, wavelets need to be selected for each transient using the same approach used in simulation (i.e., SMM). By relying on Procrustes and covariance, the waveform is compared to all of the GM wavelets to find the one which best matches the switching transient.

5.11.2.2 Processing

The Monte Carlo simulation was set to use 10-fold cross validation and 1,000 iterations. 10-fold cross validation was selected in order to verify the simulation. The 1,000 iterations were experimentally determined to be enough for confidence convergence as can be observed in Fig. 5.29.

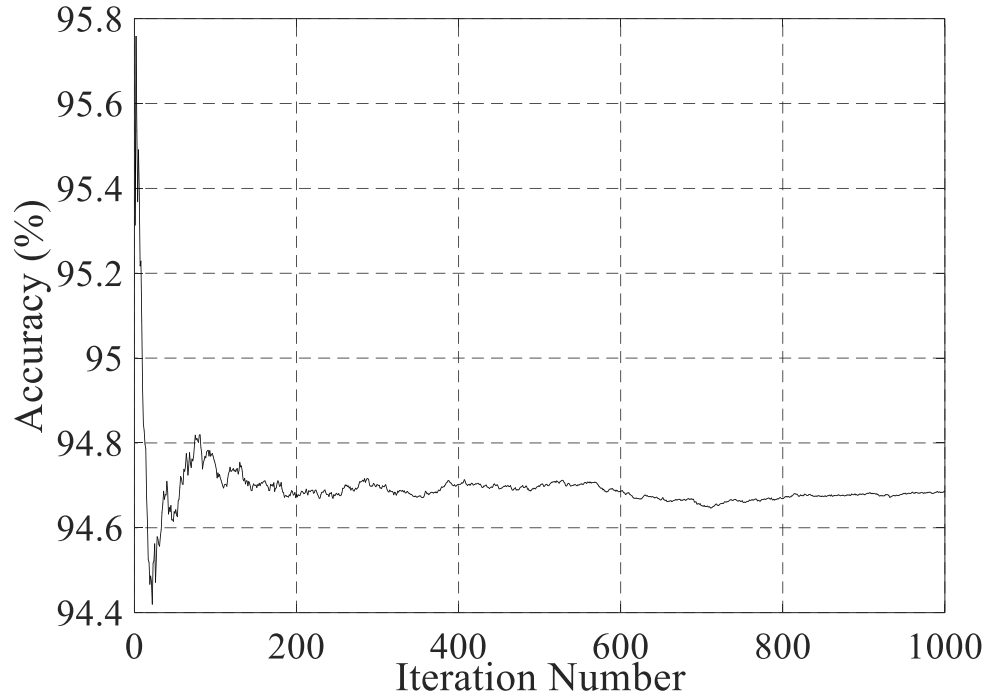


Fig. 5.29: Monte Carlo Accuracy Convergence over Iterations

MATLAB interrupts all of the transients in the experimental dataset from a CSV file, where it separates them into two different matrices. The first matrix separates the transients from their respective classes, while the second matrix extracts one cycle of information starting when switching occurred. After extracting the transient waveforms, the selected wavelets from the SMM perform the Discrete Wavelet Transform in order to calculate the energy of the coefficients. Inside the Monte Carlo loop the data was randomly sampled, with no replacement, to create 10% testing and 90% training sets. This corresponds with the 10-fold technique described in subsection 4.5.4 which provides a pseudo random set during each trial allowing the training sets to be presented to a K-NN and DT classifier in order to train the classifiers to calculate classification accuracy.

Each classifier is trained as a binary classifier utilizing the OAR approach, which implies there is a classifier for each load where if the class is 1, the load is positive, and if the class is 0 the load is negative. The trained classifiers can be tested against the testing

set previously set aside in k-fold. This step provides the accuracy and the confusion matrices of each load which allows for the analysis of the quality of the performance of each classifier.

5.11.2.3 Post Processing

The results of the Monte Carlo trials were saved to a spreadsheet, which included the accuracy of the simulation along with the confusion matrices of all of the classifiers.

5.12 Experimental Results

The results will be presented in four tables. As a benchmark for comparison, the results of the wavelets representing Daubechies of the same resolution have been presented to compare the results of the SMM. The classification accuracy is calculated utilizing Equation (5.1).

5.12.1 Length 6

Table 5.9: Length 6 – Classification Accuracy using K-NN

Load	Daubechies Wavelet	Accuracy (%)	Procrustes SMM	Accuracy (%)	Covariance SMM	Accuracy (%)
Battery	GM3.1.45	84.46	GM3.1.53	71.32	GM3.1.53	71.32
CFL	GM3.1.45	59.64	GM3.1.45	59.64	GM3.1.45	59.64
PC	GM3.1.45	81.52	GM3.1.20	89.67	GM3.1.20	89.67
ILB	GM3.1.45	62.05	GM3.1.12	76.74	GM3.1.12	76.74
Total		71.92		74.34		74.34

Table 5.10: Length 6 – Classification Accuracy for Supervised DT

Load	Daubechies Wavelet	Accuracy (%)	Procrustes SMM	Accuracy (%)	Covariance SMM	Accuracy (%)
Battery	GM3.1.45	95.89	GM3.1.53	94.04	GM3.1.53	94.04
CFL	GM3.1.45	77.14	GM3.1.45	77.14	GM3.1.45	77.14
PC	GM3.1.45	97.19	GM3.1.20	95.54	GM3.1.20	95.54
ILB	GM3.1.45	79.03	GM3.1.12	85.11	GM3.1.12	85.11
Total		87.31		87.96		87.96

5.12.2 Length 8

Table 5.11: Length 8 – Classification Accuracy for Supervised K-NN

Load	Daubechies Wavelet	Accuracy (%)	Procrustes SMM	Accuracy (%)	Covariance SMM	Accuracy (%)
Battery	GM4.1.399	81.014	GM4.1.173	90.86	GM4.1.623	80.64
CFL	GM4.1.399	63.70	GM4.1.451	68.58	GM4.1.460	68.54
PC	GM4.1.399	87.66	GM4.1.173	87.39	GM4.1.173	87.39
ILB	GM4.1.399	72.95	GM4.1.101	78.88	GM4.1.101	78.88
Total		76.33		81.43		78.86

Table 5.12: Length 8 – Classification Accuracy for Supervised DT

Load	Daubechies Wavelet	Accuracy (%)	Procrustes SMM	Accuracy (%)	Covariance SMM	Accuracy (%)
Battery	GM4.1.399	96.48	GM4.1.173	97.80	GM4.1.623	96.00
CFL	GM4.1.399	85.88	GM4.1.451	80.52	GM4.1.460	83.14
PC	GM4.1.399	96.28	GM4.1.173	96.14	GM4.1.173	96.15
ILB	GM4.1.399	83.94	GM4.1.101	82.64	GM4.1.101	82.64
Total		90.65		89.28		89.48

5.13 Discussion and Analysis

After generating the experimental dataset, the energy of the wavelet coefficients were calculated for the chosen wavelets of each load. The 10-fold cross validation for 1,000 Monte Carlo trials was performed to obtain results for each possible situation. Results for Daubechies wavelet were also presented to allow for a comparison between the wavelet

design process and the literature. This section will thoroughly describe the results presented in Tables 5.9-5.12 and these results will be compared to the simulation results when applicable.

5.13.1 Length-6

The results for length-6 and length-8 filters for K-NN and DT were presented in 5.9-5.12. The total classification accuracies for Daubechies wavelet and the wavelets selected with the SMM using Procrustes and covariance are displayed. Immediately the reader should note that the wavelets chosen using Procrustes and covariance for the length-6 filter were the same for all four loads. This is similar to the simulation work in subsection 5.6 where three of the four wavelets that were selected were the same. This is expected as only 91 wavelets exist in the set, implying the number of possible wavelets to choose from is quite low. Experimentally, it can be observed that Daubechies wavelet does not perform as well as the SMM in both K-NN and DT for the length-6 filter. In addition, since both Procrustes and covariance selected the same wavelet their results are the same at 74% and 88% for K-NN and DT, respectively. The result of DT being a better machine learning classifier for this application remain the same as what was observed in the simulation results of subsection 5.6. The simulation results are all higher than what was seen experimentally and in some cases are different by 10%. This is caused by the different transients that were produced in PSCAD against the real transients displayed in experiments and the effect of the voltages zero crossing. Overall for length-6, DT outperforms K-NN and Procrustes and covariance choose the same wavelets for each load as the number of wavelets to choose from is limited to 91.

5.13.2 Length-8

When examining the length-8 wavelet filters, it was observed in simulation that the accuracy improves, which is also the case in the experimental results. Instead of having 91 wavelets to choose from as in length-6 there are now 820 different wavelets in the wavelet library. It is expected and observed that Procrustes and covariance now start to choose different wavelets for some of the loads. Once again DT outperforms K-NN and Daubechies wavelet is only about 1% higher than the SMM wavelets in the case of DT. However, in K-NN the SMM produces a better result than Daubechies wavelet of length-8. Overall for length-8, DT produces a higher result than K-NN, while both outperform what was observed in the length-6 filter. This is the same pattern as in the simulation results demonstrating a pattern is starting to form in which as the wavelet filter length increases the accuracy improves. Moreover, Procrustes and covariance utilizing the SMM choose similar wavelets for some but not all loads as the number of wavelets to choose from is now 820.

5.14 Summary

In this chapter, a simulation is implemented to test the proposed NILM approach. To start, a dataset was generated using PSCAD for a number of different cases in which voltage, frequency and harmonics were varied, in order to test the robustness of the approach where in total 864 different cases were simulated. The transients of these cases were imported into MATLAB where the sequence subtraction was performed and the WT using six levels of decompositions was utilized to calculate the energy of each coefficient. Both K-NN and DT machine learning classifiers were trained using supervised machine learning with k-fold cross validation and tested using Monte Carlo. Classification

accuracies for each machine learning technique were presented, where results showed that DT outperformed K-NN and Procrustes and covariance analysis exceeded Daubechies wavelet in DT for length-6 and length-8 filters. Classification accuracies were highest for covariance analysis using length-8 wavelets at 97.82%. This is the best case as in length-8 there are 820 wavelets that can be selected against 91 in the case of length-6. Next, the experimental work to test the proposed NILM method was presented and from the results conclusions were drawn. This chapter discussed the experimental setup that was utilized while thoroughly describing each hardware component. This included a rigorous test of each load in both steady state and transient conditions. Upon completion of these hardware tests, a dataset was generated using the hardware described in a comparable way to the simulation. The total number of cases generated was 1,120 which forms the dataset. In order to assess the performance of the proposed algorithm, the k-fold cross validation and Monte Carlo for one cycle of the transient waveform was used along with DT and K-NN machine learning techniques. The classification accuracies were computed and compared against Daubechies wavelets with the same resolution. It was observed that the classification accuracies were lower than what was observed in simulation due to the effect of the zero crossing. However, patterns similar to the simulation were observed such as accuracy improving as wavelet length increased.

6. Conclusions and Recommendations

6.1 Conclusion

The work presented in this thesis aims to study the time-frequency analyses techniques applied to the NILM application. The work presented in the literature has been reviewed by analyzing the advantages and disadvantages of each method after grouping them into three categories: time-domain, frequency domain and time-frequency domain techniques. The NILM approach based on the time domain techniques used power quantities such as active power, reactive power, voltage, and current to predict load operation. These techniques only retain the information in the time domain, inferring the information contained in the frequency domain is lost and hence the time-domain based approaches fail to extract the features necessary to detect load operation. This significantly degrades the detection performance and the load identification process. For example, methods which rely on appliance mapping break down when only one time based parameter is available and in cases with resistive loads where only one parameter is useful in distinguishing loads. On the other hand, frequency domain approaches rely on utilizing quantities in the frequency domain such as harmonics and principal component analysis of a frequency feature set. By examining only quantities in the frequency domain, time domain information such as voltage and current waveforms are lost and hence decrease these algorithms abilities to determine load operation. For example, in harmonic analysis, a list of loads with known harmonics are compared against the harmonic profile of a home. However, some loads do not produce harmonics making them impossible to detect with these methods. The time-frequency analysis techniques combine the benefits of time and frequency domain together by preserving the information contained in both domains

simultaneously. The short time Fourier transform and Wavelet Transform are two examples of different time-frequency methods that can be used to extract features from waveforms to detect load operation. After thoroughly investigating both time-frequency analysis techniques, it has been determined that the short-time Fourier transform suffers from a fixed window size and hence any attempt to increase the frequency resolution comes at the expense of poor time resolution and vice versa. Moreover, as a Fourier-based transform, there are only two basic functions (i.e., sine and cosine) which make the transform not suitable for analyzing transients. Therefore, the work in this thesis focuses on time-frequency analysis by employing the wavelet transform with wavelet design for the NILM application.

The wavelet transform was introduced as an advanced time-frequency analysis technique that can be applied to transient waveforms. The energy of the wavelet coefficients for the different decomposition levels can be calculated in order to detect load operation with different machine learning approaches. Attention turned to which wavelet to use in the wavelet transform introducing the wavelet design procedure. The work in this thesis presented 91 wavelets of length-6 and 820 wavelets of length-8 which were designed after clustering was applied for Procrustes and covariance analysis to select a wavelet that best matches the shape of each unique load transient. These chosen wavelets were compared against Daubechies wavelet of the same length at the same resolution in which the wavelet design process with the smart matching module can be compared. Daubechies are common in literature and popular among other signal processing applications due to their orthogonality and being characterized by a maximal number of vanishing moments.

Two different machine learning classifiers were implemented in order to observe if different machine learning techniques influenced the classification accuracy. Specifically, k-nearest neighbor was selected as a lazy learner while decision tree was chosen as an eager learner. Supervised machine learning, in which all data points are labelled was utilized to train the DT and k-NN classifiers using k-fold cross validation, which is a method to split a dataset into training and testing sets. Monte Carlo trials were performed as a method to determine proper classification accuracies based on the randomness introduced by k-fold.

The proposed approach was tested through simulation by creating a dataset based on a test system in PSCAD where each possible switching configuration for four loads was performed and voltage, frequency and harmonics were varied with respect to their given standards. This resulted in a dataset of 864 cases, which was processed based on wavelets chosen through the smart matching module. The energies of the different levels of decomposition were computed in order to train and test the supervised machine learning approaches for both decision tree and k-nearest neighbor. The results demonstrated that wavelet design produced a better result than utilizing Daubechies wavelet. For example, in length-8 with decision tree the classification accuracy using the smart matching module was 97.8% when compared to Daubechies wavelet of 96.54%, which is an increase of over 1%.

Finally, an experiment was performed to verify the simulation utilizing hardware set-up in the smart grid laboratory at UOIT. The dataset was developed by switching on loads in each possible switching configuration five times as to make sure that switching had randomly occurred throughout the voltage waveform to account for the zero crossing. The load transients were processed in the same manner as what the simulation where the results

showed the same patterns that were seen in the simulation. For example, in length-8 with k-nearest neighbor the classification accuracy using the smart matching module was 81.43% when compared to Daubechies wavelet of 76.33%, which is an increase of over 5%. In the case of the length-8 filter, the classification accuracies in both k-nearest neighbor and decision tree increased substantially showing an increase in the number of wavelets to choose from increases classification accuracy, which was also depicted in the simulation results. This shows the proposed non-intrusive load monitoring approach based on wavelet design and machine learning is able to perform well, even in situations where power quality disturbances exist and where switching occurs with respect to the zero crossing.

6.2 Recommendations

Based on the work presented in this thesis, two recommendations can be established. The first is to standardize methods of testing NILM algorithms and the second is to start building smart meters that are able to capture load transients.

In order to standardize methods of testing different NILM approaches a dataset should be constructed to be utilized for all methods of NILM. In doing this, a more accurate score could be awarded to each approach based on the quality of that approach against other methods. This will in turn allow researchers to easily further research in this field while putting all approaches through the same rigorous test. Existing smart meters are allowing utilities to collect data at high sampling rates. These smart meters should be constructed to be able to capture and process switching transients in the voltage and current waveforms. Allowing NILM approaches to be built into brand new smart meters, thus reducing the amount of data being delivered to the utility. If this were commercially implemented, utilities and consumers would be able to access their data in a secure way

without concerns about third party companies processing the data for the purposes of NILM. When third party vendors gain access to the data, not only does the data security become a concern but it also asks the question of what other malicious purposes could the company use the data for.

6.3 Future Work

Some next steps that can be taken to build on the work presented in this thesis are: investigating the effect of more machine learning types, modifying the machine learning approaches to use unsupervised and semi-supervised learning, design of higher order wavelet filters, including more loads of varying load type and power consumption, modifying the technique for a lower sampling rate, and developing an improved method of one-against-the-rest where as the number of loads increases the system complexity remains constant. Each of these additions will allow the proposed technique to be more realistic in a home environment.

References

- [1] "Historical Electricity Prices: OEB", 2016. [Online]. Available: <http://www.ontarioenergyboard.ca/OEB/Consumers/Electricity/Electricity+Prices/Historical+Electricity+Prices>. [Accessed: 19- Oct- 2016].
- [2] Ontario Energy Board, "Developing an Ontario Electricity Support Program", Ontario Energy Board, Toronto, ON, 2014.
- [3] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870-1891, Dec 1992.
- [4] D. Egarter and W. Elmenreich, "Autonomous load disaggregation approach based on active power measurements," in *Proc. Pervasive Computing and Communication Workshops*, St. Louis, MO, 2015, pp. 293-298.
- [5] J. Froehlich, E. Larson, S. Gupta, G. Cohn, M. Reynolds and S. Patel, "Disaggregated End-Use Energy Sensing for the Smart Grid," in *Proc. IEEE Pervasive Computing*, vol. 10, no. 1, pp. 28-39, Jan. 2011.
- [6] H. H. Chang, M. C. Lee, N. Chen, C. L. Chien and W. J. Lee, "Feature extraction based hellinger distance algorithm for non-intrusive aging load identification in residential buildings," in *Proc. Industry Applications Society Annual Meeting*, Addison, TX, 2015, pp. 1-8.
- [7] H. Najmeddine *et al.*, "State of art on load monitoring methods," in *Proc. Power and Energy Conference*, Johor Bahru, 2008, pp. 1256-1258.
- [8] D. Jung, H. H. Nguyen and D. K. Y. Yau, "Tracking appliance usage information using harmonic signature sensing," in *Proc. IEEE International Conference on Smart Grid Communications*, Miami, FL, 2015, pp. 459-465.
- [9] J. A. Hoyo-Montaño, C. A. Pereyda-Pierre, J. M. Tarín-Fontes and J. N. Leon-Ortega, "Overview of Non-Intrusive Load Monitoring: A way to energy wise consumption," in *Proc. International Conference on Power Electronics*, Guanajuato, 2016, pp. 221-226.
- [10] C. Duarte, P. Delmar, K. Barner and K. Goossen, "A signal acquisition system for non-intrusive load monitoring of residential electrical loads based on switching transient voltages," in *Proc. Power Systems Conference*, Clemson, SC, 2015, pp. 1-6.
- [11] L. Du, D. He, R. G. Harley and T. G. Habetler, "Electric Load Classification by Binary Voltage–Current Trajectory Mapping," *IEEE Trans. Smart Grid*, vol. 7, no. 1, pp. 358-365, Jan. 2016.
- [12] H. H. Chang, K. L. Lian, Y. C. Su and W. J. Lee, "Power-Spectrum-Based Wavelet Transform for Nonintrusive Demand Monitoring and Load Identification," *IEEE Trans. Industry Applications*, vol. 50, no. 3, pp. 2081-2089, May 2014.
- [13] J. M. Gillis, S. M. Alshareef and W. G. Morsi, "Nonintrusive Load Monitoring Using Wavelet Design and Machine Learning," *IEEE Trans. Smart Grid*, vol. 7, no. 1, pp. 320-328, Jan. 2016.
- [14] J. M. Gillis and W. G. Morsi, "Non-Intrusive Load Monitoring Using Semi-Supervised Machine Learning and Wavelet Design," *IEEE Trans. Smart Grid*, to be published.
- [15] J. Chung, J. M. Gillis and W. G. Morsi, "Non-Intrusive Load Monitoring Using Wavelet Design and Co-Testing of Machine Learning Classifiers," presented at the *Electrical Power and Energy Conference*, Ottawa, ON, 2016

- [16] J. M. Gillis and W. G. Morsi, "Non-Intrusive Load Monitoring Using Orthogonal Wavelet Analysis, " in *Proc. Canadian Conference on Electrical and Computer Engineering*, Vancouver, BC, 2016
- [17] M. Dong, P. C. M. Meira, W. Xu and W. Freitas, "An Event Window Based Load Monitoring Technique for Smart Meters," *IEEE Trans. on Smart Grid*, vol. 3, no. 2, pp. 787-796, Jun. 2012.
- [18] N. Henao, K. Agbossou, S. Kelouwani, Y. Dubé and M. Fournier, "Approach in Nonintrusive Type I Load Monitoring Using Subtractive Clustering," *IEEE Trans. Smart Grid*, to be published.
- [19] T. Hasan, F. Javed and N. Arshad, "An empirical investigation of V-I trajectory based load signatures for non-intrusive load monitoring," in *Proc. IEEE PES General Meeting*, National Harbor, MD, 2014, pp. 1-1.
- [20] A. I. Cole and A. Albicki, "Algorithm for nonintrusive identification of residential appliances," in *Proc. IEEE International Symposium on Circuits and Systems*, Monterey, CA, 1998, pp. 338-341 vol.3.
- [21] S. Drenker and A. Kader, "Nonintrusive monitoring of electric loads," in *Proc. IEEE Computer Applications in Power*, vol. 12, no. 4, pp. 47-51, Oct 1999.
- [22] A. Marchiori, D. Hakkarinen, Q. Han and L. Earle, "Circuit-Level Load Monitoring for Household Energy Management," in *Proc. IEEE Pervasive Computing*, vol. 10, no. 1, pp. 40-48, Jan.-March 2011.
- [23] Z. Wang and G. Zheng, "Residential Appliances Identification and Monitoring by a Nonintrusive Method," *IEEE Trans. Smart Grid*, vol. 3, no. 1, pp. 80-92, Mar. 2012.
- [24] A. J. Bijker, Xiaohua Xia and Jiangfeng Zhang, "Active power residential non-intrusive appliance load monitoring system," in *Proc. AFRICON*, Nairobi, 2009, pp. 1-6.
- [25] X. C. Le, B. Vrigneau and O. Sentieys, "l1-Norm minimization based algorithm for non-intrusive load monitoring," in *Proc. Pervasive Computing and Communication Workshops*, St. Louis, MO, 2015, pp. 299-304.
- [26] D. Egarter, V. P. Bhuvana and W. Elmenreich, "PALDi: Online Load Disaggregation via Particle Filtering," *IEEE Trans. Instrumentation and Measurement*, vol. 64, no. 2, pp. 467-477, Feb. 2015.
- [27] S. Makonin, F. Popowich, I. V. Bajić, B. Gill and L. Bartram, "Exploiting HMM Sparsity to Perform Online Real-Time Nonintrusive Load Monitoring," *IEEE Trans. Smart Grid*, vol. 7, no. 6, pp. 2575-2585, Nov. 2016.
- [28] H. H. Chang, L. S. Lin, N. Chen and W. J. Lee, "Particle Swarm Optimization based non-intrusive demand monitoring and load identification in smart meters," in *Proc. Industry Applications Society Annual Meeting*, Las Vegas, NV, 2012, pp. 1-8.
- [29] T. Guzel and E. Ustunel, "Principal components null space analysis based non-intrusive load monitoring," in *Proc. Electrical Power and Energy Conference*, London, ON, 2015, pp. 420-423.
- [30] C. Y. Feng, H. M. Hoe, M. P. Abdullah, M. Y. Hassan and F. Hussin, "Tracing of energy consumption by using harmonic current," in *Proc. IEEE Student Conference on Research and Development*, Putrajaya, 2013, pp. 444-449.
- [31] D. Bradley, "Applying predictive maintenance to power quality, " in *Proc. Telecommunications Energy Conference*, Edinburgh, UK, 2001, pp. 229-237.
- [32] T. S. Sidhu and Z. Xu, "Detection of Incipient Faults in Distribution Underground Cables," *IEEE Trans. Power Delivery*, vol. 25, no. 3, pp. 1363-1371, Jul. 2010.

- [33] C. Duarte, P. Delmar, K. W. Goossen, K. Barner and E. Gomez-Luna, "Non-intrusive load monitoring based on switching voltage transients and wavelet transforms," in *Proc. Future of Instrumentation International Workshop*, Gatlinburg, TN, 2012, pp. 1-4.
- [34] S. B. Leeb, S. R. Shaw and J. L. Kirtley, "Transient event detection in spectral envelope estimates for nonintrusive load monitoring," *IEEE Trans. Power Delivery*, vol. 10, no. 3, pp. 1200-1210, Jul. 1995.
- [35] S. R. Shaw, S. B. Leeb, L. K. Norford and R. W. Cox, "Nonintrusive Load Monitoring and Diagnostics in Power Systems," *IEEE Trans. Instrumentation and Measurement*, vol. 57, no. 7, pp. 1445-1454, Jul. 2008.
- [36] S. Kong, Y. Kim, R. Ko and S. K. Joo, "Home appliance load disaggregation using cepstrum-smoothing-based method," *IEEE Trans. Consumer Electronics*, vol. 61, no. 1, pp. 24-30, Feb. 2015.
- [37] J. M. Alcalá, J. Ureña and Á Hernández, "Event-based detector for non-intrusive load monitoring based on the Hilbert Transform," in *Proc. 2014 IEEE Emerging Technology and Factory Automation*, Barcelona, 2014, pp. 1-4.
- [38] 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 Trans. Industry Applications*, vol. 48, no. 2, pp. 764-771, Mar. 2012.
- [39] H. H. Chang, K. L. Lian, Y. C. Su and W. J. Lee, "Power-Spectrum-Based Wavelet Transform for Nonintrusive Demand Monitoring and Load Identification," *IEEE Trans. Industry Applications*, vol. 50, no. 3, pp. 2081-2089, May 2014.
- [40] Y. C. Su, K. L. Lian and H. H. Chang, "Feature Selection of Non-Intrusive Load Monitoring System Using STFT and Wavelet Transform," in *Proc. e-Business Engineering*, Beijing, 2011, pp. 293-298.
- [41] K. L. Chen, H. H. Chang and N. Chen, "A new transient feature extraction method of power signatures for Nonintrusive Load Monitoring Systems," in *Proc. Applied Measurements for Power Systems*, Aachen, 2013, pp. 79-84.
- [42] H. H. Chang, K. L. Lian, Y. C. Su and W. J. Lee, "Energy spectrum-based wavelet transform for non-intrusive demand monitoring and load identification," in *Proc. Industry Applications Society Annual Meeting*, Lake Buena Vista, FL, 2013, pp. 1-9.
- [43] H.-H. Chang, "Non-Intrusive Demand Monitoring and Load Identification for Energy Management Systems Based on Transient Feature Analyses," *Energies*, vol. 5, no. 12, pp. 4569-4589, Nov. 2012.
- [44] M. Mathis, A. Rumsch, R. Kistler, A. Andrushevich and A. Klapproth, "Improving the Recognition Performance of NIALM Algorithms through Technical Labeling," in *Proc. Embedded and Ubiquitous Computing*, Milano, 2014, pp. 227-233.
- [45] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, 2nd Edition, Wellesley, MA, USA, 1996.
- [46] "Wavelet Toolbox - MATLAB", *Mathworks.com*, 2016. [Online]. Available: <https://www.mathworks.com/products/wavelet/>. [Accessed: 19- Oct- 2016].
- [47] P.-N. Tan, *Introduction to Data Mining*. Boston, MA, USA: Pearson Education, 2007.
- [48] E. K. P. Chong and S. H. Zak, *An Introduction to Optimization*. New York, NY: John Wiley & Sons, 2001.
- [49] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1992.
- [50] I.L. Dryden and K.V. Mardia, *Statistical Shape Analysis*, John Wiley & Sons, 1998.

- [51] Y. Geerts, W. Sansen and M. Steyaert, *Design of Multi-Bit Delta-Sigma A/D Converters*. New York, NY: Springer, 2002.
- [52] L. El Ghaoui, G. Li, Viet-An Duong, V. Pham, A. Srivastava, K. Bhaduri, "Sparse Machine Learning Methods for Understanding Large Text Corpora," in *Proc. Intelligent Data Understanding*, Mountain View, CA, 2011
- [53] J. Xu, H. He and H. Man, "DCPE co-training: Co-training based on diversity of class probability estimation," in *Proc. International Joint Conference on Neural Networks*, Barcelona, 2010, pp. 1-7.
- [54] Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. New York, NY: Chapman & Hall/CRC, 1984
- [55] A. F. Atiya, "Estimating the Posterior Probabilities Using the K-Nearest Neighbor Rule," *Neural Computing*, vol. 17, no. 3, pp. 731-740, March 2005.
- [56] R. Kohavi., A study of cross-validation and bootstrap for accuracy estimation and model selection." in *Proc. International joint conference on Artificial intelligence - Volume 2*, San Francisco, CA, 1995, 1137-1143.
- [57] N. Metropolis and S. Ulam, "The Monte Carlo method", *J. Amer. Stat. Assoc.* vol. 44, no. 247, pp. 335-341, 1949.
- [58] "Dataport: A Universe of Energy Data, Available Around the World", [Online]. Available: <https://dataport.pecanstreet.org/>. [Accessed: 19- Oct- 2016].
- [59] J. Kolter and M. Johnson. "REDD: A public data set for energy disaggregation research". in *Proc. 2011 SustKDD workshop on Data Mining Applications in Sustainability*, 2011.
- [60] V. Lizard, "PSCAD Home: PSCAD", 2016. [Online]. Available: <https://hvdc.ca/pscad/>. [Accessed: 19- Oct- 2016].
- [61] P. R. Nasini, N. R. Narra and Santosh A. "Modeling and Harmonic Analysis of Domestic/Industrial Loads", *International Journal of Engineering Research and Applications*. vol. 2, no. 5, pp. 485-491, 2012.
- [62] C. Venkatesh, D. Srikanth Kumar, D.V.S.S. Siva Sarma, and M. Sydulu. "Modelling of Nonlinear Loads and Estimation of Harmonics in Industrial Distribution System". in *Proc. National Power Systems Conference*, Bombay, 2008.
- [63] *American National Standard for Electric Power Systems and Equipment-voltage Ratings (60 Hz)*, ANSI C84.1-2011, by National Electrical Manufacturer Association, Rosslyn, VA, 2011.
- [64] *IEEE Guide for Abnormal Frequency Protection for Power Generating Plants*," IEEE Standard C37.106-2003 (*Revision of ANSI/IEEE C37.106-1987*), 2004
- [65] Philips Lighting Holding B.V, "Standard A Shape Frosted", Feb. 2016.
- [66] Dell, "Dell Studio 1535/1536 Quick Reference Guide", Dell, Round Tock, TX, 2016.
- [67] A. Gilani, "Outstanding functionality in a compact design", Siemens, Oakville, ON, 2012.
- [68] Omega, "8-Channel Simultaneous Analog Input USB Data Acquisition Modules", Omega, 2016.
- [69] S. Hughes-Jelen, "Low Power Programmable AC Source with Transients 61500 Chroma", *Chroma Systems Solutions Inc.*, 2016. [Online]. Available: <http://www.chromausa.com/product/low-power-programmable-ac-source-61500/>. [Accessed: 19- Oct- 2016].
- [70] WeMo Electronic switches home automation, Belkin. [Online], Available: <http://www.belkin.com/us/Products/home-automation/c/wemo-home-automation/>

Appendix

Hardware Component Specifications

Phillips Incandescent Light Bulb

Base	Medium (Single Contact Medium Screw)
Base Information	Aluminum Base
Bulb	A19
Bulb Finish	Soft White
Filament Shape	CC6 (Straight)
Operating Position	Universal (Any or Universal)
Main Application	Household
Atmosphere	Gas
Life with 3h/day use	1.4 years
Color Temperature	2750
Lumens (Brightness)	830
Power	60 W
Voltage	120 V

Dell Studio 1535 Personal Computer

Battery	
Type	9-cell smart lithium ion
Weight	0.34kg (0.75 lb.)
Voltage	11.1 VDC (6/9 cell)
Charge Time	4 hours
Life Span	300 discharge/charge cycles
Operating Temperature	0° to 35° (32°F to 95°F)
Storage Temperature	-40°C to 65°C (-40° to 149°F)
Coin-cell battery	CR-2032
AC Adapter	
Input	100-240 VAC
Input Current (maximum)	1.5A
Input Frequency	50-60 Hz
Output Current 90 W	4.62 A (continuous)
Dimensions	
Height	34.2 mm (1.35in)
Width	60.9 mm (2.39in)
Length	153.42mm (6.04in)
Temperature	
Operating	0° to 35°C (32° to 95°F)
Storage	-40° to 65°C (-40 to 149°F)

Omega 1608FS-PLUS Data Acquisition Module

Analog inputs	8 single-ended
A/D Sampling Rate (Continuous Scan to Computer Memory)	100kS/s maximum for any channel
Channel-Gain Current	Up to 8, ordered elements
Digital Output Current	±2.4 mA per pin
Calibration	Factory calibration only
External Clock Input	100 kHz maximum
Trigger Sensitivity	Edge or level sensitive
Counter Input	1 MHz input frequency maximum
Range	Accuracy
±10V	5.66mV
±5V	2.98mV
±2V	1.31mV

Chroma Programmable AC Source Model# 61604

Output Rating-AC	
Output Phase	2000VA
Power/Phase	2000VA
Voltage Range/Phase	150V/300V/Auto
Voltage Accuracy	0.2%+0.2%F.S.
Voltage Resolution	0.1V
Voltage Distortion*1	0.3%@50/60Hz, 1%@15-kHz
Voltage Line Regulation	0.1%
Voltage Load Regulation*2	0.2%
Max. Current/Phase R.M.S	16A/8A (150V/300V)
Max. Current/Phase Peak	96A/48A (150V/300V)
Frequency Range	DC, 15-1kHz
Frequency Accuracy	0.15%
Output Rating-DC	
Power	1000W
Voltage	212V/424V
Current(per phase)	8A/4A(212V/424V)
Range	0Ω+200μH-1Ω+1mH
Harmonics Bandwidth	2400Hz
Input	
Voltage Range	90/250V, 1Ø
Frequency Range	47-63Hz
Current(per phase)	28A Max @ 90V
Interface	GPI8, RS-232 (Optional)
Dimensions(W×H×D)	482x132.6x570mm
Weight	21kg

Dranetz PX5 Power Quality Analyzer

Analog inputs	4 differential @ 1-600 Volts RMS 4 differential @ 1-6000 Amps RMS
A/D Sampling Rate (Continuous Scan to Computer Memory)	256 samples / cycle or 15360Hz with 16 bit accuracy
Calibration	0.1% rdg + 0.05% FS
Maximum Sampling Rate	1 MHz with 14 bit accuracy
Trigger Sensitivity	Edge or level sensitive
Frequency Resolution	10mHz
Size (H x W x D)	12" x 2.5" x 8" (30cm x 6.4cm x 20.3 cm)
Weight	4.2 Pounds (1.9kg)
Additional Features	IEEE 1159 Compliant
	IEC 61000-4-30 Class A Compliant
	EN50160 Quality of Supply Compliant
	Voltage Fault recording
	Long Term Monitoring
	Continuous Data Logging with min/max and average
	Cycle-by-cycle analysis
	RMS Recordings V&I
	Wave shape Recordings
	Low and Medium Frequent Transients – V&I
	High Frequency Transients – V&I, 3% FS trigger
	Harmonics Summary
	RMS Event Characterization (IEEE or IEC)
	Transient Event Characterization
	Harmonics/Inter-harmonics per IEC 1000-4-7
	THD/Harmonic Spectrum