

Non-Intrusive Load Monitoring and Load Disaggregation of Distributed DC Networks

Tristan Kuisis

A dissertation submitted to the Faculty of Engineering and the Built Environment,
University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for
the degree of Master of Science in Engineering.

Johannesburg, May 2021

Declaration

I declare that this dissertation is my own, unaided work, except where otherwise acknowledged. It is being submitted for the degree of Master of Science in Engineering to the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination to any other university.

Signed this 17 day of May 2021



Tristan Kuisis

Abstract

The work presented extends and contributes to research in non-intrusive load monitoring and load disaggregation which focuses on utilizing these techniques by measuring the grid voltage on distributed DC electrical networks to observe and distinguish the operation of devices on the network. Although previous work in this area has produced practical guidelines and models that may be used to detect and distinguish devices operating on electrical networks, there has not been an investigation into applying the techniques to distributed DC systems. In the research presented, the techniques of non-intrusive load monitoring and load disaggregation have been applied to such systems by measuring the voltage of the network. Measurements are obtained by the design and construction of an automated system such that a combination of devices and distributed DC networks as well as a number of test protocols/scenarios can be tested. The measurement data—voltage transients of device operation events—are then run through a pre-processing pipeline (feature extraction, filtering, and normalization) followed by clustering algorithms that are used to determine if devices exhibit unique and repeatable transient signatures when operating (changing states). It is shown that it is indeed possible to distinguish device operation by measuring the grid voltage because different devices exhibit unique and repeatable transients onto the grid voltage. Limitations appear where devices emit noise onto the grid, drowning out the transients of further device switching; the limitations on the sampling rates on such systems—to achieve sufficient disaggregation performance—should exceed 20 kS/s. This sampling limitation is based on the observations that low-power devices exhibit small transients on the grid voltage and higher-powered devices have larger and longer lasting transients. This represents a unique and valuable contribution to engineers working in the field of non-intrusive load monitoring and load disaggregation as it creates visibility to the operation of devices on distributed DC electrical networks.

I would like to thank my parents for their great patience and support throughout my academic studies.

I would like to thank my two supervisors - Professors Ken Nixon and Willie Cronje for their guidance and advice throughout.

Contents

Declaration	i
Abstract	ii
Dedication	iii
Contents	iv
List of Figures	x
List of Tables	xv
Nomenclature	xvii
1 Introduction	1
1.1 Problem Statement and Motivation	3
1.1.1 Measurement Parameter	4
1.1.2 The Measurement Technique	4
1.1.3 The NILM Implementation and Load Disaggregation	4
1.1.4 Distributed DC Grid	5
1.2 Research Aims	5

1.2.1	Research Question	5
2	Background	7
2.1	The Energy Perspective	7
2.1.1	Energy Management	7
2.1.2	System Information	8
2.2	Load Monitoring	9
2.3	Intrusive Load Monitoring	10
2.3.1	ILM Criticism	11
2.4	Non-Intrusive Load Monitoring	12
2.4.1	NILM Criticism	13
2.5	ILM and NILM Applications	14
2.6	Load Disaggregation	14
2.6.1	State-Based Approach	15
2.6.2	Event Detection	16
2.7	Device Types, Signatures, and Features	16
2.7.1	Device Types	17
2.7.2	Device Features	19
2.8	Machine Learning Techniques and Inference	21
2.8.1	Supervised and Unsupervised Learning	22
2.8.2	NILM and Load Disaggregation Example	25
2.8.3	The Utility of NILM	28
2.9	DC Technologies	31

3	Problem	33
3.1	Distributed DC and NILM	33
3.2	The Theory and Operation of the System	34
3.2.1	DC Bus Voltage	35
3.2.2	The NILM Implementation	36
3.3	Switching Protocols	41
3.3.1	Disaggregation Dataset Generation	41
3.4	Validation	42
3.5	Sampling	42
4	Methodology	45
4.1	DC Test Bed	45
4.1.1	DC Battery	47
4.1.2	AC-DC Power Supply	48
4.1.3	The PeCogrid	49
4.2	DC Devices	51
4.2.1	Tested Devices	51
4.2.2	Device Connections	52
4.2.3	DC Device Comments	56
4.3	Automated Switching System	57
4.3.1	Physical System Characteristics	59
4.3.2	Event Detector Requirement	60
4.4	Measurement System	60

4.4.1	Oscilloscope Specification Requirements	61
4.4.2	Additional Measurement Details	62
4.4.3	Measurement Automation	64
4.4.4	System Coordination	65
4.5	Testing Protocols	67
4.5.1	Source Protocols	68
4.5.2	Load Protocols	69
4.5.3	Unique Protocols	70
4.5.4	Testing Protocols Implementation	71
4.5.5	Sampling Considerations	72
4.6	Data Processing System	74
4.6.1	Ground Truth	75
4.6.2	Simple Ground Truth	76
4.6.3	Complex Ground Truth	81
4.7	Downsampling	83
4.7.1	Downsampling Theory	86
4.7.2	Integer Factor Downsampling	87
4.7.3	Rational Value Downsampling	87
4.7.4	Downsampling Rates	88
4.7.5	The Utility of Downsampling	91
4.8	Feature Engineering	92
4.8.1	Feature Extraction	95

4.8.2	Feature Normalization	97
4.8.3	Feature Filtering	99
4.8.4	The Utility of Feature Extraction	107
4.9	Clustering Techniques	109
4.9.1	The Aim of Clustering	109
4.9.2	Clustering Evaluation Techniques	111
4.9.3	Clustering Algorithms	114
4.9.4	The Utility of PCA	116
4.10	Experimental Optimization Process	119
5	Analysis	121
5.1	Device Analysis	121
5.1.1	Unique Device Signatures	122
5.1.2	Repeatability Testing	133
5.1.3	Device Signature Discussion	135
5.2	Ground Truth Concept	141
5.2.1	Device Noise	141
5.3	Source Analysis	145
5.3.1	System Response Analysis	145
5.4	Sampling Analysis	147
5.4.1	System Sampling Model	148
5.4.2	Feature Analysis	150
5.4.3	Clustering Algorithm Comparison	152

6	Conclusion & Future Recommendations	154
6.1	Conclusion	154
6.1.1	Problem Statement and Motivation Outcomes	155
6.1.2	Research Question Conclusions	156
6.1.3	Future Recommendations	156
6.1.4	Physical System Setup	157
6.1.5	NILM and Load Disaggregation Techniques	160
6.1.6	Current Usefulness	160
6.1.7	Meta Information Benefits	161
	References	162
A	Appendix	180
A.1	Four State Device	180
A.2	Decreasing Grid Voltage	181
A.3	Linear Regulator Power Supply	181
A.4	Multiple Similar Devices	182
A.5	Concurrent Device Switching	182
A.6	Current Signatures	183
A.7	Source Combinations	183
A.8	Inverter Driven Loads	185

List of Figures

1.1	Comparison of ILM and NILM implementations	1
2.1	Energy Management Process	8
2.2	Generic NILM Process [16, 19]	12
2.3	Device type waveform examples described in section 2.7.1	18
2.4	Device State Transition Types [1, p. 1874-1875]	19
2.5	Features Breakdown [14, p. 12]	20
2.6	Three device clustering	24
2.7	Example Devices	26
2.8	Example Measurement	26
2.9	Possible power states with two devices	27
3.1	ILM on a distributed generation system	34
3.2	NILM on a distributed generation system	35
3.3	Clustering of three devices and their labelling	39
3.4	Unsupervised Process	40
3.5	Illustrations of different cost-performance models	43
4.1	Generic DC grid - single source	46
4.2	Generic DC grid - multiple sources	47

4.3	VI operating curve for a DC power supply	48
4.4	Concept of PeCogrid interfacing technologies	50
4.5	Type II device possible internal construction [48]	53
4.6	Trinary switch device original switch mechanism	54
4.7	Modified trinary switch	54
4.8	Device with rotary switch component	55
4.9	Switch and device placements within the circuit	55
4.10	Single direction state change switch	56
4.11	Source and device system layout (controllable relays represent the Siemens Logo!)	57
4.12	Relay positioning for different devices	58
4.13	Vertical setup of oscilloscope	63
4.14	Comparison of characteristics between different coupling types	64
4.15	Variance in event starting points due to variation in Logo! and oscilloscope (values illustrated in list)	66
4.16	Coordination of Logo! and oscilloscope for starting measurement recording and relay actuation	67
4.17	Example Protocol	72
4.18	Scatter plot of event feature vectors (two example features) - single device type I protocol	78
4.19	Scatter plot of event feature vectors (two example features) - two devices of type I	80
4.20	Complex ground truth clusters of two device protocol signatures . . .	82
4.21	Visual representation of different downsampled versions of an event .	85

4.22	Downsampling process for an event [114, p. 167]	86
4.23	Oscilloscope sampling rate options	90
4.24	Generated Renard series sampling rate options	90
4.25	Doubling Downsampled Rates	92
4.26	Common approaches to time-series clustering	93
4.27	Overall process for feature extraction, normalization, and data clustering	95
4.28	Examples of a number of time series features on an event waveform (exhaustive feature list can be found in [122])	96
4.29	Different types of feature scaling	98
4.30	Feature analysis correlation	103
4.31	Distribution of two features and their ground truth	103
4.32	ANOVA Statistic - high F-value	104
4.33	ANOVA Statistic - low F-value	104
4.34	ANOVA Statistic - high p-value	105
4.35	ANOVA Statistic - low p-value	105
4.36	Pearson correlation heatmap for different features	106
4.37	Feature performance rolloff with K-Means and Agglomerative clustering	108
4.38	Clustering data without labels	110
4.39	Clustering data with labels applied from output of clustering algorithm.	110
4.40	Example single device dendrogram	117
4.41	The comparison between the ideal case of increasing performance with increasing features compared to reality	118

4.42	Example PCA visualization application with two- and three-dimensional scatter plots with K-Means clustering	119
4.43	Example PCA visualization application with two- and three-dimensional scatter plots (colouring from K-Means clustering labels)	119
5.1	High-power type I device - state transitions	123
5.2	Low-power type I device - state transitions	123
5.3	Low-power type I device zoomed - state transitions	124
5.4	K-Means clustering plots for a low- and high-power device	125
5.5	High-power type I device dendrogram (Even numbered events are ON events, odd numbered events are OFF events)	126
5.6	Low-power type I device dendrogram (Odd numbered events are ON events, even numbered events are OFF events)	126
5.7	Type II device - state transition waveforms	127
5.8	Clustering representation of type II device	128
5.9	Dendrogram representation of type II device	129
5.10	Two devices (type I) clustering	131
5.11	ON and OFF clusterings of nine devices	133
5.12	High-power type I device transient waveform repeatability	134
5.13	Low-power type I device transient waveform repeatability	135
5.14	Type II device transient waveform repeatability	135
5.15	Replica device - same physical installation	138
5.16	Replica device - LED 2 shortened leads	140
5.17	PC fan type I device - state transitions	142
5.18	LED Light transients with fan operating	143

5.19 Visual representation of clusters formed with single noise emitting device in two device protocol.	145
5.20 Clustering performance for three device power ranges with increasing downsampling rate	149
5.21 Different types of feature scaling	151
5.22 Feature performance rolloff ANOVA	151
6.1 Benefits that an event detector has on measuring the correct samples in time.	160
A.1 Type II device (four states total) - state transition waveforms	181
A.2 DC Battery Device 1 - Transient waveform uniqueness and repeatability	183
A.3 DC Battery Device 2 - Transient waveform uniqueness and repeatability	184
A.4 DC Battery Device 3 - Transient waveform uniqueness and repeatability	184
A.5 Smart Battery Device 1 - Transient waveform uniqueness and repeatability	184
A.6 Smart Battery Device 2 - Transient waveform uniqueness and repeatability	184
A.7 Smart Battery Device 2 - Transient waveform uniqueness and repeatability	185

List of Tables

2.1	Example Load Profile of three devices (A, B, and C)	26
4.1	DC devices	52
4.2	Device type I states and corresponding relay states	58
4.3	Device type II states and corresponding relay states	59
4.4	Ground Truth for Example Protocol	77
4.5	Ground Truth for Example Protocol	79
4.6	Complex ground truth transient signature classes	82
4.7	Example protocol with features and ground truth labels	101
4.8	Clustering Evaluation with ARI of 1.0	114
4.9	Clustering Evaluation with ARI of 0.324	114
5.1	Clustering performance for increasing devices in feature space	132
5.2	Clustering performance for increasing devices in feature space comparing ON vs OFF	133
5.3	Clustering performance for single devices	136
5.4	Performance of two similar lights with different testing conditions	139
5.5	Protocol with two devices where one device emits noise (300 features selected)	144
5.6	Device power ranges	149

5.7	Ranking of best features from tsfresh [143]	152
-----	---	-----

Nomenclature

2D Two-dimensional

3D Three-dimensional

ALM Appliance Load Monitoring

ILM Intrusive Load Monitoring

NILM Non-Intrusive Load Monitoring

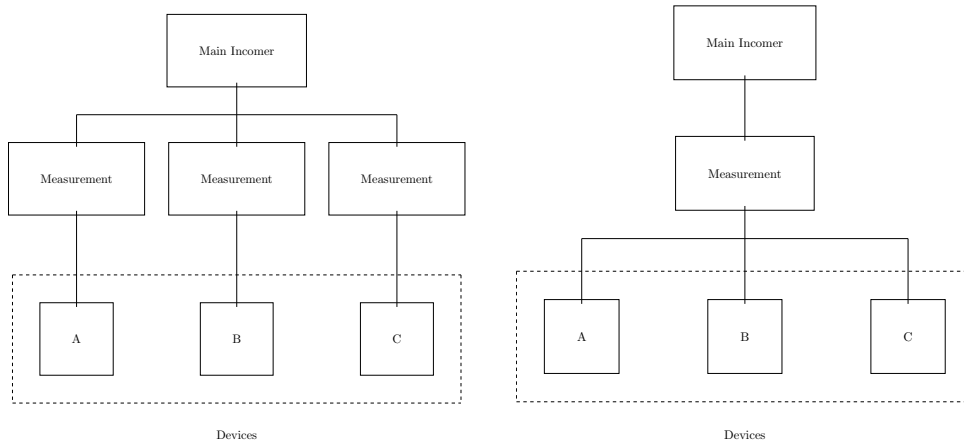
ARI Adjusted Rand Index

Chapter 1

Introduction

Over the course of the last two decades, there has been a large drive towards energy efficiency, better system visibility, and control. However, there is also an increasing need to optimise the generation and consumption of this energy. The ability to optimise and allow for greater control is often paramount to the operation of systems, section 2.1 goes into greater detail on this perspective.

In order to enact this reduction and optimization of energy, monitoring of the entire system is a first requirement. Historically, two methods have been used to accomplish this measurement: via an aggregated measurement (measuring the accumulated power flowing through some main incomer) or via measuring each device within a system and then accumulating these measurements.



(a) Diagram of ILM system

(b) Diagram of NILM system

Figure 1.1: Comparison of ILM and NILM implementations

The two methods illustrated have advantages and pitfalls, these are discussed in

greater detail in section 2.2. This research space—that of Non-Intrusive Load Monitoring (NILM) which is depicted in fig. 1.1b—has shown promise and is increasingly used in obtaining information of devices within systems.

Aggregated load data (generated by this single measurement technique) is often difficult to make use of when looking for efficiency improvements as it does not help in finding high energy intensive devices and devices which consume energy when they are not expected to; background on the work in this space is highlighted in section 2.2.

Solving for this requires the ability to monitor specific appliances such that each device can be characterized by its time of use, energy consumption, and other characteristics¹. Due to the increasing number of appliances that are commonly used in electrical systems it becomes time consuming and inefficient to measure every device on a network individually (see fig. 1.1a). This is often not possible due to the nature of the system, as it would require interfering with the system. This technique is coined Intrusive Load Monitoring (ILM) where individual sensors/monitors are required to be placed on all devices within a system.

The introduction of Non-Intrusive Load Monitoring, and additionally, load disaggregation techniques appear to be one way to solve these problems. Research in this field highlighted the fact that aggregated measurements are increasingly more accurate and provide greater benefits in terms of system installation cost and management (this is discussed in further detail in section 2.2) [1, p. 1870-1871].

Research in this area is made up in large part to perform NILM on AC systems—this is an historical artefact—in that most electrical systems currently in use utilise AC. Over the last few years, many devices used within systems are making use of DC, consequently, there has been a rise of DC grids used in diverse situations [2–4]. Some of the techniques of NILM and load disaggregation are readily applicable to such systems, with a notable exception of distributed DC systems (see section 2.9). There is an increasing adoption of distributed DC systems that make use of power electronic converters in order to construct a grid upon which devices operate.

This is precisely where the techniques of NILM and load disaggregation may be able to provide utility in observing how these distributed DC systems operate and how they may be run better.

¹Other characteristics of devices that may be monitored include systems that can notify users when devices are failing, or operating outside of normal usage.

This background highlights the requirement for research into the viability of making use of NILM techniques in order to determine the operation of a distributed DC system.

The rest of this chapter outlines the research which includes the problem statement and motivation (section 1.1), the research aims (section 1.2), and the research question (section 1.2.1).

Chapter 2 discusses non-intrusive load monitoring, load disaggregation, and DC systems in greater detail.

Chapter 3 further illustrates the research aims, questions, and the system in question as well as the general methodology that is used for this research. Chapter 4 goes into detail on the proposed methodology that is used to accomplish this research. Lastly, chapter 5 outlines how the techniques of NILM applied to this distributed DC system perform against commonly utilized metrics for measuring the performance.

1.1 Problem Statement and Motivation

There is an increasing use of distributed DC systems. Current methods of energy measurement and analysis are inefficient, and often non-existent, making distributed DC systems particularly difficult to monitor. This means that, currently, the methods of observing the operation of devices on distributed DC systems are not scalable or efficient, resulting in situations where designers and engineers do not know how the network is performing.

This points to the need for a system whereby DC networks can be metered and analysed in an efficient manner. In order to create a solution to solve this problem, however, the problem needs to be broken down into its constituent components. The material in sections 1.1.1 to 1.1.3 seek to define these problems that such that, when combined, ideally solve the major issue presented. These then create the ability to outline the research aims and questions outlined in sections 1.2 and 1.2.1.

1.1.1 Measurement Parameter

Common NILM techniques make use of current and voltage measurements on the main incomer of standard systems.² These standard parameters fail to represent the operation of the distributed DC systems. The determination of a parameter that can accurately represent the operation of the system at any given point in time is required for such an implementation to be performed.

This measurement parameter, the grid voltage, observable at all points of the network is believed to be the best identifier of operation of devices within the system. The reasoning on this is discussed in greater detail in section 3.2.1.

1.1.2 The Measurement Technique

There is currently no measurement technique that can be easily applied to a distributed DC system to determine device operation. The nature of the power flow of such a system means that standard NILM techniques cannot be applied in order to determine the operation of the system.

1.1.3 The NILM Implementation and Load Disaggregation

Due to the fact that there is no measurement technique that can be readily applied to a distributed DC system, there is also no standard NILM and load disaggregation technique which has been tested on such a system. This does not, however, imply that current NILM techniques do not work for this system. The different techniques should be evaluated to determine the most performant one in their ability to correctly identify system operations.

Commonly used validation techniques in the field of machine learning and NILM can be applied to the system to create consistent analyses on systems tested/observed. This level of validation is highly important to verify that the system is in fact capable of measuring the operation of a distributed DC system and the devices within.

²There are numerous other parameters that are used; however, these other parameters are often for unique use cases, or fail to adequately represent the operation of the system when tested for detection accuracy.

1.1.4 Distributed DC Grid

In order to create a framework for solving these problems, a distributed DC system is required such that it can be used as a test bench for the research presented. There are a number of different distributed DC network variants that are designed, constructed and tested. This would make the implementation a generalization that can be applied to other distributed DC systems with minimal adjustments. This implies that this system should be capable of operating with other DC systems, not necessarily distributed versions explicitly. These different electrical networks are then used, with a range of diverse devices to test the proposed research methodology.

1.2 Research Aims

The aim of this research is to provide a technique and framework that is capable of detecting, correctly identifying, and distinguishing between the switching of appliances within a distributed DC system. This technique and framework is made up of three sub frameworks that are required in order for the system to function. These include:

- Determine, analyse, and validate the parameter that sufficiently represents the operation of devices within the system (section 1.1.1).
- Form a measurement strategy that is capable of being fed into NILM algorithms such that they are able to correctly identify the operation of devices within the system (section 1.1.2).
- Design and evaluate the NILM algorithm that provides the greatest accuracy for identifying the operation of devices within the system (section 1.1.3).

1.2.1 Research Question

The major research question that this research seeks to answer is:

To what extent can the techniques of NILM be used to reliably and accurately identify the operation of devices within a distributed DC system with a final aim of load disaggregation within the system purely making use of the DC grid voltage of the system?

This can be broken down, in a similar manner to the research aims, into three distinct components which are as follows:

*What parameter best represents the operation of devices within a distributed DC system? What parameter consistently provides a representation (indication) that is independent of the structure of the system?*³

*What is the most applicable measurement strategy that best highlights the switching of devices within a distributed DC system? What are the limitations on the type of devices that can be used?*⁴

Which NILM algorithm and technique is capable of accurately and reliably identifying the device and its switching within a distributed DC system?

³This is to say, what parameter, when used in the field of NILM, can accurately measure and represent the operation of the system, where the structure and combination of that system is not tightly constrained and configured.

⁴This second part looks into the viability detecting the operation of the different device types and the extent to which they can be disaggregated from one another.

Chapter 2

Background

This chapter outlines the current state of the field of Non-Intrusive Load Monitoring (NILM) and load disaggregation; it pays particular attention to energy management and the benefits that it can create. It discusses the different techniques and approaches that are used in a range of different use cases. The chapter then provides a brief introduction to the rising use and integration of DC grids and devices, particularly distributed systems.

2.1 The Energy Perspective

Chapter 1, highlighted the continual technological progress that takes place the world over, the increasing use of energy to power the ways in which humans live. Large parts of this is powered directly with the use of electricity, as the energy is converted from one form to another. Trends in the energy space are increasingly looking towards methods of managing this energy in more efficient ways, understanding how the energy is converted from sun, fossil fuels, wind etc. to electrical energy [5–7]. Therefore, it is clear that better utilization of the current energy resources should be considered to make them last longer and have a greater impact. There are various methods in which this can be performed, these are discussed in sections 2.1.1 and 2.1.2.

2.1.1 Energy Management

The main task of energy management is to reduce the usage, and corresponding costs, of energy in systems without reducing the performance and/or abilities of the system.

High-impact energy management is not always focused around energy reduction strategies, it often involves the intelligent management for the goal of optimizing energy usage. This means that, in a case where a building has an HVAC, then the control—in order to reduce energy consumption—of this system should not reduce the level of comfort of occupants within the building but instead make a trade-off of energy consumption and overall comfort. This reduction in energy usage, and energy optimization can be enacted with a number of different techniques: incentives, intelligent appliance management systems, behavioural changes in appliance usage, etc. [8–10]

The standard method of energy management is to follow an iterative process. This is illustrated in fig. 2.1.

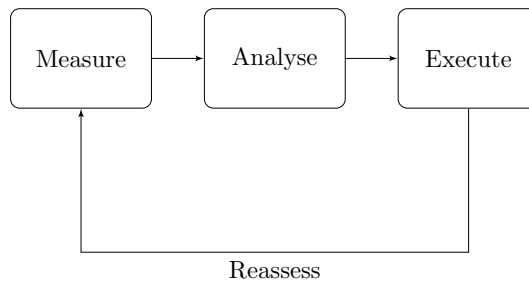


Figure 2.1: Energy Management Process

This illustrates a highly important feature of any energy management system: feedback. The ability for a user to gain feedback based on changes that they make is core to understanding how the system works [11, pp. 2–4].

However, in order to enact this change—of energy management and possibly eventual energy use reduction—an in-depth analysis and knowledge of the underlying system is required.

2.1.2 System Information

The most important conclusion to draw about the iterative process presented in fig. 2.1 is that in order to reduce and optimise energy consumption/utilization, measurement and analysis is required such that the system can be modelled and understood.

A network or system, described in this work, is a collection of devices connected to an energy source where each device within is capable of operating independently of

one another.

This concept has been extensively studied, and it is evident that the more detailed the analysis of the system, the greater the potential benefits to the end-user [11, 12]. This leads to a case where monitoring of systems, particularly the load characteristics of the system, is required for analysis and energy usage reduction and optimization. The analysis and monitoring of these systems creates information which can include individual loads, their energy consumption, as well as other useful information which can include usage patterns.

This is evidenced by the early work of load monitoring that observed different techniques of load monitoring and analysis, this spawned the research space of Non-Intrusive Load Monitoring which is covered in section 2.2 [1, p. 1870-1871].

2.2 Load Monitoring

Load monitoring the practice of determining the operating characteristics of a network of devices (in aggregate) within a power network over time. These characteristics include, but are not limited to:

- electrical characteristics:
 - voltage,
 - current,
 - power,
 - phase angle (for AC devices), and
 - frequency
- time of use, and
- ambient conditions which can include:
 - temperature
 - humidity,
 - sound levels,
 - light levels, etc.

These characteristics—the information that they provide as a whole—allow for a more in-depth analysis and understanding to the operation of the device or network. This information paves the way forward on energy management, and energy reduction efforts to, optimise the networks' operation.

The process involved in load monitoring encompasses the process that is illustrated in fig. 2.1.

An important concept in load monitoring is that efficiencies can only be realised, if inefficiencies are detected. In order to detect the inefficiencies, analysis is performed on the data (information gathered on the operation of the system over time); raw data from the monitoring system is converted into information that provides insight on how the system functions¹.

There are two major techniques which seek to alleviate this problem. These techniques allow for the analysis of entire networks, which then provide the ability to reduce and optimise this overall energy consumption. These techniques are discussed in sections 2.3 and 2.4. These two techniques fall under Appliance Load Monitoring (ALM): monitoring of a network in order to identify how energy is used within a network [13, 14].

These techniques attempt to convert the time series data generated by measurement devices into information that is actionable to allow energy management and/or reduction; this area of research therefore is based on the underlying assumption that there are inefficiencies that are unknown and can be resolved.

2.3 Intrusive Load Monitoring

Intrusive Load Monitoring (ILM) is the technique whereby multiple sensors are distributed to each appliance within a network [15–17]. These sensors measure the electrical characteristics of each device and these measurements are recorded for further analysis². There are different configurations of the sensors within ILM, the sensors can be placed at differing levels within the system, meaning that there may be cases where a sensor can measure multiple devices, this is illustrated in the list

¹Time series data is useful; however, it is often difficult to spot inefficiencies with simplistic line diagrams. This brings about the requirement for multiple visual representations of the data, as well as other analyses that provide insight.

²The measurement device may measure other characteristics, similar to those listed in section 2.2.

below³. However, the common (most accurate and representative) implementation is to place a sensor on each device as this is then able to provide an exact representation of the operation of each device within the system, this is illustrated in fig. 1.1a.

Evidently, as the name suggests, this implementation is inherently intrusive. Intrusive in the method that it monitors each device and the impact on the system. The level of measurement depends on the resolution and accuracy of required information that the system should produce, the list below illustrates the level of detail the different measurement methods enable from mostly aggregate information to individual device level resolution [15].

- Measure sub-systems (e.g. lighting, plugs, HVAC).
- Measure on circuit breaker or plug level (e.g. power strips).
- Measure each device.

2.3.1 ILM Criticism

This technique often entails the use of many sensors (for large systems with many devices). The cost of this number of sensors can often be reduced by using lower cost sensors, however, this then impacts the accuracy that these sensors can achieve [18]. This system requires the implementation of communication between each sensor and some central data storage center.⁴ This increases the complexity of each sensor and the cost of the system once again. It is often the case that loads in the system cannot be interfered with and,⁵ therefore, depending on the type of sensor, it may not be possible to monitor. This means that there are often cases where it is not possible to monitor devices in *isolation*.

The flexibility of this method is low when used in situations where devices are changed out on a frequent basis; simple examples include homes and offices where different devices are brought in and removed regularly, making it increasingly difficult to provide measurement devices ad hoc while maintaining sufficient understanding of what is happening within the network over time.

³This is commonly referred to as submetering, where multiple devices may be measured with a single measurement device. The downside of this is that these submeters can no longer provide information about the individual devices within that sub-metered region.

⁴This is for cases where data should be sent back to this central node periodically, or in real time.

⁵This will often happen in industrial settings.

2.4 Non-Intrusive Load Monitoring

Non-Intrusive Load Monitoring (NILM) is a method whereby a single measurement device is connected to the central node that power for an electrical system flows. This measurement device records a number of electrical parameters, stores this data, and then analyses it in order to detect and characterize devices operating in the system [1].

This approach is named *non-intrusive* based on the fact that only a single measurement device is required to be connected to the system, this device does not interfere with the electrical system and does not require modification (of the system) to perform these measurements. Contrasting this technique to ILM, it has a simpler installation procedure as access is not required to each device within the system.

To illustrate the technique, fig. 1.1b illustrates a system that contains three individual (and independent) devices, each device in operation receives energy through the central channel, this is where the measurement device is placed. The simplest method to acquire information from the system is to measure the current or power at this point, this provides an aggregate value which indicates the total power consumed by devices within the system over time. Contrasting this against the ILM method, this technique requires further analysis of the information to understand what individual devices do at any given point in time; the analysis, and ability to extract this information is known as load disaggregation, see section 2.6 on what this entails. The standard technique of NILM, in generating this aggregated view of the system, is not conducive to finding the inefficiencies inherent to load monitoring; this implies that the combination of NILM and load disaggregation are often required to create a more efficient system.

Many NILM applications follow a similar process in their implementations which are: data acquisition, feature extraction from the measured data, followed by classification of appliances from the features, this process—which is a generalization—of the technique of NILM is illustrated in fig. 2.2 [16, 19].

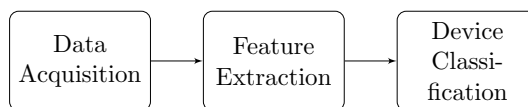


Figure 2.2: Generic NILM Process [16, 19]

2.4.1 NILM Criticism

This method is often considered to be lower in cost when compared to most ILM implementations, even though the cost of the single measurement device in this method is likely to have a higher cost and more complex than the individual measurement device in the ILM technique. Due to the nature of the NILM algorithm, the measurement device is required to be relatively accurate, this brings about the high cost of the device. This implementation relies heavily on the disaggregation capability in order to accurately function. This is where the ILM implementation performs better as it often does not need to perform any disaggregation⁶.

Due to the coupling of NILM and load disaggregation, and the fact that the art of load disaggregation often involves complex algorithms, there are situations where it is simply not possible to define the operation of devices within a system. This is seen in cases where the observed system is large and has many devices that operate within, the task of disaggregation becomes complex and often times the accuracy is sub-optimal⁷⁸.

Due to the ubiquitous nature of AC systems worldwide, the majority of the research in the NILM field is geared towards applications in AC systems. A further limitation of most NILM techniques is that they are applied to systems where there is a single source; this means that the majority of NILM applications only function for centralized systems where the network is structured such that multiple devices are fed power from a single node. There is a growing number of NILM techniques that can function on multiple phase systems which seek to provide the benefits of NILM to larger energy users [20].

This limitation to some AC systems is also observed with systems that operate with DC. This is also largely due to the historically limited systems that are DC based. There is also, only recently, a few systems that have attempted to apply NILM techniques to DC systems [21–24]. However, this is limited to single source systems where the system has a single point to measure the aggregated power draw of the devices.

⁶The fact that there is an independent measurement for each device means that the information is already disaggregated.

⁷This is partly why there is an increase in the research in this area, the advances introduced by the decreasing cost of computing tools, as well as the improvement of the algorithms, means that this technique is getting further adoption.

⁸The performance of these algorithms forms large parts of the research; this is because it is important to quantify the performance of NILM and load disaggregation on systems, this is discussed in greater detail in section 2.8.3.1

2.5 ILM and NILM Applications

The utility of NILM and ILM discussed in this work does not cover all the benefits that these techniques provide. Many tools developed by companies are providing users with the ability to remotely monitor the operation of devices, this increases the utility of the tools far beyond the simple analysis of energy consumption; the tools provide the ability to create energy audits of homes or offices, detect when devices fail [21, 25], demand side management [26], as well as a technique of security monitoring⁹ to name a few [15].

2.6 Load Disaggregation

Load disaggregation—a technique that is often highly coupled with NILM—is the process whereby aggregated load data from the operation of a network is analysed to detail the operation characteristics of devices within the electrical network. The process often makes use of the electrical and other characteristics listed in section 2.2 in order to perform the disaggregation process. This disaggregation process can analyse and illustrate the operating characteristics of the devices within the system such as: energy usage, operating times, appliance signatures/features, load models, etc.

As the disaggregation performs an algorithm on the data from a number of accumulated devices within a network, it is pertinent to explore the different types of devices that can be in the network. These four main categories of devices are discussed in section 2.7. These devices create two unique methods of performing load disaggregation, namely event-based and eventless; in other words, the two techniques are either to make use of the transients that occur from the changing of states of devices within the network, or to make use of steady state features of the network. These two techniques and their assumptions are outlined in sections 2.6.1 and 2.6.1.1¹⁰.

⁹A simple example is where a location is remotely monitored, and it can detect when devices operate when they are not supposed to, for example a security light turning on [1, p. 1872].

¹⁰Research in this field has made use of both approaches in the load disaggregation task as both of these approaches often provide unique information that one will not provide in isolation [27–29].

2.6.1 State-Based Approach

State-based approaches rely on the exact characterization of each device such that if these devices are combined, their combination during operation results in a specific state [30]. Models are used for each device within the network so that any combination of devices in any of their operating states can be utilized to match up the systems operating state with the combination of the states. The system relies on each appliance containing a number of states to operate within (this is discussed in greater detail in section 2.7). The combination of all these states allows for the system to predict which appliances are operating. The states, and how they are determined is discussed in section 2.7.

This technique does not rely on an event detector as it monitors the electrical (and other) characteristics of the network over time and uses this information to make inferences. This approach often involves the use of optimization approaches; additionally, the field has seen a rise of deep learning to solve the disaggregation problem [31–36].

A simple example of a state-based approach is provided in section 2.8.2 where three different devices operate in a network over a period of time.

2.6.1.1 Event-Based Approaches

A second approach to load disaggregation is with the use of events; this approach relies on the fact that, upon switching, each device creates a transient within the system [30]. Events are defined as the response (electrical) of the network brought about by the changing state of a device on the network over a period of time. This period of time is dependant on the network, the device, and how they both interact and the time it takes for the system to reach its steady state again. The transients are observed in the voltage, current, and other emitted electrical characteristics, different techniques use combinations of these characteristics to perform the disaggregation process [16]. These events are analysed and the corresponding device (and its state transition) to the event is provided. This means that the response observed should be characterized such that it can be associated with a device and that devices state change.

This approach also relies on devices having multiple states (see section 2.7); it is the transient that is the important signature which holds the information. The

transient contains unique features that can be used to disambiguate the different devices within the system and their state changes. This is also broadly referred to as a pattern recognition approach [37].

This technique of load disaggregation is required to include an event detector; the event detectors task is to determine the moment in time when an event takes place—when a device changes state—and to acquire the relevant electrical measurements for that period of time. This tool is discussed in section 2.6.2.

2.6.2 Event Detection

Touched upon in section 2.6.1.1, the technique of load disaggregation using the event-based approach requires the use of an event detector. The event detector, a tool that monitors the electrical characteristics of the network over time, seeks to find points in time where there are transients that indicate that a device may be operating. This tool then starts measuring and saving this information which is sent through to the next stages of the load disaggregation algorithm where the *transient signature* is analysed and matched up with a device.

There are a number of designs for event detectors, they are designed to be capable of picking up changes on the network such as a sharp increase or decrease in the power level, a change in the operating voltage, or a change in the spectral envelope observed [38]. The event detector is often provided with a threshold upon which to determine whether a signal is a transient or simply noise apparent within the system; a simple example of this is when a specified voltage fluctuation over a set period of time ($\delta V/s$) surpasses a specified threshold, then an event has occurred. There are many such considerations to make when designing event detectors and there is growing research in optimizing event detectors to be used within the NILM field [26, 38–40]. The event detector tool is also utilized in other fields, it is particularly useful in analysis and fault detection of electrical machinery [25, 41–43].

2.7 Device Types, Signatures, and Features

There are two major categories of features that can be used to distinguish the different devices within a network. These are the steady state features and the transient features. Devices within the system, due to their internal construction and the ways in which they interact with the network upon operation, produce unique signatures.

2.7.1 Device Types

To understand these two types of features, it is important to understand and model the devices within the system to find the best methodology for the context. This first section outlines the different major categories that devices are grouped into. These categories describe the general ways in which the devices function over time.

The devices are grouped into four major types listed below:

- **I:** two states,
- **II:** finite state machine (FSM),
- **III:** continuously variable, and
- **IV:** permanently energized devices.

I: These types of devices have two states, implying that they can either be ON or OFF. Consequently, these devices have two state transitions: $\text{ON} \rightarrow \text{OFF}$, and $\text{OFF} \rightarrow \text{ON}$. A simple example of such a device would be a light.

II: There are devices which can operate in different modes or states. This implies that a device has three or more possible states to operate in. This means that these devices can have multiple different/unique ON states¹¹. Two examples of these types of devices are a fan with multiple speed settings and a hairdryer with hot and cold settings.

III: These types of devices do not have specific states, they have varying electrical characteristics as they operate. These devices have the ability to continuously vary over time. An example of this type of device would be speakers, where the power output would depend on the volume and the sound level output over time. Another example would be light with a dimmer switch where the output power would change depending on the level set, this device's power output changes with user input.

IV: There are, increasingly, more devices which are permanently powered. These devices are then observed in the disaggregated load as some constant load level. A simple example of such a device would be a home security system where it is required to be powered on permanently.

¹¹These devices are often seen as multiple type I devices internally, and the different states represent the different states of the different internal type I devices.

Figure 2.3 illustrates typical power draw characteristics of these types of devices if they were to change states or operation modes over a period of time.

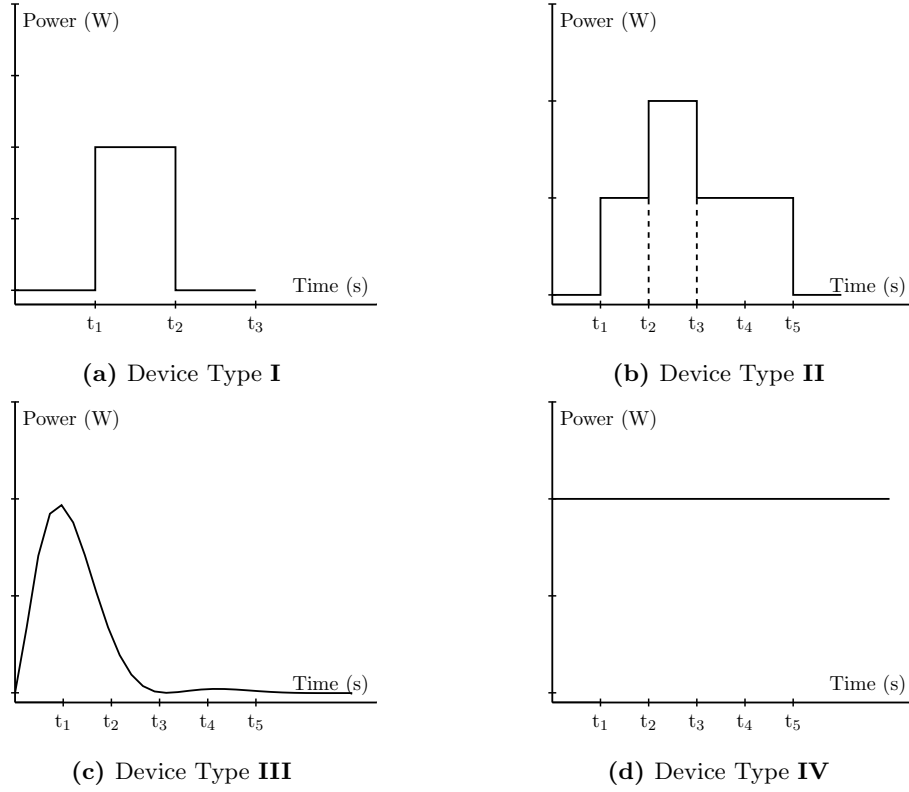


Figure 2.3: Device type waveform examples described in section 2.7.1

The first two device types can be represented by state diagrams, type I devices simply have ON and OFF states, so the transitions involve switching between these two operational modes. Type II devices depend on the number of states as well as the switching type that the devices may utilize, this means that this second device type can be modelled in a number of ways, fig. 2.4 illustrate these two device types and the possible transitions. These diagrams do not represent all of the possible state transition types which are possible with type II devices; instead this illustrates that in order to characterize these devices, their internal construction and functionality are important to understand the underlying state transition system [1, p. 1874-1875]. This illustrates the fact that type II devices have more state transition types, increasing with the number of possible operational states.

Device types III and IV pose an added complexity to the disaggregation process, this is largely due to the fact that they are particularly difficult to observe with event-based techniques due to their slow, continuously time varying nature for type III devices, not to mention the fact that type IV devices do not switch. These types

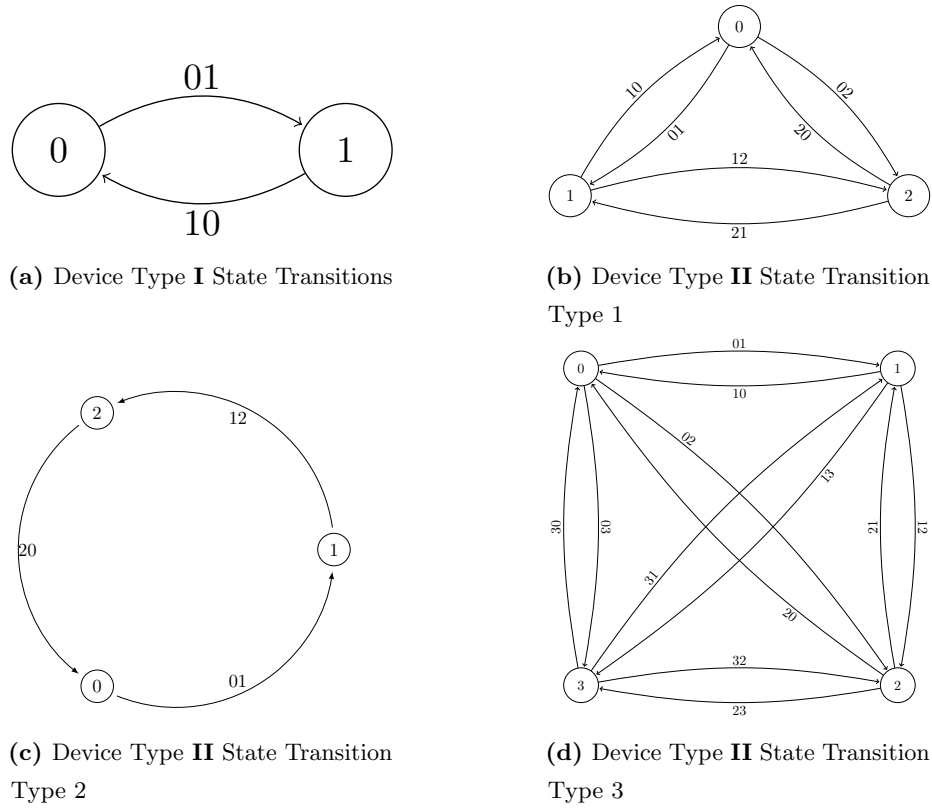


Figure 2.4: Device State Transition Types [1, p. 1874-1875]

of devices are not as prevalent in electrical systems, the majority of devices available are made up of the first two.

2.7.2 Device Features

The devices discussed in section 2.7.1 illustrate that different devices have different operating types. However, there are further characteristics of devices within these categories that allow for highly detailed models of each device to be created. This is based on the fact that all devices are designed differently, from their functionality, to the internal electronic (and mechanical) components that are used to construct them. Furthermore, these small differences in the construction of devices, coupled with the physical installation of devices within the network—resulting in further variation in resistance, inductance, and capacitance—means that the response observed on the grid from different devices is unique [1].

Research in this field has often involved observing these signatures—also known as features—that the operation of devices present on the network. A few types of features that are commonly utilized in this work is illustrated in fig. 2.5. This

highlights the fact that different load disaggregation techniques (event-based or state-based) will provide the ability to make use of different feature types.

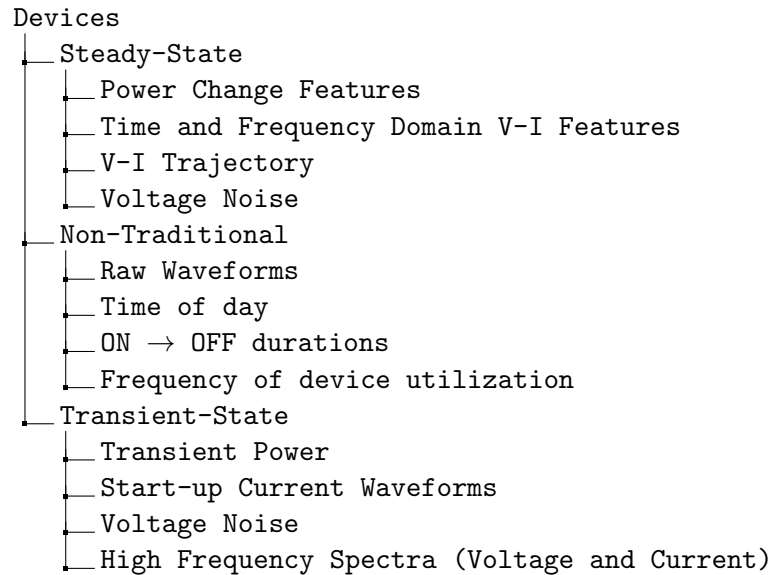


Figure 2.5: Features Breakdown [14, p. 12]

This highlights the fact that the two major feature categories utilized in this field, it also illustrates the fact that the ability to utilize both event-based and state-based techniques yields a greater diversity of features which make the disaggregation process much simpler.

Steady state features: The features extracted in this method make use of steady state information. These features include: Real Power (W), Reactive Power (Q), Voltage (V), and Current (I). The power features are often extracted from the voltage and current of the system. These features are values that reach a steady state phase within a period of time, and as such allow the algorithm to detect which state the system operating over time.

Transient state features: These features are reliant on the state transitions that occur within the system; particularly those generated by device types I and II as these have distinct transitions which yield rich event-based features. The transient state features also look at the Real Power (W), Reactive Power (Q), Voltage (V), and Current (I) of the system in these transient time periods. However, there are other characteristics that are measurable within the system; device switching features contain a wealth of information in the frequency spectrum. The devices often also emit electromagnetic interference (EMI) which can be detected and assist in classifying the device and the state transition [44].

There are additional factors that augment the signatures of devices within the system. A short list of these augmentations can include:

- temperature changes,
- other environmental changes (humidity, air pressure, etc.), and
- age of device.

These factors will often influence the steady- and transient-state features of the device. The effects on the operation of the device are often minimal and therefore can—in many cases—be neglected. These factors will, in some cases, augment the operation of the device while it is running within the system, this can happen within a normal operating period. Other factors will change over longer periods of time. Components within the devices may age and degrade, causing a shift in the operating characteristics.

2.8 Machine Learning Techniques and Inference

The information sent from the measurement device is recorded as time series data. Regardless of whether the information recorded is a continuous measurement of the operation of the system, or at events occurring within the system, the information undergoes pre-processing before it is analysed, this is to perform tasks such as the filtering of noise, remove unwanted spikes that misrepresent the original data, as well as feature extraction and feature normalization, etc. Once this pre-processing stage has taken place, a number of features are selected from the data sets; this feature filtering is used here to only utilize the features which will best represent the system operation. Finally, learning and inference is applied to the data in order to complete the process. This is the stage at which the data is finally transformed to information and the system can be analysed at a deeper level. These three stages can be illustrated as follows:

- Feature transformation (data cleansing, augmenting and operations applied to the data (e.g. FFT, rise times (T_r), settling times (T_s), etc.)).
- Feature selection (how the features are combined and factored in is important).
- Learning (this is where the machine learning algorithms take over).

The learning stage is grouped into two major categories,¹² namely supervised and unsupervised learning. These two categories are discussed in section 2.8.1.

2.8.1 Supervised and Unsupervised Learning

The two categories of algorithms/techniques for taking the feature data and transforming it into information are differentiated into two distinct categories, supervised and unsupervised machine learning.

Supervised machine learning makes use of data that is *pre-labelled* so that it is able to assign devices to specific classes where the classes represent devices and their switching state. This means that when a device is set to operate—the transients that it generates—or the features that it generates (if looking at a state-based approach) while operating, are assigned (*labelled*) for that specific device. This allows the system to create assignments from signatures and match it up to a device. The supervised case always includes a training stage such that the classifier is able to learn the features that correspond to the different devices and their state changes.

Unsupervised learning, on the other hand, makes use of unlabelled data; specifically, it takes the measurements, generates features from the data, and then attempts to group this data into a number of classes (much like the supervised approach). This is also known as clustering. This goal of this clustering is to create a system whereby the different points/data are mapped into some feature space and the groupings of these clusters represent the different devices and their state transitions.

There is a further category of semi-supervised machine learning where combinations of the two techniques are utilized such that the requirement for labelled training data is not as important and user interaction with the system is used to assist in the disaggregation, labelled data is created which has models for generic devices which can be applied in most circumstances [46].

2.8.1.1 Supervised Learning

Supervised machine learning techniques within this field fall into two broad categories, namely pattern recognition and optimization; these two categories are based upon

¹²The third, semi-supervised learning is separate, however, it can be grouped with the unsupervised methods [45].

the disaggregation approach of either event- or state-based approaches being utilized respectively [14, 47].

Optimization Approach: The optimization approach involves loading a database of known device features into the system, the algorithm then observes the current state of the system and attempts to compare the extracted features of the real world to a combination of possible device configurations within the database by attempting to reduce the error between the two [14]. Commonly utilized techniques to solve this problem use some of the following: integer programming, genetic algorithms, and least residue algorithm [14, 48, 49].

A simple example of the optimization approach is illustrated with section 2.8.2 where it is highlighted that even with a combination of a few devices, the optimization problem quickly becomes complex and computationally expensive to solve as the algorithm has an increasingly large combination of devices and operational states that may occur.

Further issues with these approaches is that due to their reliance on a device signature database, any devices that are added to the network after this process will cause the system to fail or reduce its disaggregation performance.

Pattern Recognition Approach: This approach makes up a large part of the research in this field, this is partly due to the recent progress in the field of machine learning for supervised methods; additionally, it has proved to offer greater performance than the optimization approaches have steep performance roll-offs with increasing numbers of devices within networks. This approach has the ability to make use of more information to assist in the disaggregation approach, some of these are listed in fig. 2.5 where the non-traditional features enhance the overall performance [14, 50, 51]. Some of the commonly utilized techniques to accomplish this approach include: Hidden Markov Models (HMM), Artificial Neural Networks (ANN), Support Vector Machines (SVM) [14]. This approach, much like the optimization approach often fails to perform when unknown loads (not included in the labelled training sets) are introduced.

2.8.1.2 Unsupervised Learning

Unsupervised machine learning processes are utilized within this field in an attempt to overcome the issues of requiring extensive pre-training and labelling processes where all devices within the system are characterized. This approach does not therefore

require *a-priori* information (labelled data); this approach commonly makes use of clustering techniques where events are transformed into feature vectors that can be mapped into a feature space, these points are then clustered with the proposal being that similar events are grouped into the same clusters/classes as each other. This approach assumes that state transitions of individual devices are unique and repeatable over multiple events.

Figure 2.6 illustrates a situation where three devices (all of type I) are connected to a system and three of their features are measured. The figure represents these different fictional features as Feature x , y , and z , there are many features that these techniques make use of to assist in this process¹³.

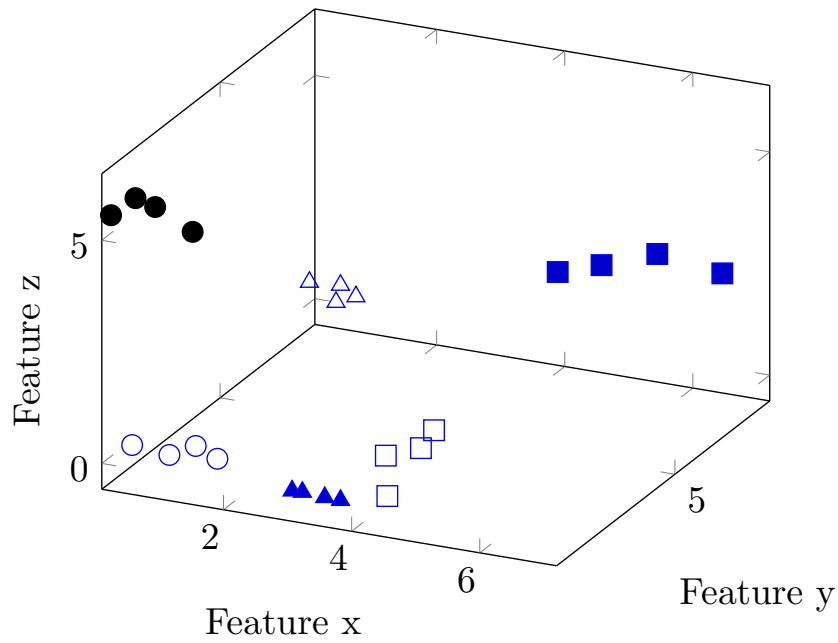


Figure 2.6: Three device clustering

This method is particularly relevant in cases where devices unknown to the system are used; in practice, this means that the system will be set to run for a period of time, after which a user will label the clusters based on underlying knowledge of the system. This can be done such that the user knows when specific appliances are turned ON and OFF, they can then find the events in the clustering space that occurred at that time, and include labels to those clusters. This allows for the ability to add new devices without re-training the system and gathering the signatures of this new device.

¹³This example illustrates a case with only three features; however, any number of features can be used to assist providing more information upon which to accomplish the disaggregation algorithm.

This approach has a number of downsides which involve the problems associated with the diversity of devices within the system: smaller devices produce *small transients* on the network which means that sometimes overlapping clusters are produced for multiple devices. A further complication is when replica devices are placed in the network, a simple example of this is households that have multiple lights that are all the same brand and model number. These devices often have similar event signatures which are difficult to disaggregate [14, 47]. These problems are touched upon in the literature where suggestions are made that it is possible to distinguish these devices from one another, however high-resolution measurements and high sampling rates are required to extract the small differences [39].

2.8.2 NILM and Load Disaggregation Example

This section details a simple procedure that makes use of the operating power of a number of devices within a network to illustrate how an optimization approach may attempt to run a load disaggregation algorithm. Figure 1.1 represents the configuration of a system of three devices.

The physical setup illustrated in fig. 1.1a is used to determine the exact operating characteristics of each device operating within the system using isolated measurements for each device. The measurement of the power draw for each device in each operating state is recorded, this is done so as to keep a record of the type of device and the signatures that it introduces upon operation. This simple example makes use of two type I devices (A and B) and a single type II device (C), which has three operating modes.

Once the devices have been characterized, the measurement placement can be moved to what is illustrated in fig. 1.1b. The measurement device, placed on the main incomer of the system, takes measurements of the power drawn from the loads. This power measurement will be the accumulated power (aggregated power) of the devices running at any given time.

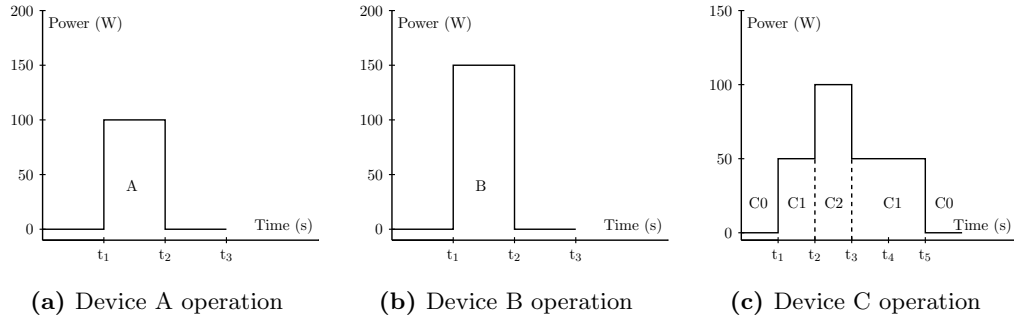
To illustrate the operation of this system over a period of time, an example load profile is constructed and is represented via table 2.1.

Where the power given for each device within the system in their different operational states are described by figs. 2.7a to 2.7c.

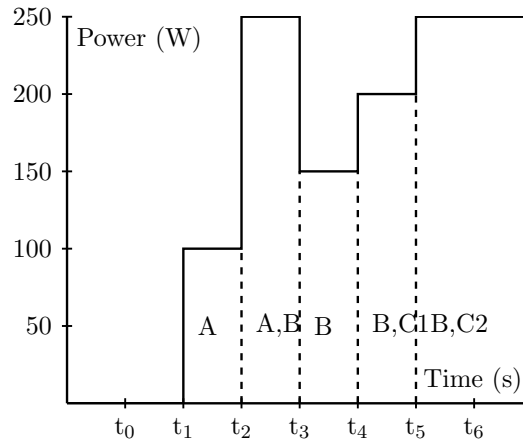
Based on the measurement point illustrated in fig. 1.1b which represents a NILM

Table 2.1: Example Load Profile of three devices (A, B, and C)

Time Interval	Device(s) Running
$t_0 - t_1$	-
$t_1 - t_2$	A
$t_2 - t_3$	A, B
$t_3 - t_4$	B
$t_4 - t_5$	B, C1
$t_5 - t_6$	B, C2

**Figure 2.7:** Example Devices

measurement the following waveform/profile is expected and illustrated in fig. 2.8. This is purely based on the accumulated power that each device produces at any given time. This means that the system can be modelled, as the characteristics of each load are known. A list of possible operating points/states can be obtained with this knowledge, each of these operating points represents a power (W). This model is illustrated with fig. 2.9. This illustrates the different possibilities that each device can operate in, and the corresponding power that is expected in each of those states.

**Figure 2.8:** Example Measurement

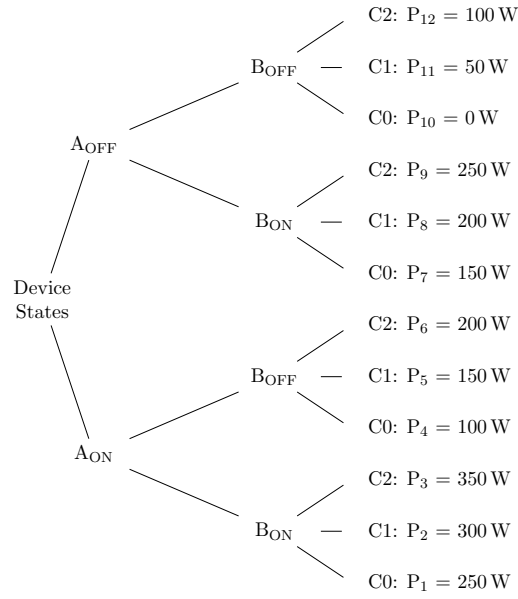


Figure 2.9: Possible power states with two devices

This process works for both AC and DC systems—which use centralized sources—as it simply measures the current (and therefore power) flowing through the main incomer to arrive at an aggregated power consumption. This is a simplistic approach towards the disaggregation process and cannot be used once the system structure is changed; a simple example is if another device is added, the database does not account for any other devices, as such the ability for the algorithm to correctly determine the state of the system at any point in time is no longer good.

This technique, even in this simple example, it is evident that the system may fail to correctly predict the operational state of the system in certain cases, this is observed in cases where the power level is at 250 W where there are two possible combinations that the system state can be in. This is also true for the power levels of: 100 W, 50 W, and 200 W which all have more than one possibility.

This simplistic example illustrates some of the core principles used in NILM research, research in this field has introduced increasingly complex solutions to this problem set. This example does not include a number of other factors and *real-world* factors that are part of the NILM process, as such it does not factor in external noise, slight variations of device power draw, etc. This example also highlights the fact that, as discussed in section 2.4.1, most of the techniques of NILM only apply to systems with a central source node.

2.8.3 The Utility of NILM

This section has multiple subsections which seek to outline a number of extra considerations for NILM.

2.8.3.1 NILM Performance

The growth of NILM research has created a need to be able to compare different techniques against one another, this has led to a common set of metrics that most NILM researchers make use of such that they can compare different algorithms. These metrics, in addition to commonly accepted datasets that the community can make use of (see section 2.8.3.3) to directly compare performance of different algorithms on the same underlying data sets. These datasets, due to the fact that they contain the labelled data, can be used to analyse the performance of NILM algorithms easily and accurately.

There are three commonly utilized accuracy metrics that can readily characterize the performance of a NILM algorithm: detection accuracy, disaggregation accuracy, and overall accuracy [49]. This relies on a number of definitions to evaluate the different accuracies which starts with defining the total number of detected events (N_{detected}) in eq. (2.1).

$$N_{\text{detected}} = N_{\text{Groundtruth}} + N_{\text{IncorrectDetections}} - N_{\text{MissedDetections}} \quad (2.1)$$

Detection Accuracy: How well can the NILM technique able to detect the switching (changing of states) of devices within the network over time. How this functions in practice is a measure of the ability of the event detector; it is attempting to pick up on type 1 errors ($N_{\text{IncorrectDetections}}$, the event detector believes it detected an event when no device switched) and type 2 errors ($N_{\text{MissedDetections}}$, the event detector did not pick up that an event took place). The detection accuracy ($\eta_{\text{DetectionAccuracy}}$) is defined in eq. (2.2).

$$\eta_{\text{DetectionAccuracy}} = \frac{N_{\text{NILMDetectedEvents}}}{N_{\text{detected}}} \quad (2.2)$$

Disaggregation Accuracy: How well can the NILM algorithm, with the detected events, accurately distinguish what each device is doing at each event, and each point

in time. The disaggregation accuracy ($\eta_{\text{Disaggregation}}$) is defined in eq. (2.3).

$$\eta_{\text{Disaggregation}} = \frac{N_{\text{NILMDetectedEvents}}}{N_{\text{detected}} - N_{\text{IncorrectDetections}}} \quad (2.3)$$

Overall Accuracy: Combining the two metrics above, how well can the algorithm relay the exact functioning of the system. The overall accuracy (η_{Overall}) is defined by eq. (2.4).

$$\eta_{\text{Overall}} = \frac{N_{\text{NILMDetectedEvents}}}{N_{\text{Groundtruth}}} = \frac{N_{\text{NILMDetectedEvents}}}{N_{\text{detected}} - N_{\text{IncorrectDetections}} + N_{\text{MissedDetections}}} \quad (2.4)$$

Some additional metrics—somewhat linked to those above—that are commonly used are the True/False Positive Rates, precision, recall, and F-score.

True/False Positive Rates: The True Positive Rate (TPR, see eq. (2.5)) and False Positive Rate (FPR, see eq. (2.6)) are both defined by using the True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN) [47]. Intuitively, the best-case for a NILM algorithm is a TPR of one, and an FPR of zero.

$$TPR = \frac{TP}{TP + FN} \quad (2.5)$$

$$FPR = \frac{FP}{FP + TN} \quad (2.6)$$

Precision and Recall: These two metrics are useful in defining the F-score, they are defined by eqs. (2.7) and (2.8).

$$Precision = \frac{TP}{TP + FP} \quad (2.7)$$

$$Recall = \frac{TP}{TP + FN} \quad (2.8)$$

F-score: This measure, also known as the F-measure, is the harmonic mean of the precision and recall, it is defined by eq. (2.9). This measure is useful as an overall classification accuracy of NILM algorithms [52].

$$F - score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (2.9)$$

Further than this, there are metrics for defining the accuracy of specific devices within the system, this is often useful to provide insight into devices that may be particularly difficult to disaggregate. Linked to this metric is the ability to compare features of multiple appliances against one another, this is particularly useful when comparing two similar devices as they may confound the NILM algorithm if their features are highly correlated [49, p. 558]. There are also a range of metrics that look further into the utility of NILM, specifically the ability of NILM algorithms to evaluate the energy utilization of systems [53].

The field of NILM is still in the early stages of development, meaning that there is still a wide range of accepted accuracy metrics and different researchers use different validation methods. This background offers some of the commonly utilized techniques used in the field, further accuracy metrics found in the literature can be found from the following references [14, 30, 47, 49, 52–58].

2.8.3.2 NILM Technical Considerations

The exact implementation of NILM varies by technique and use case. The sampling rates, vertical resolutions required, event measurement types (continuous measurement or only specific event measurements), what electrical and physical parameters are measured, and others, are all considerations that affect how effective the NILM algorithm performs.

These details are not covered here; however, they are pertinent to this research and are discussed in greater detail in chapters 3 and 4. These decisions affect many different factors of the design of a NILM implementation and some particularly relevant research on these can be found in the literature [1, 14, 59].

2.8.3.3 Disaggregation Datasets

Section 2.8.3.1 touched on the fact that research in this field is often required to be readily compared against other NILM implementations. This means that the metrics utilized should be able to be applied to the same underlying datasets such that good comparisons can be drawn. There are a number of public (and private) datasets that researchers have created to use within this field, the literature covers these in detail [58–65]. A number of these are also tracked by the community [66].

These are not discussed in detail here as section 3.3.1 highlights the fact that currently there are no datasets for DC systems and this research sets out to create a tailored version of a dataset to carry out the research.

2.9 DC Technologies

Due to the historical nature of electrical transmission and distribution, the majority of systems make use of AC for providing power to electrical networks. Section 2.4.1 touched upon the fact that, due to this historical phenomenon, much of the research in the NILM field has worked on AC systems, and therefore AC devices. The past few decades, with the increased use of transistor-based electronics (silicon in general), means that many devices within the electrical networks make use of DC to operate. These devices often make use of converters which take the AC power from the common supply and transform it into DC that is then used internally by the devices [67]. This means that it is increasingly becoming the case that most devices within a network do not natively use the AC that is utilized by most distribution systems.

This results in networks where a non-insignificant amount of energy is spent in converting this AC into DC. Consequently, designers have observed this as a potential point of efficiency increase: simply create a network that utilizes DC from the beginning and allow the devices to be directly powered from this network.

Coupled with the rise of increasingly complex and lower cost battery technologies as well as solar PV that is easily attached and interfaced to such systems, DC networks are beginning to offer simple and cost-effective alternatives to the standard AC system. This brings rise to the current gaining trend of DC networks that are capable of utilizing multiple types of generation elements at the same time to create this DC *grid* that devices operate under.

This recent adoption of DC *distributed generation* is starting to accelerate. Electronic power converters are used in these systems where the energy is transformed from one format to another; a simple example here is a situation where solar PV panels are used to provide power to a DC system, they are connection through a power electronic converter which takes the panels electrical energy and converts it to support the grid¹⁴. The technologies for running the entire system on a DC bus are becoming commonplace [2, 3, 68–71]. The major components of the system entail some energy source (generation, storage), an interfacing mechanism (the power electronic converters), and then loads.

The power conversion mechanisms are most often in the form of DC-DC power converter electronics which are capable of converting the energy from the source through to some bus, and then again from this bus to a load [72, 73]. This design does not rule out the use of AC devices as inverters are highly capable when AC output is required for devices that natively utilize AC.

This design enables the creation of systems where the sources are highly distributed throughout the network. The diversity of the sources is open to the designer, implying that they are now able to design the combination of sources to match up with the load demanded by the network which represent a least-cost implementation. This ad-hoc capability creates a highly flexible system that is often times more efficient than comparable AC systems.

¹⁴There are numerous options for designers to utilize here: batteries, solar PV panels, wind turbines, etc. This also leaves the option of AC open to the designer. They are still able to utilize an AC source and simply introduce the power electronic converter to interface through to the grid.

Chapter 3

Problem

This chapter outlines the problem and lack of research in the field of NILM and load disaggregation in distributed DC systems. The chapter discusses, in greater detail, the concepts introduced in chapter 1 where the research is contextualized. This chapter also briefly outlines the methodology for this research and discusses how this methodology will be validated to ensure that the results provide meaningful and relevant results.

3.1 Distributed DC and NILM

Due to the historical popularity of AC, there has been minimal work on the research of energy management and monitoring for DC systems, particularly in the field of NILM [22, 23, 74]. The tools and techniques developed for NILM and load disaggregation are tailored for AC systems and can be adapted for use in DC systems, evidenced by the example given in section 2.8.2.

These implementations are not, however, particularly well-suited to use in DC systems and are unable to function in cases where the system contains multiple sources, i.e. a distributed network. The presence of these multiple sources results in a case where a single point of measurement is unable to represent the operation of the system. The diverse range of possible structures of these distributed networks results in situations where it is not easy to place basic current sensors to gain an understanding of the operation of all devices within the system as the current flow (power flow) is no longer centralized¹.

¹This is a central assumption of much of the research in NILM.

The penetration of distributed DC sources in systems is increasing, as discussed in section 2.9, which implies a gap in the knowledge on monitoring the operation of such systems. This implies that in this field of distributed DC systems, there is currently no capability to apply NILM and load disaggregation in order to determine the operation of the system at any given point in time.

3.2 The Theory and Operation of the System

A system that is capable of applying NILM techniques on a distributed DC system should, first and foremost, be capable of measuring a parameter of the system that is indicative of the operation of the system. Common parameters that are used by NILM techniques are discussed in section 2.2 where the effectiveness of the different parameters are outlined. Relevant to this is the large amount of research that uses current (and hence the power) to operate the NILM system, which means that there is little research into other measurement parameters.

The use of a current measurement cannot be used in this system as it would mean placing a measurement device on each source (illustrated in fig. 3.1) which would make the measurement system intrusive (ILM); not to mention the difficulty of keeping track of the different sources, which all require their own current sensor. This means that another parameter is required which indicates the operation of the system, section 3.2.1 appears to be a significant and reliable parameter which can be used.

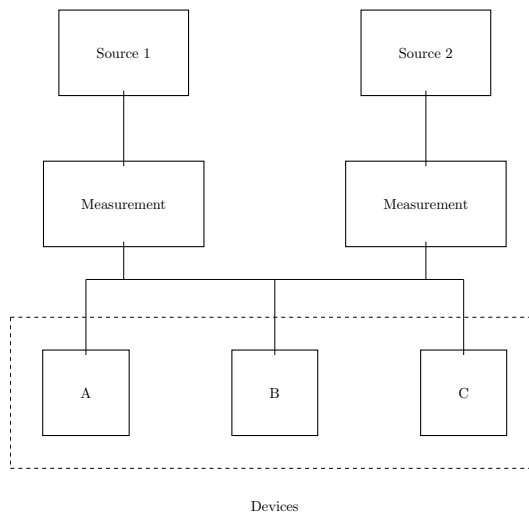


Figure 3.1: ILM on a distributed generation system

3.2.1 DC Bus Voltage

The parameter which can be observed across the entire system is that of the bus voltage, this is always the case because any device connected, needs access to the two terminals which provide the power. Measuring any point on the red indicated paths in fig. 3.2 provides the ability to detect the switching of appliances within the system.

Any device within the system, as it is switched, will disrupt the steady-state operation of the system, the bus voltage then deviates from its set position—that the distributed sources control—until it is able to stabilize to a new operating point.

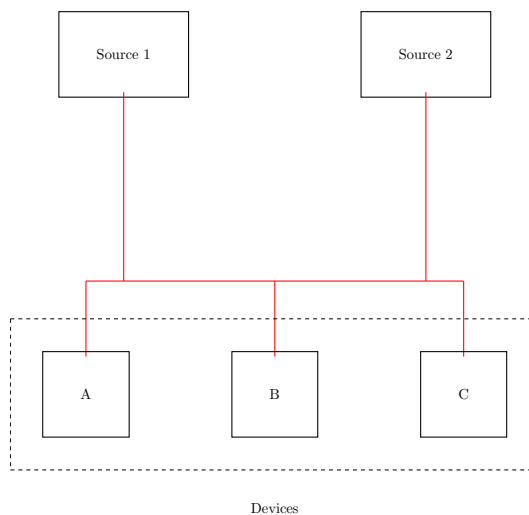


Figure 3.2: NILM on a distributed generation system

The measurement of the voltage (and current) transients within the system at these times represents the switching of the devices. The transients measured are unique to the device, and unique to the state that the device transitions from and to. The transients represented by each device are dependant on the construction of the device as well as the placement within the network: this implies that even small differences between devices results in differing transients exhibited on the grid voltage. This is the key characteristic that the system relies upon in order to distinguish between different events.

Due to the construction of a distributed system as illustrated in fig. 3.2, it is likely that no measurement point along the indicated paths (in red) will result the same transients being detected with the same system. This is due to the fact that the transmission components that are used to maintain the grid voltage are not ideal, they

have resistance, inductance, and capacitance which affect the measurements along different parts of the system. This is also due to the locality that the measurement point is in with respect to the generation, storage and load elements [72, p. 80].

This is due to the fact that these transients have limited impact based on the locality of the device switching to the measurement point based on this non-ideal nature of the transmission elements. The further away the measurement point in the distributed system is from the switching device, the smaller the impact measured.

3.2.2 The NILM Implementation

The structure of the NILM implementation requires that it be capable of disaggregating the loads (and their states) with no prior knowledge of the system or the devices contained within. This is due to the fact above whereby different measurement points result in different transients being measured for the same device.

This uniquely suits the capabilities provided by an unsupervised learning technique that makes use of the event-based approach to disaggregate the loads within the system based on their transients.

The first step in the process is from the event-based approach: entailing the measurement of the switching transients in the system from the operation of devices. Once these measurements are recorded, the unsupervised learning system takes the measured transients and applies a number of operations in order to disaggregate the loads.

The general implementation of these procedures are illustrated in section 3.2.2.1 and section 3.2.2.3.

3.2.2.1 Event Measurement

The implementation of a transient detection system is not part of this system as it requires a large set up and design in order to implement. The implementations of event detectors are covered in the literature and are thus not required to test the validity of this system as the event detectors currently available can be readily applied to such systems with minimal modification [18, 23, 39, 75, 76].

Rather, the system will be constructed such that the switching is implemented in an

automated manner where the measurement device is only required to start recording at predefined times; thus, allowing for an automated approach in measuring the transients. Section 4.5 is used to outline the exact tests and procedures used to accomplish this task.

This particular methodology for defining how the measurements are undertaken results in a case where the ground truth data can also be recorded. This is particularly important for the validation section as the data is stored in such a way so that it can be tested against itself to accurately analyse the different tests. This is discussed further in section 3.4 and the precise implementation for the methodology can be found in sections 4.5, 4.6.1 and 4.9.2.

The general measurement time required by the measurement device is determined by the characteristics of the system and the time that it takes the system to reach steady state once again, this is an important parameter that should be considered in choosing the measurement device (see sections 4.3 and 4.4) and some work has already been accomplished on attaining some of these parameters [72, p. 70].

These events are broken down into multiple combinations such that the NILM algorithm created (see section 3.2.2.3) can be demonstrated and validated on each unique case. These tests, referred to here as *protocols* are created to answer the specific research questions. There are a number of unique combinations that the system can be organized in, this implies that the NILM and load disaggregation algorithm should be tested and validated against each of these circumstances, the combinations are discussed further in section 3.3 and the validation of these in section 3.4. Each combination is validated to determine its disaggregation performance based on commonly utilized metrics in the field (see section 4.9.2).

The measurement system, the ability to accurately measure the events, needs to have a high sampling-rate and a high vertical resolution; this is required to create a baseline measurement which can be used to define the limitations of the system accounting for the smallest transients devices may exhibit. The high sampling rate allows the feature extraction system (the signatures of the events) to observe these miniscule differences in each event, and enable the clustering algorithms to correctly place events into their expected clusters for accurate characterisation. This measurement system, due to its capability has a large cost, which is why this research also looks into using the equipment to determine the lowest cost implementation that will work, this is discussed further in section 3.5. This is where trade-offs are observed to determine the feasibility of this technique.

3.2.2.2 The DC System

The distributed DC networks that are available currently fall into three major categories, each of these (and their different combinations) are used in this research. The aim of this is to be able to apply the methodology and test the feasibility of the technique to a range of different DC networks, with different electrical characteristics and generation capabilities.

The first DC grid type are the commonly utilized bench top power supplies that make use of power electronics to convert AC into variable DC to run systems.

Secondly, DC batteries are becoming increasingly common in electrical systems, they have the ability to create DC grids upon which devices can operate.

The third major platform upon which this will be applied to is currently designed and maintained by PeCo Power a company started within the school of Electrical and Information Engineering of the University of the Witwatersrand. Notably, two of the main members currently working on the system discuss the design and construction of the system in detail [72, 73]. The internal design and operation of the power electronic converters used in the system are not covered here, as the devices are seen as a lower abstraction level that is considered out of scope for this particular purpose. This does not, however, mean that the design and construction of the elements of the system are not ignored. The elements are simply viewed as to what their input/output components are, or as is more commonly known, as a *black box*. The construction of the system is covered in more detail in section 4.1.3.

3.2.2.3 Unsupervised Learning Procedure

The procedure that performs the load disaggregation is undertaken with the use of an unsupervised machine learning algorithm. The aim of this system is to make use of the switching measurements of the system over a period of time in order to define a feature space that each transient is capable of being placed into. This feature space contains multiple dimensions, where the number of dimensions is linked to the number of features that are extracted from each waveform. This is then defined in such a way so that when measurements are recorded, they are fed into the system, the system then produces a point on this feature space. This allows for multiple measurements to be placed within this feature space.

The aim for this system is such that a specific switching operation from a specific

device can be mapped to a point in this feature space where the operation of a device, and the signature it produces can be predicted and modelled. It is unlikely that each and every point recorded will be placed in the exact same point in the feature space². This allows for the grouping of clusters whereby if a measurement is recorded and the point is within a cluster, then the system is able to say which cluster it falls.

The labelling of these clusters can only be performed manually as the system does not know what these clusters represent; conveniently, however the technique proposed here keeps track of each event (device operation) such that labelling can be provided here. This is particularly important to be able to perform accurate validation and successfully assess the results.

A general outline of this clustering can be illustrated with the use of fig. 3.3 where the unlabelled system starts off as points in a feature space, a clustering algorithm then attempts to group the points into separate but distinct clusters. This system makes use of three-dimensions as it is viewable within a three-dimensional space and simpler to illustrate visually. The clustering procedure is likely to produce a high dimensional point such that visualizing the clusters is not possible.

The number of events illustrated in this figure is not indicative of the protocols utilized in this work, see section 4.5.5 which outlines the procedure to define how many events are required for sufficient analysis of systems.

The use of clustering for event-based methods is commonly used in the field of NILM and load disaggregation, making this technique readily comparable to other implementations [23, 47, 77–80].

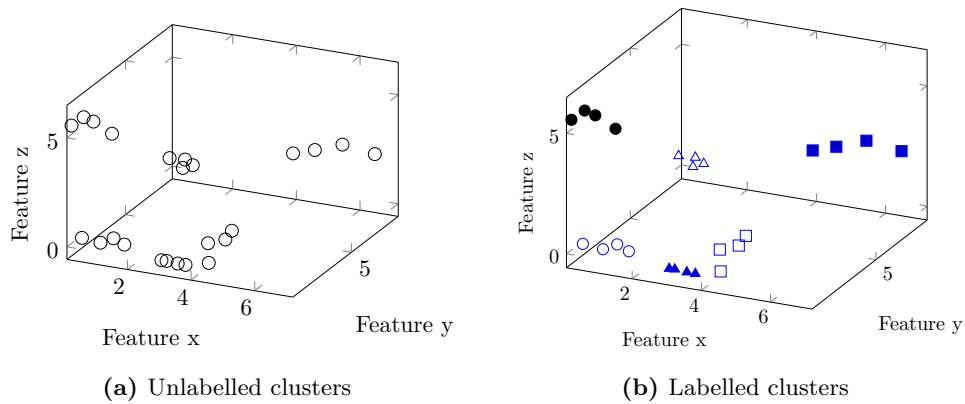


Figure 3.3: Clustering of three devices and their labelling

²This is due to the fact that small variances within the system (temperature changes, small localised interferences, etc.) cause the measurement to be slightly different.

This explanation above can be broken down into its constituent parts that make up the load disaggregation algorithm:

Feature extraction: Each measurement is run through a series of operations in order to extract a number of features from that transient signature.

Feature vector: The features extracted from each transient is then placed into a feature vector that is in a format that required for the input of an unsupervised machine learning algorithm.

The normalization: The weighting of each of these features and how prominent each feature is within the system is important. This is one of the parts of the system which is often calibrated in order to get a greater accuracy out of the machine learning process. Feature normalization assists in creating more consistent feature scales upon which the clustering algorithms run.

The clustering: Each feature vector of each event is run through the machine learning process and placed into a feature space.

Labelling: Once the clustering has taken place, each cluster should be labelled. The process of labelling clusters in this implementation is likely to be simpler as the ground truth data has been recorded for each transient.³ The labels assigned by the clustering process are compared against the ground truth data such that performance metrics can be created for analysis.

The process that can be illustrated via in fig. 3.4 which shows that each step of the process relies on the previous one and information is required to flow in both directions such that the system can function.

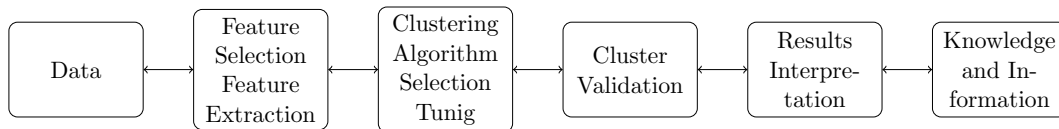


Figure 3.4: Unsupervised Process

³The use of such a system in the real world is unlikely to have the added benefit of having such precise ground truth data, this is likely to be one of the parts of this work that would require future research in order to be implemented in any system.

3.3 Switching Protocols

The presence of testing protocols forms the major framework to be able to validate a number of questions on the validity of NILM in such a system. These tests, each created to answer specific requirements, that when met, pave a path for future NILM and load disaggregation research and implementations in DC networks.

The protocols outlined here, and discussed in greater detail in section 4.5, seek to answer the research question of the capability of such a technique: can the technique disaggregate standard distributed systems with a number of devices, and what are the limitations therein.

The major tests that are created and evaluated seek to firstly determine whether, and to what extent devices are distinguishable on the different types of DC grids. The wide range of devices used, are the signatures, exhibited by the switching of devices, unique and repeatable. There are a number of specific, unique tests that are included in this research, seeking to answer specific questions about how devices operate, however, further than these tests, it is important to understand how such a technique may perform under real world conditions. This results in test protocols that seek to emulate the normal use case of an electrical network, where a diverse range of devices are operated in varying orderings and states.

This protocol testing set seeks to account for a wide range of possibilities that a load disaggregation algorithm may be used with, which is why it is important to observe these different scenarios.

3.3.1 Disaggregation Dataset Generation

Research in the field of NILM and load disaggregation requires a significant amount of data upon which to characterise their capabilities. Researchers in the field have compiled a number of energy datasets which contain measurements of electrical networks over periods of time, section 2.8.3.3 outlines the currently available datasets and what they provide. These datasets, different in their implementations, measurement types, and types of electrical networks (homes, offices, city blocks, etc.) are created such that researchers can run their algorithms to obtain performance measures.

Datasets currently available only use AC systems, which is why this research seeks

to create a unique dataset upon which to answer the questions of the research. This necessitates the design and creation of an automated system that directly controls the devices and measurement device.

3.4 Validation

Discussed in section 3.3, there is a clear and defined set of protocols for the switching of devices within the system, each with targeted tests. Based on this, there is a wealth of ground truth data which is present throughout the process, this information is uniquely suited when it comes to the validation of the system, this data is recorded as the protocols precisely control the system under test.

An important parameter of an unsupervised machine learning algorithm is that of its accuracy of detection. There are a number of metrics that are used within this knowledge area that allow for verifying the switching that is measured and how those switching transients are identified [52, 53, 76, 81].

This implementation of ground truth data validation cannot be used in *real world situations*, however, for this particular use case, it is used to assist in the validation process⁴.

The validation techniques that are utilized in this research are outlined in section 4.9.2, these are used to provide insight to the different testing protocols performed.

3.5 Sampling

Research in this area always involves looking into the physical installation of the meters, the specifications of those meters, specifically the sampling rates, and sampling types that are utilized in the system [1, 11, 30, 82–92].

This normally involves the understanding of the underlying system that devices are operating within, for example most systems operate within the 50 Hz to 60 Hz

⁴This due to the fact that the ground data generated in this implementation is generated with a priori knowledge of the switching times of each device. Standard implementations of any NILM system is unlikely to have this level of ground truth data (there are exceptions to this), which is why it is important to verify that the solution to the problems in this document can in fact be used to function as a standard NILM implementation because, ideally, it will not need the use of the ground truth data to perform the clustering.

supply frequency, and so some techniques only require sampling rates that satisfy the Nyquist criterion [47, p. 7]. Other research, particularly research that looks at the microscopic features of device signatures on the network are required to make use of higher sampling rate data acquisition units [59, p. 169]. These cases need to satisfy the Nyquist criterion for the highest possible frequency that needs measurement.

The main reason that this forms a large part of the work is that, naturally the higher the sample rate, the more information that can be extracted, yielding a greater ability to observe more signatures (from the entire spectrum from small to large devices) and hence greater disaggregation ability. The downside of this is that increasing the capability of the sensor system, by increasing the sampling rate has a direct correlation to the cost of implementing such a system. Consequently, much research is involved in determining the optimal sampling rate at which operation of devices can sufficiently be observed and disaggregated within a required accuracy range.

Often, the result of increasing the system capabilities, in terms of sampling rate, resolution, etc., result in an increasing cost. This increasing cost is not always linear; fig. 3.5 illustrates a number of possibilities of cost against the overall system performance. What is important to note here is that currently there is no possible way to reach perfect performance, however, increasing cost and complexity naturally yield greater performances up to a limit⁵.

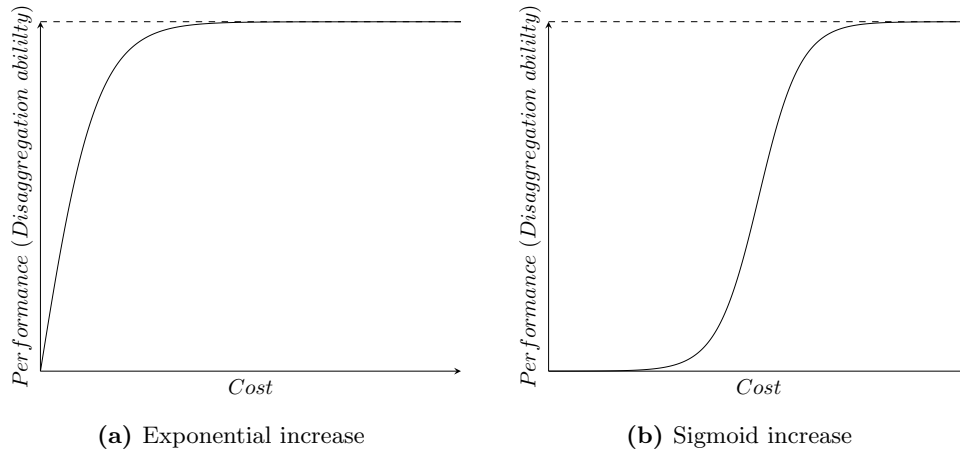


Figure 3.5: Illustrations of different cost–performance models

This research attempts to gain an understanding of how the sampling rate affects the ability to perform NILM on such a system. The aim of this is to determine some form of relationship between the sampling rate and the performance of NILM and

⁵Research in this field, due to its large diversity and different system configurations means that the system performance has a high dependence on the size of the network.

how these are related when looking at different types (and sizes) of devices when they operate on the network, specifically in the distributed DC case for this work. The implementation of this testing, discussed in greater detail in section 4.7 describes how the original sampling rate (the top benchmark) is used as the best-case scenario with the highest sampling rate, then decreasing sampling rates are then compared to observe the relationship.

Chapter 4

Methodology

This chapter outlines the reasons for, and the methodology used to carry out the research on determining whether NILM and load disaggregation are feasible techniques to analyse distributed DC systems. The section begins with the design of a distributed DC grid and covers a diverse set of devices that are tested. The design and implementation of the system control is introduced as well as how ground truth data is generated and tracked; this also includes the system that manages the data pre-processing. The chapter then outlines the technique utilized to create downsampled versions of the original raw data upon which to test a number of assumptions on sampling rates required to use NILM and load disaggregation on such systems. Finally, the techniques for running the data analysis (NILM and load disaggregation) techniques are discussed.

4.1 DC Test Bed

The DC *grid* and its variants—mentioned in section 3.2.2.2—form the major testing platform that this work relies upon.

The design and construction of the DC test bed relies on a number of different DC technologies, each of which are tested in this work. This is done to be able to test a range of different source technologies (and combinations thereof), thereby verifying how well the system performs on these different implementations. Currently, there are three types of DC systems (technologies) that are commonly available to end consumers which are the:

- DC battery,
- AC-DC power supply, and
- smart battery.

These three technologies all allow for different connections and configurations to be made when constructing the DC grid; this implies that a grid can be constructed with each of these elements in isolation, and a grid can be constructed by combining each of these different types of devices in any combination possible, due to the fact that they all operate on the same bus voltage. The flexibility of these sources means that different technologies can also be combined in different power capabilities and numbers of sources.

This major attribute of each of the sources, their ability to operate at 12 V_{DC} , means that there are seven combinations of each of these supplies that the system can be combined into¹. The first three combinations are simply each item in isolation where each terminal is connected to the relevant points on the grid, this is illustrated by fig. 4.1. The figure illustrates the fact that each of the three types of sources has two terminals, the positive and negative terminals, and these are connected directly to the positive and negative terminals of the load network respectively. The series impedance of the source depends on the source and lead characteristics; the variation in these parameters are not covered in detail in this work as it is out of scope and the system constructed has minimal resistance introduced in the form of connections. The source (V_S) represents the three types of sources that are observed in this study.

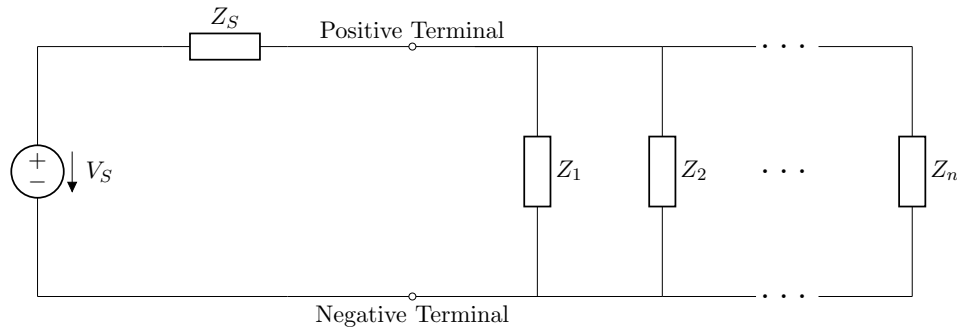


Figure 4.1: Generic DC grid - single source

¹The 12 V_{DC} is not a strict reference voltage, many of the devices are capable of operating at a larger range of voltages, this is relevant in the protocols that look at the response of the grid when the voltage has different set points, see section 4.5.1

The next set of combinations involves the parallel connection of more than one type of source to the grid, this accomplished by introducing the second, or third source in parallel again as illustrated in fig. 4.2. The different combinations possible, and the types that are used throughout testing are listed below.

- AC-DC PSU & battery,
- AC-DC PSU & smart battery,
- DC battery & smart battery, and
- DC battery & smart battery & AC-DC PSU

The illustration of loads in this figure does not represent the physical installation as the system involves a further implementation of a switching apparatus that allows for direct control of each load, this is discussed in section 4.3; however this illustrates the general layout of components.

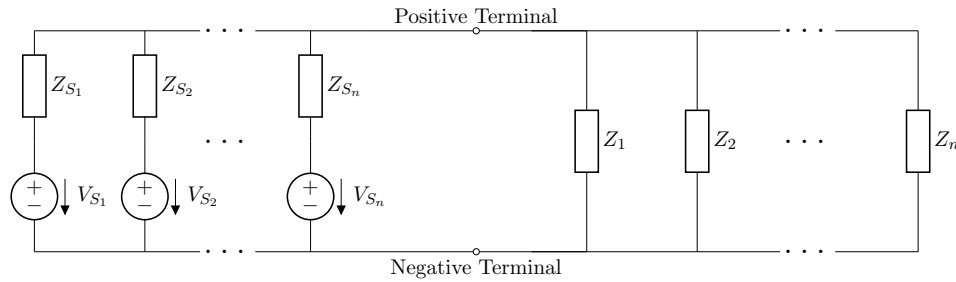


Figure 4.2: Generic DC grid - multiple sources

These three types of supplies are outlined in sections 4.1.1 to 4.1.3.

4.1.1 DC Battery

The DC battery is the simplest source implementation in the grid setup as it involves connecting the terminals of a battery to the bus terminals of the system where devices are connected.

The battery used for the range of tests is a lead-acid battery, with a 36 A h rating. During testing, the battery voltage was kept within the range of 12.5 V to 13.3 V. Charging of the battery took place outside of testing, as the battery charger is likely to interfere with the system as it is capable of feeding power through to the loads which is not desirable for the tests.

4.1.2 AC-DC Power Supply

This type of power supply, a standard variable DC voltage bench top power supply that is used in laboratory environments provides a simple interface to create the 12 V_{DC} grid. The variant of power supply used in this system is the Topward 6303D and is powered directly from mains $230\text{ V}_{\text{RMS}}$ [93]. Two of these units are utilized, each containing two power supply units, which equates to a total of four independent power supplies. Each channel from a power supply is capable of providing a maximum of 3 A output current. This setup provides the capability to supply the grid with a total of 12 A at 12 V , resulting a maximum power capacity of 144 W .

The operation of these power supplies is illustrated in fig. 4.3 where the system supplies the required current at the set voltage point.

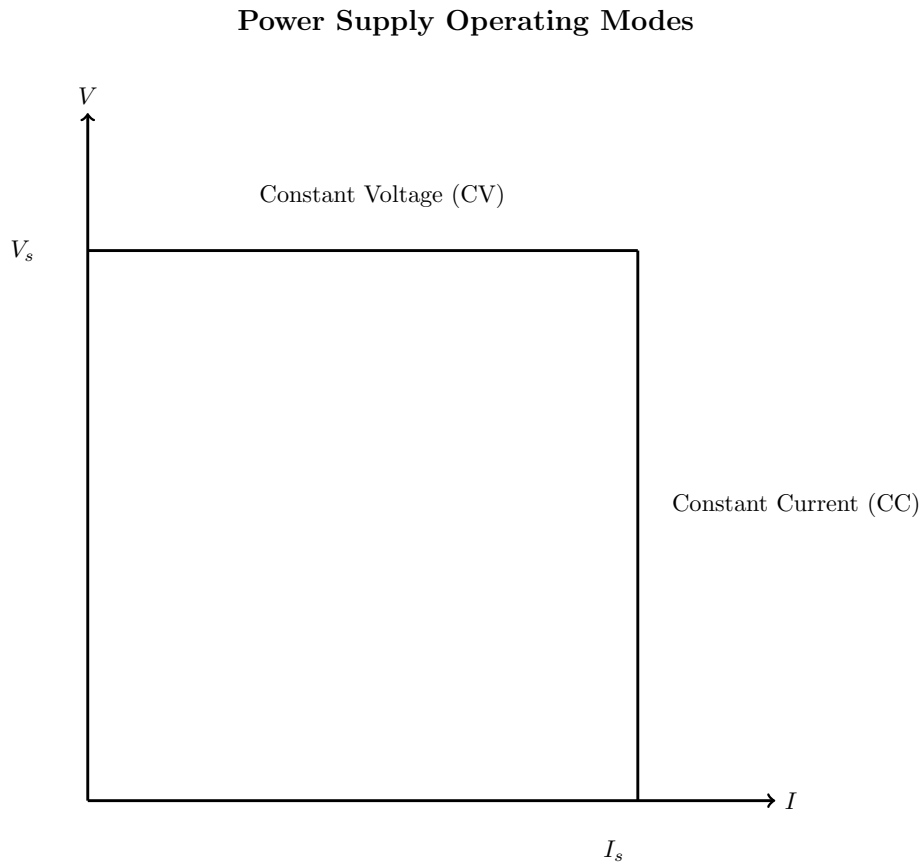


Figure 4.3: VI operating curve for a DC power supply

While operating, the power supply combination (as they are connected in parallel) splits the current load throughout the different supplies depending on the power requirements of the loads. This means that if a low-power device is operating (e.g. $\sim 5\text{ W}$), then only a single supply will supply current through to the load. The supply

that sends the power through to the system is non-determinate, implying that when a device turns on, the system will determine which supply will provide the current necessary. When a larger power device is operating, the individual power supplies will each provide current in order to maintain the grid voltage². This means that, in relation to fig. 4.3, when both power supplies are connected in parallel (ready to supply current to a load), and depending on which power supply is acting as the *master*, the current will first be supplied by the *master* supply, up until the point at which the individual power supplies current rating is reached; after that point, the *slave* power supply will provide current to keep the voltage at the set level.

The implementation of these power supplies satisfies the requirement to have a distributed source grid that devices operate on. However, as stated in section 3.2.2.2, a single power supply can be utilized and still prove the capability of observing devices operating on the network as the measurement parameter applies to single- and multi-supply configurations. These extra power supplies implemented here assist in providing more power when required³.

Unlike the DC battery, this type of power supply inherently introduces more noise on the system, however, this noise is negligible as the power supplies utilized are designed to reduce this noise introduced in the switching. The concepts of noise on the system are discussed further in chapter 5.

4.1.3 The PeCogrid

The third technology observed within this work is the increasingly popular power-electronic converters. The PeCogrid, an autonomous, plug-and-play, distributed DC grid is currently under development at the University of the Witwatersrand [94]. This system provides the unique ability to perform the function of a generic distributed DC grid. It makes use of power electronic converters to interface the source and storage elements. This interface provides the ability to generate a regulated DC voltage that devices run on. The details of this system can be found in work conducted by individuals from the university [72, 73, 94, 95]. The general structure of the system

²Note: It is not clear how the system decides which power supply will provide the current for the lower power devices, however, it is believed that it is dependant on which power supply is acting as the master and which ones as the slaves. This implies that one of the power supplies acts as the master voltage power supply, and the other supplies will set their voltage to match up with the master as they track the master. The master is then commonly the supply that will provide current when low-power devices are operating.

³There is a wealth of information that explains how parallel DC power supplies share current/power based on their master/slave relationships due to their design, this is out of the scope of this exercise.

is created with the use of three major elements: a storage, generation, and load element [72, p. 1]. These three are capable of being implemented in any combination required with the base assumption that, to create a grid, a minimum of a single storage or generation element are used. The system is constructed with connectors such that any combination of these three can be connected and interfaced to get the required system capacity.

The system creates a regulated 12 V_{DC} DC that can be tapped off at any point on the system to power devices, or connect further storage/generation elements. The PeCogrid concept, illustrated in fig. 4.4, highlights the fact that power electronic converters are used for the core operation of the system, they each have an interface with the *grid* side of the connection, this is where the 12 V_{DC} is maintained, and on the other side, they interface with a range of different sources, which can be: batteries, solar photovoltaic panels, wind turbines, etc which may all operate at different, and varying voltages. The PeCogrid unit utilized in the testing protocol makes use of a DC battery as the source for the internal storage element.

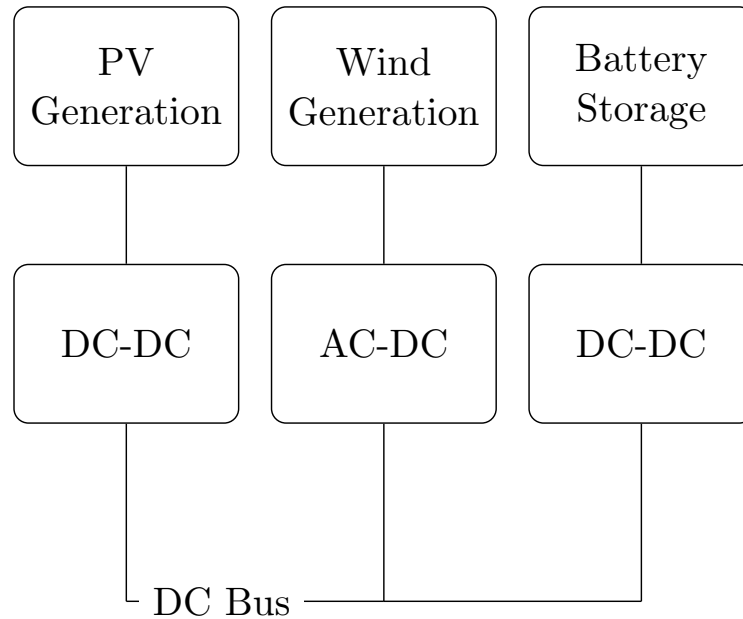


Figure 4.4: Concept of PeCogrid interfacing technologies

The system constructed resembles a generic distributed DC system as illustrated in fig. 4.2. This is particularly relevant as it provides the ability to verify the answer to the research questions outlined in section 1.1. This hybrid and dynamic *grid* is of particular relevance as the technology allows for simple integration of different sources in order to create a highly distributed system that is flexible and highly scalable: a particular case where current NILM techniques cannot accommodate.

4.2 DC Devices

Major importance is placed on the selection of devices that are operated within the system, as the range of devices should represent a generic example of a DC system. There are a number of requirements that these DC devices need to satisfy to enable the ability to undertake a range of tests that sufficiently illustrate a commonly implemented DC system: examples of this include small houses that use this technology to power common household appliances. This implies that a wide range of DC devices are required, with different functionality, as well as different power ratings; this diverse range of devices seeks to create generalizable results.

The four types of devices outlined in section 2.7.1 provide a range of possibilities for devices to use within the system. Due to the methodology employed in this work (and the limitations in the controllability of devices, see section 4.3), device types III, and IV are currently untested and is left as future work, see section 6.1.4.1⁴.

Due to the simple construction and requirements for most devices, there is a large range of devices of type I available on the market. Consequently, of the range of devices tested, the devices of type I make up the majority. The full range of devices utilized in this work is outlined in section 4.2.1.

A number of the devices require minimal modification to interface with the automated switching system as outlined in section 4.3; the different mechanisms that these devices use to switch is discussed in section 4.2.2.

4.2.1 Tested Devices

The range of devices used in the different protocols include: lightbulbs, fans, inverter, phone chargers ($12\text{ V}_{\text{DC}} \rightarrow 5\text{ V}_{\text{DC}}$), etc. The different devices in each category also have different power ratings and different manufacturers, this creates a diversity of devices available for testing; this diversity is often observed in common electrical networks.

The different devices are outlined in table 4.1 where the range of different devices includes a mix of type I and type II devices each with varying power ratings and capabilities.

⁴Additionally, device types III, and IV are not as prevalent in DC systems, much like AC systems, these devices make up a small percentage of devices used.

Table 4.1: DC devices

Device ID	Device Type	Description	Operating Current (at 13 V)	Operating Power (W)
1	I	LED Light 1	0.6339	8.2407
2	I	LED Light 2	0.6239	8.1107
3	I	High-Power LED Light	3.2121	41.7573
4	I	Small PC Fan	0.241 94	3.1452
5	I	FM Transmitter	0.033 65	0.437 45
6	I	Phone Charger 1	0.4298	5.5874
7	I	Phone Charger 2	0.420 98	5.472 74
8	I	Small Inverter	0.3319	4.3146
9	I	LED Light 3	0.5452	7.0876
10	I	LED Light 4	0.294 02	3.822
11	II	Small Air Conditioner (Hot/cool)	0.380, 10.2249	4.5214, 132.924
12	II	Hair Dryer (heating incl.)	0.5036, 11.100	6.5466, 144.300
13	II	Three Speed Fan	-	-
14	I	LED Light 2 (modified)	0.9163	11.911 44

A number of these appliances require modification to be implemented into the controllable load system, this is due to the fact that some appliances use secondary switches or selectors to change states. The connections and modifications are outlined in section 4.2.2.

4.2.2 Device Connections

Many devices implemented in the testing simply require a direct connection of their two terminals to the grid; an example of one of these devices is a lightbulb where there are no internal switches and simply placing the leads onto the grid turns the light on⁵. A number of devices are modified to connect to the grid as they have internal switches (controllers) which in normal situations would require physical

⁵These lights are often coupled with a third party switch so that it can be turned ON and OFF, for the purposes of this research, the switches are always placed onto the ON position as the automated switching system manages the state changes of these devices.

switching with the use of a button or multi-state lever. These internal switches are modified to enable the automated switching system to enable fine control of the different states.

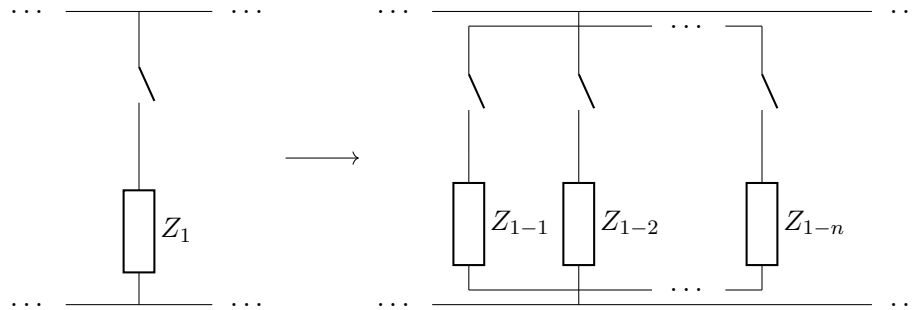


Figure 4.5: Type II device possible internal construction [48]

There are two major types of switching mechanisms that are employed in the type II devices used throughout, these are illustrated in sections 4.2.2.1 and 4.2.2.2⁶, once these switches have been modified, they need to be interfaced with the automated switching system discussed in section 4.2.2.3.

4.2.2.1 Trinary Switch Modification

The first type makes use of a Double Pole Double Throw (DPDT) switch which creates the ability to have three distinct operational states. These devices have two separate circuit components in parallel, the devices in the test set that make use of these types of switches are the hairdryer and the blower (device ID's 11, and 12 respectively). They each have a fan for the first element, then both have a heating element connected in parallel with the fan for the second state. The switches enable three possible device states: the off state, the fan-only state, and the fan and heater state.

Figure 4.6 illustrates the three possible states with the switch in different positions.

This device is easily modified to be connected into the system as it contains two separate parallel circuits when the switch is removed. These two separate circuits are provided with their own relays (from the automated switching system) where they are independently switched on demand. The resulting circuit diagram is illustrated in fig. 4.7. The automated switching system is set so that it understands that this

⁶There is a third major type which makes use of a debouncing switch; none of the devices within this test makes use of this switching method.

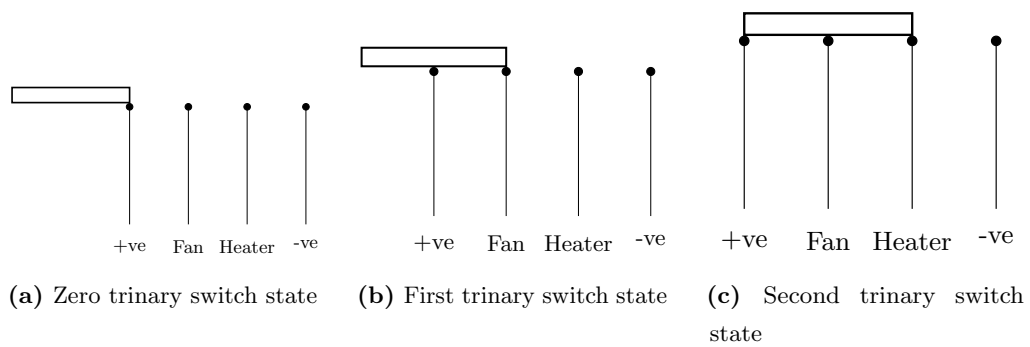


Figure 4.6: Trinary switch device original switch mechanism

device has three states, and it understands which relay corresponds to the different circuits and states (see section 4.3 for more detail on this).

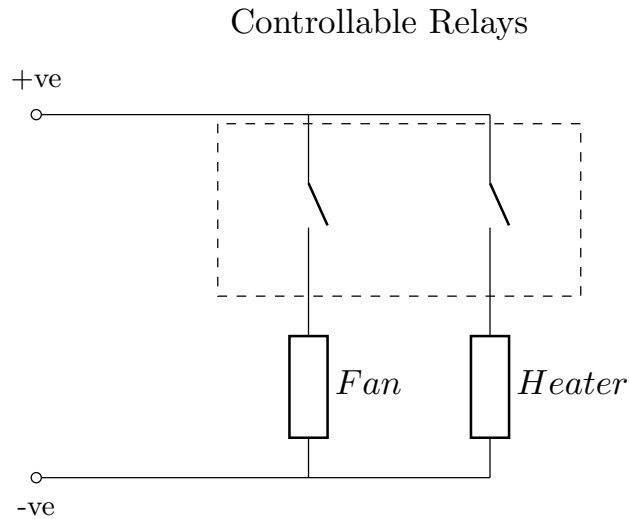


Figure 4.7: Modified trinary switch

4.2.2.2 Rotary Switch Modification

The second type of device type II connection makes use of a rotary switch, this switch brings in increasing resistance at each stage/state. Figure 4.8 illustrates how a single pole switches between different points, where each step increases or decreases the series resistance of the device as a whole. The only device that makes use of this type of switch is the three-speed fan. This type of device has four possible states: off, slow, medium, and fast.

This rotary switch device has been modified to enable the system to transition from any of the four states, to any of the other three states; this makes use of a relay for

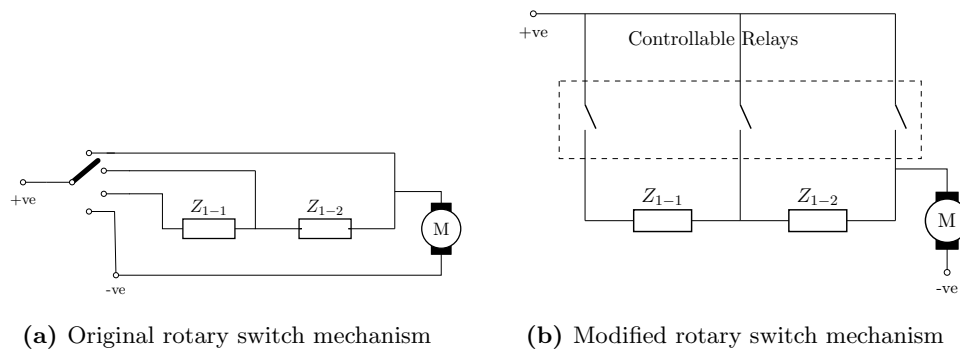


Figure 4.8: Device with rotary switch component

each line where the rotary switch was removed. Much like the other kinds of type II device, the automated switching system is designed to understand which relay corresponds to the different connections and how to connect/disconnect the different relays to set/transition different states.

4.2.2.3 Device Connection Configuration

Most electrical installations that make use of switches or relays place the switch on the high-side that is connected/disconnected when the switch actuates. This is based on safety considerations as this means that the device does not have terminals that are live except when it needs to be powered on. This reduces the risk of the device shorting to ground if anything breaks on the device when it is not meant to be operating as many devices use their chassis as a grounding point. Most of the devices in testing make use of high-side switches. There is one exception: the three-speed fan; because of its connection type, the system can only make use of low side switching, the placement of the relays in this case can be seen in fig. 4.8b. These two switch placements, high- and low-side, are illustrated in fig. 4.9.

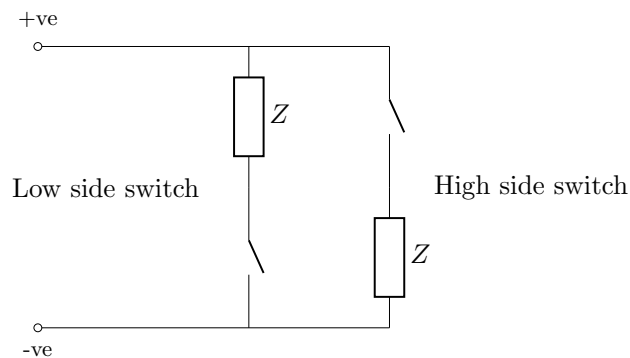


Figure 4.9: Switch and device placements within the circuit

The placement of the switches in reference to the device is not expected to augment the response on the grid as this is simply a closing of a circuit and does not affect the characteristics of the circuit operation.

4.2.3 DC Device Comments

Due to the difficulty in obtaining type II devices with directly controllable states, there is a reduced number of these tested, this is partly due to the fact that many DC devices available have a simple construction, and as such, more complex devices are not yet on the market. This is due to the fact that the methodology (no event detector) does not allow for appliances that do not have states that are selectable by a user.

Some type II devices (like the debouncing type system discussed above) make use of a single switch where each press of the switch will cycle the device through its different modes. This can be illustrated by fig. 4.10; this means that the device can only cycle through its states in this way. None of the devices utilized in this study have this switch mechanism, further work can look into how this switch mechanism may be different.

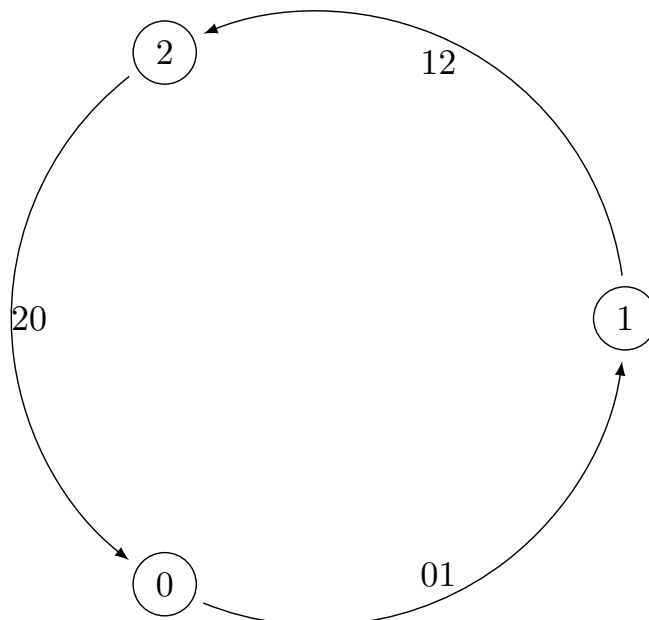


Figure 4.10: Single direction state change switch

4.3 Automated Switching System

Automating the state changes of devices within the system is required in order to accomplish the different tests/protocols outlined in section 4.5. Directly controlling the operation of devices within the grid while automating the entire procedure enables the ability to take accurate measurements while being able to repeatably run specific tests on demand. The ability to automate the entire process, from switching the devices, to measuring each event, etc, is how this system is designed to run a number of procedures which entail targeted tests/protocols to validate questions about the feasibility of NILN and eventually load disaggregation in distributed DC systems. The specific design of these procedures/protocols, is outlined in section 4.5.

The system, due to the number and nature of tests, should be capable of switching a number of relays many times, over long periods of time. This warrants the design and setup of the *automated* part of the system.

A centrally controlled system is designed here, on demand, to control a number of different devices. Section 4.2 discusses the different types of devices that require integration into the system, it outlines how the different devices are controlled with a number of individual and independent switches. This automated switching system, making use of the range of Siemens Logo! devices, uses the relays inside the devices to switch the different devices through their different operating modes/states on demand and with consistent timing. Figure 4.11 illustrates how relays from the Logo are connected to the *grid* and interfaced with the different devices.

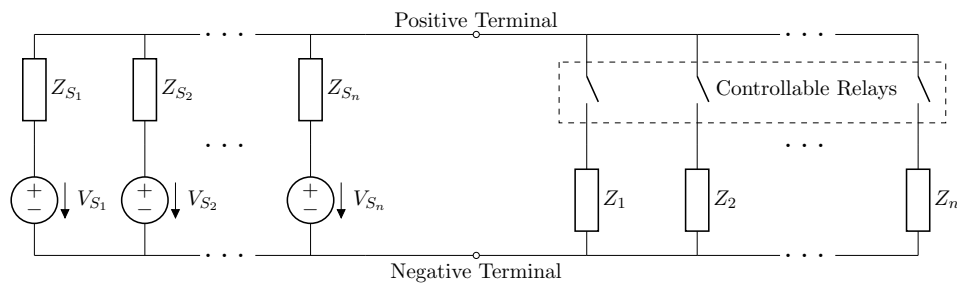


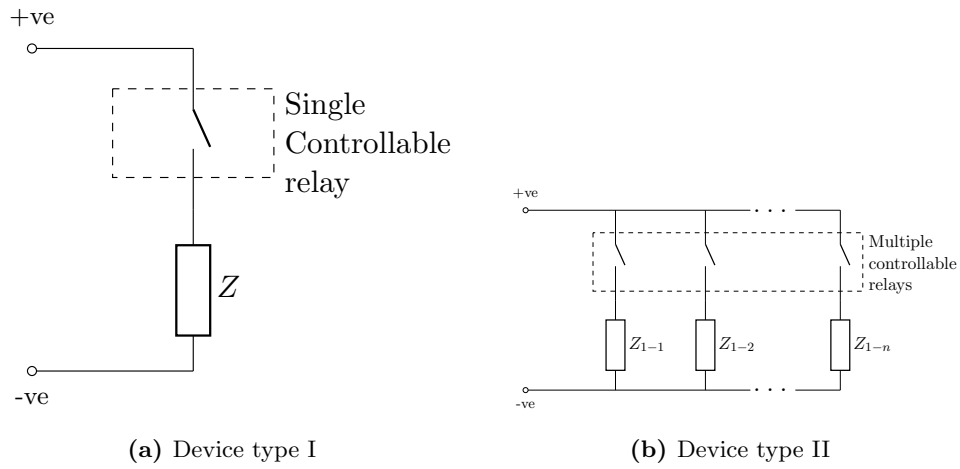
Figure 4.11: Source and device system layout (controllable relays represent the Siemens Logo!)

The different loads in fig. 4.11 illustrate how all devices are connected; it is the software, and database, integrated into the Python platform, that provide the means to control the different devices and the different device types with tight control between the timing of the different operations.

Table 4.2: Device type I states and corresponding relay states

Device State	Relay State Vector
OFF	0
ON	1

The two types of devices, and how they are connected to the relays are illustrated in figs. 4.12a and 4.12b; each device either has a single relay controlling it, or multiple relays that need to be set into specified states to change device states.

**Figure 4.12:** Relay positioning for different devices

Tables 4.2 and 4.3 illustrate what the different relay states need to be in for the different devices to operate in their different states. The relay number in the table correlates to the relay number in figs. 4.12a and 4.12b. These tables define, to the Python platform, the state that each relay should be in, in order to change the states of the different devices within the system. Tables 4.2 and 4.3 detail the state that the different relays should be in for the different states a database keeps track of each event, and the information behind each event as well as which relays the different devices and their sub-circuits are connected. This enables the ability to encode what each event means, as it is the link from the previous event to the current event that represents the transition that takes place (see section 2.7 which discusses the different devices and how they transition). This information is used throughout the process as the ground truth which keeps track of what each event signifies, this is discussed in detail in section 4.6.1.

This automated switching system, controlled via Python, is integrated with the measurement system (see section 4.4), as the system takes measurements of each

Table 4.3: Device type II states and corresponding relay states

Device State	Relay State Vector
0	00
1	01
2	11

event as devices are switched, this necessitates a centralized control and measurement system to manage the devices and measurement while keeping track of each event as it takes place according to the outlined procedure/test (see section 4.5).

This entire system is designed and constructed with the use of Python as there are libraries available for the Logo! devices which enable Python to interface with the Logo! devices over an ethernet connection as well as libraries to control the measurement device.

4.3.1 Physical System Characteristics

A main goal of implementing this automated switching system is to emulate, as close as possible, the switching characteristics that occur in normal use. This system isolates the switching apparatus of every device from every other device, the individual relays inside the Logo! make this possible, they can be independently operated from one another. The characteristics of the relay switching likely augment the signature in small ways, this is discussed as something to verify in future work (see section 6.1.4.3).

The connection to the Logo! device is accomplished via the modbus protocol where a PC, connected over ethernet, interfaces with the Logo! via its IP address, and it directly changes the values in the address space of the Logo! [96, 97]. This switching implementation creates a low latency control mechanism for the device state changes. The communication with the Logo! through the modbus protocol can have a delay of up to 100 ms⁷, this indeterminate amount of delay is factored into the timing for the measurement system such that the Logo! switches the required relays at the correct time and the event is not missed (this measurement event window is refined during the design stage), this is discussed further in section 4.4 [97, p. 321]⁸.

⁷The internal control rate of the Logo! runs at 0.1 ms, so once the command is sent through to the Logo! the delay from when the internal state is changed to when the relay actuates is miniscule.

⁸This timing from instruction the being sent to the relays actuating is non-deterministic, this is due to many factors including the cycle time of the device, hardware interrupts of the controlling

The potential of the relay contacts is isolated from the power supply that the Logo! requires, this ensures that the Logo! and its power supply does not interfere with the devices on the DC grid while operating [97, p. 51]. The Logo! device is commonly implemented in industrial control applications, as such it is ruggedized and reliable in its operation making it a convenient tool to run many hundreds of switching events for multiple tests.

4.3.2 Event Detector Requirement

The design of this system, its ability to directly control and measure each device, means that each event only contains information about the switching event as the measurement device measures only the relevant period/window in time. Consequently, this implementation does not require the use of an event detector; the absence of an event detector in this technique is discussed in section 3.2.2.1; furthermore, the benefits of creating a unique event detector for this use case is discussed in section 6.1.4.5.

Contrary to other implementations [26, 38–40], this system generates its own ground truth data. This ground truth data includes the device switched, the transition that it did, the time at which it took place, as well as other meta-data, this is stored in a database for the analysis of the protocols.

4.4 Measurement System

The measurement system implemented satisfies the following criteria:

- A sampling rate that effectively captures the transients on the system.
- A high-resolution to capture the vertical variations of the grid voltage.
- The ability to directly control the measurement and settings with an automated platform.
- The ability to communicate with the device and transfer measurement data periodically.

PC as well as random network interruptions that can affect the timing of different operations.

Each of these requirements is outlined in sections 4.4.1 and 4.4.3, the current best measurement device that accomplishes this is the oscilloscope range from Picoscope - this research uses the Picoscope 4424A [98]. The USB connected oscilloscope is connected directly to a PC where the Python platform communicates through the API supported by the device; the oscilloscope is directly controlled and set to record with settings specified from the program. The overall system designed to support this is discussed in section 4.4.4.

These requirements set a minimum standard such that the maximum amount of information can be gleaned from the grid voltage during the transient events; this standard is created by setting the highest sampling rate possible from the device. This is to set a standard at the highest level to ascertain if, even the smallest variance in grid voltage, can be used as a differentiating variable to distinguish devices. Section 4.7 covers the process that takes the different events through to the different downsampled versions to find the lower limit on the sampling rate required which forms part of the research covered.

4.4.1 Oscilloscope Specification Requirements

The main requirement that the measurement device needs to satisfy is the required sampling rate such that it can accurately measure the highest frequency transients (down to the low frequency, and DC components) that are found on the DC grid when devices change states.

The industry currently maintains a standard for monitoring and capturing the high frequency components of transients on electrical systems, however, the standard is currently geared towards AC systems as the requirements/need for DC systems is still small [99]. The concepts that are used in the standard can be readily applied/adapted in this use case as the characteristics of transients are similar for both AC and DC systems.

The aim of the standard is to lay out what types of transients occur on a system and how to accurately measure them, this work looks at the impulsive and oscillatory transients⁹.

⁹The standard lays out a number of typical characteristics/observations of electric power systems, the other types are: short-duration root-mean-square (rms) variations, long duration rms variations, imbalance, waveform distortion, voltage fluctuations, and power frequency variations. These characteristics are not applicable in DC systems and are not considered for the measurement system here [99, p. 14-15].

Impulsive Transient: This transient is characterized by a sharp rise or fall of the system voltage and/or current where it reaches its peak within a short space of time ($1.2\mu\text{s}$: the time to rise from 10 % up to 90 % of the peak value reached.) and decays over a longer period of time ($50\mu\text{s}$: the time from the start to 50 % of the peak value) to the original value [99, p. 15-16].

Oscillatory Transient: This transient has a sharp voltage and/or current movement that moves upwards and then downwards (or in the reverse) in repeated cycles until it settles to some steady-state position. This is observed as ringing where the magnitude of the parameter changes polarity continuously and decays over a period of time.

The measurement device is required to be capable of detecting both types of transients on the system; the standard implies that the highest frequency that either of these transients exhibit is up to a maximum of 5 MHz. This is an upper limit on the sampling rate that the measurement device should be capable of measuring. To satisfy the Nyquist limit, a sampling rate of at least double the highest observed frequency should be employed to accurately reflect the transients characteristics without aliasing.

In addition to the sampling rate requirement, the device should be capable of measuring the transient in its entirety. The standard, once again, sets out a typical duration for these transients with an upper limit of 50 ms. The oscilloscope measures each event for a total of 0.5 s, satisfying the requirement of the standard. This longer duration also allows the measurement device to capture any longer lasting transients that devices may present. By this time frame, any device that has been switched on to the system, has settled into its new steady state.

4.4.2 Additional Measurement Details

Due to the nature of the system observed, the fact that it *normally* operates at around the 12V_{DC} mark, it is implied that most of the vertical deviation of the voltage will be around that range as a major settling point (steady-state level).

To take advantage of the high-resolution Analog to Digital Converter (ADC) of the oscilloscope, AC coupling is used for testing¹⁰. This AC coupling allows the

¹⁰If DC coupling were to be used, the scope voltage range would have had to be set to a high value ($\pm 20\text{V}$), reducing the measurement ability (dynamic range) of the small signals present in the waveforms.

system to be setup such that the minimum and maximum measurable voltages, by the oscilloscope, will closely match up to the minimum and maximum voltages that device transients will create on the grid, allowing for some headroom on the oscilloscope range. The oscilloscope is set to capture the most amount of information with its full 12-bits of resolution.

This setting is illustrated in fig. 4.13.

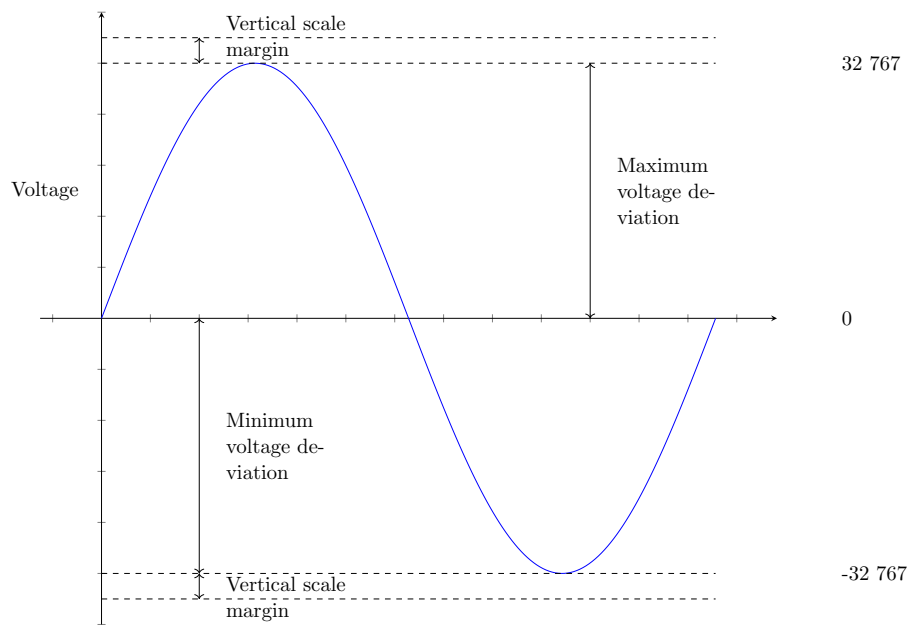


Figure 4.13: Vertical setup of oscilloscope

In a situation where DC coupling were to be used, much of the resolution that the oscilloscope provides will be lost as the voltage is unlikely to drop below $0 V_{DC}$, this would create a lot of wasted bits that could be better utilized. Figure 4.14a illustrates the possible measurement range if DC coupling were to be used where it highlights all of the space below the $0 V_{DC}$ mark that is unused.

Experimentation highlighted that a safe choice for the maximum range for the vertical scale is $\pm 20 V$ which provides enough headroom to capture the largest voltage deviation from the transients of devices tested¹¹.

Determining the dynamic range of the measurement system is relatively simple here (eq. (4.1)); this setup has $0.00977 V$ or $9.77 mV$ for each voltage level/step ($\sim 72 dB$). This is an important characteristic in the measurement setup as it has enabled the

¹¹There are a range of protocols that look at changing this vertical scale to test if the change could increase the ability in differentiating the two light bulbs from one another.

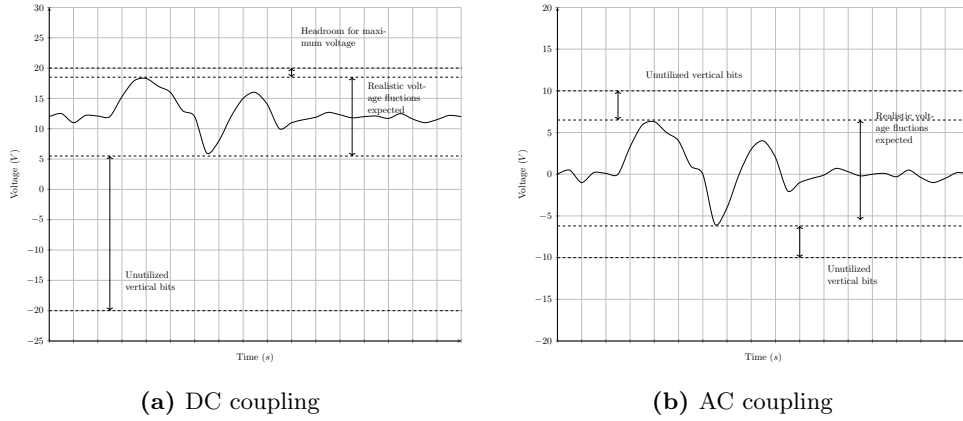


Figure 4.14: Comparison of characteristics between different coupling types

measurement of the entire range of events while maintaining the highest precision by making use of the full resolution available from the scope [100].

$$DynamicRange = \frac{AbsoluteMagnitudeRange}{2^{NumberBits}} = \frac{40}{2^{12}} \quad (4.1)$$

4.4.3 Measurement Automation

The range of oscilloscopes from Picoscope provide the ability to satisfy the second section in the criteria of the measurement device. The USB oscilloscopes are designed to be powered from a PC, where the PC is used as the interface for taking measurements. There is a standard application that allows for direct control of the oscilloscope through the PC [101]. The Picoscope range also has an API that enables the device to communicate with a PC as desired, the API enables designers to develop systems to control the scope with a range of different software languages. This API is integrated with Python in this work such that the oscilloscope can be used to automatically measure events when required [102, 103].

The Python platform designed specifically for this research enables the different devices to be controlled from the PC. The Python application is used to orchestrate the automated switching system with the oscilloscope to change the states of devices in the system while measuring the transient response on the network.

The Picoscope 4424, with four channels is utilized for this work¹², providing the capabilities specified in section 4.4.1 and those discussed in this section. The scope

¹²only a single channel is used in this work as this enables the system to sample at the highest rate and to record for the longest time

has a sampling rate of 80 MS/s (equating to a bandwidth of 20 MHz), exceeding the requirements of the highest frequency that transients may introduce into the system. This high sampling rate can not be utilised with the USB streaming setting; thus, the internal buffer of the oscilloscope is utilised to store the samples which are then transferred to the PC once recording has finished. The highest sampling frequency usable with the API is up to 20 MS/s, still above the required sampling rate for the work¹³.

4.4.4 System Coordination

The Python platform designed for this research is constructed such that it communicates with the automated switching system (Logo!) and the Picoscope in order to orchestrate the protocols. The platform is the central system that determines the device which is required to be switched, it then correlates the device and its required state to the required states each relay should be changed to for this transition to take place. Once this information is known—which depends on which protocol is being run and where in the protocol timeline the system is (see section 4.5)—the platform sets up communication links to the Logo! and the oscilloscope to run and record events in the protocols.

The Picoscope introduces a small delay from when it is enabled, to when it can begin recording. Therefore, the communication with the Logo! and the scope is done concurrently to allow the oscilloscope to start recording¹⁴ over the window of time when the Logo!’s relays actuate and the transient response occurs.

Tuning the timing for the Logo! and the oscilloscope is done such that the beginning of each waveform is fully captured without the start of the transient occurring before the measurement window. Further, the timing makes sure that the start of the measurement is accounting for some of the longer lasting transients to be completely caught/observed. The timing, due to the non-determinate nature of both devices, means that the start of each transient is placed around the first quarter of the measurement window. The placement of the event in the measurement varies from event to event due to the two devices different delays each run, this is illustrated fig. 4.15 which highlights the time shifts in each measurement. Each measurement is

¹³The highest sampling rate achievable through the streaming mode appeared to be approximately 6.6 MS/s, far below that achievable when using buffering mode with the API (20 MS/s).

¹⁴Just like the Logo!, the oscilloscope also has a non-deterministic setup time (up to 50 ms) which means that the two require further coordination.

still effectively captured by this system, not losing any important features from the transient event.

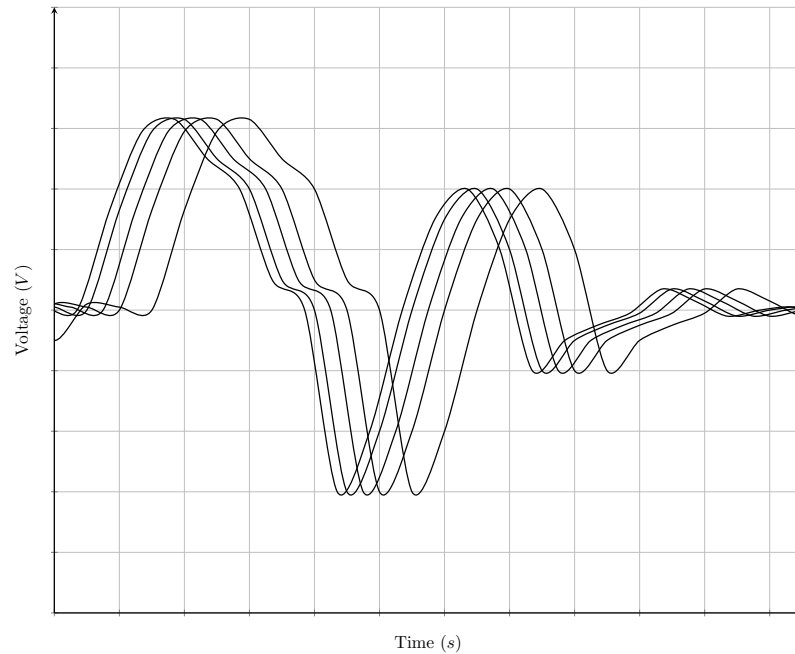


Figure 4.15: Variance in event starting points due to variation in Logo! and oscilloscope (values illustrated in list)

A simple analysis conducted on an entire protocol is used to determine some of the characteristics of these variances; the following list illustrates some timing characteristics of those observed measurements.

- Minimum Start Time: 7.305 ms
- Maximum Start Time: 9.712 ms
- Mean Start Time: 8.643 ms
- Standard Deviation Start Time: 0.651 ms

The clustering of these timings mean that the earliest and latest times when the electrical event is first detected is not larger than 2.407 ms which is a small variance which does not hugely affect the outcomes as observed in chapter 5.

The general process that takes place for each event is as follows:

1. Determine which protocol is selected

2. Determine which event should take place
3. Determine the new relay states for the new event
4. Initialize oscilloscope
5. Time delay (accounting for delays in oscilloscope and Logo!)
6. Activate new relay states on Logo!
7. Wait for oscilloscope to finish measurement
8. Store measurement
9. Return

The first three items are part of the protocol description and process which is discussed in section 4.5. To illustrate this with respect to time fig. 4.16 indicates the coordination timing of the oscilloscope and the Logo!.

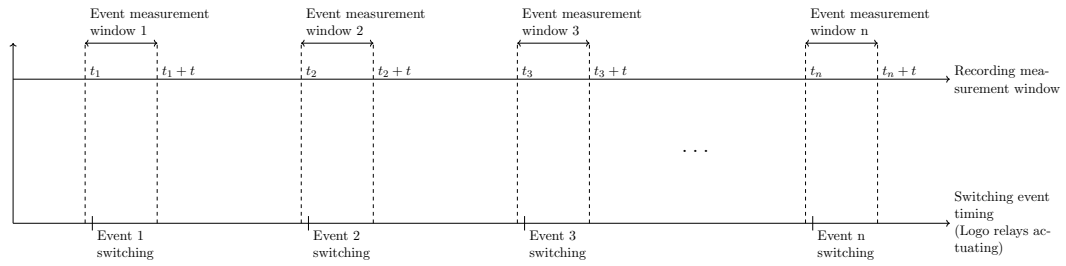


Figure 4.16: Coordination of Logo! and oscilloscope for starting measurement recording and relay actuation

4.5 Testing Protocols

Section 3.3 outline the general requirements that need to be met to enable further work on NILM and load disaggregation within a DC system using the technique outlined in this work. This section outlines the different areas of testing that are created to verify the requirements. The design of these requirements is aligned with practices used in the NILM research field.

A number of previous works designed and implemented similar *protocols* to evaluate different NILM and load disaggregation tasks, these designs were used as part of the structure of the protocols outlined here [53, 61–63, 81, 87, 104, 105].

There are three major parts to the testing to this technique:

Source Testing: A number of tests are designed with different DC sources and different methods of connecting those sources, this is discussed in section 4.5.1.

Load Testing: A number of loads are tested, each in isolation, and then in different combinations and with different sub-tests for each of the edge cases, this is discussed in section 4.5.2.

Unique Testing: This case falls under specific and unique testing procedures that highlight unexpected results or a test that involves a non-standard device, this is discussed in section 4.5.3.

These three major protocol tests groups seek to answer the major research questions; additionally, the measurements taken for a number of these protocols are used to answer the research question relating to the limitations on the sampling rate which is discussed in greater detail in section 4.7.

4.5.1 Source Protocols

The source testing is inherently coupled to the load testing - each of the different source systems may be utilized as a distributed DC network setup. The different combinations of sources are verified against a number of different devices operating.

The distributed DC system can be constructed with a number of unique DC sources (see section 4.1), where there are seven different combinations of creating the DC grid.

These different grid configurations are tested against a number of devices operating within the grid. The tests are created to verify whether it is possible to observe the operation of devices on the grid, the design of the protocols for this section seek to outline the requirements discussed in section 5.1.

In general, this technique is based on a 12 V_{DC} grid, and as such the devices within all tests operate at this voltage. However, due to the design of the devices, they are able to receive a range of voltages around this 12 V_{DC} and still operate as designed. Distinct tests are created with the variable voltage DC power supply and a single light where the test is performed in decreasing voltage steps until the light is no longer able to turn on (see appendix A.2). These tests seek to verify if the NILM

measurement technique is capable of performing over a range of non-standard grid voltages¹⁵.

In addition to the DC bench power supplies used for much of the testing, a different variant of a DC power supply is tested, the observations of this test can be found in appendix A.3. This variant of the power supply makes use of linear regulators. This type of power supply is uncommon in newer implementations as it is inefficient. These types of power supplies, due to their design exhibit much lower noise levels which makes it an interesting grid power supply to observe any marked differences between the newer and older technologies. The higher quality sources are expected to be capable of responding to transient events much better than cheaper variants, making these better power supplies more capable of *suppressing* the response on the grid voltage¹⁶.

4.5.2 Load Protocols

The task of NILM and load disaggregation relies on the fact that different devices exhibit different signatures, as such, a number of different devices (see section 4.2) are run through differing protocols to verify the requirements outlined in section 3.3. The tests created here firstly seek to verify the major requirements of NILM and load disaggregation (see section 5.1) followed by specific test cases.

There are two major test types that are created here to verify the requirements, these are the single device and concurrent device operation.

Single Device Operation: This test seeks to observe the signatures that each device expresses for their different state transitions. These protocols have a single device operating in isolation on the grid. The tests seek to verify the repeatability of the devices transients as well as the uniqueness of the different transitions of a single device. The different devices are listed in table 4.1 which includes both device types as well as a diverse range of device functions.

Concurrent Device Operation: This test looks at cases where there is already a device operating on the grid, then another device changes its state. This is testing to see if it is possible to observe the change of states of a device, while one or more device

¹⁵This is not a major requirement for NILM, rather a unique case to observe the ability to distinguish a device operating at non-standard operating voltages

¹⁶If this is the case, it makes the task of NILM and load disaggregation more complex as the signatures are suppressed.

are already in operation¹⁷. These tests are run with a combination of the two device types to observe the diverse options that may be found in a DC system.

4.5.3 Unique Protocols

The two sections before form the most important tests to verify the NILM and load disaggregation requirements. This section outlines some additional tests performed on the system that seek to answer specific questions about unique cases that may arise within the DC system. These tests and their outcomes are not integral to the end goal of NILM and load disaggregation on such a system, however they may highlight potential attributes of the system that can be utilized in further work to assist in the NILM and load disaggregation research field, the benefits of this information is discussed further in sections 6.1.5 and 6.1.7.

Multiple Replica Devices: It is not uncommon for systems to contain multiple devices that are the same. This means, for example, a light bulb from the same manufacturer, and of the same model number (the same electrical and physical construction). The larger goal of observing two—or more—individual devices on the grid is addressed in the previous sections; this seeks to determine if devices that are designed to be exactly the same, can be distinguished from one another¹⁸. This range of tests uses a number of different techniques in order to attempt to observe the differences in devices, these different techniques and what they present are discussed in section 5.1.3.1.

Multiple Similar Devices: Similar to the case above, it is common that there may be more than one device used in a system that is designed to have the same functionality. A simple example of this is a cell phone charger from two different manufacturers: they are designed to perform the same end function (supply 5 V_{DC} to a cell phone), however, the design of the actual device may vary in small ways. This test seeks to determine if two similar devices have differentiable signatures this is detailed in appendix A.4.

Random Operation: Standard operation of devices on any system are commonly performed in some form of *random order* which represents normal usage of the

¹⁷It is expected that, depending on the device that is kept in one operational state, the noise produced by that device may *hide* the signature of the device transitioning.

¹⁸In the case that two devices are indistinguishable from one another (their signatures are the same), there are other, non-device signature characteristics that NILM techniques can implement to achieve load disaggregation, this is discussed further in sections 5.1.3.1, 6.1.5 and 6.1.7.

system¹⁹. These range of tests, including a sampling of devices, create a switching order that attempts to emulate this random operation of devices on the system.

Concurrent Device State Changes: Systems are commonly constructed in a way such that multiple devices may share the same connection point. A simple example of this is where devices are connected through some form of power strip. In the case that the connection to that power strip is opened/closed, then both devices will have their states changed at the same time. This range of tests observes the signatures created when two lights are turned on and off at the same time, in essence, looking to see if their *combination* transient is different to the lights when observed in isolation, the results of this are observed in appendix A.5.

Current Characteristics: The DC systems used throughout this research are distributed in structure, hence why the research looks to use the grid voltage to see transients. Due to each source being constantly connected (even if not feeding to) the grid, they are inherently linked to any transients that occur. This range of tests makes use of current measurements on one of the sources within the network to see if there are current transients and/or reflections across different parts of the network²⁰. The results of this test can be found in appendix A.6.

Inverter Driven Loads: The availability of DC devices is still increasing; however, this means that there are still cases where there are only AC variants of the device that need to be run. This is where inverters play a role. They enable the user to make use of the AC devices within the system. These tests are created to determine if it is possible to distinguish different AC devices operating behind the inverter. The range of tests observed here are discussed in appendix A.8.

4.5.4 Testing Protocols Implementation

The Python platform, discussed in some detail in section 4.4.3, outlines the control of the oscilloscope and the Logo! to record and change states of the devices respectively. The Python platform is also responsible for accepting the settings for the different

¹⁹In actuality, there is very little randomness of device operation of the system. Lights are commonly only used at night, kitchen appliances during meal times, etc. These patterns form large parts of research in NILM because there is some level of predictability in the state changes of specific types of devices. This is not observed or built into this system as it is out of the scope, and may be used for future work. This is discussed in section 6.1.7

²⁰The current measurement is not placed over a terminal of any of the devices, reason being that as soon as that device is not operating in the network, its loop is completely disconnected and no current will flow.

protocols and setting the system up to carry out the required events and measurements. This is maintained with the use of a database that keeps track of each protocol and every event that needs to take place within that protocol.

There are two major types of protocol functions that are created to enable the different protocols to be carried out: auto-balancing, and concurrent.

Auto-balancing: This function is used for the majority of the tests carried out, the requirement that this function accomplishes is to manage the different devices operating within the system, keep track of the number of events assigned to each device and keep creating and executing events until the required minimum number of events has been accomplished.

Concurrent: This function satisfies the protocols that require specified devices to be operating in set states while another device is allowed to carry out its switching. This function is created for the concurrent device operation protocols.

The different protocols, containing different devices, and different settings for how those devices can interact, are all managed by this Python platform. Figure 4.17 illustrates what happens under each event in a protocol that contains three devices: two devices of type I, and one device of type II which are devices A, B, and C. This process will continue until it has satisfied the minimum device state change criteria, at which point, the protocol has completed and the next protocol can start.

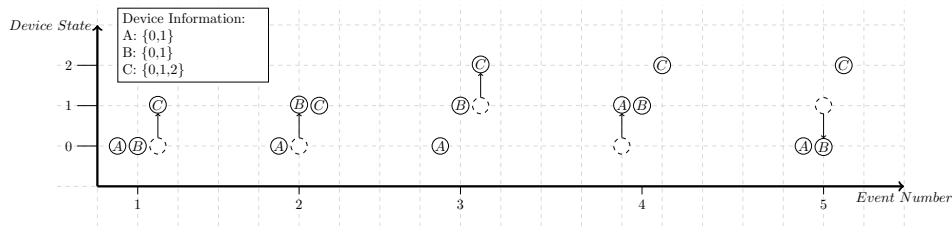


Figure 4.17: Example Protocol

The number of events on a device level is important to make sufficient conclusions about the observations of the system, the design and reasoning for this is discussed in section 4.5.5.

4.5.5 Sampling Considerations

The methodology implemented in this work, particularly sections 4.8 and 4.9 require that—in order to provide useful results and to accurately verify these results while

being statistically relevant—each unique transition of each device in a protocol needs to have a minimum number of events.

This means that, for example, in the case of a protocol with a single device of type I, there should be enough events to safely create a model and signature collection that accurately represents the two state transitions. Research in this field, making use of a similar testing technique varies in the minimum number of events required, ranging from five events upwards. There is a trade-off made in this number: time, convenience, and statistical power. This is linked to the law of large numbers [106, 107]. This law results in a case where, as the number of samples increases, the average of these results will converge on an expected value. Additionally, increase in the number of samples will reduce the effect of random events skewing the overall results, this increases the stability of the result obtained.

However, the more events required, the higher the storage space required (see section 4.6)²¹, the longer each protocol will take as each event takes a non-zero amount of time; in addition to this, increasing the number of event measurements will eventually get to a point where any further measurements do not assist in verifying the results: they simply become redundant.

Standard statistical analysis of many systems always hint that a larger number of observations is better - meaning that randomness in the system will be reduced with each consecutive sample. However, this has diminishing returns.

If the number specified for minimum events is too few a situation will result where the events and the conclusions made from those events may misrepresent the system. The more data samples, the more reliable and safe conclusions will be as the data sufficiently represents what is occurring within the system.

The important point here is to make sure that the number of samples allows for sufficient statistical power. The standard approach used in statistics makes use of specific sample size determinations by making use of specifying target variances, a target power of a statistical test, or using a confidence interval. The nature of this type of research is that these types of tools cannot be readily applied, this is seen in research in this field where it is often better to make the judgement based on the research space. The result is that it often comes down to subjective judgement, and in this case domain knowledge which has been used in previous research in this field that uses similar methodologies [11, 108, 109].

²¹Each extra event requires more storage space, computation time to extract features, filter, normalize, and cluster.

In carrying out this research, when looking at the analysis of the results, it is clear that as the number of events increases (above what is used in this study), each consecutive measurement appears to be redundant as each measurement is highly correlated with previous measurements. This is what the aim of the system is looking for, to obtain a sample size for each test that achieves satisfactory information redundancy and saturation of data[110].

This research makes use of a minimum number of 20 events which is high enough to not make false conclusions on too few data, but it also does not greatly increase the machine learning stage computational requirements (and protocol measurement stage) to become untenable, this number of events is also above the minimum number of samples that are used in other work. This means that this work, having a higher number of samples, has a larger *safety margin* when compared to other studies, making this a safe value.

4.6 Data Processing System

The data from each measured event is transferred from the memory of the USB oscilloscope through to the PC. The consequence of each event containing ten million samples results in the system built around the data storage - a little more complex than lower data requirement systems. Early in the research, comma separated value (CSV) documents were used to hold the data, however, this proved to be inefficient, the memory required to load each event in and out of memory was not feasible, in addition to this, each event took up approximately 250 MB; the number of protocols and the minimum number of events required for each protocol implies that many events will be taken throughout the research. These storage requirements were not maintainable and highly inefficient for the data analysis stage.

The use of a binary format file type is adopted throughout this work, the Hierarchical Data Format Version 5 (HDF5) standard provides a highly compatible and efficient method of storing arrays of such magnitude while keeping the overall file size for each event much smaller (~25 MB) and decreasing the time taken to load the events in and out of memory. This file format is used throughout the research where data is stored, this is with features, downsampled versions of data, clustering results, etc. This technique of data storage is utilized in other NILM research as it is an efficient method of storing large datasets for simple access [37, p. 103]

The data stored within these files is stored on the PC, and the information (the

metadata) about each event is stored in the database (see section 4.3).

This database, the ability to keep track of all settings and configurations used throughout the research, enables the capability to keep track of each protocol, each event in each protocol, and what happened in each event. This forms the basis for the construction of the ground truth data. This ground truth data is relied upon for the data analysis and validation steps, this is important as it allows the system to obtain performance metrics of the machine learning stage to gauge the ability of distinguishing devices in a protocol and hence in the system. The database keeps track of the following pieces of information of each event.

- EventID - The unique ID corresponding to the event that took place.
- ProtocolID - The unique ID of the protocol that this event corresponds to.
- DeviceID - The unique ID of the device that transitioned during the specified event.
- DeviceType - The type of device that is switched (type I, or II).
- PreviousState - The state of the device prior to this event.
- CurrentState - The state of the device that it will transition in to.
- EventStartTime - Information of when the event took place in time.
- RelayStatesVector - The state of each of the relays for the new current status of the system (what the system looks like after the event has taken place).

The measurements taken for each protocol are stored and are seen as the raw data recorded for each event. The database and how it interacts with the Python platform are integral to the entire research methodology; it is used throughout the next stages of the data analysis process, in addition to this, each of these events is replicated into multiple downsampled versions of their originals, the details of the downsampling and its tests are discussed in section 4.7.

4.6.1 Ground Truth

The methodology utilized thus far, the controlled event switching and measurement, the integration of the database, are the components of the system that enable for the ground truth data to be recorded and utilized as a validation mechanism in this

work. This work makes use of the term *ground truth* to represent the metadata that is collected throughout the event process.

This information is utilized in two major steps in the research:

- Determining the features that best represent the operation of the system.
- Performing a critical analysis and validation of the performance of the technique in the clustering process.

This ground truth, illustrated by an example protocol in table 4.4 where a single device (the light bulb) is part of a protocol, this illustrates the level of detail recorded throughout each event. The details of each of these columns are listed in section 4.6.

The techniques of feature selection and clustering (see sections 4.8.3 and 4.9) utilize the information within this database structure.

The purposes of this ground truth require it to be in a specific format, thus allowing for the further components of the system to utilize the information. There are two major types of ground truth data that are thought to provide insight into the functioning of the system. These two types of ground truth are created based on two major assumptions which seek to provide the feature filtering step and the clustering performance evaluation steps with information that allows them to understand which event corresponds to which category, cluster, or class.

The example protocol illustrated in table 4.4 is used here to illustrate how the simple ground truth method functions; the second - the complex ground truth idea is applicable to this example, however, its ability is highlighted with the use of a further example where more than a single device is used in a protocol.

4.6.2 Simple Ground Truth

Taking the example illustrated in table 4.4, where a single device is operating, it has two states: ON and OFF. Consequently, this device has two state transitions that it can perform:

- Device 1: OFF $\rightarrow$ ON.
- Device 1: ON $\rightarrow$ OFF.

Table 4.4: Ground Truth for Example Protocol

Event ID	Protocol ID	Device ID	Device Type	Previous-Current State	Event Start Time	Relay States Vector
1	1	1	I	0-1	2020-05-06 17:02:40.4	[1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
2	1	1	I	1-0	2020-05-06 17:02:51.5	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
3	1	1	I	0-1	2020-05-06 17:03:02.3	[1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
4	1	1	I	1-0	2020-05-06 17:03:12.9	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
5	1	1	I	0-1	2020-05-06 17:03:23.7	[1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
6	1	1	I	1-0	2020-05-06 17:03:34.5	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
7	1	1	I	0-1	2020-05-06 17:03:45.3	[1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
8	1	1	I	1-0	2020-05-06 17:03:56.2	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
9	1	1	I	0-1	2020-05-06 17:04:07.0	[1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
10	1	1	I	1-0	2020-05-06 17:04:18.6	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
11	1	1	I	0-1	2020-05-06 17:04:29.6	[1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]

This implies that this protocol is likely to see two distinct types of transient events corresponding to the two transition types of the device. Note that this is fictionally illustrated in fig. 4.18 where the shape and position of the *clusters* of event measurements highlight what the expected two types of clusters would form with this estimation of ground truth. The clusters illustrated in fig. 4.18 are also formed with two fictional features of the transient signatures to illustrate how the two transient state changes are grouped.

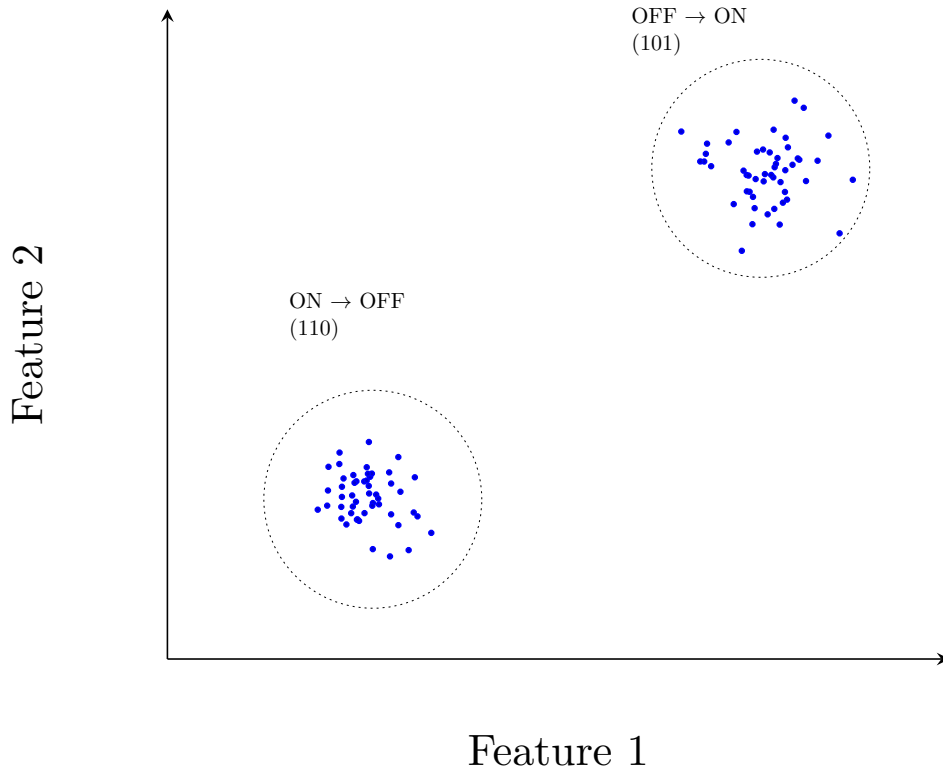


Figure 4.18: Scatter plot of event feature vectors (two example features) - single device type I protocol

This implies that events 1, 3, 5, ... (the odd events in this case) represent the points in the cluster on the right, and the events 2, 4, 6, ... (the even events in this case) represent the points in the cluster on the left. This means that there are two clusters that are formed by this type of protocol: the cluster on the left representing the signatures for *ON* to *OFF* state changes, and the cluster on the right for the *OFF* to *ON* state changes.

This ground truth system also applies to cases where there are multiple devices which are part of the protocol, table 4.5 illustrates the switching events of two devices in a protocol. This protocol, due to the number of devices present, produces four distinct transients from the two different devices in the system:

Table 4.5: Ground Truth for Example Protocol

Event ID	Protocol ID	Device ID	Device Type	Previous-Current State	Event Start Time	Relay States Vector
1	14	2	I	0-1	2020-05-07 10:15:29.6	[0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
2	14	2	I	1-0	2020-05-07 10:15:40.2	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
3	14	2	I	0-1	2020-05-07 10:15:50.8	[0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
4	14	1	I	0-1	2020-05-07 10:16:01.6	[1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
5	14	2	I	1-0	2020-05-07 10:16:12.3	[1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
6	14	1	I	1-0	2020-05-07 10:16:23.0	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
7	14	2	I	0-1	2020-05-07 10:16:33.9	[0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
8	14	1	I	0-1	2020-05-07 10:16:44.7	[1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
9	14	2	I	1-0	2020-05-07 10:16:55.7	[1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
10	14	1	I	1-0	2020-05-07 10:17:06.5	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
11	14	2	I	0-1	2020-05-07 10:17:17.3	[0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
12	14	2	I	1-0	2020-05-07 10:17:28.0	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
13	14	1	I	0-1	2020-05-07 10:17:38.7	[1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
14	14	2	I	0-1	2020-05-07 10:17:49.3	[1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]

- Device 1: OFF $\rightarrow$ ON.
- Device 1: ON $\rightarrow$ OFF.
- Device 2: OFF $\rightarrow$ ON.
- Device 2: ON $\rightarrow$ OFF.

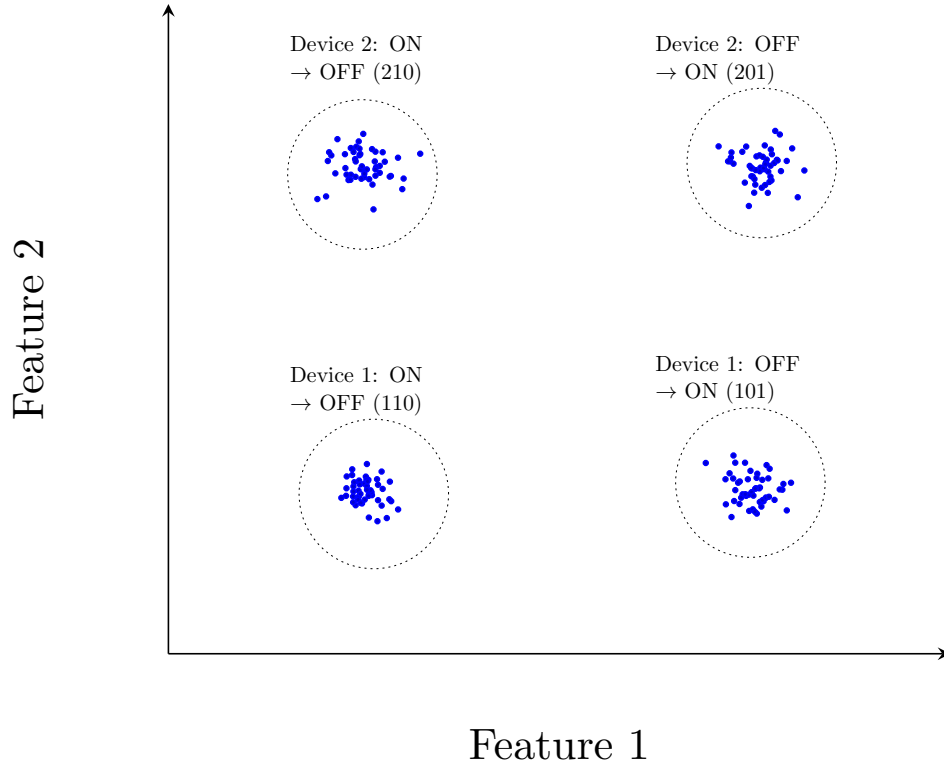


Figure 4.19: Scatter plot of event feature vectors (two example features) - two devices of type I

This simple ground truth case makes certain assumptions in how the system is performing and how the devices affect the grid voltage for the different devices' state changes. This simple case assumes that each device has its own distinct transients for its different state transitions; when further devices are added to the system, this simply increases the number of expected *clusters* by the number of state transitions of the new device.

To understand this, the number of expected clusters for a protocol is based on the number of type I devices in the protocol (x) and the number of type II devices in a

protocol (y)²², the number of clusters here can be simply defined by eq. (4.2).

$$N_{Clusters} = 2 \cdot x + 6 \cdot y \quad (4.2)$$

4.6.3 Complex Ground Truth

This second type of ground truth makes a further assumption on how the system observes transients on the system. Specifically, in the case that a device transitions its state while another device is already in an operational state, the response observed on the grid will be augmented. This means that the assumptions of the simple ground truth only hold for specific cases; those cases where there is a single device operating, or in cases where multiple devices are in the same protocol, however, none of the devices can be in any of their ON states when a device transitions.

This harkens back to the original work in NILM where the *new* state of the system is partially dependant on the state of the system at the moment before [1, p. 1870]. This means that inherently, the NILM system needs to know what the state of the system was at before the device switched (including states of the other devices within the system), and the new state that the system is now in, to determine what this means about the new state of devices within the network.

This type of ground truth is making an assumption that when the grid already has operating devices, any consequent transition of a device is now dependent on this specific state of the system.

This increases the complexity of the ground truth, in essence, it increases the total number of *unique* transients that may be observed on the system with each additional device in a protocol. This assumption is discussed in further detail, and the reasoning behind it in section 5.2.

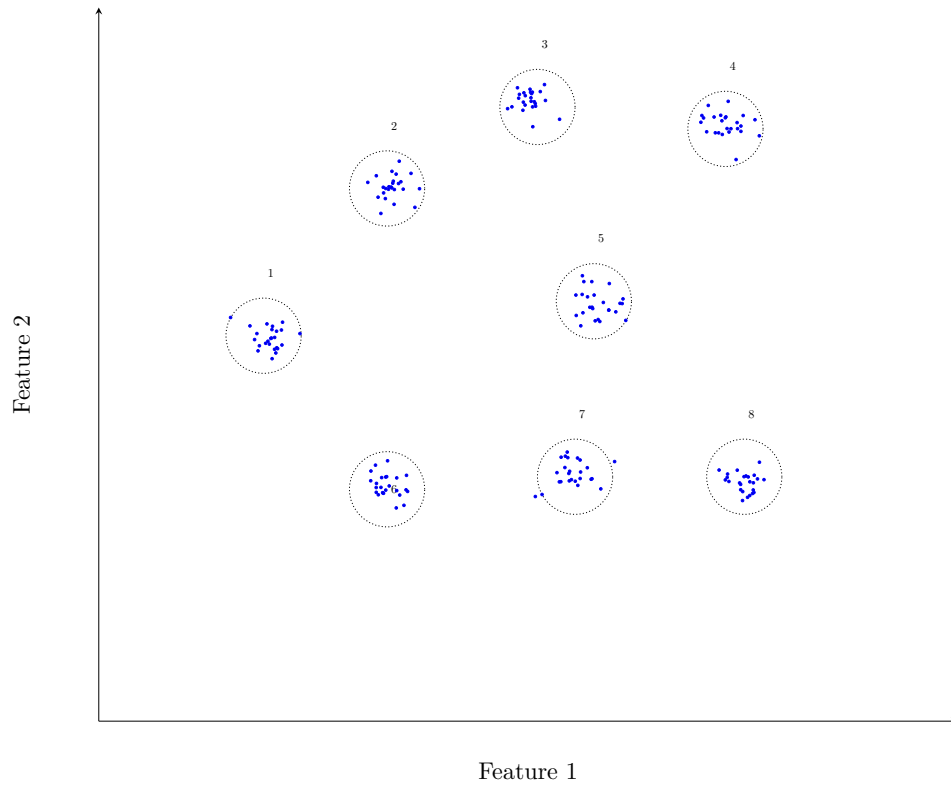
The protocol illustrated in table 4.4 will not change in this case as it represents a protocol with only one device in operation. The protocol in table 4.5 is relevant to this complex ground truth case as it has two devices where the two devices also operate in a *random* fashion. This protocol, and the different *system state changes* can be illustrated in table 4.6 which highlights the increased number of classes/clusters that may be observed.

²²For this situation, type II devices are assumed to have only three states

Table 4.6: Complex ground truth transient signature classes

System Transition	Device Stationary	Device Stationary State	Device Transitioning	Device Transition
1	1	ON	2	OFF $\rightarrow$ ON
2	1	ON	2	ON $\rightarrow$ OFF
3	1	OFF	2	OFF $\rightarrow$ ON
4	1	OFF	2	ON $\rightarrow$ OFF
5	2	ON	1	OFF $\rightarrow$ ON
6	2	ON	1	ON $\rightarrow$ OFF
7	2	OFF	1	OFF $\rightarrow$ ON
8	2	OFF	1	ON $\rightarrow$ OFF

This means that for a system with two devices, each of device type I, the system is capable of observing eight possible switching transient types. A simplistic illustration of this case can be observed in fig. 4.20 where the cluster numbers are those used in table 4.6.

**Figure 4.20:** Complex ground truth clusters of two device protocol signatures

4.7 Downsampling

The aim of this component of the platform is to create a system whereby the high sampling-rate measurements recorded for each protocol are converted into multiple, lower sampling-rate versions of the original measurements. The purpose of this is discussed in section 3.5 which highlights the requirement to test the overall methodology on a range of different downsampled versions of the original data such that a comparison can be drawn between a sampling rate and the ability to observe devices operating on the network.

The ability to provide a comparison on the performance of the methodology at different sampling rates requires data sampled over a range of different sampling rates. As mentioned in section 3.5, it is expected that as the sampling rate is decreased, the number of differentiable devices and events observable will decrease. It is unlikely that there will be a single sampling rate where, if it was to be any lower, then all devices would not be visible, and any higher, all devices will be visible. This is discussed in section 5.4 where it is highlighted how different devices impact the network based on their functionality and power rating.

As such, it is expected that different sampling rates will allow for different types (magnitudes) of *events* to be visible on the network. This therefore implies that, in order to determine the *best possible* sampling rate—that makes a trade-off between equipment cost, processing time, and accuracy (see fig. 3.5)—a range of different sampling rates should be compared against one another. There will be a point at which any increase in the cost of the system—measurement equipment, storage space, and computational requirements—that there will be diminishing returns on the performance. This performance limitation is observed throughout this research field, it is precisely this limit (which is more of an asymptote) that the researchers attempt to increase. This section seeks to provide an understanding on the optimal system that can *sufficiently* function with the capabilities commonly expected in this field [111].

To test these limits on the system at hand, to apply it within the methodology, there are two ways in which these different sampling rates can be tested:

- record each protocol (and therefore each event) with different sampling rates set on the oscilloscope or;
- convert the original measured samples to lower sample-rate versions of the

original.

The first method is limited by the capabilities of the measurement equipment used for this work (the Picoscope), it has a set range of possible sampling rates that it can sample at, and it can only select one setting for each event measurement.

Currently, the oscilloscope allows for the sampling rates to be used that are defined by eq. (4.3).

$$\text{Timebase (sampling period, } T_s) = \begin{cases} \frac{2^n}{80,000,000}, & n \in 0, 1, 2 \\ \frac{n-1}{20,000,000}, & n \in 3, \dots, 2^{30} - 1 \end{cases} \quad (4.3)$$

This means that each event will need to be recorded for each of the selected sampling rates required. This method has a number of limitations:

- The system needs to record each event at the number of different sampling rates that the test requires, this results in a large amount of time spent measuring the same event multiple times which scales up with the number of protocols, and the number of events in those protocols,
- the sampling rates that are possible are limited by the manufacturers API's, which limit the range of possible sampling rates selectable,
- this method greatly increases the switching events of each device, increasing the chance that the device may fail, and result in that device no longer being part of the range of tests possible, and
- each event of a device is likely to be slightly different each time, thus not every event measurement at different sampling rates will be readily comparable to the same event of a different sampling rate.

A positive characteristic of this method is that the oscilloscope samples at the exact required rate, creating an exact representation of what would happen if sampling at the different rates.

The second method overcomes the issues of the first method in that it only requires each protocol to be run once, measuring each event once, and recording at the highest possible sampling rate afforded by the oscilloscope in this setup (20 MS/s).

This second method makes use of signal processing techniques to downsample the original high sampling-rate measurements to a lower sampling-rate version of the original. In other words, the process, when applied to a recorded event, creates a replica of the original event as if it had been recorded with the downsampled rate selected. The general process that this follows is illustrated in fig. 4.21 where the original event is replicated into a number of downsampled versions.

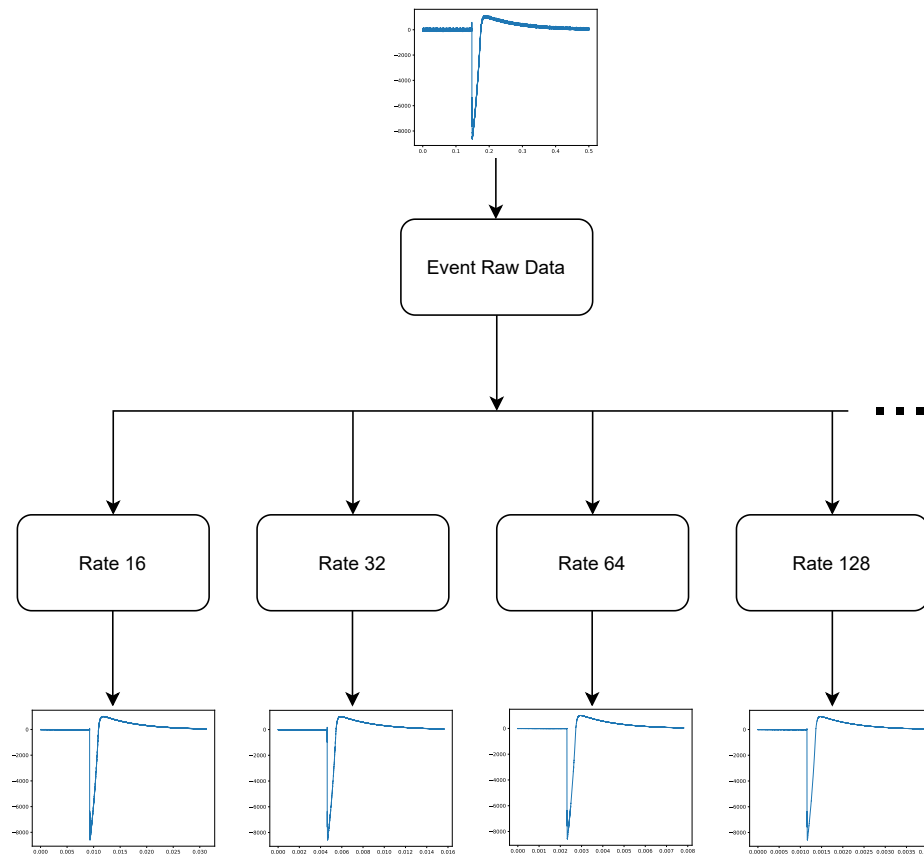


Figure 4.21: Visual representation of different downsampled versions of an event

There are three methods in which this can be accomplished, each with varying degrees of accuracy. The first method simply removes every n th sample from the original signal, such that the remaining samples represent a lower sampling-rate measurement of the original measurement. However, this is a naive approach to downsampling the data as this method loses information and does not take into consideration how the measurement was originally recorded. The problem with this approach is that the high-frequency components present in the original measurement can distort the result in the downsampled versions as this introduces aliasing to the conversion.

To overcome this issue, the following two methods make use of a low pass filtering step before removing samples from the signal. This step is band limiting the original

signal (such that anti-aliasing is not introduced), and then downsampled to the required rate. The two methods employed in this case can be found from Python's Scipy package which provides a number of methods in which to downsample time series data [112, 113].

4.7.1 Downsampling Theory

The process begins by taking the original signal recorded, represented here by $x[n]$ which is the discrete-time representation of the original continuous time signal $x_c(t)$, this is defined by eq. (4.4) where T represents the sampling period [114, p. 167].

$$x[n] = x_c(nT) \quad (4.4)$$

This signal, $x[n]$, is the raw data recorded by the oscilloscope at the original 20 MS/s sampling rate. This signal is used to create each of the different downsampled signals which are used throughout.

The downsampling process is set to create the new downsampled version, $x'[n]$, of the original signal, defined by eq. (4.5) where T' (the higher sampling period) is a set multiple of T [114, p. 167].

$$x'[n] = x_c(cT') \quad (4.5)$$

Of major importance in this process, is to ensure that the new downsampled version of the signal, $x'[n]$, is an exact (or as close to exact) replica of the original signal $x_c[nt]$ when sampled at the lower sampling rate of T' .

Generally, the process that this system follows is illustrated by fig. 4.22 where a downsampled signal $x_d[n]$ can be obtained. The downsampling rate M represented here can be defined by $T' = MT$.

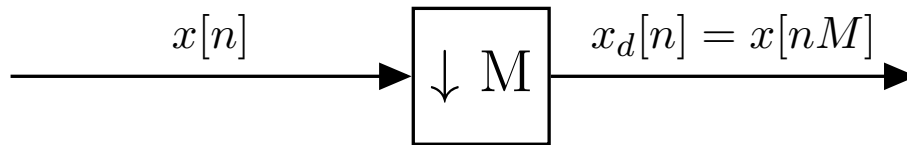


Figure 4.22: Downsampling process for an event [114, p. 167]

This process is used for each event in each protocol such that the system can generate downsampled versions of each protocol. This process is repeated for all of the different downsampling rates selected (see section 4.7.4).

Two different methods are employed to obtain the different downsampling rates required, the main difference between the two methods being that the first one in section 4.7.2 can only downsample to integer values of the original, whereas the second method (see section 4.7.3) can make use of downsampling rates with real numbers. These two methods are discussed in sections 4.7.2 and 4.7.3.

4.7.2 Integer Factor Downsampling

This process of downsampling contains two steps [115]:

1. Remove the high-frequency components with a lowpass filter.
2. Take every M^{th} sample from the intermediate signal.

The first step, in essence, limits the bandwidth of the signal so that aliasing is not presented in the output of the process, then the second step simply takes care of the removal of elements from the band limited signal.

This process only allows for integer downsampling as it is the value of M in the second step that accounts for the downsampling rate required, which can only be an integer as it selects every M^{th} sample from the signal. This method can be represented by eq. (4.6).

$$T' = M \cdot T \tag{4.6}$$

The second method in section 4.7.3 allows for rational factor downsampling, however, it introduces a little more complexity as it has three steps.

This method of downsampling is implemented with SciPy's `decimate` function [112].

4.7.3 Rational Value Downsampling

This method involves three major steps [115]:

1. Resample the original signal by interpolating by a specified factor L (upsampling step)
2. Remove the high-frequency components with a lowpass filter.
3. Take every M^{th} sample from the intermediate signal.

This process upsamples the signal, thereby increasing the sampling frequency by a factor L , then filters it with a lowpass filter, and finally, much like in section 4.7.2, it then decimates the signal by a factor of M . This method has the capability of downsampling a signal with rational values as the downsampling rate can be determined by L/M . The choice of these values will determine the resulting sampling rate. This method allows for more precise downsampling rates to be selected, it also enables the ability to downsample at rates less than two, which is the minimum that the previous method allowed.

This implementation can be represented by equation eq. (4.7).

$$T' = \frac{M}{L} \cdot T \quad (4.7)$$

This method of downsampling is implemented with SciPy's `resamplepoly` function [113].

4.7.4 Downsampling Rates

In order to analyse the dependence on sampling rate on the ability for the system to distinguish between different devices and device transitions, a system is required that is capable of observing the change in accuracy in the system with a number of different sampling rates. This section aims to compare different downsampling rates that are included as part of the oscilloscope's original system, and using these different rates to create a similar implementation that is capable of observing the trend with different sampling rates.

Standard oscilloscopes are constructed with a range of different sampling rates that are user selectable, and once selected, will set the oscilloscope to measure with that specific sampling rate for a period of time.

The Picoscope allows for the following sampling rates to be selected:

- 80 MS/s,
- 40 MS/s,
- 20 MS/s,
- 10 MS/s,
- 5 MS/s,
- 2 MS/s,
- 1 MS/s,
- 500 kS/s,
- 200 kS/s,
- 100 kS/s,
- 50 kS/s,
- 20 kS/s,
- 10 kS/s,
- 5 kS/s,
- 2 kS/s,
- 1000 S/s,
- 500 S/s,
- 200 S/s

The sampling rates selectable are clearly discrete and predefined where the difference between each consecutive option has the sequence defined by the sequence of $1 - 2 - 5$, where the lowest selectable sample rate is 200 S/s and the last selectable rate is that of 80 MS/s²³. These series are generated by a constant factor between two consecutive values, where rounding is done beforehand the sequence can be illustrated by eq. (4.8), these are known as preferred numbers [116].

$$\dots 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100, \dots \quad (4.8)$$

The result of this sequence of options upon which to sample implies that each successive sampling rate is approximately a doubling of the previous one, this is illustrated via fig. 4.23 where the blue lines represent the options selectable (the two figures illustrate the options in linear and log scale).

The selection of this range is done such that each successive element is spaced approximately equally on the logarithmic scale such that a decade can be covered in three items in the sequence.

A similar concept is applied in this work to create the different downsampling rates as this provides the ability to test a wide range of sampling rates without the number of options becoming infeasible. A solution that may perform well for downsampling follows the Renard series [116] which is used to expand the $1 - 2 - 5$ series by using

²³Note that the highest selectable sampling frequency in this list is 80 MS/s, as this is the highest selectable frequency usable in the Picoscope software, whereas the API only allows up to 20 MS/s.

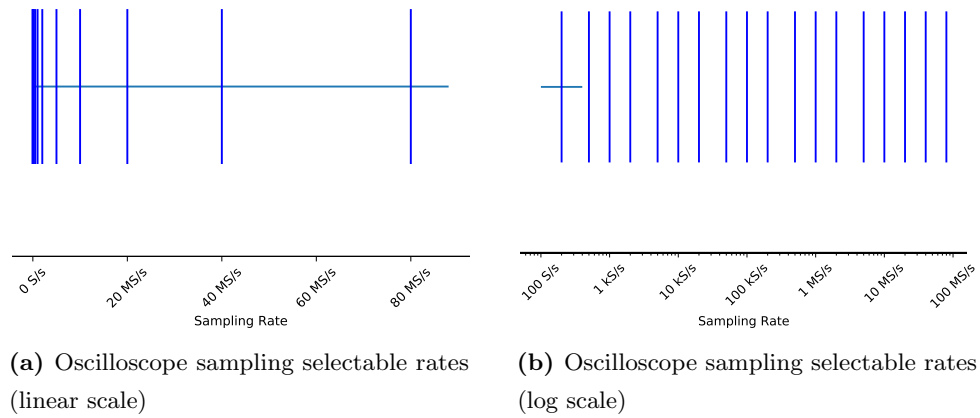


Figure 4.23: Oscilloscope sampling rate options

a smaller increment between each item. The Renard series used is the *R10* series, this allows a decade to be covered by ten items.

Making use of this method creates a list of sampling rates illustrated by fig. 4.24.

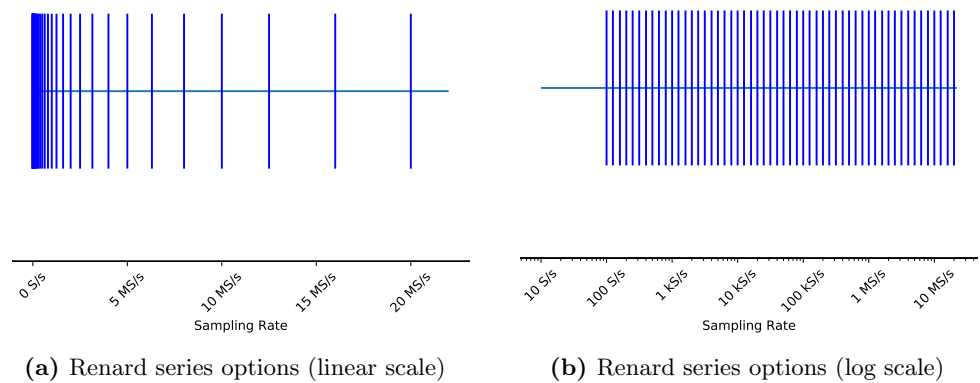


Figure 4.24: Generated Renard series sampling rate options

The downside of making use of this method is that to get down to the 1 kS/s level, it would include 54 elements to create downsampled versions for, this is currently unfeasible to utilize in this work (the computation time and storage space quickly scale up).

Thus, a compromise is taken here, such that the downsampling rates selected, starts with a downsampling rate of 16 (corresponding to a downsampled rate of 1.25 MS/s) and doubling each consecutive downsampling rate down to 2 097 152 (corresponding to

a downsampled rate of ~ 9.53 S/s). Additional downsampling rates are implemented during the analysis stage as it is observed that there are specific regions of interest in the sampling rate (these are selected based on observations for where large changes in accuracy occur); this accounts for the additional rates listed below:

• 16	• 16 384	• 212 992	• 786 432
• 32	• 24 576	• 229 376	• 851 968
• 64	• 32 768	• 245 760	• 917 504
• 128	• 49 152	• 262 144	• 983 040
• 256	• 65 536	• 294 912	• 1 048 576
• 512	• 81 920	• 327 680	• 1 179 648
• 768	• 98 304	• 360 448	• 1 310 720
• 1024	• 106 496	• 393 216	• 1 441 792
• 1536	• 114 688	• 425 984	• 1 572 864
• 2048	• 122 880	• 458 752	• 1 703 936
• 3072	• 131 072	• 507 904	• 1 835 008
• 4096	• 147 456	• 524 288	• 1 966 080
• 6144	• 163 840	• 589 824	• 2 097 152
• 8192	• 180 224	• 655 360	
• 12 288	• 196 608	• 720 896	

The two corresponding linear and log downsampled rates are illustrated in fig. 4.25.

This list is a compromise taken that allows for the analysis of different downsampling rates from the original sampling rate down to ~ 10 S/s which is capable of highlighting the degradation in ability of distinguishing the operation of devices.

4.7.5 The Utility of Downsampling

The process that this research takes, measuring the analog signal, converting it into a digitized form, is standard among all NILM approaches. However, as discussed

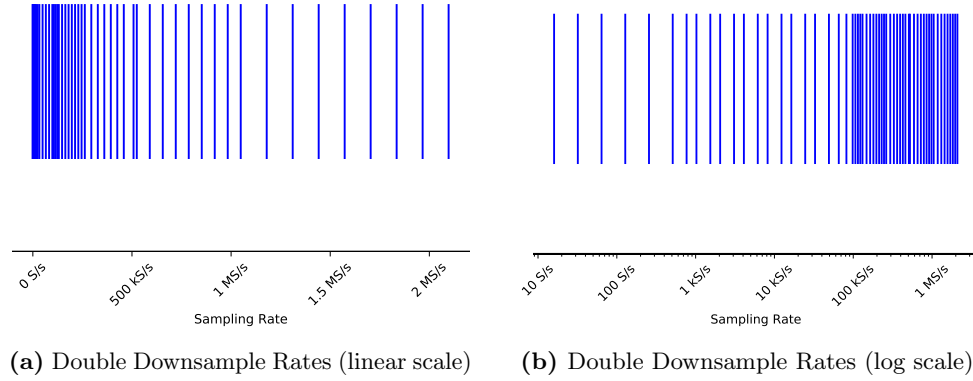


Figure 4.25: Doubling Downsampled Rates

in section 3.5, not all research in this field makes use of the same sampling rate for a similar methodology. It is these differences in sampling rates that determine the performance ability in such systems. The high sampling rate used in this research for the *raw data* is unlikely to be required for most systems, in addition to this, the measurement device used here is unlikely to be priced to make it a feasible NILM system. This process is also attempting to find an optimal sampling rate that can function for most situations while keeping the cost manageable.

This research spends a non-negligible amount of compute power in creating all of the different downsampled versions of each event (which in turn increases the computation required for feature extraction, feature normalization, etc.). Any finished product will only sample at the required rate because at that point the optimal system is created based upon the understanding that this research generates - from the sampling rate to the best features that best represent the operation of devices, etc. This idea also applies to the resolution of the measurement device; however, this analysis is not touched upon in this research.

4.8 Feature Engineering

The sections leading up to this lay out the processes that take place in order to record the unique signatures that are found as transients on the grid voltage of the system when it operates under a range of different conditions. These measurements take the form of time series representations of the grid voltage, each over a specific period of time (0.5 s). These measurements—for the most part—represent the grid

voltage making use of AC coupling (see section 4.4.2).

Section 2.7 discusses the techniques of feature extraction on data and how it can be used to extract useful information about the underlying signal. There are a number of methods that are commonly employed when applying feature extraction techniques on time series data. The simplest method is to simply make use of the time series data directly, take each sample of each event as a feature, and then pass this on to the analysis tool; this technique is a naive approach as it is highly resource intensive and often the analysis techniques are not able to obtain meaningful results out of this raw data. There are a further two types of methods that are commonly utilized when tasked with clustering time-series data, namely feature-based and model-based [117, p. 1859]. These different approaches, and their processes are illustrated in fig. 4.26.

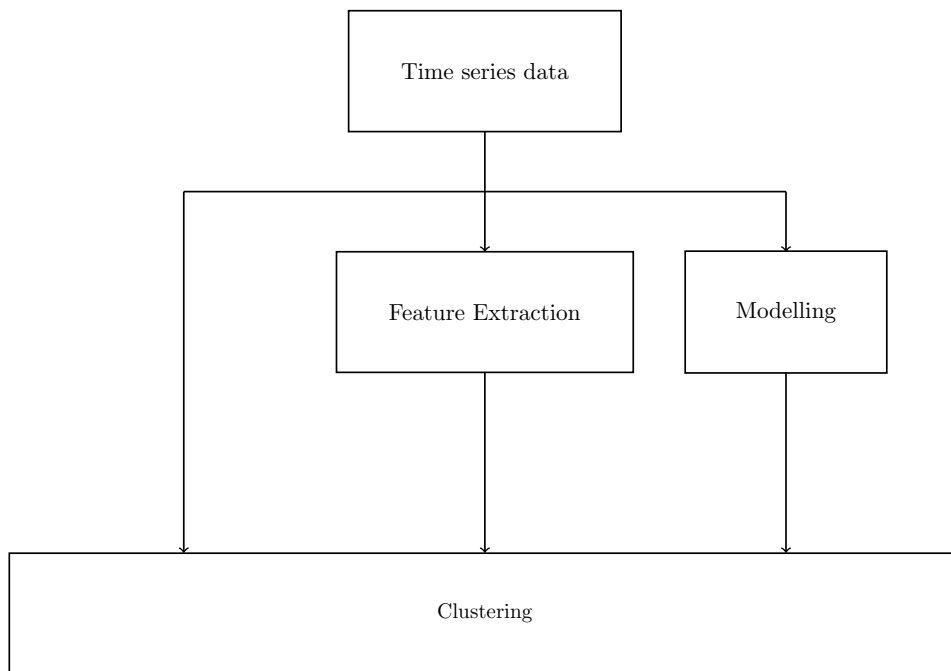


Figure 4.26: Common approaches to time-series clustering

Already having ruled out the raw-data based clustering methods, the last two types are still possible methodologies to utilize. This research makes use of feature-based clustering, reason being that current research in this field also makes use of this technique and it is readily applicable to the first sections of the methodology; additionally, the use of feature-based approaches is a generalized approach which requires less up front cost than model-based parameterizing requires [118, p. 19-20]²⁴.

²⁴Model-based approaches often require a good understanding of the structure of the system. This research, the fact that it is looking to understand the structure of the system, is not so well-suited to this method [119, p. 61]. Determining the underlying models of the *events* is also not the focus of

Further than this, there is a wide range of highly capable feature extraction tools that are available (within the problem domain) that are particularly useful for time series feature-based clustering.

The feature extraction and engineering process is useful as it is used to create a lower dimensional representation of the original waveform such that each dimension is represented by a feature and is a characteristic about the time series dataset.

The goal of this feature engineering section is to create a system whereby these events and their corresponding protocols can be placed into groups based on their attributes.

It is these features that allow for the grouping and categorizing process to take place. The choice of these features, and the level to which these features represent the underlying data is most important.

This means that a univariate time series needs to be defined by a metric, or a number of metrics to show the dissimilarity/similarity between different sets of these series (in this case, each set is the data from an event).

Consequently, the major task in this section is to determine the optimal metrics in which to represent these time series data sets. There are a number of common techniques that are used in characterising time series data, in addition to this, the time series data can be converted into the frequency domain where additional features are easily recognizable and usable as metrics²⁵.

This results in a number of features (which can form a feature vector) that represent the underlying raw data, these feature vectors are then carried forward into clustering processes in order to find underlying groupings. This is known as feature-based analysis where the features represent the time series data fed through to the clustering processes, hence representing the data in a lower dimensional format.

This highlights the fact that it is important to find as many features as possible, while attempting to also find features that best represent the underlying data. These steps are each outlined in sections 4.8.1 to 4.8.3 and this general process is illustrated by fig. 4.27.

this research.

²⁵Converting time series data into the frequency domain is a common technique that adds additional information about the dataset.

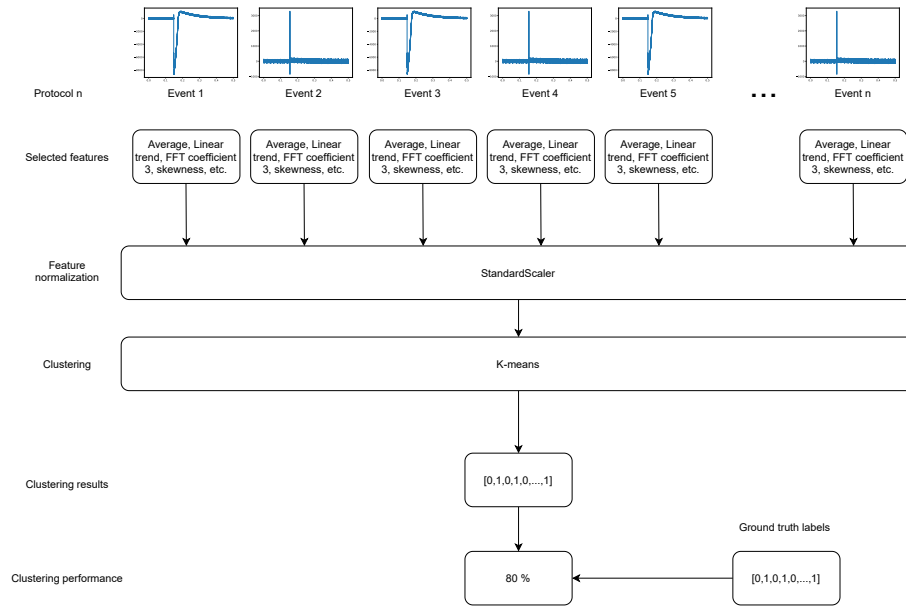


Figure 4.27: Overall process for feature extraction, normalization, and data clustering

4.8.1 Feature Extraction

The feature extraction process often forms a large part of NILM research, making use of a number of different features ranging from domain specific features to features that highlight specific characteristics of the data. Some of the domain specific features include: active and reactive power, harmonics (FFT coefficients), noise, RMS, etc. The fact that the transient information used in this work makes use of a high sampling rate, this increases the number of features that can be use as additional information from the data²⁶.

The resulting capability of such measurements means that this feature extraction process is able to obtain a large range of features from the data. Some research makes use of specifically constructed feature extractors [37, 44]. The drawbacks of these methods is that they rely on deep understanding of the clustering processes and how they might perform on different features; in addition to this, it requires understanding of which features may provide features that best represent the underlying data. This often leads to methods that employ one or two major types of features in the research.

However, a number of tools are currently available, and are used within this space that enable the automation and simple extraction of many features of a dataset that span a number of different domains [120, 121]. This enables the ability to construct a pipeline such that raw data is fed into the platform, and feature vectors are output

²⁶Take note about the fact that this does increase the cost of the system, the computational cost.

which represent the dataset now in a lower dimensional feature space. Figure 4.28 illustrates a few simple features that can be extracted from an event.

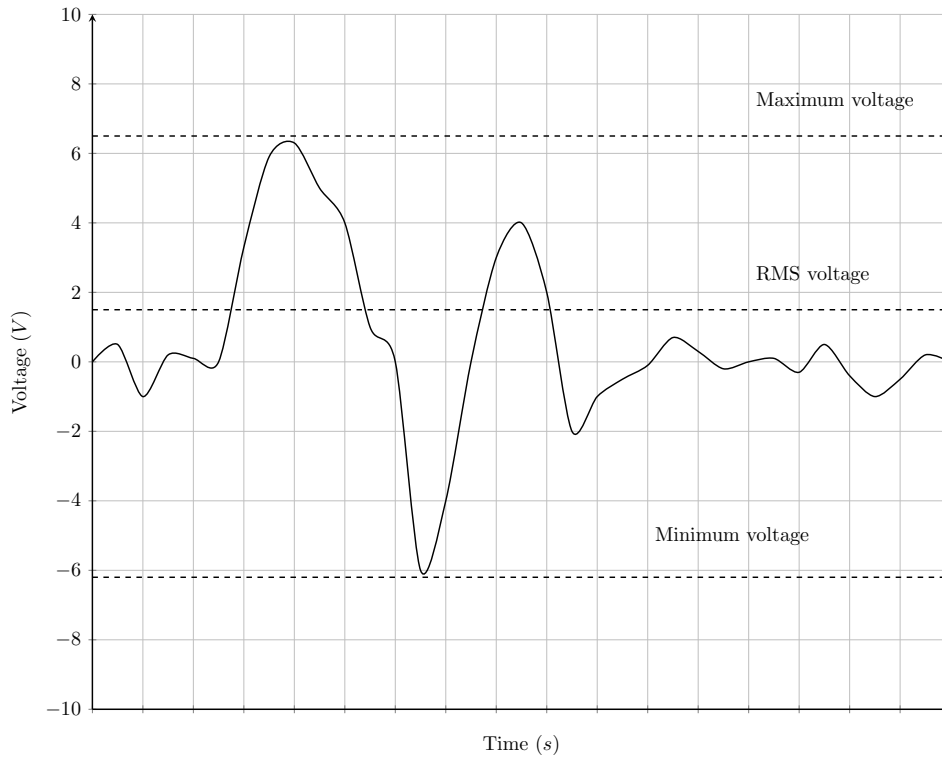


Figure 4.28: Examples of a number of time series features on an event waveform (exhaustive feature list can be found in [122])

The tool `tsfresh`, created for similar applications, provides the ability to feed the system raw time series data, it then runs through the different features that the system is set to extract, then it returns a vector with all of the different features from that data. A similar tool, `HCTSA`, built for `matlab` provides a similar functionality, this tool has been used in the NILM field as an automatic feature extractor [120].

The features made available with the use of this tool: min, max, skewness, peaks, RMS, etc. include features that are commonly used from a range of different domains which include audio analysis, stock data, statistical analysis, and more. There are various fields that make use of time series data, each using their domain specific tools and features which enable analysis for those purposes. The purpose of using such a tool is that at this point, with minimal domain knowledge, it is not simple to understand which features may perform well, and which will not. This tool, extracts all of these features, making it likely to find features that are highly correlated with the underlying dataset, as well as features that pose little usability. An example of a feature that may be useful are the minimum and maximum of each dataset, as it

is expected that different devices will create differing magnitude responses on the network, where the magnitude of these responses is likely to coincide with the power that the device requires. On the other hand, a metric such as the length of the time series is not expected to provide any useful information in this case, due to the fact that all events have the same number of samples (see section 4.6).

This is an important consideration to understand and design into the system as there will be features that represent the data well, and those that do not, all with varying *strength*. This is where the feature filtering step accounts for the different performance of different features and is discussed in detail in section 4.8.3 as it is capable of ordering the features in importance, and finding features that are redundant.

4.8.2 Feature Normalization

The feature extraction process that creates the event feature vectors in section 4.8.1 creates features that are highly variable and diverse. This means that some features vary by small amounts when comparing different events and some features have large differences.

This means that some of the feature dimensions will have a small range, whereas other feature dimensions will observe points over large ranges. These variations in different features are also affected by outliers, events which are a little different from most others can often affect systems further down the pipeline. Data normalization, a pre-processing step that assists in reducing the effects of amplitude and offset variations in the data, makes sure that these small effects do not heavily distort and reduce the performance of the system. The aim of these pre-processing procedures is to reduce the impact that some features will have on the system, particularly in cases where the values from the different points vary over large values.

Clustering algorithms often benefit from the use of normalization and/or standardization of the feature set before the clustering step [123].

The normalization and/or standardization is *often* seen to improve the results of the clustering process, in addition to this, different normalizers work well with some clustering techniques and not with others. This implies that it is worth implementing a range of different normalization types, and applying them each to the different types of clustering techniques. This will assist in determining the most performant technique (combination of pre-processing and clustering strategies) that work well for this research. The reasoning and implementation for this testing strategy is discussed

in section 4.10.

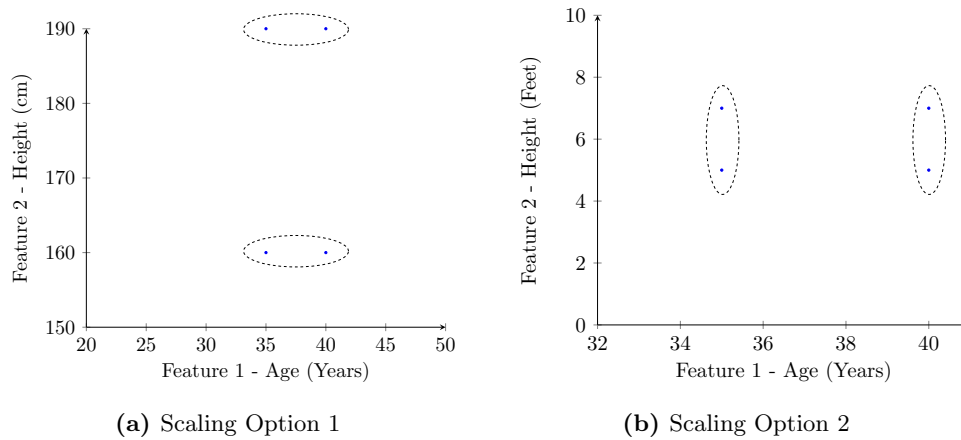


Figure 4.29: Different types of feature scaling

The points illustrated in fig. 4.29 highlight how scaling of features are important. The example here, commonly used to show how normalization is used in the data pipeline, is important when clustering different points in a feature space. The normalization, or type of feature observed, can change how the clustering is run and the ultimate clustering results.

Some of the clustering techniques utilized in this work make use of distance-based measures for creating clusters and determining how clusters are assigned. Many make use of Euclidean distance measures, this distance measure is particularly sensitive to these features that have different magnitudes; in the absence of feature normalization, these differences can skew and distort the functioning of the clustering algorithms.

There are a number of different normalization techniques that are applied in different fields for different end uses, taking heed of the discussion in section 4.10, it is simple to implement each of the commonly used normalization techniques on the data, such that they can each be compared to find which method performs best for this methodology implementation.

The platform, making use of the Python ecosystem takes advantage of the scikit-learn toolset (a large collection of data analysis and machine learning tools) to implement a range of the available normalization methods [124]. This optimization strategy is particularly useful here as it is simple to implement and test each of these normalization techniques and observe their performance and applicability to this methodology. Particular attention is paid to the case where no normalization is used against where simple normalization is used to see if it is worthwhile implementing

normalization techniques and if it increases the performance of the clustering stages.

- Standard Scaler [125],
- Min Max Scaler [126],
- Max Abs Scaler [127],
- Robust Scaler [128], and
- Quantile Transformer [129].

Each of these normalizers/transformers are built with different datasets in mind, some are robust to outliers, some not. These normalizers are run through the processing pipeline to generate normalized features.

Based on early testing, it is clear that simply implementing the standard scaler increases the performance of the clustering stages massively, this suggests that normalization is very helpful and that there are clearly features that require normalization. Consequently, for the majority of tests, the standard scaler is capable of achieving sufficient disaggregation ability, much greater than a situation where no normalization is applied.

4.8.3 Feature Filtering

The feature extraction used throughout this work makes use of a tool that enables for the extraction of many features, the total number of features that are created by this process exceeds ~ 700 . The goal of that step is to extract as many features as possible from the original data to obtain the maximal amount of useful, lower dimensional data, from the raw data; some of these features are known to provide useful information based on prior, domain knowledge, for example the RMS voltage is often useful, in addition to values such as the minimum and maximum values. However, due to the generic nature of the tool used, based on the fact that the different feature extractors are often used for widely varying end use cases, some of these feature extractors are likely to be poor predictors of what the underlying data represents. Consequently, a system is implemented in this feature engineering pipeline to enable the ability to determine which of the many features are useful and good predictors of the data. The goal of this step is to determine which of these features, in order of performance, are the most highly correlated to the switching of devices within the system.

Other research in this field makes use of *domain knowledge* or due to a lack of many features to choose from, commonly make use of simple features such as the Fourier transform and the harmonics in the Fourier transform [86]. The fact that this implementation observes a DC voltage (and the transient on top of that), it is unlikely that harmonics (well-suited to AC situations) will be readily applied to this methodology. Previous methods in this field also have limited feature sets upon which to test, making it relatively simple to test the different features [130].

This necessitates the design of this feature filtering system. This step in the process is where the ground truth becomes particularly valuable (see section 4.6.1) as it enables the ability to compare each event with its *expected outcome* and to infer the applicability of each of the features extracted.

The system used here is run through set protocols which are created to provide a generalized representation of signatures of the system. This means that, for the feature filtering, a limited number of protocols are used in order to obtain a realistic idea of how the features are related to the ground truth data. This step is a tuning step which is required to find the features that work well in representing this type of data; this step is also introduced to observe the presence of redundant features (see section 4.8.3.2).

The major problem faced by this component is that it is required to find a subset of the total feature set, such that when this subset of features is given to the clustering stages, the clustering is able to perform with the highest possible accuracy [131, p. 275].

4.8.3.1 Feature Filtering Analysis

This functions such that each events' feature vector (for selected protocols) is taken against its corresponding ground truth label, this is formed into a matrix of the protocols selected. This matrix is illustrated in table 4.7 where this process is run for a single protocol which contains a single device, this is a similar protocol to that defined in table 4.4. The system takes the ground truth information, it converts this information into categorical data, this is based on the fact that events 1, 3, 5, ... and 2, 4, 6, ... form two distinct groups into which each of the events fall into.

Statistical analysis is then performed on this matrix such that each feature is compared against the labelled category (ground truth) such that a performance score can be generated. This performance score details how correlated each feature is to

Table 4.7: Example protocol with features and ground truth labels

Event Number	Feature Number				Ground Truth	Category
	1	2	...	n		
1	-5.4	-5.8	...	3.2	101	A
2	4.8	4.3	...	-2.1	110	B
3	5.5	3.8	...	3.1	101	A
⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	4.8	4.3	...	-2.0	110	B

the categorical output labels that are given to the system from the ground truth information.

This filtering step enables the ability to choose the first n best features when performing the clustering, thus ensuring that the highest correlated features are used in descending order. These issues of dealing with high dimensional data are discussed in detail in section 4.9 where a number of techniques are introduced to deal with this high-dimensional data that is inherent to this research (based on the fact that it is possible to cluster up to the total number of features ~ 700).

The different normalization methods used in section 4.8.2 are likely to transform the data in ways such that the performance of features will be different across different normalization methods. Consequently, this process generates distinct performance metrics for each of the different features in the different normalizers.

There are a number of commonly utilized feature filtering methods used in machine learning: filter, wrapper, and embedded methods. These three techniques differ based on the requirements set out by the system.

Filter method: This method creates performance metrics for each feature based on the correlation coefficients of the features in the dataset [130, p. 1158].

Wrapper method: This makes use of a feedback mechanism such that multiple feature subsets can be selected; these are then tested on the final predictor and the performance is evaluated at that step [130, p. 1158].

Embedded method: This method is similar to the wrapper method in that it makes use of a feedback mechanism to analyse how the selected features perform with the predictor, however, in addition it uses the recursive nature to further enhance the feature subset selection procedure [132, p. 246].

The last two methods are particularly suited to supervised machine learning systems; however, they also require a more complex setup to function well. The first method works sufficiently in this research as it enables simple analysis for performance of each feature. Conveniently, the filtering method does not interfere or depend on the choice of the end predictor (clustering technique/s) as it is simply a mechanism for ranking features [130, p. 1160]. This method is also the most efficient in terms of implementation and run time cost as it assigns a score for each feature, then proceeds to order the features by these scores²⁷.

The method that is utilized for the feature selection in this system makes use of the analysis of variance (ANOVA) test statistic. This is a measure of the degree of variance that a variable/feature varies in a range, i.e. a feature with a low variance means that all samples vary over a small range. The higher the ANOVA test statistic, the lower the chance that the variable represents noise and that it is a useful predictor of the output. This means that the features that test high in this analysis are likely to represent features that correlate with the expected output.

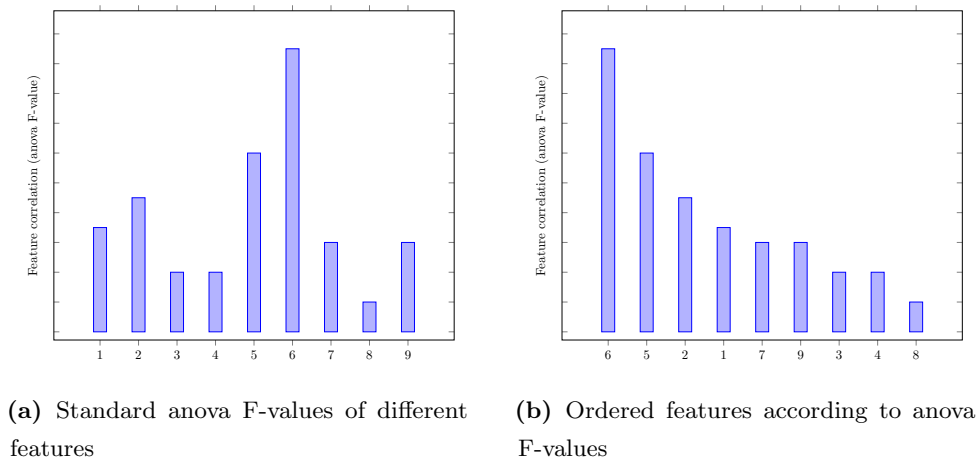
This ANOVA F-test statistic is particularly well-suited to data sets that contain numerical inputs (the feature vectors) and categorical outputs (the ground truth clusters). Python scikit-learn provides a method in which to obtain the performance of the features when coupled with the ground truth data [133, 134]. This test generates a value which describes the correlation between an input variable and the output label. This is illustrated by fig. 4.30a.

The best-case scenario for utilizing the ANOVA test statistic is when each of the groups contain the same number of samples, or observations, this makes it a balanced design which functions well with the one-way anova test [135].

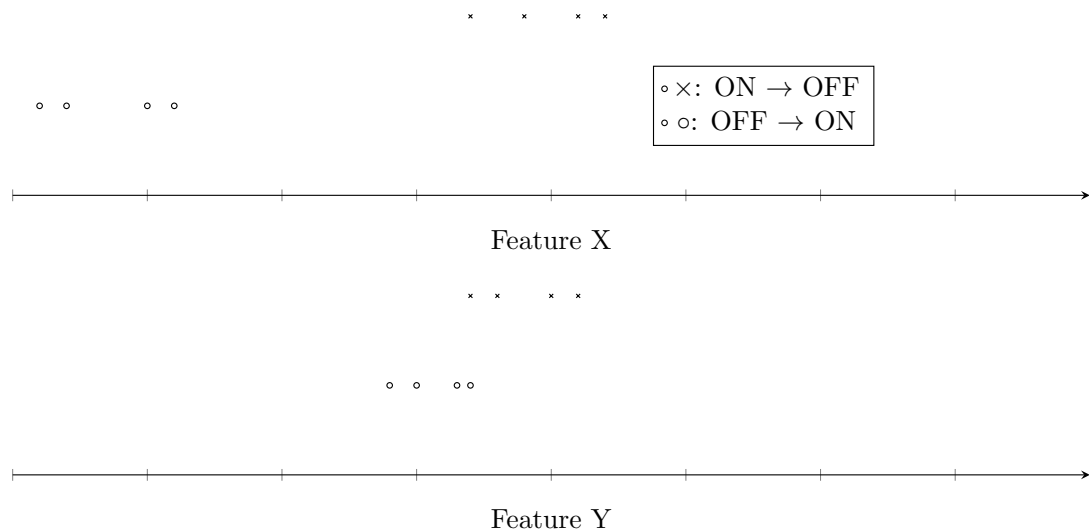
This then allows for the features to be ordered by their correlation to the expected output as illustrated in fig. 4.30b.

To conceptually understand how this algorithm functions, an example using feature X and Y illustrate how the features are scored. These two features and their significance are provided with the simplistic protocol 1 example which contains a single type I device, resulting in two ground truth labels for the two transitions: $ON \rightarrow OFF$, and $OFF \rightarrow ON$. The differences between the two features are illustrated in fig. 4.31. The distribution of samples in this example are not indicative of features in the research, this example makes use of normally distributed features,

²⁷The other methods have much higher computation requirements, increasing with each extra feature used.

**Figure 4.30:** Feature analysis correlation

however, this is not always the case.

**Figure 4.31:** Distribution of two features and their ground truth

Feature X, in this case has a large F-value from the ANOVA test, meaning that it has a large dispersion/separation between the two groups mean values; the feature Y, on the other hand, has a small F-value from the ANOVA test. This is illustrated by figs. 4.32 and 4.33 respectively.

Additionally, each of these features have corresponding p-values from the ANOVA test, these values represent the area under the groups curve past the two standard deviations. The example here illustrated by figs. 4.34 and 4.35 represent the features X and Y, where feature X has a low p-value, meaning that the two distributions are less likely to overlap (without considering their F-values) and a high p-value

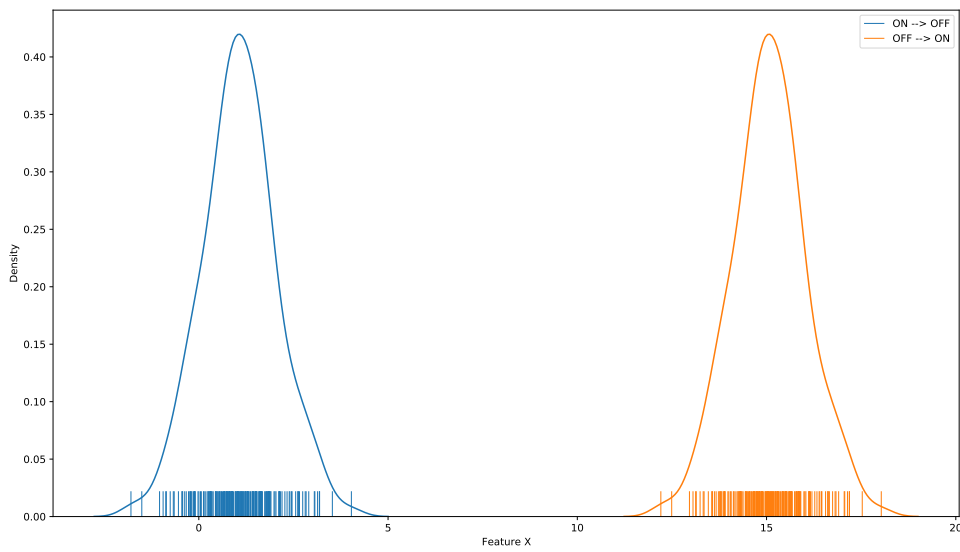


Figure 4.32: ANOVA Statistic - high F-value

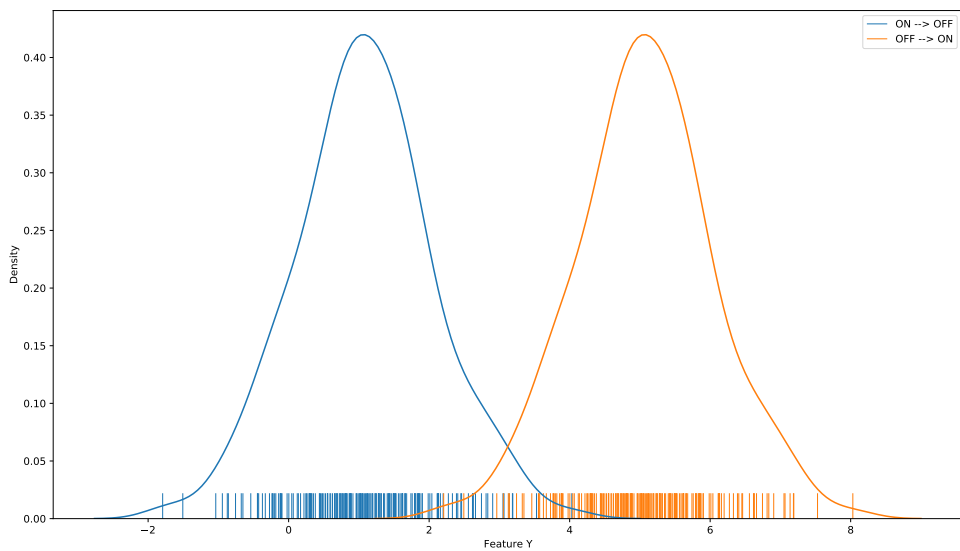


Figure 4.33: ANOVA Statistic - low F-value

will result in a higher chance of the distributions overlapping. Situations where the distributions overlap is unwanted as this makes it increasingly difficult for predictors further down the line from establishing which event is assigned to which group. The ideal scenario is illustrated in fig. 4.35 where there is no overlap of the two distributions, this will assure that any separation of the two data sets will have a much higher accuracy.

Consequently, the best-case scenario for a feature is to have a large F-value and a low p-value.

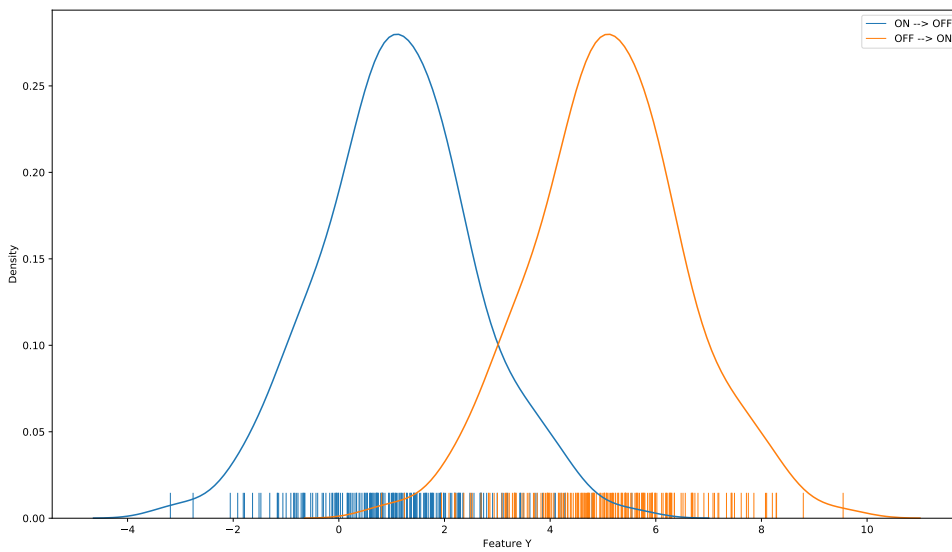


Figure 4.34: ANOVA Statistic - high p-value

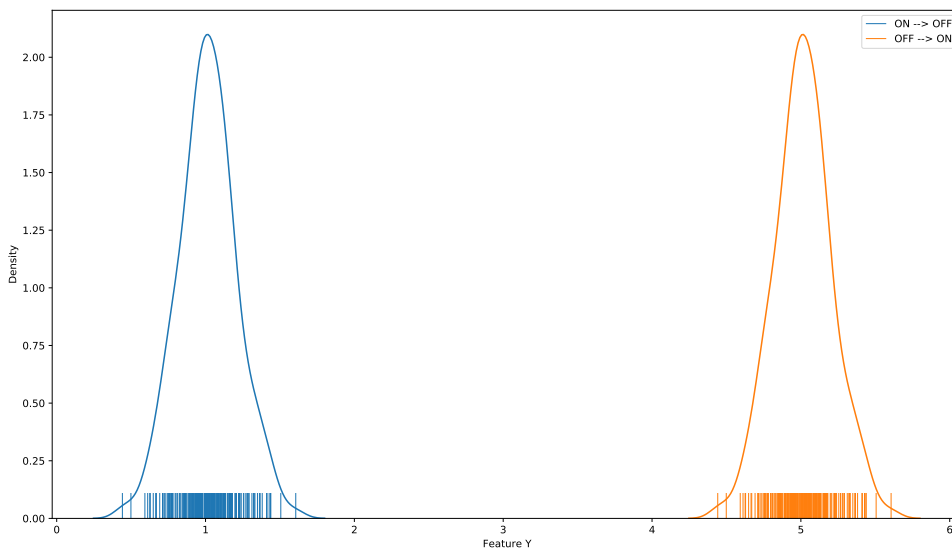


Figure 4.35: ANOVA Statistic - low p-value

4.8.3.2 Redundant Features

The process that creates feature correlation scores does not consider how different features are related with each other. Consequently, there will be features which are correlated with each other (they then will have similar scores from the previous test). These features are considered redundant, and in the case that they are used together, they are unlikely to assist with adding information and in some cases, they may reduce the performance of the clustering techniques as they add a further dimension

to the data²⁸.

A simple method to analyse the correlation between features is to make use of the Pearson correlation method. This takes each of the feature vectors and tests the level of correlation between a feature and all other features. This is illustrated by fig. 4.36 which indicates the level to which each of the features are related, and therefore redundant.

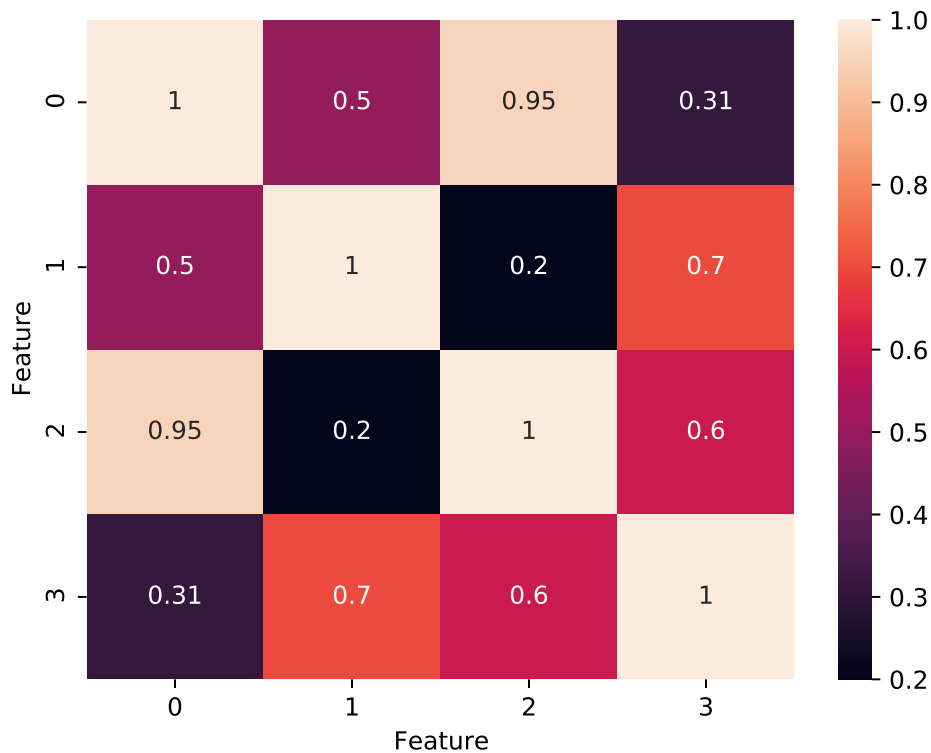


Figure 4.36: Pearson correlation heatmap for different features

The different features illustrated in fig. 4.36 indicate that each figure is perfectly correlated with itself (this is always the case) as the main diagonal shows. When comparing, for example, feature 1 and feature 3, their coefficients (0.95) are close to 1.0, indicating that they are highly correlated, and consequently it will be beneficial to remove one of those features from the feature set in order to remove this redundancy. The features 3 and 2, have a low coefficient (0.2) implying that they are not highly correlated and they are not redundant.

This further analysis should be considered in tandem with the ANOVA statistic above in that features should have a high statistical variance and correlation to the

²⁸Increasing the dimensionality of the feature space is often unwanted if it does not add sufficient information to the overall system.

expected output, additionally there should be a limitation on the introduction of redundant variables given to the predictors (clustering techniques).

4.8.4 The Utility of Feature Extraction

Ignoring the fact that more features often leads to higher and better performing predictors, this does not consider the fact that each additional feature extractor adds computational complexity and computational cost to the entire pipeline [131, p. 276]. This is in fact, one of the many trade-offs that are made when this feature filtering process is explored.

This research, due to the fact that it is currently not known which features are more applicable to this work than others, spends a non-insignificant amount of compute time extracting all of these features. Final iterations of any system that would realistically do this work in the real world would perform all of this beforehand, to get an idea on how to make the feature extraction more efficient. This means finding the best features, and only running feature extraction for that subset of features, thus reducing the complexity of such a system.

4.8.4.1 Feature Performance Roloff

Section 4.9.4 discusses the fact that many features that are extracted, are likely to be of little utility (low ANOVA scores), additionally many features are likely to be redundant. There are further concerns with using increasingly more features when clustering: the fact that the curse of dimensionality comes into play (see section 4.9.4).

This feature analysis lines up with the results of the ANOVA analysis observed in section 5.4.2 (see fig. 5.22) where the first few features contribute a large amount and have a high amount of information gain followed by a steep drop-off in the scores that each feature contributes.

To combat this, a system is created, which includes a number of singular device protocols, where an increasing number of features is selected, and for each of these selections, clustering is run, and the performance is measured. This enables the ability to determine the optimum number of features that can generate the best performance. Figure 4.37 illustrates the performance of clustering nine devices by utilizing the adjusted rand index to determine the best score. This illustrates that

starting from zero features, the performance quickly rises, then there is a drop (see section 5.4.2) until around 150 features, where the performance increases up to an ARI of approximately 0.9 which is sufficient in NILM and load disaggregation requirements (the ARI value here is discussed in greater detail in section 4.9.2). After that point, there is a steep drop-off where increasing features do not contribute to assist in the disaggregation process.

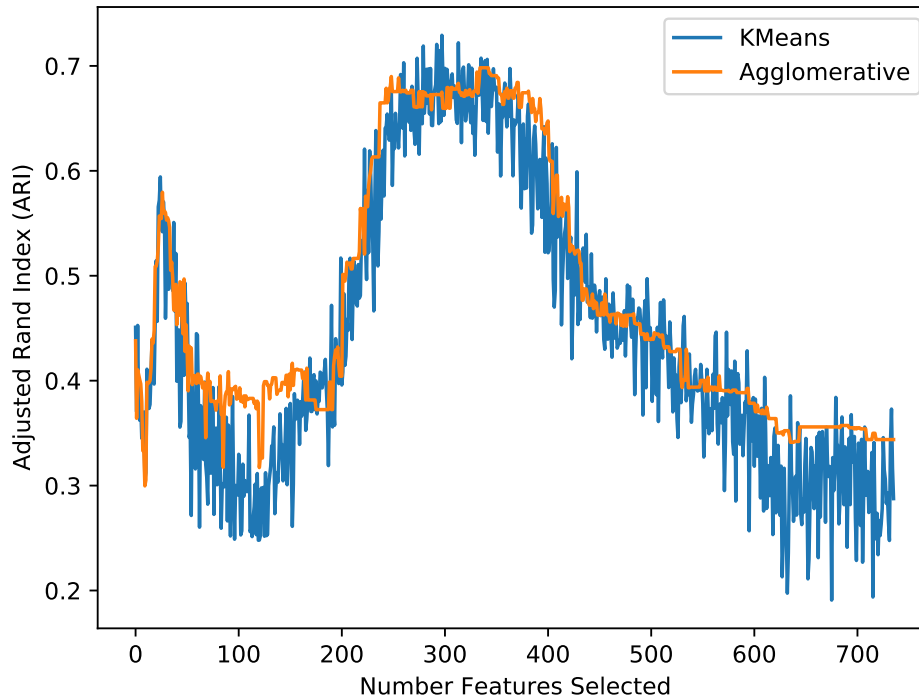


Figure 4.37: Feature performance rolloff with K-Means and Agglomerative clustering

This method of observing the clustering performance with the selection of more features illustrated in fig. 4.37 has been observed in this field in previous work; the similar time series feature analysis in [120, 136] have similar results where increasing the number of features results in minimal benefit.

The two algorithms reach their peak performance at slightly different feature counts: 300 for K-Means and 350 for Agglomerative clustering.

To maintain consistency, when analysing protocols, both clustering algorithms utilize the 300; this number is also near the apogee of the performance for agglomerative clustering. This fact is utilized in the chapter 5 when observing different tests as this is useful to find the best system to measure performance across different tests. Further than this, when analysing the performance of features, it is evident that the Fourier coefficients create a reduction in the performance of the clustering, this is

discussed in section 5.4.2.

4.9 Clustering Techniques

This section outlines the design of, and systems used, which seek to obtain a measure of the ability of different clustering techniques in differentiating the operation of devices of the different protocols.

Similar to the design of the previous systems, there are a number of different clustering techniques that perform well depending on the structure of the underlying data. Thus, in line with the principles outlined in section 4.10 this system seeks to utilize a number of clustering techniques commonly used in this research field.

Clustering techniques are utilized in many varying fields for a range of end-use cases; the structure of this methodology, is particularly designed to perform clustering on time series data [117, p. 19].

The aim of these clustering techniques and how it relates to this research is discussed in section 4.9.1, the different methods of cluster evaluation can be found in section 4.9.2 and the different specific clustering tools utilized in section 4.9.3. The addition of Principal Component Analysis (PCA) is utilized within this methodology, its use and benefits are discussed in section 4.9.4. Importantly, the role of cluster validation types and how they are utilized are discussed in section 4.9.2.

4.9.1 The Aim of Clustering

The utility of clustering tools, within the context of this research, is to provide the ability to uncover the underlying structure of the data in each of the different protocols. The tools seek to uncover the pattern that each device and its operation exhibits within the system by enabling the ability to observe and categorize the different *events* into separate clusters [137, p. 264]. Intuitively, points in a 2D feature space (i.e. each point has a feature vector of two features), where there are two known clusters of the data, will have patterns such that points in one cluster will be placed more similarity to points within that same cluster as to points within a different cluster [137, p. 265-266]. This can be illustrated via fig. 4.38 where the two clusters, each containing 5 points, contain two distinct groupings of the points. This simplistic example illustrates that the points in the same group (cluster) in fig. 4.38,

are provided with the same labelling from the clustering process where the labels are shown in fig. 4.39; the two groups contain distinct labelling from one another as the clustering identifies two separate clusters. Again, note that the shape of the clusters in this example are used to illustrate the idea, however, in practice cluster shapes vary greatly²⁹.

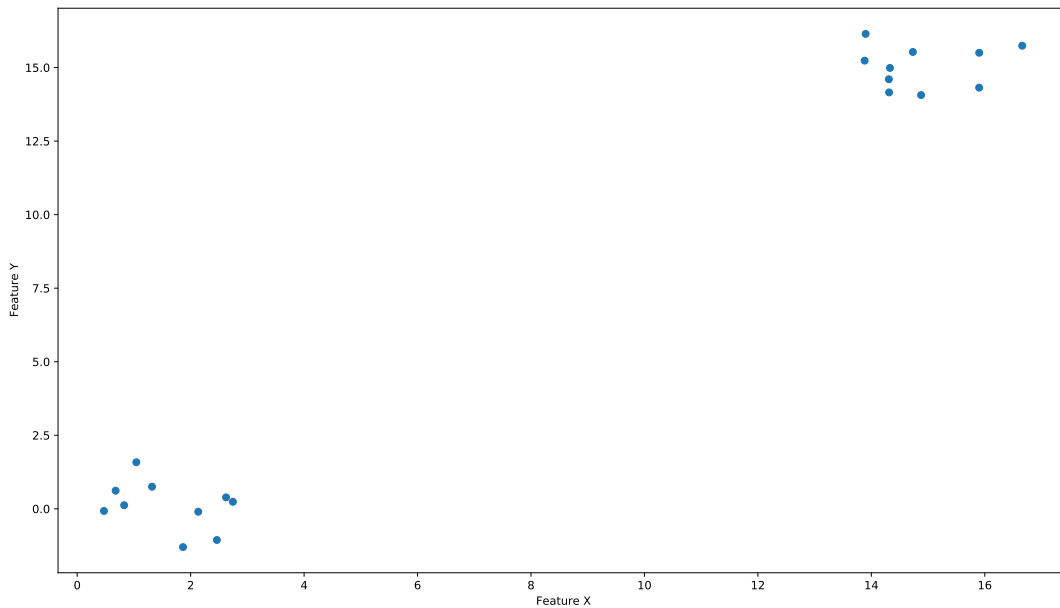


Figure 4.38: Clustering data without labels

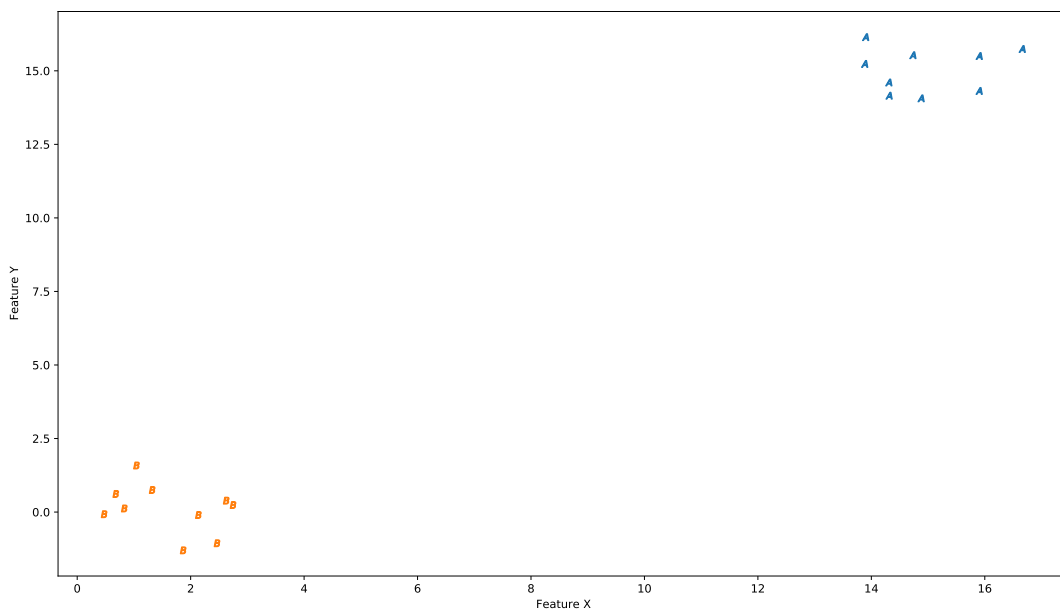


Figure 4.39: Clustering data with labels applied from output of clustering algorithm.

²⁹This is where different clustering algorithms may perform better than others as different clustering algorithms perform better with different cluster shapes.

These clusters, created by the different tools, aim to segment the events into separate, but similar *groups*. These groups, those discussed in section 4.6.1, represent the patterns exhibited by the system.

The clustering techniques take each protocol (in the case of this research), and the feature vectors of the events within that protocol, operate on this information and map these into high-dimensional feature spaces. The clustering tools then attempt—in a number of ways, depending on the clustering type—to group the different events based on their feature vectors inside this high-dimensional feature space.

Clustering, the task at hand, can be solved in many ways, these different techniques/tools—known as clustering algorithms—are used to define the clusters [137, 138]. It is the definition of what a cluster is that separates the different algorithms (see section 4.9.3).

Generally, it is understood that events that are within the same group (cluster) have more similarity (in their feature vectors) than events from other groups (clusters) [117, p. 16]. The most important characteristic of any clustering technique is that it is capable of *objectively* sorting data into groups that are similar to one another [117, p. 16]. This means that if there is an underlying structure to the data, then these tools will uncover those patterns.

This is particularly important (in this methodology) as it is then used in combination with the ground truth data (*a-priori* information) in order to compare the results from the clustering tools to the real-world ground truth that is recorded. This creates the ability to generate performance metrics for the different clustering techniques against a set standard.

A number of validation techniques, which apply to numerous clustering techniques are applicable to this methodology (see section 4.9.2), where it is apparent that the presence of the ground truth enables the use of external validation techniques to be used [108].

4.9.2 Clustering Evaluation Techniques

Cluster validation analysis, the final step in the process, is used to evaluate the performance of the clustering output with the use of a number of selected metrics. There are three common methods of cluster analysis[137, p. 267-268]:

- External: the use of a-priori metadata/information about the labels of the

points are compared to the labels the clustering assigned.

- Internal: this method attempts to determine if the structure of the clustering is appropriate to the underlying data.
- Relative (less common): this compares two clustering structures and attempts to use a relative measure to provide metrics.

The second option implies that ground truth data is not present and results in a case that the performance is evaluated by a number of heuristic measures of the final clustering. The first option is particularly well-suited to this research and methodology in that the ground truth data is available and readily utilized to observe the performance of the clustering process against what is expected.

The presence of ground truth data in real-world situations will not be available, which is why clustering techniques are referred to as unsupervised processes. The ability to use ground truth data here is relevant as it enables the ability to prove (or disprove) the fact that the transients do in fact have underlying structure which can be utilized to perform NILM on such systems.

The commonly utilized metrics for defining NILM and load disaggregation performance are discussed in section 2.8.3.1. This research, noting the fact that an event detector is not utilized, means that the performance metrics need to be adjusted to capture the performance that this system is enabling/observing. This work looks at directly assessing the feasibility of whether devices are distinguishable from their signatures, consequently this creates the benefit that a generic set of performance metrics can be utilized which can reliably provide these metrics for in-depth analysis. This is functionally based on the understanding that, with the ground truth data available, the evaluation of the algorithm can be undertaken with the use of external validation metrics [108]. This means that the ground truth labels (of each event) are directly compared to the labels that the clustering algorithm assigns each event measurement to. This is performed on a protocol-by-protocol basis in order to establish a performance value for the different tests. The metric utilized here, which provides a measure of the performance of different clustering algorithms, is the Adjusted Rand Index (ARI).

The ARI works such that it creates a measure of the similarity of the original ground truth labels to the labels that the clustering assigns. This performance measure is related to the accuracy; however, it provides a greater applicability as discussed here. The Rand Index (RI), which is what the ARI is based upon, is related to the work

discussed in section 2.8.3.1, and can be defined by eq. (4.9), in essence, this is a measure of the percentage of correctly identified cluster labels when compared to the ground truth.

$$RI = \frac{TP + TN}{TP + FP + FN + TN} \quad (4.9)$$

The ARI, with a score of 1.0 means that the ground truth exactly matches up with the labels the clustering assigned and therefore, a score of 0.0 (or a negative score, which means that the labelling's are independent) results in a case where none of the ground truth labels match up with the clustering labels. Table 4.8 illustrates the standard protocol with a single device (type I) that switches on and off; this illustrates a case where the clustering algorithm correctly predicts all of the events observed. Table 4.9 then illustrates a case with the same protocol, however in this circumstance the clustering algorithm incorrectly predicts the labels which then results in a reduced ARI measure, meaning that the clustering algorithm failed to identify the two sets of events correctly³⁰.

Other commonly utilized performance measures utilized in the research include F-measure, accuracy, precision, and recall [52]. These measures, however, are utilized with load disaggregation estimators and classifiers. These measures consequently do not translate well to this methodology as they are more suited to predictors which are those tools utilized in the supervised machine learning tools that enable the use of entire NILM algorithm pipelines to be evaluated. The ARI utilized in this research is particularly suited for use here as it is an efficient and accurate marker/measure for the separation and similarity of clustering's of data [80, p. 104].

This method of evaluating the performance of clustering algorithms is particularly suited in this work, partly because it is a reliable measure of the ability to find structure in the data: it is also highly capable of being used across different types of clustering algorithms making it a useful metric for finding clustering algorithms that are good at separating out the clusters. This means that the ARI can provide a score that can be used with K-Means as well as Agglomerative clustering, and the two clustering algorithms can be compared against one another for specific protocol tests [139, 140].

³⁰This second table (see Table 4.9) illustrates how, when there is a small dataset to work with, even a single incorrect event prediction from the clustering will result in a small ARI score.

Table 4.8: Clustering Evaluation with ARI of 1.0

EventID	Device Transition	Ground Truth Label	Clustering Label
1	OFF → ON	101	1
2	ON → OFF	110	0
3	OFF → ON	101	1
4	ON → OFF	110	0
5	OFF → ON	101	1
6	ON → OFF	110	0

Correct Clustering Assignment

Incorrect Clustering Assignment

Table 4.9: Clustering Evaluation with ARI of 0.324

EventID	Device Transition	Ground Truth Label	Clustering Label
1	OFF → ON	101	1
2	ON → OFF	110	1
3	OFF → ON	101	1
4	ON → OFF	110	0
5	OFF → ON	101	1
6	ON → OFF	110	0

Correct Clustering Assignment

Incorrect Clustering Assignment

4.9.3 Clustering Algorithms

The two clustering algorithms utilized in this work—based on previous work in this field which uses these algorithms—are the K-Means, and hierarchical (agglomerative) clustering algorithms [44, 108]. Generally, these two algorithms are divisive and agglomerative methods respectively, these are differentiated in how they create clusters. The exact implementation and specifics of these algorithms are not covered in this work as it is out of the scope.

4.9.3.1 K-Means Clustering

The K-Means clustering algorithm, a partitioning algorithm, takes in an expected cluster count where it then attempts to create clustering's of the data provided until

all points are assigned to the different clusters. The data points in the feature space are clustered, i.e. they belong to, the nearest cluster centroid.

The K , in the K-Means describes the number of clusters that are defined before the clustering process takes place. This cluster count, as discussed in section 4.6.1 is related to the ground truth labelling's for the different protocols. The ground truth labels, specifically the number of *unique* ground truth labels represent the number of clusters that the algorithm receives.

4.9.3.2 Agglomerative Hierarchical Clustering

Agglomerative hierarchical clustering, contrary to K-Means, makes use of agglomerative clustering: this technique starts out where each event is assigned to its own cluster, the algorithm then successively combines events into larger and larger groupings/clusters, until all events are merged into one single cluster. The algorithm, however, keeps track of the groupings of the clustering's in each step, thus allowing the structure of the underlying data to be observed. This is highly convenient as this algorithm can be used to uncover the structure of the data even if the ground truth data is not present.

A common technique to visualize this process is the dendrogram, this diagram illustrates how the algorithm combines the different points at each of the different steps in the process. For example, at the bottom level of the dendrogram, all of the individual events are their own clusters. The algorithm then takes each of these clusters, and in an increasing *ring radius*³¹ around each of these clusters, it will combine with the nearest clusters to show that these two clusters are now combined. At this point, the two legs will join at that *radius* to form a single cluster. This process continues until all clusters have been combined into a single cluster, thus forming a binary tree. The dendrogram illustrated with this binary tree, where the leaves of the diagram (at the bottom) are the events that, in the first step, are their own clusters, will combine into groupings which will highlight the underlying structure that the data contains.

This dendrogram, at the different vertical levels can be used to analyse how and where the clusters are differentiated from one another, thus allowing the ability to compare the ground truth labelling's (the number of clustering's) against the

³¹This term *ring radius* is used to provide a mental visual representation of a situation with only two features (x , and y) when in fact this feature space will be a high dimensional space not easily visualised.

dendrogram output. Figure 4.40 illustrates the single device protocol with a number of events (the events are observed on the bottom of the dendrogram) which start off in their own clusters. Starting from the bottom, moving upwards, the algorithm combines the events into larger and larger clusters, the length of the legs of each of these consecutive clustering's represent the level of dissimilarity that is apparent between the clusters as they merge. This highlights the labelling in the figure where B shows how events that are similar (ON or OFF clustering's) will be grouped together relatively easily, and the two areas labelled A highlights the much larger dissimilarity that is observed between the last merging of clustering into a single cluster. This shows that the best clustering assignment that the algorithm observes is that the area labelled A shows how the two clusters (and the events inside the clusters) are best assigned. This ability that the dendrogram affords, the ability to provide insight into how well the ground truth actually represents the functioning of the system, is discussed in section 5.4.3 and particularly in section 5.1.1.1.

4.9.4 The Utility of PCA

The algorithms utilized in this work (see sections 4.9.3.1 and 4.9.3.2) are both used to perform analysis on the protocols and to evaluate the research questions. However, the methods in which these tools have been utilized, the fact that the feature vectors that are available to the clustering algorithms are large (exceeding 700 features) creates two sets of problems: the curse of high dimensionality and visualization issues.

The methodology that is utilized in this work is to make use of creating feature vectors that start from using a single feature, and successively using more and more features to cluster the data. This is done in an effort to determine how many features are required to sufficiently characterize the operation of the system. This means that it is possible to draw conclusions on how well the clustering algorithms perform with more features, this can be illustrated by fig. 4.41 where the overall clustering performance increases (or decreases) when more and more features are used to assist in providing more data.

The high dimensionality problem comes into play when larger and larger features are fed into the clustering algorithms. Some clustering algorithms have reduced performance when the dimension of the data points gets sufficiently large, thus a tool can be used to alleviate this issue: Principal Component Analysis (PCA) [141].

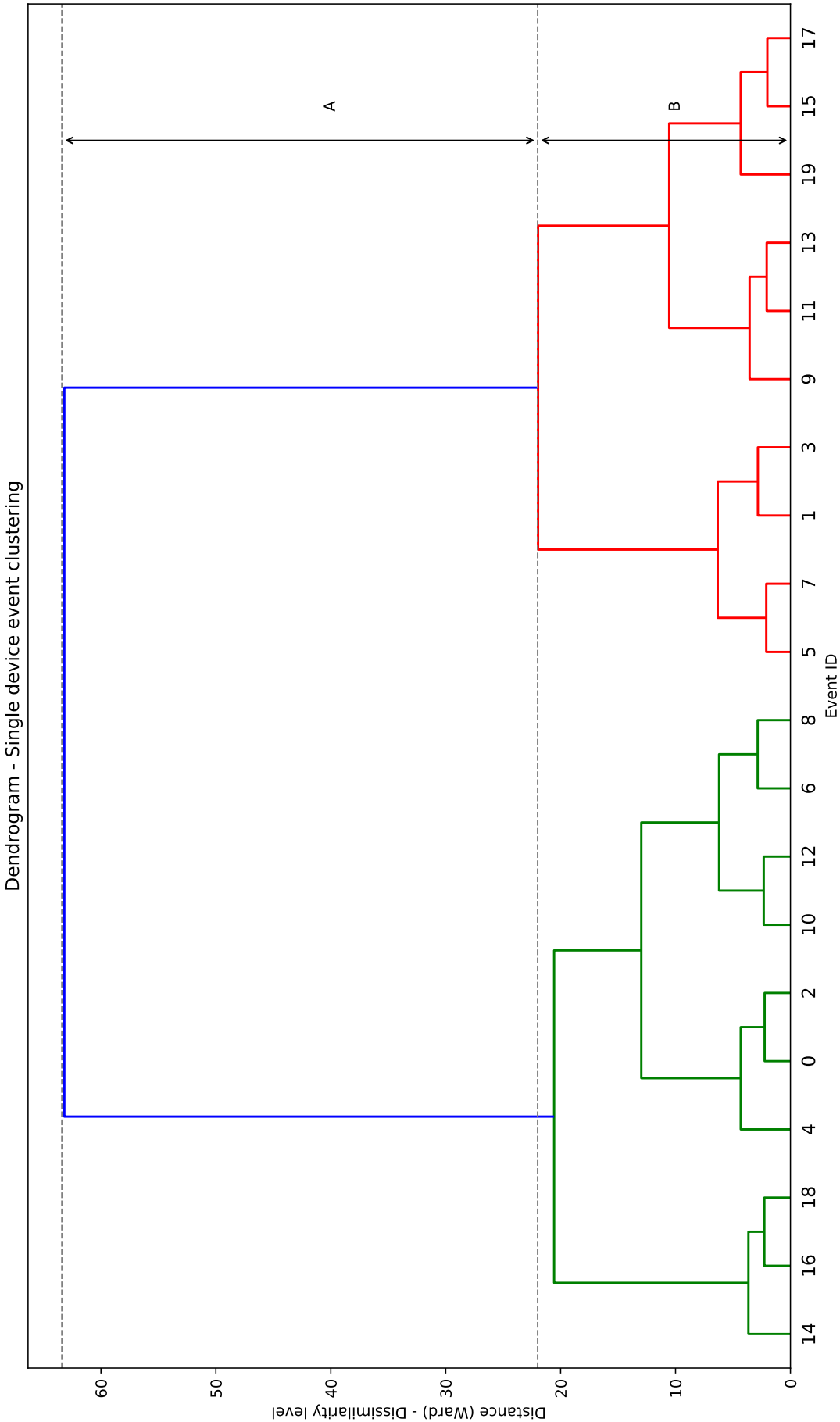


Figure 4.40: Example single device dendrogram

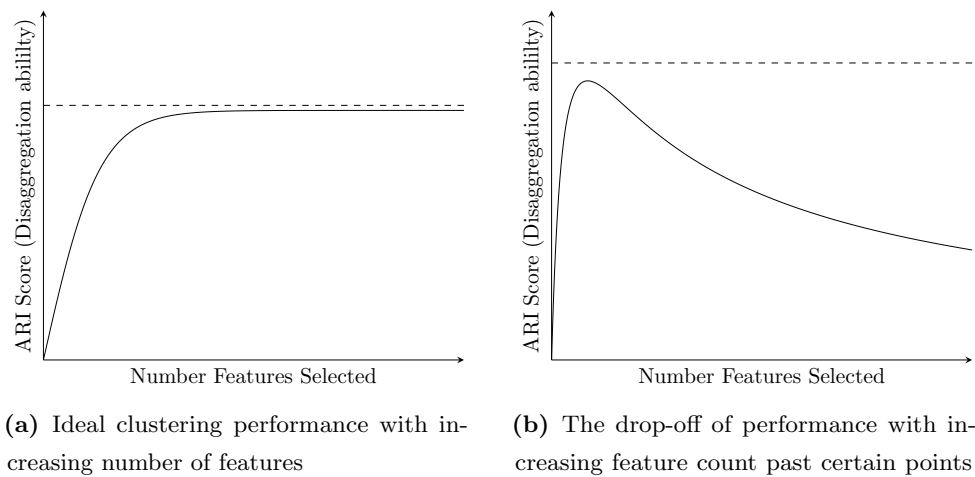


Figure 4.41: The comparison between the ideal case of increasing performance with increasing features compared to reality

The second issue is to do with the visualization of the information, these high dimensional feature spaces cannot be observed with the use of simple plotting tools, the PCA tool can assist with this as well³².

PCA functions such that it takes a dataset with two or more features (for the high dimensional cases) and transforms the original data into a new representation with a selected number of components. The data is normalized (see section 4.8.2) before being fed into the PCA algorithm where each point—in the original, high dimensional feature space—is transformed or projected onto a lower dimensional space that is selected depending on what is required.

What this means is that for the clustering algorithm case, high dimensional feature vectors can be transformed into a select number of components that still represent the underlying structure of the data.

To illustrate how one of these processes would work, the original features are converted from two, down to a single principal component; fig. 4.42 shows this process.

This implementation, the cases where the system creates component counts of two or three, can be utilized to provide a visual representation of the placement and characteristics of the clusters. The original information is now condensed into a format that lends itself to visualize in two and three dimensions³³. Figure 4.43 illustrate

³²Visualization tools are not essential, the ARI validation tool provides sufficient analysis; however, it is sometimes useful to observe how some points are clustered visually which can sometimes assist in determining how the clustering is interpreting the clusters.

³³A single dimension can also be used for visualization, although this is uncommon

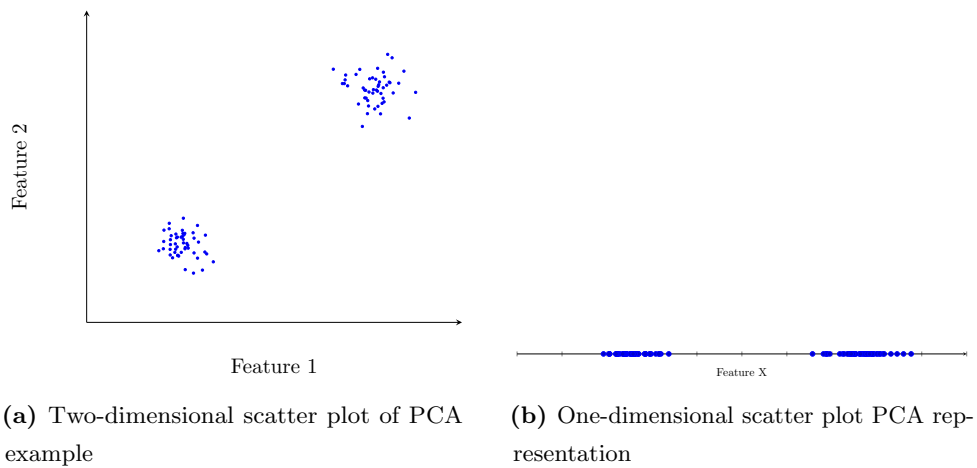


Figure 4.42: Example PCA visualization application with two- and three-dimensional scatter plots with K-Means clustering

the example protocol with PCA creating two cases of two- and three-dimensional component representations of the original data which contained 49 features.

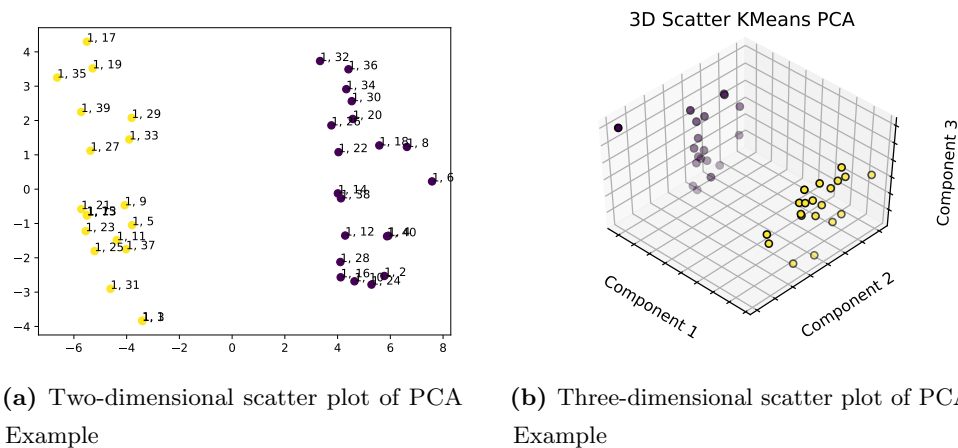


Figure 4.43: Example PCA visualization application with two- and three-dimensional scatter plots (colouring from K-Means clustering labels)

The figure mentions that the colours of the points represent the labels that the clustering grouped each point into, the labels for each point represent the ground truth of the protocol. This situation applies for all clustering figures in this document.

4.10 Experimental Optimization Process

The design of the methodology utilized here: implementing feature extractors that generate many different and unique features, implementing multiple different feature

normalizers as well as different ground truth and clustering techniques - all stems from the experimental optimization process. This design is structured in this manner because of the unique situation that this methodology has not been applied to such a system before. It is not known what combination of features/normalizers/clustering's will work well for the shapes and characteristics of the transients of the system.

The design of this system is to enable the testing of each of the different possible combinations to ascertain which performs better under these circumstances with a final goal of verifying whether NILM and load disaggregation are viable techniques to apply to such systems. This system can be visualized and understood as a tree diagram where the end leaf nodes are the different combinations of features selected, feature orderings, feature normalizers, and clustering techniques.

Chapter 5

Analysis

This analysis section is broken down into four major sections which outline the different major questions that are tested on the systems: the two key characteristics of devices are investigated in sections 5.1.1 and 5.1.2 where the uniqueness of single devices are verified and the repeatability of transients are observed. The ground truth validation techniques are illustrated, and a hybrid ground truth is observed in certain protocols based on devices noise, this is discussed in section 5.2. The sampling limitations of applying NILM and load disaggregation to such systems are then covered by observing how the disaggregation performance is tied to the power supply and load magnitudes, this is discussed in section 5.4.

5.1 Device Analysis

Section 4.5 outlined the different protocol tests that are created to answer the questions related to the characteristics of devices on the system. Specifically, section 4.5.2, which looks at a number of different devices used within the system; this section attempts to observe the two major characteristics that need to be satisfied in order to accommodate NILM and load disaggregation on these systems: device state transition signature uniqueness, and device state transition signature repeatability. These characteristics of devices on the network are integral to the research questions.

5.1.1 Unique Device Signatures

The protocols utilised for this test include each of the different devices that are included in the system (see section 4.2) where the devices are setup and connected to the standard bench top power supply as the grid. The observations in these protocols, using the bench top power supply also apply to the different source combinations, more detail specific to the sources is provided in section 5.3.

There are two parts to each of the tests that are carried out here: the first is to observe whether the different state transitions of a single device are unique. What this means, for the case of a type I device, is that the $0 - 1$ state transition, and the $1 - 0$ state transition transients are different from one another.

The second requirement here is that different devices also present different transient signatures from one another: the $0 - 1$ state transition from one device is different from the $0 - 1$ of another (the same needs to apply for the $1 - 0$ case as well as for type II device cases).

These two requirements are outlined and discussed in sections 5.1.1.1 and 5.1.1.2.

5.1.1.1 Singular Device State Uniqueness

The testing protocols for this set of tests, outlined in section 4.5.2 under the *Single Device Operation* protocol category tested each of the different devices available. The protocols include devices of type I and II, with ranging power requirements in each class of device.

To illustrate the intra-device uniqueness (differences between the state changes for a device) for this set of tests, two type I devices are observed here to highlight the unique signatures that the different state transitions exhibit on the grid voltage. The first device illustrated in fig. 5.1, due to its large power consumption (when compared to other devices tested), results in large deviations from the steady-state voltage level, and slow voltage recovery on the grid. The transient signature that this device exhibits in the two state transitions can be seen to be vastly different from another visually.

The second device, a single light bulb, which requires less power to operate than the first device, has a smaller effect on the grid voltage as illustrated in fig. 5.2.

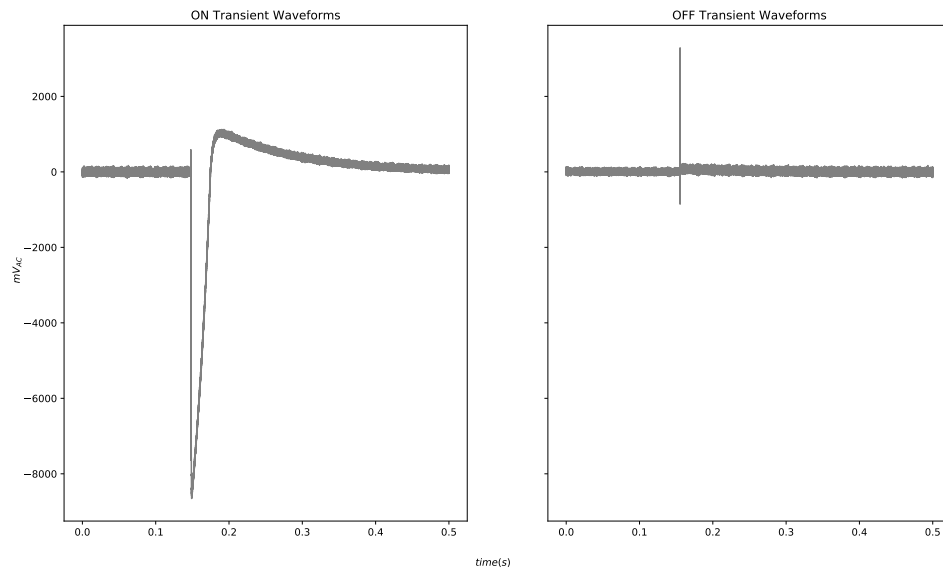


Figure 5.1: High-power type I device - state transitions

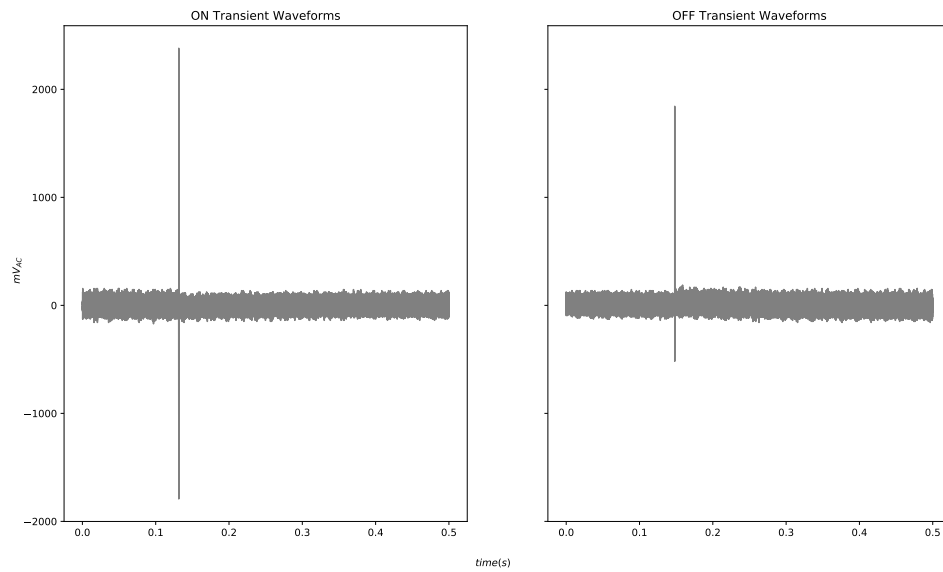


Figure 5.2: Low-power type I device - state transitions

The grid voltage does not take as long to return to its steady-state; however, an important consequence to illustrate in this circumstance is the difference in the minimum and maximum deviations that the voltage experiences in the different state transitions. The first state transition has large negative and positive deviations on the grid voltage whereas the second state transition has a large positive deviation but a smaller negative deviation. Visually, the shapes of the state transitions for this device are not as easy to differentiate as the first device observed in fig. 5.1, however, there are differences which are important for the disaggregation process because the differences enable the ability to differentiate the two state transitions

from one another.

Observing the signatures at this scale offer some benefit, however, observing just the transient (by removing the steady-state samples from either side) reveals the important signature shape that the light exhibits which is illustrated in fig. 5.3. This highlights the fact that even devices which have minimal effect on the grid voltage are still able to be observed, while also having unique state transition transients.

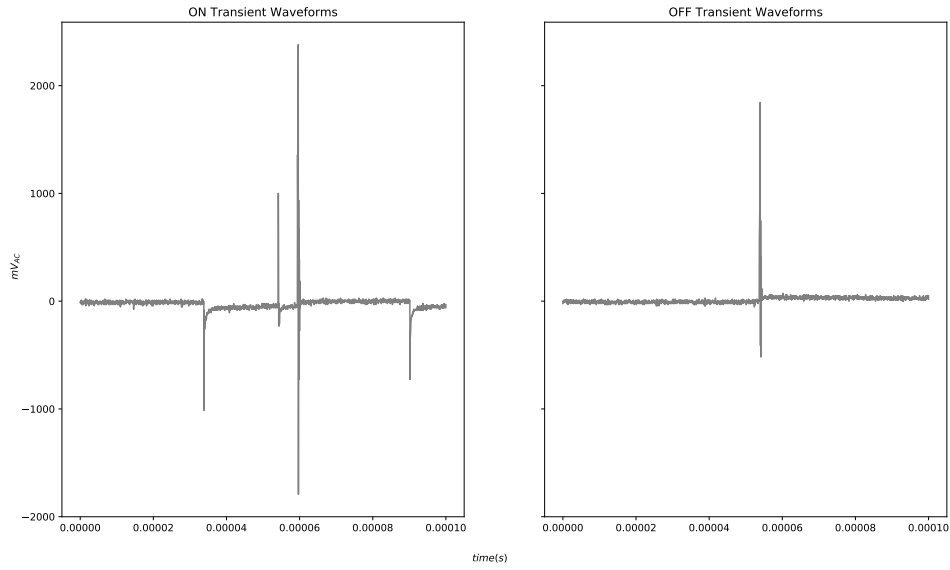


Figure 5.3: Low-power type I device zoomed - state transitions

These figures, which illustrate the time series plots of the transients, are converted into their respective feature vectors, the features which hold a lower dimensional version of the signatures are then clustered. These can be illustrated by figs. 5.4a and 5.4b where the two sets of clusters in each of the different plots, which are created from the two devices here, can easily be differentiated from one another. The clusters observed here use the ground truth information to apply the labelling onto each of the points; additionally, the clustering algorithms have a high degree of acceptance with the ground truth data (this is represented by the colours of the events), which suggests that it is feasible to observe and distinguish the differences of state transitions of devices.

This highlights the fact that even though the two different state transitions observed in fig. 5.2 visually appear to be similar, the feature extraction, filtering, and clustering process is readily capable of differentiating the two state transitions from one another.

These clustering representations of the data are created with the use of the PCA algorithm and the first two components can be visualised with the use of a 2D scatter

Figure 10 consists of two scatter plots, (a) and (b), showing the K-Means clustering of type I devices. Plot (a) is for high-power devices, and plot (b) is for low-power devices. Both plots show the device coordinates (x, y) with clusters labeled by their centroid coordinates.

(a) High-power type I device K-Means clustering

(b) Low-power type I device K-Means clustering

This clustering visualisation highlights the fact that the different state transitions are unique, however, there is still some variation from one event to another, meaning that depending on the device, the clusters may not have points that are densely populated. This is highlighted by the distribution of points in fig. 5.4a where the $0 - 1$ transients are more spread out than the $1 - 0$ transients. This characteristic is discussed in greater detail in section 5.1.2 however, it is important to note from these figures that the different state transitions are different from one another, visually and also when the events are transformed into their feature vector representations.

The important note to highlight, particularly in fig. 5.5 is the fact that the $1 - 0$ transients form a tighter cluster than the $0 - 1$ cluster, this is represented by the merging of clusters lower down on the left hand side leaves of the dendrogram.

The dendrogram illustration here lines up with the distribution of events in fig. 5.4 of the clustering.

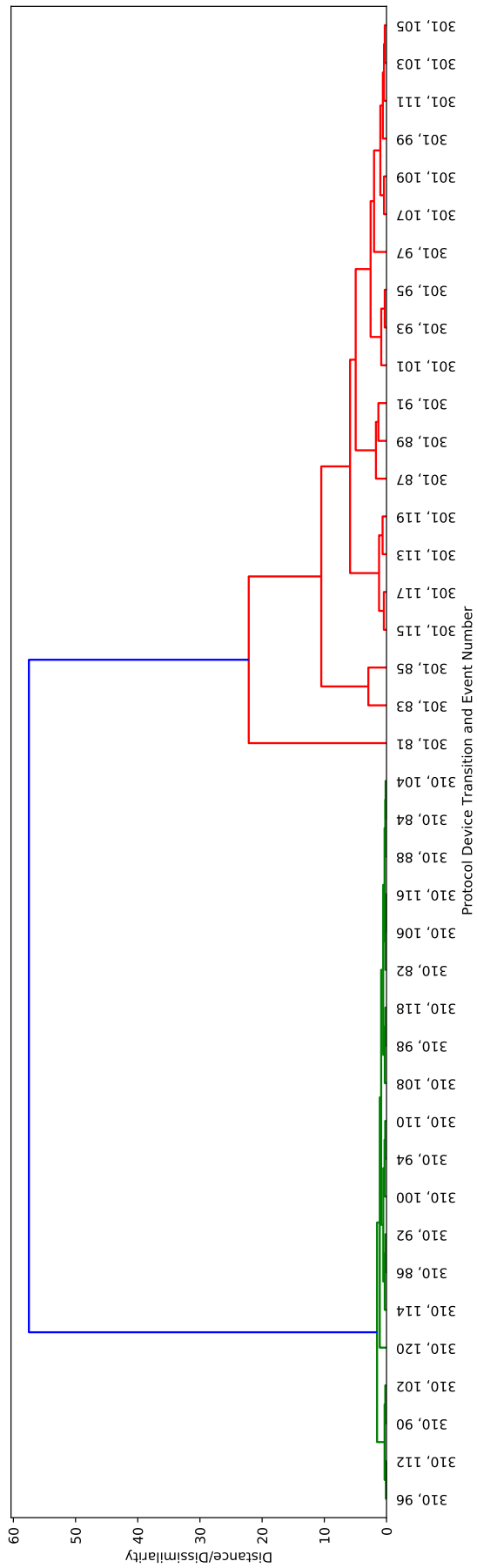


Figure 5.5: High-power type I device dendrogram (Even numbered events are ON events, odd numbered events are OFF events)

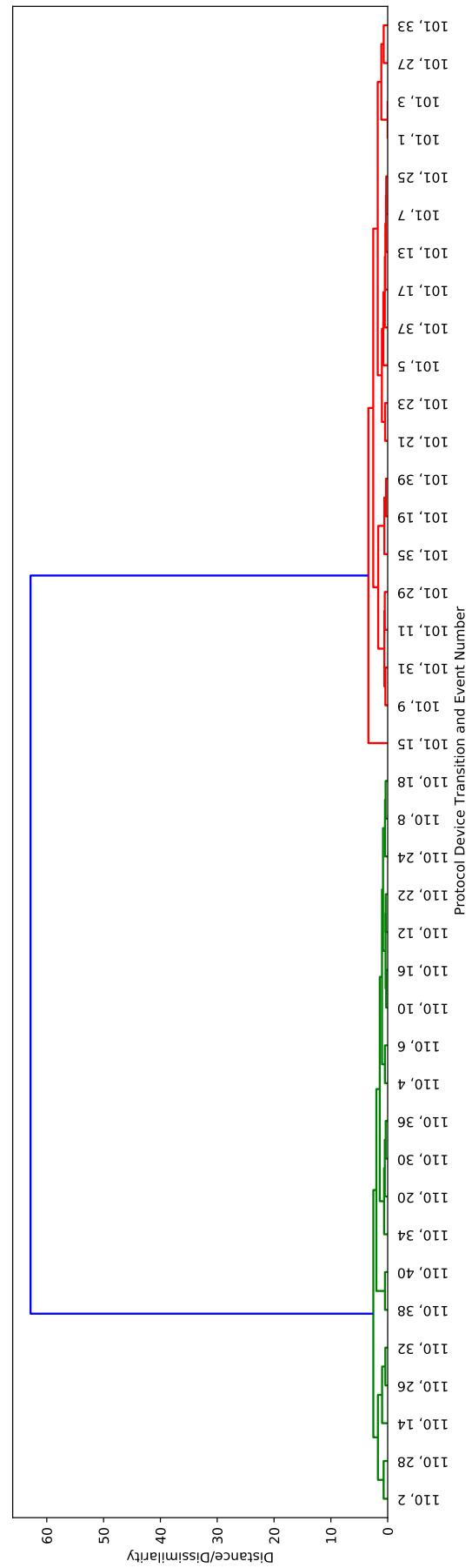


Figure 5.6: Low-power type I device dendrogram (Odd numbered events are ON events, even numbered events are OFF events)

Noting this, in reference to the requirements of load disaggregation: denser and further apart clusters are preferred in the disaggregation process as it makes it easier to tell the types of transients apart.

The observations made here also apply to devices of type II: fig. 5.7 illustrates the state transitions for a device with three possible states. This device, because it has three possible states, will have a six possible different state transitions, $0 - 1$, $1 - 0$, $0 - 2$, $2 - 0$, $1 - 2$, $2 - 1$.

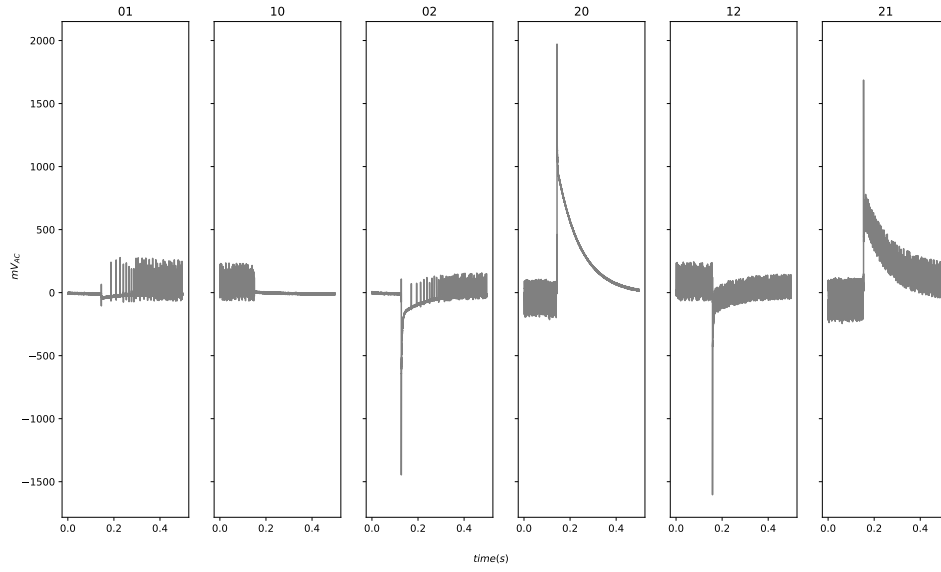


Figure 5.7: Type II device - state transition waveforms

The same conclusions made for the first two devices also apply to this type II device, further confirming that devices have unique signatures for their different state transitions. The clustering and dendrogram representation in figs. 5.8 and 5.9 illustrate this characteristic of the type II device.

What is interesting to note with this situation is the fact that there are two major sections outlined by the dendrogram. The two major branches are split by the presence of state 2. State transitions of $0 - 1$ and $1 - 0$ both represent the turning ON and OFF of the fan unit inside of the device, which has a lower power draw than the component used in any of the state transitions involving state 2: $0 - 2$, $2 - 0$, $1 - 2$, $2 - 1$. This means that the cluster formation, and how the agglomerative clustering algorithm sees the underlying feature vectors, implies that the turning ON or OFF events of the internal heating element in this device create much larger transient signatures which is also visually represented (mirrored) in fig. 5.7.

The characteristics observed for these three devices also apply each of the devices

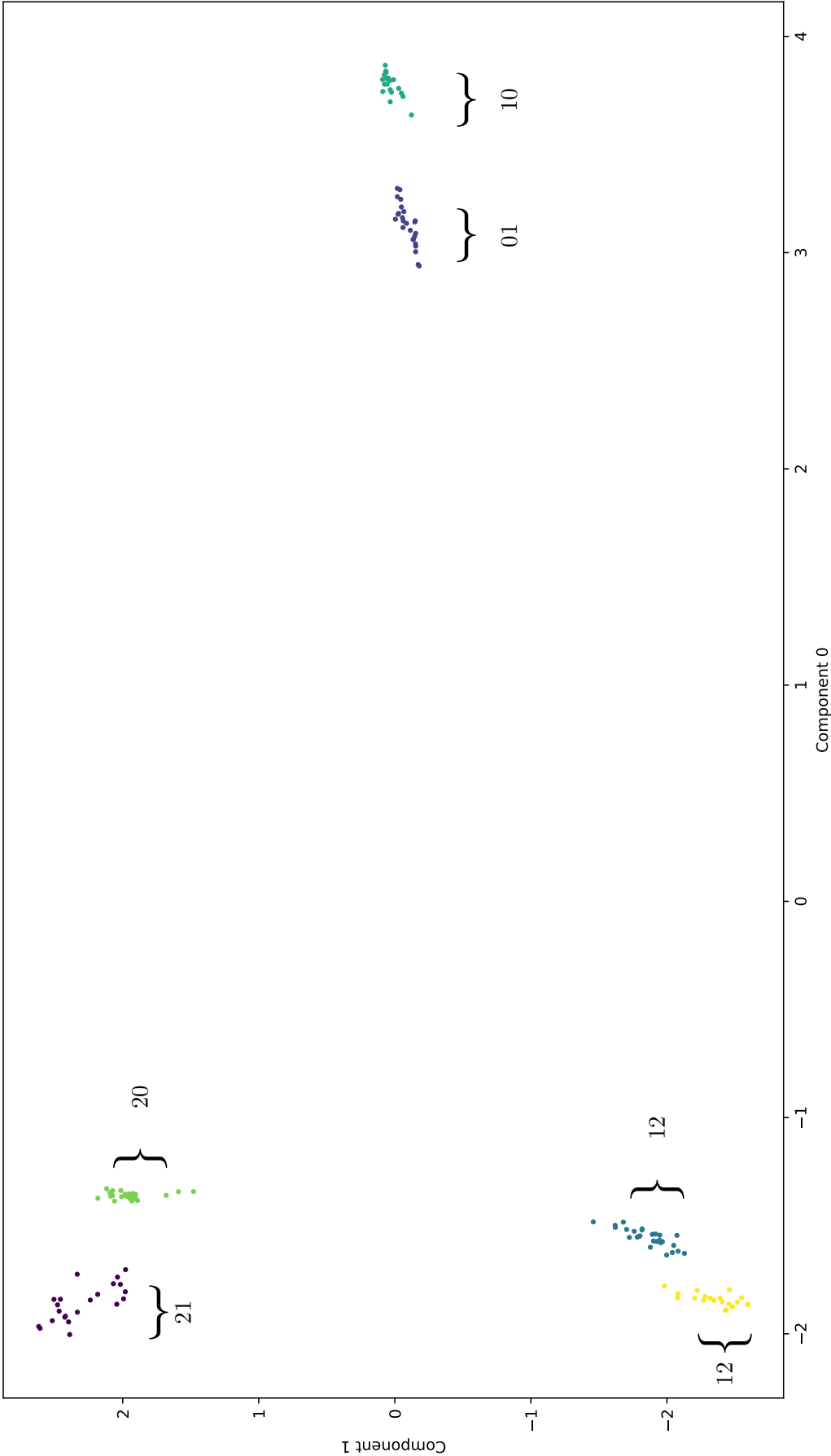


Figure 5.8: Clustering representation of type II device

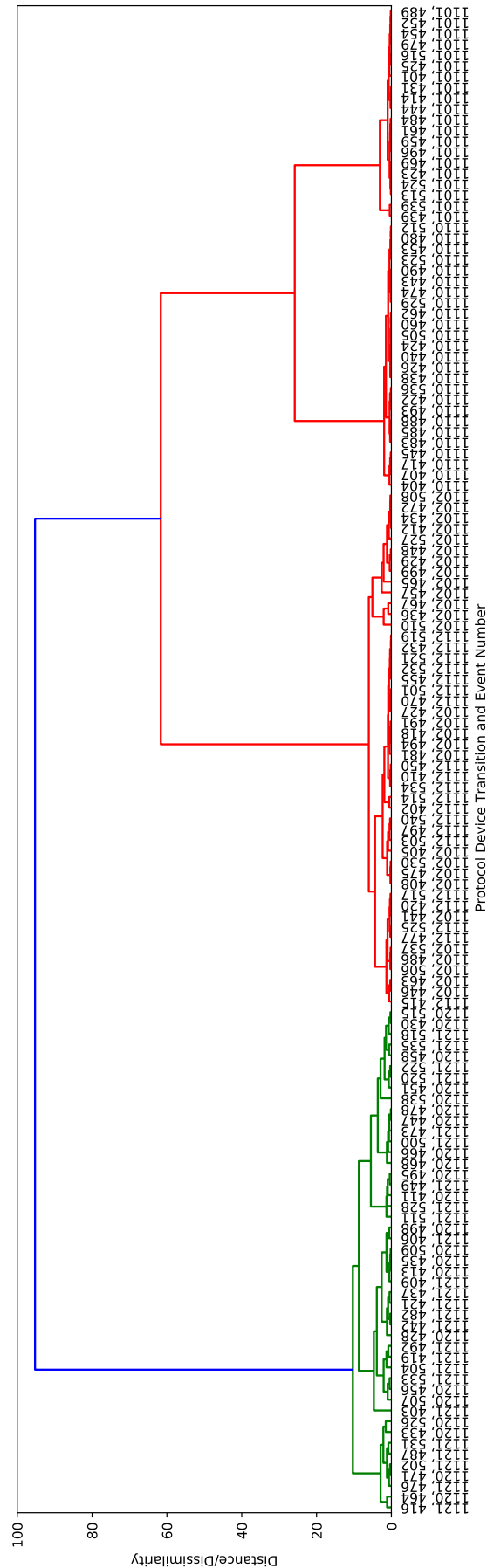


Figure 5.9: Dendrogram representation of type II device

observed within the diverse range available for this research. This means that, in general, it is indeed possible to distinguish the different state transitions of a number of devices based on their transient signatures as they are differentiable from one another.

These results also apply to the different grids tested throughout; the analysis is discussed further in section 5.3.

Furthermore, the waveforms observed in fig. 5.7 highlight the fact that this particular device exhibits a large amount of voltage noise onto the grid, this affects the ability for load disaggregation to be performed when these devices are operating in these modes; section 5.2.1 discusses the presence of this noise and the resulting concerns that this represents.

This conclusion here however, must be coupled with the results in section 5.1.2 as it is important that, not only must the signatures be unique, they must also be repeatable, this was touched upon in the clustering and dendrogram representations of the protocols here.

5.1.1.2 Different Devices State Uniqueness

This further test applied here seeks to compare the different state transitions when looking across different devices, this is where the inter-device state transitions need to be observed which is illustrated here by using the same technique of cluster visualization as in section 5.1.1.1 to illustrate this requirement. This test seeks to determine whether the 0 – 1 state transition of one device is different from the 0 – 1 state transition of another device.

This first scenario looks at two type I devices, each containing two state transitions. In order to effectively apply techniques of load disaggregation, the signatures of each of these devices all need to be unique, this means that in the feature space, four distinct clusters need to be observable. This requirement is based on the understanding that each device has a unique transient signature for all of its state transitions, and each of these state transitions need to be sufficiently different from all other transient signatures of the different devices in the system.

To illustrate this requirement of device state transition uniqueness, the event feature vectors of a number of devices are placed into the same feature space, clustering algorithms are then applied to these points in order to ascertain whether all of the

different signatures are sufficiently different from one another.

To illustrate this fact, devices two and four are clustered on the same feature space, these devices have similar power ratings.

The clustering of this feature space is illustrated in fig. 5.10 where it is evident that there are four different clusters for the two devices operating.

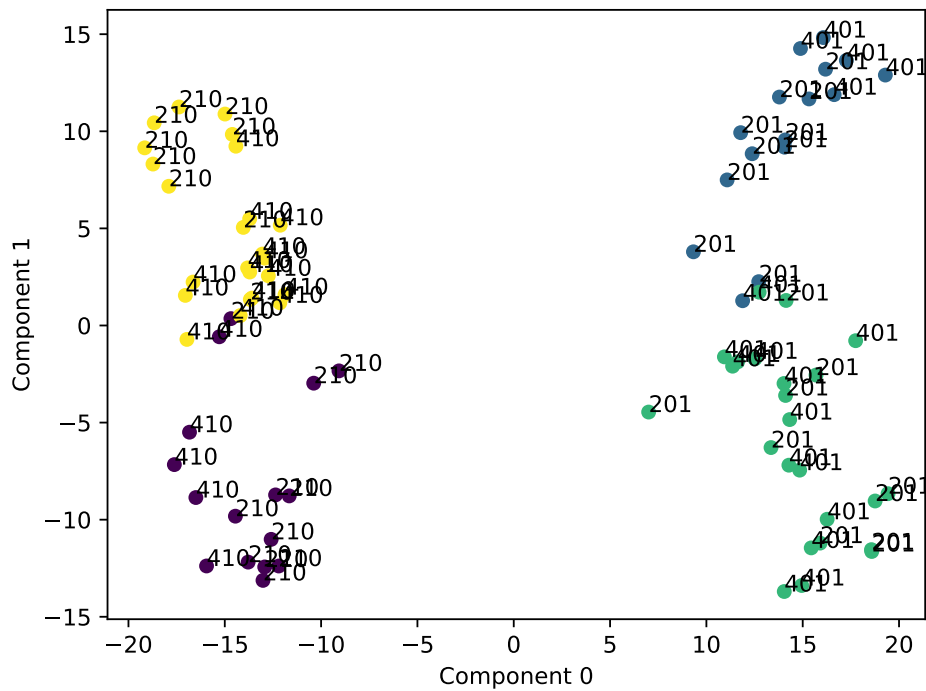


Figure 5.10: Two devices (type I) clustering

This situation represents a particularly difficult clustering case as the two devices operate at low-power which implies that they are all likely to produce similar transient signatures. However, even here, it is clear that it is indeed possible to discern the different operating state transitions of the different devices based on the fact that their different event state transitions all form their own distinct clusters.

Note however, that this illustrates a situation where some events may overlap over onto other clusters for other devices. This implies that, with increasing devices, the feature space may become crowded and the system is likely to have a corresponding drop in clustering performance.

To illustrate this, table 5.1 contains the ARI values for situations where increasing number of devices are clustered in the same feature space. It is clear from this, that as more devices are added to the network, the clustering performance will drop and

Table 5.1: Clustering performance for increasing devices in feature space

Number Devices Clustered	ARI	
	K-Means	Agglomerative
2	1.0	1.0
3	1.0	1.0
4	1.0	0.8241
5	1.0	0.9016
6	0.810 34	0.807 59
7	0.767 11	0.834 50
8	0.681 16	0.755 12
9	0.697 68	0.675 53

This situation makes use of 300 (350 for agglomerative) features, based on the results concluded from the feature analysis in section 4.8.4.1

therefore the ability for the system to successfully disaggregate.

Note here that the devices used within this test are all mostly low-power devices (within the 0 W to 15 W range) and likely produce similar feature vectors for the transients observed. Real world implementations often have wider varying devices within networks which produce increasingly unique features. Additionally, these distributed DC systems, due to the fact that they are still young in the technological penetration, are likely to be small installations with a limited number of devices, so the difficulty in having many devices on the network is unlikely to be a problem for most scenarios.

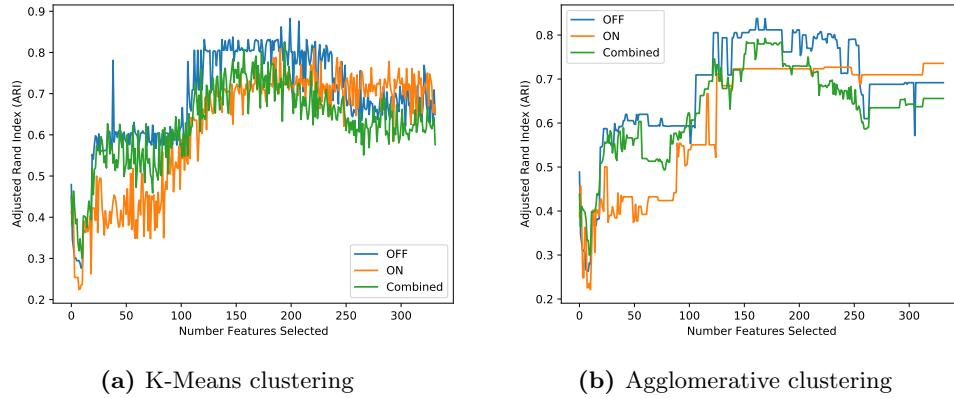
A further observation that is important to note here is the difference in the ability to differentiate between the ON transients of different devices and the OFF transients of different devices as seen in Table 5.2.

To illustrate this situation, nine devices are placed on the same clustering space, their ON and OFF events are split into two clustering runs. These are then compared with one another, fig. 5.11. This highlights the fact that the OFF transients that are created from devices appear to have greater uniqueness and so their signatures are more descriptive when used for load disaggregation purposes.

There are limitations to this however, specifically cases where there are devices which are exact replicas of one another, this unique case is discussed in greater detail in section 5.1.3.1.

Table 5.2: Clustering performance for increasing devices in feature space comparing ON vs OFF

Number Devices Clustered	On Transients ARI		Off Transients ARI	
	K-Means	Agglomerative	K-Means	Agglomerative
2	1.0	1.0	1.0	1.0
3	1.0	1.0	1.0	1.0
4	1.0	0.966 23	1.0	0.719 51
5	0.829 015	0.974 61	0.950 58	0.616 30
6	0.807 38	0.789 52	0.762 65	0.578 94
7	0.795 74	0.8221	0.717 89	0.569 82
8	0.690 08	0.7357	0.659 07	0.691 72
9	0.651 50	0.7357	0.649 52	0.691 72

**Figure 5.11:** ON and OFF clusterings of nine devices

5.1.2 Repeatability Testing

Section 5.1.1.1 illustrated the fact that devices, whether they are type I or II, or whether they are high-powered or low-powered, all have transient signatures that are different from one another, both in their intra-device state transition signatures as well as inter-device state transition signatures.

To apply NILM and load disaggregation, a further requirement needs to be satisfied: the events recorded for a device when changing from one state to another need to be the same over time. This means that successive state transitions of a device from 0 – 1 should, relatively consistently, have a similar transient signature. This repeatability characteristic of devices is required here as it enables the system to observe a transient event, place it into the feature space where it will then fall into a cluster that contains other points of the same devices state transition from the past,

thus creating much needed predictability. This predictable nature then allows the disaggregation process to continue and for a model to be formed for a devices state transition signature as the system can keep track of a state transition signature for devices within the system.

This section will use the same devices that were illustrated in section 5.1.1.1 where each of the switching events within the protocols are observed here to highlight the repeatability of the transient signatures of the devices.

The first two devices, both type I, each have two state transitions to observe, fig. 5.12 observes the time series representation of the high-powered light on the system; the different events are overlaid on one another for visual evaluation and comparison. Section 4.4.4 discussed the variance in the timing of the switching taking place and the starting of the recording, this accounts for the small x-shift of the different waveforms, where fig. 4.15 illustrates the variance and how the events may be located in the measurement window.

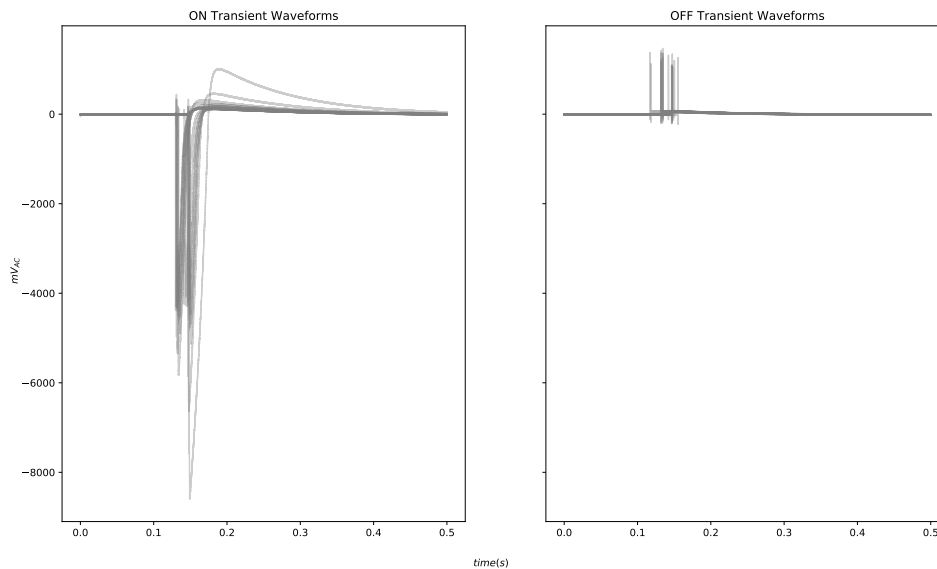


Figure 5.12: High-power type I device transient waveform repeatability

Figure 5.13 illustrates the same situation for the low-powered device where the entire event window is created as well as the zoomed in version of the same events.

Once again, a similar situation applies to devices of type II, fig. 5.14 illustrates the repeatability for the six different cases possible for the device.

Evident in these overlaid event signatures is the fact that there are still small variations from one event to another, this is evidenced when observing the clustering

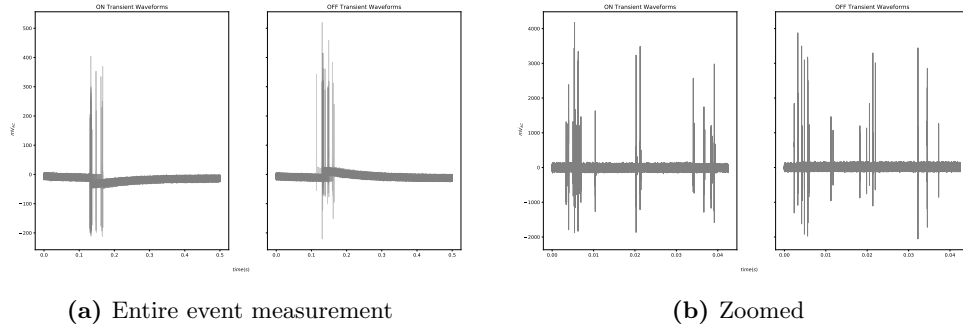


Figure 5.13: Low-power type I device transient waveform repeatability

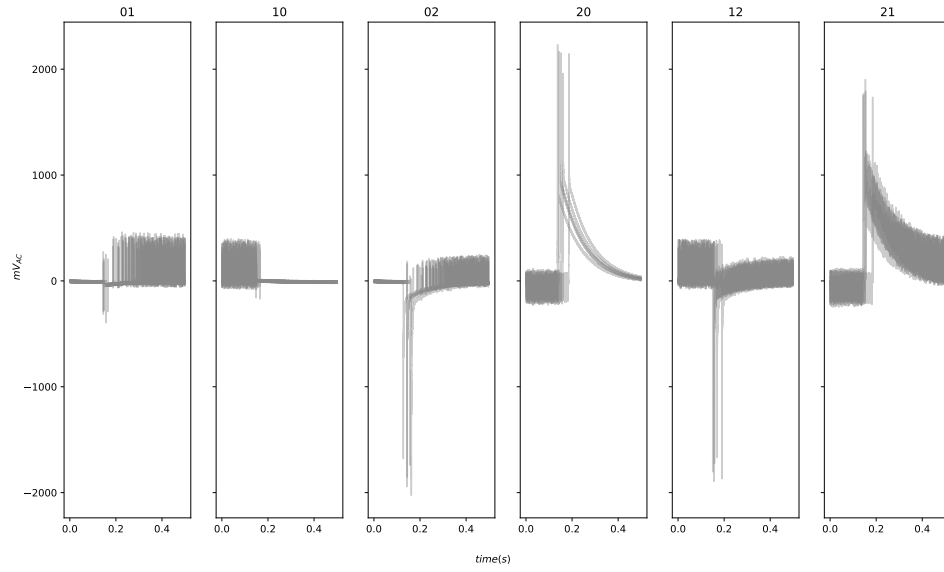


Figure 5.14: Type II device transient waveform repeatability

diagrams of the protocols of these devices which are illustrated in section 5.1.1.1.

To further solidify the presence of the repeatability of signatures, it is useful to use statistical measures to determine the level of similarity of one 01 (or 10) transient to the next. A standard technique that is capable of observing this similarity between different datasets (one event against others) is to obtain the cross-correlation.

5.1.3 Device Signature Discussion

The figures and conclusions from sections 5.1.1 and 5.1.2 cover the time series representations of devices operation on the grid. The tools utilized to create the scatter plots for the clustering process, as well as the dendrograms, in this section are created to illustrate how different feature selection types and different clustering methods perform.

Table 5.3: Clustering performance for single devices

ProtocolID	Agglomerative		K-Means	
	Feature count	ARI	Feature count	ARI
1	1	1.0	1	1.0
2	1	1.0	1	1.0
3	3	1.0	1	1.0
4	1	1.0	1	1.0
5	4	1.0	1	1.0
6	1	1.0	1	1.0
7	1	1.0	1	1.0
8	1	1.0	1	1.0
9	164	1.0	164	1.0
10	3	1.0	3	1.0
11	51	1.0	51	1.0
12	36	1.0	24	1.0
13	146	0.9185	180	0.919

Minimum number of features to achieve 1.0, or number of features that corresponds to best ARI.

To illustrate how the features here assist in the disaggregation process, the ARI scores are plotted by increasing the number of features selected. This highlights the fact that, for the first few protocols, distinguishing the differences between the state changes for individual devices is relatively easy to perform, this is understood by the fact that even with a single feature, the system has an ARI of 1.0.

The following table illustrates the performance on clustering for the individual devices. This illustrates the ease with which the two state transition types of device type I devices can be distinguished from one another: the clustering algorithms only require a single feature in order to achieve an ARI of 1.0.

This illustrates that, depending on the device, more than a single feature is required to achieve an ARI of 1.0; devices of type II (protocol ID's 11, 12, and 13) appear to require more features than other devices to achieve their highest scores.

The analysis of the features and their selection for clustering is discussed in section 5.4.2.

5.1.3.1 Multiple Replica Devices

A test that is covered by other research in this field highlights the fact that it is possible for electrical networks to have multiple devices which have the exact same electrical and physical ratings [39].

The research in this field which covers this fact discuss these situations and suggest that, even though two devices may be exactly the same in their electrical and physical designs, it is possible to find differences in the signatures of the two devices based on the physical installation of the devices. This previous work in the field suggests miniscule differences are present and can be found when observing the Fourier coefficients of the devices - using a high resolution and high sampling rate device [39].

Their reasoning on this is based on some of the following: small differences in lead lengths, manufacturing tolerances, and other small characteristics of the exact installation.

The protocols that covered these tests involve the use of two low-powered LED lights from the same manufacturer with the exact same model numbers (same power rating, light intensity output, etc.). The first test set the two lights in such a way as to ensure that their installation was as similar as possible: same lead lengths, and geometrical layout of the wiring connections to the grid. The original lead lengths of each light is ~ 3.58 m. Figure 5.15 illustrates the clustering space of this first test.

This figure suggests that, with the measurement device (measurement resolution and sampling rate) and measurement parameter used within this research (measuring the grid voltage), it is not possible to distinguish the two devices from one another. The only differences that the clustering algorithms can observe is that the 01 transients of each device look the same (the same applies for the 10 transients) as one another. The labelling of each of the points over here (unfortunately overlapping in some cases) highlight that the ON transients of one device look highly similar to the ON transients of the other device in question (the same goes for the OFF transients). What is important to illustrate here is that the clustering algorithm is assigning the colours to what it thinks are the four clusters, however, the four clusters it believes are correct, are in fact a combination of the two devices, which according to the ground truth data, is incorrect. This means that the system is capable of detecting that *a* light of a specific rating changed state, however it cannot determine *which* of the two lights changed state.

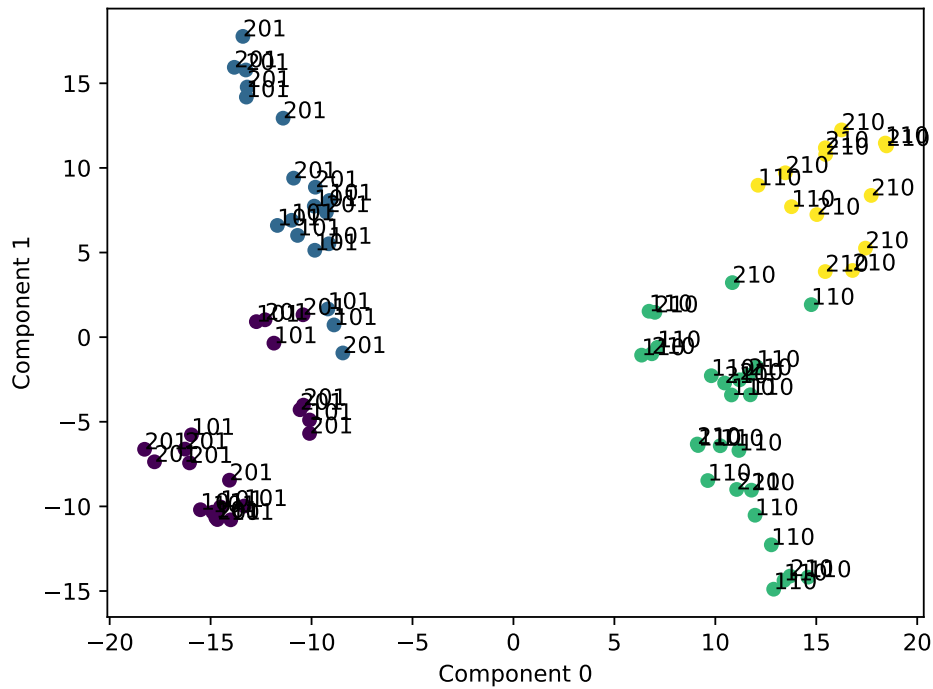


Figure 5.15: Replica device - same physical installation

Further tests are applied to the same setup which attempts to use DC coupling for the measurements (instead of AC coupling, see fig. 4.14), to reduce the vertical resolution of the measurement (see fig. 4.13), to see if this would be useful in observing any difference between the two lights. These tests proved unsuccessful, table 5.4 lists the different tests and their highest performances accomplished.

The results here suggest that none of these tests appear to achieve sufficient performance gains; the results suggest that the system is only capable of observing two distinct clusters: the ON and OFF of the two different lights.

One of the lights is then modified to have a shorter lead (1.72 m), leaving the rest of the installation exactly the same. The result of this, illustrated in fig. 5.16 where there is now a greater distinction between the four clusters for the two devices. The colours of the points represent the labels that the clustering assumes, the labels for the points represent the ground truth. The plotting tool utilized here begins to show how there is a separation of the four clusters, the further separation is found in the other components that the clustering utilized (seen in the other components, not simply in these first two components visualized here).

This small difference, the reduction of approximately 1.86 m of lead length provides the ability to distinguish the two devices much easier. This result generally lines up

Table 5.4: Performance of two similar lights with different testing conditions

Test number	Protocol Description	Coupling	Vertical Scale	Highest ARI (Normalizer)	Highest ARI (with StandardScaler)
1	Physical installation same for both lights	AC	±20 V	0.6007 (RS)	0.577 16
2	Lower vertical scale	AC	±10 V	0.565 (QTG)	0.528 816
3	Lower vertical scale	AC	±5 V	0.5303 (QTG)	0.530 301
4	Lower vertical scale	AC	±2 V	0.5727 (QTG)	0.566 24
5	Lower vertical scale	AC	±1 V	0.5644 (QTG)	0.534 378
6	DC coupling	DC	±20 V	0.5530 (QTG)	0.514 054
7	Probe set to 10x	AC	±20 V	0.4809 (QTG)	0.463 502 5

QTG: Quantile Transformer Gaussian. RS: Robust Scaler

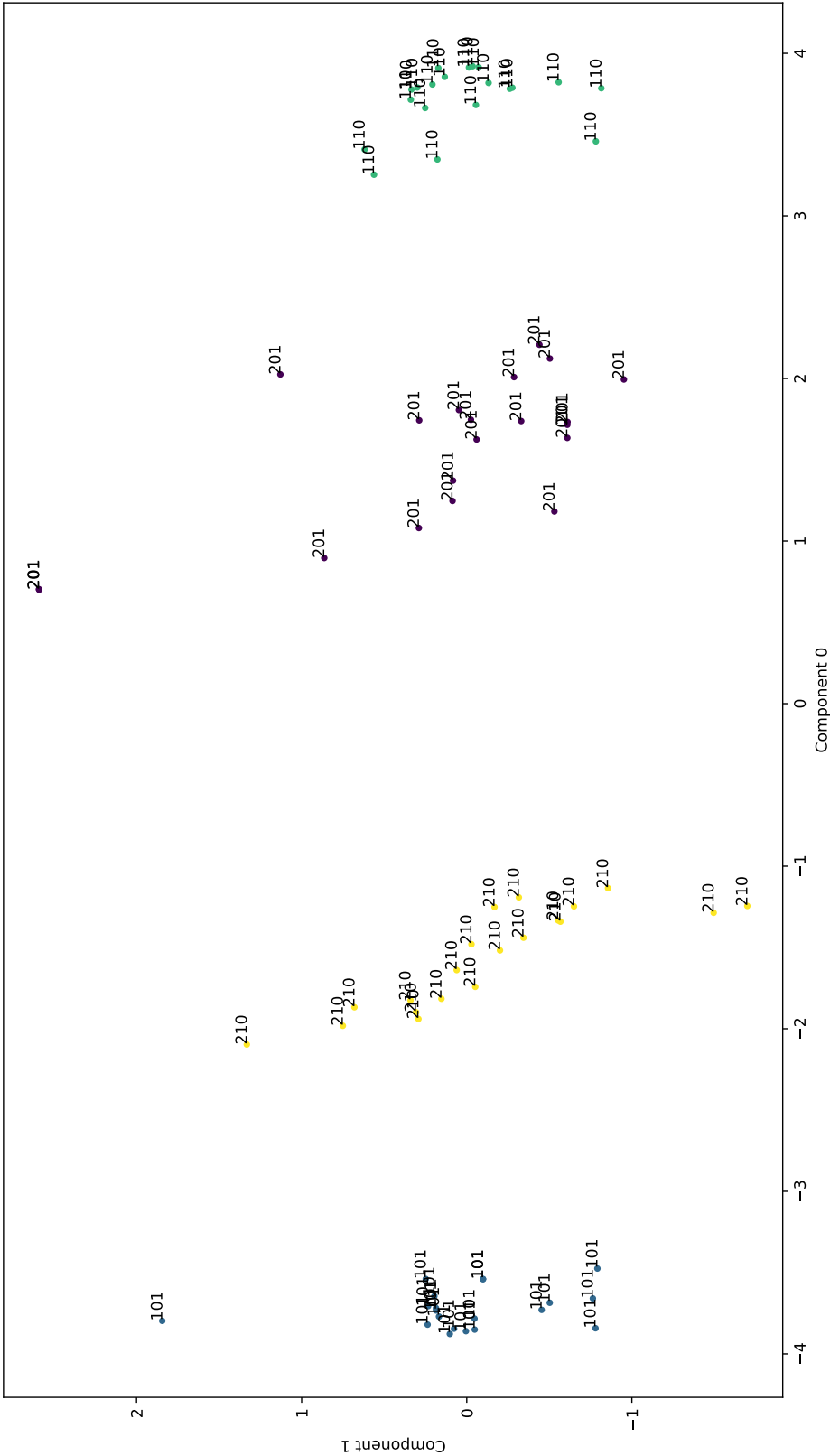


Figure 5.16: Replica device - LED 2 shortened leads

with those made with previous research in the field of this scenario. The results here show that, the sampling rate and measurement resolution available, coupled with the feature extraction tools utilized, it is not possible to distinguish the two devices from one another. The differences only appear when the devices have a different physical installation¹. The previous research was based on an AC system, however their findings are similar to the situation found here - differences in the features of two similar devices are only practically visible when the devices are placed at different *distances* from the main source (from where the measurement point is located which imply a different electrical configuration) [39, p. 8-9].

The two devices utilized in this set of tests both had low-power requirements, meaning that their impact onto the grid is minimal and so the transients, and therefore the corresponding feature vectors may only have miniscule differences. A further test to attempt is to find two high-wattage devices that are the same make and model and to further this test, this can form part of future work in the field.

5.2 Ground Truth Concept

The concept of ground truth, covered throughout this work, is integral to the functioning of the validation steps. This section outlines the key findings of the ground truth with different scenarios.

Generally, the simple ground truth concept introduced in section 4.6.2 has proved to be the most predictable method of system transient analysis.

5.2.1 Device Noise

The presence of noise emitted by certain devices when operational has been seen to confound the disaggregation process, this leads to a number of conclusions that are tied to the two types of ground truth generated for this work.

To illustrate this phenomenon, a protocol is created which has two devices, the first is a low-power LED light (type I device), and the second is a small fan, commonly used in PC's. The second device (also of type I) emits noise onto the grid during operation thus introduces an interesting scenario to observe.

¹It must be noted a high ARI is achieved (1.0) in some of the feature normalizing types and not all, this is consistent with other clustering results as well.

The first light, the transients of which are presented in fig. 5.2, is compared against the transients that are generated by the PC fan in fig. 5.17. These waveforms illustrate that the PC fan, when it is running, emits noise onto the grid, this noise is noticeable on the grid voltage.

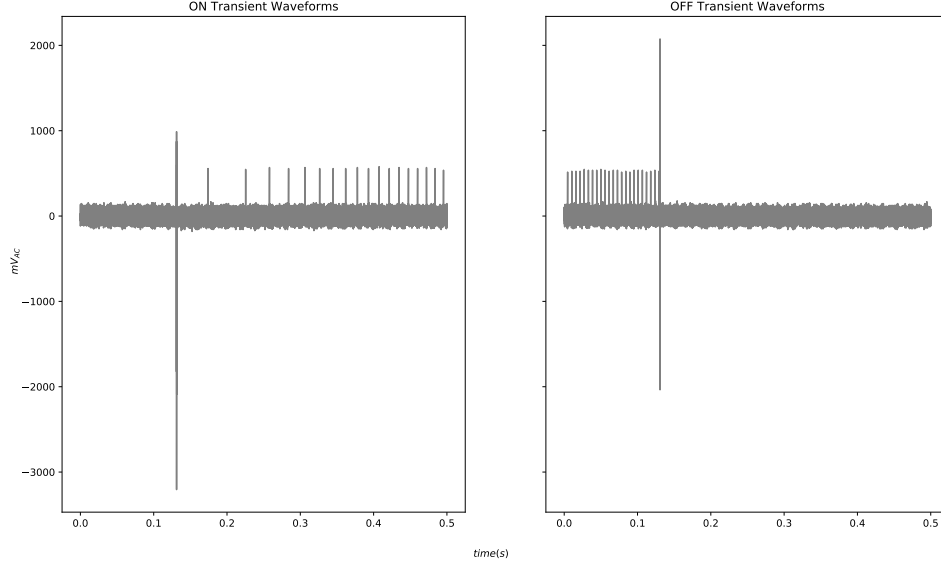


Figure 5.17: PC fan type I device - state transitions

The two protocols that test the ability to cluster these two devices in isolation can perform sufficiently by easily achieving an ARI of 1.0 (see table 5.3). However, when the PC fan is operating, there is noise introduced onto the grid which is present for the transients of the switching of the light, this is illustrated in fig. 5.18 where it is clear that the event measurements now contain more noise on the system from the operation of the fan.

This now clearly suggests that the simple ground truth is not practical as it does not capture the new system dynamics at play. This means that the signatures of the LED light are now also dependant on the state of other devices on the grid - devices that emit noise.

This result, however, is specific to cases where the device emits noise onto the grid; two further protocols are utilized in this study which only use devices that do not (or emit small amounts) of noise onto the grid when operating. These sets of tests highlight that the signature of a device is not changed (i.e. is not dependant) on the state of other devices on the system. This is relevant to the protocol test utilized in this section as it is clear that the signatures of the PC fan turning ON and OFF are not affected by the state of the light: the noise of the PC fan is the confounding

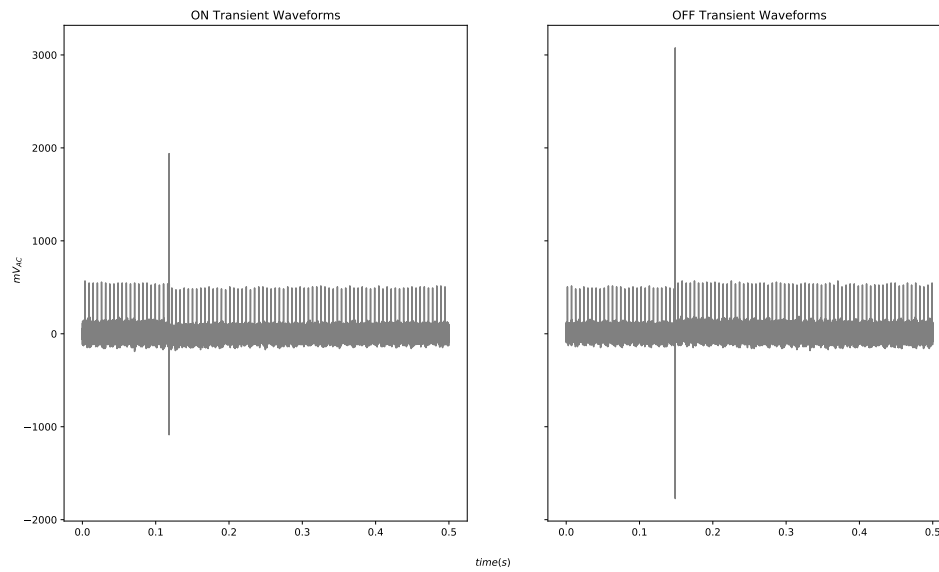


Figure 5.18: LED Light transients with fan operating

factor that is introduced into the system.

This results in a situation where only the state of the fan affects the signatures of the light: this scenario does not abide by the simple or complex ground truth. This is a one-way dependability of the two devices. This is a third situation where there are instead six clusters which can be described by:

- cluster 1: Fan OFF, light turning ON (40, 101),
- cluster 2: Fan OFF, light turning OFF (40, 110),
- cluster 3: Fan ON, light turning ON (41, 101),
- cluster 4: Fan ON, light turning ON (41, 110),
- cluster 5:
 - light ON, fan turning ON (401, 11), and
 - light OFF, fan turning ON (401, 10).
- Cluster 6:
 - light ON, fan turning OFF (410, 11), and
 - light OFF, fan turning OFF (410, 10).

This third situation is a hybrid between the simple ground truth and the complex ground truth: this situation is present based on situations where some of the transients

Table 5.5: Protocol with two devices where one device emits noise (300 features selected)

Clustering Algorithm	ARI Score for selected cluster count		
	4	6	8
K-Means	0.6703	0.8025	0.728 16
K-Means PCA	0.6703	0.7409	0.6880
Agglomerative	0.6703	0.8025	0.7970
Agglomerative PCA	0.6703	0.8025	0.7970

of devices change depending on the state of other devices on the system, if those other devices emit noise.

To illustrate this, the clustering algorithms are initiated by assigning a total of six clusters, this greatly increases the ARI score of the system; table 5.5 illustrates the choice of the different cluster counts which highlight the fact that the choice of six clusters results in a greater ARI score compared to other choices. This table also illustrates that this characteristic of the choice of clusters is consistent across the different clustering techniques, there is a high level of agreement on the separation of the different clusters.

To exhibit these different scenarios, the different number of clusters selected is illustrated by observing the K-Means PCA case and selecting the first two components, the labels for each of the points represent the event that took place, the labels line up with those in the list above. The two figures illustrated here highlight the fact that selecting six clusters appears to be (visually) a better clustering for the points in the feature space as opposed to selecting eight clusters. The colours of the points illustrate the category that the clustering algorithm assigned.

This characteristic of devices on the system, the ability for some devices to change the transients of other devices further complicates the disaggregation process. This also means that, depending on the magnitude of the noise introduced by devices, it may occur that the noise can completely mask the transients of devices that are changing state during that time. The presence of this may result in increased complexity of disaggregation; section 6.1.4.4 discusses some potential ideas that may reduce the effects of these devices as well as techniques used in the field of NILM and load disaggregation to counteract the effects introduced.

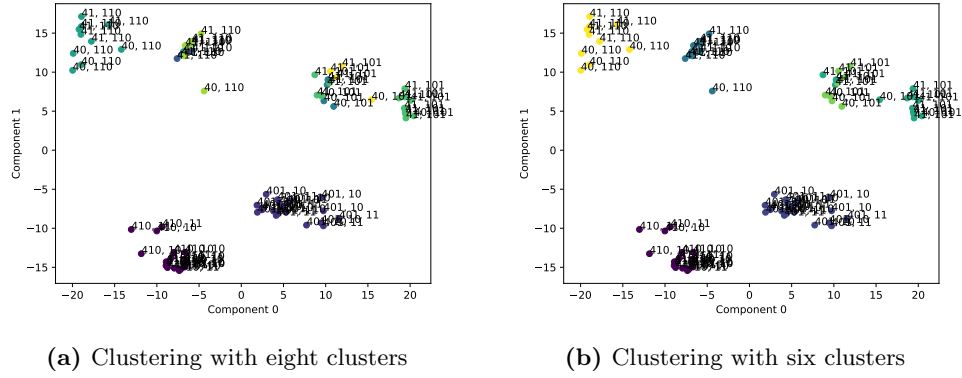


Figure 5.19: Visual representation of clusters formed with single noise emitting device in two device protocol.

5.3 Source Analysis

The different source combinations, discussed in section 4.1, are each tested with three protocols - each of these protocols test three devices. The three devices have different, unique functions: this is done to verify if it is still possible to make the same observations of load requirements in sections 5.1.1 and 5.1.2 which look for the unique and repeatable signatures also apply to the different sources.

The different source combinations, as well as the transient signatures of the different devices are illustrated in appendix A.7 where each source combination, and each protocol is given. The figures highlight the fact that, even though the systems utilize different technologies to support the grid, the operation of devices on the grid are still visible, unique, and repeatable.

There are caveats to this conclusion: similar to the cases discussed in section 5.1—which illustrated the cases for the bench top power supply—device noise still reduces the ability to observe the operation of devices on the grid, similar devices are still difficult to differentiate, and multiple low-power devices are more difficult to differentiate based on their low impact on the grid when changing states.

5.3.1 System Response Analysis

The systems analysed within this work, are made of two major components: the source, and the load. The systems, while operating, create a balance between the amount of power required by the load, and the power that the sources are

able to supply². This balance, achieved when the system is in steady-state, is the characteristic that enables the grid to function. The requirements of the load need to match up with the capabilities of the sources; however, the transient period—the time from when the system goes from a steady-state to when a device changes state and then back to the next new steady-state—is marked by the fluctuations of the grid voltage where the system attempts to re-attain the balance of the system. This transitory period, the magnitude of the deviations of the grid voltage are a function of the capabilities of the source and the magnitude of power that the load attempts to draw.

This relationship is illustrated in section 5.1.1, where a number of waveforms corresponding to different devices can be compared against one another. The effect observed in those waveforms is that the deviation of the grid voltage for devices with different power ratings are also unique: the voltage deviation of a high-power device is much greater as opposed to the smaller device.

This relationship can be defined as the voltage magnitude response on the grid and is related to the power capability of the sources; this characteristic observed here assists in the creation of a model that can be used to determine the limits of NILM and load disaggregation on generic grids. This seeks to create a generalized model that allows designers to determine the limitations of the system based on the sampling equipment. These parameters include: loads and their power ratings and the different sources and their ratings.

This results in a situation where, devices that have a smaller impact on the electrical network correspond to a smaller transient; this means that there is a limitation on the sampling rate requirements that enable the observation of these transients as they are small and ephemeral on the system. The design of the system here, with the utility of the downsampling system means the model can be used to evaluate the possibility of observing and distinguishing different devices on specified source systems with sampling rates as a parameter. Section 5.4 illustrates the relationship between the sampling rate and the ability to detect differently rated devices on the grid.

²The wiring utilized within this work has a very small resistance, as such the wires introduce negligible non-ideal effects into the system.

5.4 Sampling Analysis

Section 4.7 introduced the tools and methodology in which the original high-sampling rate event recordings are converted into multiply downsampled versions. The expectation of the system designed is: as the sampling rate lowers, the ability to detect different devices on the system will decrease to a point where the system can no longer reliably distinguish the difference between different devices and their switching events.

The reduction in sampling rate is also likely to have the effect that the lower power devices—which have a small effect on the network—will be the first devices that will no longer be distinguishable. This is largely due to the magnitude of the transients that they create on the network. This implies that there is a relationship between the power of the device, the magnitude and duration of the voltage deviation resulting on the grid, and the best ARI achievable at set sampling rates.

To model this effect, the different devices used within this research are placed into three major power ranges: low-, medium- and high-power each corresponding to $0 \leq 5\text{ W}$, $5 > 10\text{ W}$, and $\geq 10\text{ W}$ respectively. Naturally, these categories of devices are also related to the capability of the power supply used within the system because the response of the grid voltage is also dependant on the power supply. These three categories of devices are used to create a model that is able to explain the limitations on the minimum sampling rate that achieves sufficient load disaggregation ability.

Note here that the Fourier coefficient features are not utilized in this analysis, this is firstly due to the unpredictable benefit of these features (see section 5.4.2), additionally, the Fourier coefficients pose an interesting feature set.

The Fourier coefficients of events can be used to determine an upper limit on signatures of different devices, this is directly related to the Nyquist limit which states that past a certain sampling rate, no more information will be gained as the original signature can be sufficiently captured. Additionally, this means that different devices exhibit varying Fourier coefficients and magnitudes of those coefficients: some devices may exhibit Fourier coefficients which are much higher in the frequency spectrum, thus requiring a higher sampling rate to sufficiently capture the devices exact representation on the grid. This means that a device *can* be characterized by the upper limit on the highest frequency that it introduces into the grid voltage, thus enabling the system to be designed as it is then known what the upper limit of the sampling rate needs to be. Any sampling rate higher than this is not useful as it

does not add any more information (see the Nyquist limit theorem).

However, it is important to note that the FFT is not useful in determining the lower limit for the sampling rate as this is dependant on the features (not just the Fourier coefficient features) and how the features relate to the sampling rate as well as how well the features and scaling/pre-processing and clustering algorithms can perform. This suggests that even with sampling rates that are below the Nyquist limit for some devices within the system, sufficient information *can* be gained from other features to perform disaggregation which is the end goal here.

5.4.1 System Sampling Model

The model discussed in section 5.3.1 speaks about the fact that a relationship exists between the source, load, and sampling equipment that describes the ability or lack thereof to distinguish said loads on the network.

To define how the sampling rate affects the ability to observe different source and load configurations, the first step is to define how the load (the load power) and the source (the source maximum power capability) can be used to define the point at which NILM and load disaggregation techniques can no longer effectively observe and understand what the system is doing.

The two components: the source and the load can therefore be coupled here into ratios, this is defined in such a way that the load:source ratio is a parameter that can describe the sampling requirements.

The load:source ratio is defined in eq. (5.1)

$$\eta_{System} = \frac{P_{Load}}{P_{Source}} \quad (5.1)$$

Where P_{Load} is defined as the operating power that a load draws when connected to the grid and P_{Source} is defined as the maximum power that the source can supply to any load.

To illustrate this, the different loads that are utilized within this system have a wide range of power requirements, for example the low-powered light discussed in section 5.1.1 has an operating power of 5 W; this value is then coupled with the

Table 5.6: Device power ranges

Power range description	Power range (W)	Power range ratio
Low	0 to 5	0 to 0.0347
Medium	5 to 10	0.0347 to 0.069 44
High	> 10	0.069 444 to 1.0

capabilities of the power supply which has a maximum capability of 144 W. This creates a scenario where the load:source ratio η_{System} is 0.034 72.

Section 5.4 introduced the idea that the devices within this work are split into three major power requirements: low, medium, and high. Table 5.6 lists the different power ranges and the ratios that they encompass³.

These three groups of devices are then run through the full pipeline and clustered; each of the different downsampled versions are evaluated to generate an ARI which illustrates the maximum performance achievable at a given sampling rate with the device power range.

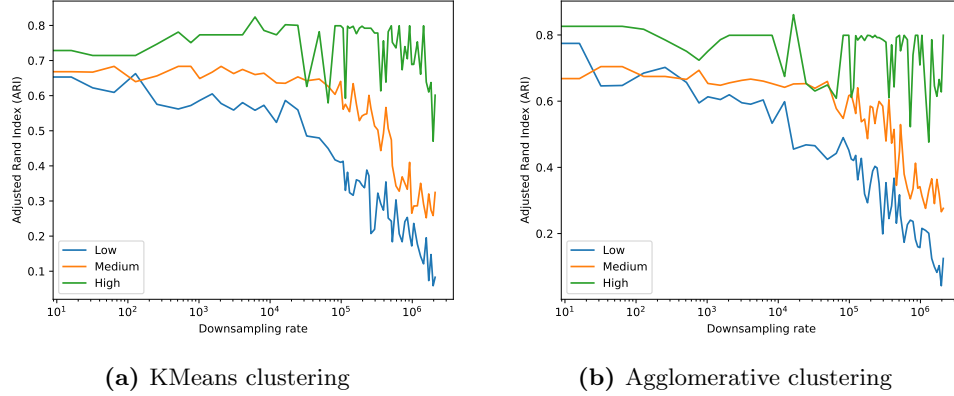


Figure 5.20: Clustering performance for three device power ranges with increasing downsampling rate

These figures illustrate the fact predicted in section 5.4 where the high-power devices, exhibiting large and long-lasting transients have features which are still unique and distinguishable with much lower sampling rates. The figures also highlight the fact that there is a limitation on the sampling rate which is dependant on the ratio of devices that need to be observed and disaggregated from one another.

³The devices within this research do not include a diverse range within the high-power section; this should be considered in further work to provide a better sampling of devices within this range.

The low-power devices have a steep drop off in performance, once the downsampling rate exceeds 16 384 which corresponds to a sampling rate of 1.3 kS/s.

This steep drop-off, however, does not coincide with an acceptable ARI, as the performance at this sampling rate is low. Sufficient performance is only realistically viable at downsampling rates below 1536 (ie. sampling rates above 13 kS/s).

This lower limit on the sampling rate defined here is dependant on the lower power devices, these devices are the limiting factor in the ability to perform disaggregation as their response on the network is small, and therefore the first group of devices that the system loses the ability to observe as the sampling rate drops.

The medium and higher-power devices can still be observed at sampling rates that are approximately an order of magnitude lower.

These results line up with the expected results introduced in section 5.4 where the high-power devices have large and long-lasting transients which are easier to observe and create signatures for when compared to the lower power devices.

This result suggested here does not represent a suitable sampling frequency for all possible use cases, however it does provide a guide on minimum sampling rates that can be used to observe specific devices operating within the grid.

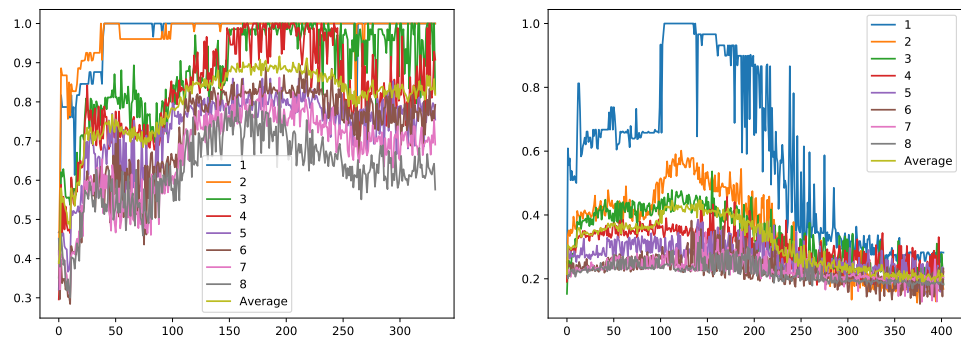
5.4.2 Feature Analysis

Figure 4.37 illustrates what appears to be a situation where the addition of specific features results in a decrease in the general ARI that can be achieved.

The placement of these features line up with the presence of the Fourier coefficient features. To illustrate how the clustering algorithms perform when using only the Fourier coefficients, and using all of the features excluding the Fourier coefficients, fig. 5.21 is created. This shows that, the presence of the Fourier coefficients do add a significant amount of information, however they also appear to reduce the performance of the clustering.

Future iterations of this should investigate the role of the Fourier coefficients in providing useful features; in some cases adding value, and in others reducing the performance.

The observation of the Fourier coefficients is that they often have large ANOVA



(a) Clustering with Fourier coefficients removed

(b) Clustering with only Fourier coefficients

Figure 5.21: Different types of feature scaling

F-values, which is beneficial however, it is often coupled with large p-values which reduces the utility of those features by confounding the clustering process with their addition. This is not consistent with all Fourier coefficients, highlighting the fact that extraction of specific coefficients should be investigated.

Figure 5.22 illustrates the ANOVA F-values of different features, it is clear that, there is a steep drop-off in the usefulness of features; this highlights the case that, approximately the first 200 to 300 features are the most useful.

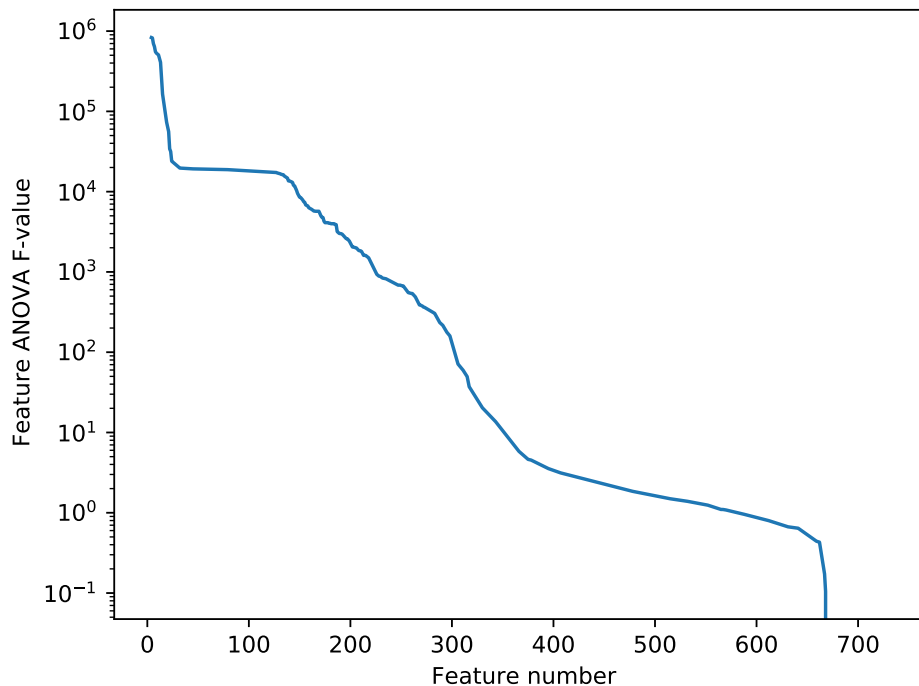


Figure 5.22: Feature performance rolloff ANOVA

Table 5.7: Ranking of best features from tsfresh [143]

Feature rank	Feature name
1	Quantile
2	Change quantiles
3	Autocorrelation
4	Agg autocorrelation
5	Skewness
6	Time series complexity
7	Agg linear trend
8	Ratio value number to time series length (num unique values / num values)
9	Minimum
10	RMS Voltage

Observing the distribution of ranks of features from their scores, there are a number of features that, overall, add the most value to clustering, these are listed in table 5.7. Essentially, the features listed here (not an exhaustive list of all features) details the best feature in terms of information gain [32, 142].

This analysis enables further designs to benefit from the fact that some features are more useful than others; future designs can make use of these conclusions to choose specific features and thus reduce the computation time and complexity to perform similar functionality.

5.4.3 Clustering Algorithm Comparison

Observing the different results in this chapter, it is clear that neither K-Means nor Agglomerative clustering perform significantly better than the other; they both appear to be sufficient tools to enable the analysis of the underlying structure of the data.

These two algorithms do have vastly different run-time performances. K-Means takes significantly longer to accomplish the same analysis as the Agglomerative clustering algorithm, suggesting that in future iterations, Agglomerative clustering is a more efficient tool to utilize.

Additionally, the introduction of feature normalization, the standard scaler in this case, greatly increased the ARI when clustering. This suggests that feature normalization

should always be part of the data processing pipeline. The presence of the normalizer also greatly reduces and enhances the Fourier coefficients, these features appear to massively skew the results if no normalization is applied.

Chapter 6

Conclusion & Future Recommendations

The research covered in this document presents a number of conclusions which appear to indicate that NILM and Load Disaggregation can in fact be undertaken on distributed DC grids. This chapter looks back at the problem statement and motivation as well as the research aims listed in chapter 1 to determine the validity of those assumptions. Over the course of experimentation and analysis, different aspects of the methodology and results were highlighted as areas where improvements could be made. These improvements include changes to the devices and sources utilized, the structure and functionality of the automated testing system, and potential changes to the data pipeline. A number of improvements also involve creating a system that can test the techniques of load disaggregation on distributed DC grids in normal installations, not just laboratory setups.

6.1 Conclusion

This research created a number of interesting outcomes, many of which directly (or indirectly) answered the research aims from chapter 1. What is clear from the material in chapter 5, it is possible to perform NILM and Load Disaggregation on LVDC distributed grids to observe the devices operating within. However, there are a number of limitations to this. To cover these, the research aims are replicated here to verify how exactly the research conducted is able to answer the questions.

On the technical level, it is shown that it is indeed possible to distinguish device

operation by measuring the grid voltage because different devices exhibit unique and repeatable transients onto the grid voltage.

Limitations appear where devices emit noise onto the grid, drowning out the transients of further device switching; the limitations on the sampling rates on such systems—to achieve sufficient disaggregation performance—should exceed 20 kS/s. This sampling limitation is based on the observations that low-power devices exhibit small transients on the grid voltage and higher-powered devices have larger and longer lasting transients.

6.1.1 Problem Statement and Motivation Outcomes

The major outcomes from the problem statement identified included four major items:

- The measurement parameter
- The measurement technique
- The NILM implementation and Load Disaggregation
- The distributed DC grid

To provide basic comments on each of these items, it is clear that they all interact and need to provide a sufficient level of utility for the research outcomes to be relevant and useful.

The measurement parameter: The work conducted in this research highlighted the fact that the grid voltage is often the best indicator of devices operating on the network. The current, depending on how it is measured is somewhat useful in adding information; however, it is clear that the voltage is the best parameter to be used for the requirements of NILM and Load Disaggregation as an end goal.

The measurement technique: The combination of using the grid voltage as a parameter while measuring in specific times created the ability to observe the response of the grid with the device. The technique utilised in this research is capable to observe the outcomes needed, and it is sufficient for most use cases. As mentioned in chapter 5 and section 6.1.6, there is the potential to enhance the measurement technique to extract more information out of the system.

The NILM implementation and Load Disaggregation: The methodology utilised in this work is particularly constrained as it is designed to seek to answer the research questions (which it is well suited for), and in this respect the design works as required. A real-world application would require a different application with a different design, however the design of such a system would be able to draw its design elements from the outcomes of this research directly.

The distributed DC grid: The aims of a distributed DC grid for this research required the design and construction of a range of types of DC grids, all seeking to answer specific questions; from this basis, the grids used are sufficient to answer the research questions with their wide technology and operating characteristics.

6.1.2 Research Question Conclusions

The major research question outlined in section 1.2, is the following:

To what extent can the techniques of NILM be used to reliably and accurately identify the operation of devices within a distributed DC system with a final aim of load disaggregation within the system purely making use of the DC grid voltage of the system?

Based on the outcomes in chapter 5, it is clear that the techniques in the NILM field can indeed be used to reliably and accurately identify the operation of devices within the distributed DC grids. There are a few limitations to the performance of such systems which are based on a number of factors: device power, device/source noise, sampling rate, number of devices operating, to name a few. These limitations are covered in depth in chapter 5, however it is pertinent to note here that the overall aim of the research, as mentioned in the question above is indeed answered.

This main research aim, broken down into three separate questions is answered throughout the document where the methodology is analysed in depth in chapter 5.

6.1.3 Future Recommendations

This section outlines the different aspects identified over the course of the research that are potential avenues for further research. Some of the items in this section are not particularly relevant to the field of NILM and Load Disaggregation; rather

they include a number of interesting observations and outcomes that can be explored further.

6.1.4 Physical System Setup

The methodology in this research is limited by the devices available, the measurement device capabilities, and the equipment that enables the automated switching system; these three systems can be improved in small ways to provide a greater understanding in how well NILM and load disaggregation can be applied to distributed DC systems.

6.1.4.1 Device Types

This work only made use of type I and II devices; this was a function of the devices available as well as the ability afforded by the automated switching system. Device types III and IV use a wider range of switching components and operational characteristics; different techniques would need to be utilized to determine if and how well these devices are detectable (and distinguishable from one another) on DC grids. The event-based approach is unlikely to perform well for these types of devices as they do not always have clearly defined switching transients over short periods of time, rather they likely exhibit other electrical characteristics over time. One potentially useful technique is to monitor the grid voltage noise - during the course of this work, it is clear that some devices exhibit noise while operating, in addition to different noise characteristics, potentially making them distinguishable from one another. Fortunately, the field of NILM and Load Disaggregation is growing and many of the techniques in the field may be applicable and usable for detecting these other device types.

6.1.4.2 Measurement Device

The measurement device utilized in this research has a maximum sampling rate of 20 MS/s, this creates a hard limit on the highest observable frequencies (based on the Nyquist criterion) that the system can observe. It is unknown if higher sampling rates would increase the disaggregation ability, however it should be noted that this sampling rate already involves a large cost for system monitoring.

A measurement device with a higher sampling rate and higher vertical resolution may be able to provide better disaggregation ability but with a trade-off of an increased

system cost. Ideally, a high sampling rate would be coupled with a high vertical resolution, it is obvious that this would negatively affect the price, however it will also introduce further complexity into the entire system as the data requirements would also increase - increased data rates required, storage space required, computational ability required, etc.

This research did not focus on how changes to the vertical resolution could provide benefit to the overall disaggregation ability, however it should be noted that a trade-off between sampling rate and vertical resolution are both important to determine an optimal measurement device. Concretely, as mentioned above, a higher sampling rate and resolution will both increase the performance of the system, however there is likely to be a point at which the incremental cost does not provide an equally incremental performance to the system as illustrated in fig. 3.5.

6.1.4.3 System Construction

Due to the construction of the automated switching system—relays available from the Logo device—there are limitations on the scalability and flexibility of the testing jig.

The first limitation of the system utilized in this work is that the Logo's utilize relays, meaning that the switching events consequently include the non-ideal effects of relays making contact as switching devices. These relays may also alter the switching action, and hence transients, of devices switching as the switching device is different from the standard installation of devices onto DC grids.

Future work should make use of event detectors (see sections 2.6.2 and 6.1.4.5) that are capable of detecting normal system usage where the automated switching system is then not required [144].

6.1.4.4 Device Noise

Section 5.2.1 introduced a unique problem that was evident when devices insert noise onto the grid voltage. The theorized ground truth was augmented depending on those devices. Further effort can be placed on the role that device noise plays on the grid; particular attention should be placed on the relationship between the noise introduced and the ability to perform disaggregation. This is particularly relevant

in cases when some noisy devices may mask the operation of other devices on the system.

A method may be utilized to counteract this issue whereby the system characterises the appliances that generate noise. Once the noise is characterised, it can be removed (and/or stored as additional information) with DSP techniques before the load disaggregation process. This can also be done with a Low Pass Filter, however the implementation of such a device would reduce some of the information within the signal that may be used in the disaggregation process. This can be used to alleviate the effects of those noisy appliances when the system needs to detect the operation of other devices when the noisy device is operating.

6.1.4.5 Event Detector

Other research in the field of NILM utilize event detectors, devices that are able to detect when devices switch, and therefore create systems that can start and stop measuring the parameters at the exact times. These devices may be beneficial in that they can precisely record moments in time when devices are switching and when those transients finish.

These event detectors, in this case represent a system that can add value to the features as the measurement event length (the amount of time from when an event starts to when the system returns back to steady-state operation) can be used as an additional feature.

The implementation of an event detector also increases the utility of NILM to systems, the event detectors are paramount to event-based approaches because in the real world, the system does not make use of automated switching systems.

Figure 6.1 illustrates the current situation, where the entire event window is recorded (this long event window accounts for the devices that have the longest lasting transients), which is compared against two events (a long and a short lasting transient) where it is clear that it makes sense to only keep select samples from the event window because it is only in those periods of time that distinguishing features are observed. Other benefits of this event detector are that data storage (and computation requirements) are reduced and optimized.

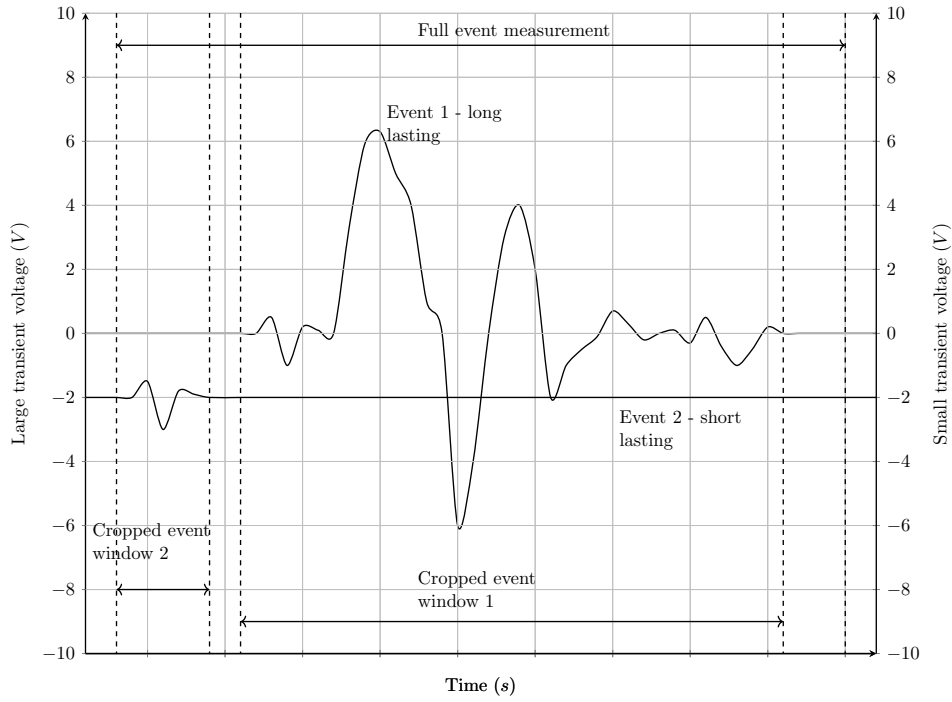


Figure 6.1: Benefits that an event detector has on measuring the correct samples in time.

6.1.5 NILM and Load Disaggregation Techniques

Research in this field makes use of a wide variety of techniques to accomplish the NILM and load disaggregation task, this involves making use of other *meta-features* which can enhance the disaggregation performance, these features are often not a direct function of the system operation but instead correlated, examples of how this can be beneficial are discussed in section 6.1.7.

Other techniques used in the NILM space may be readily applied to these distributed DC systems that can enhance and enable greater visibility of system operation.

6.1.6 Current Usefulness

In the event that the measurement of current is assistive in observing operating of devices, it may be that future work could incorporate the features of the current with the features of the voltage to further assist the disaggregation.

6.1.6.1 Energy Analysis

This research focused on verifying whether the grid voltage could be utilized to determine and distinguish devices' operation on the grid; other research in the field of NILM and load disaggregation often involves making further inferences on the power usage of these devices when operating. The current research in the field, based on the idea that it uses centralized measurement techniques (i.e. measuring current and voltage of the main incomer) makes it much easier to infer the power consumption of individual devices over time.

Further work on finding out if it is possible to infer the power usage of devices on the grid based on the characteristics of the transient events; for example, is it possible to use the voltage recovery rate to infer energy consumption? This is based on the understanding that the voltage recovery rate is a function of the power supply capabilities, the magnitude and makeup of the load (resistance, inductance, capacitance) and any other non-ideal effects present within the system.

6.1.7 Meta Information Benefits

Research in this field that observes the operation of devices in real world scenarios—for example systems that monitor homes over the course of a few weeks—often have access to a wider range of useful information available. This extra information, which includes data such as temperature, time of day, day in week, day in month, means that the load disaggregation systems have access to a wider range of features. These extra features, called *meta-features* enable greater understanding of the system; for example a device that turns on later in the day is likely to be a light, so situations where the signatures of two devices look very similar (maybe the other device is a fan), the system can then use the time of use to rule out the other device as it is more likely that a light is being turned on later in the day.

The addition of these features can enhance the performance of disaggregation and this can be looked into with future research to see if any benefit can be found.

This extra information, combined with state-of-the-art techniques in the field, which utilize more efficient and complex machine learning techniques (see supervised machine learning techniques) may increase the ability to perform NILM and load disaggregation.

References

- [1] G. Hart, “Nonintrusive appliance load monitoring,” en, *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, Dec. 1992, ISSN: 00189219. DOI: 10.1109/5.192069. [Online]. Available: <http://ieeexplore.ieee.org/document/192069/> (visited on 23/05/2019).
- [2] Junjun Zhang, Hongfei Wu, Feng Cao, Yan Xing and Xudong Ma, “Analysis and design of DC distributed DC power system with modular three-port converter,” en, in *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, Istanbul, Turkey: IEEE, Jun. 2014, pp. 416–421, ISBN: 978-1-4799-2399-1. DOI: 10.1109/ISIE.2014.6864649. [Online]. Available: <http://ieeexplore.ieee.org/document/6864649/> (visited on 05/07/2019).
- [3] J. J. Justo, F. Mwasilu, J. Lee and J.-W. Jung, “AC-microgrids versus DC-microgrids with distributed energy resources: A review,” en, *Renewable and Sustainable Energy Reviews*, vol. 24, pp. 387–405, Aug. 2013, ISSN: 13640321. DOI: 10.1016/j.rser.2013.03.067. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1364032113002268> (visited on 05/07/2019).
- [4] H. Khadilkar, V. Chandan, S. Kalra, S. Ghai, Z. Charbiwala, T. Ganu, R. Kunnath, L. C. Ming and D. P. Seetharam, “DC Picogrids as power backups for office buildings,” en, in *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Venice, Italy: IEEE, Nov. 2014, pp. 157–162, ISBN: 978-1-4799-4934-2. DOI: 10.1109/SmartGridComm.2014.7007639. [Online]. Available: <http://ieeexplore.ieee.org/document/7007639/> (visited on 20/02/2019).
- [5] S. Kurtz, J. Granata and M. Quintana, “Photovoltaic-reliability R&D toward a solar-powered world,” en, in *SPIE Solar Energy + Technology*, N. G. Dhere, J. H. Wohlgemuth and D. T. Ton, Eds., San Diego, CA, Aug. 2009, 74120Z. DOI: 10.1117/12.825649. [Online]. Available: <http://proceedings.spiedigitallibrary.org/>

- spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.825649 (visited on 18/02/2019).
- [6] A. H. Mondal and M. Denich, “Hybrid systems for decentralized power generation in Bangladesh,” en, *Energy for Sustainable Development*, vol. 14, no. 1, pp. 48–55, Mar. 2010, ISSN: 09730826. DOI: 10.1016/j.esd.2010.01.001. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0973082610000025> (visited on 25/02/2019).
- [7] Renewable Energy Policy Network for the 21st Century, “Renewables 2016 - Global Status Report,” en, *Renewable Energy Policy Network for the 21st Century*, p. 272, 2016. [Online]. Available: http://www.ren21.net/wp-content/uploads/2016/06/GSR_2016_Full_Report_REN21.pdf.
- [8] C. Seligman and J. M. Darley, “Feedback as a Means of Decreasing Residential Energy Consumption,” en, p. 6,
- [9] C. Seligman, J. M. Darley and L. J. Becker, “Behavioral approaches to residential energy conservation,” en, *Energy and Buildings*, vol. 1, no. 3, pp. 325–337, Apr. 1978, ISSN: 03787788. DOI: 10.1016/0378-7788(78)90012-9. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/0378778878900129> (visited on 19/06/2020).
- [10] J. M. Darley, “Energy conservation techniques as innovations, and their diffusion,” en, *Energy and Buildings*, vol. 1, no. 3, pp. 339–343, Apr. 1978, ISSN: 03787788. DOI: 10.1016/0378-7788(78)90013-0. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/0378778878900130> (visited on 19/06/2020).
- [11] J.-J. González-de-la-Rosa, A. Agüera-Pérez, J.-C. Palomares-Salas, O. Florencias-Oliveros and J.-M. Sierra-Fernández, “A Dual Monitoring Technique to Detect Power Quality Transients Based on the Fourth-Order Spectrogram,” en, *Energies*, vol. 11, no. 3, p. 503, Feb. 2018, ISSN: 1996-1073. DOI: 10.3390/en11030503. [Online]. Available: <http://www.mdpi.com/1996-1073/11/3/503> (visited on 29/05/2020).
- [12] K. Ehrhardt-Martinez and K. A. Donnelly, “Advanced Metering Initiatives and Residential Feedback Programs: A Meta-Review for Household Electricity-Saving Opportunities,” en, *American Council for an Energy-Efficient Economy*, p. 140, Jun. 2010.
- [13] S. S. Hosseini, K. Agbossou, S. Kelouwani and A. Cardenas, “Non-intrusive load monitoring through home energy management systems: A comprehensive review,” en, *Renewable and Sustainable Energy Reviews*, vol. 79, pp. 1266–1274, Nov. 2017, ISSN: 13640321. DOI: 10.1016/j.rser.2017.05.096.

- [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1364032117307359> (visited on 22/05/2019).
- [14] A. Zoha, A. Gluhak, M. Imran and S. Rajasegarar, “Non-Intrusive Load Monitoring Approaches for Disaggregated Energy Sensing: A Survey,” en, *Sensors*, vol. 12, no. 12, pp. 16 838–16 866, Dec. 2012, ISSN: 1424-8220. DOI: 10.3390/s121216838. [Online]. Available: <http://www.mdpi.com/1424-8220/12/12/16838> (visited on 22/05/2019).
- [15] A. Ridi, C. Gisler and J. Hennebert, “A Survey on Intrusive Load Monitoring for Appliance Recognition,” en, in *2014 22nd International Conference on Pattern Recognition*, Stockholm, Sweden: IEEE, Aug. 2014, pp. 3702–3707, ISBN: 978-1-4799-5209-0. DOI: 10.1109/ICPR.2014.636. [Online]. Available: <http://ieeexplore.ieee.org/document/6977348/> (visited on 22/05/2019).
- [16] I. Abubakar, S. Khalid, M. Mustafa, H. Shareef and M. Mustapha, “Application of load monitoring in appliances’ energy management – A review,” en, *Renewable and Sustainable Energy Reviews*, vol. 67, pp. 235–245, Jan. 2017, ISSN: 13640321. DOI: 10.1016/j.rser.2016.09.064. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S136403211630555X> (visited on 22/05/2019).
- [17] A. Ridi and J. Hennebert, “Hidden Markov Models for ILM Appliance Identification,” en, *Procedia Computer Science*, vol. 32, pp. 1010–1015, 2014, ISSN: 18770509. DOI: 10.1016/j.procs.2014.05.526. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1877050914007261> (visited on 22/05/2019).
- [18] S. J. Kang, K. Shin, R. Yang, S. Y. Lee and J. Sohn, “Cost effective disaggregation mechanism for the NIALM system,” en, *International Journal of Smart Grid and Clean Energy*, pp. 364–370, 2013, ISSN: 23154462. DOI: 10.12720/sgce.2.3.364-370. [Online]. Available: <http://www.ijsgce.com/index.php?m=content&c=index&a=show&catid=41&id=128> (visited on 23/05/2019).
- [19] I. Abubakar, S. N. Khalid, M. W. Mustafa, H. Shareef and M. Mustapha, “An overview of Non-intrusive load monitoring methodologies,” en, in *2015 IEEE Conference on Energy Conversion (CENCON)*, Johor Bahru: IEEE, Oct. 2015, pp. 54–59, ISBN: 978-1-4799-8598-2. DOI: 10.1109/CENCON.2015.7409513. [Online]. Available: <http://ieeexplore.ieee.org/document/7409513/> (visited on 22/06/2020).

- [20] A. Cole and A. Albicki, “Nonintrusive identification of electrical loads in a three-phase environment based on harmonic content,” en, in *Proceedings of the 17th IEEE Instrumentation and Measurement Technology Conference [Cat. No. 00CH37066]*, Baltimore, MD, USA: IEEE, 2000, pp. 24–29, ISBN: 978-0-7803-5890-4. DOI: 10.1109/IMTC.2000.846806. [Online]. Available: <http://ieeexplore.ieee.org/document/846806/> (visited on 24/06/2020).
- [21] S. Shaw, S. Leeb, L. Norford and R. Cox, “Nonintrusive Load Monitoring and Diagnostics in Power Systems,” en, *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 7, pp. 1445–1454, Jul. 2008, ISSN: 0018-9456, 1557-9662. DOI: 10.1109/TIM.2008.917179. [Online]. Available: <http://ieeexplore.ieee.org/document/4463663/> (visited on 29/05/2020).
- [22] Y. T. Quek, W. L. Woo and T. Logenthiran, “Anomaly Warning and Fault Detection in DC Pico-grid with enhanced k-Nearest Neighbours Technique,” en, in *2018 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, Singapore: IEEE, May 2018, pp. 728–733, ISBN: 978-1-5386-4291-7. DOI: 10.1109/ISGT-Asia.2018.8467961. [Online]. Available: <https://ieeexplore.ieee.org/document/8467961/> (visited on 05/07/2019).
- [23] —, “Smart Sensing of Loads in an Extra Low Voltage DC Pico-Grid Using Machine Learning Techniques,” en, *IEEE Sensors Journal*, vol. 17, no. 23, pp. 7775–7783, Dec. 2017, ISSN: 1530-437X, 1558-1748, 2379-9153. DOI: 10.1109/JSEN.2017.2723925. [Online]. Available: <http://ieeexplore.ieee.org/document/7970113/> (visited on 05/07/2019).
- [24] D. Salomonsson and A. Sannino, “Load modelling for steady-state and transient analysis of low-voltage DC systems,” en, *IET Electric Power Applications*, vol. 1, no. 5, p. 690, 2007, ISSN: 17518660. DOI: 10.1049/iet-epa:20060418. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/iet-epa_20060418 (visited on 29/05/2020).
- [25] A. Aboulhian, D. H. Green, J. F. Switzer, T. J. Kane, G. V. Bredariol, P. Lindahl, J. S. Donnal and S. B. Leeb, “NILM Dashboard: A Power System Monitor for Electromechanical Equipment Diagnostics,” en, *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1405–1414, Mar. 2019, ISSN: 1551-3203, 1941-0050. DOI: 10.1109/TII.2018.2843770. [Online]. Available: <https://ieeexplore.ieee.org/document/8371632/> (visited on 29/05/2020).
- [26] R. Cox, S. Leeb, S. Shaw and L. Norford, “Transient Event Detection for Nonintrusive Load Monitoring and Demand-Side Management Using Voltage Distortion,” en, in *Twenty-First Annual IEEE Applied Power Electronics*

- Conference and Exposition, 2006. APEC '06.*, USA: IEEE, 2006, pp. 1751–1757, ISBN: 978-0-7803-9547-3. DOI: 10.1109/APEC.2006.1620777. [Online]. Available: <http://ieeexplore.ieee.org/document/1620777/> (visited on 29/05/2020).
- [27] K. L. Lian, K. S. Tung and Y. C. Su, “A non-intrusive load monitoring system based on a cascaded method,” en, in *2013 3rd International Conference on Electric Power and Energy Conversion Systems*, Istanbul, Turkey: IEEE, Oct. 2013, pp. 1–6, ISBN: 978-1-4799-0688-8 978-1-4799-0687-1. DOI: 10.1109/EPECS.2013.6713027. [Online]. Available: <http://ieeexplore.ieee.org/document/6713027/> (visited on 22/06/2020).
- [28] L. K. Norford, “Non-intrusive electrical load monitoring in commercial buildings based on steady-state and transient load-detection algorithms,” en, p. 14,
- [29] Kwangduk Douglas Lee, *Electric load information system based on non-intrusive power monitoring*, English, 2003. [Online]. Available: <http://hdl.handle.net/1721.1/29633>.
- [30] A. Faustine, N. H. Mvungi, S. Kaijage and K. Michael, “A Survey on Non-Intrusive Load Monitoring Methodies and Techniques for Energy Disaggregation Problem,” en, *arXiv:1703.00785 [cs]*, Mar. 2017. arXiv: 1703.00785 [cs]. [Online]. Available: <http://arxiv.org/abs/1703.00785> (visited on 23/05/2019).
- [31] J. Kelly and W. Knottenbelt, “Neural NILM: Deep Neural Networks Applied to Energy Disaggregation,” en, *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments - BuildSys '15*, pp. 55–64, 2015, Comment: To appear in ACM BuildSys'15, November 4–5, 2015, Seoul. DOI: 10.1145/2821650.2821672. arXiv: 1507.06594. [Online]. Available: <http://arxiv.org/abs/1507.06594> (visited on 29/05/2020).
- [32] P. A. Schirmer and I. Mporas, “Statistical and Electrical Features Evaluation for Electrical Appliances Energy Disaggregation,” en, *Sustainability*, vol. 11, no. 11, p. 3222, Jun. 2019, ISSN: 2071-1050. DOI: 10.3390/su11113222. [Online]. Available: <https://www.mdpi.com/2071-1050/11/11/3222> (visited on 29/05/2020).
- [33] Z. Wang and T. Oates, “Encoding Time Series as Images for Visual Inspection and Classification Using Tiled Convolutional Neural Networks,” en, p. 7,
- [34] Y. Zheng, Q. Liu, E. Chen, Y. Ge and J. L. Zhao, “Exploiting multi-channels deep convolutional neural networks for multivariate time series classification,” en, *Frontiers of Computer Science*, vol. 10, no. 1, pp. 96–112, Feb. 2016,

- ISSN: 2095-2228, 2095-2236. DOI: 10.1007/s11704-015-4478-2. [Online]. Available: <http://link.springer.com/10.1007/s11704-015-4478-2> (visited on 02/06/2020).
- [35] Pedro Paulo Marques do Nascimento, “Applications of Deep Learning techniques on NILM,” pt, p. 85, Apr. 2016.
- [36] P. Davies, J. Dennis, J. Hansom, W. Martin, A. Stankevicius and L. Ward, “Deep Neural Networks for Appliance Transient Classification,” en, in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom: IEEE, May 2019, pp. 8320–8324, ISBN: 978-1-4799-8131-1. DOI: 10.1109/ICASSP.2019.8682658. [Online]. Available: <https://ieeexplore.ieee.org/document/8682658/> (visited on 29/05/2020).
- [37] J. Kelly, “Disaggregation of Domestic Smart Meter Energy Data,” en, p. 251,
- [38] M. C. Lee, “A M:ultiprocessorfor Transient Event Detection,” en, p. 10,
- [39] S. Gupta, M. S. Reynolds and S. N. Patel, “ElectriSense: Single-point sensing using EMI for electrical event detection and classification in the home,” en, in *Proceedings of the 12th ACM International Conference on Ubiquitous Computing - Ubicomp '10*, Copenhagen, Denmark: ACM Press, 2010, p. 139, ISBN: 978-1-60558-843-8. DOI: 10.1145/1864349.1864375. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1864349.1864375> (visited on 12/07/2019).
- [40] H. Liu, *Non-Intrusive Load Monitoring: Theory, Technologies and Applications*, en. Singapore: Springer Singapore, 2020, ISBN: 9789811518591 9789811518607. DOI: 10.1007/978-981-15-1860-7. [Online]. Available: <http://link.springer.com/10.1007/978-981-15-1860-7> (visited on 29/05/2020).
- [41] S. Shaw and S. Leeb, “Identification of induction motor parameters from transient stator current measurements,” en, *IEEE Transactions on Industrial Electronics*, vol. 46, no. 1, pp. 139–149, Feb. 1999, ISSN: 02780046. DOI: 10.1109/41.744405. [Online]. Available: <http://ieeexplore.ieee.org/document/744405/> (visited on 29/05/2020).
- [42] J. A. Lozano-Angulo, “Detection and One Class Classification of Transient Events in Train Track Noise,” en, p. 139,
- [43] David F. Peelo, F. Rahmatian, Mr.Nakul Nagpal and Daniel Sydor, *Real-time Monitoring and Capture of Power System Transients*, 2012. [Online]. Available: https://e-cigre.org/publication/B3-101_2012-real-time-monitoring-and-capture-of-power-system-transients.

- [44] T. Bernard, “Non-Intrusive Load Monitoring (NILM): Combining multiple distinct Electrical Features and Unsupervised Machine Learning Techniques,” en, p. 196,
- [45] J. Revuelta Herrero, Á. Lozano Murciego, A. López Barriuso, D. Hernández de la Iglesia, G. Villarrubia González, J. M. Corchado Rodríguez and R. Carreira, “Non Intrusive Load Monitoring (NILM): A State of the Art,” en, in *Trends in Cyber-Physical Multi-Agent Systems. The PAAMS Collection - 15th International Conference, PAAMS 2017*, F. De la Prieta, Z. Vale, L. Antunes, T. Pinto, A. T. Campbell, V. Julián, A. J. Neves and M. N. Moreno, Eds., vol. 619, Cham: Springer International Publishing, 2018, pp. 125–138, ISBN: 978-3-319-61577-6 978-3-319-61578-3. DOI: 10.1007/978-3-319-61578-3_12. [Online]. Available: http://link.springer.com/10.1007/978-3-319-61578-3_12 (visited on 27/05/2019).
- [46] O. Parson, “Unsupervised Training Methods for Non-intrusive Appliance Load Monitoring from Smart Meter Data,” en, p. 116,
- [47] B. Najafi, S. Moaveninejad and F. Rinaldi, “Data Analytics for Energy Disaggregation: Methods and Applications,” en, in *Big Data Application in Power Systems*, Elsevier, 2018, pp. 377–408, ISBN: 978-0-12-811968-6. DOI: 10.1016/B978-0-12-811968-6.00017-6. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/B9780128119686000176> (visited on 18/11/2019).
- [48] K. Suzuki, S. Inagaki, T. Suzuki, H. Nakamura and K. Ito, “Nonintrusive Appliance Load Monitoring Based on Integer Programming,” en, p. 6,
- [49] J. Liang, S. K. K. Ng, G. Kendall and J. W. M. Cheng, “Load Signature Study - Part I: Basic Concept, Structure, and Methodology,” en, *IEEE Transactions on Power Delivery*, vol. 25, no. 2, pp. 551–560, Apr. 2010, ISSN: 0885-8977, 1937-4208. DOI: 10.1109/TPWRD.2009.2033799. [Online]. Available: <http://ieeexplore.ieee.org/document/5337912/> (visited on 07/06/2019).
- [50] M. Zeifman and K. Roth, “Nonintrusive appliance load monitoring: Review and outlook,” en, *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 76–84, Feb. 2011, ISSN: 0098-3063. DOI: 10.1109/TCE.2011.5735484. [Online]. Available: <http://ieeexplore.ieee.org/document/5735484/> (visited on 29/05/2020).
- [51] Gu-yuan Lin, Shih-chiang Lee, Jane Yung-jen Hsu and Wan-rong Jih, “Applying power meters for appliance recognition on the electric panel,” en, in *2010 5th IEEE Conference on Industrial Electronics and Applications*, Taichung, Taiwan: IEEE, Jun. 2010, pp. 2254–2259, ISBN: 978-1-4244-5045-9. DOI: 10.

- 1109/ICIEA.2010.5515385. [Online]. Available: <http://ieeexplore.ieee.org/document/5515385/> (visited on 24/06/2020).
- [52] S. Makonin and F. Popowich, “Nonintrusive load monitoring (NILM) performance evaluation: A unified approach for accuracy reporting,” en, *Energy Efficiency*, vol. 8, no. 4, pp. 809–814, Jul. 2015, ISSN: 1570-646X, 1570-6478. DOI: 10.1007/s12053-014-9306-2. [Online]. Available: <http://link.springer.com/10.1007/s12053-014-9306-2> (visited on 13/07/2019).
- [53] E. T. Mayhorn, G. P. Sullivan, J. Petersen, R. S. Butner and E. M. Johnson, “Load Disaggregation Technologies: Real World and Laboratory Performance,” en, p. 13,
- [54] F. Liebgott and B. Yang, “Active learning with cross-dataset validation in event-based non-intrusive load monitoring,” en, in *2017 25th European Signal Processing Conference (EUSIPCO)*, Kos, Greece: IEEE, Aug. 2017, pp. 296–300, ISBN: 978-0-9928626-7-1. DOI: 10.23919/EUSIPCO.2017.8081216. [Online]. Available: <http://ieeexplore.ieee.org/document/8081216/> (visited on 29/05/2020).
- [55] E. T. Mayhorn, G. P. Sullivan, R. S. Butner, H. Hao and M. C. Baechler, “Characteristics and Performance of Existing Load Disaggregation Technologies,” en, Tech. Rep. PNNL–24230, 1203912, Apr. 2015. DOI: 10.2172/1203912. [Online]. Available: <http://www.osti.gov/servlets/purl/1203912/> (visited on 29/05/2020).
- [56] L. Pereira and N. Nunes, “Performance evaluation in non-intrusive load monitoring: Datasets, metrics, and tools—A review,” en, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 6, Nov. 2018, ISSN: 1942-4787, 1942-4795. DOI: 10.1002/widm.1265. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1265> (visited on 29/05/2020).
- [57] R. Bonfigli, S. Squartini, M. Fagiani and F. Piazza, “Unsupervised algorithms for non-intrusive load monitoring: An up-to-date overview,” en, in *2015 IEEE 15th International Conference on Environment and Electrical Engineering (EEEIC)*, Rome, Italy: IEEE, Jun. 2015, pp. 1175–1180, ISBN: 978-1-4799-7993-6. DOI: 10.1109/EEEIC.2015.7165334. [Online]. Available: <http://ieeexplore.ieee.org/document/7165334/> (visited on 29/05/2020).
- [58] L. De Baets, C. Develder, T. Dhaene, D. Deschrijver, J. Gao and M. Berges, “Handling imbalance in an extended PLAID,” en, in *2017 Sustainable Internet and ICT for Sustainability (SustainIT)*, Funchal: IEEE, Dec. 2017, pp. 1–5, ISBN: 978-3-901882-99-9. DOI: 10.23919/SustainIT.2017.8379795. [Online].

- Available: <https://ieeexplore.ieee.org/document/8379795/> (visited on 29/05/2020).
- [59] N. F. Esa, M. P. Abdullah and M. Y. Hassan, “A review disaggregation method in Non-intrusive Appliance Load Monitoring,” en, *Renewable and Sustainable Energy Reviews*, vol. 66, pp. 163–173, Dec. 2016, ISSN: 13640321. DOI: 10.1016/j.rser.2016.07.009. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1364032116303392> (visited on 16/06/2020).
 - [60] J. Gao, S. Giri, E. C. Kara and M. Bergés, “PLAID: A public dataset of high-resolution electrical appliance measurements for load identification research: Demo abstract,” en, in *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings - BuildSys '14*, Memphis, Tennessee: ACM Press, 2014, pp. 198–199, ISBN: 978-1-4503-3144-9. DOI: 10.1145/2674061.2675032. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2674061.2675032> (visited on 29/05/2020).
 - [61] M. N. Meziane, T. Picon, P. Ravier, G. Lamarque, J.-C. Le Bunetel and Y. Raingeaud, “A measurement system for creating datasets of on/off-controlled electrical loads,” en, in *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*, Florence, Italy: IEEE, Jun. 2016, pp. 1–5, ISBN: 978-1-5090-2320-2. DOI: 10.1109/EEEIC.2016.7555847. [Online]. Available: <http://ieeexplore.ieee.org/document/7555847/> (visited on 29/05/2020).
 - [62] T. Picon, M. N. Meziane, P. Ravier, G. Lamarque, C. Novello, J.-C. L. Bunetel and Y. Raingeaud, “COOLL: Controlled On/Off Loads Library, a Public Dataset of High-Sampled Electrical Signals for Appliance Identification,” en, *arXiv:1611.05803 [cs]*, Nov. 2016, Comment: 5 pages, 2 figures, 3 tables. arXiv: 1611.05803 [cs]. [Online]. Available: <http://arxiv.org/abs/1611.05803> (visited on 29/05/2020).
 - [63] D. Renaux, R. Linhares, F. Pottker, A. Lazzaretti, C. Lima, A. Coelho Neto and M. Campaner, “Designing a Novel Dataset for Non-intrusive Load Monitoring,” en, in *2018 VIII Brazilian Symposium on Computing Systems Engineering (SBESC)*, Salvador, Brazil: IEEE, Nov. 2018, pp. 243–249, ISBN: 978-1-72810-240-5. DOI: 10.1109/SBESC.2018.00045. [Online]. Available: <https://ieeexplore.ieee.org/document/8691975/> (visited on 29/05/2020).
 - [64] A. S. Uttama Nambi, A. Reyes Lua and V. R. Prasad, “LocED: Location-aware Energy Disaggregation Framework,” en, in *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments - BuildSys '15*, Seoul, South Korea: ACM Press, 2015, pp. 45–54,

- ISBN: 978-1-4503-3981-0. DOI: 10.1145/2821650.2821659. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2821650.2821659> (visited on 29/05/2020).
- [65] M. Kahl, A. U. Haq, T. Kriechbaumer and H.-A. Jacobsen, “WHITED - A Worldwide Household and Industry Transient Energy Data Set,” en, p. 5,
- [66] NILM Community, *Datasets - NILM wiki*, Jun. 2020. [Online]. Available: wiki.nilml.eu/datasets.html (visited on 25/06/2020).
- [67] International Energy Agency, *Gadgets and Gigawatts: Policies for Energy Efficient Electronics*. 2009, ISBN: 978-92-64-05953-5.
- [68] J. Schonbergerschonberger, R. Duke and S. Round, “DC-Bus Signaling: A Distributed Control Strategy for a Hybrid Renewable Nanogrid,” en, *IEEE Transactions on Industrial Electronics*, vol. 53, no. 5, pp. 1453–1460, Oct. 2006, ISSN: 0278-0046. DOI: 10.1109/TIE.2006.882012. [Online]. Available: <http://ieeexplore.ieee.org/document/1705636/> (visited on 05/07/2019).
- [69] J. Bryan, R. Duke and S. Round, “Decentralized generator scheduling in a nanogrid using DC bus signaling,” en, in *IEEE Power Engineering Society General Meeting, 2004.*, vol. 2, Denver, CO, USA: IEEE, 2004, pp. 977–982, ISBN: 978-0-7803-8465-1. DOI: 10.1109/PES.2004.1372983. [Online]. Available: <http://ieeexplore.ieee.org/document/1372983/> (visited on 05/07/2019).
- [70] R. Sharma, I. Majumder, S. Samantara, S. Goel and S. Samantaray, “Low Voltage DC Microgrid-Overview for Reduction in Substantial Losses,” en, in *Michael Faraday IET International Summit 2015*, Kolkata, India: Institution of Engineering and Technology, 2015, 36 (5 .)–36 (5 .) ISBN: 978-1-78561-186-5. DOI: 10.1049/cp.2015.1632. [Online]. Available: <https://digital-library.theiet.org/content/conferences/10.1049/cp.2015.1632> (visited on 05/07/2019).
- [71] L. Ott, Y. Han, O. Stephani, J. Kaiser, B. Wunder, M. Marz and K. Rykov, “Modelling and measuring complex impedances of power electronic converters for stability assessment of low-voltage DC-grids,” en, in *2015 IEEE First International Conference on DC Microgrids (ICDCM)*, Atlanta, GA, USA: IEEE, Jun. 2015, pp. 51–56, ISBN: 978-1-4799-9880-7. DOI: 10.1109/ICDCM.2015.7152009. [Online]. Available: <http://ieeexplore.ieee.org/document/7152009/> (visited on 05/07/2019).
- [72] M. R. E. Dangor, “An Interface for a Photovoltaic Module to an Ad-hoc, Low Voltage, DC Distributed Grid,” en, Dissertation, University of the Witwatersrand, Johannesburg, May 2019.

- [73] M. Aswat, “An Autonomous Smart Battery for an Isolated 12V Distributed DC Grid,” en, Dissertation, University of the Witwatersrand, Johannesburg, Oct. 2018.
- [74] Y. T. Quek, W. Woo and T. Logenthiran, “A multilevel threshold detection method for single-sensor multiple DC appliance states sensing,” en, in *2016 IEEE International Conference on Power System Technology (POWERCON)*, Wollongong, Australia: IEEE, Sep. 2016, pp. 1–6, ISBN: 978-1-4673-8848-1. DOI: 10.1109/POWERCON.2016.7753930. [Online]. Available: <http://ieeexplore.ieee.org/document/7753930/> (visited on 05/07/2019).
- [75] S. Leeb, S. Shaw and J. Kirtley, “Transient event detection in spectral envelope estimates for nonintrusive load monitoring,” en, *IEEE Transactions on Power Delivery*, vol. 10, no. 3, pp. 1200–1210, Jul. 1995, ISSN: 08858977. DOI: 10.1109/61.400897. [Online]. Available: <http://ieeexplore.ieee.org/document/400897/> (visited on 12/07/2019).
- [76] K. D. Anderson, M. E. Berges, A. Ocneanu, D. Benitez and J. M. Moura, “Event detection for Non Intrusive load monitoring,” en, in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, Montreal, QC, Canada: IEEE, Oct. 2012, pp. 3312–3317, ISBN: 978-1-4673-2421-2 978-1-4673-2419-9 978-1-4673-2420-5. DOI: 10.1109/IECON.2012.6389367. [Online]. Available: <http://ieeexplore.ieee.org/document/6389367/> (visited on 12/07/2019).
- [77] H. Wang and W. Yang, “An Iterative Load Disaggregation Approach Based on Appliance Consumption Pattern,” en, *Applied Sciences*, vol. 8, no. 4, p. 542, Apr. 2018, ISSN: 2076-3417. DOI: 10.3390/app8040542. [Online]. Available: <http://www.mdpi.com/2076-3417/8/4/542> (visited on 29/05/2020).
- [78] Z. Zheng, H. Chen and X. Luo, “A Supervised Event-Based Non-Intrusive Load Monitoring for Non-Linear Appliances,” en, *Sustainability*, vol. 10, no. 4, p. 1001, Mar. 2018, ISSN: 2071-1050. DOI: 10.3390/su10041001. [Online]. Available: <http://www.mdpi.com/2071-1050/10/4/1001> (visited on 29/05/2020).
- [79] B. Buddhahai, W. Wongseree and P. Rakkwamsuk, “An Energy Prediction Approach for a Nonintrusive Load Monitoring in Home Appliances,” en, *IEEE Transactions on Consumer Electronics*, vol. 66, no. 1, pp. 96–105, Feb. 2020, ISSN: 0098-3063, 1558-4127. DOI: 10.1109/TCE.2019.2956638. [Online]. Available: <https://ieeexplore.ieee.org/document/8917581/> (visited on 29/05/2020).

- [80] L. De Baets, C. Develder, T. Dhaene and D. Deschrijver, “Detection of unidentified appliances in non-intrusive load monitoring using siamese neural networks,” en, *International Journal of Electrical Power & Energy Systems*, vol. 104, pp. 645–653, Jan. 2019, ISSN: 01420615. DOI: 10.1016/j.ijepes.2018.07.026. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0142061517331253> (visited on 29/05/2020).
- [81] R. S. Butner, D. J. Reid, M. G. Hoffman, G. Sullivan and J. Blanchard, “Non-Intrusive Load Monitoring Assessment: Literature Review and Laboratory Protocol,” en, Tech. Rep. PNNL-22635, 1095438, Jul. 2013. DOI: 10.2172/1095438. [Online]. Available: <http://www.osti.gov/servlets/purl/1095438/> (visited on 13/07/2019).
- [82] M. Figueiredo, A. de Almeida and B. Ribeiro, “Home electrical signal disaggregation for non-intrusive load monitoring (NILM) systems,” en, *Neurocomputing*, vol. 96, pp. 66–73, Nov. 2012, ISSN: 09252312. DOI: 10.1016/j.neucom.2011.10.037. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0925231212003359> (visited on 18/06/2020).
- [83] S. N. Patel, T. Robertson, J. A. Kientz, M. S. Reynolds and G. D. Abowd, “At the Flick of a Switch: Detecting and Classifying Unique Electrical Events on the Residential Power Line (Nominated for the Best Paper Award),” en, in *UbiComp 2007: Ubiquitous Computing*, J. Krumm, G. D. Abowd, A. Seneviratne and T. Strang, Eds., vol. 4717, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 271–288, ISBN: 978-3-540-74852-6. DOI: 10.1007/978-3-540-74853-3_16. [Online]. Available: http://link.springer.com/10.1007/978-3-540-74853-3_16 (visited on 29/05/2020).
- [84] F. Sultanem, “Using appliance signatures for monitoring residential loads at meter panel level,” en, *IEEE Transactions on Power Delivery*, vol. 6, no. 4, pp. 1380–1385, Oct. 1991, ISSN: 08858977. DOI: 10.1109/61.97667. [Online]. Available: <http://ieeexplore.ieee.org/document/97667/> (visited on 18/06/2020).
- [85] K. Carrie Armel, A. Gupta, G. Shrimali and A. Albert, “Is disaggregation the holy grail of energy efficiency? The case of electricity,” en, *Energy Policy*, vol. 52, pp. 213–234, Jan. 2013, ISSN: 03014215. DOI: 10.1016/j.enpol.2012.08.062. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0301421512007446> (visited on 29/05/2020).
- [86] T. Bernard, D. Wohland, J. Klaassen and G. vom Bogel, “Combining several distinct electrical features to enhance nonintrusive load monitoring,” en, in *2015 International Conference on Smart Grid and Clean Energy Technologies*

- (*ICSGCE*), Offenburg, Germany: IEEE, Oct. 2015, pp. 139–143, ISBN: 978-1-4673-8732-3 978-1-4673-8734-7. DOI: 10.1109/ICSGCE.2015.7454285. [Online]. Available: <http://ieeexplore.ieee.org/document/7454285/> (visited on 29/05/2020).
- [87] C. Belley, S. Gaboury, B. Bouchard and A. Bouzouane, “An efficient and inexpensive method for activity recognition within a smart home based on load signatures of appliances,” en, *Pervasive and Mobile Computing*, vol. 12, pp. 58–78, Jun. 2014, ISSN: 15741192. DOI: 10.1016/j.pmcj.2013.02.002. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1574119213000370> (visited on 18/06/2020).
- [88] F. Paradiso, F. Paganelli, D. Giuli and S. Capobianco, “Context-Based Energy Disaggregation in Smart Homes,” en, *Future Internet*, vol. 8, no. 1, p. 4, Jan. 2016, ISSN: 1999-5903. DOI: 10.3390/fi8010004. [Online]. Available: <http://www.mdpi.com/1999-5903/8/1/4> (visited on 29/05/2020).
- [89] N. Sadeghianpourhamami, J. Ruyssinck, D. Deschrijver, T. Dhaene and C. Develder, “Comprehensive feature selection for appliance classification in NILM,” en, *Energy and Buildings*, vol. 151, pp. 98–106, Sep. 2017, ISSN: 03787788. DOI: 10.1016/j.enbuild.2017.06.042. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0378778817314366> (visited on 29/05/2020).
- [90] J. Froehlich, E. Larson, S. Gupta, G. Cohn, M. Reynolds and S. Patel, “Disaggregated End-Use Energy Sensing for the Smart Grid,” en, *IEEE Pervasive Computing*, vol. 10, no. 1, pp. 28–39, Jan. 2011, ISSN: 1536-1268. DOI: 10.1109/MPRV.2010.74. [Online]. Available: <http://ieeexplore.ieee.org/document/5590234/> (visited on 29/05/2020).
- [91] L. Pereira, M. Ribeiro and N. Nunes, “Engineering and deploying a hardware and software platform to collect and label non-intrusive load monitoring datasets,” en, in *2017 Sustainable Internet and ICT for Sustainability (SustainIT)*, Funchal: IEEE, Dec. 2017, pp. 1–9, ISBN: 978-3-901882-99-9. DOI: 10.23919/SustainIT.2017.8379791. [Online]. Available: <https://ieeexplore.ieee.org/document/8379791/> (visited on 29/05/2020).
- [92] Dao Liu, “Energy Saving Through Voltage Optimization & Non-intrusive Load Monitoring in Domestic House,” en, p. 213, May 2015.
- [93] Topward, *Topward - 6303D Power Supply*, English. [Online]. Available: <https://www.topward.com.tw/ps.htm> (visited on 02/07/2020).
- [94] University of the Witwatersrand, *PeCo Power*. [Online]. Available: <https://pecopower.co.za/> (visited on 03/07/2020).

- [95] W. Doorsamy and W. A. Cronje, “A NOVEL NATURE-INSPIRED PICO-GRID FOR FLEXIBLE PV APPLICATION IN RURAL ELECTRIFICATION SYSTEMS,” en, p. 6,
- [96] *Modbus Standard*. [Online]. Available: <https://modbus.org/> (visited on 02/07/2020).
- [97] Siemens, *Siemens Logo! Manual*, English, Mar. 2019.
- [98] *PC Oscilloscope, Data Logger & RF Products — Pico Technology*. [Online]. Available: <https://www.picotech.com/> (visited on 13/07/2019).
- [99] “IEEE Std 1159-2019, IEEE Recommended Practice for Monitoring Electric Power Quality,” en, p. 98, 2019.
- [100] *Picotech - Application Note: Analog Offset*. [Online]. Available: <https://www.picotech.com/library/application-note/using-analog-offset-to-maximize-oscilloscope-resolution> (visited on 02/07/2020).
- [101] Picotech, *PicoScope Oscilloscope Software*. [Online]. Available: <https://www.picotech.com/library/picoscope/picoscope-6.14.23> (visited on 02/07/2020).
- [102] —, *Picoscope API*. [Online]. Available: <https://github.com/picotech> (visited on 02/07/2020).
- [103] —, *Picoscope API Python*. [Online]. Available: <https://github.com/picotech/picosdk-python-wrappers> (visited on 02/07/2020).
- [104] S. Makonin, “Investigating the switch continuity principle assumed in Non-Intrusive Load Monitoring (NILM),” en, in *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, Vancouver, BC, Canada: IEEE, May 2016, pp. 1–4, ISBN: 978-1-4673-8721-7. DOI: 10.1109/CCECE.2016.7726787. [Online]. Available: <http://ieeexplore.ieee.org/document/7726787/> (visited on 29/05/2020).
- [105] J. Chung, “MACHINE LEARNING CLASSIFICATION TECHNIQUES FOR NON-INTRUSIVE LOAD MONITORING,” en, p. 91,
- [106] M. Dekking, Ed., *A Modern Introduction to Probability and Statistics: Understanding Why and How*, ser. Springer Texts in Statistics. London: Springer, 2005, ISBN: 978-1-85233-896-1.
- [107] K. Yao and J. Gao, “Law of Large Numbers for Uncertain Random Variables,” *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 3, pp. 615–621, Jun. 2016, ISSN: 1063-6706, 1941-0034. DOI: 10.1109/TFUZZ.2015.2466080. [Online]. Available: <http://ieeexplore.ieee.org/document/7182284/> (visited on 02/07/2020).

- [108] F. Jazizadeh, B. Becerik-Gerber, M. Berges and L. Soibelman, “An unsupervised hierarchical clustering based heuristic algorithm for facilitated training of electricity consumption disaggregation systems,” en, *Advanced Engineering Informatics*, vol. 28, no. 4, pp. 311–326, Oct. 2014, ISSN: 14740346. DOI: 10.1016/j.aei.2014.09.004. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1474034614000913> (visited on 01/07/2020).
- [109] T. Bernard and M. Marx, “Unsupervised Learning Algorithm using multiple Electrical Low and High Frequency Features for the task of Load Disaggregation,” en, p. 5,
- [110] M. Sandelowski, “Sample size in qualitative research,” en, *Research in Nursing & Health*, vol. 18, no. 2, pp. 179–183, Apr. 1995, ISSN: 01606891, 1098240X. DOI: 10.1002/nur.4770180211. [Online]. Available: <http://doi.wiley.com/10.1002/nur.4770180211> (visited on 02/07/2020).
- [111] M. C. Baechler and H. Hao, “Business Case for Nonintrusive Load Monitoring,” en, Tech. Rep. PNNL–25425, 1373019, May 2016. DOI: 10.2172/1373019. [Online]. Available: <http://www.osti.gov/servlets/purl/1373019/> (visited on 08/06/2020).
- [112] *Scipy - decimate*. [Online]. Available: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.decimate.html>.
- [113] *Scipy - resample_poly*. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.resample_poly.html.
- [114] Alan V. Oppenheim, Ronald W. Schaffer and John R. Buck, *Discrete-Time Signal Processing*.
- [115] F. Harris, *Multirate Signal Processing for Communication Systems*, en. Upper Saddle River, N.J: Prentice Hall PTR, 2004, ISBN: 978-0-13-146511-4.
- [116] H. J. Milton, “The selection of preferred metric values for design and construction,” en, p. 88,
- [117] T. Warren Liao, “Clustering of time series data—a survey,” en, *Pattern Recognition*, vol. 38, no. 11, pp. 1857–1874, Nov. 2005, ISSN: 00313203. DOI: 10.1016/j.patcog.2005.01.025. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0031320305001305> (visited on 02/06/2020).
- [118] S. Aghabozorgi, A. Seyed Shirkhorshidi and T. Ying Wah, “Time-series clustering – A decade review,” en, *Information Systems*, vol. 53, pp. 16–38, Oct. 2015, ISSN: 03064379. DOI: 10.1016/j.is.2015.04.007. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0306437915000733> (visited on 02/06/2020).

- [119] C. C. Aggarwal and C. K. Reddy, Eds., *Data Clustering: Algorithms and Applications*, eng, ser. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. Boca Raton: CRC Press, 2014, Editors: Charu C. Aggarwal (Research scientist at IBM T. J. Watson Research Center, Yorktown Heights, NY, USA); Chandan K. Reddy (Assistant Professor in the Department of Computer Science at Wayne State University, Detroit, MI), ISBN: 978-1-4665-5822-9 978-1-4665-5821-2.
- [120] H. Liu, H. Wu and C. Yu, “A hybrid model for appliance classification based on time series features,” en, *Energy and Buildings*, vol. 196, pp. 112–123, Aug. 2019, ISSN: 03787788. DOI: 10.1016/j.enbuild.2019.05.028. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0378778819305614> (visited on 29/05/2020).
- [121] M. Christ, A. W. Kempa-Liehr and M. Feindt, “Distributed and parallel time series feature extraction for industrial big data applications,” en, *arXiv:1610.07717 [cs]*, May 2017. arXiv: 1610.07717 [cs]. [Online]. Available: <http://arxiv.org/abs/1610.07717> (visited on 29/05/2020).
- [122] *Tsfresh*, Open source. [Online]. Available: tsfresh.readthedocs.io/en/latest (visited on 05/08/2020).
- [123] D. Virmani, S. Taneja and G. Malhotra, “Normalization based K means Clustering Algorithm,” en, p. 5,
- [124] scikit-learn, *Preprocessing Data - scikit-learn*. [Online]. Available: <https://scikit-learn.org/stable/modules/preprocessing.html#> (visited on 02/07/2020).
- [125] —, *Sklearn.preprocessing.StandardScaler*. [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html#sklearn.preprocessing.StandardScaler> (visited on 03/07/2020).
- [126] —, *Sklearn.preprocessing.MinMaxScaler*. [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html#sklearn.preprocessing.MinMaxScaler> (visited on 03/07/2020).
- [127] —, *Sklearn.preprocessing.MaxAbsScaler*. [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MaxAbsScaler.html#sklearn.preprocessing.MaxAbsScaler> (visited on 03/07/2020).
- [128] —, *Sklearn.preprocessing.RobustScaler*. [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html#sklearn.preprocessing.RobustScaler> (visited on 03/07/2020).

- [129] ———, *Sklearn.preprocessing.QuantileTransformer*. [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.QuantileTransformer.html#sklearn.preprocessing.QuantileTransformer> (visited on 03/07/2020).
- [130] I. Guyon and A. Elisseeff, “An Introduction to Variable and Feature Selection,” en, p. 26,
- [131] R. Kohavi and G. H. John, “Wrappers for feature subset selection,” en, *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273–324, Dec. 1997, ISSN: 00043702. DOI: 10.1016/S0004-3702(97)00043-X. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S000437029700043X> (visited on 02/06/2020).
- [132] A. L. Bluma, “Selection of relevant features and examples in machine,” en, p. 27,
- [133] scikit-learn, *Sklearn.feature_selection.SelectKBest*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html (visited on 03/07/2020).
- [134] ———, *Sklearn.feature_selection.f_classif*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_classif.html#sklearn.feature_selection.f_classif (visited on 03/07/2020).
- [135] John H. McDonald, *One-way anova*. [Online]. Available: <http://www.biostathandbook.com/onewayanova.html> (visited on 03/07/2020).
- [136] B. D. Fulcher and N. S. Jones, “Highly comparative feature-based time-series classification,” en, *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 12, pp. 3026–3037, Dec. 2014, ISSN: 1041-4347, 1558-2191, 2326-3865. DOI: 10.1109/TKDE.2014.2316504. arXiv: 1401.3531. [Online]. Available: <http://arxiv.org/abs/1401.3531> (visited on 02/06/2020).
- [137] A. K. Jain, M. N. Murty and P. J. Flynn, “Data clustering: A review,” en, *ACM Computing Surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, Sep. 1999, ISSN: 0360-0300, 1557-7341. DOI: 10.1145/331499.331504. [Online]. Available: <http://dl.acm.org/doi/10.1145/331499.331504> (visited on 02/06/2020).
- [138] V. Estivill-Castro, “Why so many clustering algorithms: A position paper,” en, *ACM SIGKDD Explorations Newsletter*, vol. 4, no. 1, pp. 65–75, Jun. 2002, ISSN: 19310145. DOI: 10.1145/568574.568575. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=568574.568575> (visited on 03/07/2020).

- [139] scikit-learn, *Clustering Performance Evaluation*. [Online]. Available: <https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation> (visited on 03/07/2020).
- [140] L. Hubert and P. Arabie, “Comparing partitions,” en, *Journal of Classification*, vol. 2, no. 1, pp. 193–218, Dec. 1985, ISSN: 0176-4268, 1432-1343. DOI: 10.1007/BF01908075. [Online]. Available: <http://link.springer.com/10.1007/BF01908075> (visited on 02/07/2020).
- [141] A. Ben-Hur and I. Guyon, “Detecting Stable Clusters Using Principal Component Analysis,” en, in *Functional Genomics*. New Jersey: Humana Press, Mar. 2003, vol. 224, pp. 159–182, ISBN: 978-1-59259-364-4. DOI: 10.1385/1-59259-364-X:159. [Online]. Available: <http://link.springer.com/10.1385/1-59259-364-X:159> (visited on 02/06/2020).
- [142] A. Reinhardt, D. Burkhardt, M. Zaheer and R. Steinmetz, “Electric appliance classification based on distributed high resolution current sensing,” en, in *37th Annual IEEE Conference on Local Computer Networks – Workshops*, Clearwater, FL, USA: IEEE, Oct. 2012, pp. 999–1005, ISBN: 978-1-4673-2129-7 978-1-4673-2130-3 978-1-4673-2128-0. DOI: 10.1109/LCNW.2012.6424093. [Online]. Available: <http://ieeexplore.ieee.org/document/6424093/> (visited on 29/05/2020).
- [143] *Tsfresh - Time Series Feature extraction based on scalable hypothesis tests*, Open Source Software. [Online]. Available: <https://github.com/blue-yonder/tsfresh> (visited on 03/07/2020).
- [144] J. Liang, S. K. K. Ng, G. Kendall and J. W. M. Cheng, “Load Signature Study—Part II: Disaggregation Framework, Simulation, and Applications,” en, *IEEE Transactions on Power Delivery*, vol. 25, no. 2, pp. 561–569, Apr. 2010, ISSN: 0885-8977, 1937-4208. DOI: 10.1109/TPWRD.2009.2033800. [Online]. Available: <http://ieeexplore.ieee.org/document/5337970/> (visited on 07/06/2019).

Appendix A

Appendix

This chapter covers the additional protocols that the methodology introduced, it illustrates some findings related to the tests therein.

A.1 Four State Device

The last type II device observed is an interesting case; this protocol represents the protocol corresponding to the floor standing fan that has four states. This device, discussed in section 4.2.2.2 which contains the rotary switch and has a different set of transient response characteristics compared to other devices.

The transients that are exposed by this device, when changing states, are indicative of the changing of the in-line resistance, the different states correspond to differing resistances.

This particular device emits a significant amount of noise on the grid when operating; above this, due to the size of the fan, the time it takes the physical fan to reach the steady state rotational speed, and this is longer than the event window. This specific case results in a situation where the dynamics of the transient event take longer than other devices tested, meaning that there may be some amount of unique information that takes place after the standard event window.

This may be one of the reasons why this device has a lower overall ARI compared to other devices tested.

Figure A.1 illustrates some of the state changes that characterise the fan. Simply

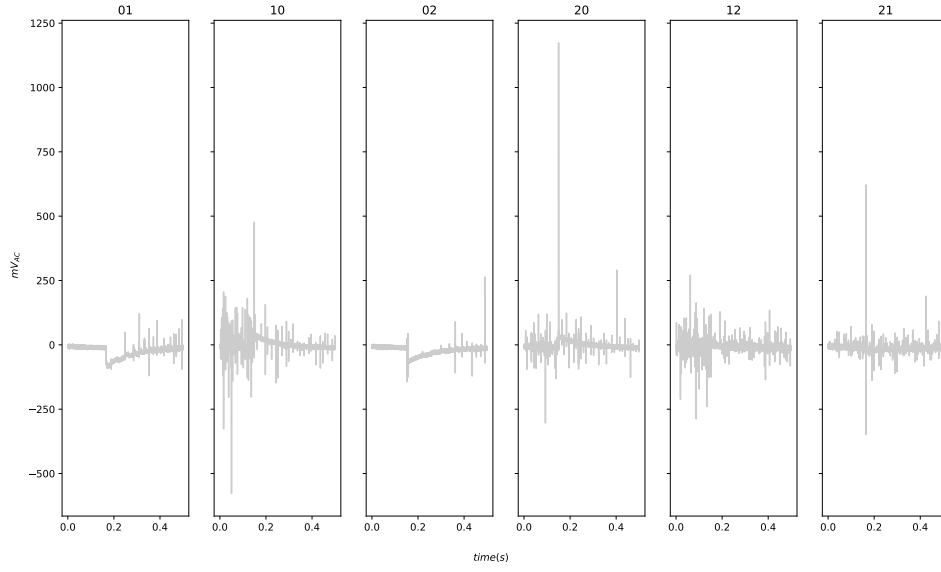


Figure A.1: Type II device (four states total) - state transition waveforms

observing the transient signatures for these state changes, it is clear that the differences between some of the state changes is very minimal. This makes it relatively difficult to understand which operating state the fan is in; however, what this protocol highlights is that although it is not possible to reliably predict the operating state of the fan, it is easy to know whether the fan is operational or not. This most basic level of understanding, of knowing whether the fan is ON or OFF, can partially assist in the disaggregation process.

A.2 Decreasing Grid Voltage

This set of protocols, containing the same light bulb throughout, contains a number of tests where the grid voltage is decreased in steps of 1 V until the light does not turn on.

The results from this simple test show that, even with lower and lower grid voltages, it is still possible to observe the operation of the light.

A.3 Linear Regulator Power Supply

One of the tests introduced in the methodology was to observe an old linear regulator power supply to see if it is also possible to perform load disaggregation on such

systems.

Based on the measurement results, it is apparent that this source is just as easy as compared to the other source types, the clustering performance easily reaches an ARI of 1.0.

A.4 Multiple Similar Devices

The situation of multiple different lights are covered in multiple other protocols in this work (see section 5.1.3.1), and based on those protocols it is clear that similar lights are easily differentiable from one another.

Additionally, the range of devices used in this work include two different DC phone chargers; these are tested in a range of protocols and the results suggest that they also follow the same characteristics of the different lights. The results, though not easily reaching an ARI of 1.0, perform similarly to the multiple similar lights situations; the best ARI that the K-Means algorithm can achieve is 0.65 which suggests that the two devices are quite similar from one another. Even though the phone chargers provide the same functionality, they can still be differentiated from one another.

A.5 Concurrent Device Switching

Standard electrical networks often include components where multiple devices are connected to a centralized switch, meaning that it is possible to have more than a single device actuated at the same instant, due to the fact that the devices are both connected to the same bus and that bus is then switched onto the grid.

This scenario is useful to illustrate here as it can function as a verification as to whether the grid observes two devices operating in isolation and then again in tandem as different signatures. This test takes two different LED lights, observes their signatures when they operate in isolation; the test then couples those two lights together such that the grid observes them as a single load switching ON and OFF.

The expected outcome for this situation is that there will be six clusters formed: firstly, the two signatures for the first device, the next two signatures for the second device, and the last two clusters for the *coupled* lighting load. This is verifying

whether the combination of the two lights operating is different from both of the two lights.

The highest achievable ARI, using K-Means for this protocol is 0.71 which suggests that there are sufficient differences from the three groups of transients: first light alone, second light alone, and two lights combined.

A.6 Current Signatures

The current signatures posed an interesting investigation as to whether it is possible to use the current as a feature for disaggregation. A protocol, outlined in section 4.5.3 suggested that the transients were not only to be found in the grid voltage.

The results found for this test show that the current is a very useful predictor for the switching of a device on the network, easily achieving an ARI of 1.0.

A.7 Source Combinations

This section illustrates the different combinations of sources that were tested within the methodology, the analysis found in section 5.3 highlights the fact that disaggregation is feasible with each of the different sources.

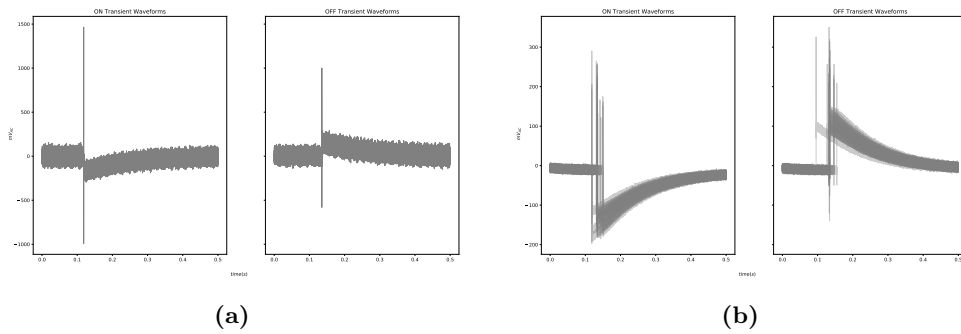
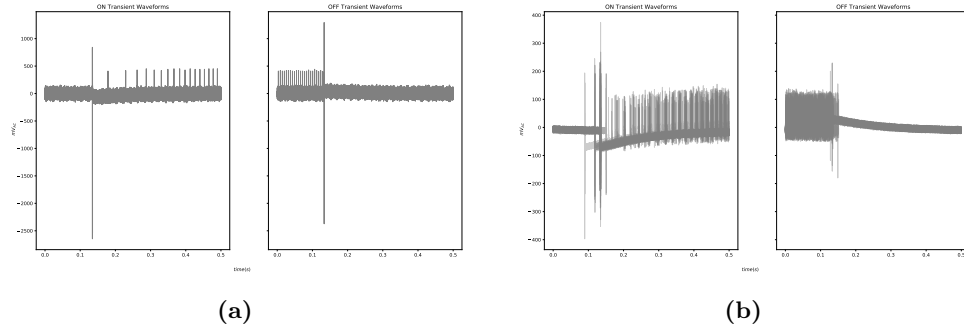
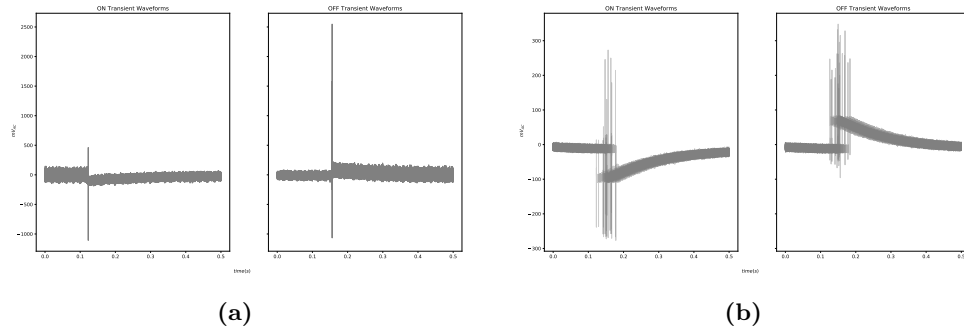
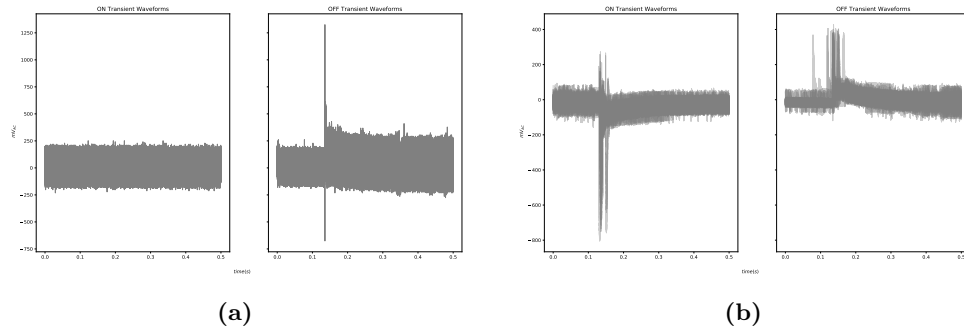
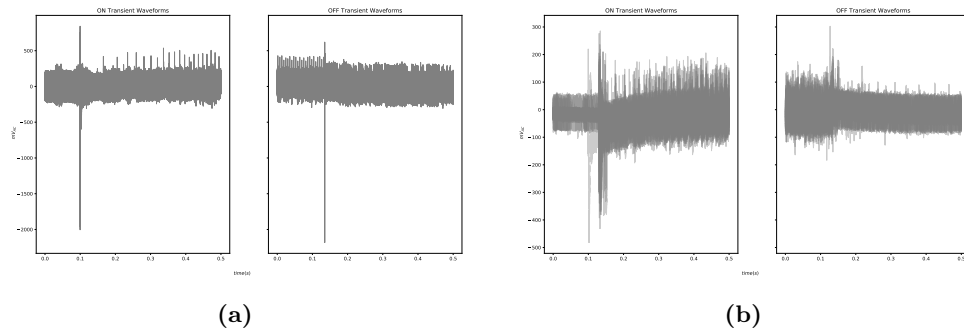


Figure A.2: DC Battery Device 1 - Transient waveform uniqueness and repeatability

**Figure A.3:** DC Battery Device 2 - Transient waveform uniqueness and repeatability**Figure A.4:** DC Battery Device 3 - Transient waveform uniqueness and repeatability**Figure A.5:** Smart Battery Device 1 - Transient waveform uniqueness and repeatability**Figure A.6:** Smart Battery Device 2 - Transient waveform uniqueness and repeatability

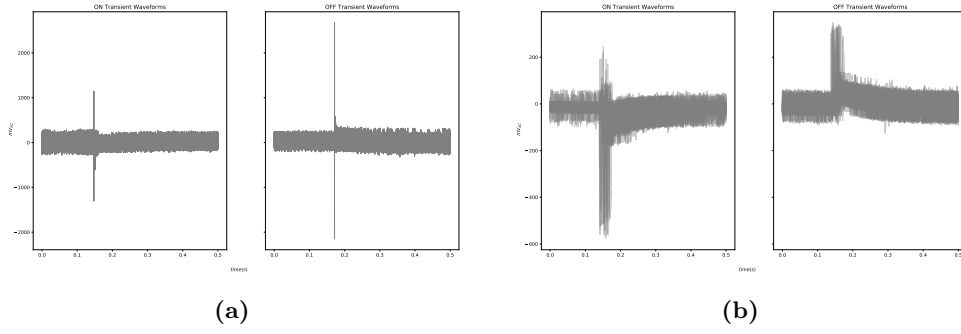


Figure A.7: Smart Battery Device 2 - Transient waveform uniqueness and repeatability

A.8 Inverter Driven Loads

Inverters are increasingly becoming useful to bridge the gap between a DC grid and old or standard AC appliances. The tests conducted in this methodology included the use of an inverter where different AC appliances were switched.

The AC devices included phone chargers and an IP camera.

Evident in these experiments is the fact these two devices, different in their end functionality, it is easily possible to disaggregate them from one another, K-Means achieves an ARI of 0.91, meaning that the clustering's of the devices are distinct and repeatable.

However, when it comes to the situation of the two phones, where the same phone charger is used but two different phone loads are introduced, the system is not able to distinguish the two loads from one another. The situation appears to be similar to that of the two replica lights, where there is a cluster for the ON transients and a cluster for the OFF transients.