



TU Clausthal

EVALUATING THE IMPACT OF DATA SAMPLING RATES ON EVENT DETECTION ACCURACY IN LOAD SIGNATURES USING A SHAPELET BASED APPROACH

BACHELOR'S THESIS

presented by

ABDUL RAHMAN HAKMEH

Energy Informatics group
Department of Informatics
Technische Universität Clausthal

ES-Bo09

Abdul Rahman Hakmeh: *Evaluating the Impact of Data Sampling Rates on Event Detection Accuracy in Load Signatures using a Shapelet based Approach*

STUDENT ID

462491

ASSESSORS

First assessor: Dr.-Ing. habil Andreas Reinhardt

Second assessor: Prof. Dr. Thorsten Grosch

SUBMISSION DATE

July 15, 2019

STATUTORY DECLARATION

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, wurden als solche kenntlich gemacht. Die Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsstelle vorgelegt.

Ich erkläre mich zudem mit der öffentlichen Bereitstellung meiner Bachelor's Thesis in der Instituts- und/oder Universitätsbibliothek einverstanden.

Clausthal-Zellerfeld, den 15. Juli 2019

Abdul Rahman Hakmeh

ABSTRACT

Non-intrusive load monitoring (NILM) is one of the practical applications of the smart electricity grid. NILM provides a comprehensive road-map to detect and determine operation circumstances of an individual appliance from an aggregate load. It can also provide the energy management system with information about appliance-level power consumption, analytical statistics, and detection of energy-hungry appliances to ensure the efficient use of the resources. The event detection is one of the NILM phases that is responsible for detecting state-changes of the appliances.

In this Bachelor's Thesis, a study is made to measure the impact of data sampling rate on the detection accuracy obtained through an event detection algorithm. Therefore, a machine learning based shapelet approach for electrical appliance use detection is implemented. The fast shapelet approach detects the occurrence of an event that is caused by switching on/off of an electrical appliance. A general architecture is introduced to evaluate the performance through feeding the algorithm with re-sampled data and comparing the obtained accuracy with the sampling rate. Different metrics are used to ensure a comprehensive evaluation; F1-score, confusion matrix, and classification accuracy.

The results of the experiment have shown that it is possible to detect appliances events with downsampled data using a shapelet based approach. Furthermore, it would have been as well possible that the downsampling improves the accuracy, but the evaluation of the event detection on noisy downsampled data have statistically verified that the lowering of the sampling rate is directly proportional to the accuracy of the detection in a linear form.

ZUSAMMENFASSUNG

Die Ereigniserkennung bzw. event detection ist eine Phase des NILM-ansatzes und spielt eine Rolle in seiner Genauigkeit bei der Disaggregation der Energiedaten. NILM kann Informationen zum Stromverbrauch auf der Geräteebeke zur Verfügung stellen und diese an das Energiemanagementsystem weiterleiten.

In dieser Bachelorarbeit wird eine Studie durchgeführt, um den Einfluss von niedrigeren Abtastraten bzw. Downsampling rates auf den Ereigniserkennungsprozess zu messen. Dafür ist ein bereits vorher bestand auf Shapelets basierender Ansatz zur Erkennung des Gebrauchs elektrischer Geräte implementiert und evaluiert worden. Die Aufgabe des Ansatzes ist es, das Auftreten eines Ereignisses zu erkennen, das durch das Ein- und Ausschalten von Elektrogeräten verursacht wird. Das Ziel dieser Arbeit ist es, die Genauigkeit des vorgeschlagenen Ansatzes zu bewerten, wobei verschiedene Abtastraten verwendet werden.

Die Ergebnisse der Studie haben bewiesen, dass es möglich wäre, mit Hilfe des Shapelet Ansatzes eine bessere Genauigkeit mit niedrigen Abtastraten zu kriegen. Es ist aber nicht der Fall mit den Daten, die ein großes Rauschen haben. Die Evaluierung der Ergebnisse hat gezeigt, dass das Shapelet Ansatz konnte keine bessere Genauigkeit mit niedrigen Abtastraten erreichen. Die Genauigkeit verringert sich linear proportional zur Verringerung der Abtastfrequenz.

ACKNOWLEDGMENTS

Throughout the writing of this thesis I have received a valuable assistance and support. Therefore, I would like to thank my thesis supervisor Dr.-Ing. habil Andreas Reinhardt for giving me the chance to join his research group and working on this research.

I would also like to extend my thanks to Msc. Jana Huchtkötter for the continuous support, supervision and the insightful recommendations she provided regarding my research and writing.

Finally, I must express my profound gratitude to my family for providing me with unfailing support and continuous encouragement throughout my years of study.

CONTENTS

1	INTRODUCTION	1
1.1	Motivation	2
1.2	Aim and Scope	3
1.3	Related Work	3
1.4	Outline of The Thesis	4
2	BACKGROUND	7
2.1	General Definitions	7
2.2	Energy Monitoring Overview	7
2.3	Intrusive Load Monitoring	8
2.4	Non-Intrusive Load Monitoring	9
2.5	Appliance Types and Signatures	11
2.6	Appliance Event Detection	12
2.7	Appliance Event Detection Approaches	13
2.7.1	Statistical Methods	13
2.7.2	Unsupervised Learning Methods	13
2.7.3	Shapelet based Supervised Learning Methods	14
2.8	Event Detection Evaluation Metrics	15
2.8.1	Classification Accuracy	16
2.8.2	CART Confusion Matrix	16
2.8.3	Precision	18
2.8.4	Recall	18
2.8.5	F1-Score	19
2.8.6	Imbalanced Data Problem	19
2.9	Sampling rate Conversions	20
2.10	Household Energy Datasets	20
2.10.1	Data-sets Summary	21
3	CASE STUDY	23
3.1	The Fast shapelet Discovery	23
3.1.1	Overview	23
3.1.2	Definitions and Notations	25
3.1.3	Generating SAX Words	27
3.1.4	Random Masking	27
3.1.5	Counting Similar Objects	28
3.1.6	Finding The Best Candidates	29
3.1.7	Algorithm Implementation	30
3.2	Data Structure	32
3.2.1	Data Source: BLUED	32
3.2.2	Data pre-processing	33

3.2.3	Sampling Frequency	37
3.3	Model Training	38
4	EMPIRICAL ANALYSIS	41
4.1	Event Detection Results and Discussion	41
4.1.1	Results of the phase A	41
4.1.2	Results of the phase B	46
4.2	Results Summary	50
5	CONCLUSION	53
5.1	Future Work	53
	BIBLIOGRAPHY	55
A	APPENDIX	65

ACRONYMS

NILM	Non-Intrusive Load Monitoring
DAQ	Data Acquisition
PPV	Positive Predictive Value
Hz	Hertz
SG	Smart Grid
kHz	Kilo-Hertz
MHz	Mega-Hertz
ILM	Intrusive Load Monitoring
FSM	Finite State Machines
CVD	Continuously Variable Devices
PCD	Permanent Consumer Devices
SAX	Symbolic Aggregate Approximation
RP	Random Projection
VA	Volt-Amperes
TS	Time Series
DTC	Decision Tree Classifier
BC	Binary Classifier
BLUED	Building-Level fUlly-labeled dataset for Electricity Disaggregation
AC	Alternating Current
GOF	Goodness-of-Fit
ADMM	Alternating Direction Method of Multipliers
REED	Reference Energy Disaggregation Dataset
UK-DALE	UK Domestic Appliance-Level Electricity
COOLL	Controlled On/Off Loads Library

WHITED Worldwide Household and Industry Transient Energy

PAA Piecewise Aggregate Approximation

INTRODUCTION

Around the globe, the energy systems are developed under the need to transition from fossil fuels to clean energy. Over the last decade the energy field witnessed a historic transformation while using new technologies. Some technological innovations have started a positive trend in fighting climate change according to *the Paris Agreement* [RDEH+16]. Today the scientific community in cooperation with all the key stakeholders are employing more efforts to provide an integrated and practical approach to reduce the energy needs and to look for new clean energy resources.

Reducing the emission of carbon dioxide is the most constraining condition for climate-change. This condition has motivated many creative ideas i.g. inventing the greenhouses by Walter Stamm-Teske in 2009 [LK11]. Such ideas make it easier to move towards a more reliable source of renewable energy. According to a study in 2017 [Mas17] a scaling up by 74 % of renewable energy generation can achieve the net-zero Carbon¹ dioxide emission by 2060. On the one hand the renewable energy gives an alternative to mitigate climate-change difficulties. On the other hand, due to high dependency on weather, these resources cannot be controlled. Furthermore, the climate-driven resources need more flexibility for consistent production and distribution of power over the network [MFZA15].

More ideas have been suggested to increase the flexibility of the distribution grid, such as an enhanced weather prediction system, and take the advantages of new technological advancement to improve the power conversion efficiency and integration of grid storage mechanisms [CGFB+06].

The improvement of grid flexibility needs, besides providing enhanced weather prediction, a power management system which ensures efficient use of the power resources. Such systems should consider not only the producer but the consumer. Active participation of both can increase the required flexibility and ensure the efficient use of the resources. Engaging the consumers in the energy management process means providing them with information about appliance-level power consumption, analytical statistics, and detection of energy-hungry appliances. [MABDV+12].

¹ Net-zero carbon refers to balance CO₂ emissions with carbon removal.

1.1 MOTIVATION

The Smart Grid (SG) is a term used worldwide to describe the next generation of power distribution network. The smart grid employs new technologies e.e. big data and analytic to expand the dimension of the traditional electricity grid like the transmission and the distribution automation [FMXY11]. In a recent study by the European Union, 72 % of the citizens in Europa by 2022 will have a smart meter to provide information about their electricity usage [GMO13]. Non-Intrusive Load Monitoring (NILM) is one of the practical applications of smart grid. NILM provides a comprehensive road-map to detect and determine operation circumstances and energy consumption of an individual appliance from an aggregate load (power consumption of households).

Such technology is used to optimize energy through the awareness of the consumers of the overall and appliance-level-specific consuming details in real time [FMXY12]. According to a study in 2016 4.5 % of power saving can be achieved by providing the consumers the real time power consumption at the device level [KK16].

In this work, the event detection is presented, a phase of NILM approach. Event detection is responsible for detecting state-changes of the appliances. In NILM the event detection process is followed by the futures extraction and the energy disaggregation phase. Energy disaggregation is the discernment of a specific applicant from a set of data which presents the consumption of the users. High appliance disaggregation accuracy is related to high event recognition accuracy. This goal demands more detailed analysis of device consumption and high-resolution² data. The nature of the collected data can influence the detection accuracy, the higher the resolution of the data the more accurate the detection [DROS14]. However, the noise can put a strain on the detection accuracy. Having a high sampling frequency data requires high-frequency Data Acquisition (DAQ) hardware, where the measurement and data collection take place. Furthermore, collecting high sampling frequency data requires proper storage solutions which demand more time and cost to handle [GRP10]. Therefore, we evaluate the impact of data sampling on the event detection in NILM process.

During the work, the following research challenge is faced: Is it achievable to detect appliance switching on or off events in downsampled data without sacrificing the detection accuracy and how does the low sampling rate affect the the event detection accuracy ?.

² Usually means several Kilo-Hertz (kHz).

1.2 AIM AND SCOPE

In this thesis, the use of fast shapelet [RK13] is presented, a machine learning based approach for electrical appliance use detection. This step corresponds to the event detection stage in typical event-based non-intrusive load monitoring [ABO+12].

The task of the fast shapelet is detecting the occurrence of an event that is caused by switching on or off an electrical appliance. The goal is to evaluate the performance of the implemented approach with re-sampled segments of load measurements, and analyze the impact of input data frequency on the detection process by changing the sampling rate as much as possible and comparing the results of the detection.

For the evaluation phase, different metrics are used; F1 score, confusion matrix, and classification accuracy to ensure a comprehensive evaluation.

1.3 RELATED WORK

NILM refers to an analytical method to break down the aggregate load consumption to the appliance level. NILM aims to analyze the collected data to come out with the load signatures which can give information about the operating state and the characteristics of the individual appliances. By using this information, NILM can provide the Energy System Management with power-consuming details and energy-hungry appliances to optimize the power-consuming and ensure the efficient use of the resources.

Since its inception in 1980, many approaches were proposed in the NILM field. NILM studies can be divided into two groups: transient-state analysis and steady-state analysis. The steady-state methods use reactive and active power [DNG+15]. Other researches are based on harmonics and current waveform [WACLog]. The transient-state researches can, in general, obtain more distinguishable appliances signatures [QLW18].

Event-Based NILM refers to many techniques for power load classification. It determine the state transition of the appliances which caused by switching on or off [QLW18]. Different classification approaches were proposed over the last 30 years for NILM event detection. Based on a statistical analyse, the Goodness-of-Fit based event detection used in [JTBS11] and Maximum likelihood classifier has been presented in [HAK+15]. On the one hand, likelihood showed

less sensitive to the grid noise. On the other hand, it only can detect the appliances type I [QLW18] (refer to the Section 2.5 for more information about appliance types). All the statistical methods can not handle slow state changes in the signal because of dividing the data into windows [DBRDD16]. Another unsupervised adaptive event detection method is presented in [BSY14b].

However, most of the proposed methods demand a high sampling rate in the order of kHz or Mega-Hertz (MHz). In a study [QLW18] on harmonics based NILM, it was suggested that the sampling rate should be at least 8 kHz. Such a sampling frequency is not accessible with the available smart meters, so the need for a low-rate NILM method has increased [QLW18].

Many machine learning methods were suggested for the low-rate NILM. A machine learning method based shapelet was proposed to solve the event detection problem. A study from university of Southern California [PPCP14] used a machine learning method of time series shapelet for electricity load disaggregation. The writers frame this task as that of event detection and event classification from BLUED energy dataset. On the one hand, they achieved approximately 98 % accuracy in the event detection phase. On the other hand, the classifier has been tested and trained with event segments which were extracted according to a fixed window around each event (5 seconds after the event and one second before). Moreover, the evaluation plan did not consider re-sampling the input instances to investigate the relationship between the sampling frequency and the algorithm's performance.

In contrast to all the previous mentioned researches, this work provides a comprehensive study which focuses on evaluating the detection of the electrical appliance's signatures while feeding the fast shapelet algorithm with re-sampled data and comparing the obtained accuracy with the sampling rates. It is expected to find a relation between the sampling rate and the classification accuracy.

1.4 OUTLINE OF THE THESIS

The thesis is structured as follows: first, in the Background Chapter 2 the borders of this work is defined, starting with providing general information about the energy monitoring and particularly about the NILM approach followed by an in-depth look into different approaches of the NILM event detection. Necessary notations with a formulation of the event detection evaluation metrics are presented, ending with an overview of the available energy datasets.

Later in the case study in [Chapter 3](#), our dependent variables; the chosen event detection approach fast shapelet based classifier and the energy dataset BLUED are presented. After that a satisfying explanation for the implementation steps, including the data pre-processing, sampling frequency, and training the classifier are discussed.

In the empirical analysis in [Chapter 4](#), the experiments, the results and the associated analysis of the sampling rate's impact on the event detection are addressed, followed by a section on evaluating the overall performance of the fast shapelet algorithm and answer the research challenge. The research challenge is already mentioned in the [Section 1.1](#).

The conclusions in [Chapter 5](#) has a summary of the results with suggestions to expand this research in the future.

BACKGROUND

In this chapter, the borders of this work are detailed, starting with providing an in-depth look into NILM and explaining different event detection methods which were mentioned before in the related work in the [Section 1.3](#), ending with addressing the used evaluation metrics and the downsampling techniques.

2.1 GENERAL DEFINITIONS

- ▷ **Active power:** the actual consumed power which is utilized in an AC-circuit (also known as true power or real power). It is given in Watt [[DTDT72](#)].
- ▷ **Reactive power:** in contrast to the active power, reactive power can do any work like lighting or heating, it only moves in the circuit in both directions (known also as the useless watts), it is measured in Volt-Amperes ([VA](#)).
- ▷ **Time series data type:** time series is samples of data which are uniformly ordered through the time with a constant sampling period [[Ful18](#)].
- ▷ **Classification algorithms:** classification algorithm is an approach in which the computer learns from the input data and then uses this knowledge to distinguish coming observations [[SSBD14](#)].
- ▷ **Power load data:** time series data which represents the power consumption of an electrical component or many components.

2.2 ENERGY MONITORING OVERVIEW

The global warming and the cost of the electricity extraction are the common challenges encountered by the power sector. In this field, many studies pointed out that because of globalization, the energy demand is increasing drastically worldwide, and it can be handled only by an efficient power management system [[AKM+17](#)].

The term 'efficient power management' has many interpretations, one of them is that the operating condition of appliances needs to be ascertained, and this task cannot be done without a proper monitoring system that takes into account more energy sustainability

measures [RVV+14]. The monitoring system facilitates the conservation of energy by utilizing energy efficient measures like the timing of the appliance's usage and defining devices with lower energy consumption and eliminating unwanted hungry-energy activities [SNN+14].

Load monitoring is a method of recognizing and analyzing the load measurement in the power system. These measures contain the power consumption and the device's state and signatures [AF15]. The load monitoring can be described as Intrusive Load Monitoring (ILM) or Non-intrusive load monitoring NILM. In the following section, a brief overview of both is presented.

2.3 INTRUSIVE LOAD MONITORING

The basic idea of ILM is involving the installation of measuring devices at every load of interest. The presence of many sets of sensors in such systems makes it expensive and more sophisticated in installation and maintenance [BDS13].

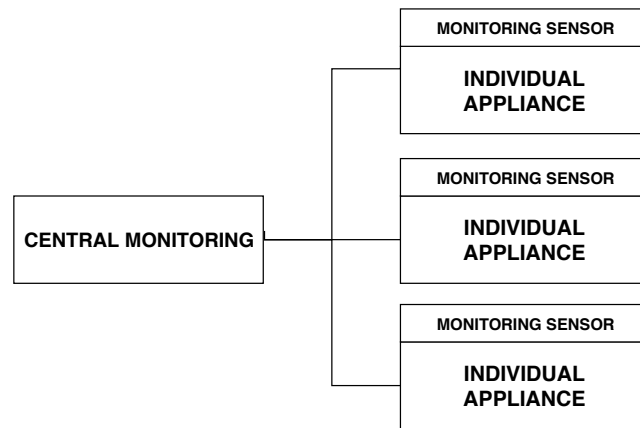


Figure 2.1: Intrusive Load Monitoring. [AKM+17]

On the one hand, the presence of submetering in the system makes it more efficient on the functionality of all submeters to ensure the accuracy of monitoring function. On the other hand in case one of these submeters fails, the accuracy of the system is affected. The general concept of ILM is represented in the Figure 2.1.

In ILM a group of appliances which is connected to a single set of sensors decides the level of the intrusiveness [RGH14]. In the praxis, ILM is divided into three layers, and every layer is connected to the next one [RGH15] :

1. **Detecting a single appliance:** this is when a sensor is installed as a smart plug to recognize the appliance's state (on/off) and its location.
2. **Middleware Phase:** the middle-ware is a software component which works as an interpreter to translate the state of the appliances.
3. **Appliance status phase:** this layer is the final stage of **ILM** that presents the actual state of the appliance for the monitoring and the control.

Although **ILM** provides accurate results as mentioned before, it demands high costs and a complicated installation process because of wiring and installing data storage components for the households [AF15].

2.4 NON-INTRUSIVE LOAD MONITORING

In contrast to **ILM**, non-intrusive load monitoring does not require intrusion into the individual appliance (or a group of appliances) when its power consumption is monitored. **NILM** is described as the process of gaining and disaggregating the electricity usage of all electrical devices in the household [PHM11].

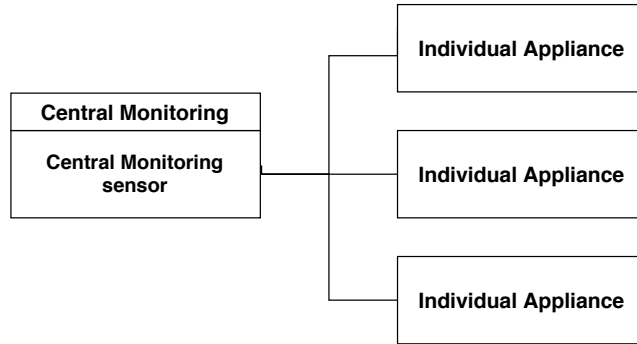


Figure 2.2: Non-Intrusive Load Monitoring. [AKM+17]

NILM analyzes the collected wave-forms to come out with the load features or load signatures, which are the measurable parameters of the whole load that can give information about the operating state and the characteristics of the individual appliances [AHS11]. The Figure 2.2 illustrates the **NILM** by showing the main difference between intrusive and non-intrusive method, namely, the only one set of monitoring sensors at a central monitoring entry. Because of this difference, **NILM** is considered as a low cost choice in comparison to the **ILM** [BKP+14].

Since its inception, **NILM** researches have been considerably developed as many new tools and methods for the features extraction and the load-disaggregation were invented [GB15].

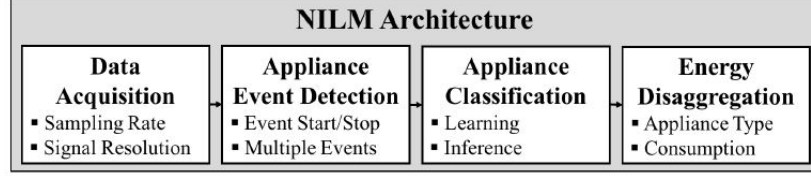


Figure 2.3: Non-Intrusive Load Monitoring architecture. [AGSA13]

NILM consist mainly of a combination of four stages, data acquisition, appliance event detection, appliance classification, and energy disaggregation as shown in the Figure 2.3. Every stage depends on the one preceding it:

- ▷ In **DAQ** phase, the aggregate load of building or house is measured (the voltage and the current) at a given time interval, regarding the area of the application and the used disaggregation method, the incoming data can be acquired at high or low frequency [AGSA13].
- ▷ **The event detection phase** is a stage of **NILM** and it is responsible for detecting state-changes of the appliances. More detailed information about the **NILM** event detection is presented later in the Section 2.6.
- ▷ **The appliance classification** is responsible for feature selection and extraction by focusing on learning and inference to build an energy consumption profile for every usage pattern. The advances in machine learning techniques and the existing machine learning algorithms, such as the k-nearest neighbor [FB08] and the artificial neural network [GFS+11; KK15a] have had a significant impact on the learning and inference stage.
- ▷ **Energy disaggregation** stage represents the detecting of appliances by using a classification approach. The last stage is comparing the power consumption of every appliance with previously known characteristics of a typical appliance in the database [BGMS10]. In addition to the disaggregation, calculating the electricity consumed by a device at a specific time gives essential information about the usage behavior of the appliances [BSY14a].

2.5 APPLIANCE TYPES AND SIGNATURES

The recognition of appliance's signatures (also known as unique energy consumption pattern) is the main area for NILM in the learning and inference phase. These patterns allow the disaggregation algorithms to identify the different operations of aggregated load measurements [ZGIR12].

Appliances signatures can be grouped into two main categories macroscopic and microscopic signatures. In a classification task, **macroscopic** features refer to the changes in the reactive and the real power. To detect such events, low sampling rate data (seconds/Hertz (Hz)) is enough. The low sampled data usually works fine for a small building and household, where a limited number of devices are working, and each device has its own unique features [NMR18]. The **microscopic** signatures refer to an event that occurs within an Alternating Current (AC) cycle. To detect such small events, high-sampled data (milliseconds/kHz) is necessary. High sampled data is used to detect events for the industrial environments such as a supermarket or a factory with high appliance density [NMR18].

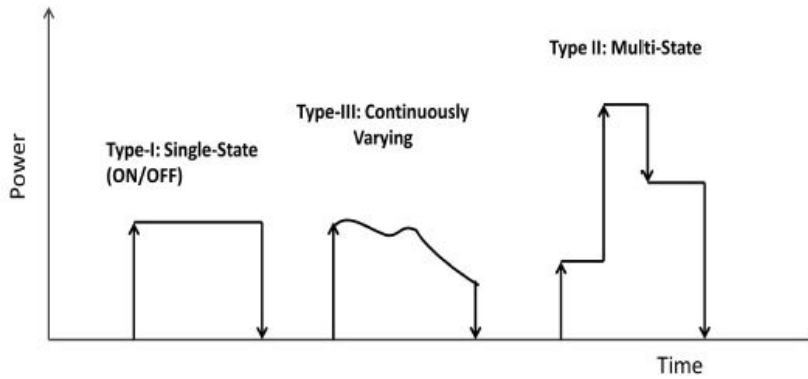


Figure 2.4: load types based on power consumption pattern [ZGIR12]

Consumption patterns are distinguished by the appliance category, as [Har92] proposed, the appliances can be classified based on their state as the follows:

1. Type I: represents the appliances that have only two states on and off, such as the majority of electric devices, e.g. toaster and lamps.
2. Type II: multi-state appliances represent the appliances with finite state of operating Finite State Machines (FSM). e.g. The washing machine. The identification of such operations is comparatively possible because the switching pattern is repeatable.

3. Type III: this category includes Continuously Variable Devices (CVD). The operation characteristics of these devices do not have a fixed number of state. i.g. dimmer lights. what makes the recognition of such appliances in the load measurements more challengeable that there is no repeatability in their switching patterns.
4. Type IV: this category is proposed by [ZR11; Har92]. the devices belonging to this category remain active for days consuming power at a fixed rate. The term Permanent Consumer Devices (PCD) refers to such devices like telephone sets. The Figure 2.4 in the following shows the explained devices types and their consumption pattern with time vs. power.

2.6 APPLIANCE EVENT DETECTION

Event detection is responsible for feature extraction where the collected data from the DAQ phase is processed in order to detect the individual appliance events [AGSA13]. The Figure 2.5 shows a sample from aggregate load collected by a smart meter from a lighting system. The reader can realize the changes in power due to the status of the different appliance.

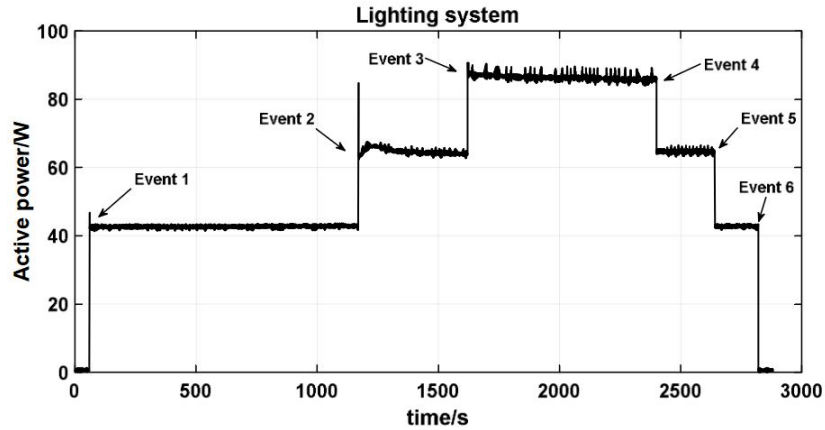


Figure 2.5: Power changes due to individual appliance status [LL19]

There are two known main classes of feature extraction; steady-state signature where the signal amplitude and its smooth variations are analyzed [NL96], and the transient feature which represents the changes in the current's waveform as a unique pattern an appliances consumption's follows and it is used to identify the different appliance. Usually, the unique pattern can be captured by switching the appliance on and off [Cha12].

As mentioned in the introduction in the [Chapter 1](#), the event detection is one of the main steps in the [NILM](#) approach and besides reducing errors in the data acquisition phase, event detection is considered as the paramount importance of the accuracy in energy disaggregation. High appliance disaggregation accuracy is related to high event detection accuracy [[DROS14](#)]. Many new event detection approaches have been presented in the last years taking the advantages of many fields in data analysis. A brief overview of the event detection methods and the used technologies is presented later in this chapter.

2.7 APPLIANCE EVENT DETECTION APPROACHES

The extraction of useful information from time series data helps the developers to improve their understanding of different data trends for the monitoring and the detection of deviations. [[AA15](#)]. Over the last years, many approaches have been proposed to identify events of electrical appliances. This section discusses some of these methods, which are already mentioned in the related works [Section 1.3](#).

2.7.1 Statistical Methods

Based on statistical analysis, many classification tools have been proposed for [NILM](#) event detection. Maximum likelihood classifier has been presented in [[HAK+15](#)], which has less sensitivity to the grid noise [[QLW18](#)]. Another statistical approach was made by Yuanwei Jin [[JTBS11](#)], a Goodness-of-Fit based event detection was proposed. The Goodness-of-Fit ([GOF](#)) method is used to deliver some probability distribution by assuming that two neighboring windows share a common distribution and the distribution of data values before and after an event is different. The method uses chi-squared test [[GN96](#)] (also written as χ^2 test) on every two neighboring windows. As mentioned before the most of the statistical methods can not handle the slow changes in the signal because of dividing the data into small windows.

2.7.2 Unsupervised Learning Methods

Unsupervised learning is a type of machine learning. The key idea is to find unknown patterns in a dataset without pre-defined labels.

2.7.2.1 Unsupervised Adaptive Event Detection

The first algorithm we discuss is a method developed for the detection and classification of events during the work on a fully unsupervised

NILM system, as presented in [BSY14b]. The method can accurately define the time limits of every transition period in a power signal. Furthermore, it detects the change intervals that occur as the appliance operates. Accurate detection of the change interval is essential for the appliances signature extraction phase in **NILM** that depend on transient features. The proposed method does not use any filtering method to avoid distorting the transient sections, especially during the detection phase. Moreover, the event detector has a dynamic adjustment to noise, so it does not need a pre-training to handle it [BSY14b].

2.7.3 *Shapelet based Supervised Learning Methods*

In the last decade, a new method called shapelets transform has attracted much attention in time series classification. Shapelets are subsequences of a time series that have high discriminative power. The key idea is that local variations can be used to distinguish between different classes of time series instead of using the global structure. In this section we introduce different classifiers based on shapelets discovery.

2.7.3.1 *Standard Shapelet-based Classifiers*

This method was proposed for the first time by Ye and Keogh in 2009. In this method, the best unique shapelets are searched across all the subsequences in time series; after this, a decision-tree is created based on the information gain. The key idea is that instead of calculating the distance between all subsequences of time series, the algorithm only compares a small subsection of the shapes [YK09]. This concept gives an efficient alternative solution to detect shapes in time series data in comparison to traditional classification methods like the nearest neighbor classifier. Despite its advantages, the search for shapelet is computationally costly.

2.7.3.2 *The Fast Shapelet*

Many solutions have been proposed to increase the efficiency of the searching process in the Standard Shapelet-based Classifier. In 2013 Rakthanmanon and Keogh [TR] proposed a novel method to make the search more efficient by reducing the dimensional space of the time series by using Symbolic Aggregate Approximation (**SAX**) representation [LKWLo7]. This method is implemented in this work, therefore is explained later in Section 3.1 with more detailed information about its work method.

2.7.3.3 *FLAG Shapelet*

Two years ago, a new different approach by Zurada and Kwok was proposed [HKZ16]. The study solves the shapelet discovery problem as a numerical optimization problem. It used fused lasso regularizer [TSR+05] and the generalized eigenvector method [MK14] to extract the features and ensure the discrimination of the obtained shapelets. Furthermore, the Alternating Direction Method of Multipliers (ADMM) is used for the optimization problem which is related to the obtained model. On the one hand, in the praxis, the experimental results show that this method is much faster than the previous shapelet-based methods while archiving similar or even better classification accuracy. On the other hand, due to its complexity in the implementation phase, this method has not been chosen for this study.

2.8 EVENT DETECTION EVALUATION METRICS

The ultimate goal of the evaluation in this work is to measure the impact of downsampled input data on detection accuracy after a comparison between the detection results and the downsampling rate. A good event detector is the one that can achieve a high detection accuracy with "unseen" downsampled data. To assess the classifier's performance, it's crucial to have the ground truth of the used data, which the classifier is evaluated with after the training phase. In this case, the truth data is a list of labels where every label is associated with one sequence of the data and defines its status: event or non-event.

Many metrics can be used to evaluate a Binary Classifier (BC)³. Every metric has a different parameter with different goals. After the searching in the literature, there is no standard metric can be found for evaluating the performance of the event detection classifiers. Therefore in the following sections a couple of possible methods are presented and discussed.

Before we start, we have to define a couple of primary building blocks which are used in this section. A **true positive** is a term that refers to a result where the model can correctly predict the positive class; likewise, a true negative refers to a result where the model can correctly predict the negative class. A **false positive** is a term that refers to a value where the model incorrectly predicts the positive class of the classification; likewise, a false negative refers to an incorrect prediction of the negative group [Met78].

³ classifying the elements of a given dataset into just two group.

2.8.1 Classification Accuracy

To start out, the classification accuracy is introduced. It has been presented first because it is an often used metric to measure the performance. The classification accuracy means the fraction of predictions the model got right [VRRVDK11]. Formally we can define the accuracy with the following definition.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \quad (2.1)$$

in our case the accuracy should be calculated by using positives and negatives terms as follows :

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \in [0, 1] \quad (2.2)$$

where

- TP True positives
- TN True negatives
- FP False positives
- FN False negatives

Although having high accuracy e.g. close to 80 % means that our classifier has shown exceptional performance. With a closer analysis, the reader will find out that the accuracy alone doesn't gain much information and can be very misleading. To explain more let's assume that we work on a dataset of 280 women who had breast cancer. Of the 280 women, 210 did not have a recurrence of cancer within five years. Now we have a model which can predict only no recurrence of breast cancer with an accuracy of $((210/280) \times 100) = 75\%$. In case of using this model alone to support doctors to take decisions it would have had 70 women sent home depending on prediction that they will not suffer a recurrence of breast cancer (high false negatives) [Met78]. This example shows how the high accuracy of a classifier alone can be misleading and not reliable in some cases.

To summarize it up, the classification accuracy has the advantage that it is a single simple number for measuring performance, but it must be interpreted with caution. [VRRVDK11].

2.8.2 CART Confusion Matrix

The limitation we faced with classification accuracy forces us to provide our evaluation scheme with some complexity: the wrong and the right

predictions had to be scored separately [Met78]. This need leads to the next metric, which is often used in the field of machine learning and statistical classification and can be considered as a basis for many common metrics [Fawo6].

The confusion matrix (also known as a contingency table or error matrix), is a visualization of the performance of a model in the form of a specific table. This matrix presents the number of false positives, true positives, false negatives, and false positives. it also provides more detailed analysis than the previous metric. Every row in a confusion matrix represents the cases in a predicted class, while every column means the cases in an actual class [VRRVDK11].

		<u>True Class</u>	
		p	n
<u>Hypothesized class</u>	Y	True Positives	False Positives
	N	False Negatives	True Negatives
Column totals :		P	N

Figure 2.6: Confusion matrix with performance metrics [Fawo6]

To understand the matrix, we assume having a binary classifier with two sets $\{p,n\}$. Each observation belongs to one of the sets. To compare the actual class with the predicted class, two labels are used $\{Y,N\}$ to name the predictions produced by the classifier. As the Figure 2.6 shows, for every outcome there are four possible cases⁴. If the observation is positive and it is predicted correctly as positive, it is scored as a true positive; likewise, if it is predicted as negative, it is scored as a false negative. If the observation is negative and is predicted correctly as negative, it is scored as a true negative; likewise, if it is predicted as positive, it is scored as a false positive [VRRVDK11].

In comparison to the classification Accuracy, The confusion matrix illustrates the wrong and the right predictions separately, so it is

⁴ In case of binary classifier otherwise can have table size $L \times L$

possible to obtain the category where the classifier could perform better and where it could not.

2.8.3 Precision

As we discussed before, in order to separate the wrong and the right predictions the precision also known as Positive Predictive Value (PPV) is used. Precision represents the ratio of positive observation, which is correctly predicted to the other observation in the same real class [Pow11]. Precision is defined as follows:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \in [0, 1] \quad (2.3)$$

In the evaluation phase, this metric can answer the question, how many events actually have been detected of all the observations that have labeled as events. Generally, a low false positive rate relates to a high precision, which ensures that the results are substantially more relevant [BG94]. The precision is often used with the recall metric which is presented in the next section.

2.8.4 Recall

Another method to separate between the wrong and the right predictions is Recall (also known as sensitivity). While precision refers to the proportion of the predicted results by giving the rate of positive observation, which is correctly predicted to the other observation in the same actual class, Recall expresses the ability to detect all relevant observations in a dataset [ABO+12]. It refers to the rate of correctly predicted positive observations to all observations in the real class [Pow11]. It can be mathematically defined as follows:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \in [0, 1] \quad (2.4)$$

To fully evaluate our model, we must examine recall and precision. It is not possible to get a high percentage of both. Improving of precision typically reduce recall and vice versa [DGo6].

2.8.5 *F1-Score*

In addition to the previous metrics, we present another metric that gives a single value to measure the performance to evaluate the effectiveness of a model. The F-score is used in the field of information retrieval and query classification [Sas+07]. F1-Score (also F-score or F-measure) is a combination of recall and precision and refers to the weighted average of both. Therefore, it takes false positives and false negatives into account [GG05]. The traditional F-measure is defined as follows:

$$F1 - score = 2 \cdot \frac{\text{precision} \cdot \text{Recall}}{\text{precision} + \text{Recall}} \in [0, 1] \quad (2.5)$$

It is not easy to separate the classification accuracy and F1-score. The main difference is that the F-measure is more useful and precise, especially when the numbers of observations in all classes are imbalanced or the data has uneven class distribution [JCDLT13].

2.8.6 *Imbalanced Data Problem*

The nature of the Building-Level fully-labeled dataset for Electricity Disaggregation (BLUED) dataset and the effect of this problem on the evaluation metrics was the motivation to discuss this issue in this section. The imbalanced data refers to a common problem in the classification where the classes are not represented equally [SWK09]. To understand this issue, Assume having a dataset 2-class with 200 observations. A total of 170 are already labeled with class 2, and the remaining 30 observations of the dataset belong to class 1. The problem is that there is one class which has more effect on the result than the other. It has ratio 170:30. We could have had a high classification accuracy, but by taking a closer look we discover that more than 90% of the data belonged to one class. Class imbalance problem can occur with two-class classification as well as multi class classification. Most datasets have no equal number of observations in each class but sometimes is just a small difference. Even in some cases, imbalanced classes are expected [SWK09] e.g. If we have a dataset that characterizes fraudulent transactions in a bank, surely this dataset has imbalanced classes because the majority of the transactions are labeled as "Not-Fraud" and a small group of observations is labeled as "Fraud". Many techniques can be followed to solve this problem. The Fast techniques can be re-sampling the data or changing the evaluation metric and others that need more analysis, like trying different algorithm or collecting more data [Cha09]. It is important to mention here that by

extracting the segments from the [BLUED](#), a massive difference between both classes (event and non-event) has been observed. A solution is proposed to solve this problem is introduced later in the [Section 3.2.2](#).

2.9 SAMPLING RATE CONVERSIONS

In many fields and applications, the re-sampling of a digitized signal is necessary for designing an efficient system. e.g. In wireless communications field changing the sampling rate is typically employed for down conversion and up conversion to meet the required frequency [[IJ02](#)]. Generally, the sampling rate can be increased by injecting additional samples within a sequence, this method is called up-sampling; likewise, the sample rate can be reduced by eliminating intermediate samples. This method is called downsampling. Both operations belong to a group of filters called multi-rate filters. This group includes a mechanism to increase or decrease the sample rate. In this section, we introduce the downsampling techniques[[RG75](#)].

As we mentioned before, downsampling is a method of decreasing the sample rate of a data sequence. Two standard methods are introduced in this study, namely the downsampling by integer factor and the downsampling by averaging. By using the first method, a signal $S[n]$ can be downsampled by factor K by holding every K^{th} sample and dropping the remaining samples. The new sample rate must be $\frac{1}{K}$ of the original signal and should have a bandwidth less than or equal to $\frac{1}{K}$ of the original signal [[IJ02](#)], otherwise an additional filter is required (low-pass filter⁵) [[SRCS91](#)]. The second method for downsampling uses the average of a block from data. We assume that $S[n]$ is downsampled by factor K so a subsequence from long k is taken and the average of its values is calculated. Every value in the resulting signal represents k values from the original signal. The advantage of this method is that no values are dropped regarding the factor k [[RG75](#)].

2.10 HOUSEHOLD ENERGY DATASETS

In the previous sections, we introduced [NILM](#) with briefly explained methods that are used to detect appliance's events, while next, we give an overview of the available household data sets that can be used to evaluate the previously explained event detection techniques. Before 2011, most of the proposed approaches in the filed of [NILM](#) used their own data sets and libraries of signatures which increase the complexity

⁵ A filter is used to passes signals which has frequency lower than a previously selected cutoff value.

because each search were working independently. After the mentioned date, many data sets got published online for public use; this gives the researchers the chance to compare their results. An up-to-date overview for available data sets for NILM is available on the NILM Wiki [com].

2.10.1 Data-sets Summary

An important metric that makes the dataset usable to evaluate any NILM algorithm is having the total power demand and the corresponding sub-metered load data. In the following, a list with some characteristics of the available energy datasets is addressed. For more information about the available datasets [Par12] [BKC+14] [MSPV13] [MEE+14]. A simple comparison between the most used datasets is given in Figure A.2 in the Appendix.

- ▷ **Reference Energy Disaggregation Dataset (REED):**
This dataset is the first available dataset which was presented mainly for NILM in 2011 [KJ11]. The REED includes the total and the sub-metered load data collected from six houses. Today the REED is one of the most commonly used datasets for evaluating disaggregation algorithms [PFH+15].
- ▷ **UK Domestic Appliance-Level Electricity (UK-DALE):**
This data set includes aggregate load data 16 kHz sampling rate and appliance-level power consumption 1/6 Hz sampling rate from five houses for 655 days [KK15b]. It was released in 2014.
- ▷ **Controlled On/Off Loads Library (COOLL):**
With 100kHz downsampling rate the COOLL dataset is collected from 24 appliances of 12 types including the current and the voltage. This dataset was collected by *Primse laboratory* from the university of Orléans in France [PMR+16].
- ▷ **Worldwide Household and Industry Transient Energy (WHITED):** This dataset includes the transient start-up records of 110 different appliances from 47 different types in six different regions (Germany, Austria, and Indonesia). The measurements have been collected with a sampling rate of 44 kHz [KHKJ16].
- ▷ **BLUED:**
Released in 2012, this data set represents current measurements and voltage for a house in the USA for a week with sampling rate 12 kHz. More information about the BLUED is introduced in the Section 3.2.1.

CASE STUDY

3.1 THE FAST SHAPELET DISCOVERY

In this chapter, the proposed method to detect the event of an appliance within the time series data is introduced. The reason behind choosing the fast shapelet as event detection algorithm is that this algorithm showed a significant performance in a previous study [PPCP14]. Moreover, it was tested with BLUED energy data set, the same dataset that is used in this experiment, besides having a detailed explanation with pseudocode in its original paper, which reduces the errors in the implementation task. Unlike the FLAG shapelet [HKZ16] that was introduced in the Section 2.7.3. On the one hand, the FLAG algorithm has performed better than fast shapelet in the praxis. On the other hand, it has been decided to drop this algorithm from the study because of its high complexity in the implementation phase, which is time-consuming.

3.1.1 Overview

As we mentioned before in Section 2.7.3, the shapelets is a machine learning method for selecting efficient discriminative subsequences within time series data. (the definition of time series data can be found in the Background Section 2.1).

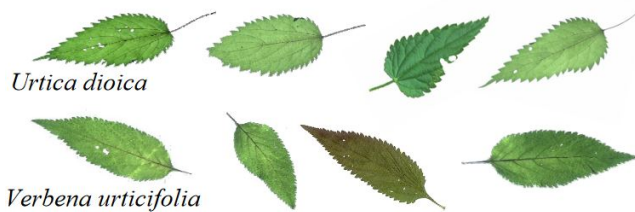


Figure 3.1: Samples of leaves from two classes [YK09]

To explain how can shapelets in some cases maximally representative of a class in a time series, in the following an example is introduced. In Figure 3.1 leaves from two different trees; *Verbena urticifolia* and *Urtica dioica* are presented as an example. The separation of both of them is difficult because of similarity in color and size within each leaf type.

Assume that these two plants need to be distinguished by building a classifier using shapelets. First, each leaf should be converted into an one-dimensional representation Time Series (TS). Then instead of making a full comparison between the entire shapes, only a small subsection (or a shapelet) of the shapes from both types that are particularly separating need to be matched.

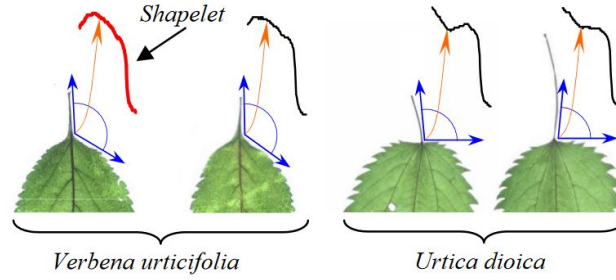


Figure 3.2: Discovered shapelet of two different types of leaves [YK09]

The Figure 3.2 illustrates the discovered shapelet by searching in the time series. The found shapelet defines a difference between the two types; the stem of *Urtica dioica* connects to the leaf at an angle of 90 degrees, In contrast, the stem of *Verbena Urticifolia* connects to the leaf at an obtuse angle [YK09].

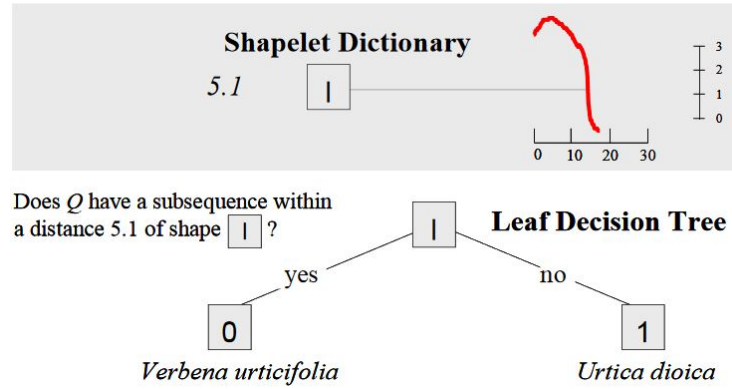


Figure 3.3: A simple Decision Tree Classifier to compare all subsequences with the discovered shapelet [YK09]

After that, the distance to the nearest matching subsequence is calculated over all sequences in the time series data. In order to build the classifier, a simple decision tree is shown in Figure 3.3 is defined. The Decision Tree Classifier (DTC) is a method from the computer science field, it is applied widely in diverse areas like radar signal classification or in data mining to create a model that forecasts the value of a target element depending on different input variables by using multistage decision making [SL91]. Since the shapelet belongs

to *Verbena Urticifolia* leaf type, all subsequences that have distance less than 5.1 are classified as that leaf type; otherwise, it is classified as *Urtica dioica*⁶ [SL91].

After explaining the general concept of the shapelets, the Fast shapelet is ready to be introduced. Fast shapelet is an alternative solution for the shapelet discovery problem with another representation. This solution is faster than the shapelets based methods; while the best-running time for proposed shapelet discovery is $O(n^2m^3)$, where m is the length of the longest sequence in time series, and n is the number of objects in the dataset, the fast shapelet achieved running time of $O(nm^2)$ [RK13].

The fundamental idea is in particular that the fast shapelet uses a transformation of the high-dimensional row data into a low-dimensional discrete representation [RK13]. Also the algorithm searches for shapelet over discrete representation, which is more efficient. In the following a description of the fast shapelets' work method. First, it builds a dictionary of SAX words for each possible shapelet length [WKX06; LKWLo7]. Subsequently, the dictionary's dimensions are reduced by masking randomly chosen letters by Random Projection (RP) [BT02]. Depending on multiple random projections, it creates a frequency count histogram for every classification group. Later a score is assigned for each SAX word based on how well these frequency counters can differentiate between groups. In the end, the n -best SAX words are calculated and mapped back to the original state (time series), which are evaluated by using the information gain.

3.1.2 Definitions and Notations

Before explaining the fast shapelet approach, necessary notation are introduced in the following paragraph [RK13].

Definition 1: T is a time series containing a list of numbers ordered by time: $T = t_1, t_2, t_3, \dots, t_p$. Every t_i can be a finite number, where p is the maximum length of time series T .

Definition 2: continues local sequence of a time series T is called subsequence S . It can be defined as $S = T_i^l = t_i, t_{i+1}, t_{i+2}, \dots, t_{i+l-1}$ where l the length of S at position i .

Definition 3: A set of pairs of TS T_i called dataset D with it own class label c_i . It can be written as

⁶ In case of Binary Classifier.

$D = \langle T_1, c_1 \rangle, \langle T_2, c_2 \rangle, \langle T_3, c_3 \rangle, \dots, \langle T_n, c_n \rangle$, Where n is the number of T inside each D . the length of each TS should not be equal. To estimate the similarity among subsequences, the distance between two subsequences is addressed.

Definition 4: The distance between two subsequences S and \hat{S} is the Euclidean length distance between S and \hat{S} . If both S and \hat{S} are Z-normalized with $std = 1$ and $mean = 0$. The distance is

$$\text{dist}(S, \hat{S}) = \sqrt{\frac{1}{l} \sum_{i=1}^l (S_i - \hat{S})^2}$$

from the definition the shapelet can have the maximum length of m . Note that in the case of comparison between two shapelets with different lengths, the length normalization should be used.

Definition 5: The minimum distance between S and any subsequence of T that has the same length as S formally called the distance between time series and subsequence of length l . $\text{dist}(S, T) = \min_i \text{dist}(S, T_i^l)$ assume that the dataset D includes n time series with c classes. So the quantity of time series in class i is n_i . The class probability is described as $p_i = n_i/n$.

Definition 6: The entropy of a dataset D can be written as

$$E(D) = \sum_{i=1}^c p_i \log(p_i)$$

A split has been defined, where the dataset can be separated into two smaller datasets:

Definition 7: A split can be described as a tuple $\langle s, d \rangle$ of a distance threshold d and subsequence S , which can divide the dataset into two smaller datasets D_R and D_L , where n_R and n_L are the number of TS in D_R and D_L .

Definition 8: Every Split has an information gain which can be formally written as

$$I(sp) = E(D) - \frac{n_L}{n} E(D_L) - \frac{n_R}{n} E(D_R)$$

Where sp is a split.

Definition 9: The separation gap of a split sp can be defined as:

$$\text{gap}(sp) = \frac{1}{n_L} \sum_{t_L \in D_L} \text{dist}(s, t_L) - \frac{1}{n_R} \sum_{t_R \in D_R} \text{dist}(s, t_R)$$

Definition 10: The shapelet is a split that divides a dataset D into two smaller datasets D_L and D_R with the maximum information gain.

After explaining the necessary notations, the fast shapelet's work method is demonstrated in the following section.

3.1.3 Generating SAX Words

As discussed before, the first step to find a unique shapelet that can represent its class is the transformation of the given time series into a symbolic representation by using the SAX method (more about SAX can be found in the literature [WKX06; LKWLo7]). Back to the previously example in the overview section, a time series from *Verbena urticifolia* leaf type is illustrated by Figure 3.4.

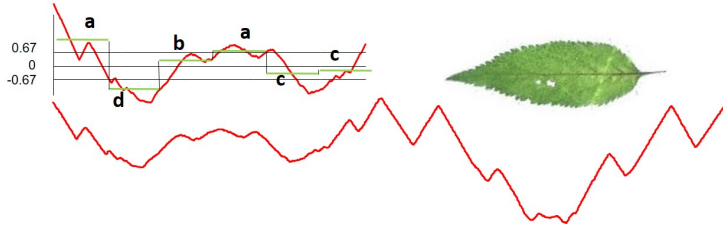


Figure 3.4: A SAX word generated from a subsequence of a dataset [RK13]

At the top part of the figure is an example of a SAX word (adbacc) generated by a subsequence of the time series. The SAX performs two stages to transform the time series into strings. First, It converts a given sequence into Piecewise Aggregate Approximation (PAA)⁷ representation. Then in the second stage, it transforms the PAA resulting data into a string by using lookup table for the cut lines coordinates (in the figure the values of the lookup table for three letters alphabet are in the top part left). To generate a SAX word for the whole given time series the sliding window technique is used. Note that a single time series generates multiple SAX words [RK13].

3.1.4 Random Masking

After having the SAX representation of the raw data, let's assume that the shapelet in this representation can be searched. It would be faster than searching in the raw data, but a problem would be faced in this case; namely the *false dismissals* problem. the *false dismissals* caused by two sequences, which have a little difference, but they generate

⁷ PAA approximates the time series data into a vector.

completely two different SAX representations. So, the best shapelet in the time series could map to two different SAX words, but in time series space, there is hardly any difference between them [RK13]. e.g., the words **a**ba**c**d and a**b**acd. To solve this problem the Random Projection is used. This idea comes from the bioinformatics field [BT02]. The key idea is to transform the high dimensionality SAX words into a smaller dimensionality. The Figure 3.5 shows the masking of SAX words; a transformation of SAX words that have dimensionality of five is randomly masked at two different positions, decreasing the dimensionality from five to three [RK13]. Using the random projection decreases the odds of facing the false dismissals problem, so event if two words have a small difference in the SAX dimension, they are counted as one object through the masking process.

	SAX Words	1 st Random Mask	2 nd Random Mask																																													
Obj 1	<table><tr><td>a</td><td>d</td><td>b</td><td>a</td><td>c</td></tr><tr><td>a</td><td>c</td><td>a</td><td>a</td><td>c</td></tr></table>	a	d	b	a	c	a	c	a	a	c	<table><tr><td>a</td><td>d</td><td>b</td><td>a</td><td>c</td></tr><tr><td>a</td><td>c</td><td>a</td><td>a</td><td>c</td></tr></table>	a	d	b	a	c	a	c	a	a	c	<table><tr><td>a</td><td>d</td><td>b</td><td>a</td><td>c</td></tr><tr><td>a</td><td>c</td><td>a</td><td>a</td><td>c</td></tr></table>	a	d	b	a	c	a	c	a	a	c															
a	d	b	a	c																																												
a	c	a	a	c																																												
a	d	b	a	c																																												
a	c	a	a	c																																												
a	d	b	a	c																																												
a	c	a	a	c																																												
Obj 2	<table><tr><td>a</td><td>c</td><td>b</td><td>a</td><td>c</td></tr><tr><td>b</td><td>c</td><td>c</td><td>c</td><td>d</td></tr><tr><td>b</td><td>d</td><td>c</td><td>d</td><td>d</td></tr></table>	a	c	b	a	c	b	c	c	c	d	b	d	c	d	d	<table><tr><td>a</td><td>c</td><td>b</td><td>a</td><td>c</td></tr><tr><td>b</td><td>c</td><td>c</td><td>c</td><td>d</td></tr><tr><td>b</td><td>d</td><td>c</td><td>d</td><td>d</td></tr></table>	a	c	b	a	c	b	c	c	c	d	b	d	c	d	d	<table><tr><td>a</td><td>c</td><td>b</td><td>a</td><td>c</td></tr><tr><td>b</td><td>c</td><td>c</td><td>c</td><td>d</td></tr><tr><td>b</td><td>d</td><td>c</td><td>d</td><td>d</td></tr></table>	a	c	b	a	c	b	c	c	c	d	b	d	c	d	d
a	c	b	a	c																																												
b	c	c	c	d																																												
b	d	c	d	d																																												
a	c	b	a	c																																												
b	c	c	c	d																																												
b	d	c	d	d																																												
a	c	b	a	c																																												
b	c	c	c	d																																												
b	d	c	d	d																																												
Obj 3	<table><tr><td>b</td><td>b</td><td>a</td><td>c</td><td>d</td></tr><tr><td>d</td><td>c</td><td>a</td><td>a</td><td>c</td></tr></table>	b	b	a	c	d	d	c	a	a	c	<table><tr><td>b</td><td>b</td><td>a</td><td>c</td><td>d</td></tr><tr><td>d</td><td>c</td><td>a</td><td>a</td><td>c</td></tr></table>	b	b	a	c	d	d	c	a	a	c	<table><tr><td>b</td><td>b</td><td>a</td><td>c</td><td>d</td></tr><tr><td>d</td><td>c</td><td>a</td><td>a</td><td>c</td></tr></table>	b	b	a	c	d	d	c	a	a	c															
b	b	a	c	d																																												
d	c	a	a	c																																												
b	b	a	c	d																																												
d	c	a	a	c																																												
b	b	a	c	d																																												
d	c	a	a	c																																												

Figure 3.5: Generated SAX words of every object. At the right side, the SAX words after masking two symbols at two masking positions which were randomly picked. Note that the random masking makes the both words **a**dbac and **a**cbac identical. [RK13]

3.1.5 Counting Similar Objects

Instead of making a full comparison of all distances between all objects, using hashing on all candidates builds a collision table with counters which determine how many time two similar objects are shown in the RP, even if there is a small difference between them. The Figure 3.6 shows two iteration of hashing of SAX words we created earlier. The reader can realize that the words **a**bbac and **a**cbac have the same signature ******bac after masking. The signatures are generated after making a mask on a specific position. So all the SAX word, even if they have small differences but having the signature bac increase their counters in the collision table A (at the right side).

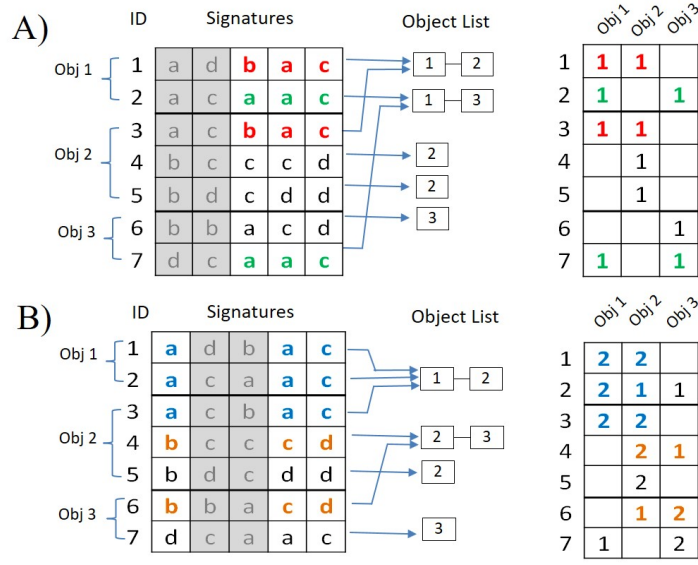


Figure 3.6: Two hashing iterations A and B are used to count the occurrence of **SAX** words with the same signature (right side). The signatures are generated after making a mask on a specific position (marked symbol in table on the left). [RK13]

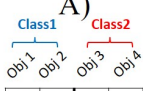
3.1.6 Finding The Best Candidates

Back to the leaves example, after r iterations of random projection we get the final collision table. Figure 3.7. Table A shows a final collision table; for example the **SAX** word number one is counted in object one five times, while it is not found in the object three or four. It means that this word can efficiently represent class1. From the final collision table two new tables are build; the first, Table B includes the sum of objects-based counts with class-based counts, and the second table (table C) represents the complementary data of table B. The purpose for using a complementing table is to determine the word that frequently appears in a class, and does not appear in the other class, this word refers to the best shapelet. By applying a single equation (table D), the distinguishing score of every object is calculated. In this example, word one has the highest distinguishing score 20 because it occurred in class1 10 times, and it is far from the objects in class2. Note that when a word has the same number of repetition in both class1 and class2 (word5 in the table C) a distinguishing score of zero is assigned, this means this word has an equally frequent pattern in the both classes.

After having the list of the high distinguishing scores, the best words are converted back to the original data form. The information gain and the separation gap are calculated in order to determine the best

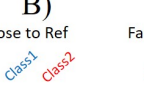
shapelet. Regarding Definition10 the best shapelet is the one with maximum information gain, so it can divide the given dataset into smaller parts.

A)




B)

Close to Ref



C)

Far from Ref



D)

Distinguishing Power

1	5	5				1	10	0				1	0	10				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0				1	10	0
---	---	---	--	--	--	---	----	---	--	--	--	---	---	----	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---	--	--	--	---	----	---

Figure 3.7: Table A the final collision table of all SAX words after 5 iterations. Table B contains the sum of all scores of objects from the same class. Table C includes the complementary data of table B. the table D has the distinguishing score of each SAX word [RK13].

In the following section, the explained steps of discovering a shapelet are formalized by a sample pseudocode.

3.1.7 Algorithm Implementation

The proposed shapelet algorithm is given in Listing 3.1 as pseudocode. The fast shapelet based algorithm is implemented by using MathWorks MATLAB 2018R as a programming platform and a programming language. MATLAB can use external compilers to call functions that are already written in other programming languages like C++ and Java. Since the open-source library for the shapelet classifier was written in C++ [TR], so importing these libraries into a MATLAB work-space is possible.

As shown in the pseudocode, the algorithm is divided into three phases. The first phase is covered by the lines (3 – 10). The potential subsequences are selected after a search in the SAX representation-based data. The second phase is the same method which has been detailed in the previous sections. It measures the quality of the selected candidates and delivers the best subsequence as the best shapelet. The last phase, where a decision tree is created based on the distance of the final shapelet Definition5. In order to choose the candidates, the algorithm selects all subsequences of length len from the time series data, which are generated by using the sliding window technique, then their corresponding SAX word is generated and stored in a SAXList.

Listing 3.1: Fast shapelet based classifier pseudocode [RK13]

<i>Algorithm: Fast Shapelet based Classifier</i>	
Input: D : Time series dataset contains class labels r : number of random projection iterations k : number of SAX word as candidates	
Ouput: A Shapelet based Classifier	
01	[TS,Label] = LoadRowData(D)
02	for len = 1 to m
03	SAXList = CreatSaxlist (TS , len)
04	Score = { }
05	for i = 1 to r
06	Count = RandProjection (SAXList , TS)
07	Score = UpdateScore (Score , Count)
08	end for
09	SAXCand = FindBestSAX (SAXList , Score , k , r)
10	TSCand = Remap(SAXCand , TS)
11	for i = 0 to TSCand
12	Cand = TSCand [i]
13	DList = NearstNeighbor (TS , Cand)
14	[gain , gap] = CallInfoGain (TSCand)
15	if (gain > max_gain)
16	((gain == max_gain) && (gap > min_gap)) then
17	max_gain = gain
18	min_gap = gap
19	FinalShapelet = cand
20	end if
21	end for
22	{ T ₁ , T ₂ } = SplitData (T , FinalShapelet)
23	if ¬ isLeaf (T ₁) then
24	buildClassifierFS (T ₁)
25	if ¬ isLeaf (T ₂) then
26	buildClassifierFS (T ₂)
27	end for

The lines (3 – 10) describe the scoring technique of the SAX words by using the RP and generating the hash signature for every SAX word. Then the top k SAX words that have the highest distinguishing score are picked from the list and remapped back to the raw time series. The parameter r and k represent the number of iterations and the initial candidates respectively.

The lines (13 – 21) represent the second phase where the information gain Definition8 and the distance between the subsequences in the list and each time series are calculated. The best one that has more significant information gain is the final shapelet Definition10.

3.2 DATA STRUCTURE

In the previous section, the proposed algorithm for event detection and its work method were presented. While Next, the energy household dataset is introduced which is used to evaluate the implemented algorithm. Furthermore, the techniques that are applied to prepare the input data to fit the training and the testing phase are mentioned including the parameter selection and the experimental tasks.

3.2.1 Data Source: BLUED

To evaluate our algorithm the **BLUED** dataset is used. The **BLUED** includes one week of the current and the voltage measurements for a family house in Pennsylvania, Pittsburgh. The data collection were taken in October 2011 from nearly 50 electrical devices in the house [Fil11]. The dataset presents the changes in the operating status of each appliance (turn on/off). The measurements were taken by using two separate systems; one is at the main distribution panel to capture the current and voltage, and the other system is used to register the time stamps for each event. The reason behind choosing the **BLUED** is that this dataset in its original form provides raw data of the current and the voltage. Moreover, it is the de facto dataset for the benchmarks; unlike all the other available energy datasets, it is annotated with the event time stamp and the triggering appliances. Many event detection approaches previously used the **BLUED** dataset to evaluate the detection performance [ABO+12; RK13]. A table listing the appliances in the **BLUED** is given in the appendix.

Because of having 3-wire in the residential buildings in USA, where the dataset is collected, the measurements are collected in two process parallel with a difference of 180 degrees. For this reason, the **BLUED** has two phases A and B [Fil11]. The algorithm is evaluated with both of them separately. The dataset is divided into 16 compressed files. Each file has two collection sampling rate of data; 12 kHz data stored in text files and 60 Hz data stored in mat files (MATLAB files). Each file represents the real and the reactive power for both; phase A and phase B for approximately 10.5 h of the load data. For the experiments the real power values from both phases A and B are used (refer to the Section 2.1 for the definitions of the reactive power and the active power).

The Figure 3.8 illustrates a part of the raw data from phase A containing 44 h of the real power (several extreme values have been dropped in order to keep the scale). This part includes more than 200

events over 9 million measurements representing multiple appliance status (on and off). The red asterisk represents the timestamp of these events in the load data. Furthermore, in the source paper of the [BLUED](#) the writer defined the event to be any change in the power consumption higher than 30 Watts at least for 5 s [[PPCP14](#)].

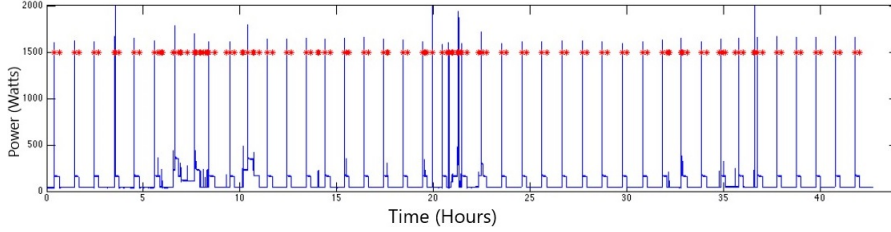


Figure 3.8: The consumption data for 44 h (the first quarter) from Phase A in the [BLUED](#). The red asterisk represents the locations of the events (on and off) [[PPCP14](#)]

3.2.2 Data pre-processing

Since a supervised machine learning based approach is used, this method requires training phase which requires in its turn a training dataset (also known as training instances) for classifying the new unseen data. Each one of the segments should have its ground truth (the label), so due to the lack of the labeled instances in the [BLUED](#), the processing of the raw data into labeled instances to fit the algorithm is needed. Therefore a window of 4 seconds duration is defined. This duration is chosen because of the hint in the [BLUED](#) paper [[Fil11](#)] which defines an event with duration at least of 5 s, so in case of an event of duration for 5 s, this event contains an on/off event. Having fewer seconds window ensures not having any overlaps events in the same segment. However, the duration of this window can be varied to get different instances according to the requirements.

All the extracted windows include the data values within 4 s. A second contains 60 data values, so the length of a segment is 240 values. The training segments with their labels represent the events of different appliances. Since the [BLUED](#) has two phases, and every phase includes different devices, two training sets for each phase A and B are extracted. Later the experiment is carried on each one separately. Moreover, every segment contains an operation of an appliance and dose not has an on or off event is labeled as a non-event segment. The [Figure 3.9](#) illustrates four samples of the training segments of the both classes event and non-event from the both phases A and B.

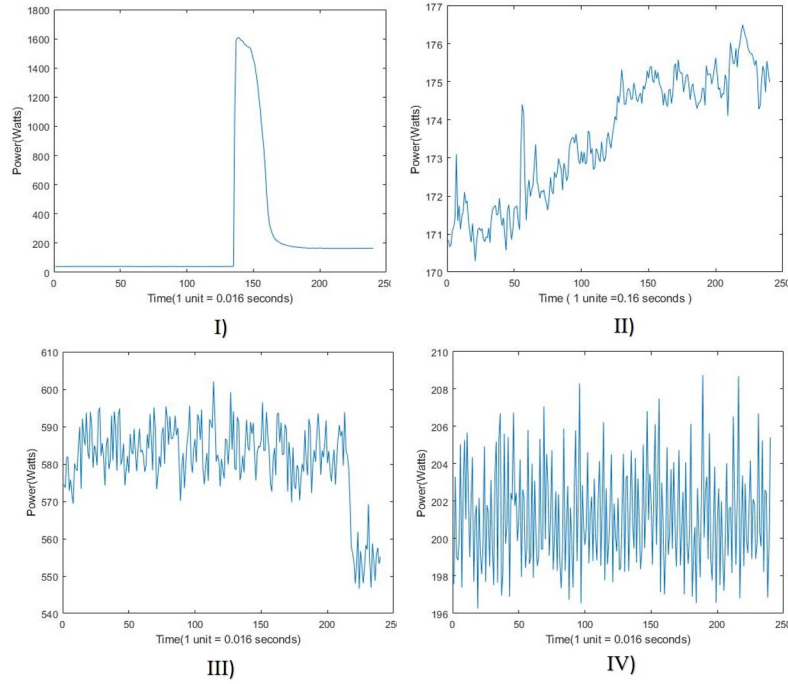


Figure 3.9: **Four samples of training Instances; figure I and II represent respectively an event and non-event segments from the phase A, while III and IV draw an event and non-event segments from the phase B.**

The reader can quickly realize from the last figure that the phase B has more noise in comparison to the phase A, which has clean signals and less noise. This fact is an addition reason to separate both phases in this experiment, so the results later represent two cases; the event detection with noisy data and with clean data.

According to the previous requirements, an algorithm is implemented to read the **BLUED** raw data as input and process it in order to deliver a set of segments with their corresponding labels ready for the training and testing phases. The testing phase is the phase where the trained algorithm is evaluated by feeding it with new unseen data without providing the corresponding ground truth. Due to the size of the raw data, all the values for the whole week can not be processed (Time-consuming), so it has been decided to apply three filters on the segments collecting process (sacrificing the quality to improve the performance).

The **first filter** splits the raw data into two parts. The both parts are used to generate the training and the testing segments. The splitting follows the rule of 60/40. This rule means 60% of the raw data is

for the training, and the rest 40 % is for the testing. The **BLUED** has precisely 170 h of data as time series, so according to 60/40 rule, the training phase consist of 105 h date and 65 h are for the test phase. This filter is not enough to reduce the massive size of the raw data, especially for the training phase, so a **second filter** is applied, this filter skips an hour during the collecting process. For example, from 10 h of data, the algorithm collects every other hour (5 h of the data). Skipping an hour is made by creating a data-collection loop which is increased by one hour every iteration. This filter guaranties covering different events from different appliances because the events are collected from different days without a massive increase in the segments number.

By the collecting process, another problem needs to be solved, namely, the imbalanced data problem (Section 2.8.6). To explain the consequences of this problem, let us assume collecting the data from the **BLUED** for 2 minutes with a window size of 4 seconds. The data represents the consumption measurements of two devices. A laptop and a room light were turned on once during the mentioned time. After the collecting process completes, the algorithm obtains 30 segments. The segments contain 2 events and 28 non-events, so the class of non-event has more effect on the results later. This difference between the both classes can be massively increased by collecting hours of the data. Therefore, one **last filter** was applied to avoid this problem. This filter ensures the collecting of data according to the following manner 1:2 ratio of events:non-events. In the previous example, after applying the collecting ratio, the algorithm obtains 6 segments, which contains 2 events and 4 non-events.

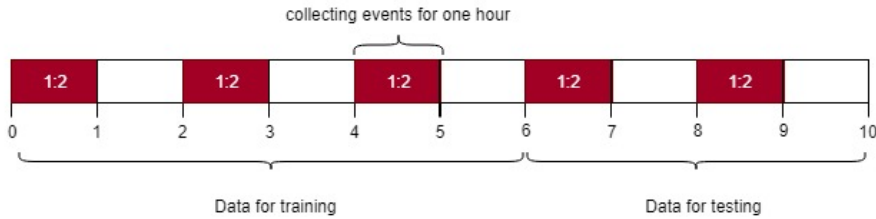


Figure 3.10: **Collecting segments for the testing and the training phases from 10 h of data by 1:2 ratio of events:non-events. The highlighted spaces are the hours where the data are collected.**

The Figure 3.10 explains the filters by applying them on an example. The highlighted spaces are the hours where the data are collected. As already shown in the figure, the 10-hours of data are split into a training set and testing set according to the 60-40 rule and 5 h of the data were skipped. The rest 5 h are collected by 1:2 ratio of events to non-events.

Using the technique of the fixed length window has disadvantages like the case of occurring two events (on/off) in the same segment. Such segments are eliminated during collecting the data for the training phase. This can ensure training the algorithm with clean data which has at most one event (on/off) in each segment. Having more than one event in the training segment can affect the outcome of the shapelet discover process. e.g., An on-event which represents a laptop generates a different SAX word when it is combined with another event in the same segment.

The elimination of the multi-event segments has a noticeable effect on the collecting process. A real example is shown in the Figure 3.11 from the phase B, where a printer generates a series of events. More than 6 events occurred in 15 s. In such cases, the algorithm eliminates about 45 events. The elimination affects the chance to cover the appliances which generate such events.

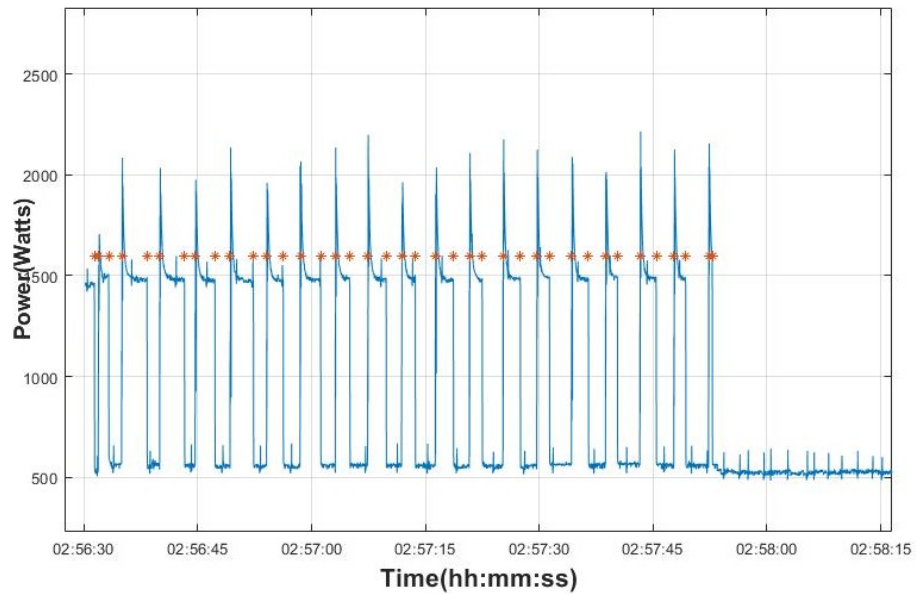


Figure 3.11: A series of events caused by a printer from the phase B. The red asterisk represents the events time stamp.

Now after explaining the fixed length window technique, the covering percent of the data pre-processing phase from the BLUED can be presented. The Figure 3.12 illustrates the statistical results of the data collecting process in comparison to all events that the BLUED includes.

The used method can collect more than 60 % of the total events in the phase A, while it achieved nearly 50 % of the total events of the phase B. The phase B has appliances which create many series of events in small period (smaller than the length of the defined window), so because

of the elimination there is a covering difference between both phases A and B. In total, the algorithm collected nearly 60 % of the overall events in the both phases. For collecting more events, there are two proposed options, defining a window with a small fixed length (can affect the downsampling stage) or abandon the second filter which were explained previously, so the algorithm do not skip hours and collect the segments from the raw data continuously. This option can cause a massive increase in the training dataset which consumes more time in the training phase.

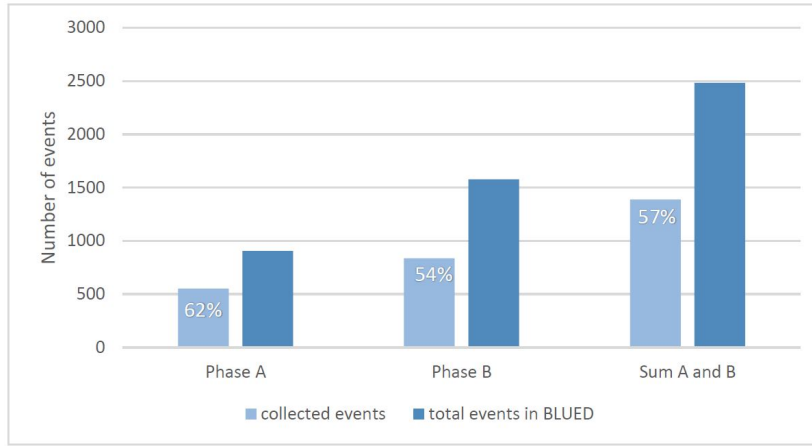


Figure 3.12: The statistical results illustrates the covering percent of the collected events by the fixed window technique in comparison to overall events in BLUED.

During the implementation, the processing of the data has been made to be as dynamic as possible. For this purpose, many parameters are defined which can control the collecting data process, such as the length of the window, the original data sampling rate, the collecting ratio. On the one hand, all these parameters can be changed to meet the requirements for the future needs. On the other hand another dynamic parameters were implemented to fit the given dataset without the need to set them manually, this includes the number of the classes and the number of the training segments. In the implementation, the class label 1 refers to the event instances, and the class label 2 refers to the non-event instances.

3.2.3 Sampling Frequency

In the last section, we explained the pre-processing data stage where the raw data from the BLUED files is read, filtered, and transformed into segments with their corresponding ground truth (labels) in order to fit the training and the testing phases. Moreover, the disadvantages

of using the fixed length window technique to process the raw data are explained.

In this section, the downsampling method which is applied to the collected segments is discussed. Previously in the [Section 2.9](#), two different methods were introduced to decrease the sampling rate of a data sequence; the downsampling by an integer factor, and the downsampling by averaging. In this experiment, the second method is used. The downsampling by averaging is chosen, because this method does not drop any value from the original segment and keeps the effect of the values by calculating the mathematical average. The range [1,32] contains the proposed downsampling factors⁸. This means a block of the length n is averaged where $n \in [1,32]$. Note that the maximum value of the downsampling factor depends on the length of the training segment. Therefore, the expansion of the range is not possible without changing the length of the pre-defined window.

3.3 MODEL TRAINING

After loading the [BLUED](#) raw files and processing them into labeled segments, training the model with these segments can be started. Later a decision tree for the classification task is generated. The last step before starting the training is setting the parameters of the model. As mentioned previously in the [Section 3.1.7](#), a set of parameters should be set to control the shapelet searching process; the number of the random projection's iterations r , the number of [SAX](#) word as candidates k , the maximum and the minimum length of the final shapelet, and the step of the search. These parameters are set to their default values as proposed in the source paper of the fast shapelet [[RK13](#)]; $K = 10$, $r = 10$, the search step = 10.



Figure 3.13: **Block diagram of the proposed stages to train the model.**

After the pre-processing stage, the [Figure 3.13](#) shows the proposed stages to train the model, starting by setting the downsampling rate of the segments then training the model, ending by the testing phase and collecting the results. These steps are repeated with different downsampling rates for every iteration.

⁸ Factor 1 means no downsampling occurs.

In the pre-processing phase, the algorithm stores the training and testing instances with their labels locally as an external data file, so the task of collection and processing the data occurs one time, unless requirements changes, such as increasing the size of the window or changing the dataset. When the training phase is finished, the algorithm generates a text file with the information of the built decision tree. The tree size includes the levels of the tree as well as the information about the found shapelet. Moreover the generated file includes the consumed training time and the used downsampling factor.

In the testing phase the generated file is loaded, then the algorithm assigns a label for each test instance depending on the calculated distance of each instance to the selected shapelet (refer to the [Section 3.1.1](#) for more information). These steps are reapplied on all test segments. In the end, the algorithm counts the correct and the wrong predictions according to the ground truth values. The final results include the classification accuracy and the confusion matrix (the both metrics are explained in the [Section 2.8](#)). The next chapter presents the results and the evaluation of the experiments for both phases A and B.

EMPIRICAL ANALYSIS

4.1 EVENT DETECTION RESULTS AND DISCUSSION

In the previous chapters the fast shapelet algorithm for NILM event detection is introduced. Earlier, the suggested metrics for evaluating the performance in the Section 2.8 are explained, and the used downsampling method in the Section 2.9 is discussed as well as the training steps in the Section 3.3. The next step is evaluating the performance of this method after feeding it with data downsampled at different rates.

In this chapter, the results of the experiments are obtained. As mentioned before, the experiments are executed on the both phases A and B separately due to the different nature of the data, and the fact that the phase B is noisier than the phase A (refer to Section 3.2.2 for more explanation). Therefore the results are split into two sections; results of the event detection on the phase A and results of the event detection on the phase B.

4.1.1 Results of the phase A

In the following the detailed results are outlined, starting with the general results in the Table 4.1 which obtained from testing the algorithm with the non-downsampled data and the downsampled data by a factor in the range from 2 to 32:

Table 4.1: Results of the event detection on the phase A appliances.

Downsampling Factor	Accuracy	F1-Score
Factor 1	94.58 %	0.92
Factor 2	93.92 %	0.91
Factor 3	90.78 %	0.87
Factor 4	85.57 %	0.77
Factor 5	87.85 %	0.80
Factor 6	90.13 %	0.85
Factor 7	85.03 %	0.76

Continued on next page

Table 4.1 – Continuation

DOWNSAMPLING FACTOR	ACCURACY	F1-SCORE
Factor 8	94.57 %	0.92
Factor 9	96.20 %	0.94
Factor 11	93.92 %	0.91
Factor 12	92.19 %	0.88
Factor 13	92.36 %	0.88
Factor 14	95.02 %	0.92
Factor 15	91.54 %	0.87
Factor 16	84.81 %	0.74
Factor 17	90.88 %	0.86
Factor 18	83.94 %	0.72
Factor 19	81.45 %	0.66
Factor 20	76.46 %	0.55
Factor 21	95.87 %	0.94
Factor 22	87.09 %	0.80
Factor 23	83.94 %	0.77
Factor 24	83.51 %	0.71
Factor 25	86.33 %	0.79
Factor 26	88.82 %	0.82
Factor 27	81.88 %	0.67
Factor 28	83.08 %	0.69
Factor 29	86.00 %	0.76
Factor 30	91.54 %	0.87
Factor 31	84.05 %	0.72
Factor 32	92.62 %	0.89

The previous table outlines the accuracy and the F1-score of the event detection on the downsampled data by all the pre-defined downsampling factors. Two facts can be extracted from the general results. The first fact, the algorithm performed well at the most of the downsampling factors. It has scored the best accuracy and the best F1-score (97 %, 0.94) by factors 9 and 10. The worst accuracy was obtained at factor 20, 67% accuracy and 0,55 F1-score. The second fact, the results in general have no recognizable pattern.

To evaluate the performance of the algorithm precisely and analyse the impact of the downsampling rate on the event detection process,

two cases are picked up from the previous table, the first one where the algorithm has been feed with the non-downsampled data, and the second one, where the algorithm performed with the downsampled data better than with the non-downsampled data. The Table 4.2 shows the results of the first case. The algorithms has showed a significant performance on the non-downsampled data as it has been proven in another related work [PPCP14]. It scored approximately 95 % accuracy and 0.92 F1-score. The false positive rate is 0 because of detecting all the non-events correctly.

Table 4.2: Results of the event detection on the phase A appliances with the non-downsampled instances.

Accuracy	F1-Score	Confusion Matrix		
		real	event	non-event
94.58 %	0.92	pred event	278	0
		pred non-event	50	585

By taking a look at the Figure 4.1, four shapelets were obtained. Three shapelets represent the on-events, and one represents the off-events. In general, all the found shapelets can represent the both classes well.

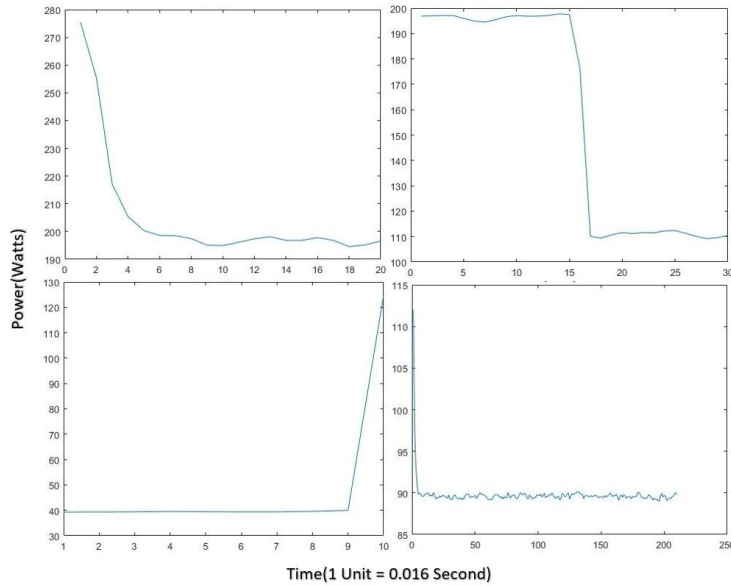


Figure 4.1: The found shapelets in the non-downsampled instances from the phase A.

In contrast, 50 segments including events have been detected as non-events (false negative segments). The Figure 4.2 shows two randomly picked up samples from the false negative segments.

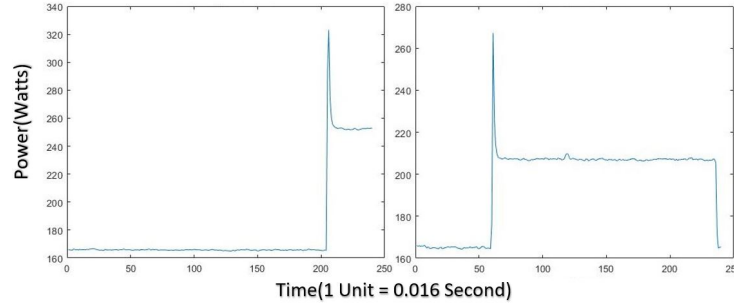


Figure 4.2: Two randomly picked up samples from the false negative segments of the non-downsampled data from the phase A.

With a quick comparison to the obtained shapelets in Figure 4.1, the reason behind not predicting such events correctly can be defined. These false negative segments have a big spike⁹ which is not exist in the obtained shapelets. This assumption can be confirmed by analysing the results of the second case, where the algorithm performed with the downsampled data by factor 10 better than with the non-downsampled data. As seen in the Table 4.3 the algorithm achieved approximately 97% accuracy and 0.94 F1-score. The confusion matrix shows that the algorithm has lost its zero false positives, but in return, the number of the false negative segments is decreased from 50 segments to 32 segments.

Table 4.3: Results of the event detection with the downsampled instances by the factor 10 from the phase A.

Accuracy	F1-Score	Confusion Matrix		
		real	event	non-event
96.31 %	0.94	pred event	305	2
		pred non-event	32	583

The Figure 4.3 proves the assumption by illustrating two randomly picked up samples from the true positive segments. Because of downsampling the data the size of the spikes decreased, which allowed the algorithm to detect more events correctly. The Figure 4.4 shows the obtained shapelets from the downsampled data by the factor 10. The found shapelets fit the events well and became smoother than the shapelets which were obtained from the non-downsampled data.

⁹ A spike is a short duration, fast electrical transient.

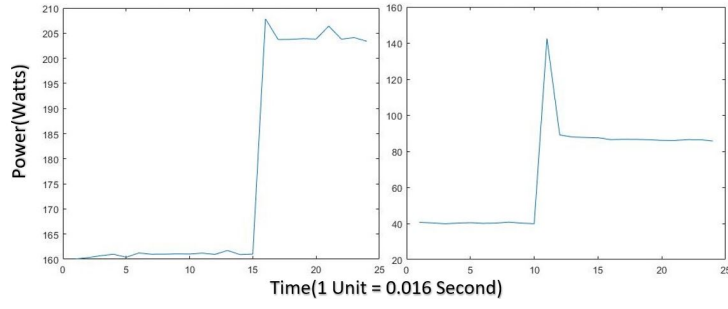


Figure 4.3: Two randomly picked up samples from the true positive instances of the downsampled data by the factor 10 from the phase A.

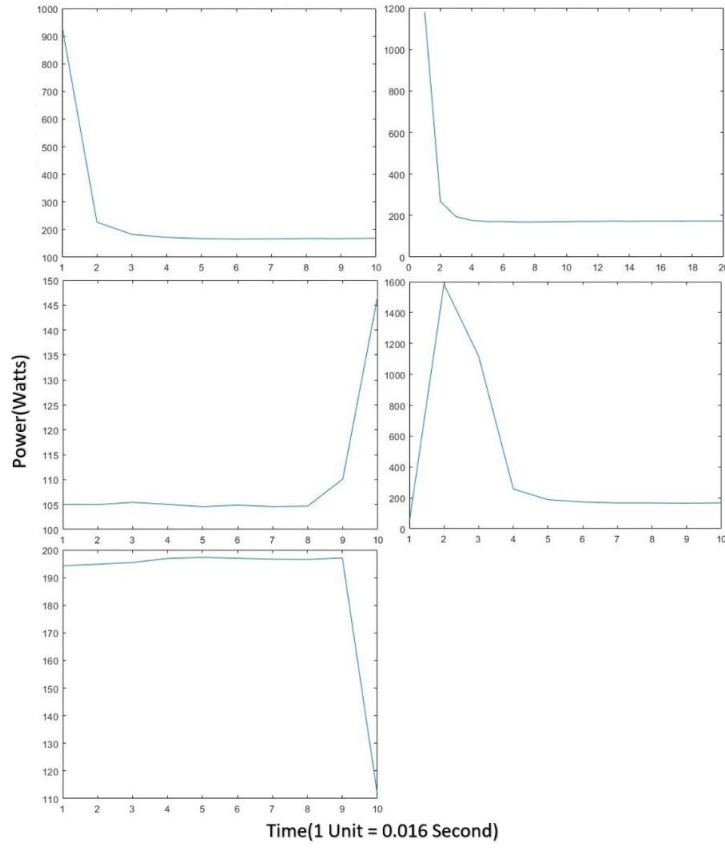


Figure 4.4: The found shapelets in the downsampled instances by factor 10 from the phase A.

The algorithm scored the best results with 97% accuracy and 0.94 F1-score by the factors 9 and 10 because of decreasing the spike size after the downsampling.

4.1.2 *Results of the phase B*

Discussing the results of the phase B starts with showing the overall results, which collected from testing the algorithm with the non-downsampled data and the downsampled data by a factor in the range from 2 to 32:

Table 4.4: Results for event detection on phase B appliances .

DOWNSAMPLING FACTOR	ACCURACY	F1-SCORE
Factor 1	89.54 %	0.85
Factor 2	88.94 %	0.84
Factor 3	88.54 %	0.83
Factor 4	88.45 %	0.83
Factor 5	88.76 %	0.84
Factor 6	88.60 %	0.83
Factor 7	87.78 %	0.82
Factor 8	88.05 %	0.92
Factor 9	78.38 %	0.81
Factor 10	86.54 %	0.79
Factor 11	86.09 %	0.78
Factor 12	85.62 %	0.77
Factor 13	84.91 %	0.76
Factor 14	84.89 %	0.76
Factor 15	84.56 %	0.76
Factor 16	84.81 %	0.75
Factor 17	83.89 %	0.75
Factor 19	81.89 %	0.70
Factor 20	81.09 %	0.68
Factor 21	78.62 %	0.64
Factor 22	78.75 %	0.62
Factor 23	77.93 %	0.61
Factor 24	78.24 %	0.62
Factor 25	78.93 %	0.64
Factor 26	80.04 %	0.64
Factor 27	79.79 %	0.66
Factor 28	79.33 %	0.65

Continued on next page

Table 4.4 – Continuation

Downsampling Factor	Accuracy	F1-Score
Factor 29	80.46 %	0.68
Factor 30	79.04 %	0.65
Factor 31	78.86 %	0.66
Factor 32	76.91 %	0.58

The general results are listed in the in the [Table 4.4](#), which includes the classification accuracy and the F1-score of the event detection on the downsampled data by the downsampling factors from the range [1,32]. Two facts can be observed from the table. First, in general the algorithm performed under the results average of the clean data from the phase A. Because the phase B is noisy, the results in general are less than the results of the phase A. The other fact can be observed by reviewing the accuracy and the F1-score of all the downsampling factors. In contrast to the phase A, the results of the phase B have a recognizable pattern. The higher the factor, the less the accuracy and the F1-score. Both metrics decrease according to the downsampling factor.

To achieve a precisely understanding, the reason behind the constantly decreasing accuracy need to be defined. Therefore, two cases are picked up from the table. The first case where the algorithm has been tested with the non-downsampled data, and the other case where the algorithm has performed obviously less than the first case. The result of the downsampled data by the factor 18 has been chosen.

Table 4.5: Results of the event detection on the phase B appliances with the non-downsampled instances.

Accuracy	F1-Score	Confusion Matrix		
89.54 %	0.85	real	event	non-event
		pred event	273	28
		pred non-event	66	532

The [Table 4.1.2](#) shows the classification accuracy and F1-Score beside a confusion matrix for event detection in non-downsampled segments (the first case). Less than 10 % difference can be observed between the results of the both phases A and B in case of the testing with the non-downsampled data. The algorithm achieved approximately 90 % accuracy and 0.85 F1-score. The number of the false negative is increased in comparison to the results of the phase A. As mentioned before, the phase B is noisy. This fact is represented not only by the

samples that are shown earlier in the [Section 3.2.1](#), but also from the found shapelets in the [Figure 4.5](#). Many non-relevant small and big spikes exist before and after the event.

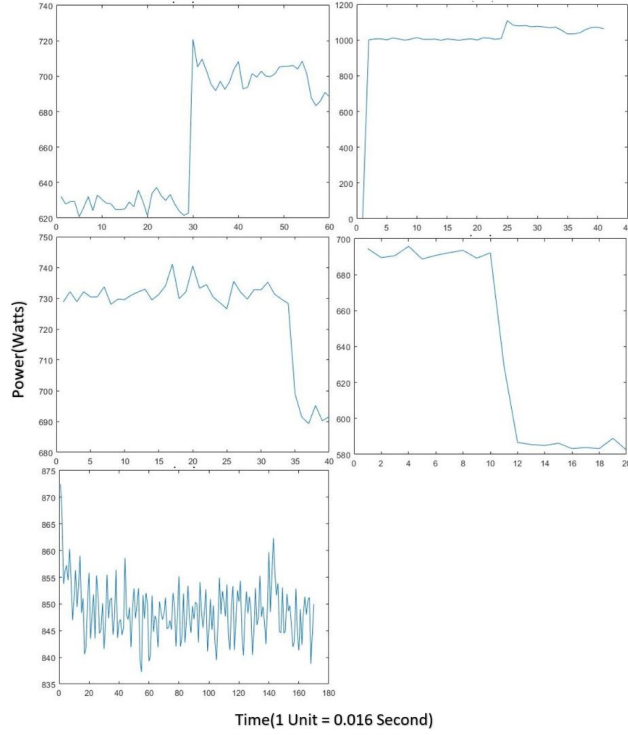


Figure 4.5: The found shapelets in the non-downsampled instances from the phase B.

By comparing the extracted shapelets with two randomly picked up samples from the true positive segments of the non-downsampled data in the [Figure 4.6](#), an observation can be taken. The algorithm could in average handle the noisy data of the non-downsampled data, as 273 of the 339 events were correctly classified, as can be seen in the confusion matrix in the [Table 4.1.2](#).

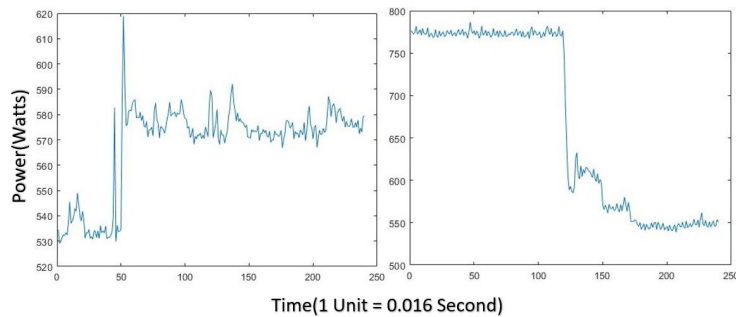


Figure 4.6: Two randomly picked up samples from the true positive segments of the non-downsampled data from the phase B.

The Table 4.6 illustrates the results of the event detection on the downsampled data by the factor18 (the second case). 10% accuracy is the difference between the both studied cases plus a significant decrease in the true positive segments ratio.

Table 4.6: Results of the event detection with the downsampled instances by the factor 18 from phase B.

Accuracy	F1-Score	Confusion Matrix		
		real	event	non-event
		pred	event	non-event
82.75 %	0.74	pred	event	215
		pred	non-event	124
				529

Including the obtained shapelets and two randomly picked up samples from the false negative segments of the downsampled data by the factor 18, The Figure 4.7, and the Figure 4.8 can define the reason behind decreasing the accuracy. The small spikes in the noisy signal have transformed into flat signals after the downsampling, which increased in its turn the complexity to distinguish between the event and the non-event.

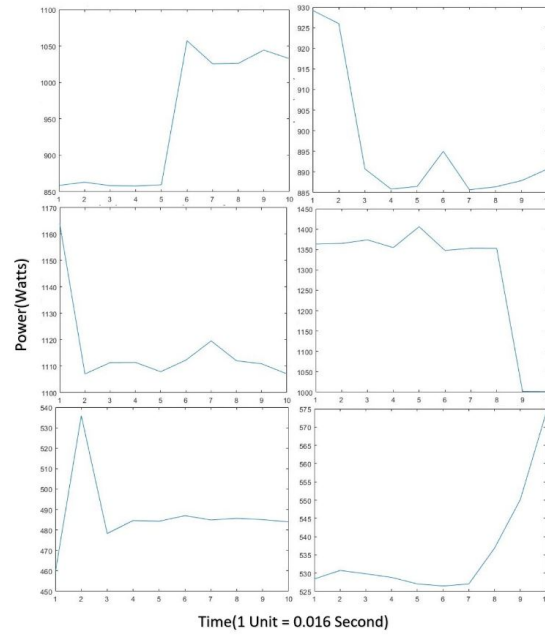


Figure 4.7: The found shapelets in the downsampled instances by factor 18 from the phase B.

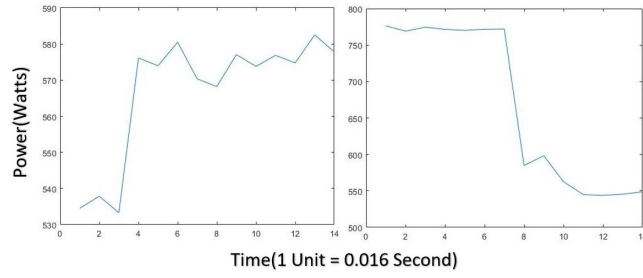


Figure 4.8: **Two randomly picked up samples from the false negative instances of the downsampled data by the factor 18 from the phase B.**

4.2 RESULTS SUMMARY

In order to make it easier to read and compare, the accuracy and the F1-Score for both phases A and B are illustrated sequentially in two plots.

The two observations which are made earlier about the correlation between the event detection accuracy and the downsampling factor from both phases A and B are confirmed by using the [Figure 4.9](#) and the [Figure 4.10](#). On the one hand, the [Figure 4.9](#) shows that the accuracy and the f1-score of the event detection from the phase A varies in a range between 76 % and 96 % classification accuracy, and between 0.55 and 0.94 F1-score. Moreover, the difference between the results of the first downsampling factor and the results of the last downsampling factor is approximately 1 % accuracy and 0.3 F1-score. This means no correlation can be found. On the other hand, the [Figure 4.10](#) shows the correlation between the downsampling factor and the accuracy. A linear correlation can be seen between the both evaluation metrics and the downsampling factor from the phase B.

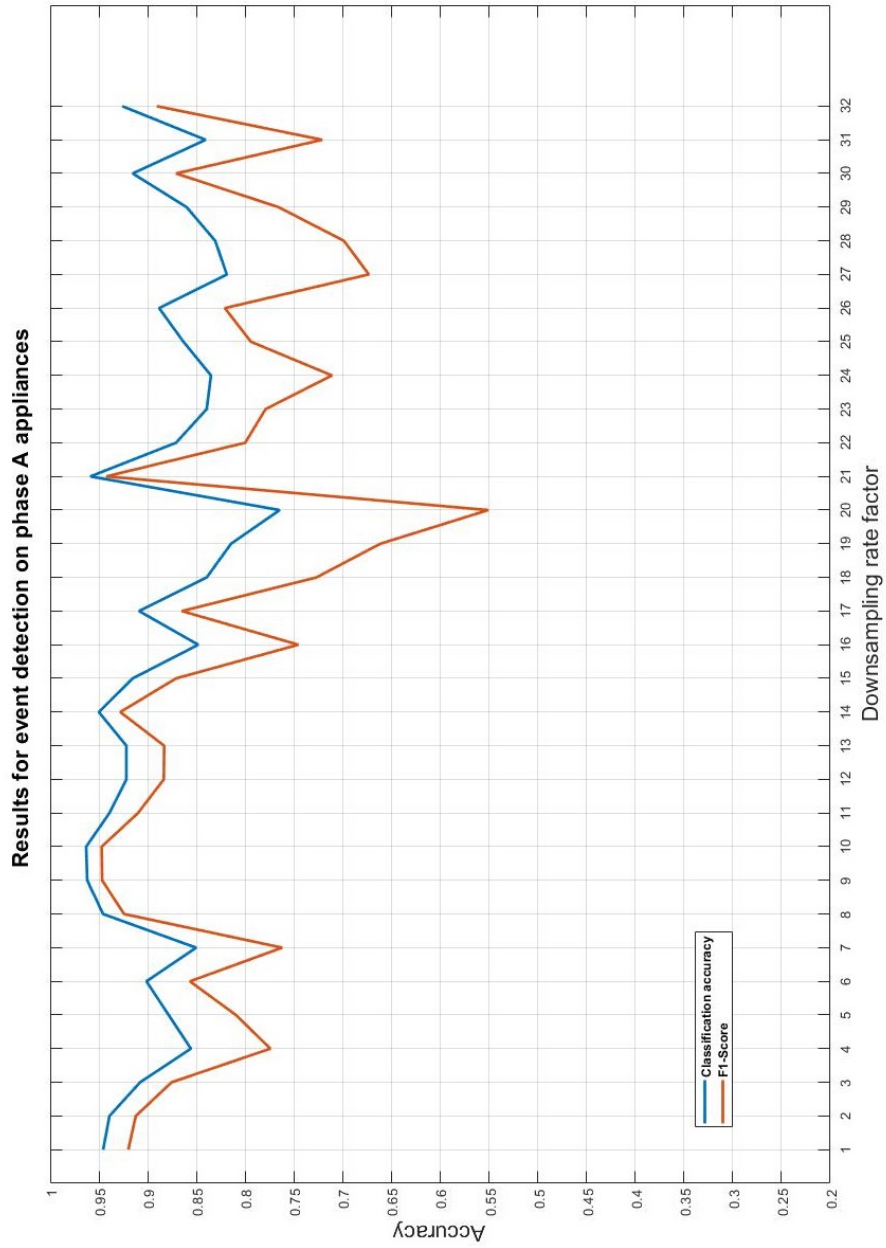


Figure 4.9: Visual representation of the event detection results from the phase A.

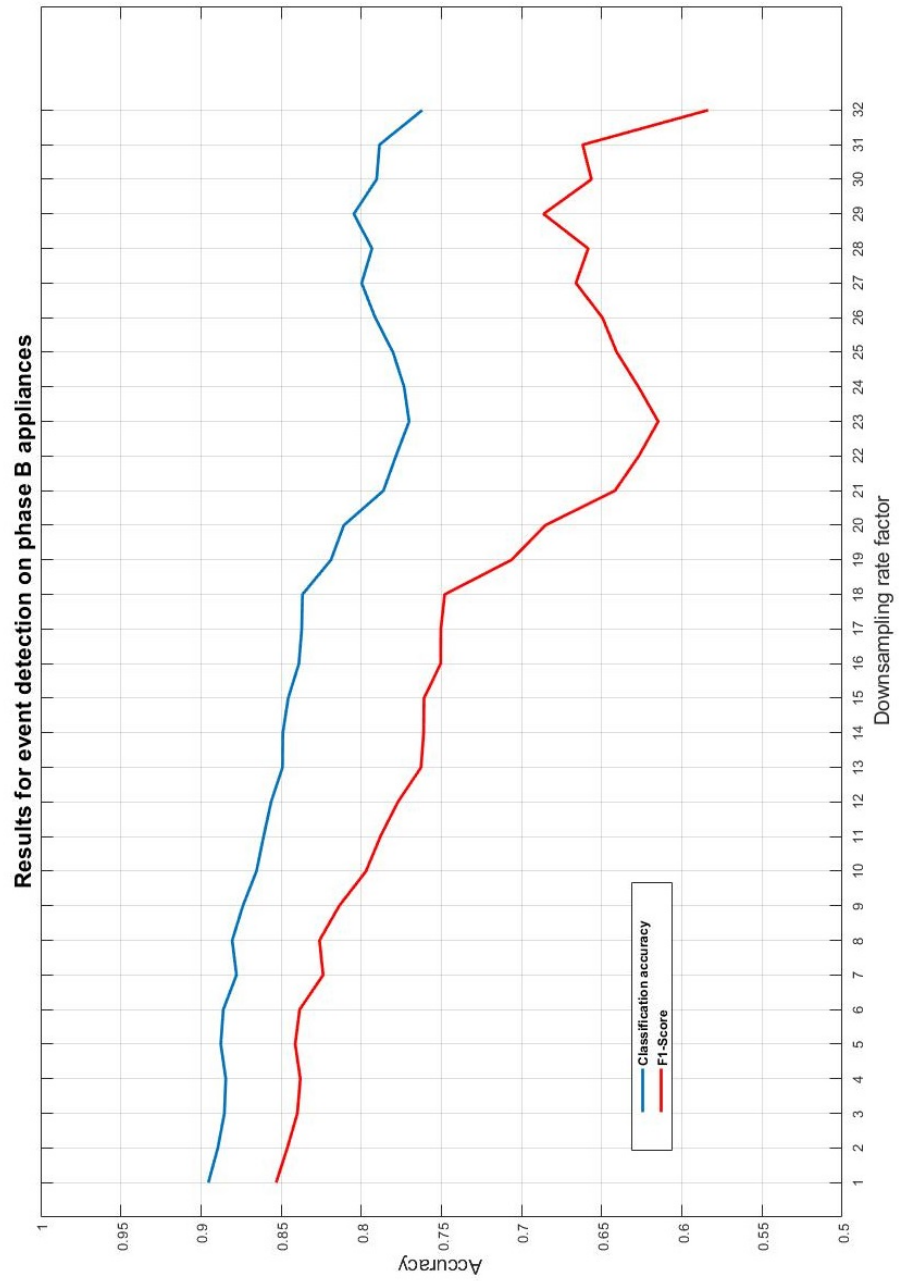


Figure 4.10: Visual representation of the event detection results from the phase B.

CONCLUSION

This thesis demonstrates that the impact of data sampling rates on detecting electrical appliance's events can be measured using a machine learning method based on fast shapelet for detecting events by selecting an efficient discriminative subsequences in the re-sampled **BLUED** dataset. In the evaluation the obtained accuracy is compared with the sampling rate using the F1-score and the classification accuracy. A general architecture for such model is proposed, in which the impact of the downsampling factor can be measured and evaluated. The research question focused on the relationship between the sampling rates for feature extraction and the **NILM** event detection.

The evaluation of the implemented approach with a downsampling rate conversion technique has shown that the fast shapelet approach generally performs better at the low sampling frequency with the clean data than the noisy data. Furthermore, analysis of the graphs shows how the sampling rate is linearly to the model accuracy on the noisy data. In particular, at low sampling frequency the detection accuracy decreases. The algorithm succeeds, however, in scoring better performance with low sampling rate on the clean data.

The findings can have a positive impact at the forefront of the **NILM** field, as collecting the data at low sampling rates for **NILM** purposes can save the recording, storage, and financial costs of the data acquisition phase, while providing similar or even better detection accuracy. This can be done only by taking into account avoiding the noise during the recording stage.

5.1 FUTURE WORK

It became clear to the reader that the impact of the downsampling rate on the event detection accuracy depends strongly on the degree of the noise in the data. In practically when the experiments was executed on the phase B of the real world **BLUED**.

For short term research, the influence of changing the implemented algorithm's parameters to improve the accuracy can be studied, such as the random projection's iterations and the number of **SAX** word candidates (refer to the **Section 3.1.7** for more information). This analysis should occur, especially in the points where the algorithm

obtained low accuracy due to the selected downsampling factor. When more time is available, other training methods can be applied, like using the cross-validation method to ensure the predictive performance of the model. Another real-world dataset could be chosen that replaces the **BLUED** and has an acceptable degree of noise; this could affect the detection accuracy according to this study.

In the long term, another event detection approaches could be researched with taking into account the impact of utilizing different models, which needs to be investigated. These models should be able to detect events from different segment lengths, so it allows abandoning the fix length window technique and covering more events hence more appliance types. A machine learning based the neural networks approach could be a proper candidate.

BIBLIOGRAPHY

- [AKM+17] I. Abubakar, S. Khalid, M. Mustafa, H. Shareef, and M. Mustapha. "Application of load monitoring in appliances' energy management—A review." In: *Renewable and Sustainable Energy Reviews* 67 (2017), pp. 235–245.
- [AA15] S. Agrawal and J. Agrawal. "Survey on anomaly detection using data mining techniques." In: *Procedia Computer Science* 60 (2015), pp. 708–713.
- [AF15] E. Aladesanmi and K. Folly. "Overview of non-intrusive load monitoring and identification techniques." In: *IFAC-PapersOnLine* 48.30 (2015), pp. 415–420.
- [AHS11] M. Alahmad, H. Hasna, and E. Sordiashie. "Non-intrusive electrical load monitoring and profiling methods for applications in energy management systems." In: *2011 IEEE Long Island Systems, Applications and Technology Conference*. IEEE. 2011, pp. 1–6.
- [ABO+12] K. D. Anderson, M. E. Bergés, A. Ocneanu, D. Benitez, and J. M. Moura. "Event detection for non intrusive load monitoring." In: *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society*. IEEE. 2012, pp. 3312–3317.
- [AGSA13] K. C. Armel, A. Gupta, G. Shrimali, and A. Albert. "Is disaggregation the holy grail of energy efficiency? The case of electricity." In: *Energy Policy* 52 (2013), pp. 213–234.
- [BSY14a] K. S. Barsim, R. Streubel, and B. Yang. "An approach for unsupervised non-intrusive load monitoring of residential appliances." In: *Proceedings of the 2nd International Workshop on Non-Intrusive Load Monitoring*. 2014.
- [BSY14b] K. S. Barsim, R. Streubel, and B. Yang. "Unsupervised adaptive event detection for building-level energy disaggregation." In: *Proceedings of power and energy student summt (PESS), Stuttgart, Germany* (2014).

- [BDS13] N. Batra, H. Dutta, and A. Singh. "Indic: Improved non-intrusive load monitoring using load division and calibration." In: *2013 12th International Conference on Machine Learning and Applications*. Vol. 1. IEEE. 2013, pp. 79–84.
- [BKP+14] N. Batra, J. Kelly, O. Parson, H. Dutta, W. Knottenbelt, A. Rogers, A. Singh, and M. Srivastava. "NILMTK: an open source toolkit for non-intrusive load monitoring." In: *Proceedings of the 5th international conference on Future energy systems*. ACM. 2014, pp. 265–276.
- [BKC+14] C. Beckel, W. Kleiminger, R. Cicchetti, T. Staake, and S. Santini. "The ECO data set and the performance of non-intrusive load monitoring algorithms." In: *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*. ACM. 2014, pp. 80–89.
- [BGMS10] M. E. Berges, E. Goldman, H. S. Matthews, and L. Soibelman. "Enhancing electricity audits in residential buildings with nonintrusive load monitoring." In: *Journal of industrial ecology* 14.5 (2010), pp. 844–858.
- [BG94] M. Buckland and F. Gey. "The relationship between recall and precision." In: *Journal of the American society for information science* 45.1 (1994), pp. 12–19.
- [BT02] J. Buhler and M. Tompa. "Finding motifs using random projections." In: *Journal of computational biology* 9.2 (2002), pp. 225–242.
- [CGFB+06] J. M. Carrasco, L. Garcia Franquelo, J. T. Bialasiewicz, E. Galván, R. C. Portillo Guisado, M. d.l. Á. Martín Prats, J. I. León, and N. Moreno-Alfonso. "Power-electronic systems for the grid integration of renewable energy sources: A survey." In: *IEEE Transactions on Industrial Electronics*, 53 (4), 1002–1016. (2006).
- [Cha12] H.-H. Chang. "Non-intrusive demand monitoring and load identification for energy management systems based on transient feature analyses." In: *Energies* 5.11 (2012), pp. 4569–4589.
- [Cha09] N. V. Chawla. "Data mining for imbalanced datasets: An overview." In: *Data mining and knowledge discovery handbook*. Springer, 2009, pp. 875–886.
- [DGo6] J. Davis and M. Goadrich. "The relationship between Precision-Recall and ROC curves." In: *Proceedings of the 23rd international conference on Machine learning*. ACM. 2006, pp. 233–240.

- [DBRDD16] L. De Baets, J. Ruyssinck, D. Deschrijver, and T. Dhaene. "Event detection in NILM using cepstrum smoothing." In: *3rd International Workshop on Non-Intrusive Load Monitoring*. 2016, pp. 1–4.
- [DTDT72] V. Del Toro and V. Del Toro. *Principles of electrical engineering*. Prentice-Hall, 1972.
- [DNG+15] C. Dinesh, B. W. Nettasinghe, R. I. Godaliyadda, M. P. B. Ekanayake, J. Ekanayake, and J. V. Wijayakulasooriya. "Residential appliance identification based on spectral information of low frequency smart meter measurements." In: *IEEE Transactions on smart grid* 7.6 (2015), pp. 2781–2792.
- [DROS14] R. Dong, L. Ratliff, H. Ohlsson, and S. S. Sastry. "Fundamental limits of nonintrusive load monitoring." In: *Proceedings of the 3rd international conference on High confidence networked systems*. ACM. 2014, pp. 11–18.
- [FMXY11] X. Fang, S. Misra, G. Xue, and D. Yang. "Smart grid—The new and improved power grid: A survey." In: *IEEE communications surveys & tutorials* 14.4 (2011), pp. 944–980.
- [FMXY12] X. Fang, S. Misra, G. Xue, and D. Yang. "Smart grid—The new and improved power grid: A survey." In: *IEEE communications surveys & tutorials* 14.4 (2012), pp. 944–980.
- [Fawo6] T. Fawcett. "An introduction to ROC analysis." In: *Pattern recognition letters* 27.8 (2006), pp. 861–874.
- [Fil11] A. Filip. "Blued: A fully labeled public dataset for event-based nonintrusive load monitoring research." In: *2nd Workshop on Data Mining Applications in Sustainability (SustKDD)*. 2011, p. 2012.
- [FBo8] T. B. Fomby and T. Barber. "K-nearest neighbors algorithm: Prediction and classification." In: *Lecture notes in Southern Methodist University, Dallas, TX* (2008), pp. 1–5.
- [Ful18] B. D. Fulcher. "Feature-based time-series analysis." In: *Feature Engineering for Machine Learning and Data Analytics*. CRC Press, 2018, pp. 87–116.
- [GMO13] F. Gangale, A. Mengolini, and I. Onyeji. "Consumer engagement: An insight from smart grid projects in Europe." In: *Energy Policy* 60 (2013), pp. 621–628.

- [GB15] S. Giri and M. Bergés. “An energy estimation framework for event-based methods in Non-Intrusive Load Monitoring.” In: *Energy Conversion and Management* 90 (2015), pp. 488–498.
- [GFS+11] L. Gomes, F. Fernandes, T. Sousa, M. Silva, H. Morais, Z. Vale, and C. Ramos. “Contextual intelligent load management with ANN adaptive learning module.” In: *2011 16th International Conference on Intelligent System Applications to Power Systems*. IEEE. 2011, pp. 1–6.
- [GG05] C. Goutte and E. Gaussier. “A probabilistic interpretation of precision, recall and F-score, with implication for evaluation.” In: *European Conference on Information Retrieval*. Springer. 2005, pp. 345–359.
- [GN96] P. E. Greenwood and M. S. Nikulin. *A guide to chi-squared testing*. Vol. 280. John Wiley & Sons, 1996.
- [GRP10] S. Gupta, M. S. Reynolds, and S. N. Patel. “ElectriSense: single-point sensing using EMI for electrical event detection and classification in the home.” In: *Proceedings of the 12th ACM international conference on Ubiquitous computing*. ACM. 2010, pp. 139–148.
- [Har92] G. W. Hart. “Nonintrusive appliance load monitoring.” In: *Proceedings of the IEEE* 80.12 (1992), pp. 1870–1891.
- [HAK+15] N. Henao, K. Agbossou, S. Kelouwani, Y. Dubé, and M. Fournier. “Approach in nonintrusive type I load monitoring using subtractive clustering.” In: *IEEE Transactions on Smart Grid* 8.2 (2015), pp. 812–821.
- [HKZ16] L. Hou, J. T. Kwok, and J. M. Zurada. “Efficient learning of timeseries shapelets.” In: *Thirtieth AAAI Conference on Artificial Intelligence*. 2016.
- [IJ02] E. C. Ifeachor and B. W. Jervis. *Digital signal processing: a practical approach*. Pearson Education, 2002.
- [JCDLT13] L. A. Jeni, J. F. Cohn, and F. De La Torre. “Facing imbalanced data—recommendations for the use of performance metrics.” In: *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*. IEEE. 2013, pp. 245–251.
- [JTBS11] Y. Jin, E. Tebekaemi, M. Berges, and L. Soibelman. “Robust adaptive event detection in non-intrusive load monitoring for energy aware smart facilities.” In: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2011, pp. 4340–4343.

- [KHKJ16] M. Kahl, A. U. Haq, T. Kriechbaumer, and H.-A. Jacobsen. "Whited-a worldwide household and industry transient energy data set." In: *3rd International Workshop on Non-Intrusive Load Monitoring*. 2016.
- [KK15a] J. Kelly and W. Knottenbelt. "Neural NILM: Deep neural networks applied to energy disaggregation." In: *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*. ACM. 2015, pp. 55–64.
- [KK15b] J. Kelly and W. Knottenbelt. "The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes." In: *Scientific data* 2 (2015), p. 150007.
- [KK16] J. Kelly and W. Knottenbelt. "Does disaggregated electricity feedback reduce domestic electricity consumption? A systematic review of the literature." In: *arXiv preprint arXiv:1605.00962* (2016).
- [KJ11] J. Z. Kolter and M. J. Johnson. "REDD: A public data set for energy disaggregation research." In: *Workshop on Data Mining Applications in Sustainability (SIGKDD)*, San Diego, CA. Vol. 25. Citeseer. 2011, pp. 59–62.
- [LKWLo7] J. Lin, E. Keogh, L. Wei, and S. Lonardi. "Experiencing SAX: a novel symbolic representation of time series." In: *Data Mining and knowledge discovery* 15.2 (2007), pp. 107–144.
- [LK11] S. Lin and B. W. Kernighan. "Experiment Holzbeton green:house." In: *Deutschen Bauzeitschrift*, (2011).
- [LL19] M. Lu and Z. Li. "A Hybrid Event Detection Approach for Non-Intrusive Load Monitoring." In: *arXiv preprint arXiv:1903.09180* (2019).
- [MSPV13] M. Maasoumy, B Sanandaji, K. Poolla, and A. S. Vincetelli. "Berds-berkeley energy disaggregation data set." In: *Proceedings of the Workshop on Big Learning at the Conference on Neural Information Processing Systems (NIPS)*. 2013, pp. 1–6.
- [MABDV+12] C. B. Martinez-Anido, R Bolado, L De Vries, G Fulli, M Vandenbergh, and M Masera. "European power grid reliability indicators, what do they really tell?" In: *Electric Power Systems Research* 90 (2012), pp. 79–84.
- [Mas17] E. R. Masanet. *Energy technology perspectives 2017: Catalysing energy technology transformations*. OECD, 2017.

- [Met78] C. E. Metz. "Basic principles of ROC analysis." In: *Seminars in nuclear medicine*. Vol. 8. 4. Elsevier. 1978, pp. 283–298.
- [MFZA15] M. Milligan, B. Frew, E. Zhou, and D. J. Arent. *Advancing System Flexibility for High Penetration Renewable Integration (Chinese Translation)*. Tech. rep. National Renewable Energy Lab.(NREL), Golden, CO (United States), 2015.
- [MK14] P. Mineiro and N. Karampatziakis. "A randomized algorithm for cca." In: *arXiv preprint arXiv:1411.3409* (2014).
- [MEE+14] A. Monacchi, D. Egarter, W. Elmenreich, S. D'Alessandro, and A. M. Tonello. "GREEND: An energy consumption dataset of households in Italy and Austria." In: *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE. 2014, pp. 511–516.
- [NMR18] B. Najafi, S. Moaveninejad, and F. Rinaldi. "Data analytics for energy disaggregation: methods and applications." In: *Big Data Application in Power Systems*. Elsevier, 2018, pp. 377–408.
- [NL96] L. K. Norford and S. B. Leeb. "Non-intrusive electrical load monitoring in commercial buildings based on steady-state and transient load-detection algorithms." In: *Energy and Buildings* 24.1 (1996), pp. 51–64.
- [Par12] O. Parson. *Disaggregated Homes*. [Online; accessed 03-June-2019]. 2012.
- [Par14] O. Parson. "Unsupervised training methods for non-intrusive appliance load monitoring from smart meter data." PhD thesis. University of Southampton, 2014.
- [PFH+15] O. Parson, G. Fisher, A. Hersey, N. Batra, J. Kelly, A. Singh, W. Knottenbelt, and A. Rogers. "Dataport and nilmtk: A building data set designed for non-intrusive load monitoring." In: *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE. 2015, pp. 210–214.
- [PPCP14] O. P. Patri, A. V. Panangadan, C. Chelmiss, and V. K. Prasanna. "Extracting discriminative features for event-based electricity disaggregation." In: *2014 IEEE Conference on Technologies for Sustainability (SusTech)*. IEEE. 2014, pp. 232–238.

- [PHM11] N. V. Perez, T. P. Harmo, and M. Manrique. "A non-intrusive appliance load monitoring system for identifying kitchen activities." In: *MS thesis, Aalto University, Faculty of Electronics Communications and Automation*. Citeseer. 2011.
- [PMR+16] 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." In: *arXiv preprint arXiv:1611.05803* (2016).
- [Pow11] D. M. Powers. "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation." In: *Journal of Machine Learning Technologies*, ISSN: 2229-399X (2011).
- [QLW18] B. Qi, L. Liu, and X. Wu. "Low-rate non-intrusive load disaggregation with graph shift quadratic form constraint." In: *Applied Sciences* 8.4 (2018), p. 554.
- [RG75] L. R. Rabiner and B. Gold. "Theory and application of digital signal processing." In: *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1975*. 777 p. (1975).
- [RK13] T. Rakthanmanon and E. Keogh. "Fast shapelets: A scalable algorithm for discovering time series shapelets." In: *proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM. 2013, pp. 668–676.
- [RVV+14] A. Ravishankar, A. Vignesh, V. Vel, D. P. SRR, and V. Vijayaraghavan. "Low-cost non-intrusive residential energy monitoring system." In: *2014 IEEE Conference on Technologies for Sustainability (SusTech)*. IEEE. 2014, pp. 130–134.
- [RGH14] A. Ridi, C. Gisler, and J. Hennebert. "A survey on intrusive load monitoring for appliance recognition." In: *2014 22nd International Conference on Pattern Recognition*. IEEE. 2014, pp. 3702–3707.
- [RGH15] A. Ridi, C. Gisler, and J. Hennebert. "User interaction event detection in the context of appliance monitoring." In: *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. IEEE. 2015, pp. 323–328.

- [RDEH+16] J. Rogelj, M. Den Elzen, N. Höhne, T. Fransen, H. Fekete, H. Winkler, R. Schaeffer, F. Sha, K. Riahi, and M. Meinshausen. "Paris Agreement climate proposals need a boost to keep warming well below 2 C." In: *Nature* 534.7609 (2016), p. 631.
- [SL91] S. R. Safavian and D. Landgrebe. "A survey of decision tree classifier methodology." In: *IEEE transactions on systems, man, and cybernetics* 21.3 (1991), pp. 660–674.
- [Sas+07] Y. Sasaki et al. "The truth of the F-measure." In: *Teach Tutor mater* 1.5 (2007), pp. 1–5.
- [SRCS91] C. Savant, M. S. Roden, G. L. Carpenter, and C. Savant. *Electronic design: circuits and systems*. Benjamin/Cummings Publishing Company, 1991.
- [SNN+14] P. H. Shaikh, N. B. M. Nor, P. Nallagownden, I. Elamvazuthi, and T. Ibrahim. "A review on optimized control systems for building energy and comfort management of smart sustainable buildings." In: *Renewable and Sustainable Energy Reviews* 34 (2014), pp. 409–429.
- [SSBD14] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
- [SWK09] Y. Sun, A. K. Wong, and M. S. Kamel. "Classification of imbalanced data: A review." In: *International Journal of Pattern Recognition and Artificial Intelligence* 23.04 (2009), pp. 687–719.
- [TR] E. K. Thanawin Rakthanmanon. *Fast-Shapelets: A Scalable Algorithm for Discovering Time Series Shapelets*. [Online; accessed 30-April-2019]. URL: <http://alumni.cs.ucr.edu/~rakthant/FastShapelet/>.
- [TSR+05] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. "Sparsity and smoothness via the fused lasso." In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.1 (2005), pp. 91–108.
- [VRRVDK11] S. Visa, B. Ramsay, A. L. Ralescu, and E. Van Der Knaap. "Confusion Matrix-based Feature Selection." In: *MAICS* 710 (2011), pp. 120–127.
- [WKX06] L. Wei, E. Keogh, and X. Xi. "Sexually explicit images: Finding unusual shapes." In: *Sixth International Conference on Data Mining (ICDM'06)*. IEEE. 2006, pp. 711–720.

- [WACLo9] W. Wichakool, A.-T. Avestruz, R. W. Cox, and S. B. Leeb. "Modeling and estimating current harmonics of variable electronic loads." In: *IEEE Transactions on power electronics* 24.12 (2009), pp. 2803–2811.
- [YKo9] L. Ye and E. Keogh. "Time series shapelets: a new primitive for data mining." In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2009, pp. 947–956.
- [ZR11] M. Zeifman and K. Roth. "Nonintrusive appliance load monitoring: Review and outlook." In: *IEEE transactions on Consumer Electronics* 57.1 (2011), pp. 76–84.
- [ZGIR12] A. Zoha, A. Gluhak, M. Imran, and S. Rajasegarar. "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey." In: *Sensors* 12.12 (2012), pp. 16838–16866.
- [com] N. community. *NILM Wiki*. [Online; accessed 20-June-2019]. URL: <http://wiki.niln.eu/>.

APPENDIX

Index	Name	Average Power Consumption (W)	Number of events	Phase
1	Desktop Lamp	30	26	B
2	Tall Desk Lamp	30	25	B
3	Garage Door	530	24	B
4	Washing Machine	130-700	95	B
6	Kitchen Aid Chopper	1500	16	A
9	Fridge	120	616	A
10	A/V Living room	45	8	B
12	Computer A	60	45	B
13	Laptop B	40	14	B
16	DVR, A/V Receiver, Blu-ray Player, Basement	55	34	B
19	Air Compressor	1130	20	A
20	LCD Monitor A	35	77	B
21	TV Basement	190	54	B
23	Printer	930	150	B
24	Hair Dryer	1600	8	A
25	Iron	1400	40	B
26	Empty living room socket	60	2	B
28	Monitor B	40	150	B
29	Backyard lights	60	16	A
30	Washroom light	110	6	A
31	Office Lights	30	54	B
32	Closet lights	20	22	B
33	Upstairs hallway light	25	17	B
34	Hallways Stairs lights	110	58	B
35	Kitchen Hallway light	15	6	B
36	Kitchen overhead light	65	56	B
37	Bathroom upstairs lights	65	98	A
38	Dining room overhead light	65	32	B
39	Bedroom Lights	190	19	A
40	Basement Light	35	39	B
41	Microwave	1550	70	B

Figure A.1: List of the appliances are monitored in the BLUED [Fil11].

Data set	Institution	Location	Duration	Number of houses	Number of Sub-meters	Appliance frequency Sampling	Aggregate data frequency
REDD	MIT	MA, USA	3-19 days	6	9-24	3 sec	1 sec & 15 kHz
BLUED	CMU	PA, USA	8 days	1	✗	✗	12 kHz
Smart*	UMass	MA, USA	3 months	3	54	1 sec	1 sec
Tracebase	Darmstadt	Germany	✗	✗	✗	1-10 sec	10 sec
Sample data set	Pecan Street	TX, USA	7 days	10	12	1 min	1 min
IHEPCDS	EDF R&D	France	4 years	1	3	1 min	1 min
HES UK	DECC	UK	3 months	250	13-51	2 min	2 min
AMPds	Simon Fraser U.	BC, Canada	2 years	1	19	1 min	1 min
Colden Common	U.Southampton	UK	1 year	117	✗	✗	✗
BERDS	Berkley University	California	7 days	1	4	20 sec	20 sec
Dataport	Pecan Street	TX, USA	3 years	722	10-23	1 min	1 min
DRED	TU Delft	Netherlands	6 months	1	9	1 sec	1 sec
ECO	ETH Zurich	Switzerland	8 months	6	6-10	1 sec	1 sec
GREEND	Alpen-Adria Universitat Klagenfurt	Italy-Austria	1 year	9	9	1 sec	✗
iAWE	Indraprastha Institute of Information Technology	India	73 days	1	33	1-6 sec	1 sec
REFIT	U. of Strathclyde	UK	73 days	21 months	9	8 sec	30 min
UK-DALE	U. of Southampton	UK	2.5 years	5	54	6sec	1-6 sec & 16kHz

Figure A.2: Characteristics overview of the available energy datasets [Par14].