8-26-2019 3:00 PM

Classifying Appliances Operation Modes Using Dynamic Time Warping (DTW) And K Nearest Neighbors (KNN)

Abdelkareem M. Jaradat, *The University of Western Ontario*

Supervisor: Lutfiyya, Hanan L., *The University of Western Ontario*
Co-Supervisor: Haque, Anwar, *The University of Western Ontario*
A thesis submitted in partial fulfillment of the requirements for the Master of Science degree in Computer Science

Abstract

In the Smart Grid environment, the advent of intelligent measuring devices facilitates monitoring appliance electricity consumption. This data can be used in applying Demand Response (DR) in residential houses through data analytics, and developing data mining techniques. In this research, we introduce a smart system approach that is applied to user's disaggregated power consumption data. This system encourages the users to apply DR by changing their behaviour of using heavier operation modes to lighter modes, and by encouraging users to shift their usages to off-peak hours. First, we apply Cross Correlation to detect times of the occurrences when an appliance is being used. We then use two approaches to recognize the operation mode used: The Dynamic Time Warping (DTW), and Machine Learning using K-Means and K-Nearest Neighbors (KNN).

Keywords: Dynamic Time Warping, Machine Learning, K-nearest Neighbors, Smart Systems, Time-Series Analysis, Smart Grid, Disaggregated Power Consumption, Demand Response, IoT

Summary for Lay Audience

Technology is rapidly evolving. The electricity network is changing. This network is getting smarter with the use of smarter measuring devices which are capable of seamlessly collecting various types of information on consumer electricity consumption. This information is collected frequently by devices that sense through the wired connections then store the electricity consumption data in a storage device that is accessible by the hydro company. Consequently, the hydro company performs analysis on the stored information and then gets back to the consumer with tips and recommendations on reducing electricity bill by urging the consumer to change certain habits in operating house appliances.

In this research, we introduce a smart system approach that is applied to a household's electricity consumption data for each appliance. The main objective of this system is to encourage the consumers to change their behavior with certain appliances by switching using heavier operation modes to lighter operation modes. Also, the system aims to urge users to shift their appliances usage to the times of the day where the demand is lower and electricity is cheaper.

To achieve the goals of our study, we analyzed publicly available electricity consumption data for a group of appliances and came up with a representation of this data. We then built an algorithm to mimic this data with the same form, so that we can generate as much data as we need. Thereafter, we developed an algorithm that can search this data and find the times when an appliance is turned on using a searching technique (cross-correlation) that relies on prior knowledge about each appliance electricity consumption pattern. Using the detected time, we applied two other algorithms to determine the operation mode (light, medium, heavy) for each detected instance of the house appliances. We used an algorithm based on comparison (Dynamic Time Warping), and the other algorithm is based on artificial intelligence (K Nearest Neighbors).

Acknowledgment

First and foremost, I would like to express my heartiest gratitude to the Almighty, most gracious, and most merciful, for blessing me with the ability to proceed successfully to accomplish this thesis.

I would like to express my utmost gratitude to my supervisor Prof. Hanan Lutfiyya for giving me her valuable time, constant encouragement and guidance, and ample knowledge. Her supervision in conducting the research and writing of this thesis was extraordinary. Her encouragement, kindness, spot on comments, insightful suggestions, and her prompt responses are unforgettable. Without her assistance, this work could not have been accomplished.

My sincere gratitude goes also to my co-supervisor Dr. Anwar Haque for his continuous help and support in all stages through my study. Without his involvement, this work will not have been possible.

I am deeply and forever indebted to my family. The greatest source of love, support, and inspiration. My father and my mother, words stop speechless when it comes to express my thanks. My brothers Yaser and Hamzeh, whom I get my strength. My sisters Asma (God bless her soul), Arwa, Sahar, and Eman whom I belong to. Without all of them, I would never took a step forward.

To my friends scattered around the world, thank you for your well-wishes Also, I would like to thank my friends scattered around the world for your well-wishes, and giving the support, consultations and opinions whenever I need it.

Lastly, my heartful thanks to my love, my wife, who gave me the endless emotional support to keep going into achieving my goal. And of course my two cute little kids, the best friends.

Dedication

To my mother, Fatima Jaradat–

The heart that I live in..

To my father, Mahmoud Jaradat–

My source of inspiration..

To my son, Bilal–

The good-hearted..

To my daughter, Asal–

The affectionate..

To my wife, Amal Jaradat–

My love ..

Contents

List of Figures

List of Tables

List of Appendices

List of Abbreviations and Symbols

GEM	:	*General Electricity Meters*
SM	:	*Smart Meters*
PCDS	:	*Power Consumption DataSet*
RAE	:	*Rainforest Automation Energy*
IHD	:	*In-Home Display*
SUP	:	*Single Usage Profile*
DUP	:	*Daily Usage Profile*
WUP	:	*Weekly Usage Profile*
PMF	:	*Probability Mass Function*
PMF	:	*Probability Mass Function*
CDF	:	*Cumulative Distribution Function*
ITS	:	*Inverse Transform Sampling*
DTW	:	*Dynamic Time Warping*
MST	:	*Moving Step-Test*
DR	:	*Demand Response*
NRC	:	*Natural Resources Canada*
KWH	:	*kilowatt-hour*
AMI	:	*Advanced Metering Infrastructure*
MDMS	:	*Meter Data Management System*
CIS	:	*Consumer Information System*
PLC	:	*Power Line Carrier*
FERC	:	*Federal Energy Regulatory Commission*
NILM	:	*Non Intrusive Load Monitoring*
ED	:	*Energy Disaggregation*

ToUT	:	*Time Of Use Tariff*
NFC	:	*Near Field Communication*
BLE	:	*Bluetooth Enabled*
OSN	:	*Online Social Networks*
FDK	:	*Facebook Developers Kit*
SMLP	:	*Simple Method of prediction of daily Load Profile*
MCM	:	*Markov Chain Model*
ALPG	:	*Artificial Load Profile Generator*
DSM	:	*Demand Side Management*
GLR	:	*Generalized Likelihood Ratio*
GoF	:	*Goodness of Fit Chi-squared test*
SFT	:	*Short-time Fourier Transform*
EM	:	*Expectation Maximization*
SOM	:	*Self-Organizing Map*
ANN	:	*Artificial Neural Network*
PKNN	:	*Probabilistic K-Nearest Neighbor*
TPW	:	*Transient Power Waveform*
ILDACP	:	*Iterative Disaggregation based on Appliance Consumption Pattern*
IHMM	:	*Iterative Hidden Markov Models*
CW	:	*Current Waveform*
HAR	:	*Harmonics*
CDM	:	*Committee Decision Mechanism*
DSP	:	*Digital Signal Processing*
f_s	:	*Sampling Frequency*
α	:	*Cycle Power Variation Coefficient*
β	:	*Duration Variation Coefficient*
γ	:	*Turn Off Noise Coefficient*
δ	:	*Low Amplitude Canceling Coefficient*
ϕ	:	*DTW Local Cost Function*
ℓ	:	*Step-Test Window Size*
ζ	:	*Low Amplitude Canceling Multiplier*
t_{on}	:	*Turn ON Time*
i_h	:	*Household Usage Intensity*

CHAPTER 1

Introduction

Greenhouse gas emissions in the atmosphere are one of the contributors in increasing the severity of climate change [28]. Half of the current CO_2 emissions are absorbed by oceans and land ecosystems [22]. The World Energy Outlook [72] identified new supply options, mainly renewable energy resources as a way to redistribute resources to limit the effects of their shortage of resources in the future.

As stated by the International World Outlook, by the year of 2040, renewable sources will supply only about 15% of the world's energy needs [108]. This suggests that energy sources conservation should be a high priority. Therefore, the need of sustainability that keeps resources availability arises. This could be achieved by regenerating resources at a rate more than harvesting. Also, energy conservation can save costs in residential uses up to 56% [1].

1.1 Statistics About Canada's Electricity Generation And Consumption

Canada is considered one of the highest six countries in the world in terms of power consumption in the period 2000 to 2014 [113]. Figure 1.1 shows the electricity consumption per capita for the top six electricity consuming countries in the world. The total electricity usage in Canada in 2014 was 1755 petajoules (PJ), where 582 PJ are for residential purposes and 760 PJ are for industrial usage, while the rest is divided between commercial and other kinds of usage. The provincial electricity usage widely varies among provinces. Provinces with plentiful and cheap electricity from large-scale electricity projects are in British Columbia and Quebec, which tend to use more energy than the other provinces. In particular, Quebec uses 37% of the total Canadian electricity, then Ontario at 23%, British Columbia at 11% and all the other provinces together use 29% [113].

According to the Natural Resources Canada (NRC) [19], on an international scale, Canada is the world 6^{th} highest generator of electricity. It generates 3% of world electricity after China, USA, India, Russia, and Japan. The total generated electricity in Canada in 2015 was around 635 terawatt-hour. Hydro (water) is the largest producer of electricity putting Canada as the world's second largest producer of hydro-electricity, followed by nuclear plants at 15%, and

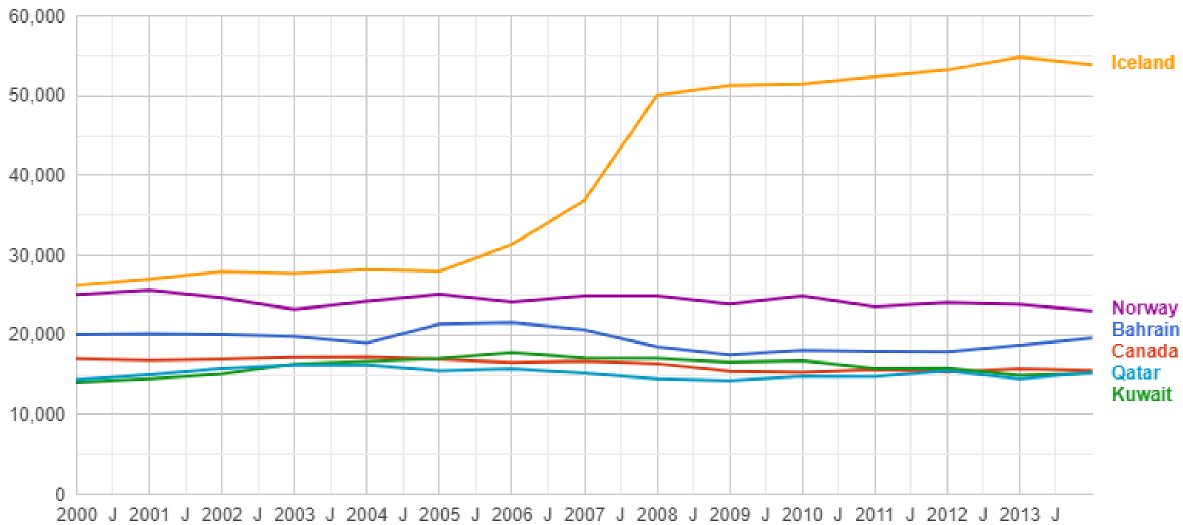


Figure 1.1: Electric power consumption (kWh per capita) for the top 6 countries in electricity consumption [113].

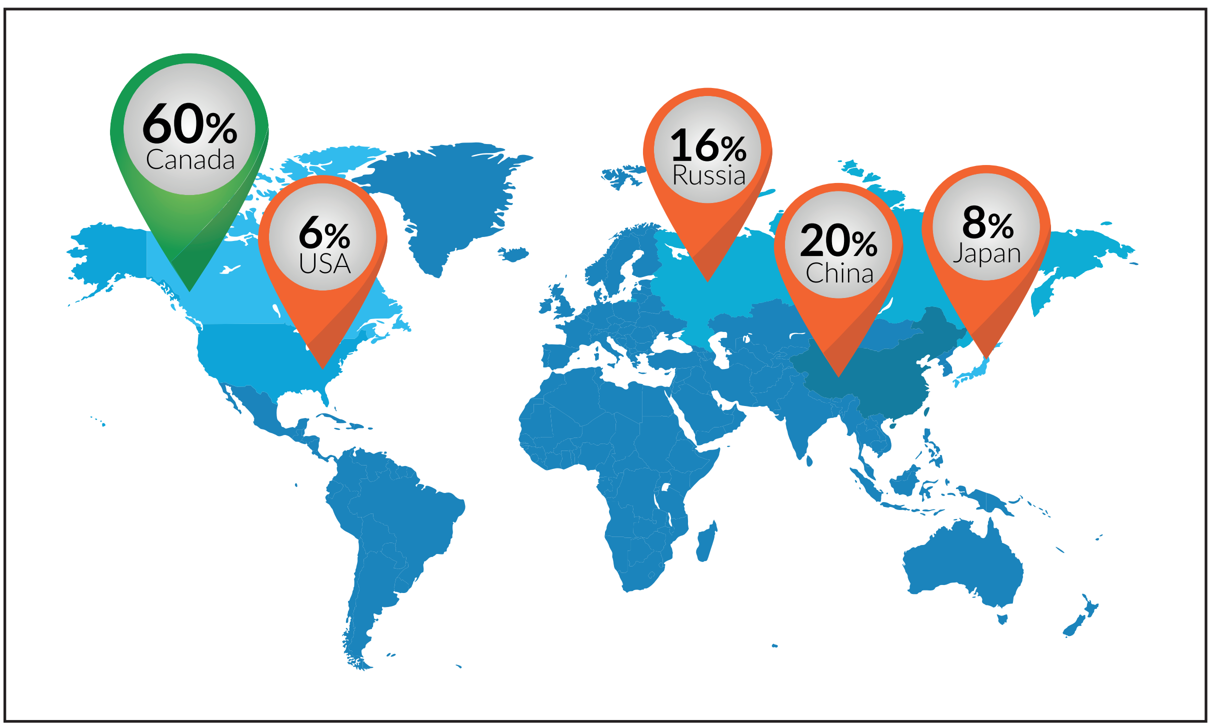


Figure 1.2: Percentage of the Electrical power consumption using hydroelectric compared to the total electricity production in 2015. A comparison between 5 of the leading countries in the world; Canada, USA, Japan, Russia and China [113].

the rest (coal, gas and non-hydro) at 26% (see Figure 1.2). For nearly 11% of the total power generated in Canada, there are 34 active transmission lines between Canada and the USA that transmit power to the US for export purposes [19].

In terms of pricing, Canada is considered one of the lowest prices in the world [113]. The average electricity price in Canada is 11 cents/KWH. Figure 1.3 depicts the average electricity prices in US dollar per KWH. In Canada, the underlying infrastructure and the means of producing electricity play an important role in determining electricity prices across provinces in Canada [9]. Consequently, prices vary among provinces in terms of residential and industrial prices. The average residential electricity prices in cents per kilowatt-hour in April 2016 is the lowest in Alberta at 10.89 and the highest in Ontario which is 20.12 [19].

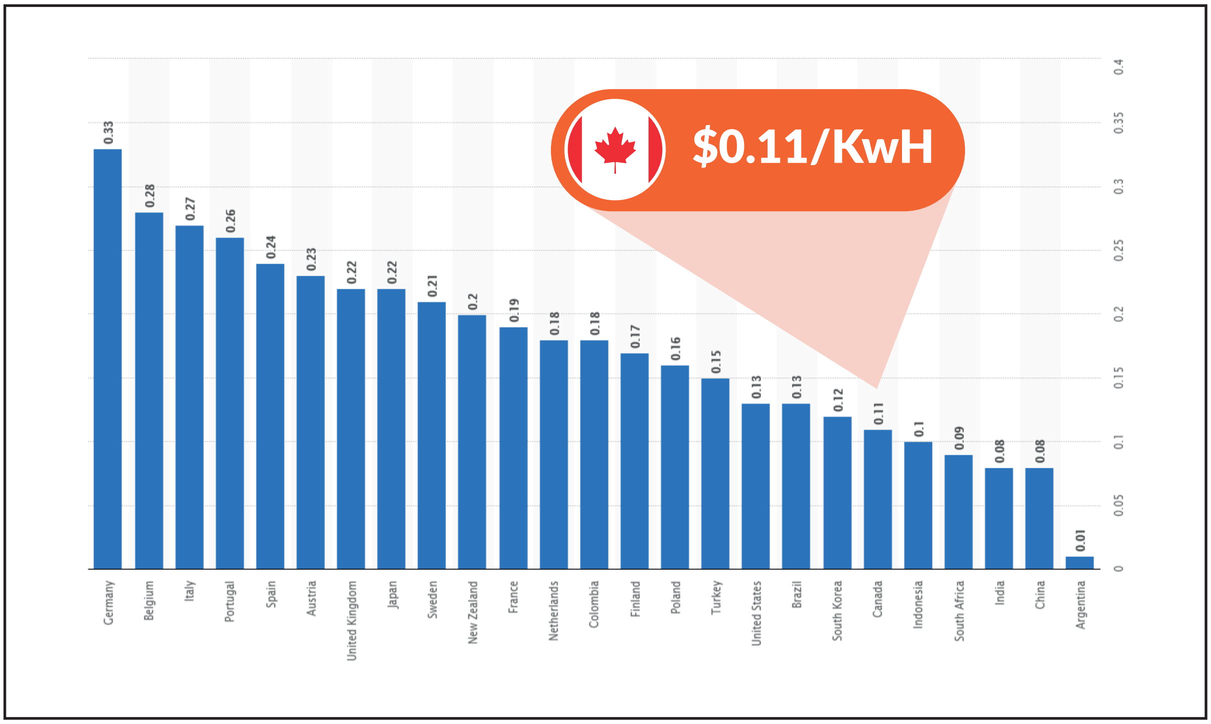


Figure 1.3: This chart shows electricity average prices in selected countries worldwide in 2018. Price is in U.S. dollars per kilowatt hour. [98]

1.2 The Smart Grid

Smart Grid is a cyber-physical system which includes communication system with the power flow structure that can also be used to transmit data, to gain intelligence and automated control [12]. Smart Meters (SMs) are capable to communicate with sensors that sense power usage and store this data into a database that belongs to a Meter Data Management System

(MDMS) [100]. MDMSs are smart systems that involves multiple tools and services to the consumer. These tools rely on the power consumption data that is collected by the Advanced Metering Infrastructure (AMI) and stored in the database.

Examples of MDMSs tools are the following: Consumer Information Systems (CIS) which provide the consumer analytics about consumption, billing systems for reporting bills, etc as shown in Figure 1.4. Other tools focus on reducing home energy use by analyzing consumers' consumption data and provide information that helps consumers gain more knowledge about their consumption patterns and maintain good habits of using electronic devices in the house.

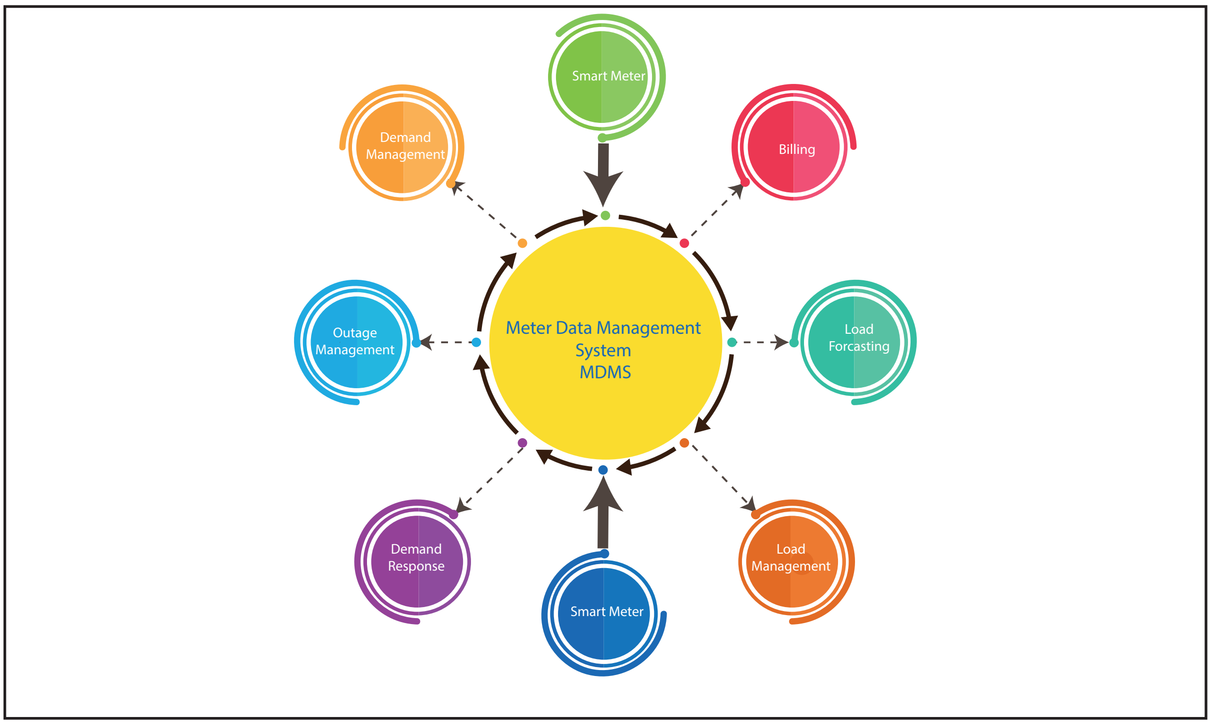


Figure 1.4: Meter Data Management System (MDMS). [12]

1.3 Demand Response DR

Demand Response DR is defined as: "Changes in electric usage by end-use customers from their normal consumption patterns in response to changes in the price of electricity over time, or to incentive payments designed to induce lower electricity use at times of high wholesale market prices or when system reliability is jeopardized" [11].

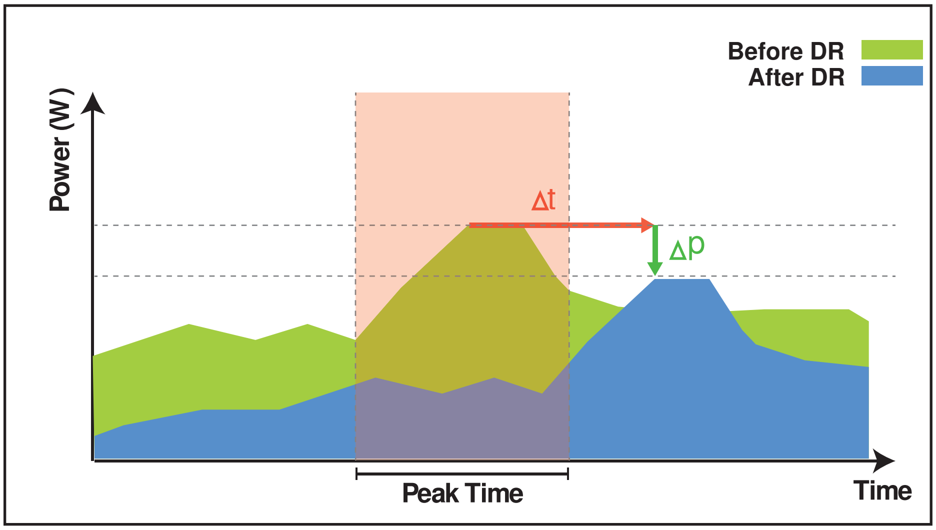


Figure 1.5: Power consumption data before and after applying DR.

Demand response systems are designed to decrease electricity consumption either by shifting it from on-peak to off-peak periods based on consumers preferences, or changing the power consumption level by changing the way how people consume power [96]. The intention is to reduce cost to consumers and reduce network peak load. The reduction of network peak load may limit the need for new infrastructure.

The main two objectives of DR for residential consumers is to reduce electricity consumption and to shift the heavy loads from on-peak hours to off-peak hours where the price of electricity is less expensive and thus reduce bill for consumers. Figure 1.5 shows a power consumption plot for certain household in two different situations. The highlighted area represents the on-peak time when the energy price is relatively higher than the rest of the day. The green area shows the power consumption through time with a relatively high consumption in the highlighted area. The blue area shows the two desired objectives after applying DR. The adaptation with DR shows that the power consumed over time is reduced by Δp. Second, the high load appears within the on-peak before DR is shifted by Δt outside the period where the energy price is less expensive.

1.4 Problem Statement

One of the main goals of Smart Grids is to provide the data that is required to build technologies that encourages consumers to reduce their energy consumption. Therefore, households reinforce the efforts to conserve energy, with the collaboration with the smart home technology.

There are different approaches used in the literature that suggest methods of providing consumers of electricity to reduce their consumption. Some of the approaches [32, 33, 77, 44] focus on gamification to engage people more and change their behaviour in energy consumption in more enjoyable way. Other approaches [74, 29, 39, 23, 74, 29], focus on visualizing consumption data and presenting it to the consumers in a way that is easy to understand. Socialization is also used [99, 106, 101] by utilizing social networks to connect households and encourage energy conservation by comparisons, competitions among households.

Data analytics is applied on energy consumption data collected by smart meters. The analysis focused on detecting and recognizing different appliances and loads in the data. Profile detection algorithms [30, 79, 5] are proposed to detect events in the energy consumption time series data. Other methods [46, 78, 60] are used in order to extract each appliance consumption from the aggregated consumption data from the mains.

Typically, home appliances may run in one of several operation modes. Each operation mode is characterized by its running time and different cycles that the appliance go in. Activating an appliance in a certain operation mode consumes energy differently than other modes.

To the extent of our knowledge, the literature lacks approaches that focus on analyzing disaggregated power data and detect the activation of certain appliances, then classify each use of the appliance in one of its operation modes. Analyzing disaggregated power consumption data consumed by each class of device helps suppliers in the development and testing of evidence-based energy-efficiency policies and power conservation applications targeted at reducing consumer's spendings [38]. By doing this approach on disaggregated power data the consumer has the opportunity of getting feedback in a form of tips to change his behavior of selecting the operation time, and the operation mode for each individual appliance.

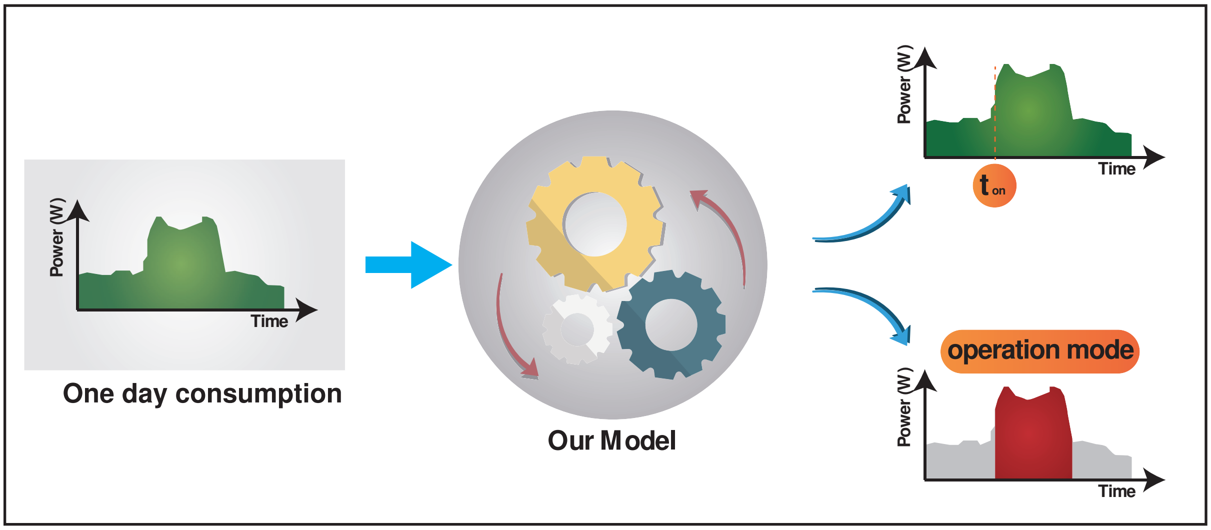


Figure 1.6: Inputs and outputs for the proposed model.

1.4.1 Thesis Approach

The model we propose conforms with the DR in terms of goals since our model's goals is to encourage consumers mainly to shift high consumption loads to off-peaks periods, and to lower their overall daily power consumption. To achieve these goals, we propose a system that processes the disaggregated power consumption data day by day for appliances. We focus on certain appliances that the DR concept can be applied on and the results can be observed easily.

To apply DR, we define what makes a consumption within certain period of time as high consumption so that we can apply DR to shift this high load and to lower its power level. Therefore, we focus on certain appliances that have similar behaviour in which they are turned on by human intervention, but they turn off on a timely basis depending on the operation mode that the consumer selects. Between these two time marks (turn on, and turn off) the appliance is in running state, performing different types of preprogrammed cycles to achieve its task. These cycles consume different levels of power and run for different periods of time, all depend on the appliance type and the operation mode that the consumer selected. This is what we call **Single Usage Profile (SUP)**. A **SUP** is defined as: The power consumption behaviour for a certain appliance running in a certain operation mode within specific period of time.

Figure 1.6 demonstrates the two DR goals achieved by our model. The input to the model is a single day power consumption for a certain appliance. The processing of the power consumption data has two main objectives:

1. **Find the time when the SUP starts:** The power consumption data is processed in order to find the turn on times for the appliance i.e. the start time of the SUP. By doing so, we obtain the time when the high load starts. Consequently, a recommendation generated by the system could be sent to the consumer if the detected time falls within the on-peak time advising the consumer to shift the load either before or after the current time to avoid higher energy prices. This can save the consumers up to 51.5% of their spending on energy [50] . Figure 1.6 shows the detected time t_{on} of the high load region (SUP).

2. **Recognize the operation mode of the SUP:** Most of modern appliances have the option to run in different operation modes. For example, a washing machine could be programmed with three or four different modes, each of them runs the inner components of the washing machine differently in order to meet the person's needs. Each of these operation modes has its own timing, cycles activated, and power levels during the activation time. Therefore, one way to apply DR is to recognize the mode that the detected SUP uses then the system recommends to the user to avoid using heavier modes on the long run. Figure 1.6 shows how our model recognizes the SUP and isolates in red, then it distinguishes its operation mode.

1.4.2 The Proposed Model

Our work focuses on building a web based prototype that resembles a tool belongs to the MDMSs. The main goal of this prototype is to monitor and analyze residential power consumption for certain appliances in order to give the household tips to lower the daily consumption. These tips are based on detecting the times when an appliance is activated so the system advises the user to shift the usage to off peak hours. The other way to give tips is to recognize the operation mode that appliances in activated with and advise the user to use lighter modes that consume less power instead of using heavier modes that consume more energy.

The prototype generates a group of households. Each household uses several appliances in daily basis with certain behaviour on when and how these appliances are used. The application monitors the power consumption data for each appliance using detection algorithms to detect the time when an appliance is activated so that a tip is given to the user to shift the use time. Also, recognition algorithms are applied to recognize the operation mode used at the detected time so that the application gives a tip to use lighter mode.

Our Model consists of five modules working sequentially to obtain the two goals that facilitate applying DR. Figure 1.7 shows the overall architecture of the model and how the flow of the execution is going top-down. The main modules of the model are Data Analysis Module, the Simulator Module, the SUP Detection Module, and two classification modules that we compare their results. These two modules are the DTW Classification Module and the Machine Learning Classification Module.

1.4.3 Contribution

To the extent of our knowledge, the combination of all these methods together is not done before in the literature. Analyzing existing dataset to extract behaviours of each appliance as configuration object. Then using this configuration to simulate the existing dataset with the ability o control these behaviours and inject certain bias in the simulated data. Then run a detection algorithm to determine the timing on when these behaviours take place and after that recognize these behaviors using two different approaches to classify the behaviours into different classes based on the analysis of the existing dataset. This combination of methods and techniques in not done yet in the literature.

1.5 Thesis Outline

The content of this thesis has been arranged into 10 chapters as follows: Chapter 2 describes the background and relevant work about this research area. Chapter 3 discusses the data analysis conducted on a publicly available power consumption dataset. Chapter 4 discusses the simulation of power consumption. Chapter 5 talks about the detection of appliances usages in the consumption data. Chapter 6 walks through the recognition of appliances profile using signal processing approach. Chapter 7 discuses the recognition of appliances profile using machine learning approach. Chapter 8 describes the technical implementation. Chapter 9 evaluates the performance of all the proposed approaches. Chapter 10 concludes the thesis.

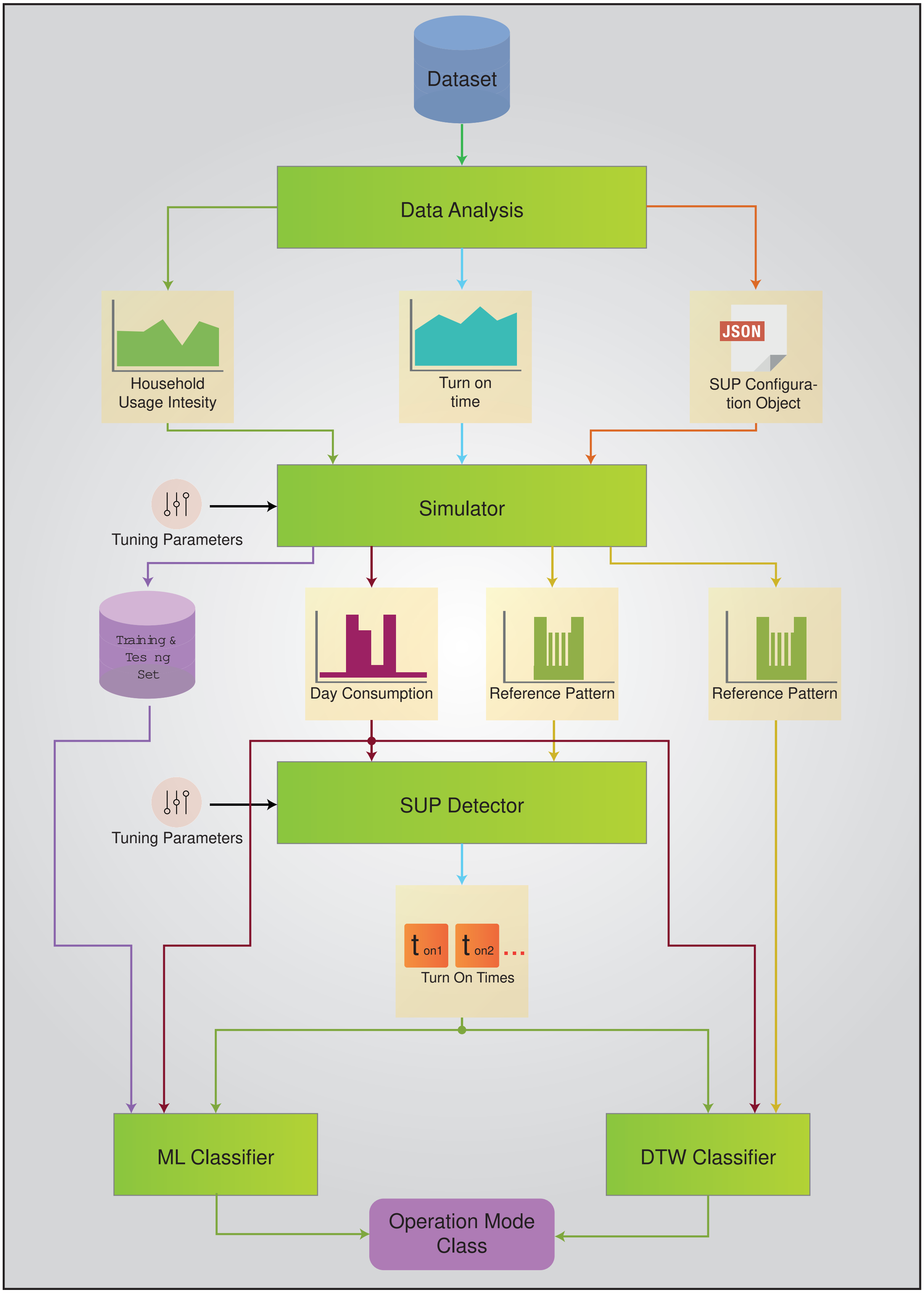


Figure 1.7: The overall architecture.

CHAPTER 2

Background And Literature Review

2.1 Background

2.1.1 Electricity Connectivity

This section discusses the electricity distribution inside the house and how the electricity consumption is monitored.

2.1.1.1 The House Main Service Panel

The utility company generates the power and distributes it to the main service panels installed in houses in a certain area. The main service panel is considered like the switchboard for all the electricity lines in the house. It receives the power from the incoming utility company power line and distributes it to each circuit which supply the various loads such as: power outlets, lights, appliances, etc. Every outgoing power line out of the service panel can be turned on or off at the main service panel using a Circuit Breaker. A circuit breaker is a switch that controls one outgoing line from the main panel. It is an auto/manual switch that can be switched by hand or automatically when the line is overloaded.

2.1.1.2 Electricity Meters

Each service panel comes with an electricity meter attached to it (before the circuit breakers) to monitor the entire house power consumption. There are two types of meters:

1. **General Electricity Meters (GEM):** This type of meter is installed before the the circuit breakers as depicted in Figure 2.1 . The meter shows power consumption as an accumulated reading from the time of installation. Other meters may store multiple readings such as power, voltage and current measured frequently. One example of this type of meter is shown in Figure 2.2.

2. **Smart Meters (SM):** A smart meter is an electronic device that logs the data of electricity consumption and communicates this data to the electricity supplier's central system for the purpose of monitoring and billing. Barai et.al. [12] demonstrated the functionalities of smart meters in the smart grid. The smart meter model is shown in Figure 2.4. Smart meters can communicate with sensors that are installed on an appliance. These

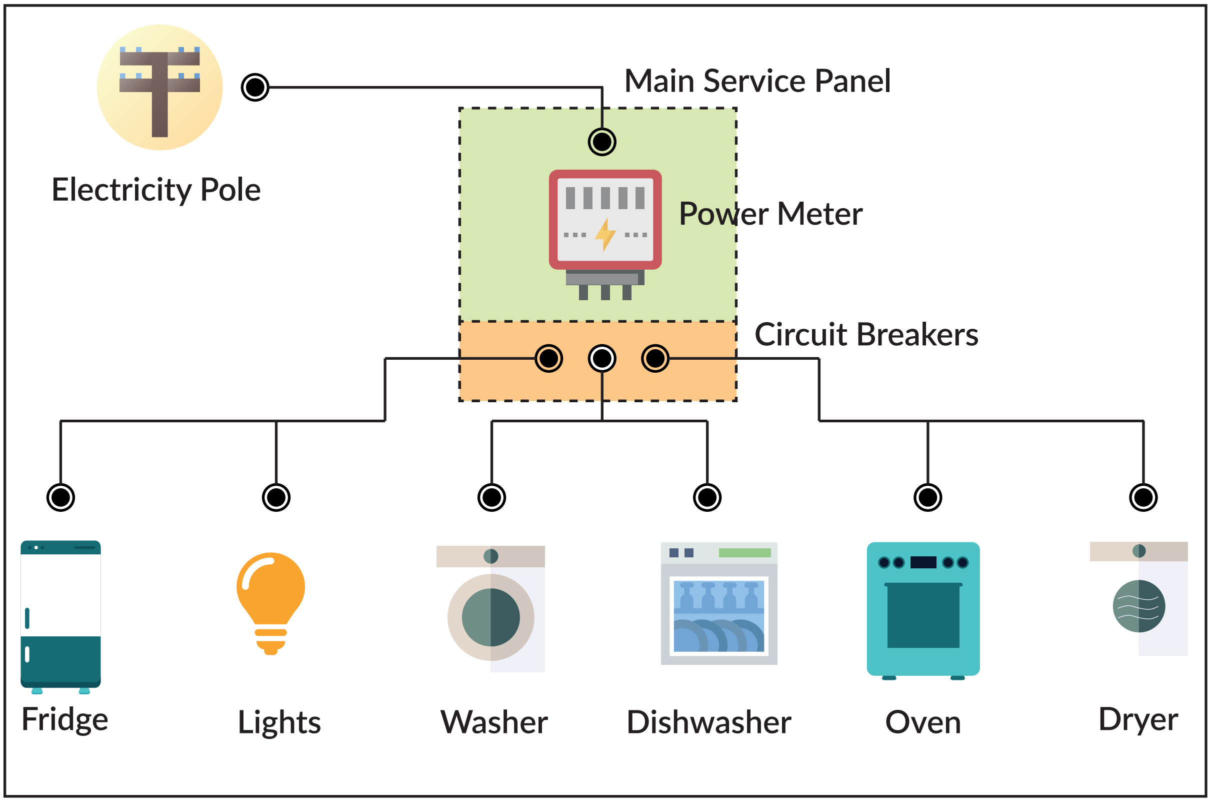


Figure 2.1: General Power Meter model. This model shows that General Power Meter is used to distribute power from electricity poles to house appliances.

sensors have the ability to collect data at fixed intervals from the appliance that they are attached to. This data includes power, voltage and current. The smart meter receives this data and logs it to a storage device. Additionally, smart meters enable two-way communication between the meter itself and the supplier's central system. This communication to the network may be established wirelessly through 3G, or via wired connections e.g. Power Line Carrier (PLC) which uses the existing power lines to transfer the sensor data. The Smart Meter can also receive control signals sent from the supplier for the purpose of applying power conservation policies. One example of these meters is shown in Figure 2.3.

2.1.2 Electricity Consumption Metering

Electricity Metering is the task of recording the power consumed by a certain load. This load could be a light bulb, a microwave, a dishwasher, or even the house as a whole. This task

Figure 2.2: Three-phase electromechanical induction meter, metering 100 A 240/415 V supply. Horizontal Aluminum rotor disc is visible in center of the meter.(online at bit.ly/2QH1i1i)

Figure 2.3: Newer retrofitted U.S. domestic digital electricity meter Elster REX with 900MHz mesh network topology for automatic meter reading and (EnergyAxis) time-of-use metering.(online at bit.ly/2QH1i1i)

is performed by the electricity meters we discussed in the previous section.

Since the late 18^{th} century as the use of electricity started to spread and the electrical grid expanded to reach residential areas, the concept of installing a meter that monitors the electricity consumption over time emerged. Sebastian Ferranti offered the first meter which was a mercury motor meter with a register so that consumers can read electricity consumption values [41]. These meters are important for residents and power suppliers. Residents want to minimize power consumption bills. Power suppliers rely on these meters to be aware of the load consumed by each household for billing purposes.

Traditionally, GEMs are installed in each house by the utility company in order to monitor

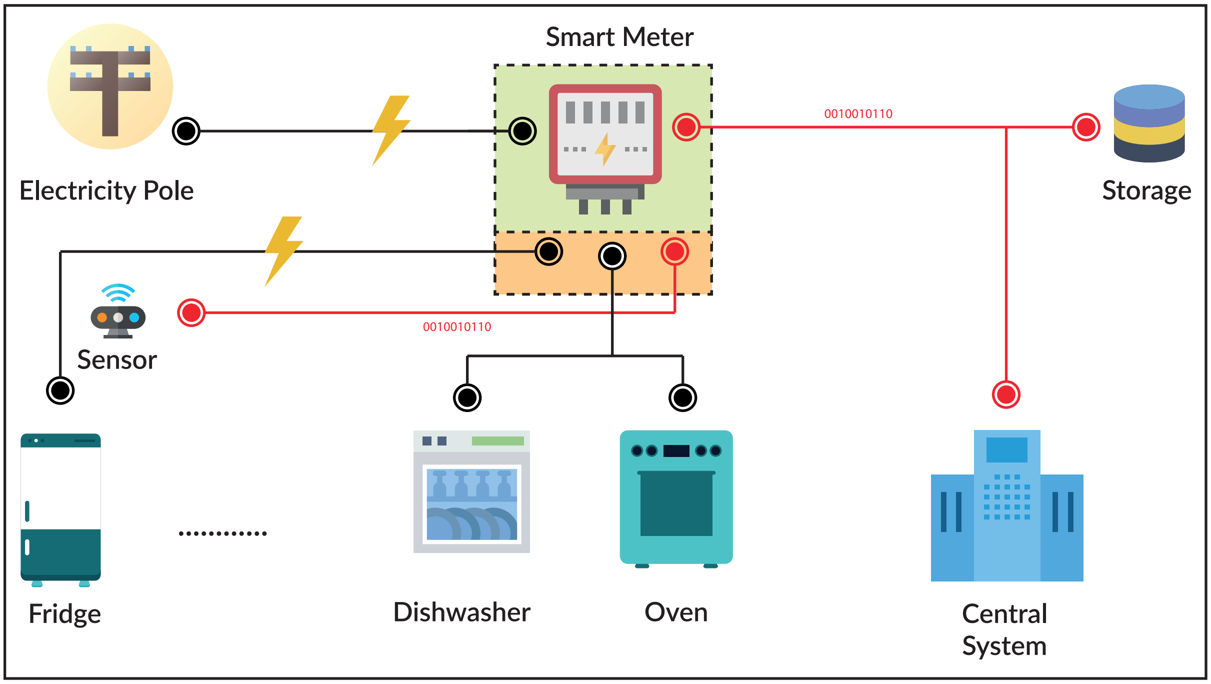


Figure 2.4: Smart Meter connection model. Smart meter act as General meters in terms of ability to log power consumption data. On the other hand, the model shows how the Smart Meter can connect to sensors installed on the appliance wall outlet and log consumption data for each appliance individually. furthermore, the Smart Meter connects two-way with the supplier central system.

the electricity consumption in the house. With the advent of the smart grid and the equipments that come with it in terms of metering devices and sensors such as smart meters SMs, the types and quantity of data differed. There are two forms of power consumption data:

- **Disaggregated Electricity Consumption Data:** This type of metering is done by a smart meter with the collaboration of special sensors installed on each breaker line or a single outlet that samples the power reading frequently and send them to the SM as shown in Figure 2.4.

- **Aggregated Electricity Consumption Data**: This is the traditional way of monitoring electricity consumption. The traditional meters are used to measure the power consumed on the mains line before the circuit breaker regardless of how the lines are distributed in the house as shown in Figure 2.1. This aggregate power reading could be transformed into disaggregated powers reading by applying Energy Disaggregation (ED) techniques.

> Energy disaggregation is a wide research area where algorithms are developed to decompose the aggregate power into the the different loads and appliances. Energy disaggregation, also referred to as a Non Intrusive Load Monitoring (NILM), is the process of using an aggregated energy signal, such as what is coming from a whole-home power (which is known as mains), to infer each of the different individual loads of the mains.

Consumers typically struggle to understand an appliance's power consumption because of the lack of the technical knowledge that prevents them from knowing which types of electrical components consume more power than others [36]. These misunderstandings of how energy is used in the house make it difficult for consumers to determine how to conserve it. Furthermore, people commonly overestimate the effectiveness of conservation measures which depend on short-term changes in behavior e.g., turning off lights when leaving the house. In contradiction, residents underestimate technology related solutions such as deciding to replace the house insulation or the use of energy-efficient appliances [36].

Reporting disaggregated power consumption data consumed by each class of device to the supplier is critical in terms of development and testing of evidence-based energy-efficiency policies and power conservation programs targeted at reducing capital spendings. Disaggregated data provide opportunities to the power companies to improve the power systems planning process, electric load forecasting, new ways of billing, and more accurately solving customer complaints [38].

With regard to residents, several studies [21, 56, 112] show that consumers immensely underestimated the energy used for certain uses while overestimating the energy used for appliances and other uses. Mettler-Meibom et.al [68] interviewed 52 households in Munich, West Germany, and asked them to approximate the proportional energy cost of certain power usages over others. They then compared these responses to the actual usage. The results showed that there is a wide gap between estimations and actual power usages as they underestimated power used for heating and overestimated the power used for lighting, cooking, and appliances.

2.1.3 Time Of Use Tariffs

Time Of Use Tariffs (ToUTs) are price policies that support consumer flexibility by realigning price signals in favor of more flexible energy use [92]. ToUTs aim to encourage consumers to shift their energy usage from high energy demand times to low energy demand times. Therefore, ToUTs divides each day (or week) into several periods depending on the demand rate, and assign prices to each period accordingly by setting cheaper energy prices in the low demand period compared to the high demand periods.

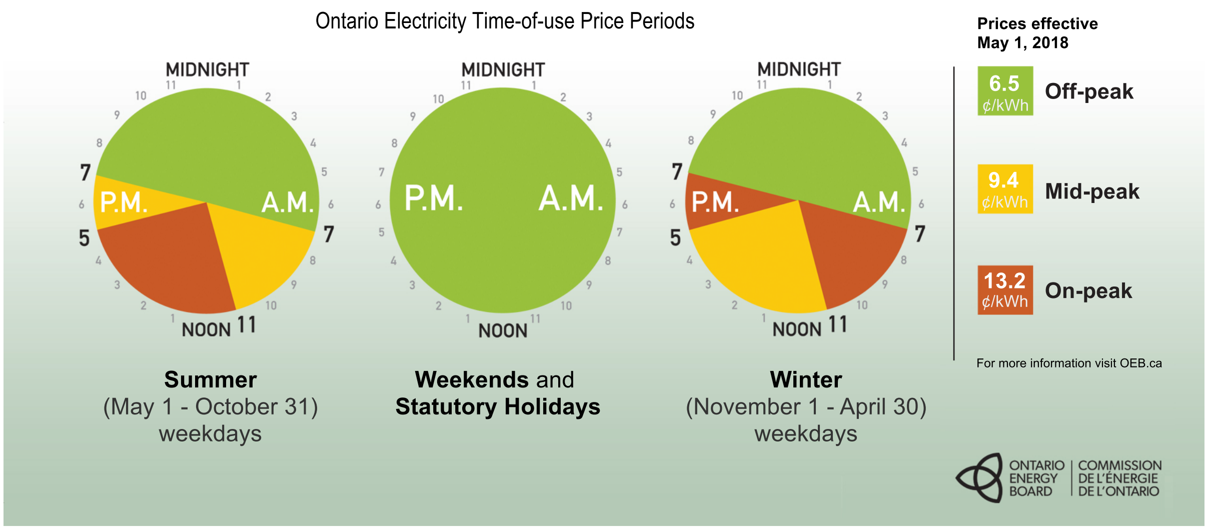


Figure 2.5: Ontario Time Of Use Tariff starting from May 1st, 2018. Ontario Energy Board website [76].

ToUTs differ depending on when the usage of the electricity takes place. Based on the Ontario Energy Board, there are three periods in Ontario:

- **Off-peak:** This period is when demand for electricity is lowest. Ontario residential areas and small businesses use nearly two thirds of their electricity during off-peak hours.
- **Mid-peak:** This period is when demand for electricity is moderate. These are the daytime periods, but not the busiest.
- **On-peak:** This period is when demand is highest. These are the busiest times of day –generally when people are very active cooking, and running heaters or ACs.

The overall objective of applying ToUTs is to shift energy demand from on-peak to off-peak hours, instead of reducing the total energy use. The design of ToUTs reflects the real energy generation expenses throughout the day.

2.2 Literature Review

2.2.1 Demand Response

This sections discusses different approaches in the literature to apply demand response in the residential electricity consumption.

2.2.1.1 Gamification

According to Deterding et al. [25] the definition of gamification is *"The use of game design elements in non-game contexts"*. They discuss on how to benefit from gamification in non-game applications. For instance, video games are designed with the purpose of entertainment, yet it can motivate players to engage with its elements in certain intensity and playing duration. Hence, adding gaming elements to non-game products and applications will encourage users to engage more in an enjoyable matter. For example, the work conducted by Flatla et. al. [32] focused on calibration in interactive systems like calibrating a new input device of a computer, which is tedious and boring process. They introduced calibration games that collects calibration data in an entertaining manner.

Gamification approaches can be considered as an engaging means of intervention that aims to stimulate certain desired behavior by tapping into intrinsic motivation as mentioned in Fogg's book [33], which defines intrinsic motivation as *"spontaneous energizing force based on activity or situation"*. This is considered very powerful in persuading people to do certain actions.

Gamification is one of the tools that is used in persuading consumers of electric power to reduce their consumption. This could be done through engaging them with a gamified environment that includes some gaming elements. Project **ChArGED** by Papaioannou et al. [77] describes the IoT-enabled gamification approach by employing a multi-channel plug-level meter, Near-Field Communication (NFC) tags, Bluetooth Enabled (BLE) beacons and the user smartphones to collect disaggregated power consumption data. They used this data to introduce the GreenCup challenge game and weekly competitions among teams of employees. **Social Power**, is a mobile application developed by Wemyss et al. [107]. Its objective is to stimulate social engagement in households, encourage change of personal behavioral in order to save electricity by forming groups of neighbors in competitive or collaborative environments with a points system. Ro et al. [90] collaborated with Cool Choices, a non-profit organization located in Madison, Wisconsin to develop the **Cool Choices** game, in which players submit environmental sustainable cards (actions) and receive monetary awards. Gustafsson el al. [44] focused on determining if post game effects (sustainable) on behavior can be achieved through gamification. They proposed **Power Explorer** game that uses real-time sensing for appliances cables. The game is capable of providing instant feedback to consumers when consumers switch on/off home appliances.

2.2.1.2 Visualization And Eco-Feedback

Visualization is considered one of the most effective approaches used in the literature [48], where systems are developed to visualize consumption data in various forms suitable for technical and non-technical people so that everyone can understand the data and the decision making will be more accurate.

The literature shows that depiction of energy consumption and saving advisories are key concepts to save energy and leads to change of behavior that in people is performing in energy consuming, lowering their power demands, and upraise energy consumption awareness for them [31]. Through previous studies [74, 29, 39], it is commonly expected that by offering visualized feedback of the household electricity consumption at a certain frequency, a range between 5% and 15% of the household consumption can be saved [23]. These savings are the outcome of the behavior changes in energy consumption of users who consume less or more efficiently based on the feedback they get.

Eco-feedback is defined as any technology which provides feedback on individual or group behaviors with a target of reducing environmental impact. An important way to change individual behavior is to render energy consumption visible through personalized feedback [2]. This feedback may take many forms, and is typically designed to motivate individuals to increase sustainable behaviors. For example, goal setting and public commitment may both enhance motivation [37]. Sending alerts to employees screens when they left their screens open in meeting time was efficient [74]. Fan et al. [29] suggested ranking of users based on their habits based on rules extracted from the Internet about efficiency of using appliances.

Electricity usage data visualization not only allows users to intuitively understand the household electricity consumption, but also gives the users the ability to optimize home energy efficiency by following up with their habits of using electricity, in order to achieve home energy saving, efficiency and convenience. Ghidini et al.[39] introduced a mobile application to distinguish and display power usage when there is no one in the house and when there is no one outside the house using GPS. Apperley et al. [8] developed an iOS application for an iPad that displays energy sources as icons (water, wind, coal, etc.). These icons move in proportional speed with the amount of power that they produce. Holmes et al. [47] introduces the idea of converting usages (in KW) the number of trees need to be planted in order to absorb the produced CO_2.

2.2.1.3 Socialization

Monitoring technologies for electricity consumption by its own are mostly not enough to make the desired behaviour change regarding consuming power [23]. This section presents some of the work on the use of Online Social Networks (OSNs).

These days, OSNs interconnect billions of people together e.g., Facebook now has over 2.19 billion active users [99]. Studies [106, 101] demonstrate that, through social networks, users read other people's postings, play group games, post comments on pictures, show admiration, and add to their own activity log many times daily. These networks provide a powerful mechanism that delivers dedicated applications to groups who have similar tastes or opinions and thoughts in the real-world in a manageable and pleasant manner. There may be a possibility in leveraging the engaging power of small applications, offering rich social interactive features to help change energy behaviour.

Using existing social networks may be considered a solution to developing certain applications on top of it, so that the underlying social infrastructure can be utilized. Foster el al.

[34] developed **Wattsup** a Facebook application using the Facebook Developers Kit (FDK) allowing users to compare domestic energy consumption on Facebook. **Wattson** home energy monitor devices [34] was used to collect usage data from household wirelessly and rank usage among friends. Grevet et al. [42] developed a web application to display personal use graphs, and the ability to compare with other individuals anonymously in a college dorm. Petcov et al. [80] focused on comparative feedback by developing the mobile application **EnergyWiz** that enables users to compare with consumer's past performance, their neighbors, or contacts from social networking sites e.g. Facebook and other EnergyWiz users.

2.2.2 Power Consumption Data Simulation

One of the fundamentals of the smart grid applications is the electricity consumption data. This data is collected mainly by smart meters as we discussed in Section 2.1.2. In some cases, real data that is collected by smart meters are not available. In this situation synthetic data is the solution to overcome this limitation. This section discusses several approaches to simulate electricity consumption for residential uses.

A behaviour based load profile generator is proposed by Pflugradt et.al. [81] to generate realistic load profiles for every single household in a low voltage grid using MATPOWER tool. Load profiles are generated based on psychological models that describe domestic load profiles to be results of household residents' daily activities. The generated loads are disaggregated loads for different appliances. These loads show fewer power fluctuation than real power readings. Kong et.al. [57] proposed a rule based domestic load profile generator that allows the users to choose appliance profiles and add them to the household. The simulation is controlled by rules that represents the typical resident load patterns and schedules and appliance's attributes.

Yao et.al. [114] introduced a Simple Method of prediction of daily Load Profile (SMLP). They applied cluster analysis method based on several scenarios of occupancy patterns that include information about each individual in the household such as job type, number of residents, unoccupied period of the house, etc. SMLP can be applied at both regional (macro) and house (micro) levels. A similar approach is used by Richardson et.al. [89] to generate high-resolution synthetic lighting demand data. The model depends on two physical/environmental input factors: Outdoor Irradiance, and Active Occupancy within the house.

An Adaptive Markov Chain Model (MCM) approach is presented by Zufferey et.al [115] to generate appliance level power consumption. This approach takes into consideration the medium to long-term seasonality using low number of markov states without decreasing performance. Hoogsteen et.al. [49] presented a novel, open source, Artificial Load Profile Generator (ALPG) developed specifically to test and compare the performance of Demand Side Management (DSM) methodologies. They considered several parameters to control the ALPG such as: Penetration of emerging technologies like electric vehicles, geographical location of the neighborhood, and predictability of persons.

2.2.3 Load Profiles Detection

In Section 2.2.1 we discussed different approaches applied in order to conserve electricity consumption in residential areas. This section discusses the techniques used to analyze the electricity consumption data so that end user applications could be built on top of these approaches. These techniques are mainly about event detection in time series data. The events that we are interested in are the events of activation of appliances.

Event detection algorithms are used to detect starting and transition states for loads/appliances from aggregated power consumption data, then to extract each load profile separately. According to Anderson et.al. [7], there are three types of these algorithms: (a) Expert Heuristics, (b) Probabilistic Models (c) Matched Filters.

2.2.3.1 Expert Heuristics Algorithms

Expert Heuristics algorithms assume that a prior knowledge of the appliance that is detected to be exist. Farinaccio et.al. [30] proposed a rule-based approach to detect the transitions of a selected appliances (turning on and off). The ”state change detection” rule scanned the difference periods of time of the total power data and compared with pre-determined power periods for the start and end (on and off) associated with each appliance. Such a technique is used by Baranski et.al.[13] using dynamic programming and genetic algorithms. Hart et.al.

[45] predefined certain periods of time as Steady State Periods where the power value through these periods doesn't vary more that a certain tolerance value. They used this definition to detect the transition between states with different power levels to indicate the activation of appliances.

2.2.3.2 Probabilistic Models

Probabilistic Models make decisions about the occurrence of certain events by providing a probability of how likely an event will happen. A training step is required to learn the statistics models of different power datasets [6]. Luo et.al. [64] introduced the concept of a Generalized Likelihood Ratio (GLR). This approach calculates a "decision statistic from the natural log of a ratio of probability distributions before and after a potential change in mean". This approach involves training of several statistic parameters offline. Pereira et.al. [79] used a GLR variant to detect events of turning appliances on/off and to label a data set. The study reported that it can detect 95% of events. Bergesr et.al. [16] used another variant of GLR that uses fewer number of statistic parameters to be trained. Also they introduced a voting process to maximize the detection statistic. Jin et.al. [54] proposed event detector that utilizes a Goodness of Fit Chi-squared test (GoF) for detecting load activation using the power mean calculation, then followed by a change point detector that estimates the transition point of the signals using the first harmonic component of the power signals extracted by applying the Short-time Fourier Transform (SFT) on the power signal.

2.2.3.3 Matched Filters

Matched Filters involve a known signal known as the (mask signal or template signal) to be correlated with unknown signal to detect the occurrence of the template in the unknown signal. Weiss et.al. [105] proposed AppliSense algorithm that uses a database of load templates in order to detect appliances presence within unknown power signal. They utilized Derivative Filters approach. Alcala et.al. [5] proposed an approach using Hilbert Transform to extract the envelope of the current signal. Then, by using Average Filters, Derivation Filters, and thresholding to cut off the signal, transition events is detected. Remscrim et.al. [88] and Shaw et.al. [94] showed approaches that use Field-Programmable Gate Array (FPGA) to monitor power signals and extract spectral envelopes from several harmonics, then a Kalman Filter is used to track events. Other studies [95, 59, 58] used pre-stored load signatures to detect appliance's activation.

2.2.4 Load Profiles Classification

NILM algorithms basically consist of two major steps: Load Detection and Load Classification. Once a load have been detected, we need to classify this load into the correct appliance. Consequently, we need to extract load profile (signature) from the power signal. Load signatures should be unique for each appliance in the house, and they can be extracted during steady states while the appliance is in operational modes, or transient states while an appliance transition between two operational modes.

Hassan et.al [46] evaluated appliance load signatures to disaggregate residential total energy use and predict individual appliance profiles. They built their approach based on V-I trajectory which is "the mutual locus of instantaneous voltage and current waveforms" for precision and robustness of prediction in classification algorithms used to disaggregate residential overall energy use and predict constituent appliance profiles. Parson et.al. [78] proposed an approach using Iterative Hidden Markov Models (IHMM) in which prior known models of general appliances are tuned to specific appliance instances using load signatures extracted from the total load.

Multi signatures are used by Liang et.al. [60] where different snapshots and delta form signatures are extracted including Current Waveform (CW), harmonics (HAR), active and reactive power (PQ), admittance waveform (IAW) etc. They then apply multiple algorithms on these signatures such as Least Residue (LR), Integer Programming, Genetic algorithm and Neural Nets. On the presence of a event in the aggregate load, all the mentioned signatures are extracted; then each algorithm classify the signature into an appliance class. Lastly, a voting algorithm is applied to decide the appliance class among algorithms outcomes. A Committee Decision Mechanism (CDM) is chosen for voting.

2.2.4.1 Machine learning algorithms

Machine learning algorithms are emerging in the context of load signature detection with both supervised and unsupervised algorithms. Barsim et.al. [15] proposed an approach that uses the typical event-based NILM system with only unsupervised algorithms to eliminate the need for training stage. The event detector is a grid-based clustering algorithm to segment the power signals into transition-state and steady-state segments. They extracted Macroscopic Features from the detected events and used them in a Mean-Shift Clustering algorithm. They proposed another approach [14] that uses bucketing technique and Expectation Maximization (EM) clustering for event detection. Then Mean-Shift clustering is utilized to detect recurrences. Kang et.al. [55] used Probabilistic K-Nearest Neighbor (PKNN) to infer the device states from home appliances electrical power usage signal and also from sensor data including temperature and humidity. Prudenzi et.al. [84] proposed a procedure that provides three different sequential back propagation Artificial Neural Networks (ANNs) to process the load shape and identify load signatures. Then the classification is performed by an unsupervised network implementing the Self-Organizing Map (SOM) of Kohonen.

2.2.4.2 Dynamic Time Warping (DTW)

Dynamic Time Warping (DTW) [17] is an algorithm used to measure the similarity between two signals. DTW prevents the impacts of the potential location shifting and/or temporal scaling between the compared signals. DTW is chosen as similarity measure for several reasons: First, It is easy to implement, and the enhancements [73] to the original algorithm are plenty and easy to implement as well. Second, DTW is quite robust and effective and gives good results [104] proofing its robustness. Third, it is a good fit for the nature of the time series that have slight distortion it terms af amplitude or in phase shift. Accordingly, DTW is a good choice for measuring similarities between functions that has differences in amplitude and time warping.

Liao et.al. [61] proposed an approach for appliance load classification using DTW. The approach relies on load signature DB and predetermined DTW thresholds (distances) for different appliances. Liu et.al. [62] used a nearest neighbor transient identification method to identify the appliance creating the Transient Power Waveform (TPW) sample time-series, then the DTW-based integrated distance is utilized to calculate the similarity of TPW signatures and a template time series for an appliance. Wang et.al. [103] contributed an approach which uses Iterative Disaggregation based on Appliance Consumption Pattern (ILDACP). This approach combines Fuzzy C-means clustering algorithm to detect appliance operating status, and DTW search that identifies single energy consumption based on the appliance typical power consumption pattern (a template pattern).

Throughout our review in the literature, we found that the literature so far lacks approaches [83] that focus on disaggregated power data analysis that focus on detection of the activation of certain appliances, then classify each use of the appliance in one of its operation modes. In this work, we focus on analyzing disaggregated power consumption data consumed by each class of device to detect the activation time for some appliances and determine the operation mode used in each activattion. This serves as a basis to support consumer feedback applications which target behavior change aiming for reducing energy expenses and lowering the energy demand.

CHAPTER 3

Data Analysis

This chapter discusses the statistical analysis performed on a publicly available Power Consumption DataSet to understand the characteristics of the different loads that resemble the dataset.

3.1 The Dataset

This section focuses on the analysis of a Power Consumption DataSet (PCDS) collected by a smart meter as illustrated in Figure 2.4. The data represents electricity measurements such as Voltage (V), Current (I), Power (P), Energy (E). Smart meters periodically collect the data throughout the sensors connected to the Smart meter. The frequency of data collection is known as the Sampling Frequency (f_s). If the Smart meter receives readings every second from the sensors then the Sampling Frequency f_s is $1Hz$.

3.1.1 Publicly Available PCDSs

This section presents the analysis of one of the publicly available PCDSs. The analysis aims to understand the characteristics of loads and appliances over time when they are turned on. The obtained characteristics help to formalize the behaviour of each appliance/load so that we can simulate this behaviour later on and generate synthetic data that have the same behaviour of the dataset.

3.1.1.1 Data Collection

The Rainforest Automation Energy (RAE) dataset [66] represents power consumption data of appliances. The dataset was published to help smart grid researchers evaluate their solutions that assume disaggregated data [67]. Figure 3.1 demonstrates the data disaggregation for one day in house 1.

The currently available release of the RAE dataset contains 1 Hz data from two residential houses in Burnaby, BC. The schematic hardware model used to collect this dataset is illustrated in Figure 3.2 which comprises of two functionalities:

1. Real-Time In-Home Display for the current usage in terms of Kilowatts/hour and cost in

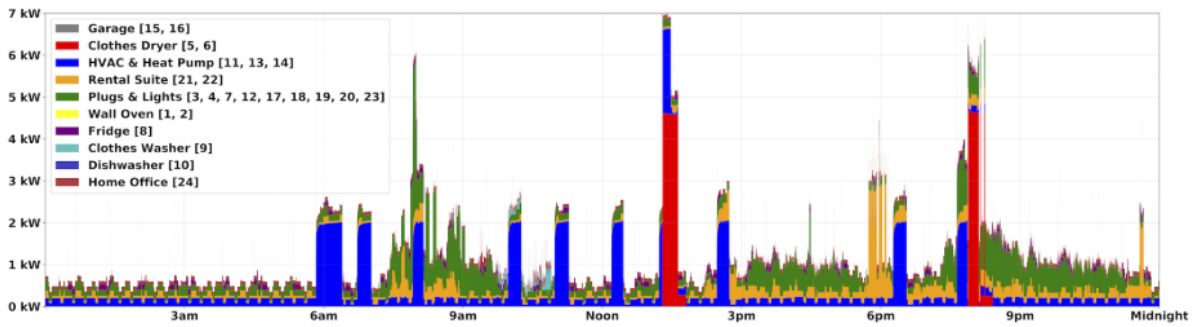


Figure 3.1: Plot of all disaggregated readings over full day on Sunday, 20 March 2016 for House 1 [66].

Dollars/hour. This functionality is achieved by connecting the Smart Meter [52] to the In-Home Display [86] through the ZigBee protocol. With this connection the SM sends the readings to the In-Home Display every 15 seconds.

2. The Breaker Panel submeter [24] captures the instantaneous power consumption readings for each port, while each port is connected to an appliance or a load line. Once the readings are captured, the data is sent through a USB cable to a Data Acquisition Unit, Raspberry Pi [87], which logs the gathered data into a storage device in a CSV formatted files.

3.1.1.2 RAE Data Description

The RAE dataset contains more than 11.3 million readings. There are up to 24 sub-meters, one for each breaker in the main distribution panel. The Sampling Frequency $f_s = 1Hz$, captures several electrical measurements such as power, voltage, etc. Figure 3.3 shows a snapshot of the raw data.

In our study we focus on the power (Watts) and energy (kWh) readings only, since these readings are the most commonly used to describe power consumption and consumers are familiar with them [53]. Energy is the total amount of work done over a period of time, while power is how fast a work could be done, or in other words, power is an instantaneous measurement of work at a moment of time [27].

Each row in the dataset corresponds to a sample of readings at a specific time. The first column (Unix *ts*) represents the Unix time stamp for that particular sample. The IHD column

Figure 3.2: RAE data capturing hardware setup [67].

shows the reading of the data that the smart meter communicated to the In-Home Display which is transmitted every 15 seconds. The NULL value indicates no reading. The Mains column corresponds to the total power consumption for the house at the time stamp specified in the first column, which is the summation of column data that appears after the Mains column. The columns $\{sub_1, sub_2, ..., sub_n\}$ are the power readings for the sub-meters in the house where n represents the number of sub-meters in the house. All the power readings in the dataset $\{Mains, sub_1, sub_2, ..., sub_n\}$ are measured in watts.

3.2 Data Analytics

One crucial step in this work is to gather information about different electrical loads in such a way that the information will help us to build our models and validate our proposed algorithms. The main problem with this process is the large number of electrical loads that could be available in residential houses. The available data is limited and does not represent the spectrum of electrical loads. This will lead to difficulty in analyzing these different types of

Unix Ts	Ihd	Mains	Sub1	Sub2	Sub3	Sub4	Sub5	Sub6	Sub7	Sub8	Sub9	Sub10	Sub11	Sub12
1,454,832,024	null	629	0	1	0	2	0	0	5	0	0	0	141	100
1,454,832,025	null	628	0	1	0	2	0	0	5	0	0	0	140	99
1,454,832,026	null	627	0	1	0	2	0	0	5	0	0	0	140	100
1,454,832,027	null	627	0	1	0	2	0	0	6	0	0	0	140	100
1,454,832,028	null	626	0	1	0	2	0	0	5	0	0	0	140	100
1,454,832,029	638	627	0	1	0	2	0	0	5	0	0	0	140	100
1,454,832,030	null	632	0	1	0	2	0	0	5	0	0	0	140	100
1,454,832,031	null	624	0	1	0	2	0	0	5	0	0	0	140	100
1,454,832,032	null	625	0	1	0	2	0	0	5	0	0	0	140	100
1,454,832,033	null	628	0	1	0	2	0	0	5	0	0	0	142	100

Figure 3.3: A snapshot from House 1 shows an example of the shape of the data. The columns $\{Sub_1, Sub_2, ..., Sub_n\}$ are the power readings for the sub-meters in the house.

	House1	***House 2***
Annual Energy Consumption (kWh)	7816	2959
Daily Energy Consumption (kWh)	21	8
Monthly Energy Consumption (kWh)	642	243
Total Energy Consumption (kWh)	1542	478
Number of Occupants in the house	3	3
House Area (m^2)	100	105
Number of collection days	72	59
Start Date of the collection	February 7^{th}2016	September 13^{th}2017
End Date of the collection	May 7^{th}2016	November 11^{th}2017

Table 3.1: Basic information about the two houses.

loads. To tackle this problem, we introduce a classification for the electrical loads in residential houses such that we group together all the loads which share similar characteristics.

3.2.1 Analysis of Aggregated Loads

This work focuses on appliances that represent the three classes of loads presented in Table 3.2 . These selected loads/appliances are dishwasher, clothes washer, clothes dryer, refrigerator, and lights. Figure 3.4 shows the daily average energy consumption for selected appliances during the entire data collection period.

From Figure 3.4 we make several observations. Firstly, the chart shows five main values out of the 24 loads in the house. Mains, as we mentioned, is the total consumption for that day

Load Type	**Examples**	**Selection**
Timer Based Loads	Air conditioning, water heater, refrigerator	refrigerator
Preprogrammed Loads	Dishwashers, Ovens, Broilers, washing machines, tumble dryers	Dishwashers, washing machines and tumble dryers
Variable Loads	electric cooktop, lighting, television, hair dryer, laptops, and almost the rest of house appliances	Lighting

Table 3.2: Classification for the electrical loads based on operation behavior.

but since we show only five loads the summation of the loads is always less than the mains reading. Secondly, there was no data between February 16,2016 to March 5, 2016. Thirdly, continuous usage is seen for the entire period for the refrigerator. The refrigerator belongs to the Timer Based Loads class. The average daily energy consumption is around 1.3kWh/day and the consumption ranged between 1.05kWh to 1.6kWh. The deviation of the daily average is 0.12kWh which reflects minor differences in refrigerator usage. Fourthly, Preprogrammed Loads (e.g. dishwasher, clothes washer and dryer) have intermittent usage ranging from very low amounts of consumption due to a lack of usage to high amounts of energy consumption. This behaviour is expected since the appliances in the Preprogrammed Loads category are typically not used on a daily basis. The actual energy consumption varies among households, but generally, the households show intermittent usage. Lastly, a varying continuous usage in the whole period is observed for the kitchen plugs, which is a Variable Loads class member. These plugs are used by several kitchen devices such as coffee maker, toaster, water kettle, microwave, range hood, sandwich maker, etc. These devices do not abide to any fixed usage policy since usage is determined by the individual. We observe that there are highs and lows, high deviation between days, which is approximately 0.21kWh compared to the maximum 0.9kWh and minimum 0.01kWh.

In terms of general behavior for the total energy consumption for the houses, Figure 3.5 shows the total energy consumption for the two houses as the summation of all sub-meter consumption in each hour of the day for all the readings. The chart splits the energy consump-

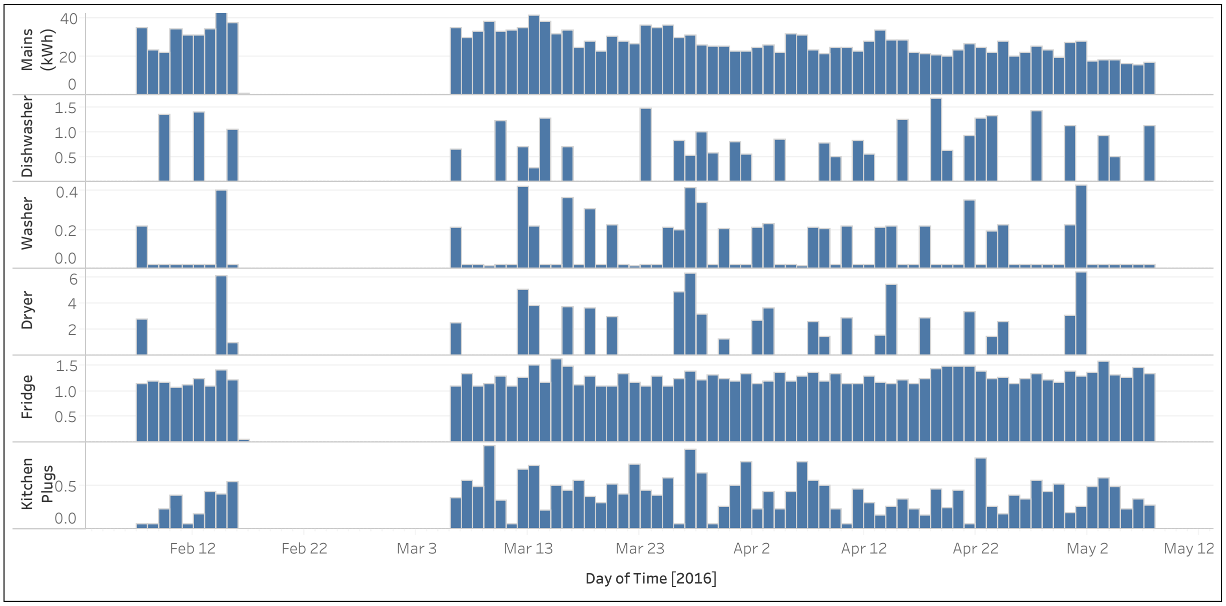


Figure 3.4: Daily Energy consumption for selected appliances and plugs in House1. The x-axis shows the day of time, the y-axis shows energy consumption in (kWh).

tion by day of the week and distinguishes weekdays from weekends as well. We observe that consumption on the weekends represents 29% to 49% of the total usage, while each weekday ranges between 13% to 20% of the entire consumption. One interesting observation in this graph is that there is an obvious increase in the energy consumption in certain hours of the day, compared to the timespan surrounding this increase. The first surge is at 6-8 AM, and the second surge is at 7-9 PM for both houses. These surges reflect behaviour typically associated with waking up. The later period likely corresponds to the time when household members have returned home.

3.2.2 Analysis Of Individual Loads

Our discussion of load consumption is based on load profiles. We introduce three types of load profiles: Single Usage Profile (SUP), Daily Usage Profile (DUP), Weekly Usage Profile (WUP). **Single Usage Profile** (SUP) represents the power consumed by an appliance from the moment of turning it on to the moment of that it is turned off. Hence, SUP with $f_s = 1Hz$ is $SUP = \left\{p_i\right\}_{i=1}^{t_e}$ where p_i is the instantaneous power reading at time i, t_e is the turn off time of the appliance and $i \in [1, t_e]$ represents the i^{th} sample. The **Daily Usage Profile** (DUP)

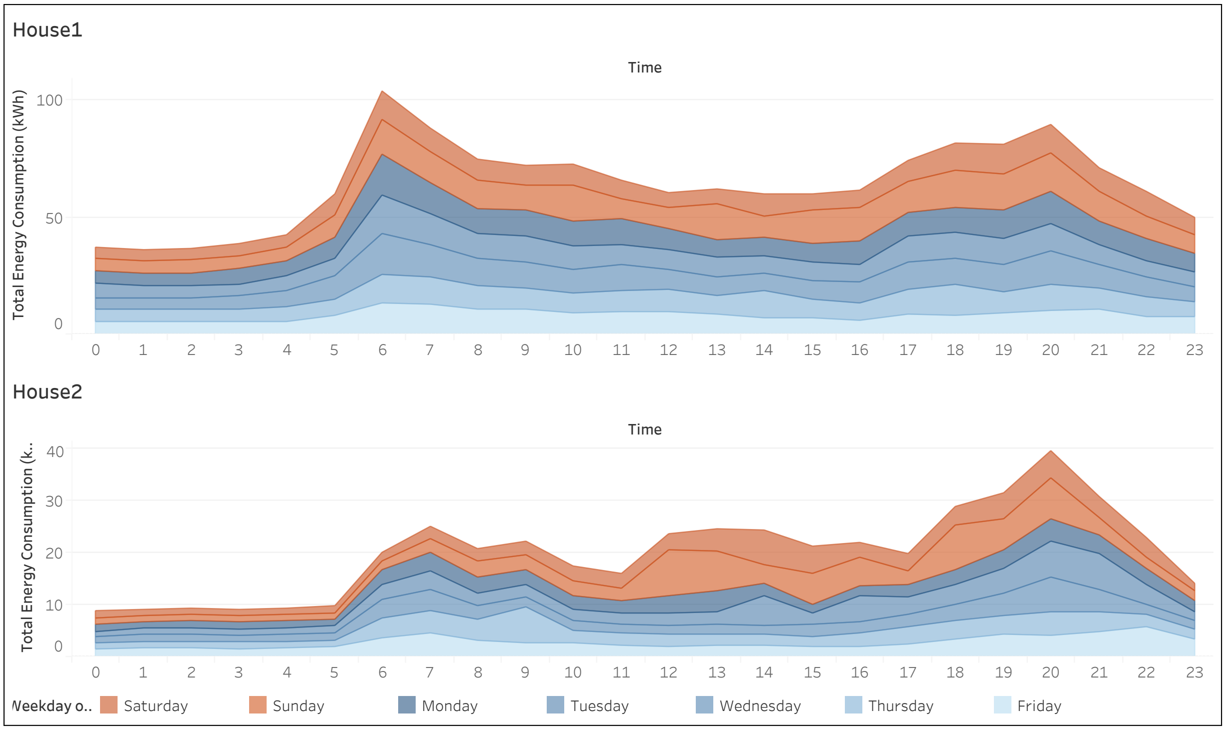


Figure 3.5: The total energy consumption for the two houses distributed by hour of the day, showing weekdays in blue and weekends in orange.

corresponds to the power consumption of a single load/appliance in a single day based on a sampling frequency. For example, a DUP with $f_s = 1Hz$ is $DUP = \{p_i\}_{i=1}^{86400}$, where p_i in the instantaneous appliance power reading at time i, and $i \in [1, 86400]$ represents the number of samples in a day with the determined sampling frequency f_s. Lastly, **Weekly Usage Profile** (WUP) is similar to DUP, but for a period of a single week.

3.2.2.1 The Refrigerator

A refrigerator belongs to the Timer Based Loads class. A refrigerator 's power consumption is primarily associated with two main circuits: Compressor and Defroster. Figure 3.6 shows the power consumption graph over three hours of operation, during which both circuits were activated.

Figure 3.6 shows four spikes as seen with high amplitudes. These spikes are called Inrush Current [63]. High inrush current is the result of the maximum instantaneous input current drawn by an electrical component each time it is turned on. It is always higher than the normal drawn current due to the high impedance of the device, which is the equivalent of resistance in the direct current systems [82]. Thus, it could happen for an appliance level, or for a certain component within the appliance such as the compressor or the defroster in this case.

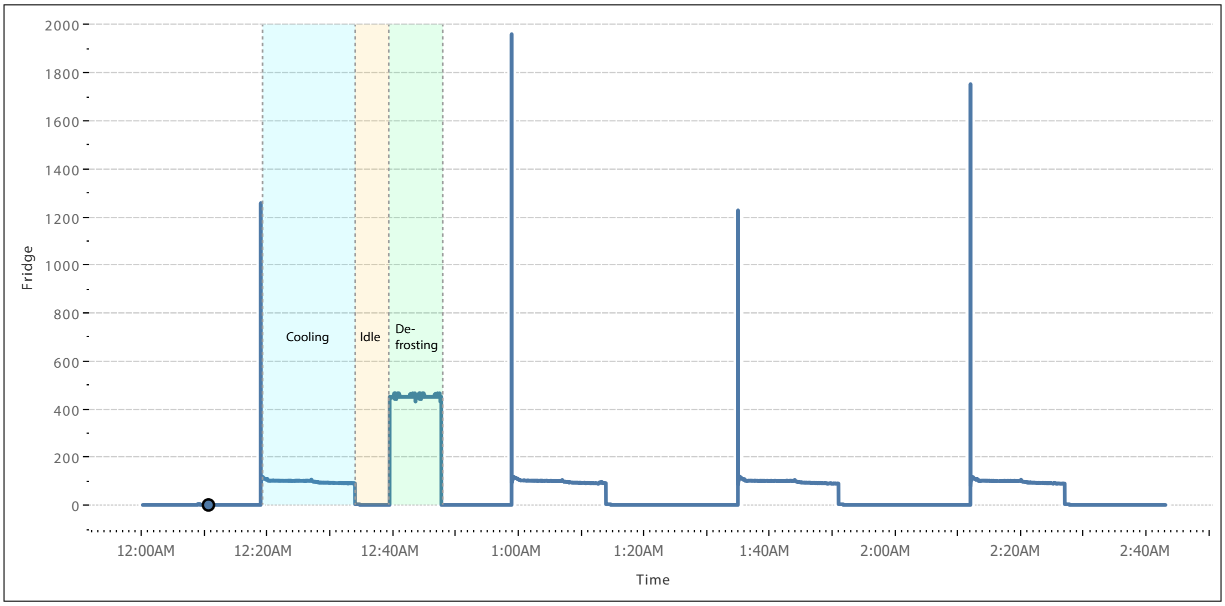


Figure 3.6: Power consumption for a refrigerator within three hours.

During the operation of the refrigerator, three different states took place: Cooling, Defrosting and Idle, which can be seen in Figure 3.6. The Cooling state appears four times at time intervals (12:18 - 12:34), (12:58 - 1:14), (1:34 - 1:50) and (2:11 - 2:26) with power consumption of 106W. In this state the compressor of the refrigerator is turned on, and thus we observe the inrush current at the start of each occurrence of the cooling state with power consumption at 1200W-2000W for only one second. Turning the compressor on drives the interior temperature down until it reaches a certain lower threshold when the thermostat interrupts the compressor and turns it off. After that, the refrigerator is in the Idle state. An example of this is seen at the interval (1:14 - 1:34), where the main components of the refrigerator are turned off. At this point the power consumption falls as low as 4W.

The timing of the Cooling and Idle states is predictable since a thermostat regulates the timing of the state. The defrosting state occurs when the heater ignites in order to get rid of the accumulated frost on the evaporator and thus the timing of the state is less predictable. The duration of the defrosting operation takes about 10 minutes. The defrosting state is presented in Figure 3.6 in the time interval (12:39 - 12:49) in which the power consumption is around

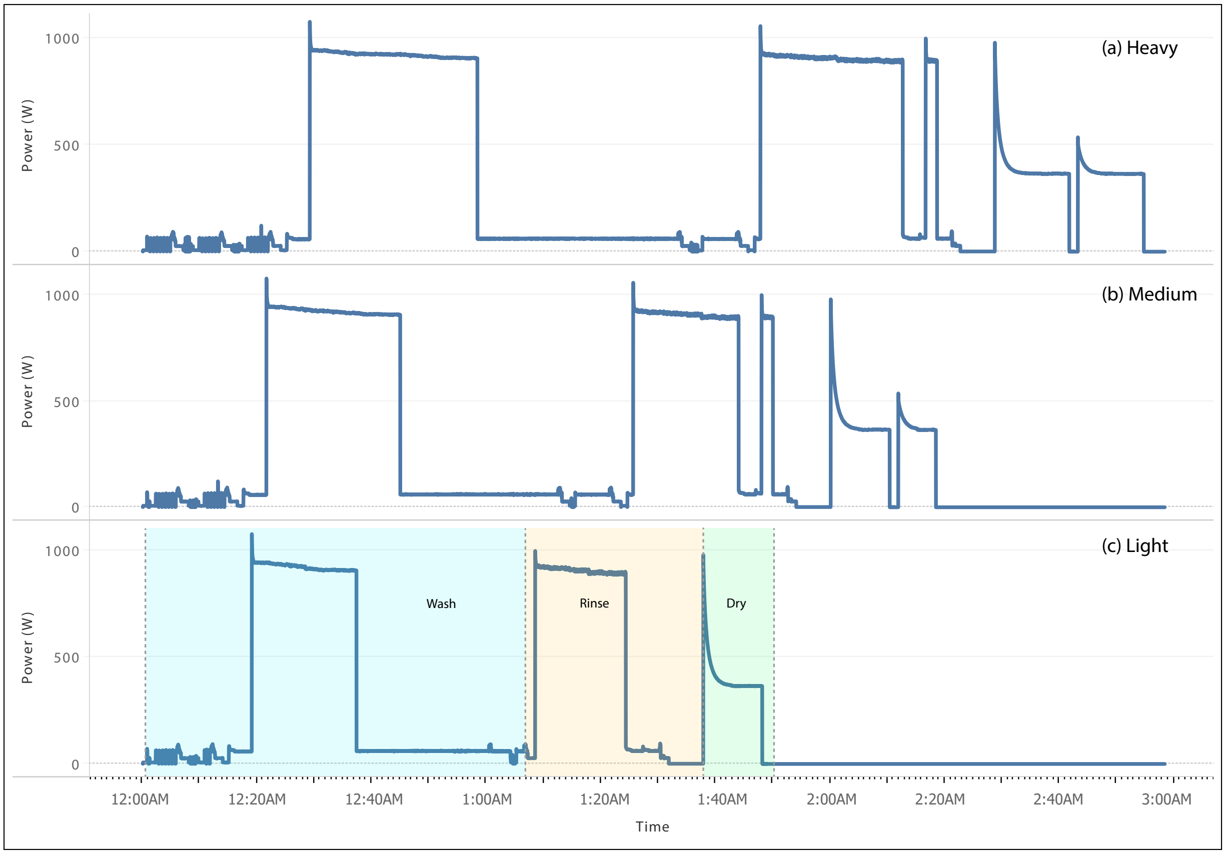


Figure 3.7: Single use (SUP) for the dishwasher showing three modes of operation: (a) Heavy, (b) Medium, (c) Light.

450W. It is higher than the cooling state as the defroster utilizes a heating element.

3.2.2.2 The Dishwasher

The SUP for the dishwasher is shown in Figure 3.7. The dishwasher has three main operating modes: Heavy, Medium, and Light. Each single use of the dishwasher has three operations: Wash, Rinse and Dry, regardless of the operation mode. For the Light mode the first 70 minutes are associated with the wash state. The rinse state follows from minutes 70 to 97. The last is the dry state which ends at minute 108.

The first 15 minutes of the operation during the wash state is the pre-wash phase. Pre-wash involves filling the dishwasher with water and spraying the water through jets to get the first round of spray with regular water temperature. The dishwasher consumes in this period around 10–80 W. The wash state takes place when the water is heated by the heating element inside

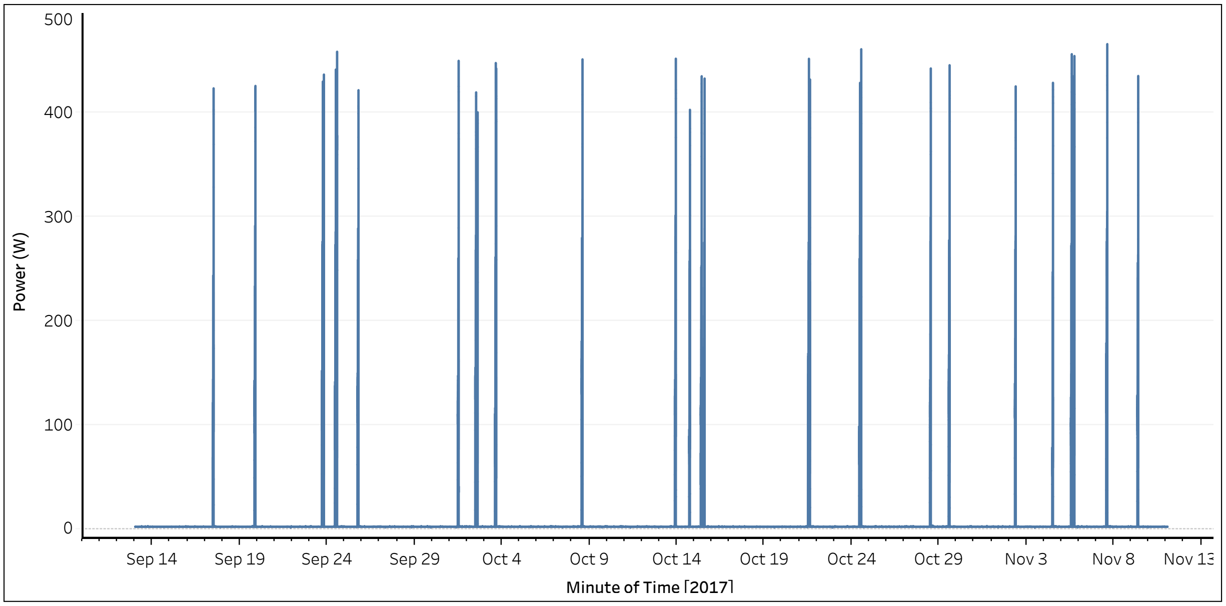


Figure 3.8: Usage times for the clothes washer in House 2. Each spike shows a single operation for the washer.

the machine until the water temperature reaches the desired temperature setting. This will take 20 minutes to complete. During water heating the dishwasher consumes approximately 925W. The sprinkler then starts washing by rotating and spraying hot water on the dishes consuming 61W for 25 minutes. The dishwasher then drains dirty water at 1:05, consuming about 26W. The rinse state that follows repeats the steps in the wash state by filling in water and heating the water to the desired temperature. Heating starts at 1:08 consuming 925W and the rinsing starts at 1:24 consuming 62W. Sprinkling fresh water to rinse the dishes takes less time (6 minutes) since dishes are almost clean. The water is drained at 1:30. Finally, the dry state starts at 1:37 in which the dishwasher starts increasing the interior air temperature of the machine for 10 minutes consuming an average of 390W. The insulator installed in the dishwasher keeps the high temperature until the dry states completes.

3.2.2.3 The Clothes Washer

Washing machine usages are depicted in Figure 3.8. The graph shows the power readings over two months for the washer. We observe that when the washer is turned on there is a high spike in the graph which indicates that the power consumption was almost zero before turning on the machine. Power consumptions returns back to zero after the washer is turned off.

Figures (3.9,3.10,3.11) graphically depicts the Single Use Profiles (SUP) of the clothes washer for three wash modes: Heavy (normal), Medium (permanent press), and Light (delicates). Each mode takes different amounts of time. Each of the figures has two plots: the upper plot is the real consumption data plotted over time. The lower plot corresponds to a smoothed version of the real data plot using moving median so that the different states in each washing mode is easily identified.

To deeply comprehend how clothes washers work and what are the different phases of washing, we will analyze Figure 3.9 which shows the power consumption of the washer for a single use. The washer has three states of operation: wash, rinse, and spin. These three states occur for each single use of the the washer. This washer starts its wash state at 12:00 by filling in the water to the main cavity, during which the washer consumes relatively low power at approximately 2–80 W. As the water fills in, the washer starts rotating. Initially, the power consumption is low but gradually increases. This reflects a rise of the motor electrical load as the water level climbs which needs more power to rotate. Frequent variations in motor speeds are observed during the wash state, during which the washer consumes around 30–1900 W. The wash state is then followed by the rinse state at 12:55 which cleanses clothes with only clear water. In this state, the washer consumes 20-270 W. The last state is the spin state at 1:30 - 1:40. In this state the machine stops pumping water to the cavity and starts to dry clothes by spinning at a very high speed so that water escapes by centripetal force. During this state the machine consumes 30–450 W.

3.2.2.4 The Dryer

Figure 3.12 shows a SUP for a dryer in three different modes of operation: Heavy, Medium, and Light. A single operation of the dryer uses these states:Max Heat, Half Heat, No Heat. During the Max Heat state the dryer drum is heated to the maximum temperature in the dryer setting which is in our case is around 5300W, and the duration of this state is relatively longer than others with 8 minutes of operation. The next state is the Half Heat state which sets the core temperature to half of the maximum heat with duration of roughly 1.5 minutes. Finally, with the No Heat state the heating element in the dryer is turned off. The dryer maintains its state until the control sensor turns the heat back again. The common thing shared between all

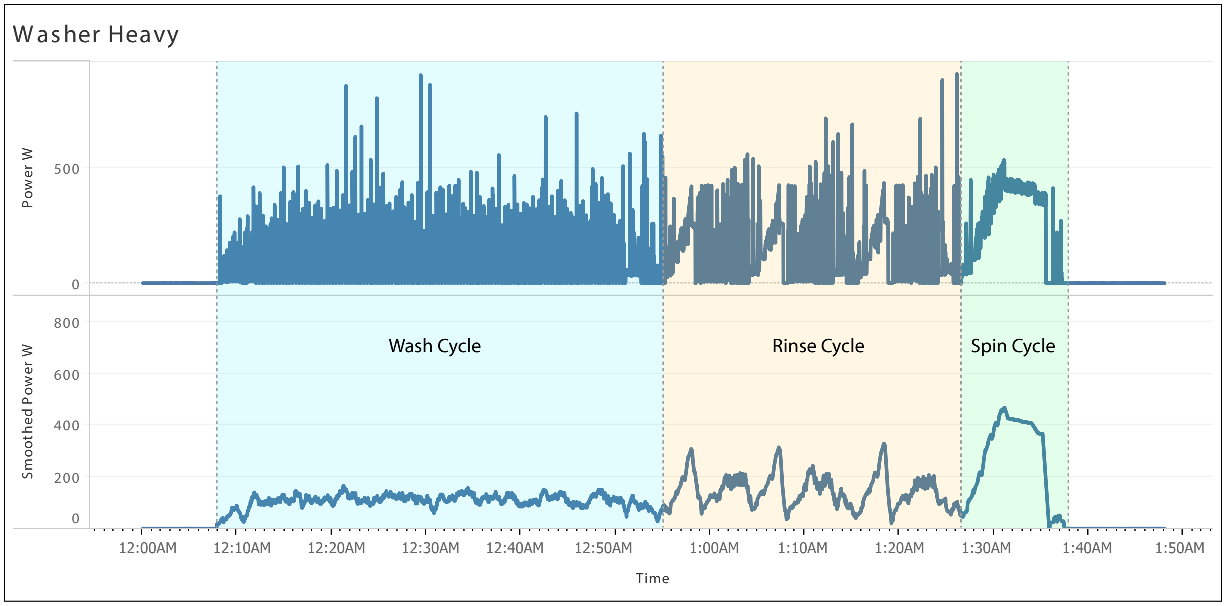


Figure 3.9: Single use (SUP) for clothes washer using heavy mode.

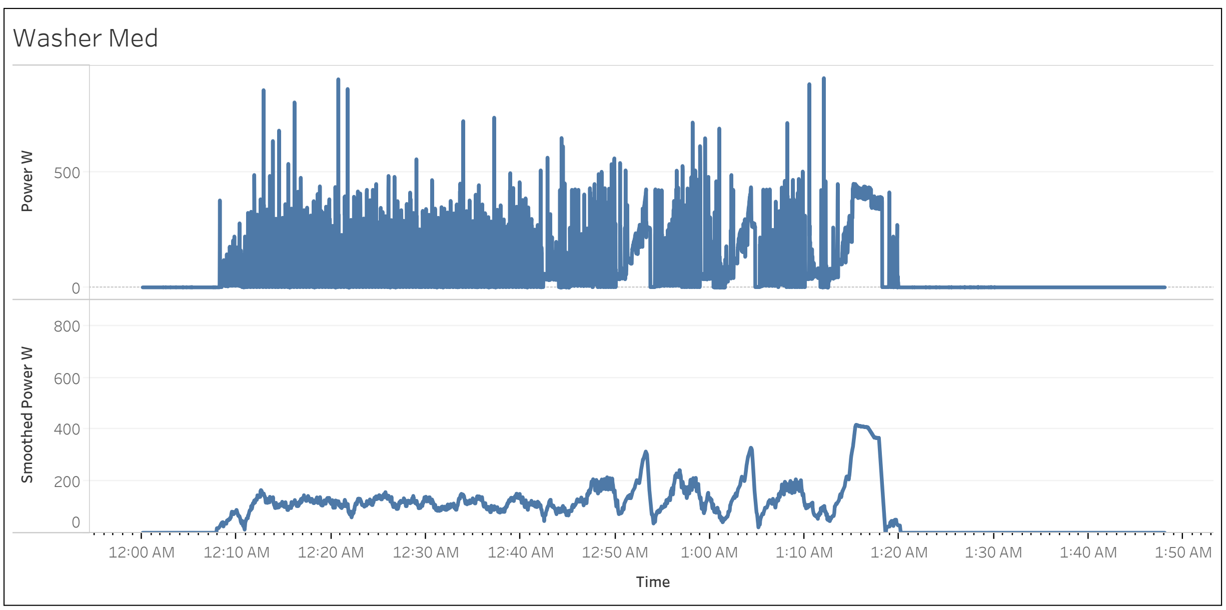


Figure 3.10: Single use (SUP) for clothes washer using medium mode.

the three states is that the dryer is continuously tumbling regardless of which state is taking place.

The first state is always of type Max Heat, followed by several alternating subsequences states of Half Heat and No Heat depending on the mode of operation. At the end, the dryer switches back to a Max Heat state followed by a long No Heat state for about 10 minutes to cool down.

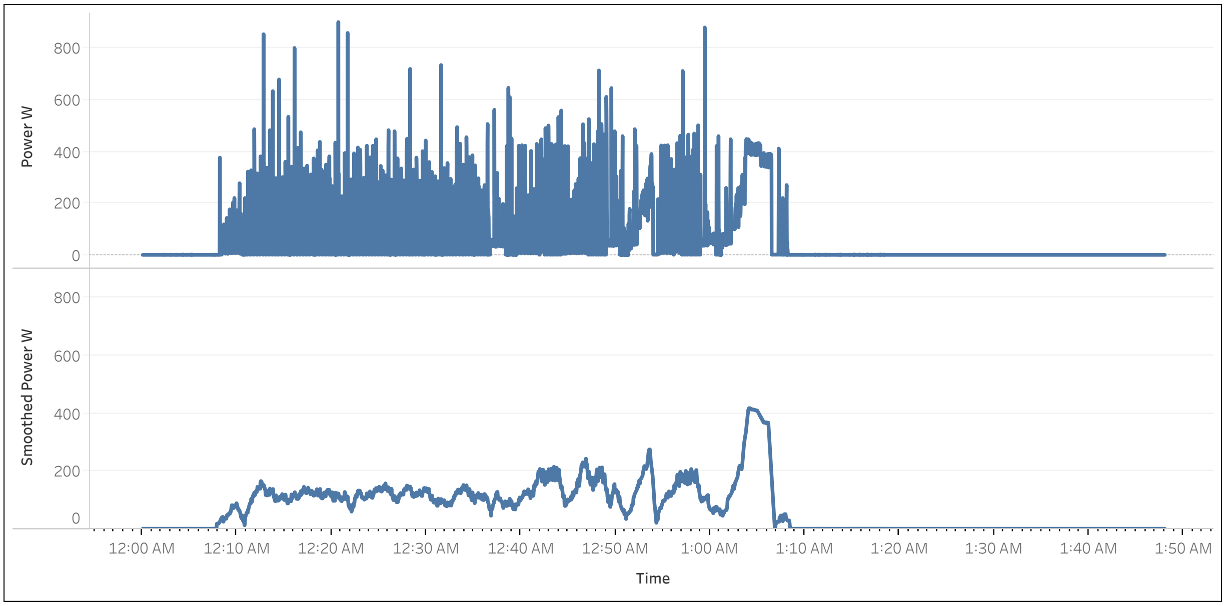


Figure 3.11: Single use (SUP) for clothes washer using light mode.

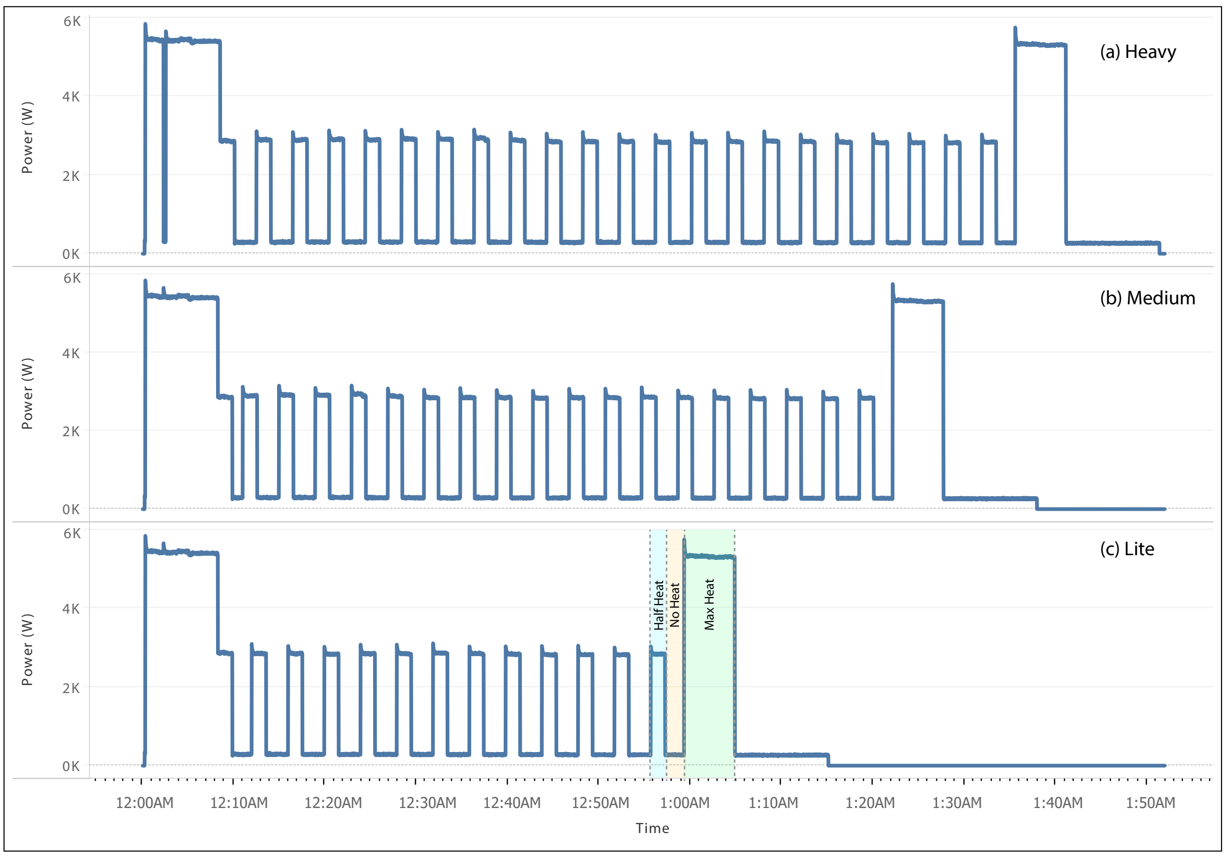


Figure 3.12: Single use (SUP) for the dryer showing three modes of operation: (a) Heavy, (b) Medium, (c) Light.

3.2.2.5 The Lights

Lighting is one of the electric loads that is highly stochastic due to the many factors that affects light usage. We will focus into two factors: Natural lighting, and Active Occupancy. Human perception of the natural light level within a building is a major factor in deciding the use of electric lighting or not. On the other hand, the number of people who are awake at home (Active Occupancy) is the other key factor for determining domestic lighting usage as presence and activity in the house is a logical reason for power consumption.

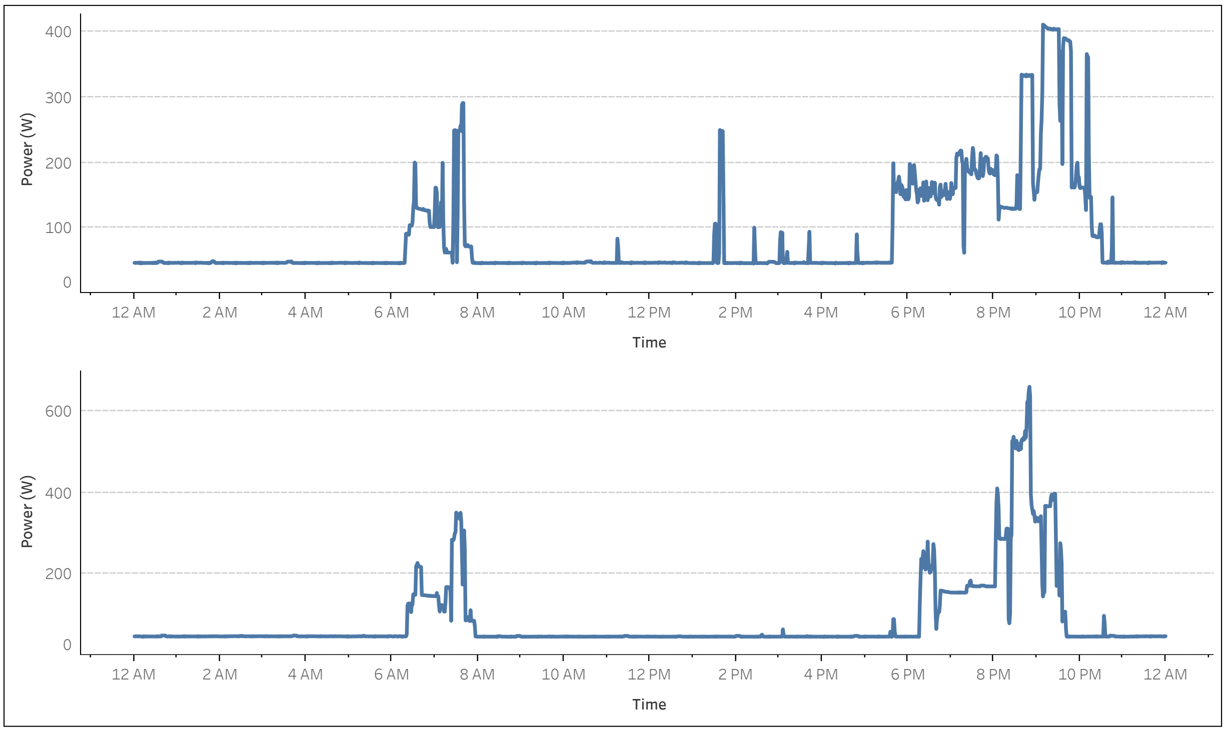


Figure 3.13: Power consumed by turning lights on for two different days.

The Daily Usage Profile (DUP) for lights is shown in Figure 3.13. The plot shows two separate days of light usage in House 2. Light consumption is high at the start of the day as residents are often getting ready for the day. As the day moves on, there is little use of lights since there is natural lighting. The activity of turning on lights in the house is observed in two plots at similar times of the day: In the morning 6:15AM - 8:00AM where the power consumption reaches 330W and in the evening 6:00PM - 10:30PM where the consumption ranges in 155W - 500W.

Regarding the daily consumption of lights, it is challenging to find regular daily patterns or states as we found in the previously discussed loads. Instead we studied energy consumption data over a long time period as demonstrated in Figure 3.14. A single day consumption in Figure 3.13 conforms with the long-term plot showing that there is a relatively more activity in lights usage during the morning time (6:00AM - 8:00AM) and the evening time around (7:00PM - 9:00PM).

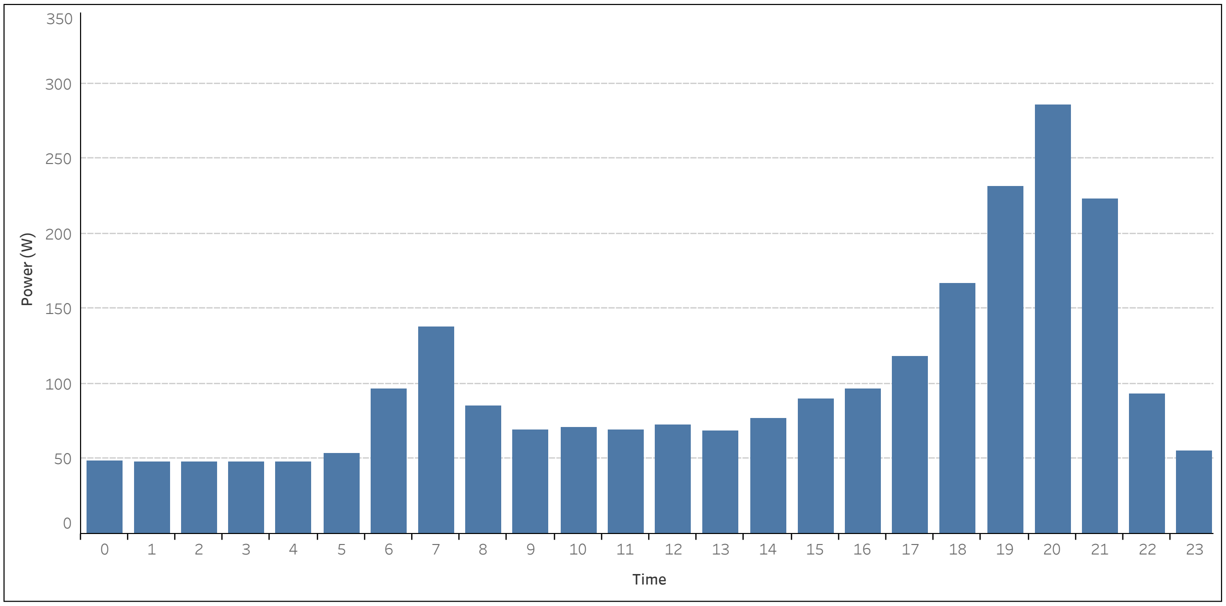


Figure 3.14: The average power consumption in House 2 during the entire data collection period, distributed over hour of the day.

3.2.3 Electrical Load Classification

Based on the characteristics of electrical loads we propose the classification presented in Table 3.2. Appliances with **Timer Based Loads** are regulated using a timing regulation device, e.g. thermostat that responds to temperatures. For example with the refrigerator, the compressor (that cools the refrigerator inside) is activated when the interior temperature reaches the preset upper threshold while it switches back to standby (off) when the temperature falls to the lower threshold. In contrast, the on-off of the water heater is in the opposite since the thermostat senses the tank water temperature and turn on the heating coil when the water get colder and then turned back off when the water reaches the desired water temperature. Figure 3.15 demonstrates this process.

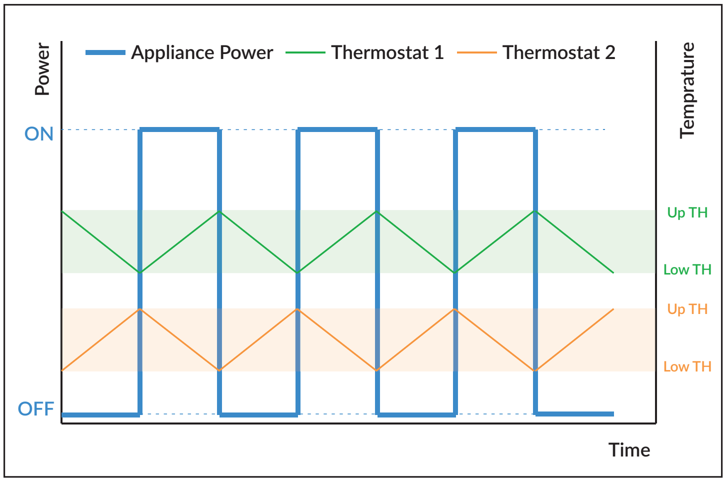


Figure 3.15: A plot shows how the setting of the thermostat affects the operation of a device. The green line corresponds to Low Threshold Activated thermostat which turns on the device when the temperature falls below the Low Threshold. The orange line corresponds to High Threshold Activated thermostat which activates the opposite way.

Appliances with **Preprogrammed Loads** are those appliances that are operated manually by human intervention. When an appliance is turned on it will go through a set of preprogrammed set of operations that have a specific duration for each process until it automatically shuts off. For example, a dishwasher has operations for water filling, washing state, and rinse state. Each state has its preconfigured duration, and the dishwasher will turn off when finishing the last state.

The class associated with most appliances/devices is the **Variable Loads**. The appliances in this class are triggered by human behaviour. These interventions are related to a vast number of parameters related to households such as wakeup times, bed times, work times, number of people inside the house, ages, etc. Since these parameters are unpredictable, it is hard to know precisely the start time nor the operation duration for loads belong to this class. Consequently, the electrical load behavior of this kind of loads relies on their usage statistics.

3.3 Conclusion

In this chapter we discussed the importance of data in smart grid applications, specially residential disaggregated power consumption data and the general model on how to collect this data realistically. We then discussed the public dataset RAE [66], the data description, and the model which is used to gather this dataset. Finally, we analyzed the dataset to understand the different loads that resemble the dataset by classifying these loads and analyzing each load to comprehend its behavior either as a long-term, or per single use.

CHAPTER 4

Power Consumption Simulation

It is sometimes necessary to create simulated data when it is impractical to obtain real data that may have protected information, may not be available for legal reasons or there is an insufficient amount of data [26]. Simulated datasets have the potential to provide the sufficient data needed to evaluate algorithms.

The publicly available power consumption datasets (PCDSs) are scarce [83, 81]. Furthermore, the available PCDSs are often limited to aggregate power consumption of a residence with little information specific to the power consumption of an appliance. When there is information specific to an appliance there is a lack of diversity of usages for the same appliance over a long period of time and the different operation modes for an appliance are not captured.

The reason for these limitations is that there is still limited use of sensors in homes that can monitor disaggregated power consumption. This makes it difficult to evaluate algorithms used to make recommendations on reducing power consumption. This chapter focuses on the simulator developed for generating synthetic data sets.

4.1 Inverse Transform Sampling (ITS)

Assume that $F_X(x)$ represents the Cumulative Distribution Function (CDF) of random variable, X. The CDF is the probability that the variable takes a value less than or equal to x. That is $F_X(x) = Pr[X \leq x] = \alpha$. The function takes as input x and returns values from the [0,1] interval (probabilities).

The inverse of the CDF is represented by $F_X^{-1}(\alpha)$ and is referred to as the inverse distribution function. $F^{-1}(\alpha)$is the value $x_\alpha$such that $P(X \leq x_\alpha = \alpha)$.

The Inverse Transform Sampling (ITS) is a method for generating random numbers from any probability distribution by using its inverse cumulative distribution function. The first step is to generate a random number, u, from the uniform distribution. The second step is to calculate $X = F^{-1}(u)$. X follows the distribution governed by the CDF. We use ITS for randomly generating turn on times and household usage intensity values.

4.2 Simulator Parameters

In this section we discuss the parameters that is used by the simulator. These parameters are the following: Turn on times which represent the time of the day that an appliance is activated by the user. The second parameter is the Household Usage Intensity which refers to the pattern of operation modes that a household selects over time. The third parameter is the SUP Representation Object which is an object that contains a formal representation for any SUP.

4.2.1 Turn On Time

Turn on time (t_{on}) refers to the time when an appliance is activated by the user during the day. We use the ITS method to generate values of t_{on}.

The power consumption for the appliance is aggregated over a single day (24 hours). Figure 4.1 (a) shows the average power consumption from the RAE dataset discussed in Section 3.1.1 for a dishwasher over the data collection time, aggregated over a 1 hour period over a single day. Each value of the y-axis represents the average power consumption for that specific hour over the data collection period (72 days) i.e., each value of the y-axis represents the probability of turning this dishwasher on at this hour. The plot is normalized so that each value of the y-axis represents a percentage of the total usage.

The probability of turning on appliance, a, at time, t, is represented by $f_T^a(t)$ such that:

$$f_T^a(t) = Pr(T = t) \quad , \quad t \in D \tag{4.1}$$

where D is a set of discrete time partitions that represents one day. For example, if the day is partitioned into 24 hours then $D = \{0, 1, 2, ..., 23\}$.

The next step is to compute the Cumulative Distribution Function (CDF), $F_T^a(t)$, as follows:

$$F_T^a(t_x) = \sum_{t=0}^{t \in D, t \le t_x} f_T^a(t) \tag{4.2}$$

ITS uses the CDF presented in Section 4.1 as the basis for generating random values of

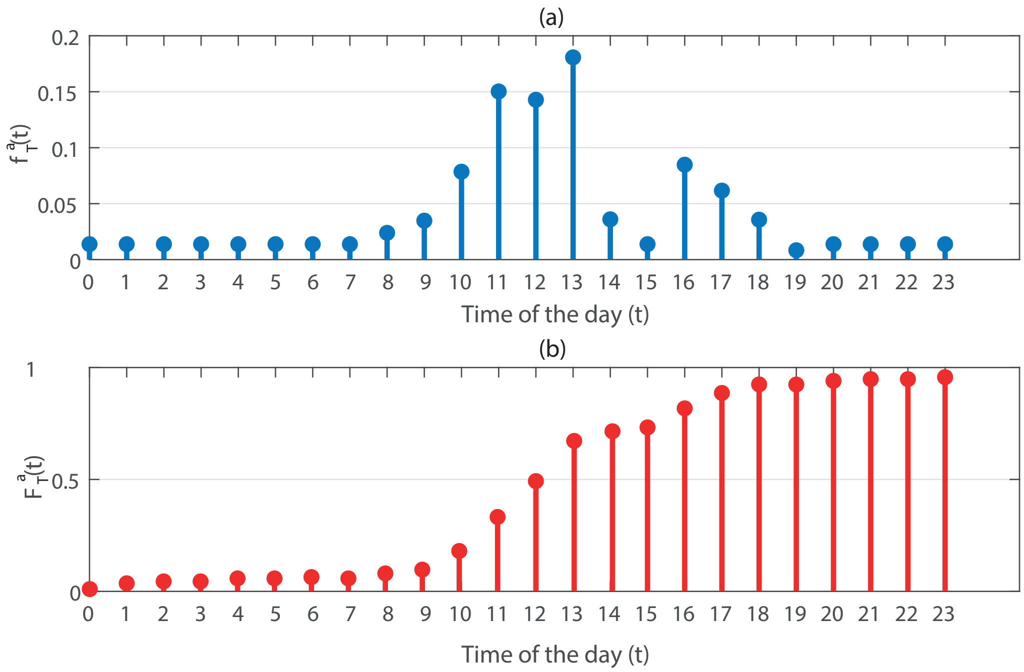


Figure 4.1: **(a)** The Probability Mass Function (PMF) for turning on an appliance over time. **(b)** Cumulative Distribution Function (CDF).

turn on times.

4.2.2 Household Usage Intensity

Household Usage Intensity refers to the distribution of operation modes that a household uses over time. It takes the values: Low, Medium, and High. The operation mode that is used more frequently than the others is called the *Major Operation Mode*, while other less frequently used operation modes are the *Minor Operation Modes*. Households are considered as **high intensity** when they use an appliance in heavy operation mode more than using it with the light or medium operation modes, so in this case the major operation mode is the heavy mode. Also, households are considered as **medium intensity** when they use an appliance in medium operation mode more than using it with heavy and light operation modes, so in this case the major operation mode is the medium mode. Finally, households are considered as **low intensity** when they use an appliance in light operation mode more than using it with heavy and medium operation modes, so in this case the major operation mode is the light mode. For example, let us consider a household with medium usage intensity that uses a dishwasher for a month. This household uses the medium operation mode (major) for the dishwasher most of the time, e.g., let's say 60% of the time e.g., 18 days. The light and heavy modes (minor) are used 20% of the time each e.g., 6 days each.

We assume that the selection of an appliance mode for a household is based on a multinomial distribution function. The frequencies of this distribution is as follows:

$$\text{operation mode} = \begin{cases} minor & ,20\% \\ major & ,60\% \\ minor & ,20\% \end{cases}$$

The major mode occupies the highest frequency since the household will use this mode most of the time. The minor modes are selected a fewer number of times.

ITS is used to select an operation mode.The mode is selected based on the value of i_h which is sampled from a multinomial distribution function with the the aforementioned frequencies. By selecting these values the majority of the samples (60%) will be selected from the major mode, while each of the other two minor modes will be selected with lower chance.

4.2.3 SUP Representation

The *SUP Representation Object* is a representation of a SUP in a particular operation mode. This is input to the simulator (more in Section 3.2.2). This section describes the representation of a SUP. This is used by the simulator to generate synthetic power consumption data.

Figure 4.2 shows three SUPs for a clothes dryer corresponding to three different operation modes. The clothes dryer has three different cycles within each single use: No Heat, Half Heat, Max Heat. These cycles are characterized by their time, duration, and power consumption level. These cycles are also repeated through the duration of the usage by either singularly (such as the max heat cycle) or in pairs (such as the no heat followed by the half heat). The duration of each occurrence of a cycle changes e.g., for the no heat cycle, all the occurrences have a 3 to 4 minute duration except the last no heat cycle which runs for over 8 minutes. Also, the repetition of pairs of cycles differs among operation modes. For example, Half Heat and No Heat cycles in the heavy mode are repeated 22 times per phase. However, in the medium

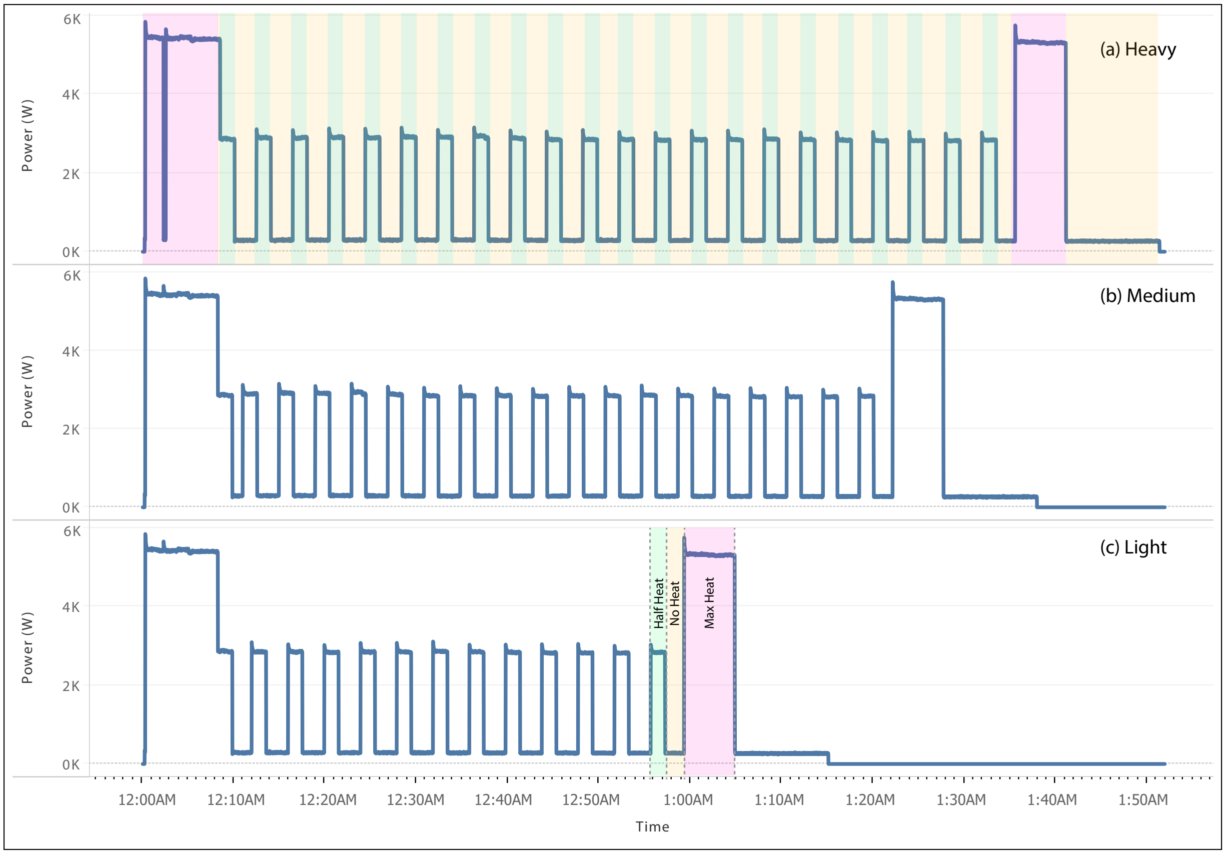


Figure 4.2: SUP for clothes dryer with three modes of operation **(a)** Heavy **(b)** Medium **(c)** Light. The types of cycles are shown in **(c)**. The repetition of phases is shown in **(a)**.

operation mode there are 19 repetitions.

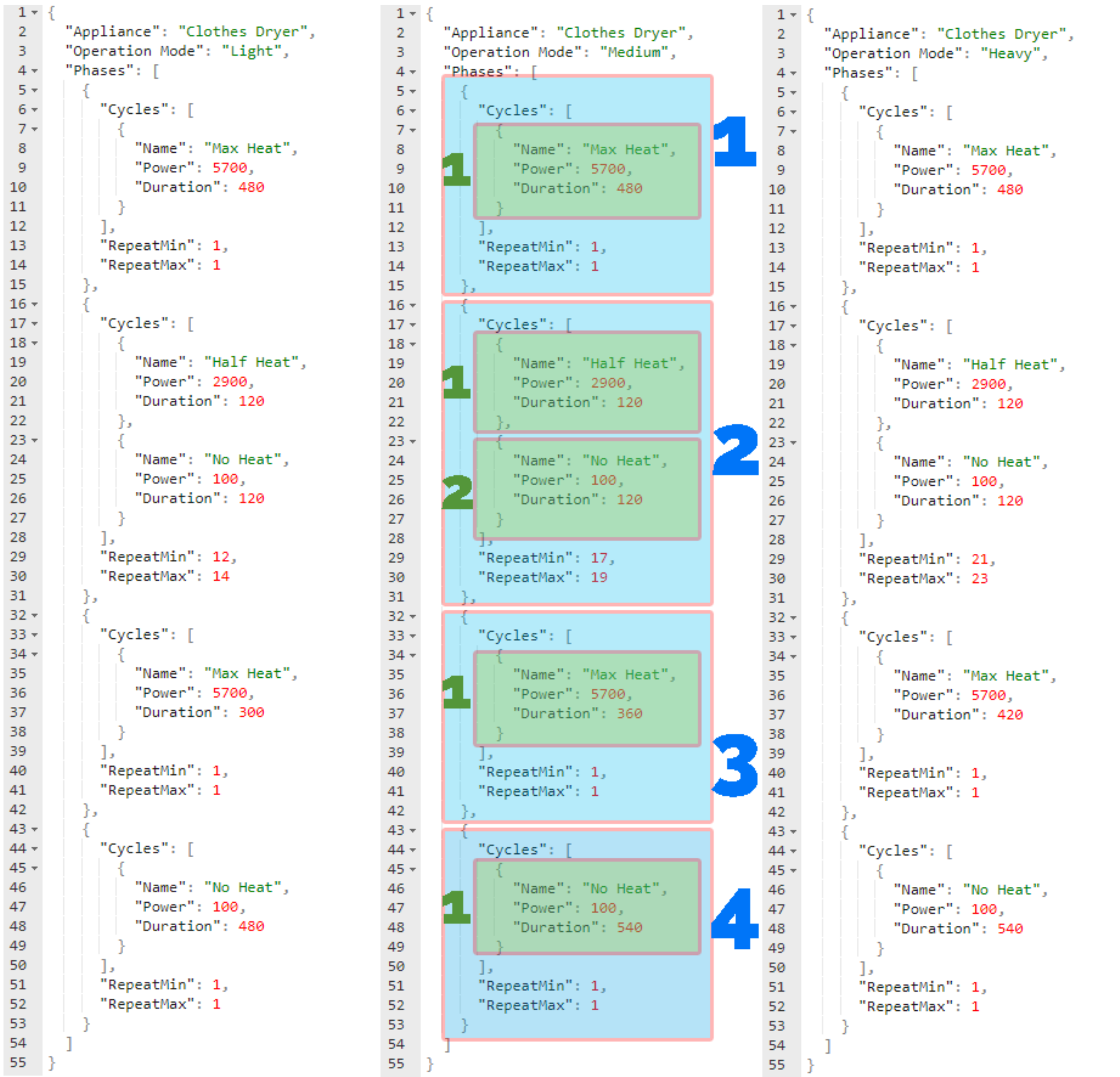
The representations of the three SUPs in Figure 4.2 are depicted in Figure 4.3. In Figure 4.3, three SUP representation objects correspond to the SUP plots. They share the same structure but with different values. For example, Figure 4.3 (a) shows the light SUP representation. The following is the description of the attributes found in a SUP representation object:

1. **Appliance:** This attribute is used to specify the appliance.

2. **Operation Mode:** This attribute is used to specify the operation mode.

3. **Phases:** This is an array where each element is a phase. Each phase in this array contains the following attributes:

   (a) **Repeat Minimum:** This is used to determine the lower bound of the number of repetitions of the phase.



(a) (b) (c)

Figure 4.3: SUP representation objects for a clothes dryer, phases are highlighted and numbered in blue, while cycles are in green.Three operation modes are shown **(a)** Light operation mode. **(b)** Medium operation mode. **(c)** Heavy operation mode.

(b) **Repeat Maximum:** This is used to determine the upper bound of the number of repetitions of the phase. The number of repetitions is selected randomly between the lower and upper bounds.

(c) **Cycles:** This is an array where each element is a cycle. Each cycle in this array contains the following attributes:

i. **Name:** This is used to determine the name of the cycle.

ii. **Power:** This attribute represents the average power value (amplitude) of this cycle.

iii. **Duration:** This attribute represents the duration of the cycle in seconds.

A graphical depiction of the SUP representation object structure is presented in Figure 4.3 where a SUP consists of multiple phases, illustrated as blue boxes. Each phase is duplicated one or more times based on the lower (Repeat Minimum) and upper bounds (Repeat Maximum) e.g., phase-1 occurred a single time while phase-2 is repeated multiple times. The reason behind introducing these two bounds is that the number of repetitions varies among models for the same appliance type. The use of lower and upper bounds can be used by the simulator to provide more realistic data that reflects the variation among different manufacturers.

4.3 Simulator Data Generation

This section describes the approach used to generating a day of consumption data and the processing of generating a single synthetic SUP.

4.3.1 SUP Generator

This section discusses the simulation of a single SUP, how the phases are repeated and how the cycles are generated.

4.3.1.1 SUP Phases Simulation

In the SUP representation object, each phase is associated with a lower bound (RepeatMin) and with an upper bound (RepeatMax). The number of repetitions of a phase is based on the selection of random value in the interval [RepeatMin, RepeatMax].

$$repetitions = rnd(RepeatMin, RepeatMax)$$

4.3.1.2 SUP Cycles Simulation

To simulate a cycle of an appliance, we **first** calculate the simulated cycle power p_{sim} using the equation:

$$p_{sim} = p_{rep} * (1 + \alpha) \quad , \quad \alpha \in (0, 1] \tag{4.3}$$

where p_{rep} is the power value of the cycle in the representation object. A noise component is added to the power value to make the power value more realistic in terms of randomness. p_{sim} is the power value after adding the noise element and α is the Cycle Power Variation Coefficient which is selected within the interval $[-1, 1]$. The value of α is chosen based on a normal distribution with the assumption that $\mu = 0, \sigma = 0.2$ so that the variation values are mostly focused around the power value in the SUP representation p_{rep}. Also, since $\alpha \in [-1, 1]$, the power variation could be positive or negative, so the simulated power value p_{sim} could be more, less, or equal to the power values in the representation p_{rep} .

The **second** parameter that controls the shape of the cycle is the Duration Variation Coefficient β, which changes the duration of the simulated cycle proportional to the value in the SUP representation as the equation:

$$d_{sim} = d_{rep} * (1 + \beta), \beta \in (0, 1] \tag{4.4}$$

The value of β is obtained from a normal distribution function with $\mu = 0, \sigma = 0.2$.

The Simulator uses Algorithm 1 to generate synthetic SUPs. The inputs of the algorithm is the SUP Representation Object for an appliance. The output is a synthetic SUP for the specified appliance for the specified mode. Lines 1 and 2 initialize the mode of the appliance and the array that stores the power values of the synthetic SUP. Lines 3 to 16 is repeated for each phase in the SUP representation object. Line 4 specifies the number of phases to iterate over within the variation limits specified in the SUP representation object. Lines 5 to 14 is repeated for all cycles in a phase. Line 6 initializes the array that holds the power values of the current phase. Lines 7 to 13 iterate over all cycles in the current phase. Line 8 calculates the duration (as number of samples) of the cycle based on the input representation and the variation coefficient. Lines 9 to 12 loop over every randomly generated power value in the current cycle to simulate. Lines 10 and 11 calculate the randomly generated power value and add this value into the phase power array. Line 14 joins the synthetic SUP power array with the current phase power array.

Algorithm 1: getSUP()

```
   input : SUP Representation Object
   output: Synthetic Single Usage Profile (SUP)
1  config = getRepresentationObject().getOperationMode() ;
2  SUPpower = [] ;
3  for i = 1 to size(config.phases) do
4      repeat = randomBetween(config.repeatMin,config.repeatMax) ;
5      for j = 0 to repeat − 1 do
6          PhasePowers = [] ;
7          for k = 1 to size(config.phases[i].cycles) do
8              duration = config.phases[i].cycles[k].duration*(1+ β ) ;
9              for l = 1 to duration do
10                 power = config.phases[i].cycles[k].power*(1+ α );
11                 PhasePowers.add(power) ;
12             end
13         end
14         SUPpower.addRange(PhasePowers) ;
15     end
16 end
17 return SUPpower;
```

4.3.2 Generating Daily Power Consumption

This section describes the generation of synthetic data for an appliance for one day. This is accomplished in the following steps: **First**, the turn on times and operation modes are generated using the techniques described in Section 4.2.1 and Section 4.2.2. **Second**, the generation of the synthetic SUPs. The generation of a synthetic SUP depends on the representation object that is selected based on the operation mode determined in the previous step. **Third**, the idle power values between synthetic SUPs is generated. These values correspond to the time between SUPs that represents the rest of the day time when the appliance is turned off or idle [6, 77] . These periods of time are represented by random low amplitude noise that correspond to low consumption components of the appliance such as LEDs, displays, etc. This noise is generated randomly and is tuned by Turn Off Noise Coefficient γ which we assume to take values $\gamma \in (0, 40)$ W based on the the consumption data in the dataset in Section 3.1.1.

Algorithm 2: generateDayPowerUsage()

input : Turn on times distribution, Usage Intensity distribution, and SUP Representation Object
output: One day of power usage for an appliance

```
1  t_on = getRandomTurnOnTime() ;
2  mode = getRandomIntensity();
3  dayPowers = [] ;
4  SUP = getSUP(SupRepresentationObject);
5  for j = 1 to t_on do
6  |   dayPowers.add(randomBetween(0,γ));
7  end
8  dayPowers.addRange(SUP);
9  for size(SUP) + t_on to DayEnd do
10 |   dayPowers.add(randomBetween(0,γ));
11 end
12 return dayPowers ;
```

Algorithm 2 is used to generate one day of power usage including a single synthetic SUP within this day for simplicity. The inputs of the algorithm are the following: Turn on times distribution, Usage Intensity distribution, and SUP Representation Object. The output is simulated power consumption for a certain appliance for a single day. Lines 1 and 2 initialize t_{on}, and the synthetic SUP operation mode using the techniques described in Section 4.2.1 and Section 4.2.2. Line 3 initializes the power array with the number of samples of the day. This array stores the power values for the whole day. Line 4 generates a new synthetic SUP with the generated operation mode from line 2. Lines 5 to 7 generate random power noise values before the appliance is turned on. Line 8 appends the synthetic SUP power values to the day power values at the specified turn on time. Lines 9 to 11 generate random power noise values after the appliance is turned on .

4.3.3 Synthetic SUP Smoothing with Low-Pass Filter

A Smoothing Function or (Low-Pass Filter) is commonly used with time series data [45] to smooth out short-term fluctuations (high frequency noise) and highlight longer-term trends or the general shape of the data (low frequency wave). The threshold that defines short-term and long-term depends on the application, and the parameters of the smoothing function is set accordingly.

Commonly used smoothing functions are the following: Moving Average and Moving Median [110]. A moving median returns a set of local k-point median values, where each median is calculated over a sliding window of length k across neighboring elements of the input set. This window jumps one sample at a time and each jump produces a new median value based on the neighboring samples enclosed by the window of length k. The Moving Average is similar except it calculates the average of the values.

In the context of our work, a time series A represents power samples for an appliance. A window of size k is selected from the time series A, the median is computed from all the samples within the window. The window slides by one neighboring sample and the median of the new k samples is computed. This process repeats until the number of values obtained from the sliding window equals the size of A. We use a smoothing function to smooth the impact of applying Cycle Power Variation Coefficient. Figure 4.4 shows the smoothing effect on a synthetic SUP for a dryer.

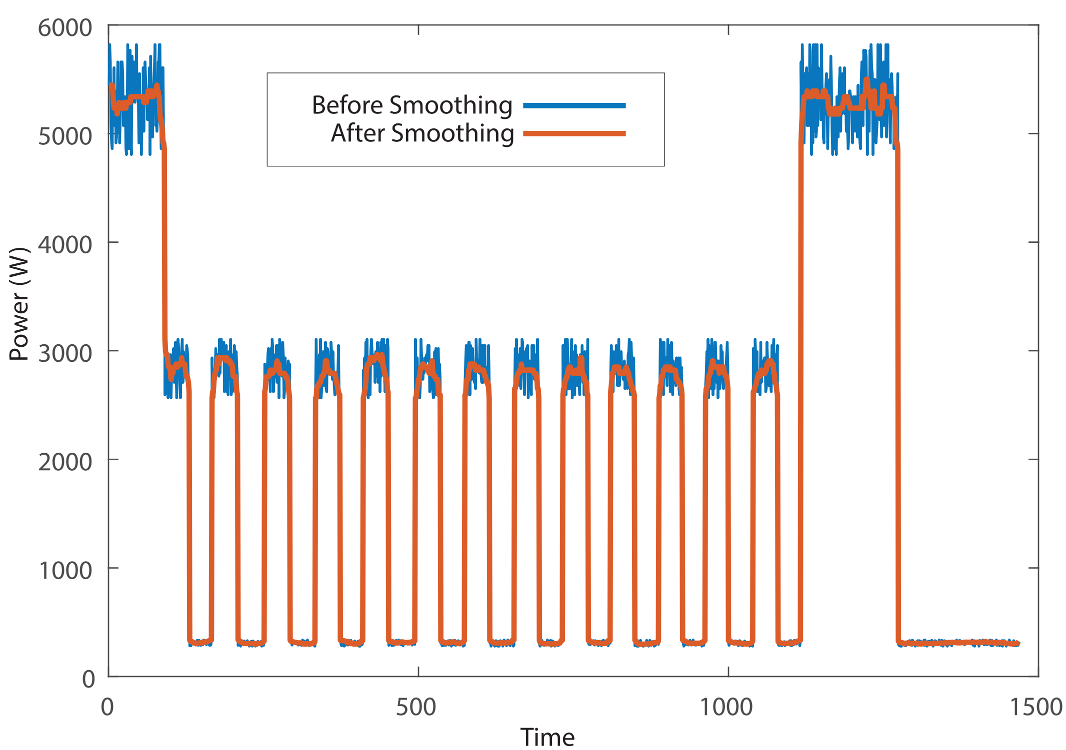


Figure 4.4: Synthetic SUP before and after smoothing.

This work uses the moving median. The general shape of the appliance power consumption data is square waves or with sharp (vertical) edges in the transition between different cycles. The moving median maintains these sharp edges in place (time) while the moving average distorts the edges and skews them so that the transition is smoother. Figure 4.5 shows two different smoothing functions applied on the same input data. The moving median filter lines up perfectly with the input data while the moving average skews around the edges.

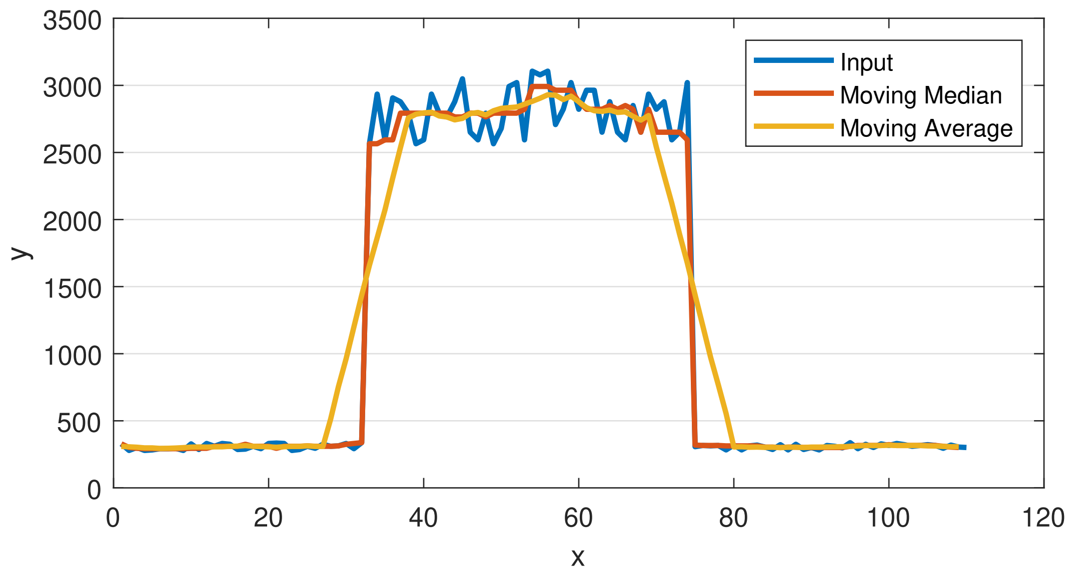


Figure 4.5: The difference between using moving average and moving median in smoothing the same input data.

4.4 Simulator Architecture

Figure 4.6 depicts the architecture of the simulator used with the inputs and outputs. The SUP Representation Object is used to shape the simulated consumption data for a single usage. The turn on time distribution is used determine the timing of which the activation of each appliance take place. The Household Usage Intensity distribution is used to determine the household the majority of the operation modes used over time. Tuning parameters are also provided to the simulator to add variation and stochasticity to the output. These parameters are the following: Cycle Power Variation Coefficient (α), Duration Variation Coefficient (β), Turn Off Noise Coefficient (γ), and the Smoother Window Size k. The output of the simulator represents a simulated power consumption data for a single appliance in a single day.

This section graphically depicts the approach used by the simulator to generate simulated consumption datasets. The SUP Representation Object is passed to the SUP Generator to generate a new synthetic SUP. A smoothing function is applied to the synthetic SUP to cancel out unwanted high frequency noise. Finally, turn on time t_{on}, household intensity i_h, and the smoothed synthetic SUP are passed to the One Day Generator. In this module a group of turn on times and operation modes are generated based on each SUP Representation Object. A new synthetic SUP is generated and placed at the corresponding t_{on}. This represents the power consumption for a single day for a particular appliance.

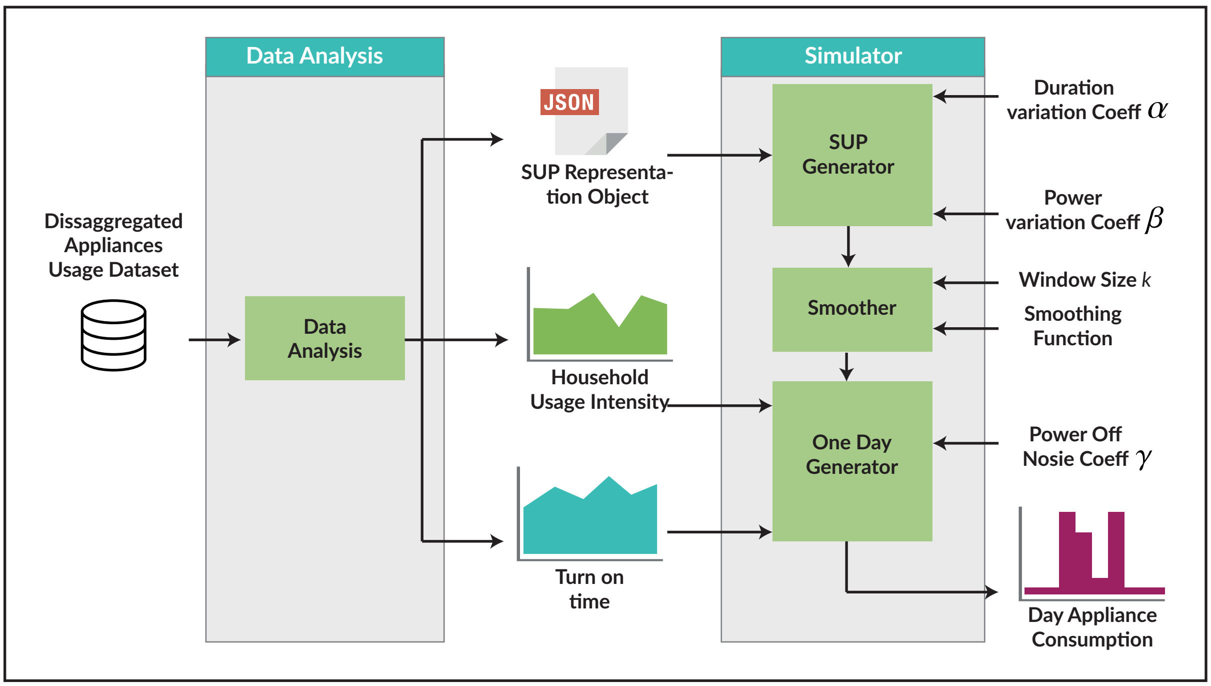


Figure 4.6: The Simulator components, inputs, and outputs.

The number of synthetic SUPs is determined by the number of sampled t_{on} times through the day. The location of each synthetic SUP is determined by the generated t_{on} value. Also, the operation mode of the synthetic SUP is defined by the sampled mode from the household intensity distribution. lastly, the tuning parameters affect timing t_{on}, power amplitude and the noise outside the synthetic SUP. This plot resembles the general profile shape of the Preprogrammed Loads class discussed in Section 3.2.3 that we will focus on in this study.

4.5 Conclusion

In this section we introduced the Power Generation Simulator that generates synthetic power consumption data on a daily basis for appliances. We discussed the parameters which control the simulator output these parameters are turn on time distribution, household usage intensity distribution, and SUP representation object.

CHAPTER 5

Single Use Profile (SUP) Detection

With Demand Response, consumers are encouraged to use appliances in off peak hours. This requires the need to detect when appliances are activated. Hence, we can advice the user to shift the load or not based on the detected time. This chapter introduces a mechanism that is applied to the generated daily consumption data to search for occurrences of an appliance being turned on. This serves as a basis for further analysis in determining appliances operation modes.

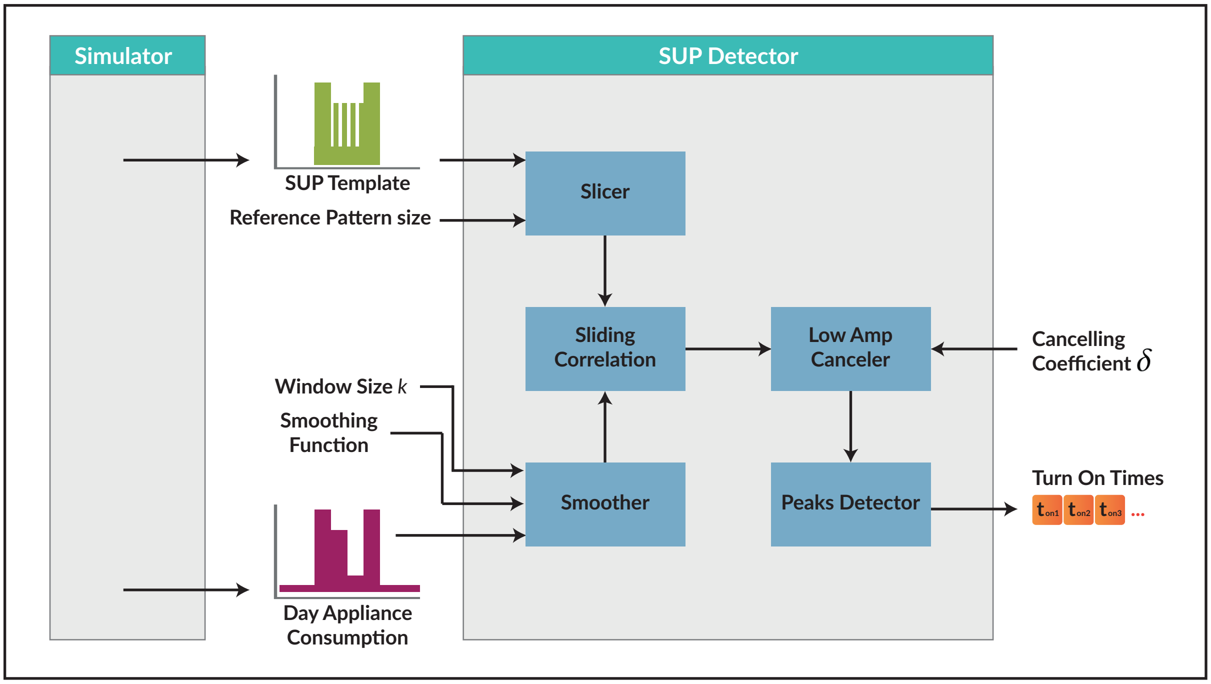


Figure 5.1: The SUP Detector components, inputs, and outputs.

5.1 Detecting SUP Instances Within A Day

This section describes the steps required to identify usage occurrences of an appliance from the daily consumption information for the appliance. Figure 5.1 shows the overall steps in the SUP detection module.

5.1.1 The Reference Pattern

A time series is a sequence of data points indexed by time. We represent a daily time consumption as a function $D(t)$ where t is time and $D(t)$ returns the data point associated with

t within a single day. We use a reference pattern representing the start of a SUP. This reference pattern is a sequence of data points. We represent the reference pattern as $S(t)$ which returns the data point that represents the power value associated with t. This chapter focuses on matching the reference pattern to a subset of the time series represented by $D(t)$.

The reference pattern is derived from a generated SUP by taking a slice of the generated SUP corresponding to turning on the appliance. The slice is a subsequence of the generated SUP. This sequence is represented by $S(t)$. Figure 5.2 shows a SUP with a sliced reference pattern at the the start of the SUP.

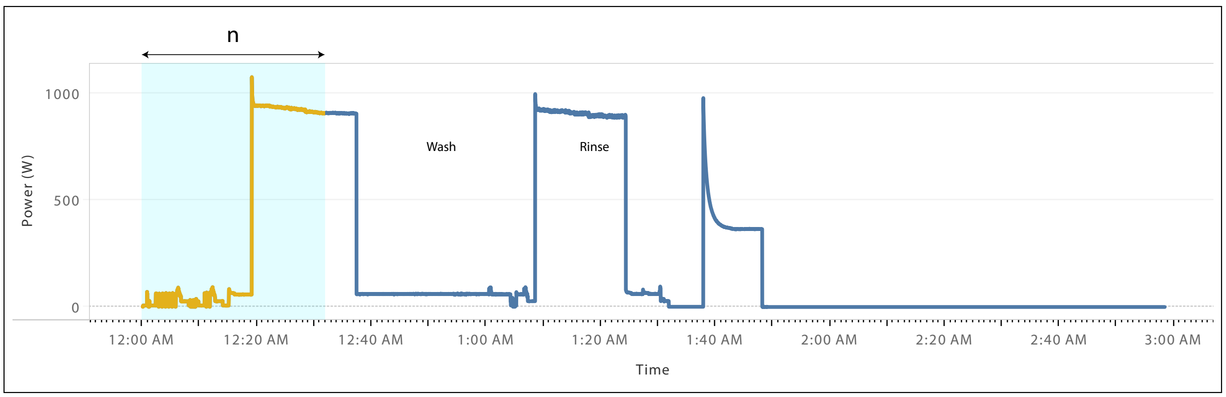


Figure 5.2: A SUP template for a dishwasher showing different cycles. A reference pattern with size n is sliced at the very beginning of the template SUP.

5.1.2 Smoothing Day Power Consumption

The one day usage data that is simulated and ready to pass to the correlation step, it is filtered by a smoothing component. We use a low pass filter with a moving median function with sliding widow of size n. We use a low pass filter to remove the high number of frequency fluctuations from the time series. This allows us to more clearly see trends.

5.1.3 Cross Correlation (XCorrelation)

This section discusses the definition of cross correlation and how it is used to determine if an appliance has been activated.

5.1.3.1 The Definition Of Cross Correlation

In time series analysis, cross-correlation [40] (or moving correlation or running correlation) is a measure of similarity of two series as a function of the displacement of one series relative to the other series.

Assume two time series represented by $f(t)$ where $t \in [1, n]$ and $g(t)$ where $t \in [1, m]$ and $n < m$. The cross-correlation, $X(t)$, for $f(t)$ with $g(t)$ at different points of time is defined as follows:

$$X(t) = (f \star g)(t) \; = \sum_{k=-\infty}^{\infty} f(k)g(k+t)$$

assuming that the number of moving windows is finite, we define the equation for cross correlation as follows:

$$X(t) = (f \star g)(t) \; = \sum_{k=1}^{n} f(k)g(k+t) \tag{5.1}$$

Another option is to use the absolute difference instead of multiplication such that:

$$X(t) = (f \star g)(t) \; = \sum_{k=1}^{n} \left| f(k) - g(k+t) \right| \tag{5.2}$$

If we use the absolute difference, the correlation function $X(t)$ between $S(t)$ and $D(t)$ is the following:

$$X(t) = (S \star D)(t) \; = \sum_{k=1}^{n} \left| S(k) - D(k+t) \right| \quad , \quad t \in [1, m-n] \tag{5.3}$$

where m represents the number of daily samples in $D(t)$ and n is the number of samples used in $S(t)$. The use of absolute difference instead of multiplication is due to the resulting huge numbers (13-14 digits) that is hard to deal with when we use multiplication. while the absolute difference produces manageable number length.

After the cross correlation function $X(t)$ is calculated, a normalization step [10] takes place to invert $X(t)$ over the y-axis by subtracting all correlation values from the average of the maximum power value for both $S(t)$ and $D(t)$. This normalization makes the similarity between the two functions relative to their maximum values. Therefore, the two functions are more similar when the normalized correlation value is closer to the average of maximum values found in the sequences represented by $S(t)$ and $D(t)$. The normalized cross correlation function

$X(t)$ is the following:

$$X(t) = \frac{Max(S(t)) + Max(D(t))}{2} - \frac{1}{n}\sum_{k=1}^{n}\left|S(k) - D(k+t)\right| \quad , \quad t \in [1, m-n] \tag{5.4}$$

5.1.3.2 Visual Intuition Of Cross Correlation

This section visually illustrates cross-correlation. Figure 5.3 shows a graphical depiction of the two functions $S(t)$ and $D(t)$ used in the cross correlation operation and shows the output of the cross-correlation $X(t)$. The function $S(t)$ is shown in Figure 5.3 (a). The function represents 600 samples. Second, the cross correlation function $X(t)$ between $D(t)$ and $S(t)$ is shown in Figure 5.3 (b). $D(t)$ shows an idle time from midnight till 9:00PM. When the appliance is turned on, the SUP appears from 9:00PM - 9:30PM. The plot then shows an idle period until midnight. Third, $X(t)$ as shown in Figure 5.3 (b) and shows almost a constant function from midnight until turning the appliance on at 9:00PM which means that there is no significant changes in the value of $X(t)$. The fluctuation in the value of $X(t)$ starts around 8:50PM as illustrated in Figure 5.3 (c) which depicts a zoomed in view for the period 8:45PM to 9:45PM. These fluctuations represent the overlap between the two correlated functions. Higher values of $X(t)$ means that there is more overlap present between the functions, therefor, higher similarity between the functions. The correlation function $X(t)$ shows its maximum value at approximately 9:00PM, which means that the potential SUP that is being searched for starts at this time. The turn on time of the appliance in $D(t)$ is observed at 9:00PM as seen in Figure 5.3 (c) representing the maximum value of $X(t)$.

The formulation of the cross correlation curve is illustrated in Figure 5.4 which shows the sliding reference pattern moving over time through three example points (a,b,c). At point **(a)** the reference pattern $S(t)$ overlaps with the day consumption $D(t)$ at the start of $D(t)$. The overlap is almost perfect since both functions have the same shape at this point of time. Both functions contain a large cycle and then a few smaller cycles. The green shaded area corresponds to the correlation value $X(t)$, which is at its maximum value across the entire period. At point **(b)** the reference pattern slides forward making the overlap with $D(t)$ smaller

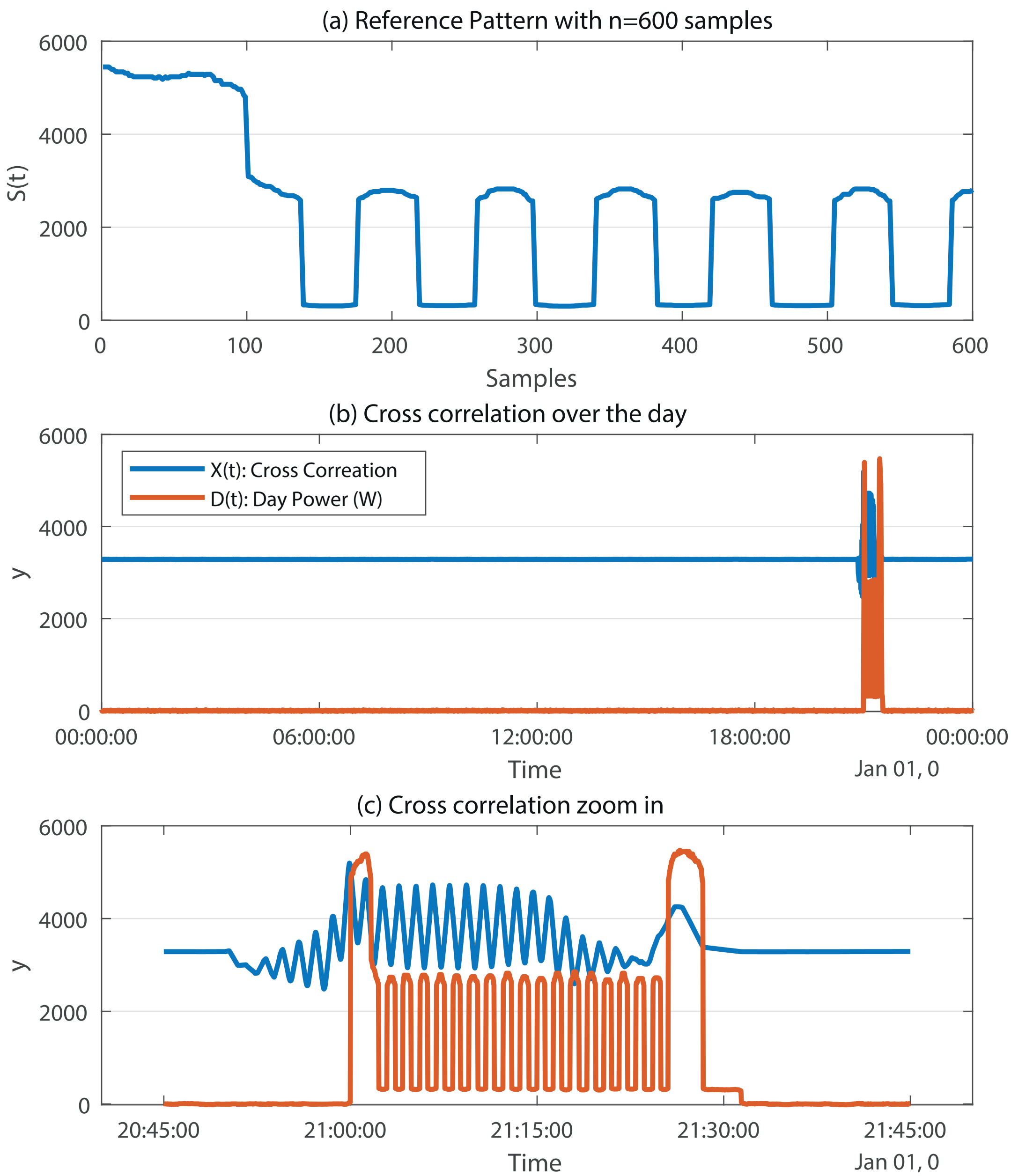


Figure 5.3: **(a)** The reference pattern represented by the function $S(t)$. **(b)** The cross correlation plot $X(t)$ and the day consumption function $D(t)$ over the day time. **(c)** A zoom in version from (b) focusing on the period where the the appliance is activated.

in value. The overlapping at this point of time represents a local minimum in $X(t)$ since both curves are almost out of phase. At point **(c)**, $S(t)$ slides forward and again become in-phase

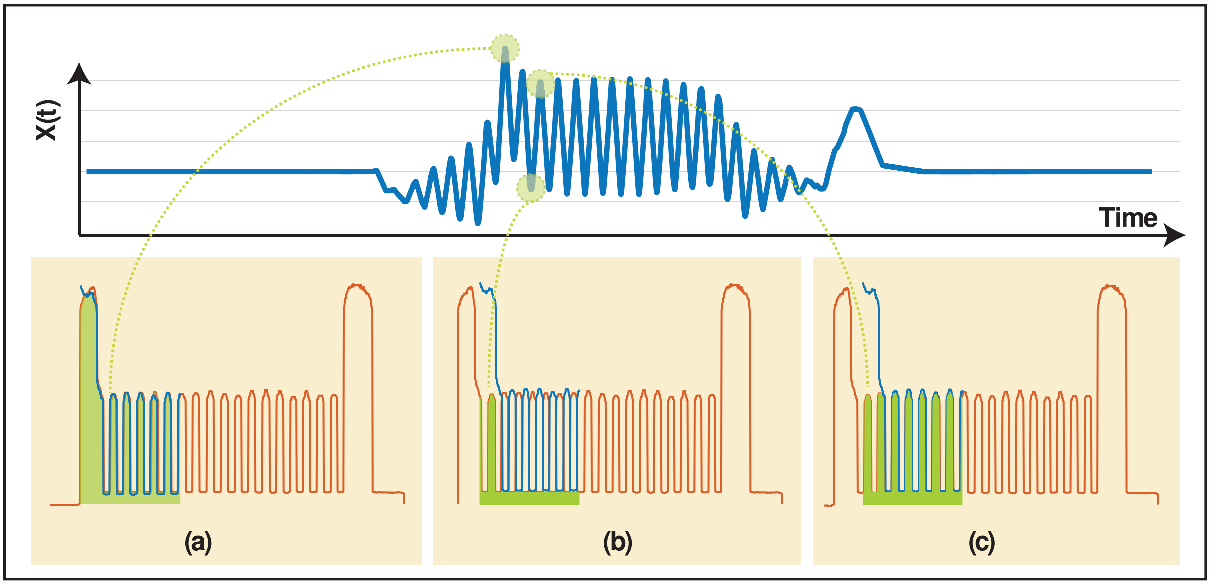


Figure 5.4: The reference pattern function $S(t)$ sliding over day consumption function $D(t)$ through three points (a,b,c)

with the day consumption curve with the overlap rising to a local maximum. However, at this point, $X(t)$ is at local maximum despite the in-phase situation. That is because the larger cycle of the reference pattern is overlapping with a smaller cycle in $D(T)$ leading to a smaller correlation value.

5.1.4 Low Amplitude Cancellation

We use X(t) to determine the potential turn on times. It is infeasible to consider all times at the local maximums of $X(t)$ as potential turn on times since $D(t)$ could contain multiple periods where there is not a considerable level of similarity with the reference pattern. Also, the noise contained within the correlation function may cause local maximums even after a smoothing function is applied. To overcome these problems we introduce a Low Amplitude Canceling Coefficient δ that limits the range of where to look for the local maximums. The role of the δ coefficient is to cancel out all values of $X(t)$ that is lower than a threshold τ determined by δ, so that the residue function $\overline{X}(t)$ of the correlation function after applying δ is the periods that contain the local maximums that is greater than τ. The value of δ is selected with the assumption that $\delta \in (0, 1)$. It is then multiplied by the average of the two absolute maximum values found in $S(t)$ and $D(t)$. The result of this multiplication is the threshold value τ. The residue function then is defined as the following:

$$\overline{X}(t) = X(t) - \tau \tag{5.5}$$

Where the value of the threshold τ is calculated as follows:

$$\tau = \delta\left(\frac{Max(S(t)) + Max(D(t))}{2}\right) \quad , \quad \delta \in (0, 1) \tag{5.6}$$

Figure 5.5 shows the plot of $X(t)$ of the previous correlation example in Figure 5.3. It shows also the residue function $\overline{X}(t)$ zoomed in around the period of time where $\overline{X}(t) > 0$ where $\overline{X}(t)$ shows high values that indicate a SUP is detected.

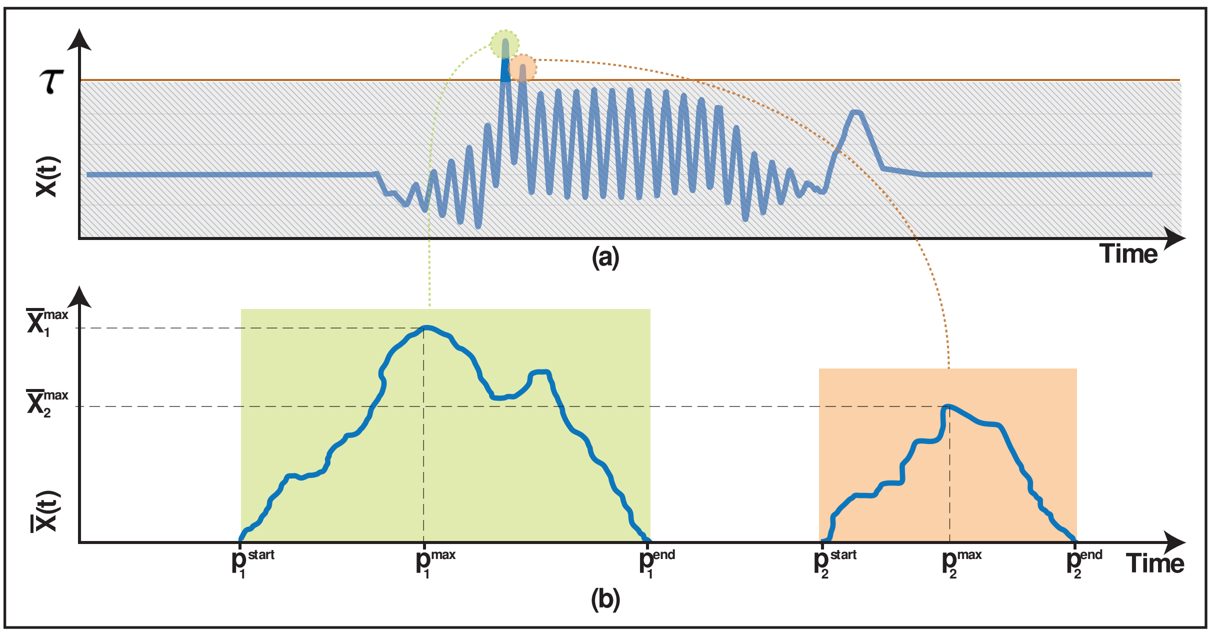


Figure 5.5: **(a)** The positioning of the threshold τ relative to the residue function $\overline{X}(t)$. **(b)** A zoomed in version of $\overline{X}(t)$. The plot shows the absolute maximum X_i^{max} for each tip of spike i enclosed in the period $p_i = [p_i^{start}, p_i^{end}]$

5.1.5 Determining Peaks Of The Residue Function

This section focuses on extracting the turn on times from the residue function $\overline{X}(t)$. Figure 5.5 (b) shows the plot of $\overline{X}(t)$. Since $\overline{X}(t)$ is a trimmed version of $X(t)$ where $X(t)$ is thresholded by τ value to produce $\overline{X}(t)$, then $\overline{X}(t)$ equals to zero all the time except for short periods of time. These periods of time are what remained from $X(t)$ after applying the threshold τ. $\overline{X}(t)$ contains isolated periods of time where the value of the correlation is relatively the highest among the entire period of the day. The shape of the $\overline{X}(t)$ is a group of continuous concave down curves that resembles the highest values of $X(t)$ where $X(t) > \tau$. Therefore, the position of the potential turn on time would be at the absolute maximum of each of these concave down curves. These periods in Figure 5.5 are p_1, p_2 where $p_1 = [p_1^{start}, p_1^{end}]$ and $p_2 = [p_2^{start}, p_2^{end}]$. These periods have their own absolute maximum values at p_1^{max}, p_2^{max} respectively within their domains. This maximum value represents the highest value of the cross correlation at this period, which means that the time when this maximum values occur is a potential turn on time. Consequently, the absolute maximum in p_1 is X_1^{max} at $t = p_1^{max}$, and the absolute maximum in p_2 is X_2^{max} at $t = p_2^{max}$. We then conclude that $\{p_1^{max}, p_2^{max}\}$ is the set of the potential turn on times.

Algorithm 3 shows the pseudo code for detecting turn on times. The way that the algorithm gets the times of the peaks is by looping over every single point in $\overline{X}(t)$ and flag when a point is located on the x-axis right after a point positioned above the x-axis, or vise versa. Line 1 calculates the value of the threshold τ, and the values of $\overline{X}(t)$ is calculated in line 2. In line 3, the flag *OnGround* is initialized to true. This flag indicates weather the value of $\overline{X}(t)$ is positive. Line 4 initializes two pointers: start and end. These two pointers indicate the index of the start and the end of each tip in $\overline{X}(t)$ as depicted in Figure 5.5. Line 5 initializes the reference pattern size, and line 6 initializes the turn on times set. Lines 7 to 19 loops over all values of $\overline{X}(t)$ to check their positions. Lines 8 to 11 checks if the current $\overline{X}(t)$ value is positive and the previous is zero, it then lowers the *OnGround* flag and sets the start pointer. Lines 12 to 18 checks if the current value of $\overline{X}(t)$ is zero and the previous value is positive. It then raises the *OnGround* flag, and sets the end pointer. It then gets the index (time) of the absolute maximum in the period $[start, end]$, adds this time to the turn on times set, and finally resets the start and end pointers.

Algorithm 3: getTurnOnTimes(X,δ)

input : Cross Correlation sequence X, Low Amplitude Canceling Coefficient δ,Reference Pattern sequence S, One Day Usage sequence D

output: Turn On Times Set T

```
τ = δ× (max(S)+max(D)/2 ;
Xbar = X - τ ;
onGround = true;
start = end = -1;
k = size(S);
T = [] ;
for i = 1 to size(D) − k do
    if onGround AND Xbar[i] > 0 then
        onGround = false;
        start = i;
    end
    if !onGround AND Xbar[i] <= 0 then
        onGround = true;
        end = i;
        t_on =X.getIndexOfMaxBetween(start, end);
        T.add(t_on);
        start = end = -1;
    end
end
return T ;
```

5.2 Conclusion

In the previous sections we discussed how to get a simulated day consumption data and search for SUP instances within this day, trying to find out what times does the SUP is activated through the day. We saw how we chose a reference pattern function that is capable of detection different SUPs in different modes of operation for the same appliance. Also, we explained the cross correlation function and how the correlation curve indicate the similarity between the two correlated functions. Finally, we introduced a trimming mechanism to eliminate the less probable periods of time that have potential turn on times for the appliance. Then we discussed the algorithm that extracts the potential turn on times for the residual Correlation function and pass this set of times to the next module which will determine the mode of operation that each SUP belongs to.

CHAPTER 6

Single Use Profile (SUP) Classification Based On Dynamic Time Warping (DTW)

Chapter 5 presents a method that determines when an appliance has been activated. For each time that the appliance has been activated we need to determine the appliance operation mode. This requires a comparison of the time series at the point that the appliance has been activated and the reference patterns for each of the operational modes. This chapter describes the approach using Dynamic Time Warping (DTW) which is depicted in Figure 6.1.

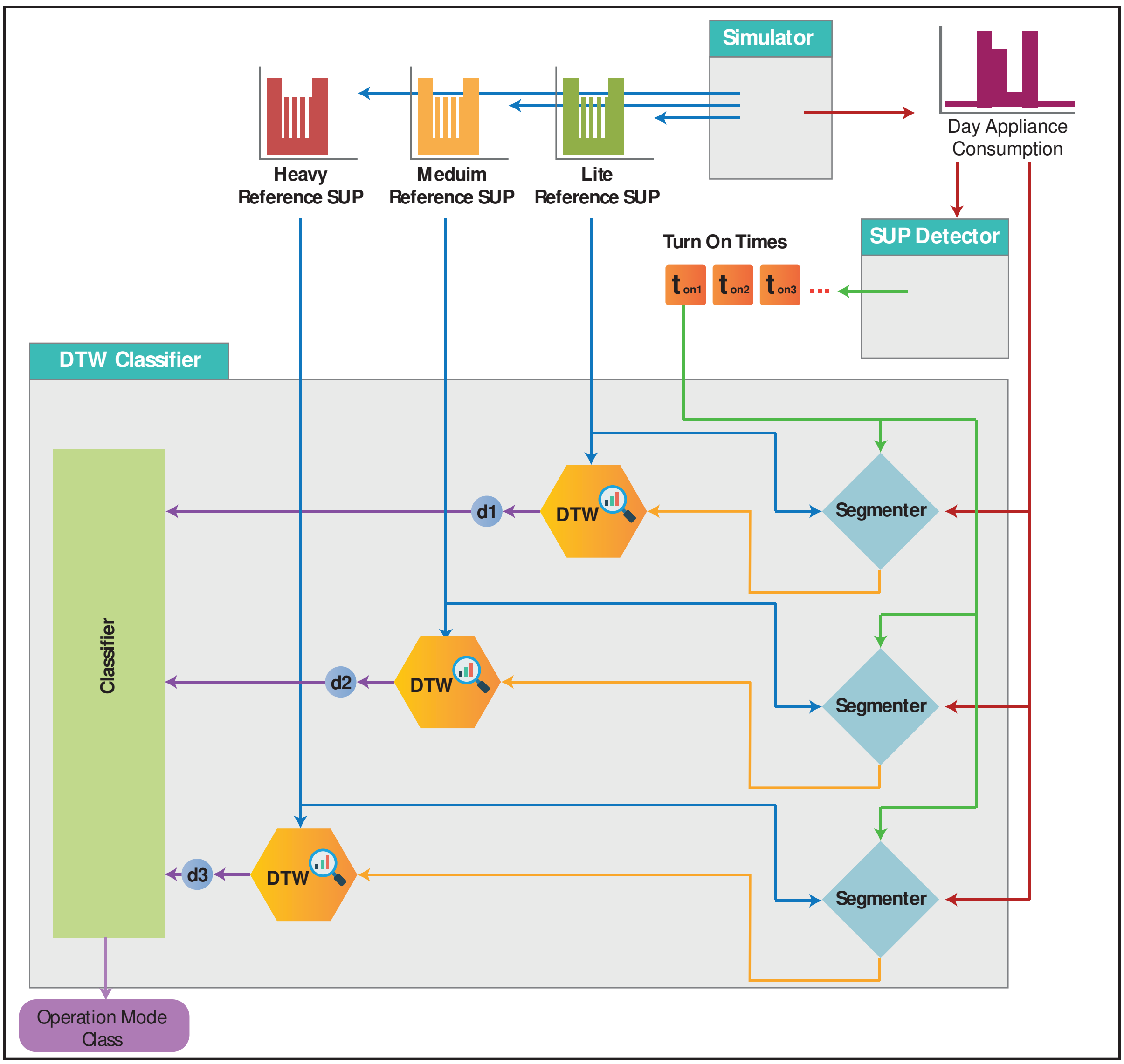


Figure 6.1: The DTW Classification module and the connections with other modules.

6.1 Dynamic Time Warping (DTW)

We use Dynamic Time Warping (DTW) algorithm for measuring the similarity between two temporal series, which may vary in speed or phase [109]. Let us consider two sequences $X = \{x_1, x_2, ..., x_n\}$ of length n and $Y = \{y_1, y_2, ..., y_m\}$ of length m. The DTW distance between X, Y denoted by $D(X, Y)$ is defined as follows:

$$D(X, Y) = f(n, m) \tag{6.1}$$

$$\begin{aligned} f(i, j) &= \phi(x_i, y_j) + min \begin{cases} f(i, j-1) \\ f(i-1, j) \\ f(i-1, j-1) \end{cases} \\ f(0,0) &= 0, \\ f(i,0) &= f(0, j) = \infty \\ i &\in [1, n], j \in [1, m] \end{aligned} \tag{6.2}$$

The values of n and m represent the lengths of the two series. $\phi(x_i, y_j)$ is a cost function that computes the distance between two points x_i, y_j. There are many distance functions that could be used in the algorithm e.g., Euclidean Distance where $\phi(x_i, y_j) = (x_i - y_j)^2$ or Manhattan Distance where $\phi(x_i, y_j) = |x_i - y_j|$ etc. Figure 6.2 provides a numerical example for computing DTW distance between two sequences.

6.2 Day Consumption Segmentation

Let us assume that an appliance runs in n operation modes denoted by $M = \{m_1, m_2, ..., m_n\}$. The daily consumption is represented by $D(t)$ which contains subsequences representing appliance usages. For each mode m_i the reference pattern is represented by $P^{m_i}(t)$ where k^{m_i} is the size of a single usage profile for the operation mode m_i.

A segment $G^{m_i}(t)$ is a sub-sequence of the daily consumption function $D(t)$, starting from the point of the turn on time t_{on} and with the size k^{m_i} representing the size of operation mode

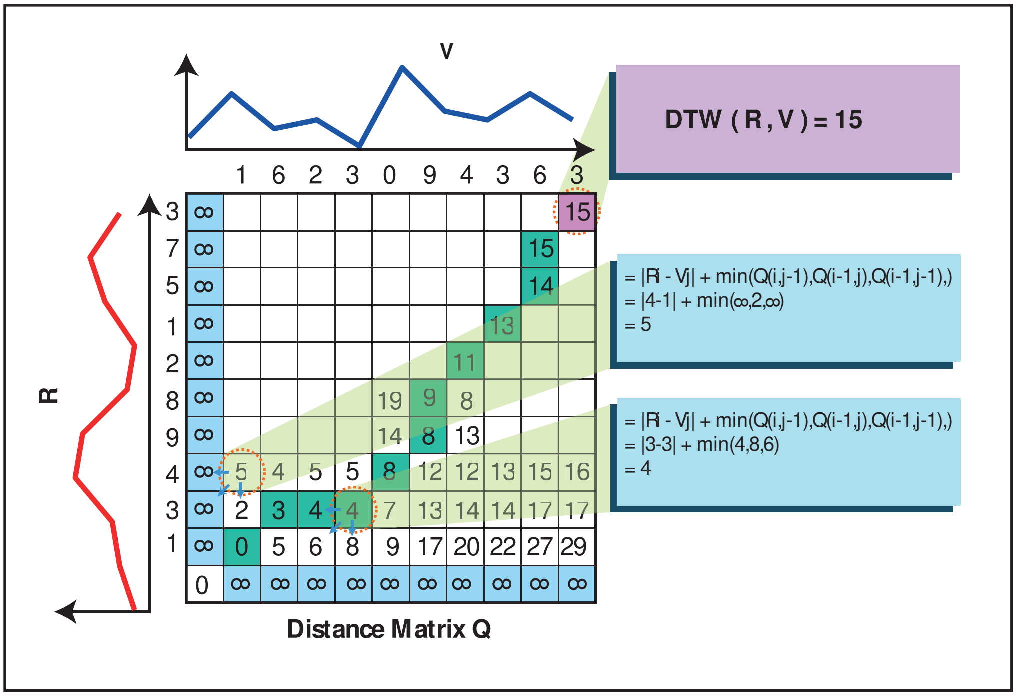


Figure 6.2: The calculations used to fill out the distance matrix. Souza [97]

m_i. This is defined as follows:

$$G^{m_i}(t) = D(x) \quad : \quad x \in [t_{on}, t_{on} + k^{m_i}] \tag{6.3}$$

For each reference pattern $P^{m_i}(t)$ for an operation mode m_i and turn on time t_{on}, segments of the daily consumption, D(t) can be extracted using Eq. (6.3) .Figure 6.3 depicts the inputs to the segmenter.

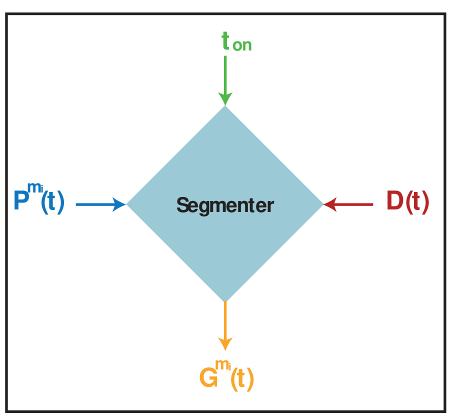


Figure 6.3: The segmenter inputs and outputs.

The segmentation is visually presented in Figure 6.4. In part (a) $D(t)$ is shown, and the segmentation starting point is indicated by the vertical dashed line at t_{on}. The segments sizes are highlighted in different shades so that each shade corresponds to the size of the segment for a specific operation mode. Section (b) shows the generated reference functions $P^{m_i}(t)$ that are used to specify segments sizes $\{k^{m_1}, k^{m_2}, .., k^{m_n}\}$. Lastly, in (c) the segment functions $G^{m_i}(t)$ are listed, each segment function is displayed across the corresponding reference function preparing them to be compared by the DTW algorithm.

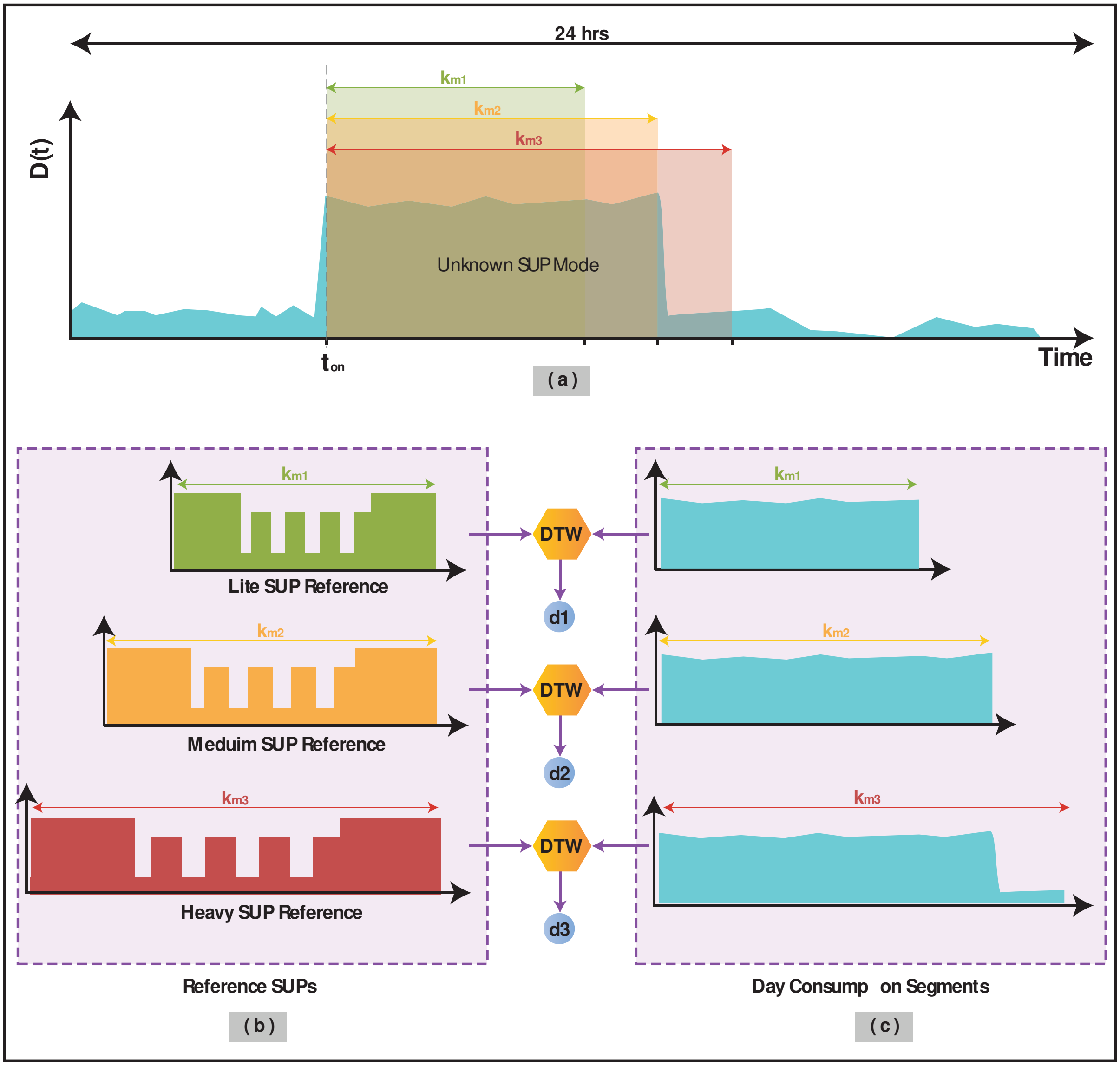


Figure 6.4: Visual representation for the segmentation.

6.3 Classification of SUPs Based On DTW Distances

This section describes how DTW determines the operation mode being used or each SUP. The DTW algorithm uses two inputs which are two functions representing sequences. It then performs a calculation to find the distance (or similarity) between these two functions as the output. Figure 6.4 shows the inputs and the outputs of the DTW algorithm. The DTW is invoked n times where n equals the number of reference patterns representing operation modes for an appliance. For each reference pattern and the corresponding segment, the distance is calculated.

Let X denote the sequence represented by $P^{m_i}(t)$ and let Y represent the sequence $G^{m_i}(t)$. The DTW distance is formulated as follows:

$$\begin{aligned} d_i &= \mathrm{DTW}(X, Y) \\ M &= \{m_1, m_2, ...m_n\} \end{aligned} \tag{6.4}$$

The value of the distance calculated by DTW is inversely proportional to the similarity. Thus we assume that the most similar reference pattern is the reference pattern with the minimum distance. This means that the mode m^* with the minimum distance d^* that is the operation mode of the appliance at the SUP detected at t_{on} as:

$$d^* = min(d_1, d_2, ..., d_{mi})$$

6.4 Conclusion

In this chapter we discussed the problem of recognizing certain SUPs within a day consumption and classifying the instance into the proper operation mode that most likely run with. The technique we used is based on time series processing started by forming references with all operation mode possibilities that is configured in the appliance. Then extracted segments from the day with equal sizes for each reference. These segments are taken at the detected turn on time. After that, we used DTW algorithm to measure the similarity between reference-segment pairs in order to find the most similar pair among the constructed pairs. We found that the operation mode corresponds to the SUP instance within the day is the the mode of the reference that has the least DTW distance with its corresponding segment.

CHAPTER 7

Single Use Profile (SUP) Classification With K Nearest Neighbors (KNN)

In this chapter we propose a machine learning based approach to classify the operation modes of the SUP.

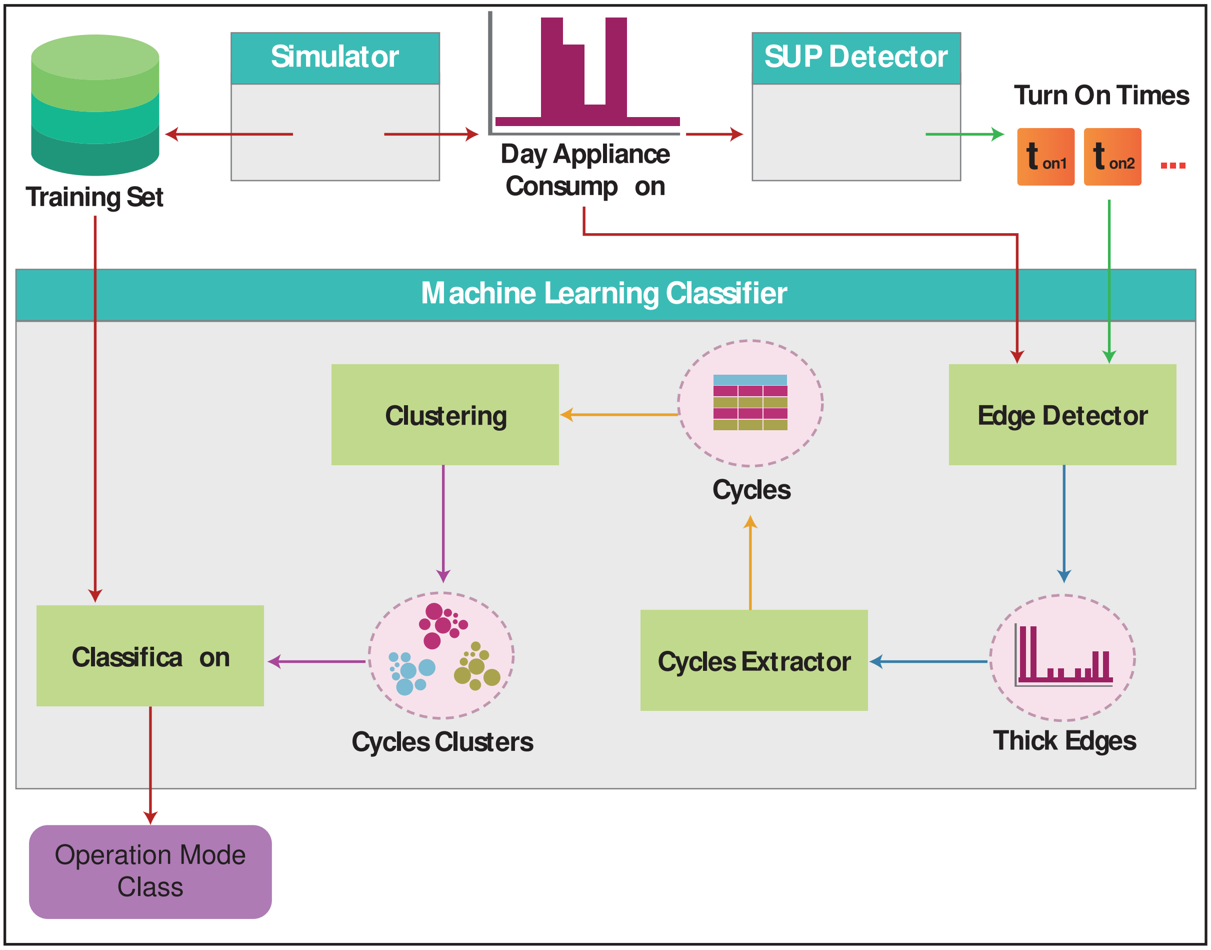


Figure 7.1: The basic architecture for the classification.

7.1 Feature Extraction

Chapter 5 presents a method that determines when an appliance has been activated. For each time that the appliance has been activated we need to determine the operation mode. This requires a comparison of the time series at the point that the appliance has been activated and the reference patterns for each of the operational modes. This approach is discussed in Chapter 6.

This section discusses the approach of extracting features from the the detected SUP. These features are represented the cycles that form a SUP. Each cycle is characterized by two abrupt changes in the power value. These features are used by the classifier to classify the detected SUP into operation mode.

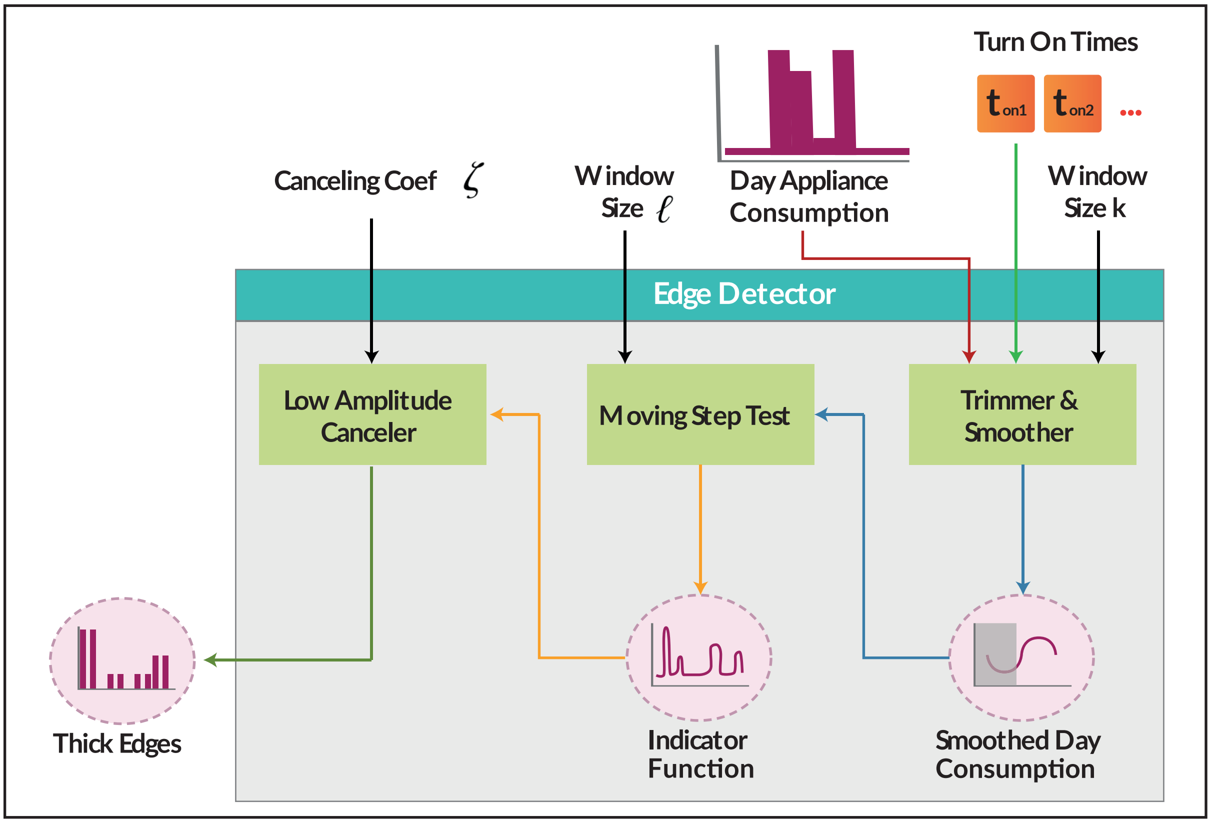


Figure 7.2: Edge detector architecture.

7.1.1 Edge Detector

The classification of the detected SUP into operation modes requires determining the cycles (as described in Chapter 4) that form the SUP.

The day consumption is represented by $D(t)$. Given a turn on time for a SUP, t_{on}, we define $D'(t)$ as follows:

$$D'(t) = D(t): \quad t \in [t_{on}, midnight] \tag{7.1}$$

where $D'(t)$ represents the consumption after the turn on time to the rest of the day. A median smoother function is then applied to $D'(t)$ to remove low amplitude noise component.

Edge detection is based on the Moving Step-Test (MST) [35]. MST is characterized by the following equation:

$$I(t) = |m_{t+\ell} - m_{t-\ell}| \tag{7.2}$$

where $I(t)$ is defined as the *Indicator Function*. In this work $m_{t+\ell}$ is the median value of the power consumption of the sequence that starts at $D'(t)$ and ends at $D'(t+\ell)$ and $m_{t-\ell}$ is the median value of the power consumption values that starts at $D'(t-\ell)$ and ends at $D'(t)$. The median function is chosen over the mean since the median gives more accurate edge location since it eliminates the presence of points that are located at the other end of the transition around the edge.

If the value of $I(t)$ is higher than a certain threshold, this indicates the start or the end of a cycle. Each period of time (t_s, t_e) where $I(t)$ is greater than the threshold forms a *Thick Edge* i.e, a thick edge is period of time defined by a starting time t_s and ending time t_e where $I(t)$ is greater than a certain threshold. A thick edge in $I(t)$ indicates that the values of the two medians $m_{t-\ell}$ and $m_{t+\ell}$ are far from each other.

A cycle is defined by two abrupt changes in the value of $D'(t)$. An abrupt change in $D'(t)$ is a single point of time $t_m \in (t_s, t_e)$ which is referred to as *Exact Edge*. Exact edges are determined based on the values of $I(t)$.

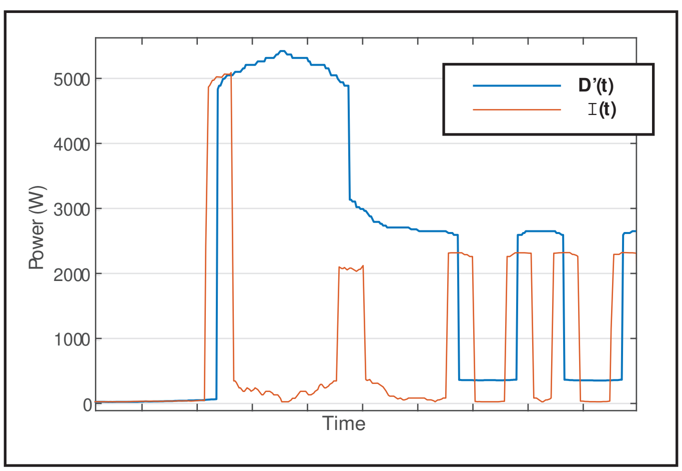


Figure 7.3: The response of the indicator function $I(t)$ for the abrupt changes in $D'(t)$.

The function $I(t)$ facilitates finding abrupt changes in $D'(t)$, rather that looking directly in $D'(t)$ for finding the abrupt changes. These abrupt changes in $D'(t)$ occurs in different power levels i.e., an abrupt change could occur when the value of $D'(t)$ is around 5W and jumps to 500W, or it could jump from 3000W to 5000W. Also, abrupt changes in $D'(t)$ occur from lower values to higher values or vise versa i.e., an abrupt change could occur when the value of $D'(t)$ is around 5W and jumps to 2500W or from 2500W to 5W. On the other hand, $I(t)$ shows all the abrupt changes in $D'(t)$ in a single reference i.e., whenever there is no abrupt changes in $D'(t)$, $I(t)$ shows a value close to zero. Also, whenever an abrupt change occurs in $D'(t)$, $I(t)$ shows a spike with high value in a short period of time.

The threshold τ depends on the standard deviation of $I(t)$ and Low Amplitude Canceling Multiplier ζ. This multiplier adjusts the threshold in order to cancel whatever values of $I(t)$ that is less than the threshold value. This is graphically depicted in Figure 7.4.

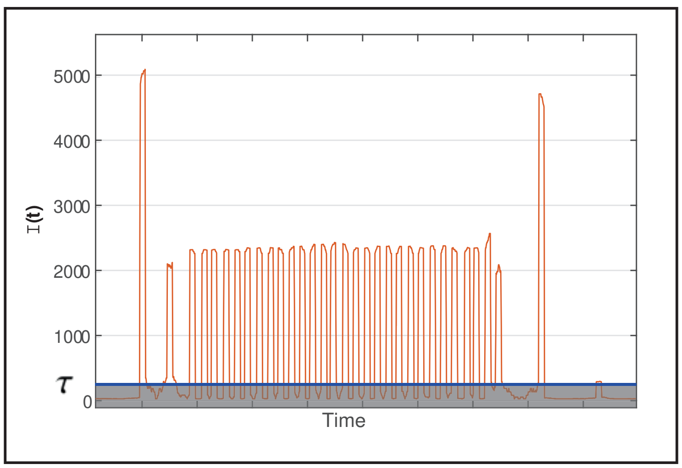


Figure 7.4: The formation of $I'(t)$ after placing the threshold line τ.

The value of the threshold τ is defined as follows:

$$\tau = \zeta \times \sigma(I(t)) : \quad \zeta \in \mathbb{Z}^{+} \tag{7.3}$$

The trimmed version of $I(t)$ as presented in Figure 7.4 defines $I'(t)$ as the following:

$$I'(t) = I(t) - \tau \tag{7.4}$$

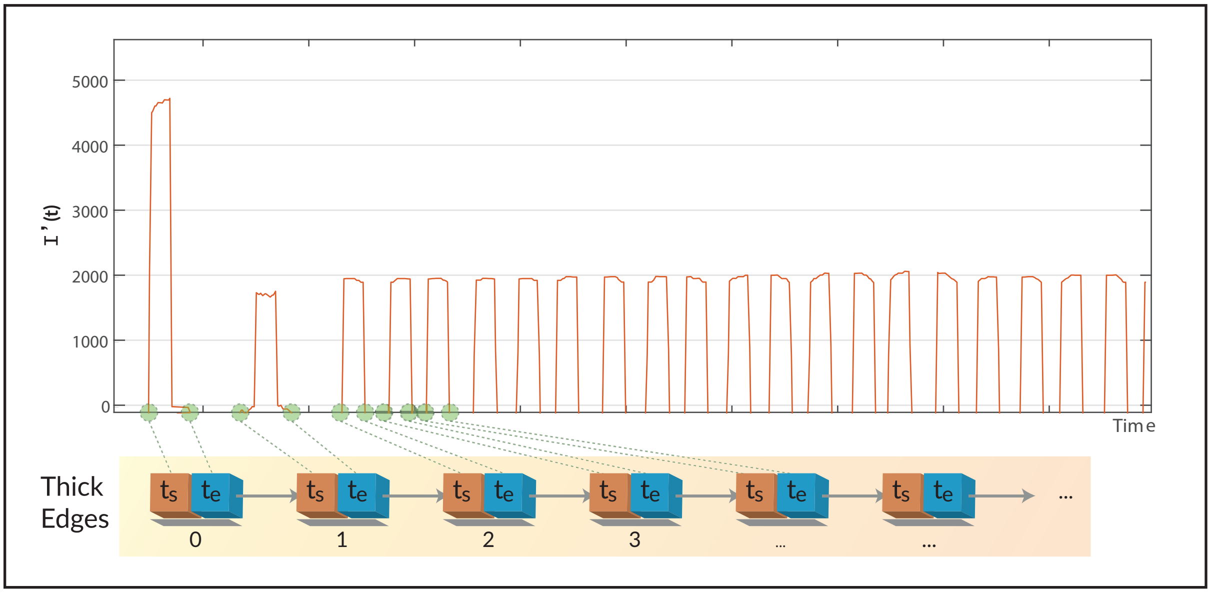


Figure 7.5: Thick edges extraction from $I'(t)$ as list of pairs.

We use $I'(t)$ to determine the thick edges , E, within a SUP such that:

$$\begin{aligned} E &= \{e_0, e_1, .., e_{n-1}\} \quad \text{where n is the size of E} \\ e_i &= (t_{si}, t_{ei}) : \quad 0 \le i < n \end{aligned} \tag{7.5}$$

where t_{si} and t_{ei} represent the start time stamp and end time stamp of thick edge i respectively. Figure 7.5 illustrates the extraction of the thick edges. It shows the plot of $I'(t)$ and a list of pairs of time-stamps. The i^{th} pair defines the start time (t_{si}) and end time (t_{ei}) of the i^{th} thick edge.

7.1.2 Cycles Extractor

In this section, we discuss the process of extracting cycles of a SUP based on the thick edges determined in Section 7.1.1.

7.1.2.1 Edge Thinning

Edge thinning refers to selecting the point of time t_m of the exact edge from a thick edge (t_s, t_e). We pick t_m such that it falls in the middle of the thick edge, so t_m is defined as:

$$t_m = t_s + \frac{t_e - t_s}{2} \tag{7.6}$$

where t_s is the start time of the thick edge and t_e is the end time of the thick edge. Eq. (7.6) is applied across all thick edges in E. All thick edges are thinned and grouped in the Exact Edges Set $X = \{t_{m_0}, t_{m_1}, t_{m_2}, ...t_{m_n}, \}$ where t_{m_i} represents the i^{th} exact edge and n is the number of thick edges, consequently, equals the number of exact edges.

Figure 7.6 (a) demonstrates edge thinning where plots of $I'(t)$, $D'(t)$ are shown. The set of thick edges E is displayed as pairs of times (t_s, t_e). These time stamps are shown in Figure 7.6 (a) as gray dotted lines surrounding each thick edges in both sides. As the definition of the exact edge t_m in Eq. (7.6), exact edges set X are displayed as the yellow time stamps pointing to the middle of each thick edge period with a dashed green pointer.

7.1.2.2 Extracting Cycles

The next step is to extract cycles of $D'(t)$ where each cycle is defined by two consecutive exact edges $\{t_{m_i}, t_{m_{i+1}}\}$ and the power value of this cycle m_i. To calculate the power value for the cycle i, we define the sequence Y as the power values of $D'(t)$ between the two exact edges $[t_{m_i}, t_{m_{i+1}}]$. We then calculate the median of Y such that:

$$m_i = Median(Y): \quad t \in [t_{m_i}, t_{m_{i+1}}] \tag{7.7}$$

The calculations of the cycle values are demonstrated in Figure 7.6 (b). The exact edges are displayed in yellow boxes pointing upwards towards $D'(t)$ with dashed green lines. The cycle power is defined by the median of $D'(t)$ between two adjacent exact edges. This is depicted as a black horizontal line representing the cycle's power value within the exact edges (t_{mi}, t_{mi+1}).

Finally, the cycles set C is formed by collecting what defines a cycle into single tuple,

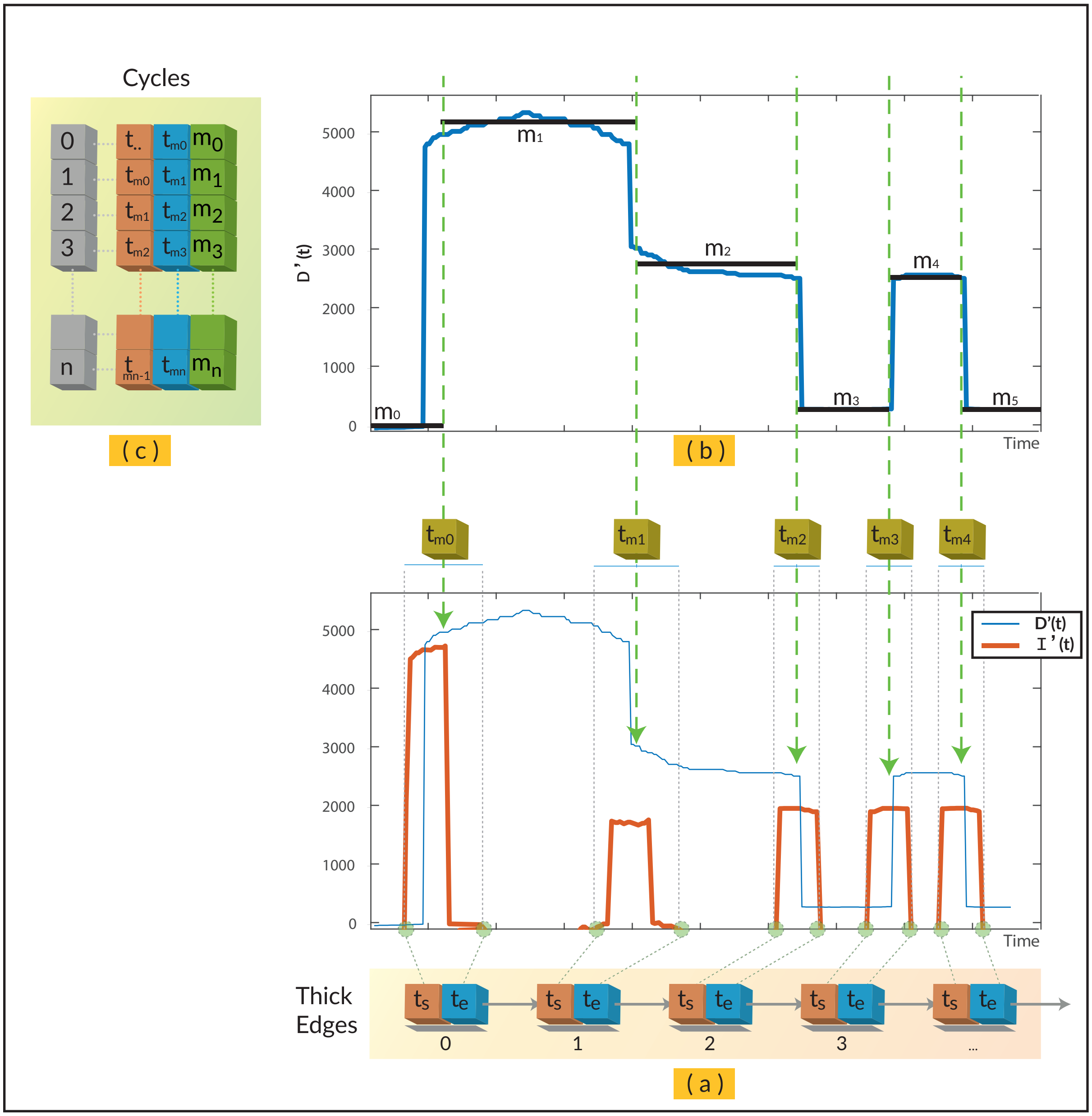


Figure 7.6: **(a)** Edge thinning. **(b)** Extracting the exact edges. **(c)** The cycles set.

where each tuple e_i hold the start time, end time, and the power value. Therefore, the cycles set C with n tuples is defined as the following:

$$\begin{aligned} C &= \{e_0, e_1, .., e_{n-1}\} \quad \text{where n is the size of C} \\ e_i &= (t_{s_i}, t_{e_i}, m_i) : \quad 0 \leq i < n \end{aligned} \tag{7.8}$$

where, the cycle i is defined by the enclosing two exact edges, t_{s_i} the edge at the cycle start, and t_{e_i} the edge at the cycle end, and the power value of the cycle m_i.

Figure 7.7 shows the steps to extract the features. Figure 7.7 (a) shows the $D'(t)$ after smoothing in order to detect edges and extract cycles. In Figure 7.7 (b) the cycles are extracted in the cycles set C. Then using this set, a traced version of $D'(t)$ is sketched based on the information enclosed in the cycles set C.

7.1.3 Clustering Cycles Into Levels

This section discusses the use of the cycles set C to extract features that is used to classify the detected SUP into the operation mode.

7.1.3.1 Clustering Cycles Using K-Means Algorithm

In this section, the goal is to cluster the cycles in C into clusters based on the power level of each cycle. We chose the k-Means Clustering Algorithm [65, Chapter 20]. K-means algorithm is considered a good choice for numerical based features as stated by Niennattrakul et.al.[75].

The observations that are fed to the k-means algorithm consist of the power level m_i for each cycle $e_i \in C$ except the last cycle e_{n-1}. This cycle represents the rest of the day in case of a single SUP detected, or it represents the time period between multiple detected SUPs. During this cycle the power value is close to zero which corresponds to a turned off appliance, which is not part of the SUP to be classified. The observations set $B = \{m_0, m_1, ..., m_n\}$ where m_i is the cycle $e_i \in C$ power level, and n is the number of SUP cycles. The k-means algorithm requires the definition of the number of clusters k in advance. Therefore we choose the value of k based on the data the we have analyzed in Chapter 3. We choose $k = 3$ since the power values of the cycles for all appliances that we analyzed is three power levels. Therefore, K-Means algorithm produces k mutual exclusive clusters sets $L_0, ..., L_{k-1}$ where the union of all the clusters represents the observation set such that:

$$\bigcup_{i=0}^{k} L_i = L_1 \cup L_2 ... \cup L_{k-1} = B$$

K-Means computes k centroids set $R = \{r_0, r_1, ..., r_i, ..., r_{k-1}\}$. Each centroid, r_i , represents the center point where all the elements m_j belong to the cluster L_i have minimum distance with.

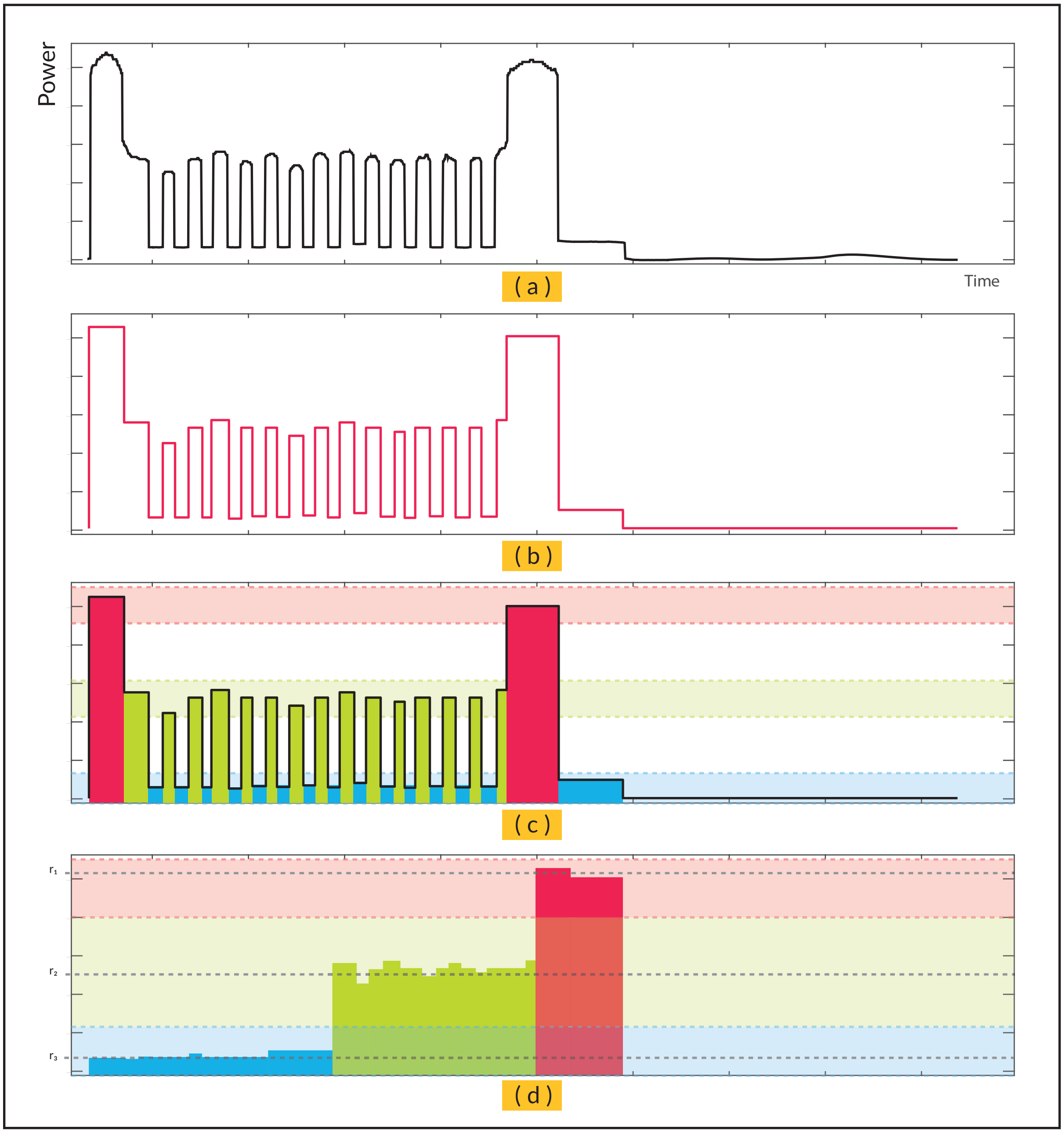


Figure 7.7: **(a)** The smoothed function $D'(t)$. **(b)** a traced version of $D'(t)$ using the cycles extracted. **(c)** The cycles in different colors based on the power value of each. **(d)** Cycles clusters.

Figure 7.7 (c) shows an example of the cycles in different colors based on the power value of each cycle. The three highlighted areas indicated by the centroid values points r_1, r_2, r_3 show the power range for each group of cycles.

The features set X is calculated based on the clusters in B, plus the centroids set R. We first sort the cluster ascending order based on the centroid value for each cluster such cluster 0 has the lowest centroid value and cluster $k - 1$ has the highest centroid value. We then use each cluster L_i to define a feature x_i in the features set X. Therefore, the feature set X is defined as:

$$X = \{x_0, x_1, ..., x_{k-1}\} \tag{7.9}$$

where each feature x_i is defined as the total duration of the cycles e_j within the cluster, L_i, multiplied by the average power level of the cluster, which in this case is the value of the centroid of the cluster, r_i. Each feature x_i is modeled as follows:

$$x_i = r_i \sum_{j=0}^{size(L_i)} |t_{e_j} - t_{s_j}| \tag{7.10}$$

where t_{e_j}, t_{s_j} are the two points of time which define a cycle, as stated in Eq. (7.8). Figure 7.8 visually depicts features formation.

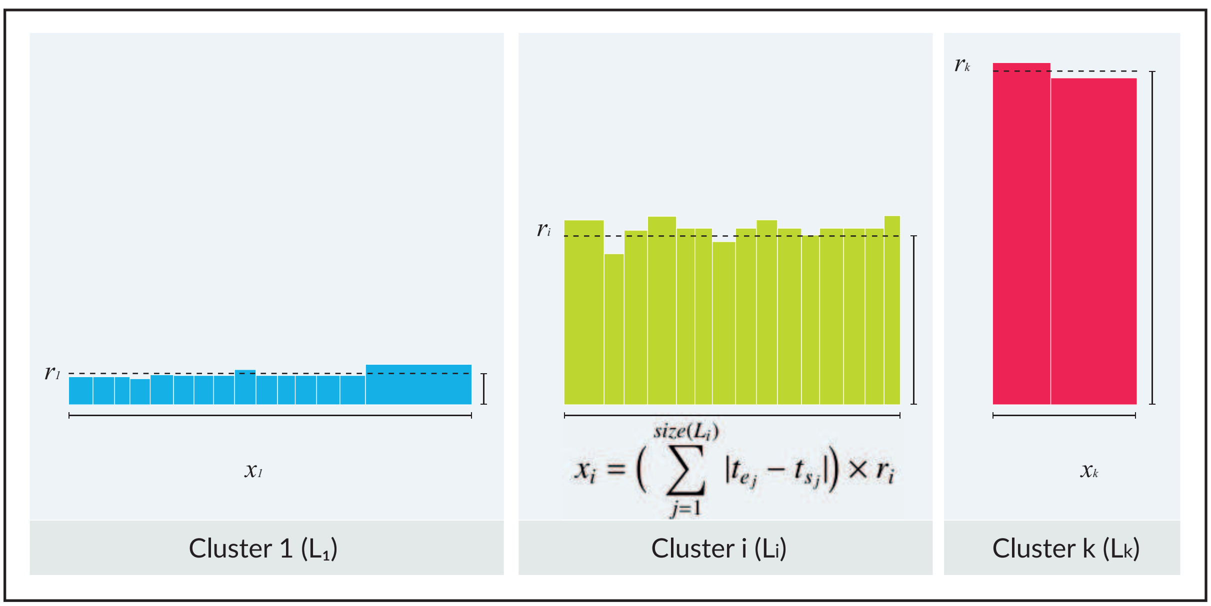


Figure 7.8: Visual representation for feature calculation.

7.2 Classification Of SUP To Operation Modes

This section discusses the classification method of the SUPs into operation modes based on the features set X that is calculated based on a clustering mechanism for the SUP cycles.

7.2.1 Training Data

Since we are solving our main problem using a classification technique, it is necessary to have a dataset that is used to train the KNN model. In other words, because of the nature of the KNN algorithm which is a lazy learning algorithm, it is important to initialize the KNN model and compute initial values for its centroids so that it can later on work online and classify any new queries. This initialization is considered as a kind of training stage for the KNN model which can not be obtained without an already labeled dataset.

The training dataset we use is a synthetic dataset which is generated by our simulator discussed in Chapter 4. The dataset consists of three subsets, where each subset consists of 4000 observations for certain appliance (dishwasher, clothes dryer, clothes washer). The training set that is generated to train the KNN model consists of a group of observations where each observation represents a single SUP as a set of features X with size k as defined in Eq. (7.9). A SUP is generated, then using the Representation Object the average power value m_i of each cycle type is calculated. Afterwards, Eq. (7.10) is applied to calculated the value of each feature x_i. This is done for each operation mode to each appliance.

The model demonstrated in Figure 7.1 shows that the classification module depends on the data generated from the simulator. The simulator generates a synthetic dataset that consists of all possibilities of operation modes for appliance SUP with variations through changing tuning parameters that makes the dataset diverse. The simulator is already configured by giving the description of the SUP cycles in terms of duration, power level, and repetition. Since all this information is already known by the simulator, it is straight forward to convert these configurations into a form of feature vector that matches what we have described in this chapter.

7.2.2 K-Nearest-Neighbors KNN Algorithm

The algorithm we use in the classification is the K-Nearest-Neighbors (KNN) [18]. It is a classification algorithm that is based on a voting scheme. An item is classified by a plurality voting of its neighbors. The item is then assigned to the class most common between its k nearest neighbors where $k > 0$. For example, If $k = 1$ the item then is assigned to the class of that single neighbor.

One of the advantages that KNN gives to our study is that KNN is a Lazy Learning Algorithm. This means that the generalization of the training set is postponed until a query is made to the algorithm. In other words, KNN keeps track off (at most) all available data all the time until a query of classification is made so it does the calculations for all the available data with the query. In contrast with the Eager Learning Algorithms, the training set is summarized into a model so that when a query is made the model is enough to do the decision apart form using the training data again.

Another advantage for using KNN is its suitability for Online Recommendation Systems such online stores that recommend certain items to the customer [51]. The reason for its suitability is that the data is continuously updating, as it updates part of the data my be considered obsolete because of certain trend in the market. Therefore, summarizing the available data into a model and classify upcoming queries based on this model my lead to low precision results.

As the feature set is extracted (see Section 7.1), the features set takes the form $X = \{x_0, x_1, .., x_i, .., x_{k-1}\}$ where k is the number of features in the features set. Then both training set and features set are fed to the KNN classifier in order to classify each SUP into one of n operation modes within the set $M = \{d_0, d_1, ..., d_{n-1}\}$.

CHAPTER 8

Implementation

In this chapter we present the implementation. Figure 8.1 shows the architecture used in the application development. We used the Model View Controller (MVC) design pattern with the N-tier software architecture. MVC provides separation of concerns between the data, functionality, and presentation. N-tier provides layering concept that gives fine grain separation of concern within each component of the MVC [93].

Three different endpoints are shown in Figure 8.1: DB server, web server, and a client.

- **DB Server:** The DB server stores the data in database. This database is managed by a Database Management System (DBMS) to create the tables and seed them with initial data. We used Microsoft SQL Server (MSSQL Sever).
- **Web Server:** This is the web server that hosts the web application. We used the Internet Information Services Express (IIS).
- **Client:** Which is in this case a web browser.

8.1 The Database

We use Microsoft SQL Server 2018 (MSSQL) Database engine [70]. The Entity Relation Diagram (ERD) is depicted in Figure 8.2. This is a graphical representation of our system that depicts the relationships between the system entities which include people, objects, and concepts with the attributes that define each entity [102].

The following is the description of each entity:

1. **Household:** This entity represents a single household. It is identified by a serial number (ID), a household name (name), and it has a usage intensity attribute (UsageIntensity) which correspond to level of intensity this household belongs to. The (UsageIntensity) attribute takes one of the following values: High, Medium, and Low, as discussed in Section 4.2.2.
2. **ApplianceType:** This entity refers to the different appliances that could be used in a house. It is identified by a serial number (ID), and a name (Name). The values of

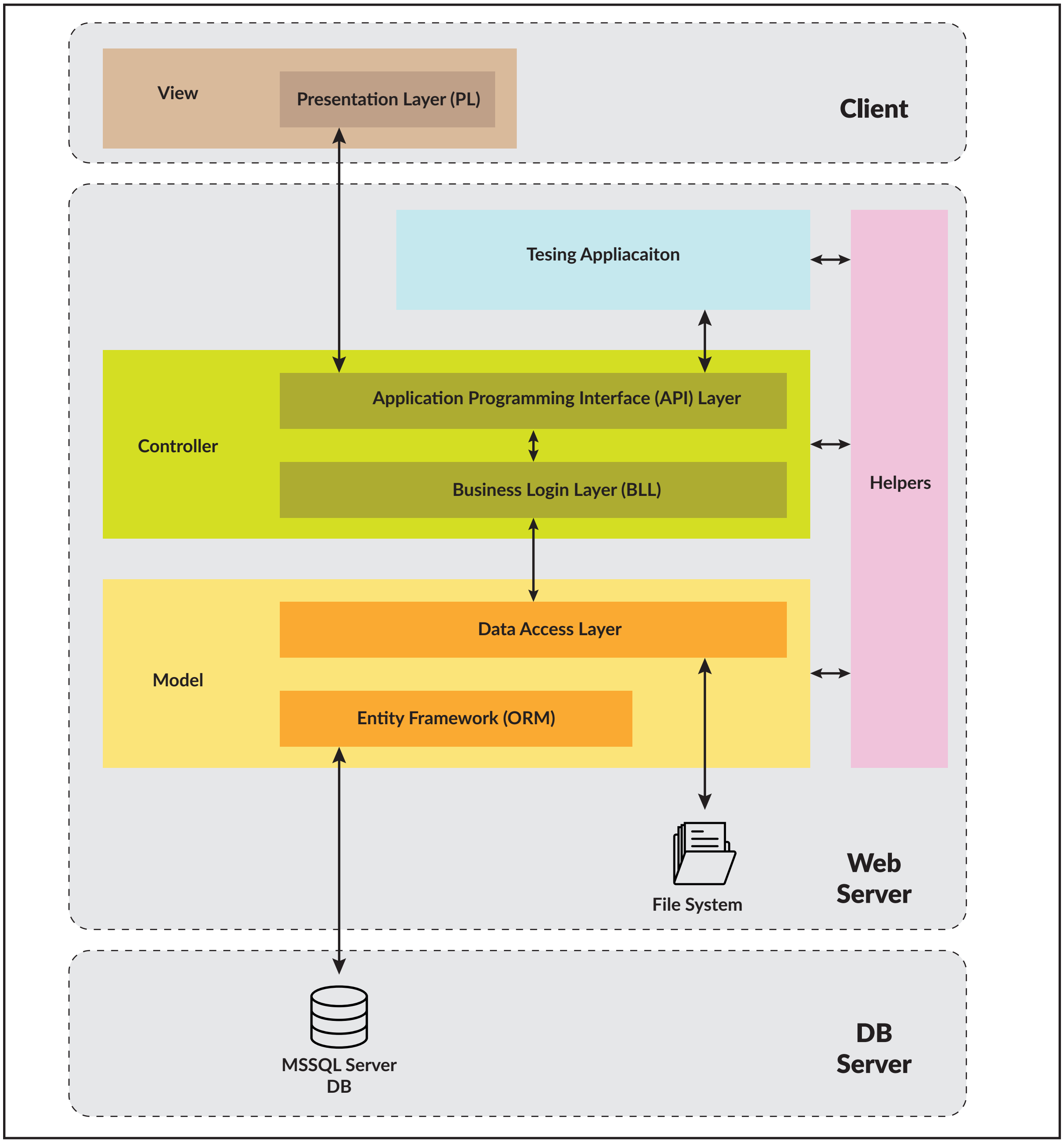


Figure 8.1: The architecture of the application.

this entity are the names of appliances monitored. The appliances include: dishwasher, clothes washer, clothes dryer.

3. **OperationMode:** This entity corresponds to the operation mode that has been used when activating a certain appliance. It is identified by a serial number (ID), and a name (Name). The values of this entity are the operation modes that can be run by the appliances, these

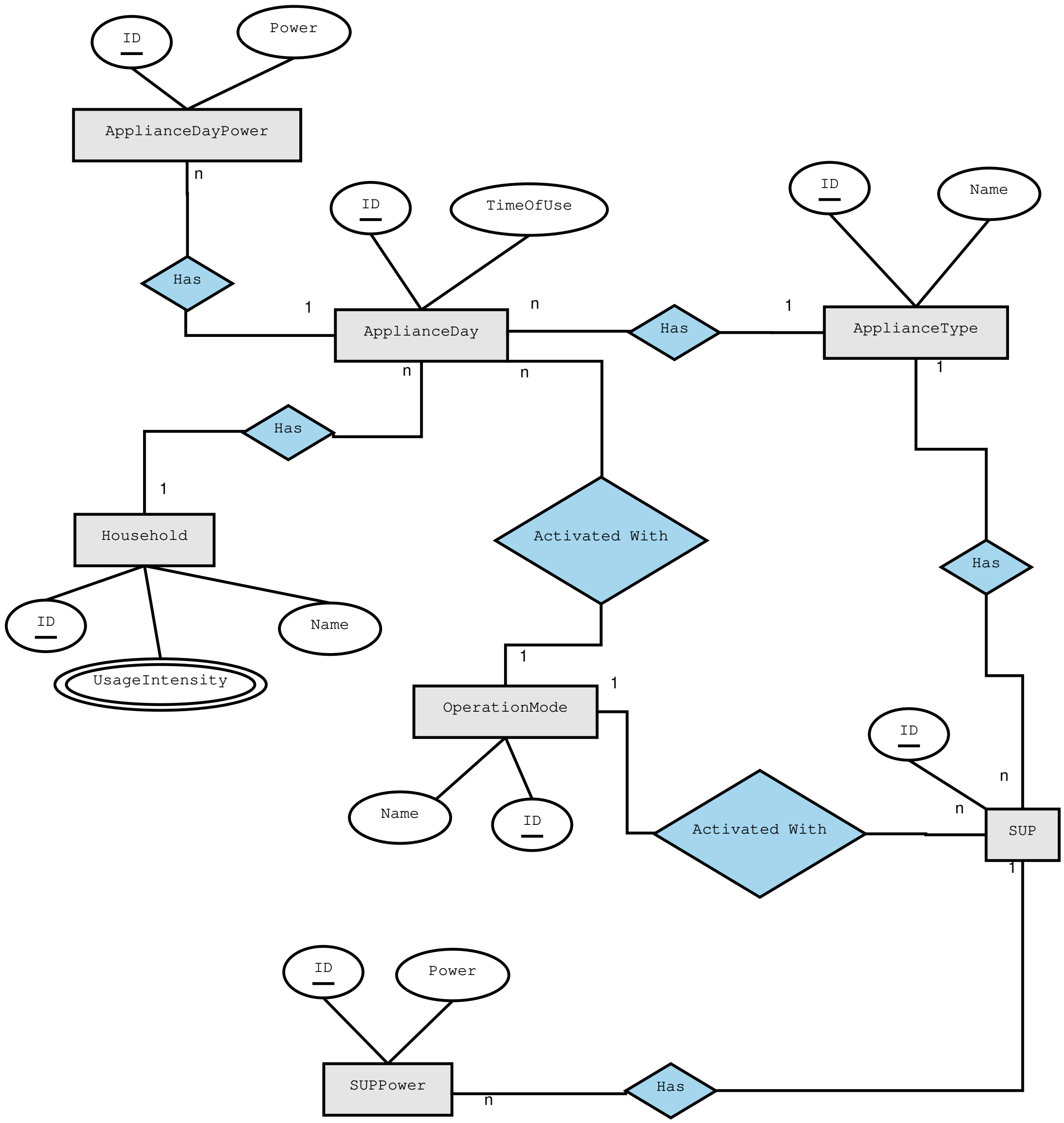


Figure 8.2: Entity Relationship Diagram (ERD)

operations modes are: Light, medium, and heavy.

4. **ApplianceDay:** This entity refers to a single day of consumption for an appliance that is run by certain household. The activation time is specified by the attribute (TimeOfUse).

5. **ApplianceDayPower:** This entity corresponds to the power samples that belongs to a single day. Each power sample has a value in watts (Power) and identified by a serial

number (ID).

6. **SUP:** This entity represents the Single Usage Profile (SUP) for an appliance using a specific operation mode.

7. **SUPPower:** This entity corresponds to the power samples that belongs to a single SUP. Each power sample has a value in watts (Power) and identified by a serial number (ID).

8.2 The Web Application

This section discusses the components used in building the web application.

8.2.1 ASP.Net

We used The ASP.NET framework to develop our system. Model View Controller (MVC) is an open source web application framework from Microsoft, which applies different design patterns such as the traditional Web Forms and the Model View Controller (MVC). It is now an open-source software under the Microsoft Public License (MS-PL) [69], apart from the ASP.NET Web Forms component which is proprietary[20]. ASP.Net propose different development environments that suits different preferences and situations.

In this work, we used ASP.net MVC framework with C# as a backend development language. The development took place using Microsoft Visual Studio 2019.

8.2.2 Model View Controller (MVC) Framework

The MVC pattern splits an application into three parts: the Model, the View and the Controller. A Model represents the application data that is part of the persistent state and also contains the logic for accessing and manipulating this data. The View is used for rendering the state of the model. The Controller is responsible for translating user input into actions to be handled by the model. Moreover, it is responsible for selecting what to view based on the user inputs and the resulting operations of the model [43]. Figure 8.3 shows the MVC architecture.

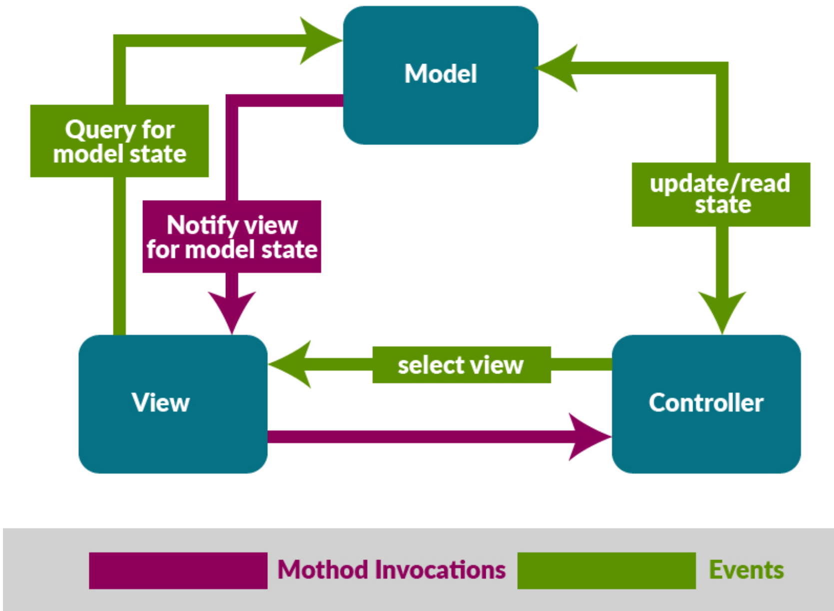


Figure 8.3: Model View Controller (MVC) Design Pattern Architecture

8.2.3 The Model

The Model component corresponds to **all** the related data logic that the user deals with. It includes the data itself and the logic behind processing and communicating this data. In this work, the data comes from two main sources: The file system , and the database. First, we store three different files in the file system and our application read them through file steaming. These files are the SUP configuration object as a JSON file, the household usage intensity and turn on time distributions as simple CSV files. Second, the data stored in the database fetched using an Object Relational Mapper (ORM) which is a technique that is used to query and manipulate data from a database using an object-oriented programming languages.It encapsulates the logic needed to process the data from and to the database. We use Entity Framework (EF) [71] as the ORM implementation that is provided by Microsoft. EF produces classes and facilitates reading and writing data back and forth from the DB. In case of querying the DB, it retrieves the data as objects of the mapped classes, or as arrays of these objects in case of vector results.

To execute custom queries and send arguments, we introduce a layer on top of the EF that serves as container for the queries to the DB. This layer is the Data Access Layer (DAL). DAL contains functionality for Creating, Returning, Updating, Deleting (CRUD) items into the database, plus the custom queries. In this layer, each entity (or class) has its own repository of functions to read and write the data. The other main role for DAL is to convert Entity Objects (EOs) produced by EF, to Data Transfer Object (DTOs). A data transfer object is an object that carries data between functions in different conceptual layers. What is different about a DTO is the data and functionality hiding, since this object is then exposed to the upper layers, then to the end user. For example, the table *ApplianceType* has the attributes *ID, Name*. The EO which corresponds to it is a class *ApplianceType* that has the fields *ID, Name*. Since the user is only interested in the *Name* of the appliance, not the *ID*, a DTO is produced with the name *dtoApplianceType* that has only the *Name* field.

8.2.4 The Controller

The controller is responsible for controlling the application's logic and coordinates between the View and the Model. The controller receives data from the users through the View or from the DB through the Model. The controller then processes this data, and either passes the results to the View or push the data to the DB via the Model. In our implementation, we divided this component into two layers: The Business Logic Layer (BLL) that contains the core business logic of the application , and the Application Programing Interface (API) Layer which exposes the data to the front-end.

8.2.4.1 The Business Logic Layer

In this layer, all the algorithms proposed in this work is fully implemented. This layer is divided into five maim components:

1. **The Simulator Component:** This component implements all the functionalities used by the simulator. We implemented the ITS algorithm using hash tables to get random samples of usage intensity and turn on times. The algorithm that reads the SUP representation object and translate this representation with the phases and cycles into a list of integer power values that represent a single SUP. Also the algorithm that puts an SUP at the specified time of the day is implemented.

2. **The SUP Detector Component:** This component contains the implementation for the

templates slicing mechanism, the cross correlation process, low amplitude cancellation, and the peaks detector.

3. **The DTW Component:** This component mainly focuses on the implementation of the DTW algorithm. We implemented the basic DTW algorithm but we didn't use the recursive version since it is expensive in terms of processing time. Instead, we implemented the DTW algorithm using 2D array. Also in this component we implemented the segmentation mechanism of the daily power list and and classification logic for SUPs.

4. **The Machine Learning Component:** In this component we used a ready made machine learning framework that is built on top of .Net framework using C#. We used Accord.Net [3]. The Accord.NET Framework is a free .NET based machine learning framework that is capable for building production grade computer vision, signal processing, and computer audition applications. In this work, we used the core functionalities of the framework with the machine learning module to complete our clustering in k-means and classification in KNN.

5. **The Statistics Component:** This component contains the implementation for all statistics functions that is not supported by the .Net framework. For example, the mean, maximum, minimum are ready made functions in .Net. On the other hand, an implementation from scratch is performed for functions such as median, mode, slandered deviation. Also we implemented the smoothing filters such as moving average and moving median.

8.2.4.2 The API Layer

This layer contains the final functionality that is exposed as RESTful APIs to the View or other applications that can integrate to our system over HTTP. We utilize the ASP.NET WebAPI to achieve this.

The ASP.NET Web API is an expandable framework for building HTTP based web services. These services can be accessed and consumed from different applications using different platforms such as web , windows, mobile etc., regardless of the invoking language. It sends the results on invoke as HTTP Response that takes many formats. We configured our API to respond to the requests as JSON format. For example, an API service is *getAllHouseholds()*, this service pulls all the available households in the database, convert the names of these households into an array of strings and encapsulate this array within a JSON object. This object is returned back to the invoker as a body inside an HTTP Response.

8.2.5 The View

The View is known also as The Presentation Layer (PL). This layer is responsible to deal directly with the end user. It has two main duties: First, it renders the data consumed from the back-end. Second, it manages the data and interactions performed by the end user. We utilized the following technologies in the presentation layer: jQuery, and Bootstrap.

8.2.5.1 jQuery

jQuery is a free open-source JavaScript library designed to make it easier for developers to manipulate and traverse HTML DOM tree, as well as event handling, CSS animation, and AJAX. jQuery simplifies listening to the controls events that is fired by user interaction, and simplifies the code required to collects forms data, initiate AJAX requests to the server and post form data, receive AJAX responses that contains data from the server, and bind the data back to the controls and forms in the web page.

8.2.5.2 Bootstrap

Bootstrap is an open-source front-end framework developed by Twitter. It is a combination of HTML5, CSS3, and Javascript code designed to build user interface components in the client side, without the intervention of the server. Bootstrap is a free toolbox for creating a web application. It has all the common web controls that suits all kinds of controls that is used to gather inputs from the end user or to display server data to the end user.

8.3 The Hardware

The hardware setup used for deploying the prototype is a PC with Intel(R) Core(TM) i5-7500T CPU @ 2.7GHz Quad core with 8GN RAM. The operating system is Microsoft

Windows 10 Enterprise N 64-bit. The SQL server is deployed on the same host machine as well as the IIS web server.

CHAPTER 9

Performance Evaluation

9.1 Detecting SUPs Results

This section discusses the performance evaluation of the SUP detection.

9.1.1 Evaluation Metrics

We used two metrics to evaluate the SUP Detection performance in our experiments. These are the number of detected SUPs, and the time difference between the actual and the detected turn on time.

The first metric is *the number of detected SUPs* for certain appliance, which corresponds to the number of reported turn on times for certain appliance by the SUP detection module. We assume that a single SUP presents daily for an appliance based on the analysis performed in Chapter 3, where we found that an appliance is activated on average several time a week or once a day. For example, if there is a single SUP activation in a certain day for a dishwasher, and after running the SUP detection algorithm it detected five SUPs while it is actually one SUP.

The other metric we use is *the time difference between the actual and the detected turn on time*. This corresponds to the time difference between the actual turn on time of a SUP and the reported turn on time by the SUP detection algorithm.

For the subsequent sections, we assume that the value of the Low Amplitude Canceling Coefficient δ is equal to 90 and the sampling frequency used is $f_s = 1Hz$ in the following tests.

9.1.2 The Impact of Reference Pattern Size

This section discusses the impact of the reference pattern size on the number of detected SUPs and on the time difference between the actual and the detected turn on time.

9.1.2.1 Impact of Reference Pattern Size on The Number of Detected SUPs

This section analyzes the impact of the size of the reference pattern has on successful detection of SUPs (see Chapter 5). A single SUP appears in each day of consumption.

A reference pattern is a time series sequence representing an SUP from the start of the SUP and is of size n where n is less than the length of the SUP. We generated reference patterns of different sizes based on a SUP generated with a randomly selected operation mode (see Chapter 4). Since the frequency of sampling (fs) is $1Hz$, the running time of the appliance in seconds can be calculated from the number of samples. For example, if a dishwasher has a SUP with 110 minutes, then this SUP is represented by 6600 samples. The reference patterns generated is a subsequence of the samples representing the generated SUP. For the dishwasher, the reference pattern sizes used ranged from 50 to 1400 samples. The range of the values of n is selected to show different cases of correlation, consequently, different results.

Figure 9.1 shows the results for a dishwasher. Figure 9.1 shows a very large number of detected SUPs found in the day (of 86400 samples) when n is between 50 and 400. This is because of the shape of the SUP of the dishwasher. When the reference pattern size is between 50 and 400, there is a high number of SUPs detected despite the existence of only one SUP. However, the number of detected SUP settles down to one when the reference pattern size typically increases within the range 400 and 1400 samples. When the reference pattern size exceeds 1400 samples, the number of detected SUPs is zero because of the low chance to find such a long reference pattern during the day.

The number of matching SUPs is higher for shorter reference pattern than a longer reference pattern. This is because there is higher chance that this small reference pattern occurs more frequently within the consumption function than a longer reference pattern. The reason is that a SUP has repetitions of cycles that have similar shape. This shape could be similar to the reference pattern. This leads to reporting multiple potential SUPs even where is only one SUP. Thus, as the size of reference pattern decreases, the probability of reporting these repetitions within a SUP is higher. Therefore, the accuracy increases as the reference pattern size increases.

Figure 9.2 and Figure 9.3 show results for clothes dryer and clothes washer respectively. The same range of the reference pattern size is used where n is between 0 and 1400 samples. Figs. 9.2 and 9.3 show that these two appliances have fewer fluctuations in the number of detected SUP in response to the change in n. The dryer shows that the value of n that gives best results is when n is between 600 and 1100 samples where it shows the number of detected SUPs equals to one. Nevertheless, the results show some choppiness in the number of detected SUPs

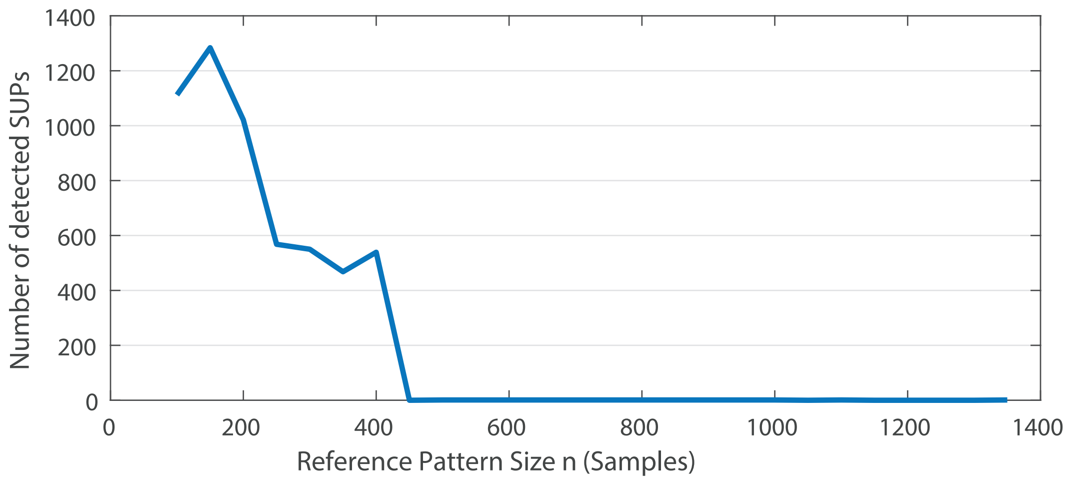


Figure 9.1: The impact of changing the reference pattern size n on the number of detected SUPs in the day consumption date for a dishwasher.

within the same range (when n is between 600 and 1100 samples). This is a common problem for all appliances. It is most likely due to the variations in cycles duration (see Chapter 4) where cycle durations are generated with variation factor that affects the cross correlation by reducing the overlapping between the cycles in the reference pattern and the detected SUP. Therefore, it shows fluctuation in the number of detected SUPs between zero and one depending on the variation coefficient selected by the simulator. Figure 9.3 shows the number of detected SUPs for the washer. The range of reference pattern size is between 100 and 1100 samples. In this range, the number of detected potential SUPs is always one. That is probably because the the repetition in the washer consumption curve is minimal, thus, a unique one high value of cross correlation in every test.

9.1.2.2 Impact of the Reference Pattern Size on The Detected Turn On Time

As we discussed in Chapter 5, the reference pattern is used to detect the start time of the appliances when it is turned on. This detection is computed by moving correlation, which may not always find the exact point of time that represents the actual t_{on}. The reference pattern size plays a role in the accuracy of finding the closest time to the actual t_{on}. The selection criteria of the reference patterns size n is the same as the criteria mentioned in Section 9.1.2.

The absolute time difference (in number of samples) between the actual time and the de-

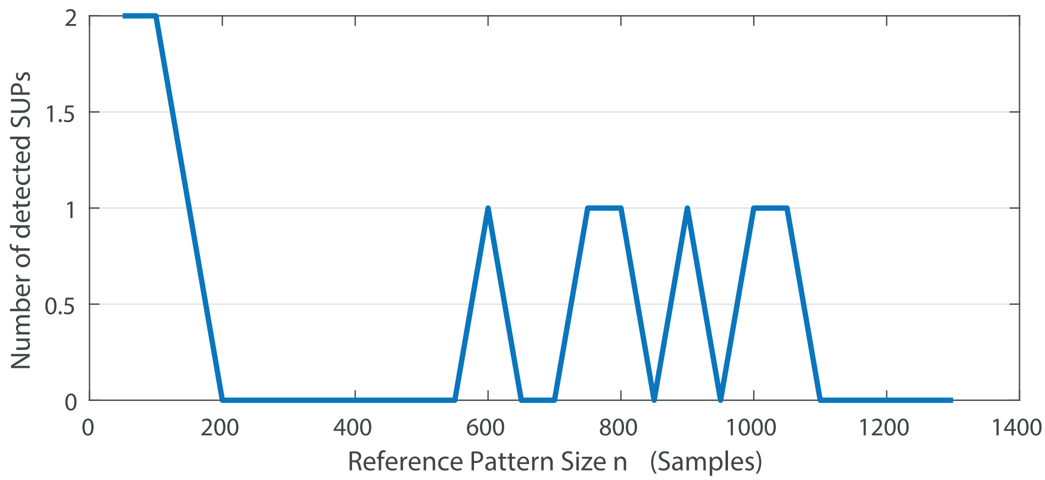


Figure 9.2: The impact of changing the reference pattern size n on the number of detected SUPs in the day consumption date for a dryer.

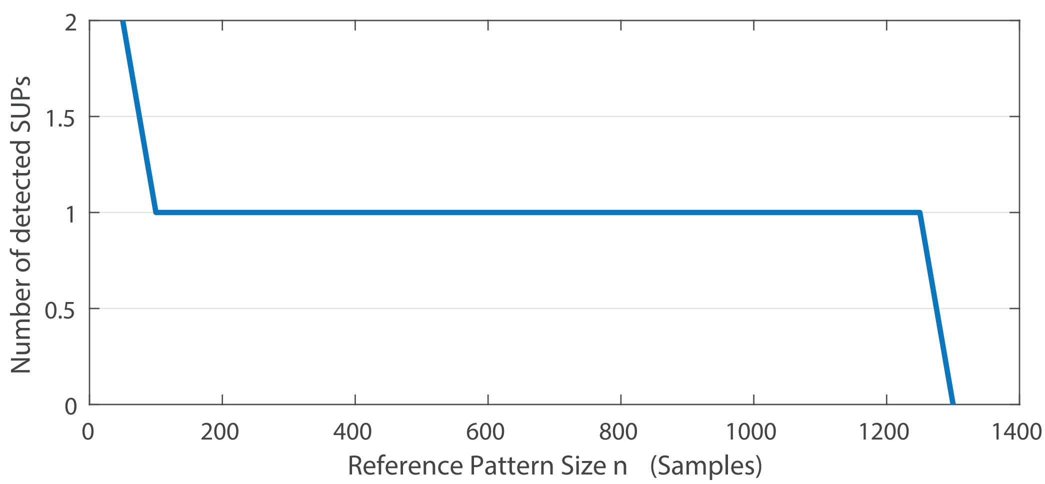


Figure 9.3: The impact of changing the reference pattern size n on the number of detected SUPs in the day consumption date for a washer.

tected time is as follows:

$$\Delta t_{on} = \left| t_{on}^{actual} - t_{on}^{detected} \right|$$

The relation between the reference pattern size n and the time difference for the clothes dryer is presented in Figure 9.4. The time difference trends higher as n increases. Generally, the time difference is greater than 20 samples for most of the reference patterns sizes. The best range of values for n is when it falls between 1050 and 1500 samples (1050 to 1500 seconds) where the lowest time difference is observed in this duration.

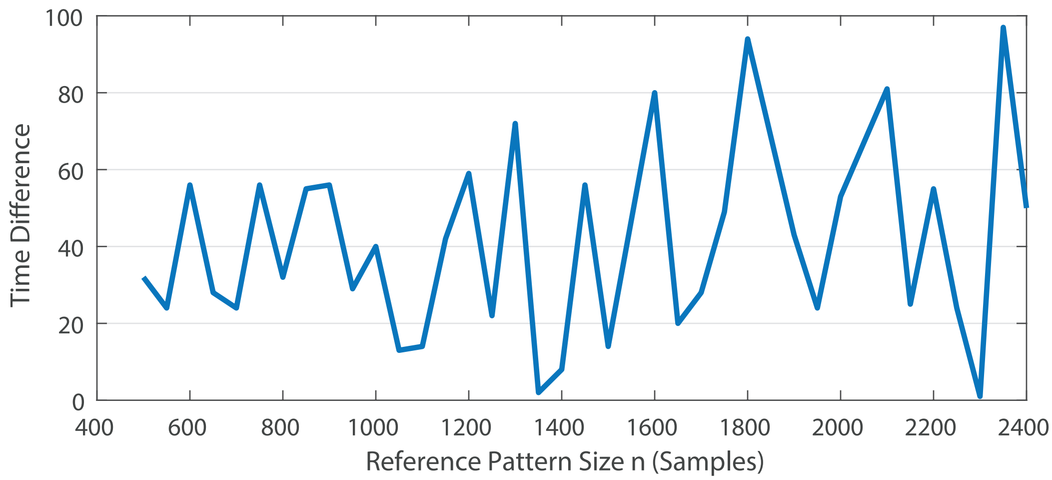


Figure 9.4: The impact of changing the reference pattern size n on the detected t_{on} for a dishwasher.

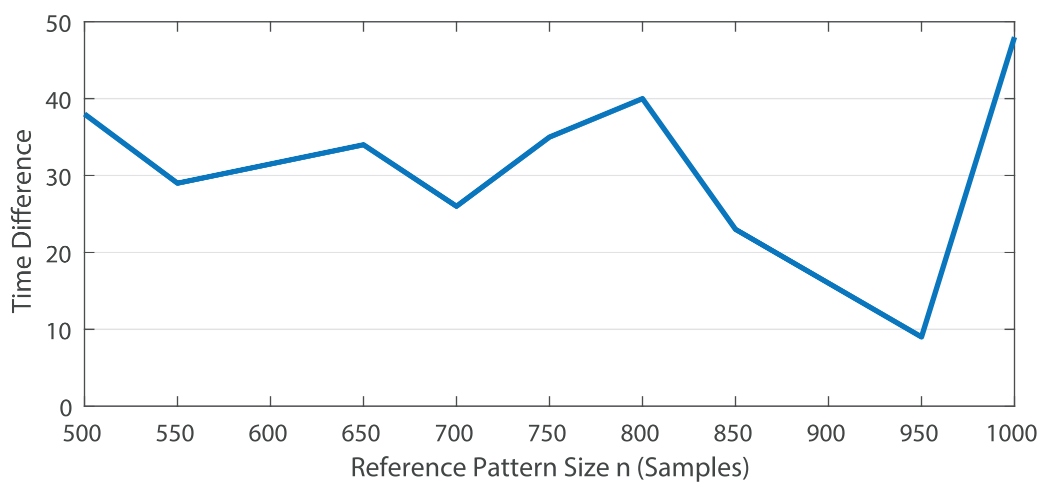


Figure 9.5: The impact of changing the reference pattern size n on the detected t_{on} for a dryer.

Figure 9.5 shows the relation between the reference pattern size n and the time difference for the dryer. The plot shows that the optimal size for the reference is when n between 900 and 950 where the time difference is minimum.

The impact of the reference pattern size n on t_{on} for the clothes washer is shown in Figure 9.6. There is a higher time difference when n is less than 700 because the reference pattern is relatively short, so it is more more likely to determine an incorrect t_{on}. Otherwise, the experiment shows the optimal size when the value of n is between 700 and 1050 where the error in time difference doesn't exceed 10 samples in this period.

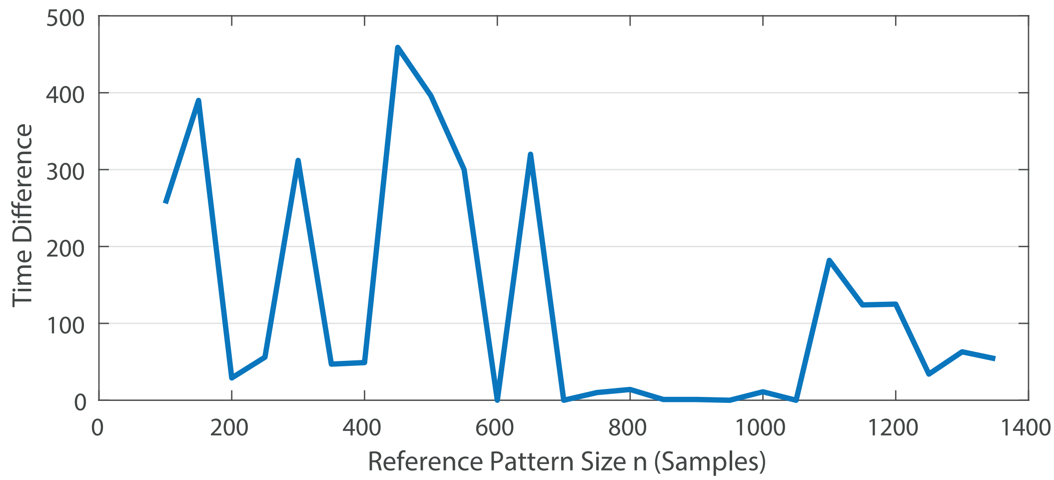


Figure 9.6: The impact of changing the reference pattern size n on the detected t_{on} for a washer.

9.1.3 The Impact of The Low Amplitude Canceling Coefficient δ on The Number of Detected SUPs

This section discusses the impact of the of the Low Amplitude Canceling Coefficient δ on the number of detected SUPs in a single day of consumptions data. The δ coefficient is used to form a threshold on the correlation function so that it trims what is under the threshold value and ends up with the residue function. This function has local maximums that represents the existence of a potential SUP. Generally, higher values of δ means a higher threshold, thus, spikes are less likely to be caught. On the other hand, a lower value for δ catches lower value spikes which may not resemble a correct SUP.

The following tests are performed as follows: For each appliance, a day of consumption data is generated. We assume that the actual number of SUPs in the day is one for each appliance. The operation mode is selected randomly. We repeated the test starting with the value of $\delta = 50$ to $\delta = 100$, increasing the value of δ by one every round. In each round, the number of the detected SUPs is reported.

Figure 9.7 shows the impact of changing δ on the number of detected SUPs for the dishwasher. The plot shows fluctuations between 4 to 2 detected SUPs when δ is between 50 and 80. In this period, the threshold value is relatively low which leads to detecting low amplitude correlation spikes. As the value of δ increases beyond 80, the threshold becomes higher making

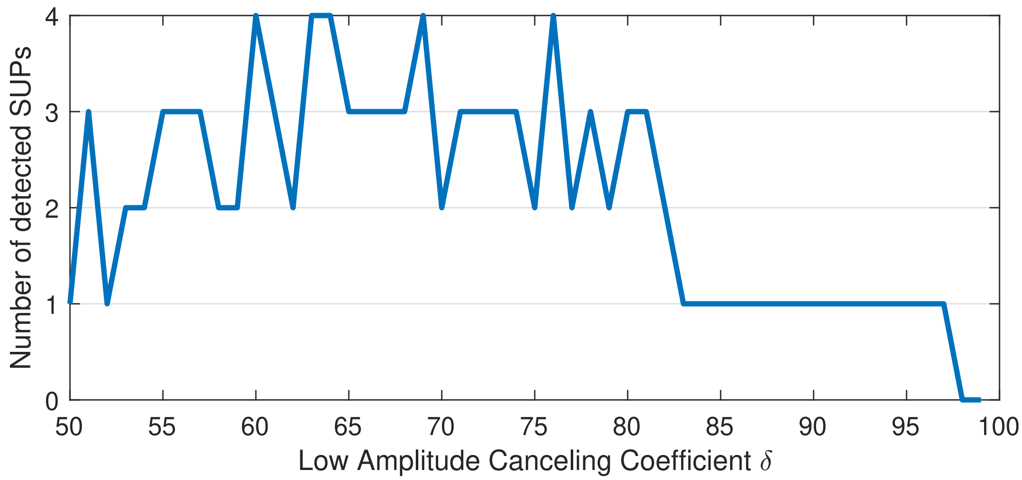


Figure 9.7: The impact of changing the Low Amplitude Canceling Coefficient δ on the number of detected SUPs in the day consumption date for a dishwasher.

it difficult to catch spikes except the actual spikes where the SUP really exists.

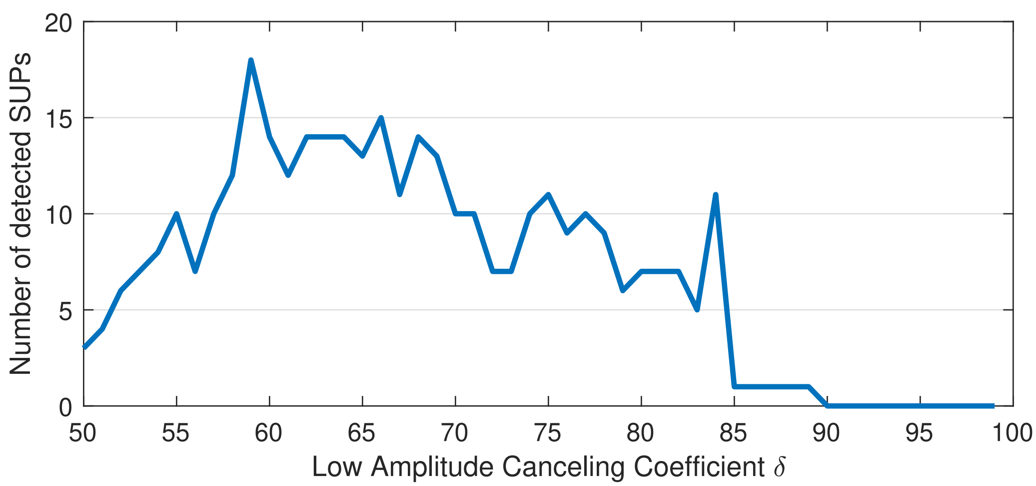


Figure 9.8: The impact of changing the Low Amplitude Canceling Coefficient δ on the number of detected SUPs in the day consumption date for a dryer.

For the case of the clothes dryer Figure 9.8 shows the impact of δ on the number of SUPs detected. The plot shows a wide fluctuation in the number of SUPs from 5 to 18 when the canceling coefficient δ is between 50 and 85. This is due to the shape of the dryer SUP where it has many segments of the data that is relatively similar to the reference pattern. For example, when $\delta = 84$ the algorithm identifies 10 potential times during the day where actually there is a single SUP. On the other hand, as δ increases between 85 to 90, Figure 9.8 shows the real number of SUPs which equals to one.

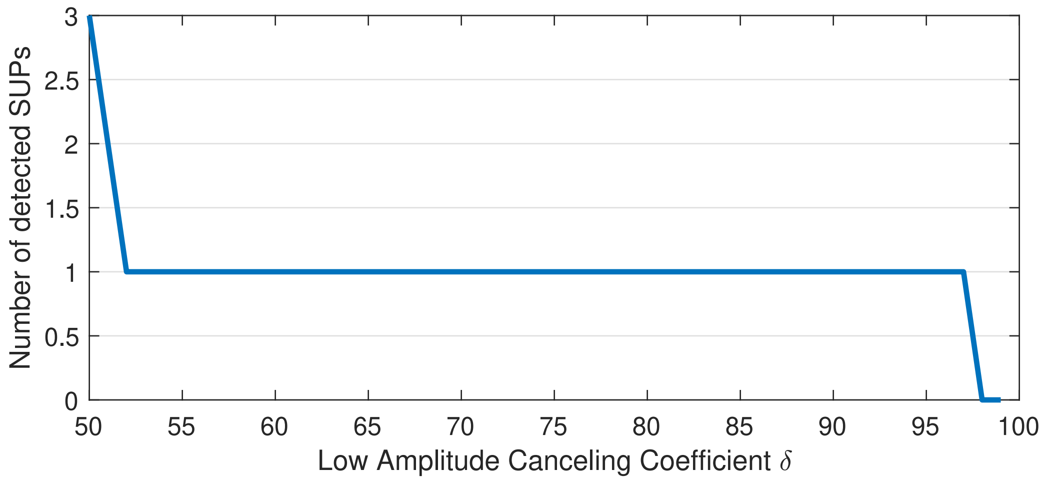


Figure 9.9: The impact of changing the Low Amplitude Canceling Coefficient δ on the number of detected SUPs in the day consumption date for a washer.

The clothes washer case is more straight forward as illustrated in Figure 9.9. The plot shows that regardless of the threshold value, the correct number of SUPs is usually picked. This is because the reference pattern doesn't find any matches within the day except the one at the beginning of the SUP.

9.2 SUP Classification Using DTW and KNN Results

This section discusses the recognition of SUPs and the classification of the detected SUPs into different operation modes. Since we work in two different methods in classifying SUPs, we test each method with its appropriate setup. Then we evaluate the performance of each method and provide evaluation measures to compare the results of the two methods.

9.2.1 Generating Datasets

Using the approach described in Chapter 4, we generated power consumption data for three different households. These datasets are distinct in appliance usage.

We assume that for each appliance, there are three different levels of usage intensity: High, Medium, and Low intensity where High Intensity refers to households who use the Heavy mode the majority of the time, Medium Intensity refers to households that use Medium mode the majority of the time, and Low Intensity refers to households that use Light mode the majority of the time. The generated test dataset setup is shown in Table 9.1.

Household	Intensity	Appliance	Major Mode	Minor Modes
1	**High**	Clothes Washer	Heavy	Medium, Light
		Clothes Dryer	Heavy	Medium, Light
		Dishwasher	Heavy	Medium, Light
2	**Medium**	Clothes Washer	Medium	Heavy, Light
		Clothes Dryer	Medium	Heavy, Light
		Dishwasher	Medium	Heavy, Light
3	**Low**	Clothes Washer	Light	Heavy,Medium
		Clothes Dryer	Light	Heavy,Medium
		Dishwasher	Light	Heavy,Medium

Table 9.1: The setup of the testing datasets.

The simulator discussed in Section 4.2.2 generates daily power consumption data based on the Usage Intensity parameter that determines which operation mode is used mostly over the time. The simulator samples the modes from a normal distribution function. We roughly assumed that $\mu = 50$ and $\sigma = 9$ so that the major operation mode is selected in 68% of the time, while the other two minor modes are selected 16% of the time, each of them.

Each dataset consists of daily power consumption for two thousands days. We assume that there is only one SUP occurrence daily per appliance so that a single turn on time is provided.

9.2.2 Evaluation Metrics

We use four metrics to measure the performance of the classifiers we use. The metrics are the following: Precision, Recall, F1-score and Accuracy [111]. These metrics are defined as follows:

$$
\begin{aligned}
Accuracy &= \frac{TP + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \\
Precision &= \frac{\text{TP}}{\text{TP} + \text{FP}} \\
Recall &= \frac{\text{TP}}{\text{TP} + \text{FN}} \\
F1\ score &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\end{aligned}
$$

where TP is True Positives, TN is True Negatives, FP is False Positives, FN is False Negatives.

9.2.3 SUP Classification Results

Table 9.2 shows the classification accuracy for DTW and KNN applied on the nine data sets described in Section 9.2.1. The table is divided by appliance. In terms of the average accuracy for each appliance using DTW, clothes washer accuracy is around 85%, for the clothes dryer the average accuracy is 88%, and for the dishwasher's accuracy, it is on average 84%. The average accuracy of the DTW for all appliances is 85%. For the KNN algorithm, clothes washer accuracy is around 79%, the clothes dryer is with average accuracy 92%, and for the dishwasher's accuracy, it is 91%. The average accuracy of all appliances for the KNN is 84%. Finally, the confidence interval for the DTW is $[0.8513, 0.8575]$ and for KNN is $[0.8388, 0.8451]$.

Figure 9.10 shows the relationship between usage intensity, operation mode, and the metrics. The chart shows the precision, recall, and F1-score on the y-axis. The x-axis is divided into three lanes representing the usage intensity values (low, medium, high). In each lane, the operation mode is depicted. The chart shows high values of the metric for the major operation mode in each usage intensity lane. e.g, in the low intensity lane, the light mode is the operation mode that is used by the household the majority of the time. Therefore, the three metric values for the light mode are relatively higher than other modes. The heavy operation mode in the high intensity lane behaves in the same way. The medium operation mode in the medium intensity lane show this property in the precision, but generally it has higher performance than other usage intensity.

The performance of the DTW and the KNN algorithms is summarized in Table 9.3. It

Appliance	Usage Intensity	DTW	KNN
Clothes Washer	Low	0.95	0.80
	Medium	0.86	0.80
	High	0.73	0.76
Clothes Dryer	Low	0.85	0.94
	Medium	0.89	0.92
	High	0.89	0.92
Dishwasher	Low	0.76	0.74
	Medium	0.84	0.80
	High	0.92	0.90
Average		**0.85**	**0.84**

Table 9.2: Accuracy for the classification of the 9 test datasets using DTW based and KNN.

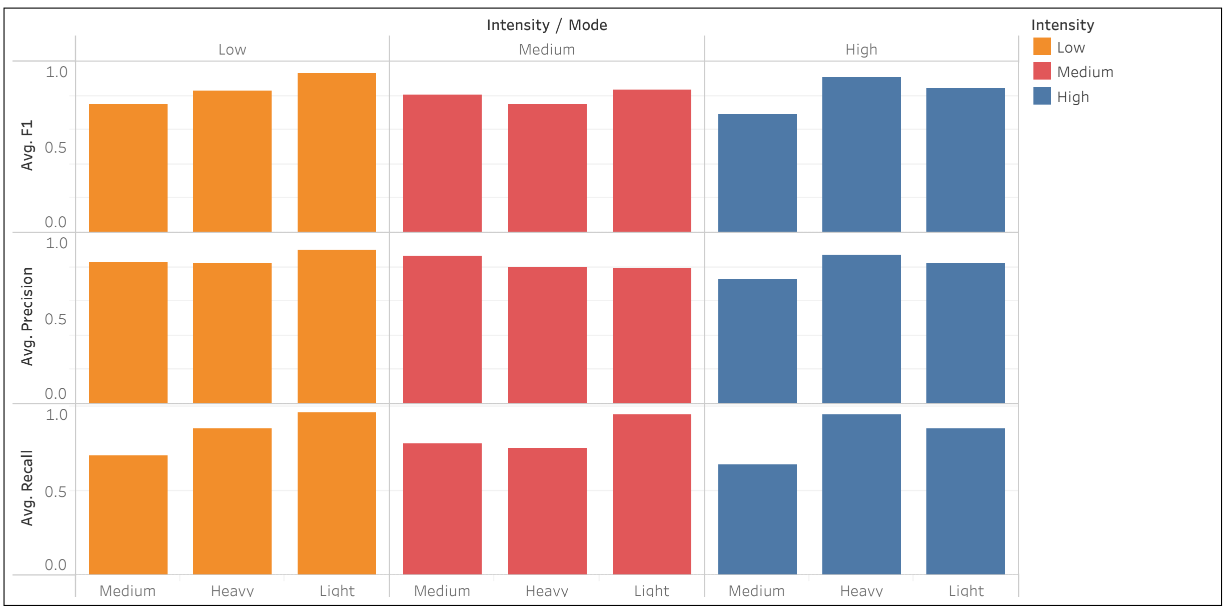


Figure 9.10: The precision, recall, and F1-score for each operation mode in every usage intensity.

shows the average precision, recall, and the F1-score across all the experiments conducted over the datasets. The precision and recall values of the DTW are 81% and 81% respectively. While these values are generally higher for the KNN with 84% and 85%.

The evaluation of the performance of the two algorithms used in terms of operation modes is depicted in Figure 9.11 and Figure 9.12. The charts shows the precision, recall and F1-score

Algorithm	Precision	Recall	F1-score
DTW	0.81	0.81	0.80
KNN	0.84	0.85	0.83

Table 9.3: Average precision, recall, and F1-score values for all datasets using DTW based and KNN.

for each operation mode using box and whiskers. Figure 9.11 shows that for the DTW the metrics values are averaged around 82% for the light and heavy operation modes. Otherwise, the metrics value is approximately 79%. On the other hand, the KNN chart presented in Figure 9.12 shows more balanced results, for the three operation modes the metrics average values are around 82%. The average values for the metrics per operation mode for both algorithms is summarized in Table 9.4.

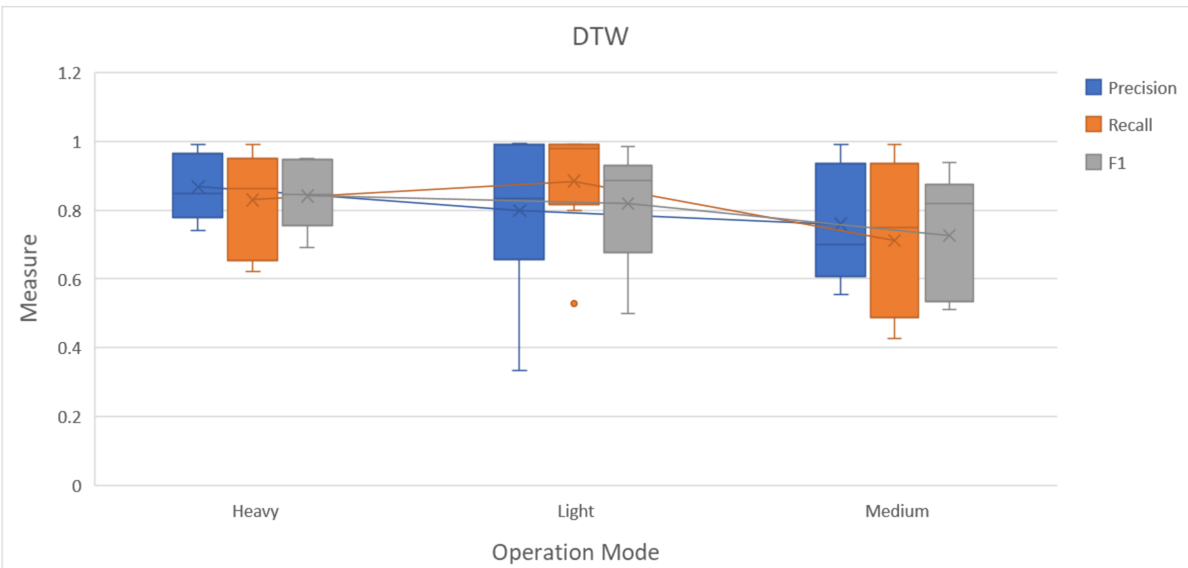


Figure 9.11: The precision, recall and F1-score for each operation mode using DTW.

Operation Mode	Precision	Recall	F1-score
Light	0.83	0.92	0.87
Medium	0.81	0.71	0.75
Heavy	0.83	0.86	0.83

Table 9.4: Average precision, recall, and F1-score values for each operation mode.

The comparison between the performance of DTW and KNN per appliance is illustrated in

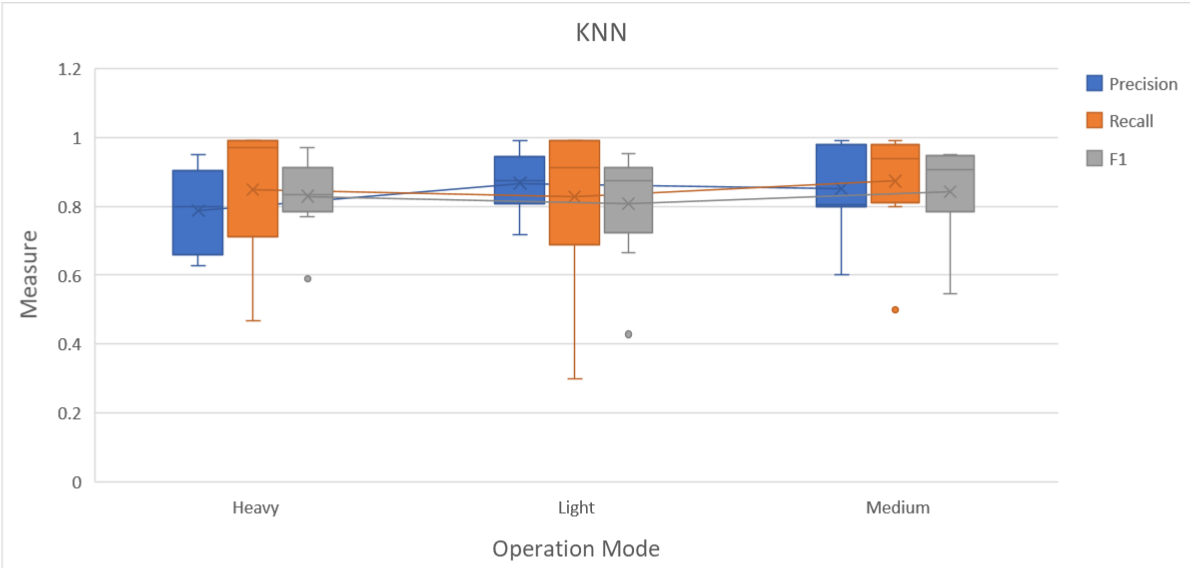


Figure 9.12: The precision, recall and F1-score for each operation mode using KNN.

Figure 9.13. We focus on the precision, recall, and F1-score values for each appliance. DTW has metrics values range between 82% and 83% for the clothes dryer. For the clothes washer DTW scored between 77% and 80%. For the dishwasher, DTW has higher metrics scores with a of values between 78% and 82%. KNN scored higher precision, recall, and F1-score values for the clothes dryer with the range 90% to 91%. For the clothes washer KNN metric values are ranging between 81% and 88%. for the dishwasher, KNN has lower metrics values than other appliances with the range 75% and 78%.

The breakdown of the operation modes performance for each appliance is shown in Figure 9.14. The chart is divided into three main lanes, each lane corresponds to an appliance. Within each lane a comparison between KNN and DTW classification results over the operation modes. Generally, the chart shows higher performance for KNN over DTW. In the clothes washer lane, KNN has higher metrics values for heavy and light modes than DTW. Also, in the clothes dryer lane, KNN has higher performance for heavy and light modes than DTW. Lastly, for the dishwasher, the metrics values of the DTW is higher than the KNN.

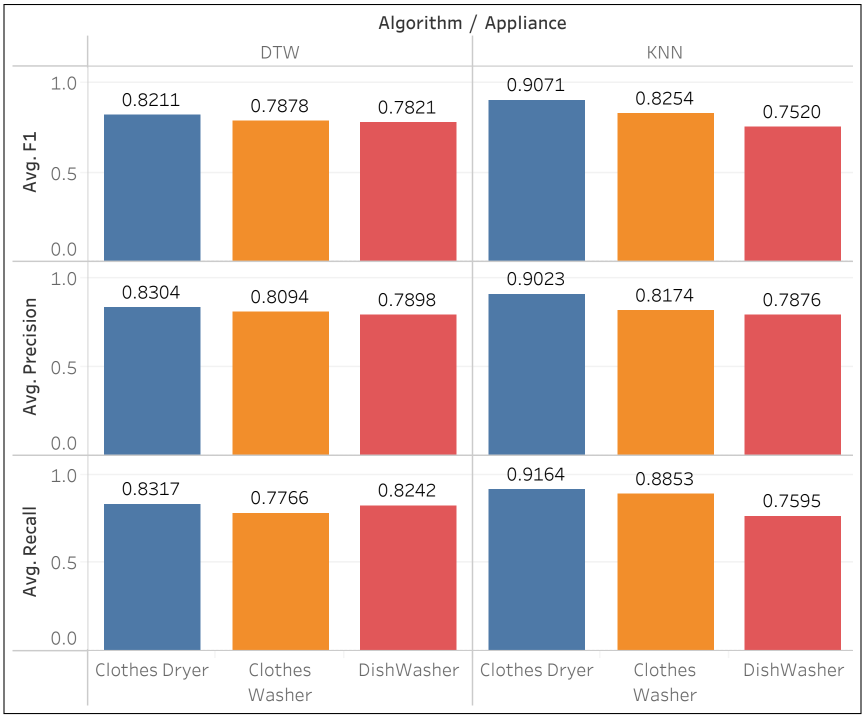


Figure 9.13: Average precision, recall, and F1-score values for each appliance using DTW and KNN.

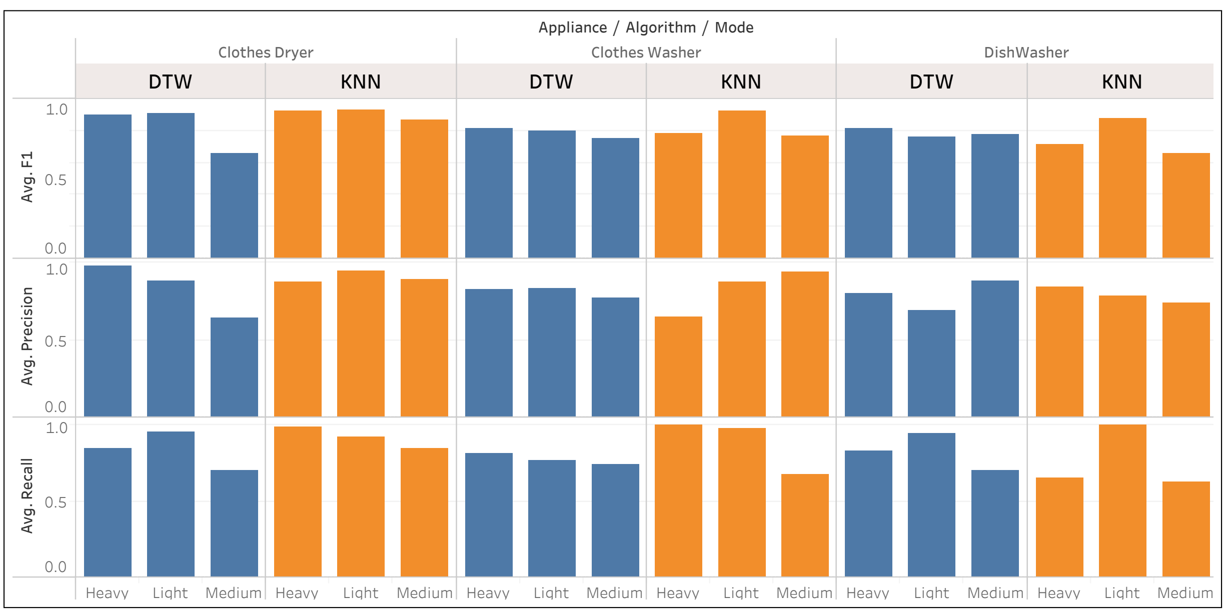


Figure 9.14: The breakdown of the operation modes performance for each appliance using DTW and KNN algorithms.

9.3 Discussion

We discussed the impact of reference pattern size on the number of detected SUPs for each appliance in Section 9.1.2.1. We observed that the detection algorithm worked at its best performance when the the reference pattern size in the first 20 minutes of the SUP. For the dishwasher the best reference pattern size was between 7 to 20 minutes, for the dryer it was 10 to 17 minutes, and for the washer the best performance was when the size was between 3 to 20 minutes. We found that within these ranges of sizes, the detection performed fairly with around 74%, 60%, and 99% successfully detecting one SUP for the dishwasher, dryer, and washer respectively. We also found that the detection performed poorly when the reference pattern was less than the aforementioned ranges because the it is highly likely to detect short reference patterns in SUPs that have many repetitions. However, when the reference pattern size exceeded the ranges mentioned before, the detection poorly performed and didn't detect any SUP.

We then discussed the impact of the reference pattern size on the detected turn on time in Section 9.1.2.2. The average time difference between the actual time and the detected time is 40 samples, 31 samples, and 124 samples for the dishwasher, dryer, and washer respectively. The main contributor to this time difference is the cycle duration variation that changes the start time of a cycle differ by couple of minutes (120 samples). We believe that with this performance, the detection algorithm could be used practically since the time difference is at most 2.8% of the average SUP length.

The impact of the low amplitude canceling coefficient δ on the number of detected SUPs is discussed in Section 9.1.3. We found that for every appliance there is a range of values for δ that gives good performance in detecting the correct number of SUPs. We found that the results align with the expected role of δ for the dishwasher and the dryer. When the value of δ is very close to 100, the number of detected SUPs is zero since the matching between the reference pattern and the actual SUP is not perfect. When the value of δ is relatively small, the number of detected SUPs is high since the threshold is not high enough to cancel out the spikes which report incorrect SUPs. Finally, when the value of δ in the upper eighties and lower nineties, the number of detected SUPs in one since the threshold is placed at the best location. For the clothes washer, changing δ didn't have significant impact on the number of detected SUPs since the shape of the washer SUP doesn't have many repetitions, consequently, the cross correlation produces less number of high spikes. Therefore, increasing or decreasing the value of δ doesn't change the detection of the actual SUP. We believe that these results are fair enough in case of practical use of the model. For each appliance, the value of δ could be tuned in the aforementioned ranges to give the best results.

The results of SUP classification was presented in Section 9.2.3. We found that both DTW and KNN algorithms performed almost the same with an extra 1% for the DTW. The average accuracy for the DTW and KNN are 85% and 84% respectively. However, since the F1-score for the KNN (83%) was higher the the f1-score for the DTW (80%), the performance of the KNN is considered better than the performance of the DTW. In terms of algorithm performance per appliance, KNN generally performed better that DTW for the clothes washer and dryer, while DTW performed better for the dishwasher. As a result, there is a potential of applying this classification practically for some appliances since the results of each algorithm differ from appliance to other.

CHAPTER 10

Conclusion and Future Work

This chapter gives concluding remarks for the research work that has been conducted in this thesis. It also discusses potential future enhancements in this research topic.

10.1 Conclusion

Our work is focused on building a prototype to monitor and analyze residential power consumption for certain appliances in order to give the household tips to lower the daily power consumption. These tips are based on detecting the times when an appliance is activated so the system advises the user to shift the usage to off peak hours. The other way to give tips is to recognize the operation mode that appliances in activated with and advise the user to user lighter modes that consume less power.

We analyzed publicly available disaggregated power consumption dataset [66] by applying statistical analysis to find out statistical distributions about consumption for the households in the dataset. Furthermore we used some of the analysis in the literature [83, 114] to understand operation modes for appliances and extract their characteristics. A simulation engine then processes the statistical models resulted from the data analysis and generates power consumption data to simulate households use their appliances in different operation modes. This simulated data is used to test our models. An SUP detection algorithm is proposed using cross correlation between reference patterns and daily usage data to detect the activation times of appliances. These activation times are used by the recognition algorithms to classify SUPs into their operation modes. We used two algorithms to classify SUPs. Fist, Dynamic Time Warping (DTW) is used to measure the similarity of the detected SUP with reference patterns that represent all different operation modes for the appliance. Second, K-Nearest Neighbors (KNN) is used to classify SUPs. The features used are cycles enclosed by edges that form the SUP which are extracted using Moving Step Test (MST) for edge detection.

10.2 Future Work

It is important to have gold standard datasets in order to evaluate any proposed model. It is crucial to have a real dataset that contains power consumption readings from different types of appliances with different operation modes. This dataset should also consider different manufacturers for the same appliance so that different usage profiles is considered.

Furthermore, building a real application that resembles a MDMS for a hydro company and integrate the work described in this thesis. This integration opens the doors for different opportunities including the following: First, this work provides tips to the consumers based on their own consumption, this facilitates the evaluation of behaviour change in terms of demand response. Second, this work can be used in the gamification applications in a way that serious games can be developed on top of the results that come from the detection and the recognition modules.

Another improvement that could be applied is to enhance the machine learning based classification by using different feature extraction methods such as computing mean, median, standard deviation of the SUPs. Taking the cycles order within SUP can enhance the SUP classification, since we didn't pay attention to cycles order rather than considering the average power value and cycle duration.

More improvements could be applied is to use the enhanced DTW version called *SparseDTW* [4] or the Sakoe-Chiba-Band version proposed in [91] which both enhances the speed of distance computing by limiting the need for large number of recursive calls, or by limiting the need for calculating all the elements of the distance matrix. Using either of these improved DTW versions will provide opportunities to apply this work on online power consumption data streams collected by smart meters and sensors.

Bibliography

[1] Hafiz M Abd-ur Rehman, Fahad A Al-Sulaiman, Aamir Mehmood, Sehar Shakir, and Muhammad Umer. The potential of energy savings and the prospects of cleaner energy production by solar energy integration in the residential buildings of saudi arabia. *Journal of cleaner production*, 183:1122–1130, 2018.

[2] Wokje Abrahamse, Linda Steg, Charles Vlek, and Talib Rothengatter. A review of intervention studies aimed at household energy conservation. *Journal of environmental psychology*, 25(3):273–291, 2005.

[3] accord. accord-framework.net, 2019. [Online; accessed 16-April-2019].

[4] Ghazi Al-Naymat, Sanjay Chawla, and Javid Taheri. Sparsedtw: A novel approach to speed up dynamic time warping. *Conferences in Research and Practice in Information Technology Series*, 101, 01 2012.

[5] José M Alcalá, Jesús Ureña, and Álvaro Hernández. Event-based detector for non-intrusive load monitoring based on the hilbert transform. In *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, pages 1–4. IEEE, 2014.

[6] Jose Manuel Alcala. NILM for Activities of Daily Living. 2016.

[7] Kyle D Anderson, Mario E Bergés, Adrian Ocneanu, Diego Benitez, and José MF Moura. Event detection for non intrusive load monitoring. In *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society*, pages 3312–3317. IEEE, 2012.

[8] Mark Apperley and Jishaal Kalyan. A Mobile Personal Residential Electricity Dashboard. In *2015 19th International Conference on Information Visualisation*, pages 195–199. IEEE, jul 2015.

[9] Barbara Baker, Ioulia Sklokin, Len Coad, and Todd Crawford. Canada's electricity infrastructure: Building a case for investment. *Conference Board of Canada*, 2011.

[10] Stuart N Baker and George L Gerstein. Determination of response latency and its application to normalization of cross-correlation measures. *Neural Computation*, 13(6):1351–1377, 2001.

[11] VSK Murthy Balijepalli, Vedanta Pradhan, SA Khaparde, and RM Shereef. Review of demand response under smart grid paradigm. In *ISGT2011-India*, pages 236–243. IEEE, 2011.

[12] Gouri R. Barai, Sridhar Krishnan, and Bala Venkatesh. Smart metering and functionalities of smart meters in smart grid - A review. *2015 IEEE Electrical Power and Energy Conference: Smarter Resilient Power Systems, EPEC 2015*, pages 138–145, 2016.

[13] Michael Baranski and Jürgen Voss. Detecting patterns of appliances from total load data using a dynamic programming approach. In *Fourth IEEE International Conference on Data Mining (ICDM'04)*, pages 327–330. IEEE, 2004.

[14] Karim Said Barsim, Roman Streubel, and Bin Yang. Unsupervised non-intrusive load monitoring of residential appliances.

[15] Karim Said Barsim, Roman Streubel, and Bin 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.

[16] Mario Berges, Ethan Goldman, H Scott Matthews, Lucio Soibelman, and Kyle Anderson. User-centered nonintrusive electricity load monitoring for residential buildings. *Journal of computing in civil engineering*, 25(6):471–480, 2011.

[17] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994.

[18] G. Biau and L. Devroye. *Lectures on the Nearest Neighbor Method*. Springer Series in the Data Sciences. Springer International Publishing, 2015.

[19] Natural Resources Canada. Electricity facts, 2018. [Online; accessed 27-June-2018].

[20] Wikipedia contributors. Asp.net mvc — wikipedia, the free encyclopedia, 2018. [Online; accessed 14-April-2019].

[21] Mark Costanzo, Dane Archer, Elliot Aronson, and Thomas Pettigrew. Energy conservation behavior: The difficult path from information to action. *American psychologist*, 41(5):521, 1986.

[22] Peter M. Cox, Richard A. Betts, Chris D. Jones, Steven A. Spall, and Ian J. Totterdell. Erratum: Acceleration of global warming due to carbon-cycle feedbacks in a coupled climate model. *Nature*, 408(6809):184–187, nov 2000.

[23] Sarah Darby et al. The effectiveness of feedback on energy consumption. *A Review for DEFRA of the Literature on Metering, Billing and direct Displays*, 486(2006):26, 2006.

[24] Dent. Dent powerscout 24 multi-circuit power submeter, 2019. [Online; accessed 14-January-2019].

[25] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. From game design elements to gamefulness: Defining gamification. *Proceedings of the 15th International Academic MindTrek Conference on Envisioning Future Media Environments - MindTrek '11*, pages 9–11, 2011.

[26] J. Dickert and P. Schegner. A time series probabilistic synthetic load curve model for residential customers. *2011 IEEE PES Trondheim PowerTech: The Power of Technology for a Sustainable Society, POWERTECH 2011*, (July), 2011.

[27] Energy Education. Energy vs power, 2019. [Online; accessed 15-January-2019].

[28] EPA.

[29] Xiaodong Fan, Bo Qiu, Yuanyuan Liu, Haijing Zhu, and Bochong Han. Energy Visualization for Smart Home. *Energy Procedia*, 105:2545–2548, may 2017.

[30] Linda Farinaccio and Radu Zmeureanu. Using a pattern recognition approach to disaggregate the total electricity consumption in a house into the major end-uses. *Energy and Buildings*, 30(3):245–259, 1999.

[31] Corinna Fischer. Feedback on household electricity consumption: a tool for saving energy? *Energy Efficiency*, 1(1):79–104, feb 2008.

[32] David R Flatla, Carl Gutwin, Lennart E Nacke, Scott Bateman, and Regan L Mandryk. Calibration games: making calibration tasks enjoyable by adding motivating game elements. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 403–412. ACM, 2011.

[33] Brian J Fogg. Persuasive technology: using computers to ch@articleliang2010load, title=Load signature study—Part II: Disaggregation framework, simulation, and applications, author=Liang, Jian and Ng, Simon KK and Kendall, Gail and Cheng, John WM, journal=IEEE Transactions on Power Delivery, volume=25, number=2, pages=561–569, year=2010, publisher=IEEE e what we think and do. *Ubiquity*, 2002(December):5, 2002.

[34] Derek Foster, Mark Blythe, Paul Cairns, and Shaun Lawson. Competitive carbon counting: can social networking sites Make saving energy more enjoyable? *Energy*, (May 2014):4039–4044, 2010.

[35] Roland Fried. On the robust detection of edges in time series filtering. *Computational Statistics & Data Analysis*, 52(2):1063–1074, 2007.

[36] Jon Froehlich. Promoting Energy Efficient Behaviors in the Home through Feedback: The Role of HumanComputer Interaction. *Biometrika*, 73(1):13–22, 1986.

[37] Jon Froehlich, Leah Findlater, and James Landay. The design of eco-feedback technology. *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*, page 1999, 2010.

[38] Jon Froehlich, Eric Larson, Sidhant Gupta, Gabe Cohn, Matthew S Reynolds, and Shwetak N Patel. Disaggregated End-Use Energy Sensing for the Smart Grid The Value of Disaggregated Data. *Energy*, pages 1–17, 2011.

[39] Giacomo Ghidini, Sajal K. Das, and Vipul Gupta. Fuseviz: A framework for web-based data fusion and visualization in smart environments. *MASS 2012 - 9th IEEE International Conference on Mobile Ad-Hoc and Sensor Systems*, pages 468–472, 2012.

[40] S.G. Gindikin and L.R. Volevich. *Distributions and Convolution Equations*. Gordon and Breach Science Publishers, 1992.

[41] Graeme JN Gooday and Graeme JN Gooday. *The morals of measurement: Accuracy, irony, and trust in late Victorian electrical practice*. Cambridge University Press, 2004.

[42] Catherine Grevet, Jennifer Mankoff, and Scott D. Anderson. Design and evaluation of a social visualization aimed at encouraging sustainable behavior. In *Proceedings of the Annual Hawaii International Conference on System Sciences*, pages 1–8. IEEE, 2010.

[43] Nadir Gulzar. Fast track to struts: what it does and how. *Retrieved on March*, 30:2010, 2002.

[44] Anton Gustafsson, Magnus Bång, and Mattias Svahn. Power explorer: a casual game style for encouraging long term behavior change among teenagers. *ACE '09 Proceedings of the International Conference on Advances in Computer Entertainment Technology*, pages 182–189, 2009.

[45] George William Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891, 1992.

[46] Taha Hassan, Fahad Javed, and Naveed Arshad. An empirical investigation of vi trajectory based load signatures for non-intrusive load monitoring. *IEEE Transactions on Smart Grid*, 5(2):870–878, 2014.

[47] Tiffany Grace Holmes. Eco-visualization. *Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition - C&C '07*, page 153, 2007.

[48] Tiffany Grace Holmes. Eco-visualization: combining art and technology to reduce energy consumption. *Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition*, pages 153–162, 2007.

[49] Gerwin Hoogsteen, Albert Molderink, Johann L. Hurink, and Gerard J.M. Smit. Generation of flexible domestic load profiles to evaluate Demand Side Management approaches. *2016 IEEE International Energy Conference, ENERGYCON 2016*, 2016.

[50] HydroOne. Time-of-use (TOU), 2019. [Online; accessed 27-August-2019].

[51] FO Isinkaye, YO Folajimi, and BA Ojokoh. Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, 16(3):261–273, 2015.

[52] Itron. Itron openway® centron, 2019. [Online; accessed 14-January-2019].

[53] Jun Jun Jia, Jin Hua Xu, and Ying Fan. Public acceptance of household energy-saving measures in Beijing: Heterogeneous preferences and policy implications. *Energy Policy*, 113(October 2017):487–499, 2018.

[54] Yuanwei Jin, Eniye Tebekaemi, Mario Berges, and Lucio Soibelman. A time-frequency approach for event detection in non-intrusive load monitoring. In *Signal Processing, Sensor Fusion, and Target Recognition XX*, volume 8050, page 80501U. International Society for Optics and Photonics, 2011.

[55] SeungJun Kang and Ji Won Yoon. Classification of home appliance by using probabilistic knn with sensor data. In *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 1–5. IEEE, 2016.

[56] Willett Kempton and Laura Montgomery. Folk quantification of energy. *Energy*, 7(10):817–827, 1982.

[57] Weicong Kong, Z. Y. Dong, Guo Chen, and Youwei Jia. A rule based domestic load profile generator for future smart grid. *2014 Australasian Universities Power Engineering Conference, AUPEC 2014 - Proceedings*, (October):1–5, 2014.

[58] Christopher Laughman, Kwangduk Lee, Robert Cox, Steven Shaw, Steven Leeb, Les Norford, and Peter Armstrong. Power signature analysis. *IEEE power and energy magazine*, 99(2):56–63, 2003.

[59] Steven B Leeb, Steven R Shaw, and James L Kirtley. Transient event detection in spectral envelope estimates for nonintrusive load monitoring. *IEEE Transactions on Power Delivery*, 10(3):1200–1210, 1995.

[60] Jian Liang, Simon KK Ng, Gail Kendall, and John WM Cheng. Load signature study—part ii: Disaggregation framework, simulation, and applications. *IEEE Transactions on Power Delivery*, 25(2):561–569, 2010.

[61] Jing Liao, Georgia Elafoudi, Lina Stankovic, and Vladimir Stankovic. Power disaggregation for low-sampling rate data. In *2nd International Non-intrusive Appliance Load Monitoring Workshop, Austin, TX*, 2014.

[62] Bo Liu, Wenpeng Luan, and Yixin Yu. Dynamic time warping based non-intrusive load transient identification. *Applied energy*, 195:634–645, 2017.

[63] Sunpower Electronics LTD. What is inrush current?, 2014. [Online; accessed 20-January-2019].

[64] Dong Luo, Leslie K Norford, Steven R Shaw, and Steven B Leeb. Monitoring hvac equipment electrical loads from a centralized location–methods and field test results/discussion. *ASHRAE Transactions*, 108:841, 2002.

[65] David JC MacKay and David JC Mac Kay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[66] Stephen Makonin. RAE: The rainforest automation energy dataset, 2018. [Online; accessed 14-January-2019].

[67] Stephen Makonin, Z. Jane Wang, and Chris Tumpach. RAE: The Rainforest Automation Energy Dataset for Smart Grid Meter Data Analysis. pages 1–9, 2017.

[68] B Mettler-Meibom and B Wichmann. The influence of information and attitudes toward energy conservation on behavior, 1982.

[69] Microsoft. Microsoft public license (ms-pl), 2018. [Online; accessed 14-April-2019].

[70] Microsoft. Microsoft sql server, 2018. [Online; accessed 14-April-2019].

[71] Microsoft. Entity framework, 2019. [Online; accessed 4-July-2019].

[72] Luca Morganti, Federica Pallavicini, Elena Cadel, Antonio Candelieri, Francesco Archetti, and Fabrizia Mantovani. Gaming for Earth: Serious games and gamification to engage consumers in pro-environmental behaviours for energy efficiency. *Energy Research and Social Science*, 29(September):95–102, 2017.

[73] Abdullah Mueen and Eamonn Keogh. Extracting Optimal Performance from Dynamic Time Warping. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, pages 2129–2130, 2016.

[74] Fumihiko Nakazawa, Hiromitsu Soneda, Osamu Tsuboi, Akinori Iwakawa, Masahiko Murakami, Masahiro Matsuda, and Naoyuki Nagao. Smart power strip network and visualization server to motivate energy conservation in office. *IEEE International Conference on Industrial Informatics (INDIN)*, pages 352–357, 2011.

[75] Vit Niennattrakul and Chotirat Ann Ratanamahatana. On clustering multimedia time series data using k-means and dynamic time warping. In *2007 International Conference on Multimedia and Ubiquitous Engineering (MUE'07)*, pages 733–738. IEEE, 2007.

[76] Ontario Energy Board (OEB). Managing costs with time-of-use rates, 2018. [Online; accessed 11-April-2019].

[77] T G Papaioannou, D Kotsopoulos, C Bardaki, and S Lounis. " IoT -Enabled Gamification for Energy Conservation in Public Buildings ". 2017.

[78] Oliver Parson, Siddhartha Ghosh, Mark Weal, and Alex Rogers. Non-intrusive load monitoring using prior models of general appliance types. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[79] Lucas Pereira, Filipe Quintal, Rodolfo Gonçalves, and Nuno Jardim Nunes. Sustdata: A public dataset for ict4s electric energy research. In *ICT for Sustainability 2014 (ICT4S-14)*. Atlantis Press, 2014.

[80] Petromil Petkov, Felix Köbler, Marcus Foth, and Helmut Krcmar. Motivating Domestic Energy Conservation through Comparative, Community-Based Feedback in Mobile and Social Media.

[81] N Pflugradt, J Teuscher, B Platzer, and W Schufft. Analysing low-voltage grids using a behaviour based load profile generator. In *International Conference on Renewable Energies and Power Quality*, volume 11, page 5, 2013.

[82] PhysLink. What is the difference between resistance and impedance?, 2018. [Online; accessed 3-February-2019].

[83] Manisa Pipattanasomporn, Murat Kuzlu, Saifur Rahman, and Yonael Teklu. Load Profiles of Selected Major Household Appliances and Their Demand Response Opportunities. *IEEE Transactions on Smart Grid*, 5(2):742–750, mar 2014.

[84] A Prudenzi. A neuron nets based procedure for identifying domestic appliances pattern-of-use from energy recordings at meter panel. In *2002 IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No. 02CH37309)*, volume 2, pages 941–946. IEEE, 2012.

[85] Psychology Encyclopedia. Normal distribution, 2019. [Online; accessed 24-June-2019].

[86] Rainforest. Rainforest emu-2^{TM} energy monitoring unit, 2019. [Online; accessed 14-January-2019].

[87] RaspberryPi. Raspberry pi, 2019. [Online; accessed 14-January-2019].

[88] Zachary Remscrim, James Paris, Steven B Leeb, Steven R Shaw, Sabrina Neuman, Christopher Schantz, Sean Muller, and Sarah Page. Fpga-based spectral envelope preprocessor for power monitoring and control. In *2010 Twenty-Fifth Annual IEEE Applied Power Electronics Conference and Exposition (APEC)*, pages 2194–2201. IEEE, 2010.

[89] Ian Richardson, Murray Thomson, David Infield, and Alice Delahunty. Domestic lighting: A high-resolution energy demand model. *Energy and Buildings*, 41(7):781–789, 2009.

[90] Michael Ro, Markus Brauer, Kathy Kuntz, Raj Shukla, and Ingo Bensch. Making Cool Choices for sustainability: Testing the effectiveness of a game-based approach to promoting pro-environmental behaviors. *Journal of Environmental Psychology*, 53:20–30, 2017.

[91] Hiroaki Sakoe and Seibi Chiba. Computer for calculating the similarity between patterns and pattern recognition system comprising the similarity computer, June 11 1974. US Patent 3,816,722.

[92] Geertje Schuitema, Lisa Ryan, and Claudia Aravena. The Consumer's Role in Flexible Energy Systems: An Interdisciplinary Approach to Changing Consumers' Behavior. *IEEE Power and Energy Magazine*, 15(1):53–60, 2017.

[93] Tony C Shan and Winnie W Hua. Solution architecture for n-tier applications. In *2006 IEEE International Conference on Services Computing (SCC'06)*, pages 349–356. IEEE, 2006.

[94] Steven R Shaw and Christopher R Laughman. A kalman-filter spectral envelope preprocessor. *IEEE Transactions on Instrumentation and Measurement*, 56(5):2010–2017, 2007.

[95] Steven R Shaw, Steven B Leeb, Leslie K Norford, and Robert W Cox. Nonintrusive load monitoring and diagnostics in power systems. *IEEE Transactions on Instrumentation and Measurement*, 57(7):1445–1454, 2008.

[96] Omid Ameri Sianaki and Mohammad AS Masoum. A fuzzy topsis approach for home energy management in smart grid with considering householders' preferences. In *2013 IEEE PES Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–6. IEEE, 2013.

[97] Carlos F. S. Souza, Carlos E. P. Pantoja, and Francisco C. M. Souza. Verificação De Assinaturas Offline Utilizando Dynamic Time Warping. pages 1–5, 2016.

[98] statista. Global electricity prices in 2018, by select country (in u.s. dollars per kilowatt hour), 2018. [Online; accessed 10-April-2019].

[99] Statista.com. Number of monthly active facebook users worldwide as of 1st quarter 2018, 2018. [Online; accessed 4-July-2018].

[100] NETL Modern Grid Strategy. Advanced metering infrastructure. *US Department of Energy Office of Electricity and Energy Reliability*, 2008.

[101] Kaveri Subrahmanyam, Stephanie M Reich, Natalia Waechter, and Guadalupe Espinoza. Online and offline social networks: Use of social networking sites by emerging adults. *Journal of applied developmental psychology*, 29(6):420–433, 2008.

[102] Techtarget. entity relationship diagram (erd)r, 2014. [Online; accessed 14-April-2019].

[103] Huijuan Wang and Wenrong Yang. An iterative load disaggregation approach based on appliance consumption pattern. *Applied Sciences*, 8(4):542, 2018.

[104] Xiaoyue Wang, Abdullah Mueen, Hui Ding, Goce Trajcevski, Peter Scheuermann, and Eamonn Keogh. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 26(2):275–309, 2013.

[105] Markus Weiss, Adrian Helfenstein, Friedemann Mattern, and Thorsten Staake. Leveraging smart meter data to recognize home appliances. In *2012 IEEE International Conference on Pervasive Computing and Communications*, pages 190–197. IEEE, 2012.

[106] Barry Wellman, Anabel Quan Haase, James Witte, and Keith Hampton. Does the internet increase, decrease, or supplement social capital? social networks, participation, and community commitment. *American behavioral scientist*, 45(3):436–455, 2001.

[107] Devon Wemyss, Roberta Castri, Vanessa De Luca, Francesca Cellina, Evelyn Lobsiger-kägi, Pamela Galbani Bianchi, Christian Hertach, and Vicente Carabias. Keeping Up With the Joneses : Examining Community-Level Collaborative and Competitive Game Mechanics To Enhance Household Electricity-Saving Behaviour. *4th European Conference on Behaviour and Energy Efficiency (Behave 2016)*, (September):8–9, 2016.

[108] IEA WEO. World energy outlook 2004, 2005.

[109] Wikipedia contributors. Dynamic time warping — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Dynamic_time_warping&oldid=895166895`, 2019. [Online; accessed 16-May-2019].

[110] Wikipedia contributors. Median filter — Wikipedia, the free encyclopedia, 2019. [Online; accessed 17-April-2019].

[111] Wikipedia contributors. Precision and recall — Wikipedia, the free encyclopedia, 2019. [Online; accessed 18-June-2019].

[112] Richard A Winett, Michael S Neale, and H Cannon Grier. Effects of self-monitoring and feedback on residential electricity consumption. *Journal of Applied Behavior Analysis*, 12(2):173–184, 1979.

[113] worldbank. Electricity production from hydroelectric sources ([Online; accessed 10-April-2019].

[114] Runming Yao and Koen Steemers. A method of formulating energy load profile for domestic buildings in the UK. *Energy and Buildings*, 37(6):663–671, 2005.

[115] Thierry Zufferey, Damiano Toffanin, Diren Toprak, Andreas Ulbig, and Gabriela Hug. Generating stochastic residential load profiles from smart meter data for an optimal power matching at an aggregate level. In *2018 Power Systems Computation Conference (PSCC)*, pages 1–7. IEEE, 2018.

APPENDIX A

Performance Evaluation For DTW and KNN Classification

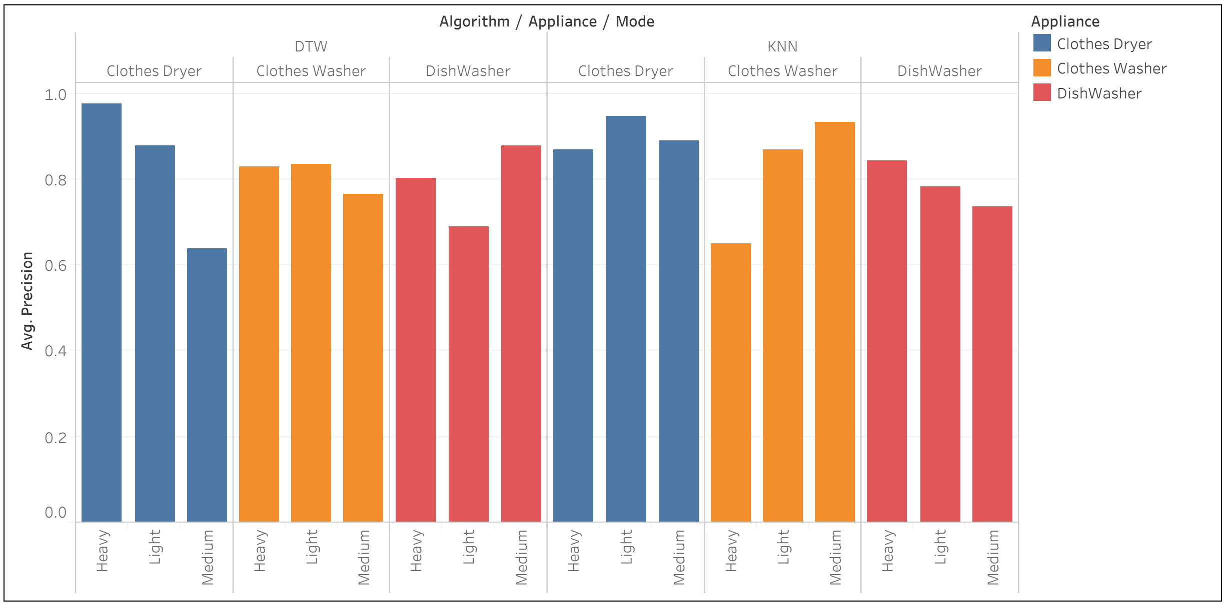


Figure A.1: Precision comparison between DTW and KNN in terms of each operation mode within appliances.

Algorithm	Measure	Class Light	Class Medium	Class Heavy
DTW	Precision	0.9925	0.9	0.8055
	Recall	0.9779	0.7941	0.9666
KNN	Precision	0.8125	0.98	0.6667
	Recall	0.99	0.71	0.981

Table A.1: Precision and recall percentage for 3-classes classification using DTW and KNN for a clothes washer that belongs to low-intensity consumption household.

Algorithm	Measure	Class Light	Class Medium	Class Heavy
DTW	Precision	0.989	0.8391	0.8484
	Recall	0.8	0.96	0.6222
KNN	Precision	0.9166	0.991	0.65
	Recall	0.9825	0.5	0.992

Table A.2: Precision and recall percentage for 3-classes classification using DTW and KNN for a clothes washer that belongs to medium-intensity consumption household.

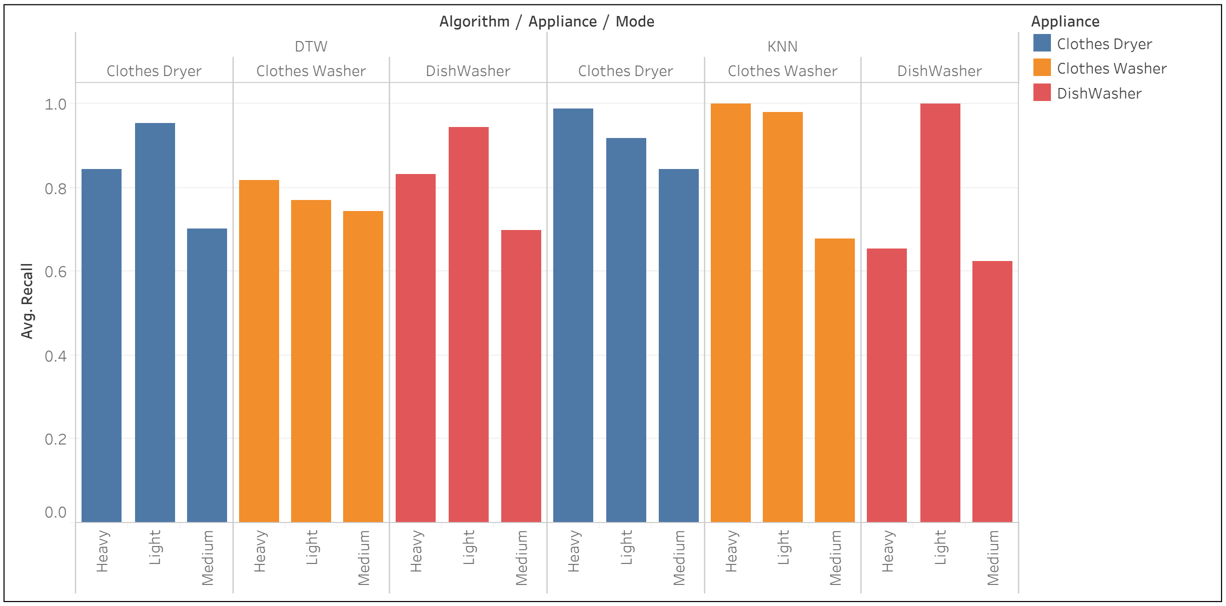


Figure A.2: Recall comparison between DTW and KNN in terms of each operation mode within appliances.

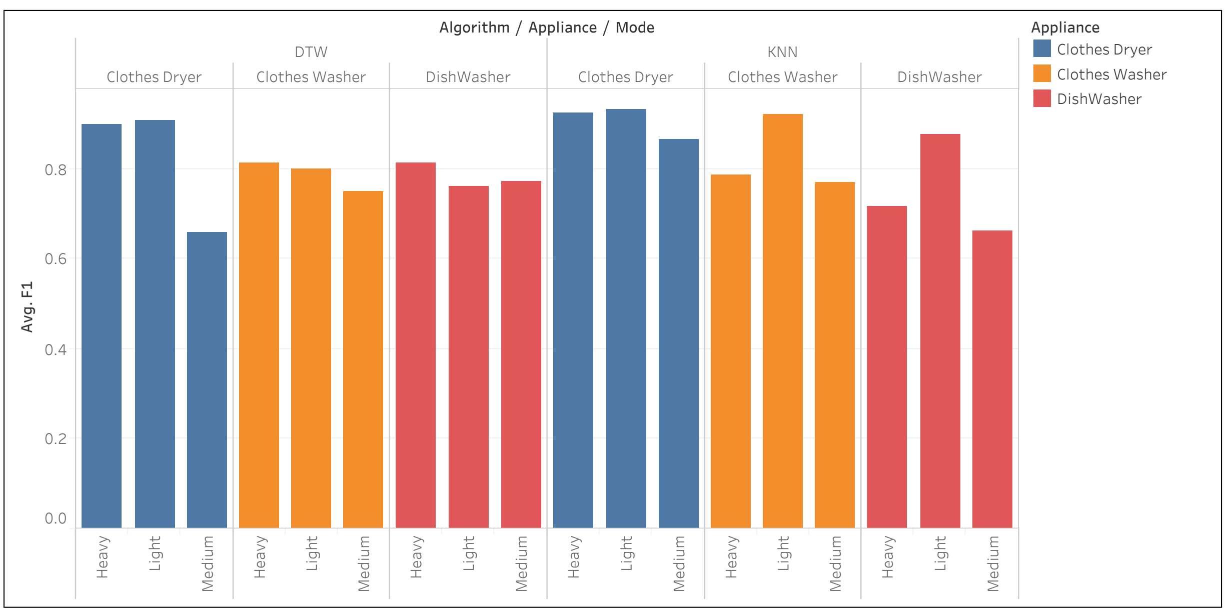


Figure A.3: F1-score comparison between DTW and KNN in terms of each operation mode within appliances.

Algorithm	Measure	Class Light	Class Medium	Class Heavy
DTW	Precision	0.5142	0.5556	0.8294
	Recall	0.5294	0.4762	0.8629
KNN	Precision	0.8823	0.8	0.6285
	Recall	0.9375	0.82	0.9895

Table A.3: Precision and recall percentage for 3-classes classification using DTW and KNN for a clothes washer that belongs to high-intensity consumption household.

Algorithm	Measure	Class Light	Class Medium	Class Heavy
DTW	Precision	0.9917	0.7	0.9896
	Recall	0.857142857142857		0.6667
KNN	Precision	0.9687	0.9	0.875
	Recall	0.9687	0.8181	0.9901

Table A.4: Precision and recall percentage for 3-classes classification using DTW and KNN for a clothes dryer that belongs to low-intensity consumption household.

Algorithm	Measure	Class Light	Class Medium	Class Heavy
DTW	Precision	0.8	0.6121	0.9655
	Recall	0.99	0.5	0.9333
KNN	Precision	0.875	0.9687	0.8
	Recall	0.875	0.9117	0.9872

Table A.5: Precision and recall percentage for 3-classes classification using DTW and KNN for a clothes dryer that belongs to medium-intensity consumption household.

Algorithm	Measure	Class Light	Class Medium	Class Heavy
DTW	Precision	0.8333	0.6	0.9629
	Recall	0.9983	0.6	0.9285
KNN	Precision	0.9869	0.8	0.9333
	Recall	0.9090	0.8	0.9655

Table A.6: Precision and recall percentage for 3-classes classification using DTW and KNN for a clothes dryer that belongs to high-intensity consumption household.

Algorithm	Measure	Class Light	Class Medium	Class Heavy
DTW	Precision	0.9	0.6666	0.7419
	Recall	0.9904	0.4285	0.8518
KNN	Precision	0.7179	0.8	0.8333
	Recall	0.9798	0.4667	0.7142

Table A.7: Precision and recall percentage for 3-classes classification using DTW and KNN for a dishwasher that belongs to low-intensity consumption household.

Algorithm	Measure	Class Light	Class Medium	Class Heavy
DTW	Precision	0.3333	0.9687	0.75
	Recall	0.9931	0.9117	0.6428
KNN	Precision	0.8	0.8056	0.75
	Recall	0.9982	0.9062	0.3

Table A.8: Precision and recall percentage for 3-classes classification using DTW and KNN for a dishwasher that belongs to medium-intensity consumption household.

Algorithm	Measure	Class Light	Class Medium	Class Heavy
DTW	Precision	0.8333	0.9893	0.9142
	Recall	0.8333	0.75	0.9951
KNN	Precision	0.8333	0.6	0.9487
	Recall	0.9930	0.5	0.9487

Table A.9: Precision and recall percentage for 3-classes classification using DTW and KNN for a dishwasher that belongs to high-intensity consumption household.

Curriculum Vitae

Name:	**Abdelkareem Jaradat**
Education and Degrees:	Computer Science (MSc) University Of Western Ontario (UWO), 2019 London, Canada
	Computer Engineering (BSc) Jordan University of Science and Technology (JUST), 2009 Irbid, Jordan
Honours and Awards:	Western Graduate Research Scholarship (WGRS) 2017-2019
	President's list of honors / Dean's list of honors 2005-2008
	Ministry of Higher Education Scholarship (MHES) 2006-2008
	King AbdullahII' Fund for Development (KAFD) 2006-2008
Related Work Experience:	Graduate Teaching and Research Assistant The University of Western Ontario 2017 - 2019
	Senior .Net Web Developer / Business Systems Analyst EJADA Systems 2015 - 2017
	Full Stack Developer Jordan University of Science and Technology (JUST) 2010 - 2015
	Software Development Engineer Systems and Electronic Development Co (SEDCO) 2009-2010