CONTRIBUTIONS AND DEVELOPMENTS ON NONINTRUSIVE LOAD MONITORING

Juliano Freitas Caldeira

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Elétrica.

Orientador: Mauricio Aredes

Rio de Janeiro
Março de 2018

CONTRIBUTIONS AND DEVELOPMENTS ON NONINTRUSIVE LOAD MONITORING

Juliano Freitas Caldeira

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

_____

Prof. Mauricio Aredes, Dr.-Ing.

_____

Prof. Adilson Elias Xavier, D.Sc.

_____

Prof. Eduardo Antônio Barros da Silva, Ph.D.

_____

Prof. José Ferreira de Rezende, Dr.

_____

Prof. Lisandro Lovisolo, D.Sc.

RIO DE JANEIRO, RJ - BRASIL
MARÇO DE 2018

Caldeira, Juliano Freitas

Contributions And Developments On Nonintrusive Load Monitoring/ Juliano Freitas Caldeira. – Rio de Janeiro: UFRJ/COPPE, 2018.

XVII, 201 p. il.; 29,7 cm.

Orientador: Mauricio Aredes

Tese (doutorado) – UFRJ/ COPPE/ Programa de Engenharia Elétrica, 2018.

Referências Bibliográficas: p. 127-138.

1. Machine Learning. 2. Nonintrusive Load Monitoring. 3. Smart Grid. I. Aredes, Mauricio. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

Agradecimentos

"The good thing about science is that it's true whether or not you believe in it".

Neil deGrasse Tyson

Inicialmente agradeço aos meus pais, Viviane e Mauricio, por todo o suporte e dedicação e confiança que sempre conferiram a mim. Agradeço também ao meu professor e orientador Mauricio Aredes por sempre me motivar, aconselhar e ensinar, principalmente, a seguir minha intuição em todos os momentos. Obrigado também a minha filha, Rafaela, e minha namorada, Bruna, pela paciência, compreensão, dedicação e carinho comigo durante a execução deste trabalho.

Agradeço aos amigos queridos que me ajudaram nesta difícil tarefa de escrever sobre um tema inovador e complexo. Jorge Caicedo, pela enorme amizade, genialidade, e excelente trilha sonora. Guilherme Lello, pelos valiosos conselhos, conversas, pela serenidade e irmandade, na gestão do tempo, motivação e companheirismo, além da enorme amizade. Sem a ajuda de vocês este trabalho não teria sido possível.

Também agradeço a todos os amigos do LEMT, em especial aos colegas Bruno França e André Castro, pela incrível troca de conhecimento. Aos meus alunos e ex-alunos que executaram trabalhos fabulosos durante este período, sempre disponíveis para ajudar e trocar ideias singulares sobre este projeto, incluindo o brilhante Pedro Nascimento. Também minha eterna gratidão ao Prof. Carlos José Ribas D´Avila (Casé), por ter participado ativamente da minha formação como engenheiro.

Por fim agradeço aos professores que participaram sempre positivamente de alguma forma da minha carreira acadêmica e/ou profissional: Professores Adilson Xavier, Eduardo Silva, José Rezende meu muito obrigado por tudo. Aos professores Saifur Rahman e Mario Bergés por toda cordialidade e disponibilidade. Meus sentimentos aos que se foram durante este período: Prof. Sergio Barbosa Villas-Boas e Prof. Mauros Campello Queiroz. Vocês foram exemplos de excelência. Muito obrigado por tudo que fizeram por nós da UFRJ.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

CONTRIBUIÇÕES E DESENVOLVIMENTOS NO MONITORAMENTO NÃO INTRUSIVO DE CARGAS

Juliano Freitas Caldeira

Março/2018

Orientador: Mauricio Aredes

Programa: Engenharia Elétrica

Eficiência energética é um assunto essencial na agenda mundial. No Brasil, o desperdício de energia no setor residencial é estimado em 15%. Estudos indicaram que maiores ganhos em eficiência são conseguidos quando o usuário recebe as informações de consumo detalhadas por cada aparelho, provocando mudanças comportamentais e incentivando os consumidores na conservação de energia.

Monitoramento não intrusivo de cargas (NILM da sigla em inglês) é um termo relativamente novo. A sua finalidade é inferir o consumo de um ambiente até observar os consumos individualizados de cada equipamento utilizando-se de apenas um único ponto de medição. Métodos sofisticados têm sido propostos para inferir quando os aparelhos são ligados e desligados em um ambiente.

Dentro deste contexto, este trabalho apresenta uma metodologia para a definição de um conjunto mínimo de características elétricas e sua taxa de extração que reduz a quantidade de dados a serem transmitidos e armazenados em servidores de processamento de dados, preservando níveis equivalentes de acurácia. São utilizadas diferentes técnicas de aprendizado de máquina visando à caracterização e solução do problema.

Como adendo ao trabalho, apresenta-se um banco de dados de eletrodomésticos brasileiros, com amostras de equipamentos nacionais para desenvolvimentos futuros em NILM, além de um medidor inteligente de baixo custo para desagregação de cargas, visando tornar o consumo de energia mais sustentável.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

CONTRIBUTIONS AND DEVELOPMENTS ON NONINTRUSIVE LOAD MONITORING

Juliano Freitas Caldeira

March/2018

Advisor: Mauricio Aredes

Department: Electrical Engineering

Energy efficiency is a key subject in our present world agenda, not only because of greenhouse gas emissions, which contribute to global warming, but also because of possible supply interruptions. In Brazil, energy wastage in the residential market is estimated to be around 15%. Previous studies have indicated that the most savings were achieved with specific appliance, electricity consumption feedback, which caused behavioral changes and encouraged consumers to pursue energy conservation.

Nonintrusive Load Monitoring (NILM) is a relatively new term. It aims to disaggregate global consumption at an appliance level, using only a single point of measurement. Various methods have been suggested to infer when appliances are turned *on* and *off,* using the analysis of current and voltage aggregated waveforms.

Within this context, we aim to provide a methodology for NILM to determine which sets of electrical features and feature extraction rates, obtained from aggregated household data, are essential to preserve equivalent levels of accuracy; thus reducing the amount of data that needs to be transferred to, and stored on, cloud servers.

As an addendum to this thesis, a Brazilian appliance dataset, sampled from real appliances, was developed for future NILM developments and research. Beyond that, a low-cost NILM smart meter was developed to encourage consumers to change their habits to more sustainable methods.

Contents

List of Figures

List of Tables

1. Introduction

Energy efficiency is an important subject in today's world agenda; not only because of greenhouse gas emissions, which contribute to global warming, but also because of the possibility of supply interruptions and delivery constraints in some countries and regions of the world.

Recent research (Armel, et al., 2013) has estimated that about 20% of the total demand for commercial and residential energy in the U.S. could be reduced by energy savings. This corresponds to about 8% of the energy consumed by, and distributed through, different end uses, as shown in Table 1. In Brazil, energy waste is conservatively estimated to be around 15% in the residential market (CERNE, 2017).

Table 1 - Electricity Consumption in the USA, by end use, (EIA, 2010)

End Use	Total Consumption (Billion kWh/yr)	Consumption per Household (kWh/yr)	Total Cost (in billions)	Cost per Household
Space Heating	125.5	1,089	$13.2	$115
Air Conditioning	222.2	1,928	$23.5	$204
Water Heating	130.4	1,131	$13.8	$119
Lighting	210.3	1,824	$22.2	$193
Appliances	699.3	6,067	$73.8	$641
Total	1,387.7	12,039	$146.5	$1,271

According to the Electric Power Research Institute's (EPRI, 2009) research, numerous studies have suggested that feedback on specific electrical appliance's consumption could support and encourage consumers in energy conservation. It has proved to be one of cheapest and most eco-friendly ways to improve energy measurement, in order to benefit businesses and environment (THE ECONOMIST, 2014). The term, "Negawatt-hour", was coined by Amory Lovins (Lovins, 1989) to describe a theoretical representation of the amount of energy (measured in kilowatt-hours) saved.

Investment in energy saving is a large and growing market. In 2011 alone, expenditure on energy savings, by governments, utilities and businesses, was estimated to be around $300Bi (THE ECONOMIST, 2014), with the International Energy Agency (IEA, 2011) reporting that the same amount of investment was made to generate electricity from oil, gas and coal. However, a 2009 ScienceDirect paper (Marianne R., 2009) showed that the actual energy saved, in several global energy efficiency endeavors did not have impressive results (Armel, et al., 2013).

Experts postulate that the poor results stem from behavioral barriers (Armel, et al., 2013), and some studies (Pereira, et al., 2012; Quintal, et al., 2010) have found that consumers progressively lose interest in eco-feedback solutions over time (about four weeks after their deployment); thus, diminishing the potential of energy savings.

The 'human element' tends to reduce the feasibility of most solutions; the behavioral culprit is known as the "rebound effect" (Gillingham, et al., 2014). This occurs when the money that is saved through energy economies is used to buy new appliances; thus, generating a new cycle of increased power consumption. This phenomenon demonstrates that consumers would benefit from an ongoing solution that would maintain their understanding of consumption so they would continue to save energy. Moreover, spending billions of dollars, to install "smart meters", without first making a careful consideration of their ability to maximize energy efficiency, is also not a clever solution.

Past studies by the American Council for an Energy-Efficient Economy (ACEEE) indicated that the most savings were achieved, depending on the level of consumer feedback information, which prompted behavioral changes (ACEEE, 2010). Reported energy savings were normally achieved by simple changes in consumer habits or by replacing low-cost, inefficient appliances, such as halogen light bulbs with LED bulbs, and investments in new energy-efficient equipment.

More than fifty studies have attempted to investigate the effects of feedback information's being shown on electricity bills. Significantly, results indicated that the best rates of saving were achieved when users were shown feedback such as appliance-load information (Armel, et al., 2013). Armel's research indicated that solutions based on appliance feedback increased savings by over than 12% using automated and

personalized recommendations, and by sending the information directly to customers' smart phones or to dedicated displays inside their homes.

According to the ACEEE (ACEEE, 2010), and as illustrated Figure 1, there was a correlation between the results of the level of feedback, given to energy consumers and the average percentages of savings. The shown characterization was based on EPRI´s first categorization (EPRI, 2009).

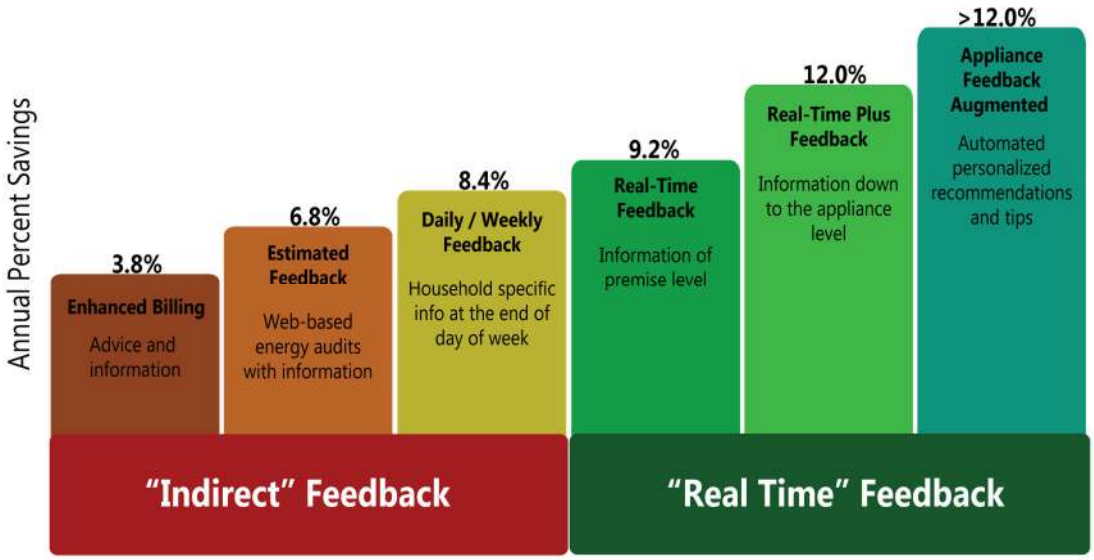

Figure 1 - Average Household electricity savings by feedback [adapted from ACEEE (ACEEE, 2010)]

In addition, utility companies also exhibit a growing demand for information about how consumers are using energy: which appliances are the 'energy villains', and at what time of the day will the demand for power increase? A deeper awareness of energy expenditure and the possibility of use-predictions will help the distribution sector make better contracts with the power plants; thus, reducing energy costs. Moreover, utilities want to be able to provide large *demand side management* (DSM) strategies in order to prevent demand peaks and overloads. For these reasons, electric companies require appliance-level metering to differentiate the individual loads from the global consumption.

Unfortunately, most utilities use intrusive methods to monitor their customers, such as the placement of smart plugs and circuit breakers that sensor and control appliances, individually. Smart equipment, with communication and metering features, are expensive ($25-$100+ per device) (Armel, et al., 2013). Their adoption can incur an extra cost of hundreds to thousands dollars per home or office. This solution would not be feasible for mass adoption and, in most cases, the payback on investment is uncertain.

Nonintrusive Load Monitoring (NILM) is a relatively new term. Its purpose is to disaggregate global consumption down to appliance level, using only a single point of measurement, which is placed on the home breaker panel. Previously, a variety of methods have been proposed, to infer when appliances are turned **on** and **off,** using information taken from an analysis of current and voltage aggregated waveforms. Some methods used the same energy meters that were used by power utilities, to charge consumers. In more advanced solutions a custom designed meter was used to inform users about their consumption.

Several NILM researchers have suggested using the data from the utilities' already installed smart meters, to disaggregate power consumption. This would reduce the cost compared to using dedicated NILM-ready hardware. However, typically, most of current affordable smart meter solutions export basic data only (in some cases only energy values), with a long sampling period (five to sixty+ minutes) of aggregated load records. At these feature extraction rates, NILM's methods remain very inaccurate for low power appliances and offer fair levels of confidence in the absence of a large a-priori known reference database (Weiss, et al., 2012).

In a 2014 study, the U.S. International Trade Commission (Alejandro, et al., 2014) showed that the smart meter market has grown quickly. They estimated that the global market for smart meters could rise, from USD 4Bi in 2011 to USD 20Bi in 2019, with the global market for smart grid equipment expected to rise to more than USD 400Bi by 2020. In the United States, as shown in Figure 2, smart meter penetration rate was lower than 25% in 2014 (Alejandro, et al., 2014). In other developed countries, smart meters may already be a reality; however, the market for selling "NILM-ready" smart meters, to utility companies is huge and is expanding.

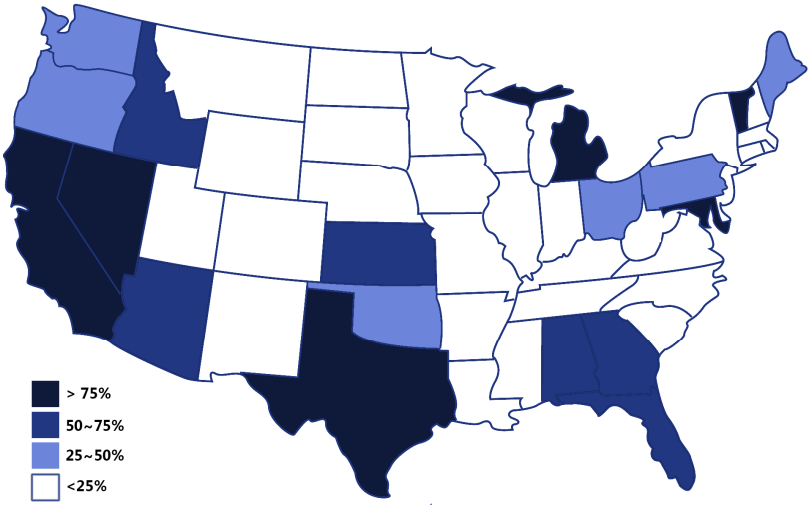

Figure 2 - United States Smart Meter Penetration by 2014 [adapted from (GTM RESEARCH, 2013)]

Beyond that, in Brazil and other developing countries, the majority of electricity meters are still electromechanical induction watt-hour meters. Brazil´s government and utilities are planning to install 63Mi new smart meter units by 2021 (VEZ-BR, 2012) and in India, the government is planning to install 130Mi new units (VEZ-BR, 2012). Therefore, this shows that the smart meter market remains open to opportunities for new "*NILM ready*" hardware custom solutions, and is growing.

Given this situation, we aim to contribute to NILM problem solving, by **providing a methodology for feature extraction rate and feature selection**[1]. The result of proposed methodology is the reduction of the amount of recorded data, necessary for NILM processing, and, therefore, communication bandwidth required, from smart meters to cloud-based servers, lowering the overall cost of the solution, while preserving similar levels on six metric scores for three different classification algorithms. The proposed methodology applies machine learning techniques to search for the lowest feature extraction rate necessary and then to achieve a feature ranking. Moreover, the methodology proposes a minimum set of electrical features, based on

[1] Main contribution of the thesis

feature ranking, which maintains equivalent levels, on the studied scenario, of classification accuracy for each one of the three NILM classification algorithms.

Another outcome of this development is the **delivery of a Brazilian appliance dataset**[2] for present and future NILM developments and research, as well as a **novel method for mixing individual appliances' high frequency measurements in order to create synthetic mixtures of datasets**[3].

Additionally, this thesis will **deliver a smart metering hardware platform**[4], which can potentially help energy consumers develop more sustainable habits. The proposed solution will collect data from a unique point of measurement.

---

[2] BRazilian Appliance Dataset - BRAD

[3] Will mix distinct time sampled appliance data to create simulated household aggregate data

[4] Hardware for NILM community

2. Nonintrusive Appliance Load Monitoring

2.1. Background

Nonintrusive appliance load monitoring focuses on researching how to identify appliances without ambiguity. If power consumption is assumed to be in discrete time, with real values $x(i) \in \mathbb{R} : i \in [1, n]$, with m appliances, and n states for each appliance overall, an aggregate signal can be defined by the formula:

$$x(i) = a_1(i) + a_2(i) + \cdots + a_m(i) + \varepsilon \qquad (2.1.1)$$

where $a_j(i)$ is a particular appliance's consumption, m is the appliance index, ε is noise and $i \in [1, n]$. Based on the principle of superposition, current and its product, power, could be expressed as the overlap of the individual measurements. (Prasad, et al., 2013).

Therefore, a disaggregation problem aims to obtain every single appliance's contribution a_j given a combined $x(i)$. The number of appliances m and each distinctiveness characteristic is, in most cases, indefinite. Thus, most of these signatures must be learned, either individually or, in some cases, from an aggregate with supervised or unsupervised methods. The exact solution for NILM inference is known to be intractable (Pattem, 2012), but most NILM methods attempt to identify as many indistinguishable characteristics of an appliance (such as equipment fingerprints), as possible, to solve the problem.

In general, NILM algorithms can be described in five steps (Berges, et al., 2011):

I. **Data acquisition** –The voltage and/or current signals, from a home´s breaker panel, are acquired and digitalized;

II. **Event detection** – After observing one or more of the measurements that were obtained from the acquired data, events are detected and flagged for later identification.

III. **Feature extraction** – Using the samples of an known appliance, electrical features are calculated, to measure the single characteristics of a piece of equipment, such as active power, reactive power, frequency, Fourier harmonics, the amount of samples, size of step change, transitory time etc.;

IV. **Classification** – Events are labeled and clustered, using statistics or machine learning algorithms, to create a model of how events are designed by those systems;

V. **Energy computation** – Using separate events and transitions, with associated power levels, the specific consumption of each appliance can be estimated, using multiple or single machine learning methods.

Feature extraction is crucial for a complete load identification process. The process of extraction transforms an observation space into a feature space, which, generally, has fewer dimensions than the observation space (Lin, et al., 2011). The focus of past studies has been on which features are absolutely necessary, and sufficient, to achieve results, and how algorithms should process the data from the meter (Prasad, et al., 2013).

George Hart (Hart, 1992) was a pioneer of this research and introduced an NILM load monitor in his article written for the Proceedings of the IEEE, in 1992. In this, he proposed a method to disaggregate appliances from single steady-state metering data, with step changes, calling it an *Event Based* approach. His work proposes an NILM solution for a whole-house power metering system, with different appliances overlapping their different consumption steps, as shown in Figure 3.

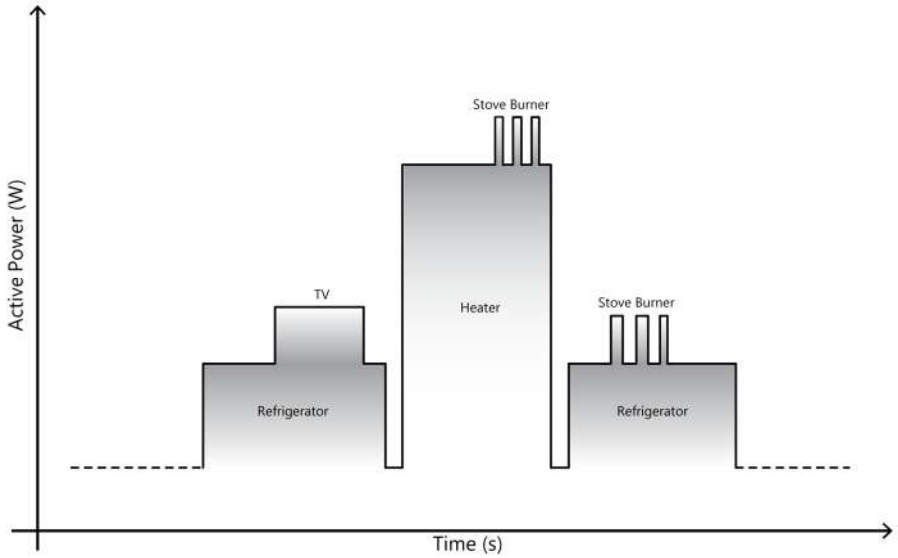

Figure 3 - Total Load vs. Time patterns of appliances [adapted from (Hart, 1992)]

Hart's main idea was to reconstruct the individual power that each load drew, using the aggregated information. In this way he could obtain information about specific appliances, in a non-intrusive manner, and without using any additional hardware to measure the individual appliances (shown in Figure 4).

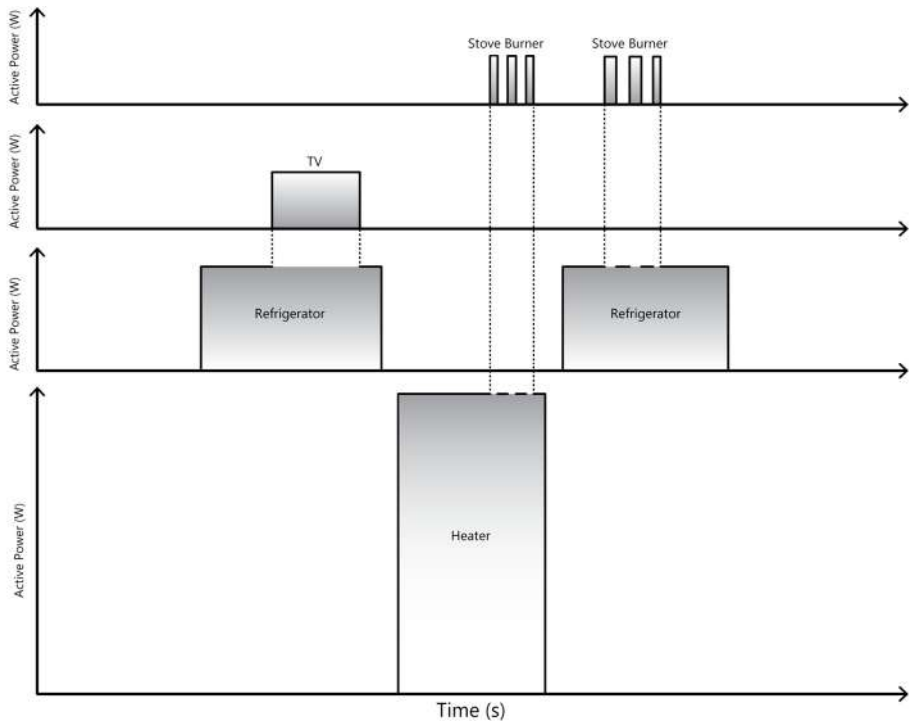

Figure 4 - Disaggregation of an overall signal into individual load appliances

Hart (Hart, 1992) proposed describing the use of measured active power and voltage to calculate "normalized power" as:

$$P_{norm}(t) = \left(\frac{120}{V_{rms}(t)}\right)^2 P(t) \quad (2.1.2)$$

His work suggested normalizing the drawn power by adjusting the measured voltage with nominal voltage values (120V USA), in order to deal with voltage fluctuations. Rapid step changes can give a voltage fluctuation of more than 10% and can affect the power consumption of loads. Hart's normalization proposal is based on the load's admittance (in siemens):

$$Y(t) = \frac{P(t)}{V_{rms}{}^2(t)} \quad (2.1.3)$$

which is a voltage independent property of a linear load, in preference of just power or current signatures. For non-linear devices, the generalization given was:

$$P_{norm}(t) = \left(\frac{120}{V_{rms}(t)}\right)^{\beta} P(t) \quad (2.1.4)$$

where β was empirically parameterized for each individual appliance. Hart's tests suggested values ranging from (0.7, 2.4) for active power. His procedure for power normalization could also apply for reactive power. The measurement results are shown in Figure 5.

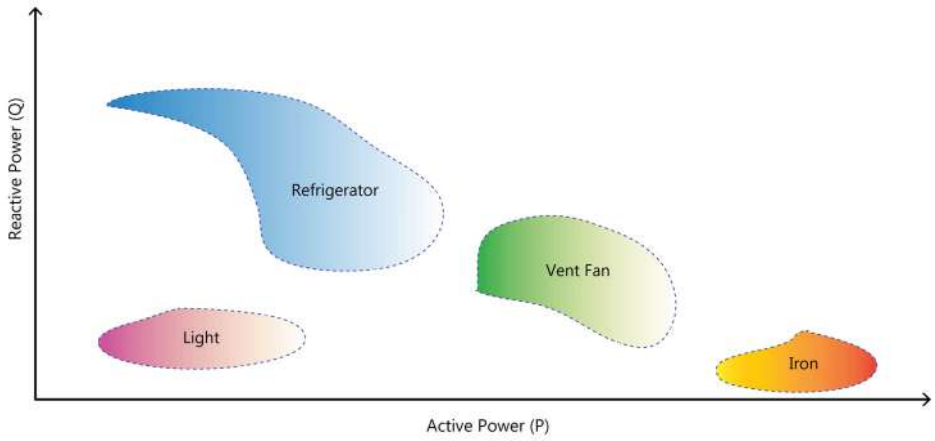

Figure 5 - Signature space on a plane, active power vs. reactive power [adapted from (Hart, 1992)]

Hart (Hart, 1992) also suggested four different groups to characterize appliances; described as follows:

I. **On-Off Appliances** – Appliances such as a light bulb or a water pump with only one **on-off** switch;
II. **Finite State Machines (FSM) or Multistate devices** – Household appliances with a high-load, power-consumption profile such as refrigerators, air conditioners, washing machines and dryers. These appliances pass through definite states, using cycles of different power levels automatically or via human action;
III. **Continuously Variable Devices** – Consumer appliances with a variable range of operations, such as dimmable lights, inverter controlled motors etc.;
IV. **Permanent Consumer Devices** – Appliances which remain **on** for twenty-four hours a day, with approximately constant power consumption, such as security cameras and monitoring systems.

Hart´s method only detects appliances from groups I and II. The power consumption graph for the three first groups is shown on Figure 6.

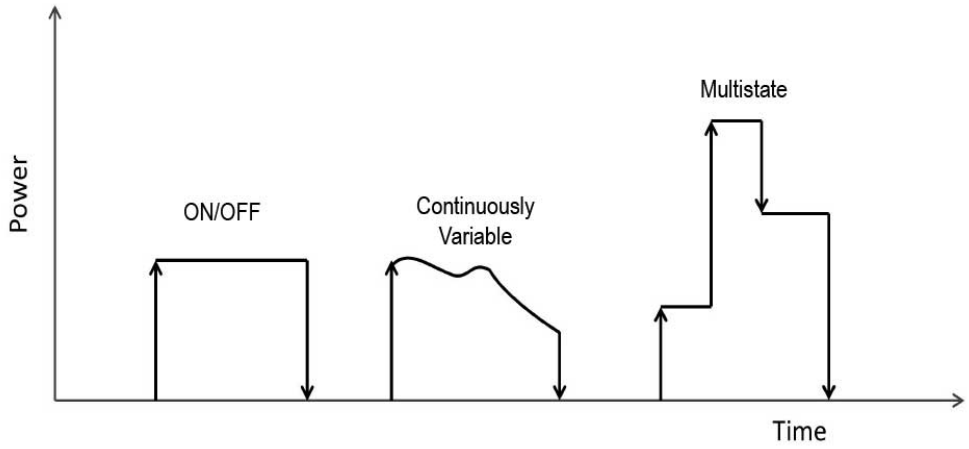

Figure 6 - On/Off, Continuously Variable and Multistate Devices [adapted from (Zoha, et al., 2012)]

Several years after Hart's research, Laughman (Laughman, et al., 2003) showed that the two-dimensional signature space, P*x*Q, became crowded whereas the number of loads increased, which generated ambiguity and indistinguishable results for the separation of loads. Furthermore, Laughman et al.'s illustrated their research with a case where the turning **on** and **off** of a computer and an incandescent light bulb were only distinguishable when a 3rd Fourier harmonic component (in phase) was used in event detection, as illustrated in Figure 7.

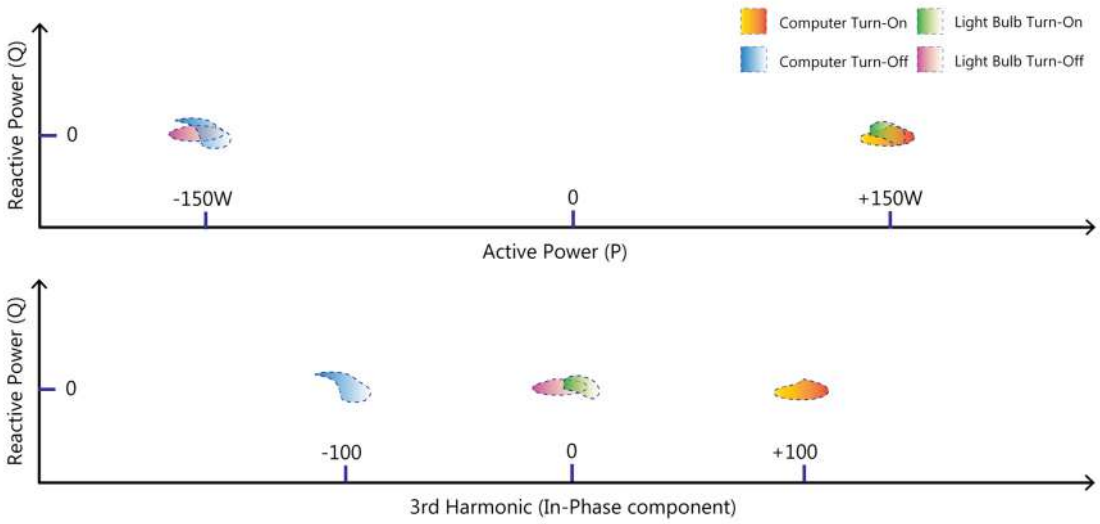

Figure 7 - Computer and light-bulb in plane P x Q (up) and Q x 3rd harmonic (down) [adapted from (Laughman, et al., 2003)]

Instead of steady-state detection, Laughman and his colleagues (Laughman, et al., 2003) introduced transient detection. Employing this extra feature meant appliances could be distinguished via their intrinsic manufacturing characteristics, as the turn-on sampled transients could be compared with the pre-learned, known values of individual measurements, called "ground truth" data.

Additionally, the identification of loads using transient and spectral analysis gives NILM the ability to detect loads that are continuously variable, instead of only FSM or On-Off devices. However, as previously stated, this requires a training period in order to assemble an *a priori* database of known devices and their transient waveforms.

Regardless of the efforts made in NILM, its algorithms have only a marginal accuracy with low-power appliances. For this reason, the global disaggregation results for household appliances are accurate to only approximately 80% (Zeifman, et al.,

2011). The major reason for this is the potential overlap between the different power-drawing levels of appliances, which makes them indistinguishable.

Notwithstanding that, steady-state and/or transient analysis of a load features, such as active power (P), reactive power (Q), transient waveform, harmonic components and their magnitudes, crest factor, current waveform, admittance and power factor have been used to assist in the identification of the individual appliance energy in specific NILM problem solving.

Modern NILM systems, which have feature extraction rates that are larger than 1Hz, have the possibility to disaggregate a wider range of appliances. Armel and her colleagues (Armel, et al., 2013) made a review of publications about scientific energy efficiency and NILM, and related the feature extraction/sampling rate to the maximum number of detectable distinguishable appliances and their categories, as shown in Table 2.

Table 2 - Feature extraction/sampling rate vs. number of detectable appliances [adapted from (Armel, et al., 2013)]

Feature extraction rate	**1 per 15min~1h**	**1 per 1s~1min**	**1Hz to 60Hz**	**60Hz to 40kHz**	**> 1Mhz**
Number of Appliances	~3	<10	10-20	20-40	40-100
Type	Continuous load, time-dependent and correlated with outdoor temperature.	Refrigerator, ACs, heaters, pool pumps, washers, dryers etc.	Steady State steps. Good range.	Toasters, computers etc.	Can differentiate even 2 light bulbs from the same manufacturer.

2.2.NILM Overview

A. I. Cole (Cole, et al., 1998) said, "Any load can be distinguished by its transient changes, its steady-state power levels, or some combination of the two". The objective of NILM is to recognize multiple individual loads in aggregate and to assign the energy for each appliance correctly.

The correct assignment of disaggregated energy for each appliance is a complex process of recognition and classification, and multiple changes in the load states can give unclear results. Additionally, many loads have oscillating, transitory or ramping changes in their power waveforms, which hinder the disambiguation process. Therefore, whenever the magnitude of an electrical feature, from the various appliances, overlaps or falls below the measurement uncertainties, load recognition may not be ensured (Bouhouras, et al., 2012). For this reason, NILM researchers have begun to look for different methods to **detect**, **learn** and **classify** both transient and steady-state power transitions, in order to solve the disaggregation problem.

The majority of NILM classification methods consider that only one appliance may be activated in one instance of time, using the "*Switch Continuity Principle*" (Hart, 1992), which assumes that in a small timeframe, two or more appliances cannot be turned on or off at the exactly same moment.

The identification and classification of existing NILM approaches are based on the multi-label classification techniques, used in the field of information theory. A classification learner maps a vector through different appliance labels, using historical data. According to Basu and his colleagues (Basu, et al., 2015), there are two broad approaches to handling these classification algorithms: the first is by problem transformation; that is reducing a multi-label problem into one or more single-label problems, such as SVM (Support Vector Machines) or DT (Decision Tree). The second is by using an algorithm adaptation method, such as ML-KNN (Multi-Label K-Nearest Neighbor), to modify existing single-label algorithms for multi-label classification.

The Hidden Markov Model (HMM), and its variations, have also been used in NILM. This method uses the load measurements (observations) to find the best set of parameters, in order to maximize the probability of observing a sequence of particular events, in a given probability model.

Electric loads can also be characterized by their front-end circuit topologies. This strategy makes it possible to extract common "*fingerprints*" for each group of appliances, and the resultant categories are described by Du, in his research work (Du, et al., 2015):

I. **Category R** (*resistive loads*) – There is no phase angle between voltage and current. The load usually contains a resistance connected to the front-end;

II. **Category X** (*reactive loads*) – There is a large phase difference between voltage and current. Typically an inductance is connected to the front-end circuit;

III. **Category NP** (*electronic loads without power factor correction*) – The current's waveforms contains abundant harmonics. This normally consists of a rectifier, filters (EMI, voltage or current), and a DC-DC converter;

IV. **Category P** (*electronic loads with power factor correction*) – Generally, this consists of a front-end EMI, a rectifier, a voltage regulator, PFCM (power factor correction module) and a DC-DC converter**;**

V. **Category T** (*linear loads*) – Commonly consists of a transformer and other downstream electronics;

VI. **Category PAC** (*phase angle controllable loads*) – The PAC equipment continuously adjusts its load current by controlling the firing angle of a switch (e.g. a thyristor or IGBT);

VII. **Category M** (*complex structure loads*) – In this category, the current's waveforms are made up of multiple categories - the result of a multiple front-end power-supply unit;

As shown in Figure 8, an electric load can also be identified by its V-I trajectory graphical signature. Under this condition, the classification of an unknown load can be achieved by comparing its patterns with one of the known, listed categories and without using Fourier transformations or any frequency domain analysis.

In the NILM load detection process, *variable* and *always-on* loads require discreet techniques in order to be classified correctly. The classification of variable speed drives, power electronic controlled loads (inverters) and similar, are often labeled as non-identifiable loads.

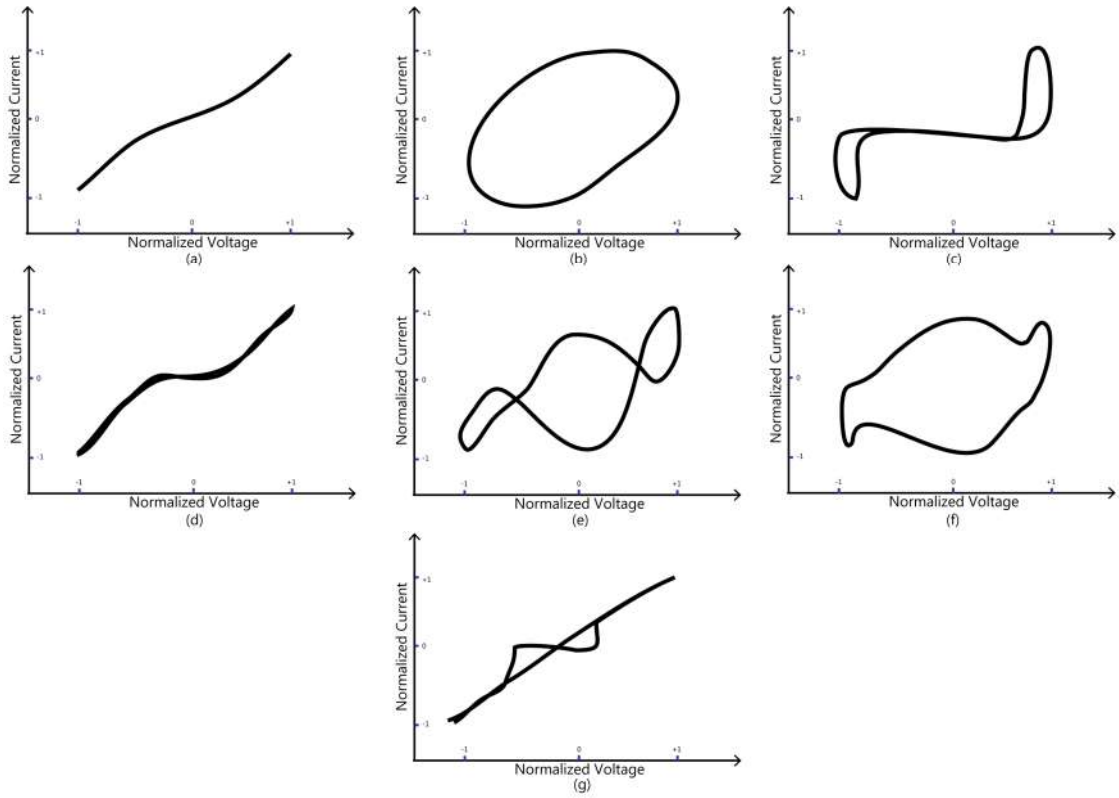

Figure 8 - Normalized expected V-I trajectories of the seven load categories
(a) R (b) X (c) NP (d) P (e) M (f) T (g) PAC [adapted from (Du, et al., 2015)]

Wichakool and his colleagues (Wichakool, et al., 2015) presented a method that uses frequency domain and harmonic spectrum to identify power electronic controlled loads, by observing the intrinsic properties of a current's waveform. Using sampled and synthetic waveforms of variable speed drives (VSDs), computers, light dimmers and other variable loads, to observe the common properties of the loads, such as periodicity, the presence of zero-current regions, symmetry and other features, with very promising results.

The classification of NILM algorithms usually involves *Event Based* (EB) and *Non-Event Based* (NEB) methods. Event based methods can easily detect **on-off** appliances and *multistate devices* (FSM) assuming the system has been previously trained, but they

usually have trouble distinguishing *variable* and *always on* loads. Conversely, Non-event based methods generally have a higher computational complexity, and use the entire waveform, or a long time-series of computed features mean values, to evaluate problems. They employ *data mining* or *hidden Markov* modeling techniques and do not require any previous knowledge of the appliances (Pereira, et al., 2015).

NILM algorithms have also been classified by their feature extraction/sampling rate; however, there is no consensus about the rates and related terminology. As an example, a recent work (Parson, 2014) classified *High Frequency* measurements as being a feature extraction rate that was greater than 1 Hz. Conversely, Patel's work, from 2007 (Patel, et al., 2007), put the *High Frequency* sampling rate between 50 kHz and 100 MHz.

This thesis used sampling rates of 10 kHz for current and voltage, which gave a large Fourier harmonic spectrum for measurements, at 60 Hz (Brazil, USA etc.) and 50 Hz (most European countries etc.). Thus, as concerns the feature extraction rate, our measurements occurred from 60 Hz (~16.67ms) to 1/15 Hz (15s).

In general, machine learning algorithms, as well as NILM classification methods, could also be categorized using their learning technique. *Supervised*, *Semi-supervised* and *Unsupervised* methods could be used for NILM problem solving.

2.3. NILM Approaches

2.3.1. Event Based Methods

Recent Event Based (EB) implementations use digital "de-noising" to plane edges' transition. A median filter is also widely used, but has the disadvantage of frequently smoothing edges (Liu, et al., 2014). Basseville (Basseville, 1986) was the first to propose a method for detecting sequential processes that he called the *generalized likelihood ratio* (GLR). A modified GLR detector was also implemented by Luo (Luo, et al., 2002), who benefited from a reduced number of parameters to be set.

A group of researchers from CMU Inferlab (Berges, et al., 2011; Berges, et al., 2009) used a modified version of the GLR to detect abrupt changes in power

measurements and a *nearest neighbor* approach, in *Euclidean* space, to perform NILM classification.

Research from Carnegie Mellon University (Giri, et al., 2013) sought an automated EB unsupervised method, to label step changes in real power, using a pre-defined threshold (50W in their approach). When, on average, power measurement stays above 50% of a previous event and for at least two seconds, transition is labeled as *on*. For the *off* labels, parameterization is to 80%. This step is characterized as *event detection*, as explained in item 2.1.

In the *Feature extraction* step of their work (Giri, et al., 2013), researchers extracted three seconds of transient for each event and, subsequently, normalized it. The normalization process for *on* events includes four steps: (1) a *smoothing process*, done by a moving the average window of 10ms; (2) a *threshold selection,* calculating the RMS values of cycles and choosing the threshold using 5% of the lowest established value; (3) a *non-negative threshold normalization*, which removes the threshold offset, point-wise, and changes negative values to zero; and (4) *normalization by maximum,* where all the data were normalized by their highest RMS value.

Feature extraction for *off* events was different, because *off* transients looked the same for all the devices they studied in their work (Giri, et al., 2013). The solution they adopted was to subtract the aligned current waveforms by phase (finding zero-crossing of the voltage) before and after the timestamp of *event detection*. The waveform generated by the difference (between the before and after measurement) was then normalized and used as a classification feature.

Kamat (Kamat, 2004) used Fuzzy Logic theory, originally developed by Zadeh (Zadeh, 1973), to introduce a supervised pattern recognition approach, using specialist knowledge to identify appliance patterns, without the need for extensive training or the implementation of an artificial neural network. This technique proposed matching transients in waveform with a pre-set-up database. An algorithm compared two vectors (a and b) of length n.

Ducange and his colleagues (Ducange, et al., 2014) also implemented an approach based on Fuzzy Logic Transitions for NILM multistate and **on-off** devices. Their work associated linguistic variables and triangular fuzzy sets and fuzzy transitions $T(S_i \rightarrow S_j)$ through the $\boldsymbol{IF\text{-}THEN}$ rule:

$$R_{i,j}: \textbf{IF } State \textbf{ is } S_i \textbf{ AND } \Delta P \textbf{ is } A_{\Delta P,k_{i,j}} \textbf{ AND } \Delta Q \textbf{ is } A_{\Delta Q,k_{i,j}} \textbf{ THEN } State \textbf{ is } S_j$$

where S_n is a set of the working states of generic appliances; ΔP and ΔQ are linguistic variables defined from a variation of active power and reactive power steps for each appliance, in a given rule universe, as displayed in Table 3.

Table 3 - Transition rules and working states of monitored appliances (Ducange, et al., 2014)

Appliance	State	Description	If State is	And ΔP is	And ΔQ is	Then State is
Food Cutter	FC0 FC1	OFF ON	FC0 FC1	FS10 FS7	FS13 FS2	FC1 FC0
Hair Dryer	HD0 HD1 HD2	OFF COLD HOT	HD0 HD1 HD1 HD2	FS12 FS5 FS12 FS5	FS7 FS8 FS7 FS8	HD1 HD0 HD2 HD1
Refrigerator	R0 R1 R2 R3	OFF PEAK ENGINE 1 ENGINE 2	R0 R1 R2 R1 R3 R2	FS13 FS4 FS13 FS4 FS8 FS8	FS14 FS5 FS14 FS4 FS5 FS4	R1 R2 R1 R3 R2 R0
Microwave oven	MO0 MO1 MO2 MO3	OFF ON Power 1 ON Power 2 ON Power 3	MO0 MO1 MO2 MO2 MO3	FS10 FS14 FS3 FS3 FS9	FS14 FS1 FS13 FS14 FS14	MO1 MO2 MO0 MO3 MO1

Certainly this approach is not feasible for a real world application, as it manually generates all the fuzzy rules and transitions for every appliance. However, if the rules and states could be automatically generated from previously-learned data, the fuzzy approach might guarantee a simple way to do state transitions on an EB NILM approach, with a very low computational cost.

2.3.2. Non-Event Based Methods

The labeling process for event based methods is considered an error-prone process, which could decrease the confidence of a classification method. Non-event-based NILM approaches eliminate the necessity to label every event, in available datasets, in order to

provide information about the "ground truth" data. Every feature extraction is used to determine the actual and future appliance status.

Blind source separation (Cardoso, 1998), a signal processing subject, could also be used as a *Non-Event Based* NILM classification algorithm. Processing audio for channel separation– Independent Component Analysis (ICA) - where the aim is to disaggregate multiple sound sources, could be applied to NILM, which has electrical mixed measurements with appliances being their different sources. However, Parson (Parson, 2014) described this kind of approach as not getting any benefits from the dependencies of metering correlation among sequential events and, said that it offers low scalability to a large number of simultaneous sources (the number of appliances in a home or office).

Hidden Markov models have been widely used in the scientific community as machine learning algorithms (Ridi, et al., 2014). They are in the group of probabilistic temporal graphical models, and are predominantly acknowledged as useful in problems concerning temporal pattern recognition. HMM compute the probability of a particular sequence of events.

Kim and his colleagues (Kim, et al., 2011) used four variants of a HMM to model data: the factorial hidden Markov model (FHMM); the conditional factorial hidden Markov model (CFHMM); the factorial hidden semi-Markov model (FHSMM); and a combination of FHSMM and CFHMM. These variants of the original HMM were proposed in order to incorporate additional features to be modeled, such as the time of the day, appliances' dependencies and other sensor measurements. Data were collected from seven homes, over a six-month period, from different appliances, as illustrated in Figure 9.

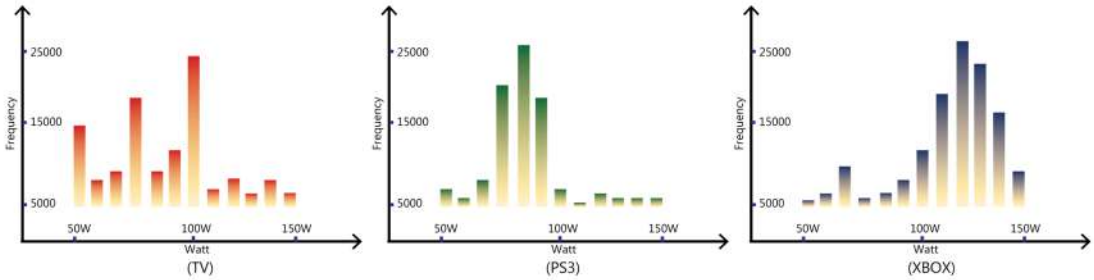

Figure 9 – Histograms of TVs, PS3 and XBOX of different homes [adapted from (Kim, et al., 2011)]

Pattem (Pattem, 2012) also used a Viterbi algorithm to find the most likely state sequence and, in particular, to find the ´zero´ state (i.e. the state where all appliances are at start of measurement) with a high degree of confidence, thus preventing errors from one part of a sequence affecting the other parts.

Ridi (Ridi, et al., 2014) also used HMM. This group of researchers used 0.1Hz as their feature extraction rate and a window length for each analysis of up to 50 seconds, so they could cluster the states of different appliance categories with a reasonable success ratio.

2.4.Feature Extraction Rate

2.4.1. Low frequency

Low frequency, feature extraction rate algorithms, also called steady-state methods, aim to use the data that is gathered from existing installed utilities' smart meters. Parson´s work (Parson, 2014) established a minimum feature extraction rate of approximately 10 seconds, to achieve reasonable NILM accuracy. The majority of NILM *Low frequency* algorithms, in the literature, have used feature extraction rates from 1Hz to 1/10Hz.

However, most of these smart meters send measured data to utilities with a periodicity of fifteen or more minutes and show a reduced set of features; thus, they require the installation of extra custom-hardware (or, at least, modified billing meter firmware) to solve the disaggregation problem.

Dinesh and his colleagues (Dinesh, et al., 2015) developed a method that could "tolerate very low sampling rates" (they said of 1 Hz or less) without any significant level loss in NILM metrics. Their method used an auto-correlation function matrix (ACM) and the Karhunen-Loève Transform (KLT) (Maccone, 1994), which is another expansive method of representing a signal spectrum, as an electrical feature measurement disambiguation, using voltage and current waveforms.

As a result, they (Dinesh, et al., 2015) computed the uncorrelated spectral components of a washing machine and an LCD TV, presented for both Fourier and Karhunen-Loève frequency domain information. Their results demonstrated a broader and richer orthogonal harmonic content for KLT, which suggested they might be useful and promising features for future NILM disambiguation analysis.

2.4.2. High frequency

High Frequency NILM methods that use a higher sampling and/or feature extraction rate, can provide better results when more appliances are in the mixture. Typically, these methods use transients to detect waveform changes, or, extended feature extraction.

Feng and his colleagues (Feng, et al., 2013) used the waveforms of currents to take advantage of the harmonic features in the identification process. They used the odd Fourier harmonics, from 3^{rd} to 15^{th}, and their work combined eight similar types of active power loads (Compact Fluorescent Lamps - CFLs, PC´s and T5 lamps) mixed in a controlled test bed environment, with individual switches. Their results showed that when new features were combined with existent active and reactive power measurements, classification algorithms could increase disaggregation rates.

A researcher who sought more features to assist in the disambiguation process, proposed the use of wavelet transform coefficients (WCTs), discrete wavelet transforms (DWT) and multi-resolution analysis (MRA) (Chen, et al., 2013; Chang, et al., 2014) for additional feature extraction. They also suggested the use of back-propagation artificial neural networks (BP-ANN) (Chang, et al., 2014) and inner product (Chen, et al., 2013) to resolve NILM disaggregation problem. Their (Chen, et al., 2013; Chang, et al., 2014) sampling frequency was 15 kHz and it included six different loads, with *variable* and *multistate* power loads. After processing the data, the achieved results were superior to those methods that only used P (active power) and Q (reactive power) as the electrical characteristics for disambiguation.

2.4.3. Very high frequency

In 2007 researchers (Patel, et al., 2007) presented a new NILM method that explored the broadband electrical noise produced by the swift switch of electrical transient loads between hot and neutral (normal mode noise) and from neutral to ground (common mode noise). Patel (Patel, et al., 2007) stated that observed noise was close to AM radio frequencies and typically lasted only a few microseconds. Radio transient noise flows through power lines and can be detected at any electrical outlet.

The device generated noise, returned to power lines, cannot surpass certain limits that are defined by the Federal Communications Commission (FCC, 2016), for a spectrum of up to 30 MHz. Patel's research (Patel, et al., 2007) acquired data from 100 Hz to 100 MHz. The selected point of measurement was a specific outlet in the home. This meant that the electrical measurement point did not necessarily have to be at the home breaker panel, in order to sample the complete current load flow.

The test results, for *event detection,* were promising but, as expected, the system had some limitations to mass adoption: the hardware that generated such remarkable sampling rates was expensive. Nevertheless, this method is promising as a way of lowering installation costs and complexity, because it can be plugged into any outlet as an ordinary appliance; thus obviating the need for handling the home´s breaker panel.

2.5. NILM Learning Process

The process of classification is an extensive topic when discussing NILM approaches. Methods vary; some use supervised learning techniques that require training and validation from direct observations; while others use unsupervised learning, where an algorithm automatically detects steps and creates a self-labeling and clustering process. Algorithms that could be used, might range from the k-nearest neighbor (Pereira, et al., 2012) to Deep Learning techniques (Kelly, et al., 2015), using advanced neural and machine learning systems in order to achieve the correct regression and/or energy classification process.

Prasad (Prasad, et al., 2013) used a multi-algorithm approach, through a technique called a committee decision mechanism (CDM). This uses a voting scheme, of multiple classification algorithms, to solve NILM. The method performs successive eliminations for the classification filtering problem, where the results of each algorithm are evaluated one at a time, sorted by their computational effort, to distinguish multiple loads.

Automatic clustering techniques that use unsupervised learning could also be used to discover the loads' multiple power steps. In recent research, Giri (Giri, et al., 2015) proposed a method to create a clustering map, automatically, based on a variation of active (ΔP) and reactive (ΔQ) power and, a state transition model, using transition probability matrices and historical data, as shown in Figure 10. Bergman (Bergman, et al., 2011) suggested that genetic algorithms, and dynamic programming, could also be applied to build *finite state machines* (FSM) from the events that occurred in a temporal series.

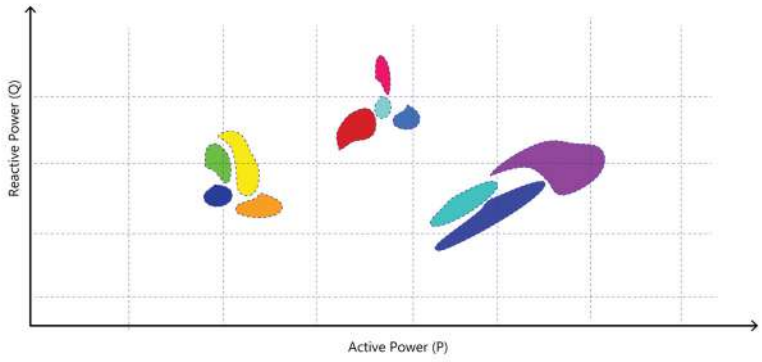

Figure 10 - Clustering results for a fridge, in the plane ΔP x ΔQ [adapted from (Giri, et al., 2015)]

Additional approaches have been implemented using machine learning algorithms that fall within the multi-label classification, such as the *boosting* technique (Breiman, 1998), a method which selects "weak learners" in a voting scheme, to create a "strong learner"; *multi-label k-nearest-neighbor* (ML-KNN) (Zhang, et al., 2007) a method that extends the single-label k-NN to a multi-label classifier; and *back-propagation multi-label learning* (BP-MLL) (Zhang, et al., 2006), an extension of back-propagation algorithm.

Lin developed a *Neuro-Fuzzy* classification method, with a *Fuzzy C-Means* clustering algorithm (Lin, et al., 2014). Pereira (Pereira, et al., 2012) also references a Java machine learning (JML) and Weka APIs for NILM classification.

2.5.1. Supervised learning

At present, the methods that are conventionally associated with *supervised* approaches use previously human-labeled training samples to support their classification algorithms. However, the manual provisioning of labels, concerning the state of each appliance on a timeline, could potentially be at risk of human error in real-life scenarios. Additionally, brand new appliances are released every day and number of manufacturers is growing exponentially, making such labeling almost impossible to maintain.

Regrettably, manual labeling demands homeowners switch each monitored appliance and then label all the states separately. This task is inconvenient and future users may hesitate to accept this task (Iwayemi, et al., 2015).

In an effort to create a signature database, that would support and help future supervised methods, Klein (Klein, et al., 2014) acquired data from a set of appliances such as refrigerators and freezers, microwave ovens, and washing machines. After processing the data, the researchers found similar characteristics in these classes. This is illustrated in Figure 11, which is uses the example of cooling devices.

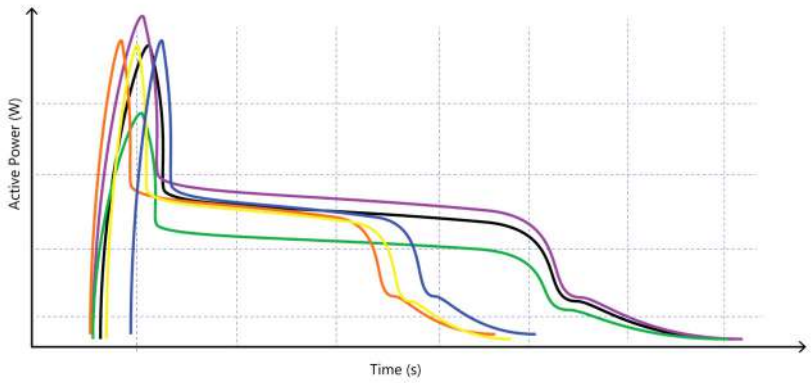

Figure 11 - Power signatures of consumer cooling devices from six different manufacturers [adapted from (Klein, et al., 2014)]

In 2011, researchers presented an interactive supervised learning method called dNILM (Bergman, et al., 2011). This distinct solution reacts continuously when a new appliance is found and, after asking users, it appends the new, unique appliance signature to a dynamic table in its memory. After insertion, a timeout counter begins, to prevent unused or broken electrical devices remaining in its memory indefinitely.

2.5.2. Unsupervised learning

Under NILM *unsupervised learning,* the labeling process is skipped and no human intervention is needed. The aim is to acknowledge the differences in features automatically and to cluster distinctive equipment together, using only aggregated data. *Dynamic time warping* (DTW) (Müller, 2007) is an example of a scalable unsupervised method, which does not require re-training of the entire dataset when new appliances are added to the set-up.

Liao and his colleagues (Liao, et al., 2014) compared three different NILM unsupervised methods: *decision tree* (DT), *dynamic time warping* (DTW) and the *hidden Markov model,* on five distinct appliances: one toaster, a dishwasher, a washing machine with integrated dryer, a microwave oven and a refrigerator. Their results showed that each classification algorithm performed better on a specific kind of appliance, reinforcing the idea of the benefits of using multiple algorithms, for disaggregating several appliances.

2.5.3. Semi-supervised learning

Semi-supervised learning exploits the benefits of a limited quantity of previous knowledge, garnered from labeled data and the processing of large amounts of unlabeled data. The process of learning appliance signatures from one home to another is called **generalization**. Poorer results were always obtained when the generalization was applied. This approach aims to provide a stronger generalization of a hybrid, machine-learning algorithm.

Support vector machine (SVM) methods, such as *semi supervised support vector machine* (S3VM) and *transductive support vector machine* (TSVM) are examples of

methods in this class. By definition an SVM constructs a hyperplane (n-dimensional space) which could be used for classification tasks. As a drawback to this method, the computational load of an SVM may need to be very high to ensure a reasonable outcome.

Iwayemi (Iwayemi, et al., 2015) designed a semi-supervised method that returned promising results, even in generalization scenarios. However, no classifier is currently available, which works for any device and under all circumstances (Liu, et al., 2014).

2.6.Privacy Concerns

Today, automated systems that are able to obtain knowledge, from large amounts of data are a reality. "Big Data" is a huge trend in computer-aided analysis; however, the mining of personal data, to gather valuable information, brings questions about NILM usage. The extraction of valuable awareness about user behaviors is positive when it is done to improve efficiency in energy consumption. However, the utilities and companies that possess these data could easily discover other facts, such as when a consumer is at home, what time they took their last shower or when they watched television.

If the aim is to safeguard privacy, it is mandatory to store data anonymously. However, in order to inform each user about their own consumption habits, NILM systems must process information in a personalized way. Thus, directly or indirectly, NILM solutions carry the risk of disclosing personal information in the disaggregation process.

In research conducted in 2013, Chen (Chen, et al., 2013) associated appliances with daily activities. Researchers identified simple household tasks such as *waking up*, *cooking*, *taking a shower*, and *arriving home* with an average accuracy of around 85%, after 100 days of artificial intelligence training and 50 days of testing.

Researchers (Alcala, et al., 2015) also used NILM to monitor the health of elderly people, living alone, which would seem a positive use of NILM behavioral knowledge. Using the recognition of the weekly usage from kettles and probabilistic models, the system triggered alarms when it detected a deviation in patterns of routine, permitting

earlier healthcare interventions in houses that had previously demonstrated a consistent routine in more than 80% of cases.

However, NILM could potentially expose companies internal secrets to competitors, when used in industrial settings, revealing information about occupancy, server loads, and industrial processes (Adabi, et al., 2015) etc. Nevertheless, it is clear that the use of NILM could benefit most consumers, by improving efficiency in energy consumption and even changing how customers select and acquire their new household appliances.

2.7.Energy household datasets

Energy datasets are a basic requirement of NILM academic research, providing the means to compare multiple methods and algorithms' performance using the same source of data. Both household (aggregate) and individual appliance energy data (ground truth) must be sampled to enable performance comparison over different techniques.

Table 4 displays public datasets for NILM. The selection was made based on Bonfigli (Bonfigli, et al., 2015) and Monacchi's (Monacchi, 2014) research. REDD (Kolter, et al., 2011) was the first to release a public household dataset that was available for NILM research. This dataset has been used extensively in NILM performance evaluation and contains both the data of aggregate power consumption (sampled at 15 kHz) and circuit-level data (up to four seconds for each measure).

In 2015, researchers proposed (Pereira, et al., 2015) the creation of a standard file format for datasets, based on the standard, resource-interchange file format – waveform (RIFF-WAVE) format. The researchers remarked that the file format supports the storage of data, and the relevant metadata, for all NILM applications. Kelly (Kelly, 2014) further proposed a standard metadata scheme to represent appliances, meters, buildings, and all known actors, which could be included in NILM research. It creates a tier-based model to describe how appliances and meters connections are interrelated. Geo-based and weather information could also be represented.

In other studies, researchers (Pereira, et al., 2014) provided detailed information about the included metadata, in SURF file format for an NILM *Event Based* dataset, with the associated timestamps and information about appliances.

PLAID (Gao, et al., 2014) was designed to be a crowd-sourcing individual appliance dataset, which sampled voltage and current waveforms at 30 kHz. It sampled eleven different appliance types, from different manufactures and models, for a few seconds, and the data is available for public download.

Table 4 – Energy Household Datasets

Dataset	Location	Duration per house	Number of houses	Appliance sample resolution	Aggregate sample resolution	Features	Sensors	Reference
ACS-F2	SWZ	1 hour	N/A	10 sec	10 sec	I, V, Q, f, Φ	225 devices in total (10types)	(Ridi, et al., 2014)
AMPds	CDN	1 year	1	1 min	1 min	I, V, pf, F, P, Q, S	19	(Makonin, et al., 2013)
BERDS	USA	1 year	N/A	20 sec	20 sec	P, climate	55	(Maasoumy, et al., 2013)
BLUED	USA	8 days	1	state transition label	12 kHz	I, V, switch-events	Aggregated	(Anderson, et al., 2012)
COMBED	IND	18 months	8	30 sec	30 sec	8 parameters	200	(Batra, et al., 2014)
ECO	CH	8 months	6	1 sec	1 sec	P, Q	6-10	(Kleiminger, et al., 2015)
GreenD	AT/IT	1 year	9	1 sec	1 sec	P	9	(Monacchi, 2014)
HES	UK	1 or 12 months	251	2 or 10 min	2 or 10 min	P	13-51	(Zimmermann, et al., 2012)
iAWE	IND	73 days	1	1 or 6 sec	1 sec	V, I, f, P, S, E, Φ	33 sensors 10 appliance	(Batra, et al., 2013)
Pecan Street Sample	USA	7 days	10	1 min	1 min	S	12	(PECAN STREET, 2016)
PLAID	USA	Few seconds	55	30 kHz	N/A	V, I	11 appliance types	(Gao, et al., 2014)
REDD	USA	3-19 days	6	3 sec	1 sec & 15 kHz	Aggregate: V, P Sub meter: P	9-24	(Kolter, et al., 2011)
SustData	PT	5 years	50	50 Hz	50 Hz	P, Q, S, V, I	50	(Pereira, et al., 2014)
Tracebase	DE	N/A	15	1-10 sec	N/A	P	158appliance (43 types)	(Reinhardt, et al., 2012)
UK-DALE	UK	3-17months	4	6 sec	1-6 sec & 16 kHz	Aggregate: P Sub meter: P, switch-status	5 (house 3) 53 (house1)	(Kelly, et al., 2015)
UMass Smart	USA	3 months	3	1 sec	1 sec	Aggregate: P, S Sub meter: P	25 circuits, 29 appliances	(Barker, et al., 2012)

Unfortunately, there is no standardization, for datasets specifications, for a minimum sampling rate of aggregate data, a sampling rate of individual appliances, the number of appliances, or even any data format or associated metadata. NILMTK (Batra, et al., 2014) is an open-source toolkit for NILM which contains a parser for the various datasets available, as part of an effort to offer data interchangeability between the different datasets. Nevertheless, appliance generalization between different datasets is still is a difficult development.

2.8.Metrics

Differing NILM approaches and algorithms require an objective manner, to evaluate and compare performance. In order to achieve accurate characterization in NILM, a tradeoff will be necessary between *sensitivity*, when missing detections occur, and *specificity*, when false-positives are discovered. Researchers (Zeifman, et al., 2011) suggested using *receiver operating characteristic* (ROC) curve analysis (Egan, 1975) to improve the fine tuning parameterization of NILM methods.

A set of metrics was also used, to compare applications in energy disaggregation researches. In Batra and his colleagues' work on NILMTK (Batra, et al., 2014), and in other studies (Bonfigli, et al., 2015), researchers described each metric used in NILM and its mathematical definition and meaning. Their objective was to validate appliance states; for example, **on** or **off** (shown as y), and each energy contribution (stated as E) from different perspectives, and assuming that both the classification of the correct state and the reconstruction of the power waveform were essential for a complete NILM algorithm.

Thus, measured data, as real measurements from appliances, are defined as:

- E_i - energy measured of appliance i;
- y_i - measured state of appliance i;

Reconstructed estimated data, as inferred output from the NILM algorithm, are defined as:

- $\breve{E}_i$ energy reconstructed of appliance i;

- $\breve{y}_i$ reconstructed state of appliance i;

where:

- K is the number of appliances;
- T is the number of sample time index;
- t is the time index;

and:

- True positives (TP) are defined as:

$$TP = \sum_{i=1}^{K} [(y_i == on)\ AND\ (\breve{y}_i == on)]$$

- True negatives (TN) are defined as:

$$TN = \sum_{i=1}^{K} [(y_i == off)\ AND\ (\breve{y}_i == off)]$$

- False positives (FP) are defined as:

$$FP = \sum_{i=1}^{K} [(y_i == off)\ AND\ (\breve{y}_i == on)]$$

- and False negatives (FN) are defined as:

$$FN = \sum_{i=1}^{K} [(y_i == on)\ AND\ (\breve{y}_i == off)]$$

This thesis selected the metrics that were used the most in previous studies, to measure NILM performance. They were defined as:

I. **Total energy correctly assigned (TECA)**

$$\boldsymbol{TECA = 1 - \frac{\sum_{t=1}^{T}\sum_{i=1}^{K}|E_i(t) - \breve{E}_i(t)|}{2\sum_{i=1}^{K} E_i}}$$

II. **Normalized error in assigned power (NEAP)**

$$\boldsymbol{NEAP = \frac{\sum_{i=1}^{K}|(E_i - \breve{E}_i)|}{\sum_{i=1}^{K} E_i}}$$

III. **Precision (P) –** The ratio of correctly predicted ON observations to the total predicted ON observations:

$$\boldsymbol{P = \frac{TP}{TP + FP}}$$

IV. **Recall (R) –** The ratio of correctly predicted ON observations to all ON observations:

$$R = \frac{TP}{TP + FN}$$

V. **Accuracy (A) –** The ratio of correctly predicted observation to the total observations:

$$A = \frac{TP+TN}{TP+FP+TN+FN}$$

VI. **F-measure / F-score (F_1) -** The weighted average of Precision and Recall

$$F_1 = \frac{2.P.R}{P+R}$$

Empirically, we can infer that, as the complexity of the aggregated data increases, so NILM's methods obtain a poorer performance. One example can be observed in the work of Kim and his colleagues (Kim, et al., 2011), where performance, which was measured by F-score, falls as the number of appliances is increased (shown in Figure 12). Similarly, the same behavior is expected when sampling frequency, or the number of features, decreases.

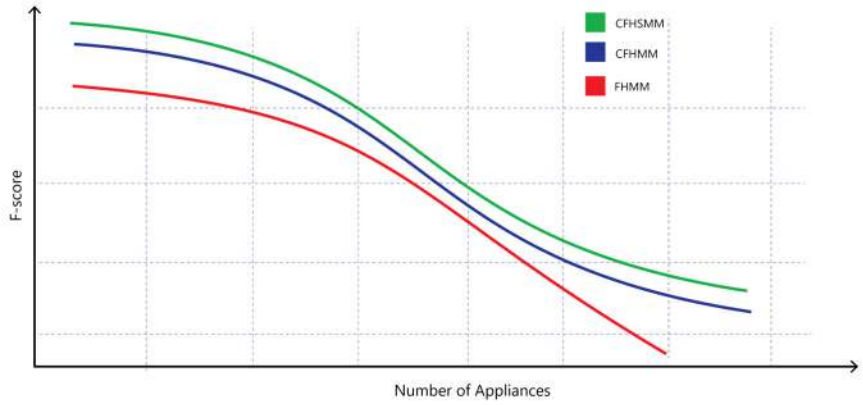

Figure 12 – Performance comparison of multiple models vs. number of mixed appliances [adapted from (Kim, et al., 2011)]

Most implementations of NILM only compare the effectiveness of load level recognition rate with defined metrics. However, the efficiency of NILM algorithms also depends on the time it takes to compute the method, especially for those requiring real-time applications (Bouhouras, et al., 2012).

3. Methodology overview

The term ‘machine learning’ was invented by Arthur Samuel in 1959 (Koza, et al., 1996). It refers to a computer’s ability to solve problems and to execute tasks without being explicitly programmed to do so. Today, in a world where computers “see” things and “understand” people’s voice commands; where cities are “smart”, human life is logged and connected twenty-four hours a day over social networks and sensing health and environment are a reality, developing systems to help analyze these vast amounts of data is crucial.

Recent progress in the field of machine learning algorithms is formidable. However, large amounts of data are required for training large models successfully, including the highly advanced techniques of Deep Learning. These deeply structured and hierarchical learning mechanisms were inspired by nature, in a similar manner to the functioning of the brain’s neurons.

As mentioned earlier (in Section 2.7), much effort has been made to create datasets that epitomize human behavior inside households, requiring the collection of years’ worth of data, to achieve the mass information required to train machine learning algorithms.

A similar approach was successful in 1996 and 1997, when the supercomputer, IBM Deep Blue, (IBM, 1996) beat Garry Kasparov in a pair of six-game chess matches. Deep Blue mimicked tactics by learning many human chess moves. Unfortunately, this approach to the problem is limited by the creativity and intuition of the human brain.

The AlphaGo Zero Project, developed a machine algorithm, to play the Chinese abstract strategy, 19 x 19 grid, board game, *Go*. This new version surpassed all previous versions of AlphaGo (including the Lee Sedol 9p player, version) in only forty days of training, entirely from self-play, and without human intervention or the use of human historical data (DEEPMIND, 2017). The technique used reinforcement learning, where a machine acts as its own trainer. As the algorithm plays, the computer tunes itself, creating a better and improved version of itself; thus, establishing cutting-edge artificial intelligence. Using this strategy, AlphaGo Zero showed unusual and non-trivial movements compared to humans, but they were still efficient and winning strategies. Scientists called this as machine intuition (Kohs, 2017).

This thesis takes the AlphaGo team's approach as inspiration and proposes a new method for the automatic generation of non-trivial appliance dataset mixtures, which will also create a new category of synthetic datasets. The method, detailed in Chapter 4, will permit training to be as long as required, and will create a voltage and current mixture by using individual real data samples from single appliances. Hence, a synthetic mixtures generator develops the capability to generate innumerable singular combinations.

Because of the potential importance of a widespread adoption of NILM and its positive consequences for society, we propose to move the research forward with various contributions. They are:

(I) A *BRazilian Appliance Dataset* (BRAD);

(II) A new method for creating synthetic dataset mixtures that uses voltage and current waveforms from individual appliances' only;

(III) A method for determining the minimum set of features (feature selection) and desired feature extraction rate, using three different disaggregation methods; and

(IV) The development of custom hardware for energy and features measurements.

This Chapter presents an overview of each of these contributions, explaining the methods used and referencing the chapters where they are detailed.

3.1.Overview of BRAD

In 2015, Adabi and his colleagues (Adabi, et al., 2015) postulated, "The required sampled training data period needed to achieve a certain high accuracy in pattern recognition for a specific algorithm decreases as the sampling rate increases, and increases with the number of devices whose signatures are in the data to be disaggregated". As seen in Section 2.8, algorithms' performance drops, when the number of aggregated appliances is increased.

Furthermore, using higher sampling rates produces key features and resources to distinguish overlapped appliance signatures. Zeifman's studies (Zeifman, et al., 2011)

showed how an ordinary home might include anywhere from thirty to fifty different appliances and these numbers generally increase each year.

Using a Yokogawa DL850EV ScopeCorder (YOKOGAWA, 2015), which was available at the LEMT Laboratory at COPPE/UFRJ, our first step was to acquire voltage and current signals, as illustrated in Figure 13.

This thesis adopted a sampling rate of 10 kHz. This choice was made based on the analysis from a Chinese study (Liu, et al., 2014). They considered Fourier harmonics up to the 20th adequate electric features for disambiguation, and suggested the use of a sampling rate of at least 8 kHz for feature extraction. The Yokogawa DL850EV ScopeCorder works with sampling frequency-steps of 5 kHz, thus, the lowest rate after 8 kHz was the rate used (10 kHz). Beyond that, Armel (Armel, et al., 2013) correlated sampling frequency intervals with the maximum number of distinguishable appliances, as listed in Table 2. Therefore, this suggests that using this sampling rate, the number of distinguishable appliances will be between 20 to 40 unique signatures.

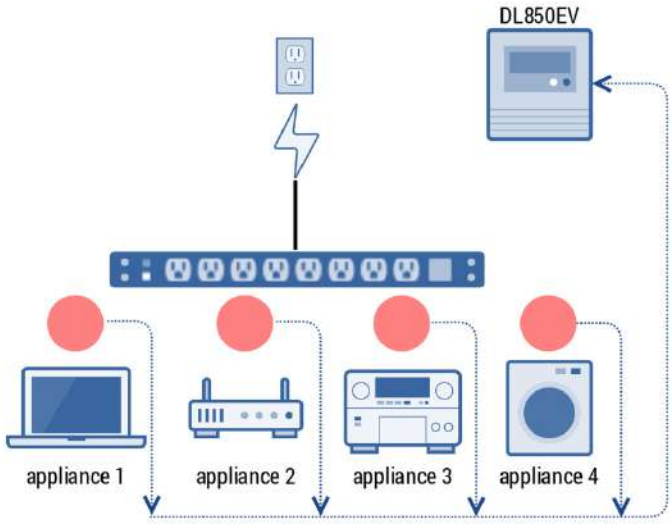

Figure 13 – Experiment's setup using Yokogawa DL850EV

We acquired up to four single-phase appliances at a time - the number was limited by the available current channels of the DL850EV - using the 127V_{rms} reference provided by the Rio de Janeiro energy distribution utility. A sample of waveforms is shown in Figure 14. It is important to highlight that the sampling procedure guaranteed that all activations of appliances, as states in a Finite State Machine, were recorded

entirely. The procedure forced all measured appliances into a comprehensive cycle for activating all power states.

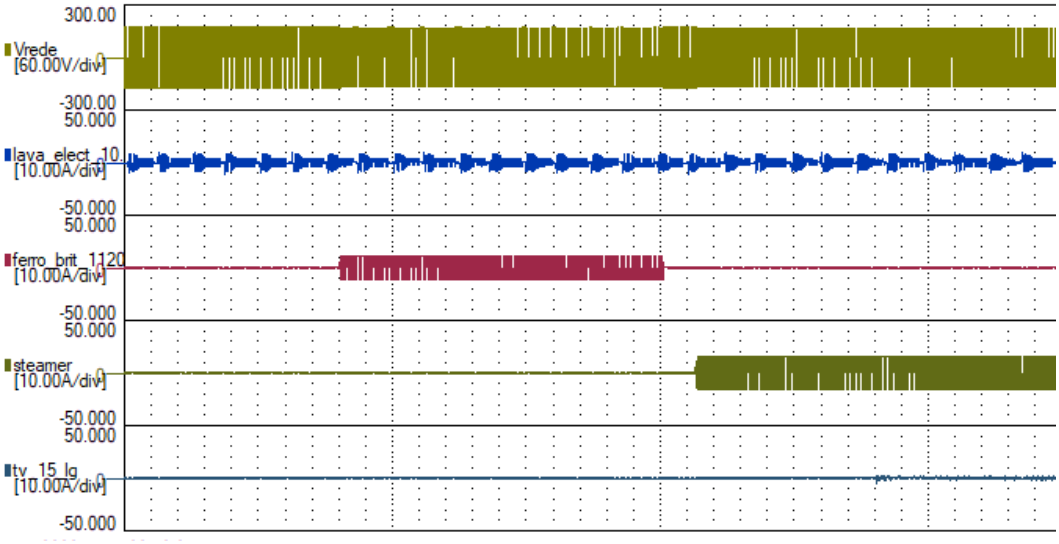

Figure 14 – Example of four individual current and voltage samples acquired by the DL850EV

The list of selected appliances followed directives obtained from the PROCEL research (PROCEL, 2005) on the most-used appliances in Brazil. Thus, we included at least one appliance from each residential electricity category as follows: **refrigerators**, contribute to 22% of power consumption in Brazilian residences, **air conditioning** (20%), **lightning systems** (14%), **televisions** (9%), **freezers** (5%), **sound systems** (3%), **irons** (3%) and **others** (24%), including computers (desktops and laptops), and printers. Hair dryers, microwave ovens, dishwashers, washers and dryers were also included in current dataset.

The availability of appliances was also a challenge, and we used several homes in Rio de Janeiro, Brazil, to achieve a total number of 94 appliances. The sampling of individual appliances was done randomly, and was limited by the maximum current of the selected transducer evaluated with DL850EV (100A @ $127V_{rms}$). The duration of each measurement was done based on each complete appliance power cycle; taking from a few minutes (light bulbs, for example, turned **on** and **off** at least three times) up to 6h20m (a washer with integrated dryer through a complete activation). The complete list of acquired appliances is shown in Table 5. Details about the adopted file format and metadata are described in Appendix A.

Table 5 – List of the acquired waveforms of 94 different appliances

Appliance Code	Description	Category
hd_seagate_4tb	External Hard drive	Office and Computer
Ibebedouro	Drinking fountain	Office and Computer
Icafe	Coffee maker	Office and Computer
mon_philips23led	FullHD LED Monitor	Office and Computer
monitor_lg_29_wd	FullHD UW LED Monitor	Office and Computer
monitor_s_23lcd	FullHD LCD Monitor	Office and Computer
net_digital_sd	SD Cable TV Tuner	Office and Computer
net_hd_digital	HD Cable TV Tuner	Office and Computer
notebook_14_vaio	Notebook Corei5 14LED	Office and Computer
pc_6cor_gtx960	PC with GPU	Office and Computer
pc_amd_apu	PC with APU	Office and Computer
xerox_laser_mf	Laser Printer	Office and Computer
air_fryer_ford	Kitchen air fryer	AC, Home & Kitchen
ar_consul7500btu	Window air conditioning	AC, Home & Kitchen
aspirador_po	Vacuum Cleaner	AC, Home & Kitchen
coifa_electr	Exhaust hood	AC, Home & Kitchen
ferro_brit_1120w	Iron	AC, Home & Kitchen
filtro_gela_colr	Water filter with cooling	AC, Home & Kitchen
fogao_dako_5boca	Gas stove with electric starter	AC, Home & Kitchen
fogao_eletrico	Electrical stove	AC, Home & Kitchen
foreman_grill	Kitchen Grill	AC, Home & Kitchen
freezer_consul	Freezer	AC, Home & Kitchen
frigobar_consul	Small Fridge	AC, Home & Kitchen
gas_rinnai	Water heater	AC, Home & Kitchen
geladeira_electr	Fridge	AC, Home & Kitchen
grill_ford	Kitchen Grill	AC, Home & Kitchen
Igeladeira	Fridge	AC, Home & Kitchen
iogurteira	Yogurt maker	AC, Home & Kitchen
lava_elect_10.5k	Washing Machine	AC, Home & Kitchen
lavalouca_braste	Dishwasher	AC, Home & Kitchen
lavaseca_10ksams	Washing Machine with dryer	AC, Home & Kitchen
liquid_osterizer	Blender	AC, Home & Kitchen
microondas_lg	Microwave over	AC, Home & Kitchen
microondas_pana	Microwave over	AC, Home & Kitchen
panela_walita	Electrical pan	AC, Home & Kitchen
pipoqueira	Popcorn maker	AC, Home & Kitchen
secador_taiffrs3	Hair dryer	AC, Home & Kitchen
steamer	Steamer iron	AC, Home & Kitchen
waffle_maker	Waffle machine	AC, Home & Kitchen
walita_processad	Food processor	AC, Home & Kitchen
fluo_W_osram_!5W	Fluorescent lamp bulb	Lightning
fluo_W_tasch_13w	Fluorescent lamp bulb	Lightning
fluo_Y_18w_osram	Fluorescent lamp bulb	Lightning
fluo_Y_kian_15W	Fluorescent lamp bulb	Lightning
fluo_Y_light_9W	Fluorescent lamp bulb	Lightning
fluo_Y_osram_13w	Fluorescent lamp bulb	Lightning
fluo_Y_osram15w	Fluorescent lamp bulb	Lightning

Appliance Code	Description	Category
fluor_Y_flc_23W	Fluorescent lamp bulb	Lightning
halo_tasc_40w_cb	Incandescent lamp bulb	Lightning
led_grow_28wrgb	LED lamp bulb	Lightning
led_rzd80_5W	LED lamp bulb	Lightning
led_Smart_7w	LED lamp bulb	Lightning
led_W_ecp_9w	LED lamp bulb	Lightning
led_Y_foxlux_10w	LED lamp bulb	Lightning
led_Y_galax_4W	LED lamp bulb	Lightning
osram_11w_fluor	Fluorescent lamp bulb	Lightning
osram_5.5w_led	LED lamp bulb	Lightning
osram_60w_halog	Incandescent lamp bulb	Lightning
osram_7w_led	LED lamp bulb	Lightning
osram_halogen_70	Halogen lamp bulb	Lightning
amazon_echo_dot	Echo DOT home AI system	Home Theater and Network
android_mp	Android Media Player	Home Theater and Network
harmonyhub	Logitech Harmony hub	Home Theater and Network
ht_bluray_samsun	Home-theater all-in one	Home Theater and Network
modem_net_aires	Internet Cable Modem	Home Theater and Network
rec_denon_1403	Audio Receiver	Home Theater and Network
som_2.1_logitech	Audio system	Home Theater and Network
tp_link_850re	Wi-Fi repeater	Home Theater and Network
tplink_archerc59	Wi-Fi router	Home Theater and Network
tv_14_lg	Small CRT Television	Home Theater and Network
tv_15_lg	Small Flat CRT Television	Home Theater and Network
tv_50_plasma_pan	Plasma TV	Home Theater and Network
TV_LG_50	LED TV	Home Theater and Network
xbox_one	X-box one	Home Theater and Network
barb_philips_qcs	Electric Shaver	Others
bose_mini2	Bluetooth sound system	Others
chrg_dash_one+	Cell-phone charger	Others
chrg_hoverB84W	Hoverboard charger	Others
chrg_ipad10w	Cell-phone charger	Others
chrg_iphone5w	Cell-phone charger	Others
chrg_lg_8W	Cell-phone charger	Others
chrg_orico40w	Cell-phone charger	Others
chrg_samsung10w	Cell-phone charger	Others
chrg_xiaomi_10w	Cell-phone charger	Others
f38Wdh	Cell-phone charger	Others
fragment_aurora	Paper shredder	Others
furadeira_350w_b	Drilling Machine	Others
furadeira_port	Drilling Machine charger	Others
powersave_outlet	Smart outlet meter	Others
sonnoff_1ch	Smart relay	Others
ssocket_eekbes	Smart outlet controller	Others
ssocketSWA1_bivt	Smart outlet controller	Others
vent_teto_3vel	Ceiling fan	Others
vent_teto_spirit	Ceiling fan	Others

3.2.Overview of synthetic mixture generator

Using the individual appliances samples, we propose a method to mix individual measurements that will give an aggregated stream of synthetic data waveforms similar to those sampled in a real-life scenario. In addition, the proposed method is capable of controlling whether appliances are **on** and **off** in an environment and also the appearance times for each sampled domestic or industrial device. As previously stated in this thesis, the creation of a broad and comprehensive data stream is essential for successfully training large classification models. The proposed sequence for creating synthetic appliance mixtures is illustrated in Figure 15 and comprises four steps to prepare data for processing. The first step, sampling individual appliances, has already been described in Section 3.1, with the creation of BRAD.

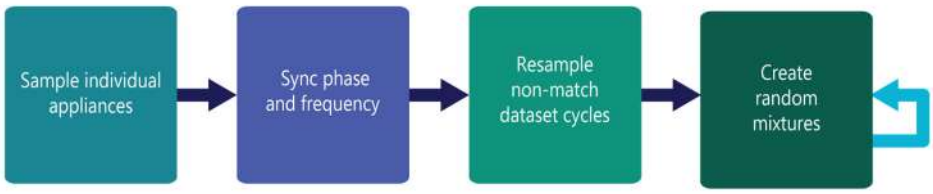

Figure 15 – Sequence for creation of synthetic data to different machine learning algorithms

The next step of the proposed method concerns the synchronization of the phase and frequency of the voltage and current for all appliances. This is done to guarantee that, for all cycles of voltage and current, (1) all sampled cycles will have exactly the same number of samples; thus, forcing cycles to have the same oscillating frequency $\omega(t)$; and (2) all cycles must be in-phase $\phi(t)$. The combination of these two premises enables us to merge individual current waveforms by simply manipulating the sampled data.

Frequency and phase are interrelated by:

$$\omega(t) = \frac{d\phi(t)}{dt} \quad \textbf{(3.2.1)}$$

$$\phi(t) = \phi(0) + \int_0^t \omega(t')d(t') \quad \textbf{(3.2.2)}$$

where $\omega(t)$ is the angular frequency and $\phi(t)$ is the phase of the fundamental frequency in a sinusoidal signal.

The purpose of *a phased-locked loop* (PLL) is to replicate and track the frequency and phase presented at input, at its output, and preferably without offset, allowing one oscillator to track with another. In this case, a complete feedback system includes a VCO at output, a phase detector and a low-pass loop filter (UCSB, 2011).

In order to measure the frequency and phase of voltage channels and, consequently, the current channel, we proposed using a *PLL* algorithm, which would provide the phase $\phi(t)$ and frequency $\omega(t)$ of the original signals. The general scheme for a PLL is illustrated in Figure 16.

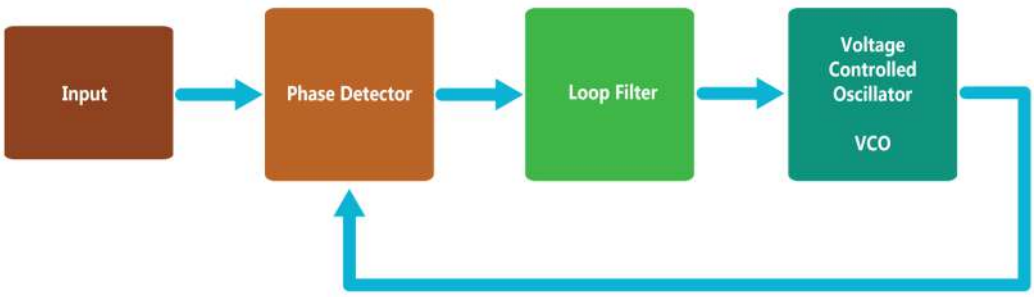

Figure 16 – Description of a *phased-locked loop* (PLL) general scheme for tracking and locking frequency and phase

The synchronization process provided phase $\phi(t)$, as a sawtooth waveform, as illustrated in Figure 17, and, as a result, the number of samples per acquired cycle. We evaluated that each cycle of voltage and current would not present the same number of samples, because the sampling rate (10 kHz) and voltage fundamental frequency (60 Hz) were not multiples. The ratio between both frequencies was, in most steady state measurements (excluding starting and ending sampled data), a number between 166 and 167 data points. Thus, it is proposed that, for each cycle which does not equal 167 points, a resample be made to achieve a new simulated sampling rate of 10,02kHz, thus, forcing all the cycles to equal 167 points.

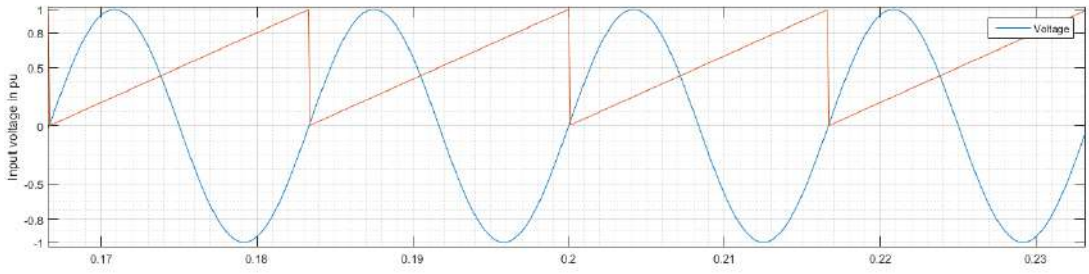

Figure 17 – Example of a PLL phase $\phi(t)$ lock as a normalized sawtooth waveform

Now, with all the waveforms resampled and synchronized (see further details in Sections 4.6 and 4.7) new mixture scenarios can be developed synthetically. This is achieved by manipulating individual appliances' current channels as an aggregate - the last phase of proposed method. Using Kirchhoff's circuit laws, and the superposition principle, it is possible to infer that, for linear systems, an aggregate of two or more currents, flowing into a node, could be represented by the sum of each current separately.

Therefore, using previously synched, resampled appliances' waveforms, each current cycle could be merged randomly; thus, creating a synthetic current mixture. In addition, we propose to select the main voltage channel using the longest duration sampled voltage from the set of appliances. Synthetic datasets could perform sets of appliances with or without repetition, by combining M by N appliances as needed (with M being the number of appliances to be combined and N being the minimum number of each appliance's appearance, proportioned over all datasets).

Therefore, we used ten non-repeated appliances per synthetic generated data stream mixture; so, as result, each appliance appeared at least ten times in 93 generated datasets, using a "branch and cut" optimization algorithm. The full description and results of creating random mixtures is detailed in Section 4.7.

3.3.Relevant Features

The purpose of NILM is to disaggregate energy. Therefore, for each individual appliance, active power (P) was computed as a main feature. In 2012, Pattem (Pattem, 2012) showed that active power (P), reactive power (Q) and Fourier harmonic components were sufficient to detect most appliances (~90%). Unfortunately,

researchers didn't define exactly which set of features was capable of disaggregating which category of the appliances for each NILM method tested.

Therefore, based on the preceding chapter's examination of scientific papers concerning the state of the art of NILM, we compiled a list of the more prevalent features for both aggregate and ground-truth data. The listed features are described in Table 6.

Table 6 – Thirty selected features for offline computation

Feature	Representation or Unit	Extraction Rate
I_{RMS}	A	16.67ms to 15s
Crest Factor	Dimensionless	16.67ms to 15s
Active Power	W	16.67ms to 15s
Reactive Power	VAr	16.67ms to 15s
Apparent Power	VA	16.67ms to 15s
Non-Normalized Fourier Transform Spectrum *up to the 25th harmonic, including fundamental*	One value for each 60Hz harmonic	16.67ms to 15s

An example of the five features (root mean square (rms) voltage - V, root mean square current - I, active power - P, apparent power - S, reactive power - Q) over feature measured samples (x axis) is shown in Figure 18, and is measured from a LED light bulb.

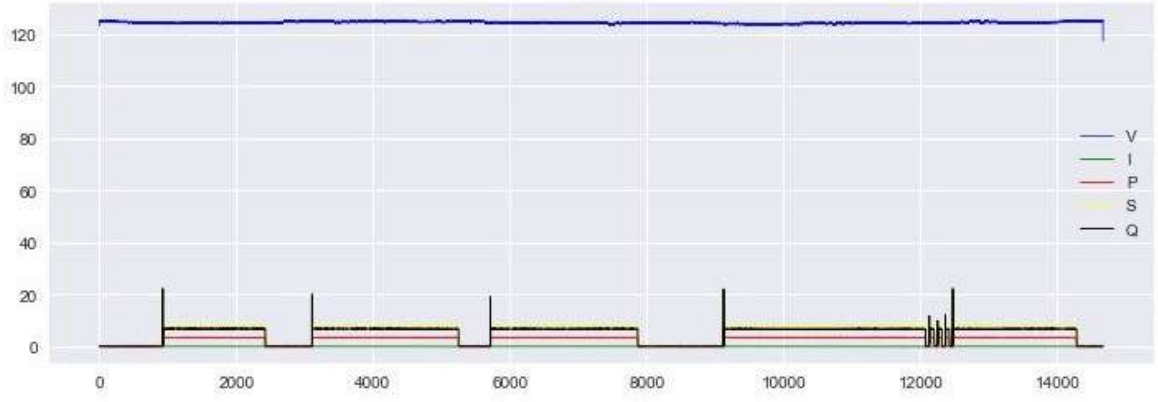

Figure 18 – Example of a set of features from a 5.5W LED light bulb

The algorithms that computed features were previously tested using synthetic sinusoids which generated different in-phase harmonics' components, with assorted

amplitudes, in order to guarantee correct feature extraction for each appliance. A sample of a synthetic Fourier spectrum analysis is shown in Figure 19. It has a sinusoid with 90% of the 2nd harmonic, 70% of the 3rd harmonic, 50% of the 5th harmonic, 30% of the 7th harmonic, 20% of the 9th harmonic and 10% of the 11th harmonic. We applied Zero padding and a Blackman window for the Fourier analysis to provide a better resolution of the sampled data. The software was developed in Python, using auxiliary libraries to calculate features as NumPy 1.14.0 (NUMPY, 2017) and ScyPy 1.0.0 (SCYPY, 2018).

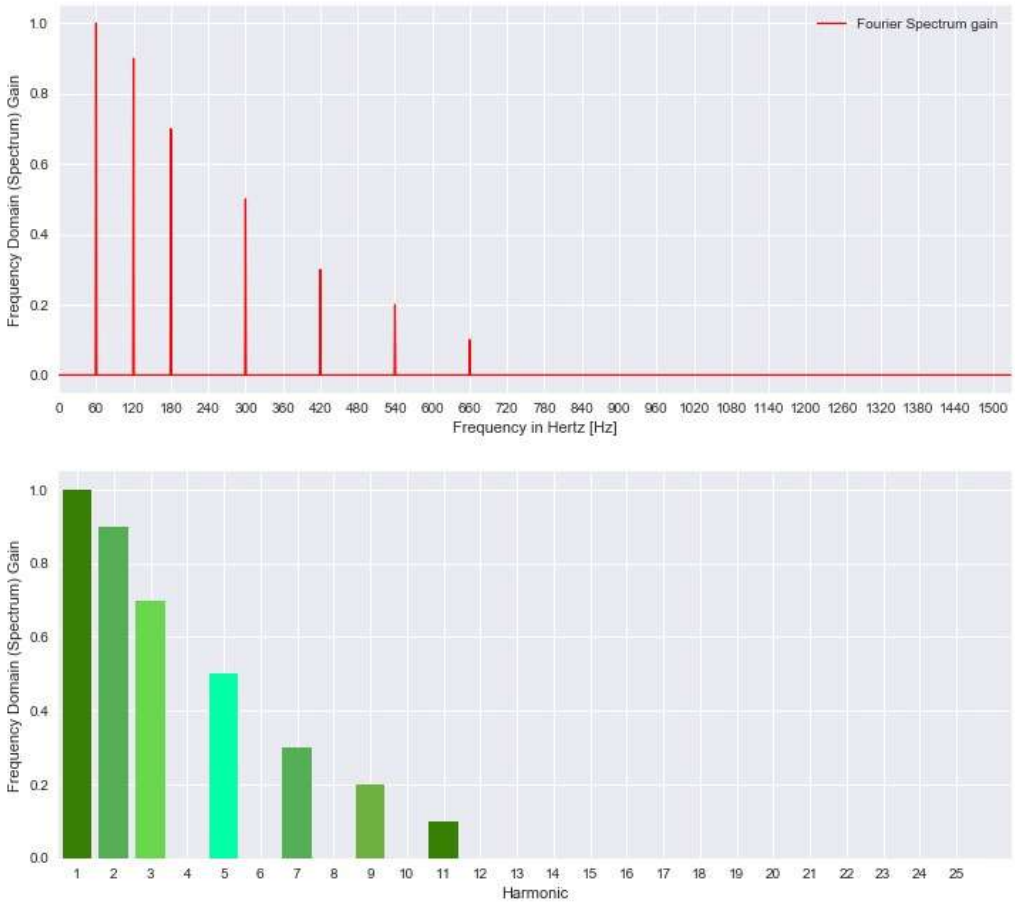

Figure 19 – Example of a Fourier transform calculation using a synthetic sinusoid with pre-defined harmonic spectrum content for testing

Note that the individual 10 kHz data samples from the voltage and current waveforms were not used by this thesis. The raw data was considered inadequate for streaming over actual smart meter communication channels' bandwidth. However, manipulation of waveforms could be used in future works, to generate more features and to exploit more advanced and distinctive variables.

3.4. Overview of feature extraction rate and feature selection

As noted in Chapter 1, NILM platforms have the potential to improve energy efficiency in most residential scenarios. Unfortunately, the use of smart meters requires a great deal of new telecommunications infrastructure and storage capacity, as illustrated on Figure 20. Utilities currently use remote smart metering, in order to avoid field services, for example to remove the need for manual data collection from power meters.

Figure 20 – Actors involved in the smart metering process

For the utilities, each user that sends data to a datacenter represents a pre-defined cost for the telecommunications channel and the storage of historical data. However, billions of energy users in every big city around the globe would need to be attached to a smart grid infrastructure. Consequently, reducing the amount of generated data to be transmitted and, consequently, processed and stored becomes a binding premise for the success of load disaggregation technologies mass adoption. With this in mind, we propose the development of a method to reduce the amount of data transferred, by answering two questions:

Q1. Given a set of calculated features, what is the minimum *feature extraction rate* required, to preserve similar levels of disaggregation performance?

Q2. Given a defined feature extraction rate, what is the minimum set of electrical features, obtained from the aggregated household data, which will maintain equivalent levels of accuracy as the maximum extraction rate?

As described in Section 2.8, we can infer, empirically that as the complexity of the aggregated data increases, NILM's methods obtain a poorer performance in metrics. Therefore, we expect the same behavior when data transfer is decreased, by dropping the feature extraction rate, or the number of extracted features, as shown in Figure 21.

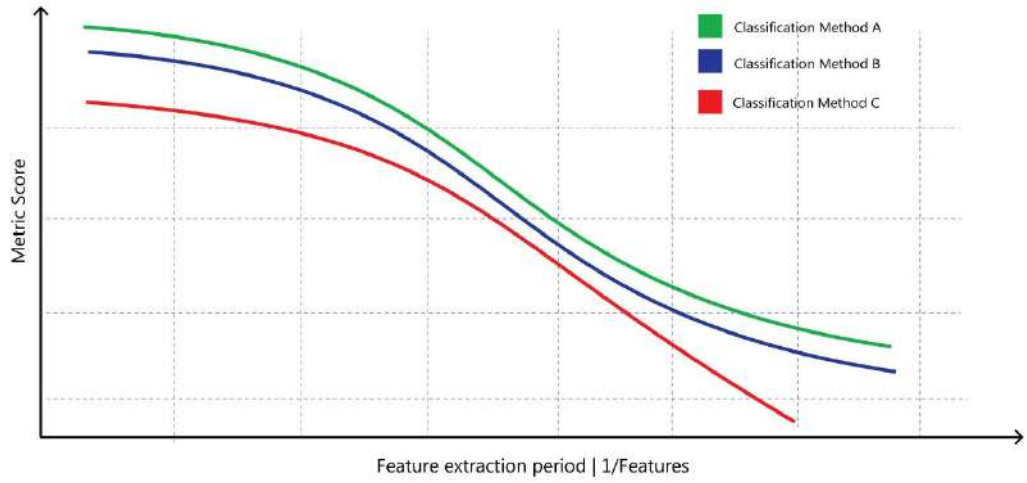

Figure 21 – Expectation for performance of NILM by the reduction of data transfer

In order to evaluate NILM's performance, we used six metrics (as listed in Section 2.8); each one defined as a *Metric Score*. In our quest to find an optimum combination of performance and feasible computing time, we also included two more metrics to assist in the evaluation of the problem: (1) *Fit time* - the required time for the training process of the NILM problem solving algorithm and, (2) Predict time - the time required to formulate predictions about load disaggregation. Finally, eight *Metric Scores* were used to evaluate performance:

I. **Total energy correctly assigned (TECA)**
II. **Normalized error in assigned power (NEAP)**
III. **Precision (P)**
IV. **Recall (R)**
V. **Accuracy (A)**
VI. **F-measure / F-score (F1)**
VII. **Fit Time (training)**
VIII. **Predict Time**

To find the desired *minimum feature extraction rate*, we propose to make a comprehensive survey over an interval of extraction rates, measuring all eight *Metric*

Scores; thereafter, to discover when performance drops below a chosen set point. In order to achieve this, the first step of proposed method is to identify the maximum value for each metric displayed by a NILM classification algorithm. As previously stated, we expect that the maximum disaggregation performance will be achieved using the *maximum feature extraction rate*.

As described in Section 3.3, we adopted a sampling rate of 10 kHz for both voltage and current channels. After a consideration of the premises of earlier researches (see Chapter 2 of this thesis) concerned with feature extraction rate selection, we defined rates empirically, in an interval between 60Hz (period of 16.67ms) and 0.067Hz (period of 15s) in order to evaluate our results.

The seventy one (71) pre-defined period intervals, listed as feature extraction rates, are displayed in Table 7. For simplification, we adopted a specific notation for feature extraction rates by numerically counting the number of 60Hz cycles. For example, **one (1) cycle** rate represents a 16.67ms period, and **nine hundred (900) cycles** rate**,** is a 15s period, all referenced to a 60Hz voltage frequency.

Table 7 – The pre-defined feature extraction rates intervals

Period start	Period end	Period intervals
From 1 cycle (16.667ms)	to 60 cycles (1.000s)	60 bins (1, 2, …, 60 cycles)
From 61 cycles (1.167s)	to 100 cycles (1,667s)	4 bins (70, 80, 90, 100 cycles)
From 101 cycles (1.683s)	to 149 cycles (2,483s)	1 bin (120 cycles)
From 150 cycles (2.500s)	to 300 cycles (5.000s)	4 bins (150, 200, 250, 300 cycles)
From 301 cycles (2.517s)	to 900 cycles (15.000s)	2 bins (600, 900 cycles)

This means that, using the full set of 30 features (as detailed in Section 3.3) to train and predict load disaggregation with synthetic datasets, the proposed method will plot each *Metric Score* alongside the feature extraction rate interval. We propose to establish an *acceptable metric score* value as a set point for each *best metric score*, which we expect to be at the *maximum feature extraction rate* (defined as a number of 60Hz cycles). Thus, whereas the feature extraction rate (displayed in cycles) advances, and

each metric score starts to fall, the *minimum feature extraction rate* is determined when *acceptable metric score* is achieved. We used 5% less than the *best metric score* as a heuristic premise value for the *acceptable metric score*. Figure 22 shows our proposed scheme to find the *minimum feature extraction rate*.

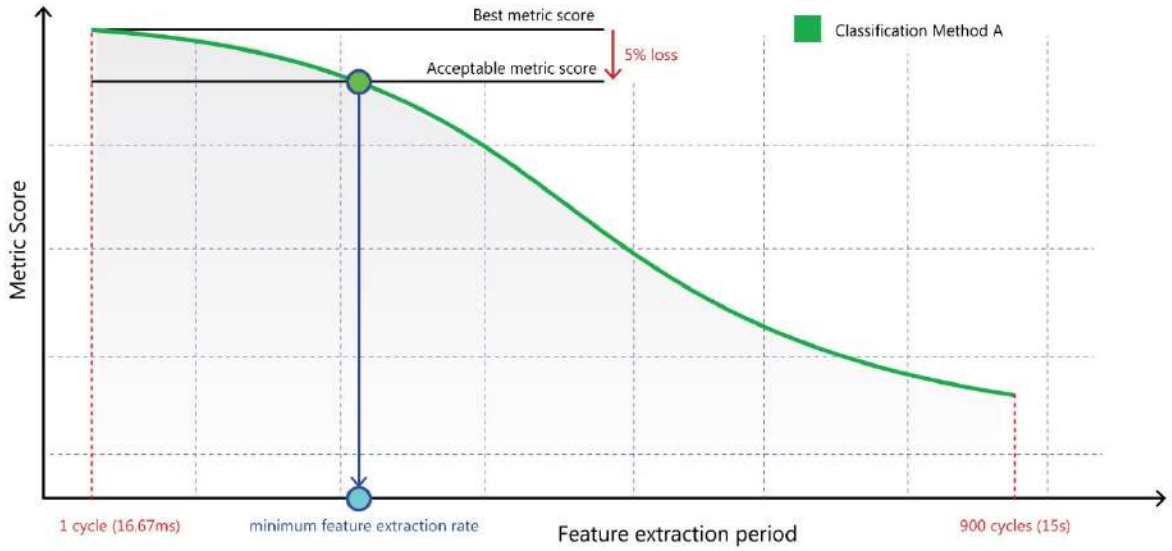

Figure 22 – Scheme for finding the *minimum feature extraction rate* for an appliance

Furthermore; more than just reducing the feature extraction rate, a smart selection of a set of features also guarantees that any data that does not need to be transferred and stored is expunged. This is a useful point, given that redundant or irrelevant features could potentially complicate computational models, extend training and predicting times, and reduce generalization (Bermingham, et al., 2015).

Feature selection algorithms are usually categorized as (1) **filter methods**: only the intrinsic correlation between the properties of data are scored, thus, features with low scores are removed; (2) **wrapper methods**: different sets of features are evaluated, coupled with a computational model (for regression or classification); and, (3) **embedded methods**: a feature's importance is obtained directly by a training model (Huijskens, 2017).

Feature selection has been studied extensively and several algorithms have already been proposed to realize an effective selection of discriminant features. In an attempt to improve this analysis, we started using three different, widely-used algorithms for a further analysis of their performance in the feature selection process.

Apart from NILM, the **AdaBoost algorithm** (Freund, et al., 1997) was used in previous research (Silapachote, et al., 2005) for feature selection within the problem area of facial recognition. Another work (Zhou, et al., 2007) used AdaBoost for feature selection in acoustic event detection, to select features that outperformed classical speech feature sets. Wang and his colleagues (Wang, 2012) also mentioned AdaBoost as a widely used algorithm, which offered a promising performance for feature selection.

Other researchers proposed a multi-class exponential loss function with low computational cost (Zhu, et al., 2006) as a method of extending the original AdaBoost algorithm (Freund, et al., 1997) in order to solve multi-class classification problems. The implemented algorithm, detailed in their work (Zhu, et al., 2006), was called AdaBoost *'Stagewise Additive Modeling using a Multi-class Exponential'* (SAMME) loss function. After examining the studies mentioned above, and previously, we selected multi-class AdaBoost SAMME as the first algorithm to be analyzed.

Mutual Information (Kozachenko, et al., 1987) measures the mutual dependence, or the amount of information, between two random variables. This concept is linked to the entropy of a variable, an important subject in the Information Theory field, and has been detailed in Ross's work (Ross, 2014). Cang and his colleagues (Cang, et al., 2012) used feature selection, using Mutual Information, in three experimental classification problems, finding that it was an efficient, robust and stable method. In 2014, researchers (Vergara, et al., 2014) presented all of the successful, state-of-the-art feature selection methods based on mutual information field. We have also chosen to use Mutual Information for feature selection.

Within the subject field of NILM, one study on feature selection (Sadeghianpourhamami, et al., 2017) proposed using a Random Forest (Ho, 1995) algorithm as the main classification model. Similar algorithms, such as **Extremely Randomized Trees** (Extra-Trees) (Geurts, et al., 2006), have also been used successfully for feature selection in other subjects, such as polarimetric synthetic-aperture radar (SAR) image classification (Zou, et al., 2009), seamounts classification derived from bathymetry data (Lawson, et al., 2017), protein disorder prediction (Smith, et al., 2018), and has even proved to be efficient with noise data. This has been therefore selected as the last algorithm to be evaluated.

We propose a method based on recursive feature elimination (RFE) (Guyon, et al., 2002), to determine which minimum sets of features are strongly correlated. The proposal begins by constructing a classification model using all the features and evaluating its performance using the (previously listed) eight *Metric Scores*. The second proposed step of the method will use a greedy solution, to rank feature importance using the three feature selection algorithms: **mutual information** (Ross, 2014); a **multi-class AdaBoost** (Zhu, et al., 2006) and **extra-trees** (Geurts, et al., 2006).

After a supervised analysis of the generated ranking, all features with an importance below an empirically pre-defined parameter will be expunged from the main set of features. It should be noted that the proposed feature selection is a heuristic procedure which does not corresponds to the best feature set. However, the use of this procedure is intentional in that it allows each high-correlated feature to appear more than once in a set.

From the three different sets, obtained from the feature selection algorithms, we propose to continue on to a validation of the sets, for different NILM solving methods; checking for robustness, accuracy and stability of the disaggregation process.

In line with a comparison of previous studies, published by the Imperial College (Kelly, et al., 2015) and COPPE/UFRJ (Nascimento, 2016), our validation of the disaggregation methods will not use algorithms to solve regression problems (reconstructing the output waveform). Instead, the focus will be on classification algorithms, using quantized pre-clustered power outputs, as each appliance is declared as a finite state machine (FSM). As Nascimento pointed out, in his dissertation (Nascimento, 2016), models, where each state was associated with a power value given by distribution, appeared to have better performance, for inferring and correctly classifying estimated individual energy contributions, than regression algorithms.

3.5. Overview of quantization of active power

Logically, the disaggregation problem in NILM is a regression problem. This means our development must:

i. Reconstruct active power demand,

ii. Correctly assign power consumption, and

iii. Establish turn **on** and **off** moments of each appliance.

However, regression problems are usually difficult to develop in machine learning systems, for the most traditional L2 metric criterion, as they suffer from the presence of outliers and this can raise the importance of meaningless features. For this reason, this thesis proposes to manage NILM problems using a different approach. We propose quantizing active power outputs from appliances, using an unsupervised clustering algorithm. An example of quantized active power (P) output is illustrated in Figure 23. In this manner, the algorithm could automatically determine the clusters values, associating their active power (P) values as a state of a FSM for each appliance. In this way we could manage the NILM problem as a classification problem.

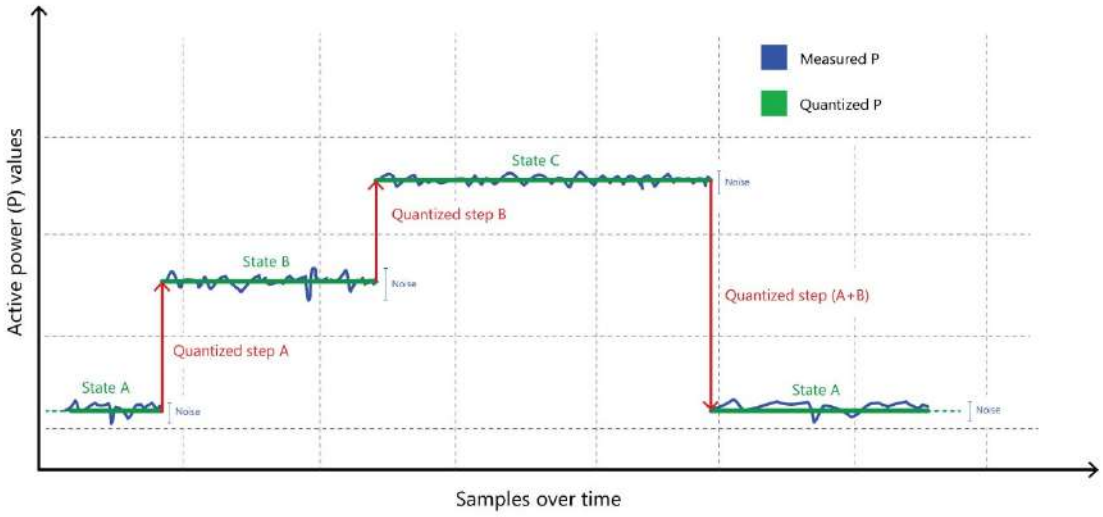

Figure 23 – Quantization of active power (P) by clustering near measurements in a fixed state

Clustering algorithms, with various implementations, are widely available in scientific papers. Out of the possible available algorithms in literature, we found that the approach of *Density-based spatial clustering of applications with noise* (DBSCAN) (Ester, et al., 1996), that was implemented by the *Hierarchical Density-Based Spatial Clustering of Applications with Noise* (HDBSCAN) library (McInnes, et al., 2017), presents good performance and scaling for massive datasets implementation on Python.

The quantization of active power, with detailed parameterization, which will be described in Section 5.1, creates predefined levels (quantization) in order to reduce

number of possible outputs for all appliances, from two (2) to six (6) cluster bins. An example of both clustered and original active power is shown in Figure 24. The green line is the output of HDBSCAN algorithm, as detailed in the HDBSCAN library documentation (McInnes, et al., 2017).

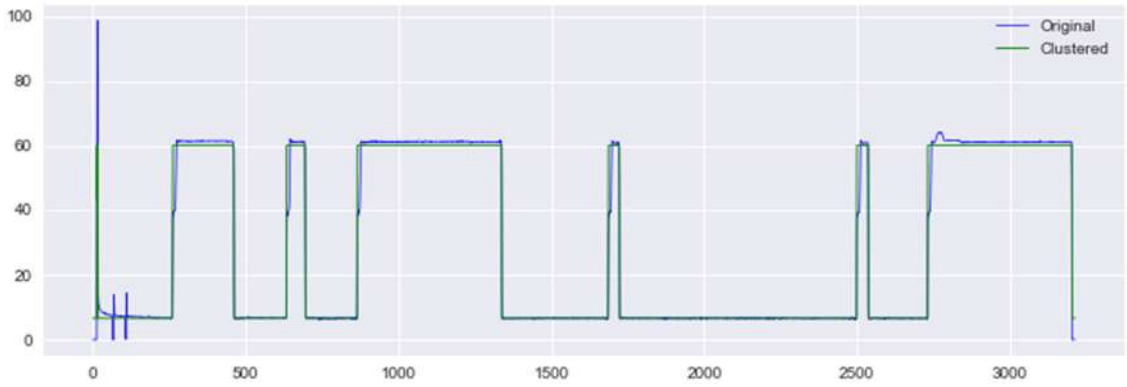

Figure 24 – Stacked original active power (blue) and clustered output (green) of a 15 inch TV alongside measured samples in x axis

3.6. Overview of classification algorithms

Outside of the previously stated algorithms for choosing the set of features, the list of classification methods is extensive. Moreover, the correct energy classification of individual appliances on NILM is a finely-tuned exhaustive recursive development. Each algorithm requires multiple fine-tunings of its parameters and different levels of computational resources.

After feature extraction of the mixtures, and the quantization of the sampled individual appliance active power (P), called 'ground truth' data, the next step of this study is to train machine learning models to predict load disaggregation, as illustrated in Figure 25. Therefore, we propose to evaluate three different supervised classification algorithms, thereafter providing **three trained models for each appliance-feature set**, in order to complete a comprehensive comparison of the results.

Figure 25 – Training with appliance and mixture voltage and current samples

The input for each training model is the normalized features set, which standardizes features by removing the mean and scaling to unit variance; thus, forcing all features to have variance of the same order and to be centered on zero. The quantized active power (P), known as 'ground truth' data, is used as the output to be fitted. The proposed architecture for disaggregating appliances is shown in Figure 26, where the stream of a normalized feature sets is multiplexed for several pre-trained classifiers.

For clarification, the proposed method provides twelve different trained models, for each appliance, one for each of the three classification algorithms, by each one of the four features sets: (1) using a full set of 30 features; (2) using mutual information (Ross, 2014), (3) using multi-class AdaBoost (Zhu, et al., 2006), and (4) using extra-trees (Geurts, et al., 2006).

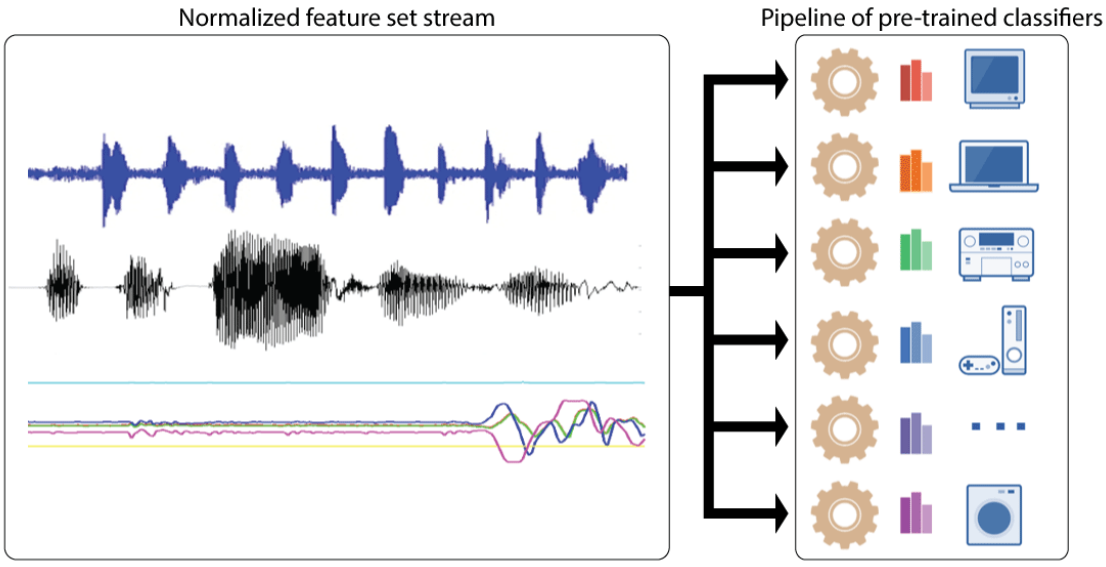

Figure 26 – Proposed scheme for individual classification for each appliance as a pipeline

Logistic Regression was first introduced by Berkson (Berkson, 1944) in 1944, and has been widely used in social and biomedical sciences (Allison, 2015) since then. Despite its name, Logistic Regression, or the logit model, estimates probabilities between independent variables with pre-trained categories of dependent variables. Multi-class logistic classifier evaluates a *Softmax* function that turns scores, or logits, into proper predictions probabilities for each output class. Details about the algorithm are explained in Kleinbaum and Klein's book (Kleinbaum, et al., 2010).

Outside the field of NILM, Logistic Regression has been widely used for subjects from statistical models in epidemiology (Clayton, et al., 1993) to language recognition (Leeuwen, et al., 2006). Two articles (Shulga, 2018; Allison, 2015) reinforced the importance of Logistic Regression as an important model to be evaluated. Thus, the first classification algorithm to be tested by this thesis is Logistic Regression, and uses the implementation described by Freedman (Freedman, 2009).

Support Vector Machines (SVM), described in detail by Campbell (Campbell C., 2011), have been previously used for financial time series forecasting (Tay, et al., 2001), text categorization (Joachims, 1998), gene selection for cancer classification (Guyon, et al., 2002) and other machine learning applications. Formally, classification tasks from SVM constructs a set of hyperplanes (or a unique hyperplane) in high-dimensional space (or infinite dimensional space) in order to separate the nearest training data points into different classes. Due to its prominence as a classification algorithm this study selected SVM as the second classification algorithm to be evaluated. The implementation used by this thesis, Support Vector Machines with Support Vector Classification (SVC), is detailed in LIBSVM (Chang, et al., 2013).

Deep Learning, a metaphoric expression, is an extension of Artificial Neural Networks, with improvements having been made to create deeper and larger artificial neurons connections. Its advanced mathematical models try to mimic the activity of the brain's neo-cortex, where approximately 80% of human thinking occurs (MIT Technology Review, 2013). Huge companies have applied Deep Learning to several hardware and software solutions; ranging from voice recognition (Amazon Alexa, Google Home and Apple Siri for example) to image recognition where, since 2015, algorithms (Vanhoucke, et al., 2015) have surpassed humans in fast image recognition

tests. Thus, the evaluation of a model that uses this technology is clearly promising as a potential solution to the disaggregation problem.

However, creating deeper neuron networks introduces many training problems, such as overfitting. Deep Learning concepts such as layers, activations, initializers, optimizers and regularization methods are summarized below in order to mitigate those related problems.

Dense Layer

Dense layer in Deep Neural Networks is a fully connected artificial feed-forward neural network. Neurons can contain linear or non-linear activations and, when combined, have the potential to create a sophisticated universal approximator.

Activations

Each neuron in an artificial neural network has an activation function, describing how it works intrinsically. Activations map non-linear or linear functions, using training weights and bias, in neurons connections. Several activation functions have been developed over the years, ranging from the simple linear function:

$y(x) = \alpha x$;

To a non-linear tanh:

$S(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$,

as shown in Figure 27 (a);

Passing by the rectified linear unit - ReLU (Nair, et al., 2010):

$f(x) = \max(0, x)$,

as shown in Figure 27 (b).

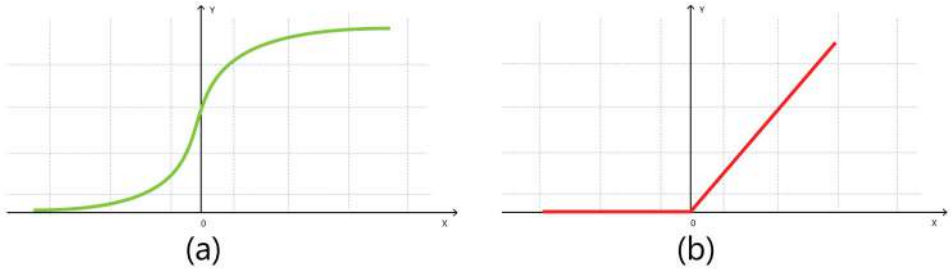

Figure 27 – (a) tanh or sigmoid shape activation and (b) rectified linear unit (ReLU)

Early stopping

The Oxford English Dictionary (OXFORD, 2018) defines overfitting as "*The production of an analysis which corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably*". In machine learning, overfitting impairs a trained model's ability to generalize about unseen datasets, by creating a model with more parameters than necessary.

For this reason, artificial neural networks and, consequently deep learning, developed multiple techniques to prevent overfitting. As shown in Figure 28, one efficient method is to discontinue training when the loss of validation dataset stops decreasing and starts to rise. At that specific moment, the model starts to memorize training data samples rather than being able to generalize and "understand" training.

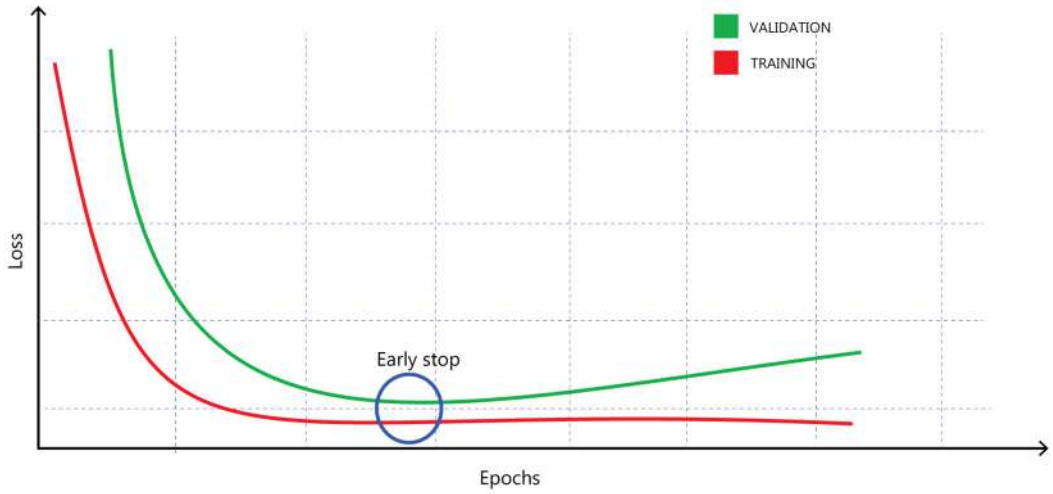

Figure 28 – Early stop moment

Dropout

Dropout (Srivastava, et al., 2014) is a technique to prevent overfitting in large and deep networks, where random activations between artificial neurons are dropped, preventing them from co-adapting too much. This algorithm promotes robustness in training, by developing a consensus of an ensemble, where different areas of a network are activated, resulting in redundant training, as shown in Figure 29.

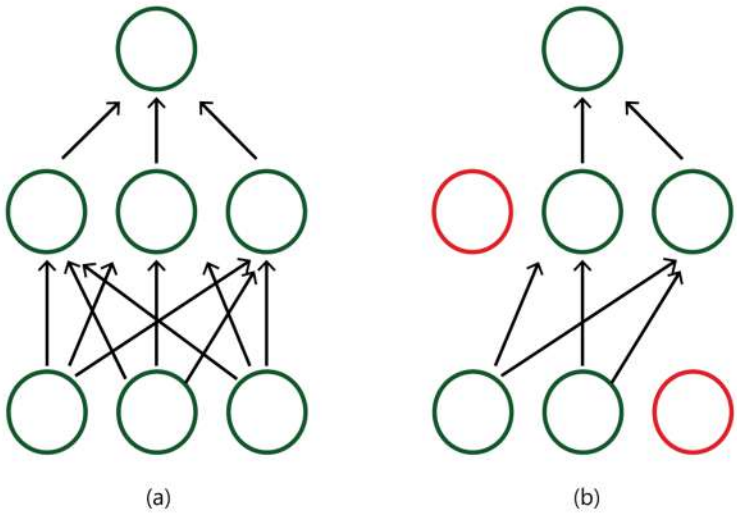

Figure 29 – (a) Fully connected neural network and (b) Dropout applied on neural model

Batch normalization

Training Deep Neural Networks is a time-consuming task. The majority of previous research into Deep Learning has used specialized hardware, such as graphics processing units (GPU) or tensor processing units (TPU), to accelerate the training process.

In the search for potential improvements to the software, a team from Google (Ioffe, et al., 2015) developed a technique, which they called Batch Normalization (BN) that reduces internal covariate shift by normalizing layer inputs. This technique achieved same accuracy with up to fourteen-times fewer training steps (Ioffe, et al., 2015) in an image recognition dataset.

Initializers

Each artificial neural network "synapse" must be initialized with values for weight and bias. Assorted distributions were created for different purposes, enabling designers to choose between then. However, each distinct distribution contributes differently to network convergence. In 2010, a work (Glorot, et al., 2010) presented the *Xavier normal initializer*.

Optimizers

Classic neural networks, calculate the gradient descent of an artificial neural network many times, in order to adjust weights and bias. The *Stochastic Gradient Descent* (SGD) optimizer, detailed in the work of Bottou and his colleagues (Bottou, et al., 2008), gets a computational advantage over classic gradient descent (GD) by calculating the gradient over only some parts of the entire dataset.

SGD has also improved on classic GD algorithms, in order to avoid a training problem that is known as "Local Minima". This is where it is possible to identify the gradients "getting stuck" at "Local Minima", as shown in Figure 30. It should be noted that the achievement of "Global Minima" is an ideal which is not usually achieved by the majority of algorithms. The Adam optimizer (Kingma, et al., 2014) has also shown promising results in minimizing classification errors.

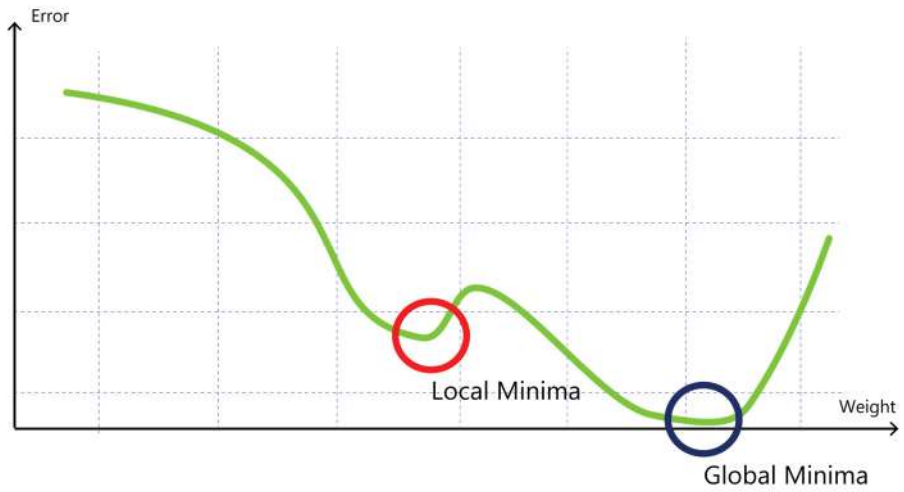

Figure 30 – Local Minima problem associated with "Back Propagation"

The process of discovering a necessary size for each layer, as well as the number of required layers, and which activations, initializers and optimizers or regularization methods your Deep Neural Network will request, is a recursive and thorough procedure. There is no scientific method that describes "how to" manage those parameters. For these reasons, our developed model was designed heuristically, by validating the related architectures, and by comparing similar complexity problems and, then, verifying the convergence of networks to a subsample of datasets recursively.

The initial neural network was very deep, and was based on previous Deep Learning NILM research from the Imperial College (Kelly, et al., 2015) and COPPE/UFRJ (Nascimento, 2016). After each test, we began to remove layers; to reduce and replace models whenever there was any significant loss in measured metric scores. It is important to note, here, that the reduced model used only nine fully connected (Dense) layers, without the advantage of more advanced deep layer architectures. However, we decided to maintain the definition of "Deep Learning" for our classifier, based on the definition that if a technique performs learning in more than two hidden neuron layers, it could be called 'deep'.

One Deep Neural Network model was evaluated for each electrical/electronic device. Our models used dense layers, using ReLU, Softmax and Linear activations; we also used a Batch normalization layer and Dropout layer for training. Early stopping was applied in the training phase, which was parameterized for a maximum number of 2560 epochs of training. Our tests also revealed that the *Xavier normal initializer* and Adam optimizer reduced training time on most of the appliances. The complete evaluated architecture is detailed in Section 5.3.3.

Input sampled data was treated as single data measurement, without any *Sliding Window* approach, or a *Curriculum Learning* strategy (Zaremba, et al., 2014). Dataset samples were randomly shuffled, keeping 10% hidden from training, as data for validating metrics, and 90% for training. The development of these non-linear classifiers, using Deep Learning techniques, was built with TensorFlow v1.5 (GOOGLE, 2018) library and KERAS high-level neural networks API (Chollet, et al., 2015).

3.7. Overview of smart meter development

After an analysis of feature selection and feature extraction rate methods, this research focused on the development of an NILM capable hardware, which was able to deliver a metering, storage and computational capacity to evaluate metrics, at a desired frequency, and to communicate with cloud servers. It is important to mention that this hardware can potentially help electricity customers to improve energy efficiency.

To achieve this, two different and interrelated requirements must be evaluated: cost of hardware and computational limits. Platform restrictions include power processing, memory, storage capacity, and communication technology. Newer technologies, offering more powerful data rate transfers, usually are more costly. In contrast, established and already mass-adopted technologies are more common and are usually cheaper.

In order to realize a powerful cost-effective solution for a large scale scenario, we decided to evaluate NILM algorithms on cloud servers. Local processing tasks should only include feature extraction and communication skills. Our proposal takes advantage of cloud computing (Kondo, et al., 2009) whereby shared resources, software and storage are provided over a network (*Internet of Things* - IoT paradigm).

Anticipating the architecture, detailed in Appendix B, Figure 31 shows the proposed platform. The NILM meter holds data in local storage (microSD card) until a connection from the user begins the download process. With Bluetooth technology (BLUETOOTH, 2016) users can use their own computers, smart phones or tablets to connect directly to the NILM meter wirelessly.

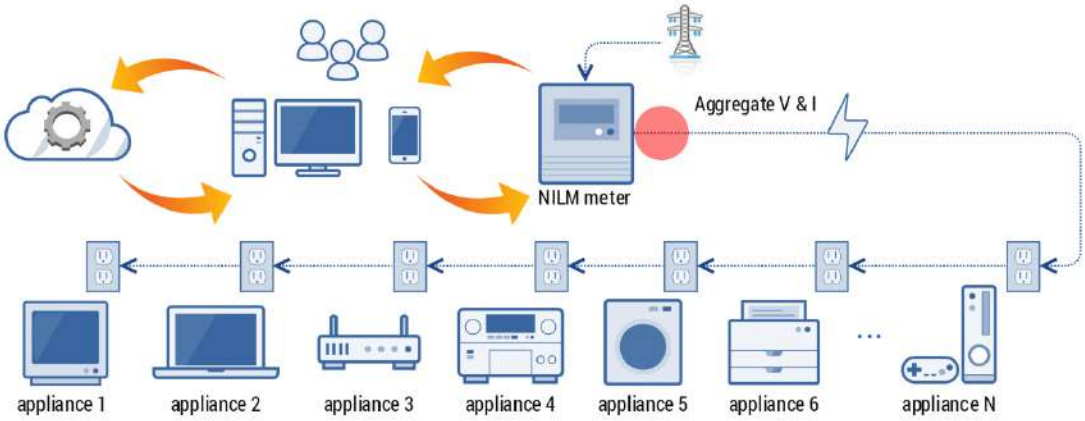

Figure 31 - Proposed architecture for NILM platform

The downloaded data from the meter is sent to cloud servers over the user's own internet connection. The processed data, containing the results of the load disaggregation, are then returned to the user's location to be shown as valuable information about their consumption. An additional Wi-Fi module could potentially be integrated into the platform, therefore avoiding users' need to download data manually, sending measurements directly to cloud servers.

4.Synthetic mixture generator

As previously described in Section 3.2, waveforms must be synchronized for the fitting and merging individual current waveforms. This implies that all cycles of each voltage and current channels must have the exact same number of samples.

In power electronic systems, *phased-locked loops* (PLLs) are widely used to synchronize active filters, *Flexible Alternating Current Transmission Systems* (FACTS) equipment and *Uninterruptible Power Supply* (UPS) systems. Numerous studies on power systems have applied the *Enhanced PLL* (EPLL) to industrial equipment (Karimi-Ghatemani, et al., 2004). Ziarani´s method for the extraction of nonstationary sinusoids, was adapted from control systems for power electronics, and in this thesis is called *Zirani's phased-locked loops* (ZPLL) (Ziarani, et al., 2004).

In an attempt to compare both PLL´s (EPLL and ZPLL) and to determine which is transiently safer (with faster locking and lower offset in frequency and phase), we propose an initial comparison of both algorithms.

4.1.PLL comparison for voltage phase synchronization

Our proposed *Phased-Locked Loop* (PLL) comparison aims to guarantee the stability of steady state and transient frequency and phase extraction, using four different tests:

1. *Voltage magnitude step response*
2. *Frequency step response*
3. *Voltage magnitude step response on presence of harmonics*
4. *Frequency step response on presence of harmonics*

In our study, a Brazilian energy-distribution standard, called PRODIST (ANEEL, 2018), was used to set the parameters for the characterization of different electrical

disturbances on distribution lines, for both transitory events and for a steady-state operation (described in Table 8).

Table 8 – Brazilian energy-distribution standard limits, from PRODIST (ANEEL, 2018) Module 8 (Power Quality) for short-time voltage momentary variation and steady-state for V ≤ 1.0kV

Name	Description	Minimum Value	Maximum Value	Duration
SAG	Short duration reduction in RMS voltage	0.1pu	0.9pu	Up to 3s
SWELL	Short duration increase in RMS voltage	Larger than 1.1pu	NA	Up to 3s
Short Interruption	Relevant short duration reduction in RMS voltage	NA	Lower than 0.1pu	Up to 3s
Frequency	Maximum Accepted Frequency Range	56,5Hz	66Hz	Up to 10s
THD	Total Harmonic Distortion	NA	10%	95% of Steady-State
EHD	Harmonic Distortion for Even harmonics	NA	2.5%	95% of Steady-State
OHD	Harmonic Distortion for Odd harmonics	NA	7.5%	95% of Steady-State
3rdHD	Harmonic Distortion for 3rd harmonic	NA	6.5%	95% of Steady-State

In order to analyze the PLL's response to a group of transitory steps and operations, this study used PRODIST as its standard to introduce steps in voltage and frequency, providing limit conditions for each PLL higher than those established by the references. The chosen test intervals are described in Table 9. Our objective was to excite the control loop of a PLL in the presence of transitory events and steady state harmonics, to verify how frequency and phase were affected in the presence of disturbances; something which occurs frequently in real distribution systems.

Table 9 – Selected PLL test interval limit values for magnitude and frequency tests

Name	Values	Duration
SAG	0.8pu	0.5s
SWELL	1.2pu	0.5s
Interruption	0.0pu	0.25s
Frequency range	54Hz ~ 66Hz	0.5s
THD	26%	100% of Steady-State
EHD	5%	100% of Steady-State
OHD	21%	100% of Steady-State
3rdHD	7%	100% of Steady-State

The sets of parameters for each tested PLL are described in Table 10, and are derived from Ziarani´s work (Ziarani, et al., 2004) and from an M.Sc. dissertation from COPPE/UFRJ (T. A. Brasil, 2013), and are adjusted for an adequate response at 60Hz.

Table 10 – Parameters used for adequate response at 60Hz input

EPLL Parameters		ZPLL Parameters	
Kpamp	100000.0	**u1**	100.0
Kp	150.93	**u2**	10000.0
Ki	22485.0	**u3**	0.0

4.2. Voltage magnitude step response test

The first test intended to discover the PLL´s behavior in the presence of voltage steps, with a clean 60Hz signal at the input of the tested system. Our test showed a brief startup moment, up to 0.75s of 1.0pu magnitude, followed by a SWELL (voltage increasing) of 1.2pu magnitude for 0.5s, before going back to 1.0pu up to 1.75s. After 1.75s, the system exhibited a short interruption (0.0pu) of 0.25s and then restored the supply's voltage signal to 1.0pu for 0.5s. The next step had a SAG (voltage decreasing) of 0.8pu magnitude for 1s, and then restored supply at 3.5s up to 4.0s, for the entire duration of the simulated test. The complete setup for the magnitude steps is shown in Table 11 and illustrated in Figure 32.

The sequence of disturbance test zoom is displayed for SWELL in (Figure 33), SWELL out (Figure 34), Interruption in (Figure 35), Interruption out (Figure 36), SAG in (Figure 37) and SAG out (Figure 38).

Table 11 – PLL Test (1) Setup for magnitude of voltage steps

Time interval	Voltage input	Frequency	Harmonics
0.00 ~ 0.75s	1.0 pu	60Hz	none
0.75 ~ 1.25s	1.2 pu	60Hz	
1.25 ~ 1.75s	1.0 pu	60Hz	
1.75 ~ 2.00s	0.0 pu	60Hz	
2.00 ~ 2.50s	1.0 pu	60Hz	
2.50 ~ 3.50s	0.8 pu	60Hz	
3.50 ~ 4.00s	1.0 pu	60Hz	

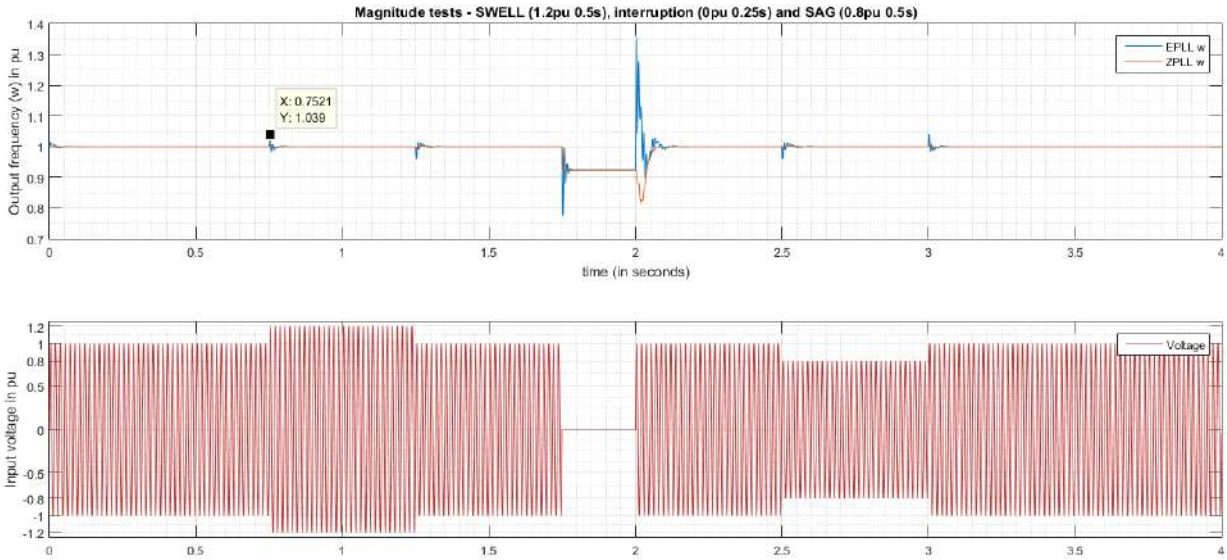

Figure 32 - PLL Tests - Full view of voltage input (in pu) and frequency output (in pu) of EPLL and ZPLL

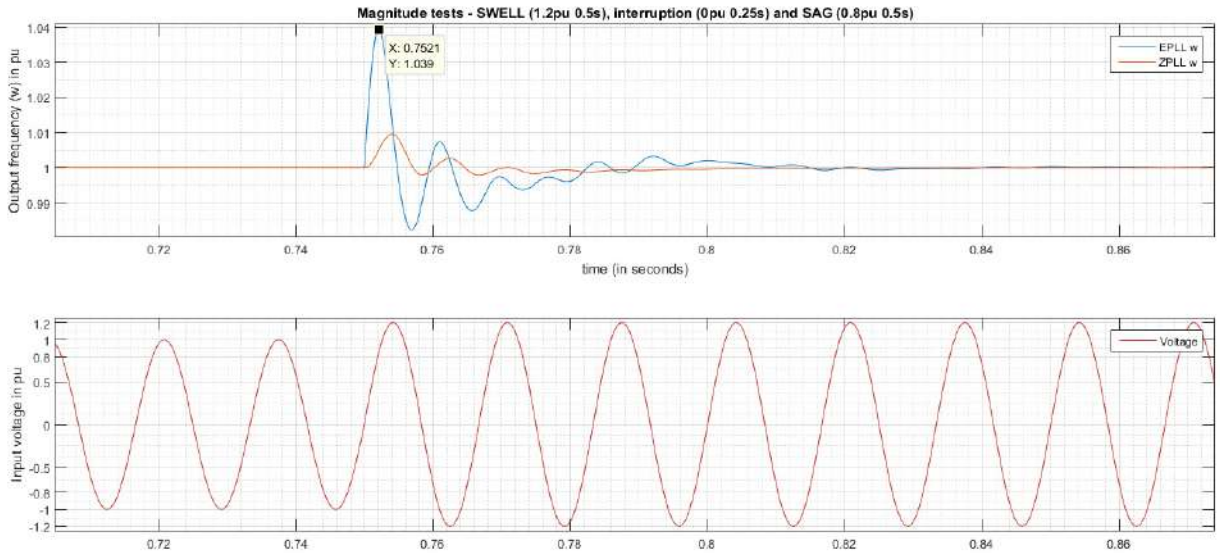

Figure 33 - SWELL In (1.0 to 1.2pu) - Zoom of frequency output (in pu) of EPLL and ZPLL

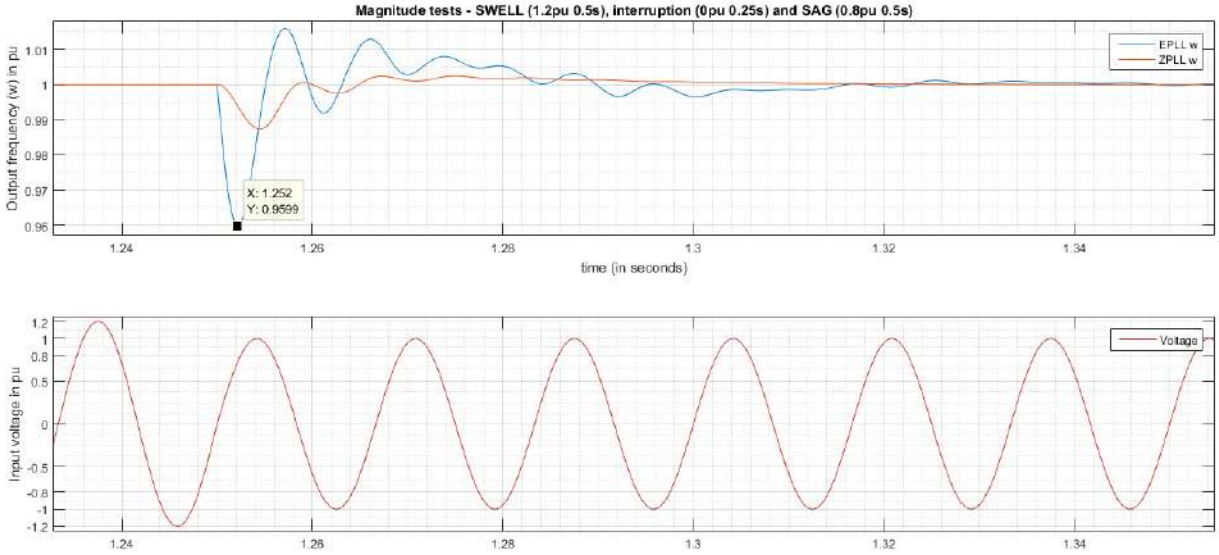

Figure 34 - SWELL Out (1.2 to 1.0pu) - Zoom of frequency output (in pu) of EPLL and ZPLL

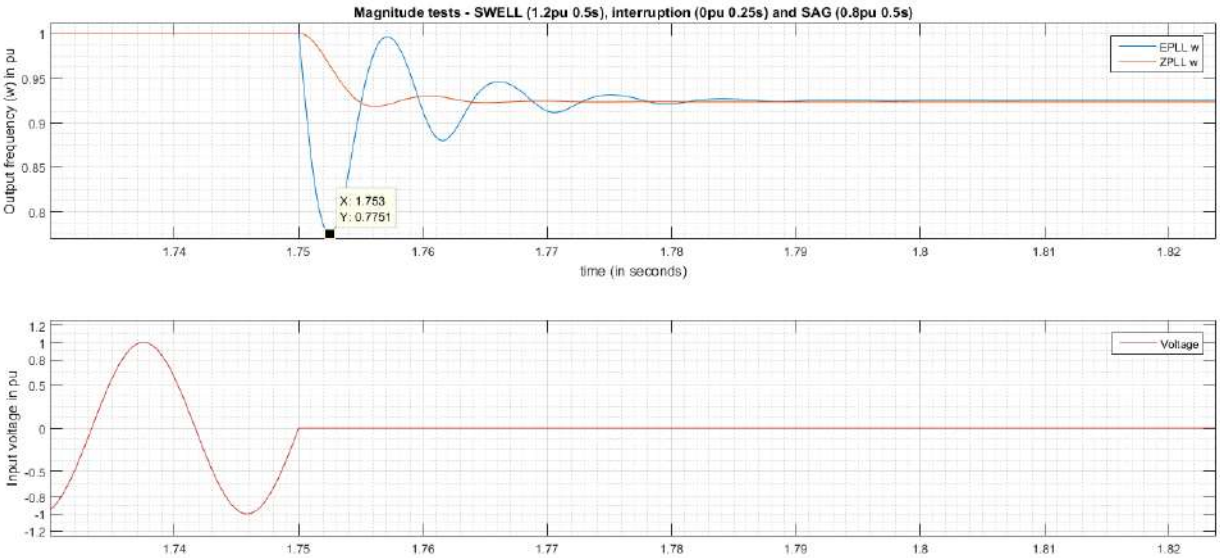

Figure 35 - Interruption In (1.0 to 0.0 pu) - Zoom of frequency output (in pu) of EPLL and ZPLL

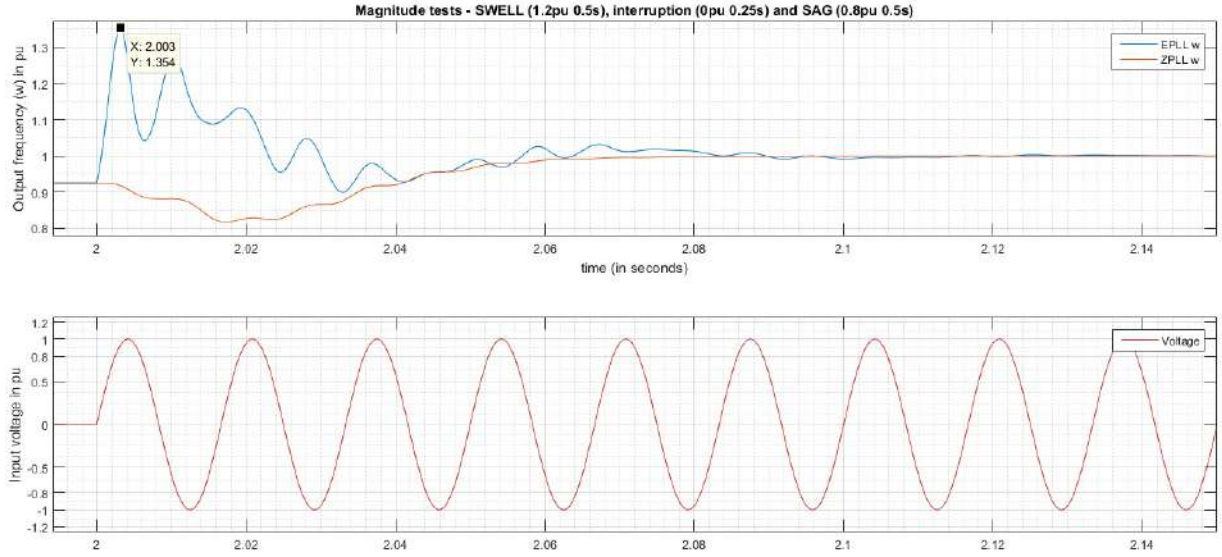

Figure 36 - Interruption Out (0.0 to 1.0pu) - Zoom of frequency output (in pu) of EPLL and ZPLL

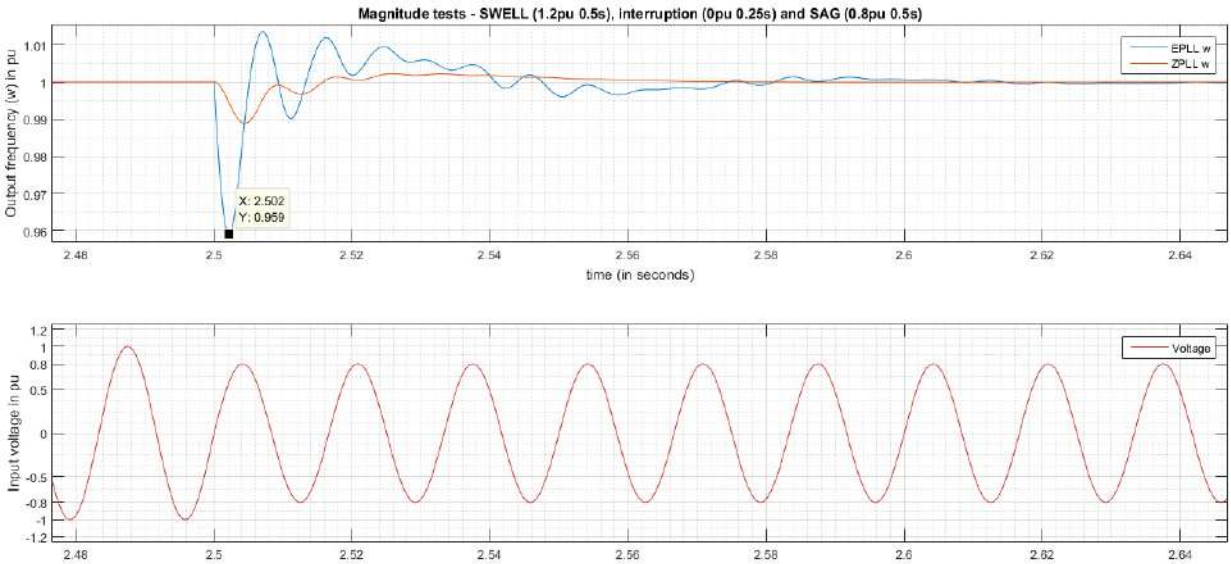

Figure 37 - SAG In (1.0 to 0.8pu) - Zoom of frequency output (in pu) of EPLL and ZPLL

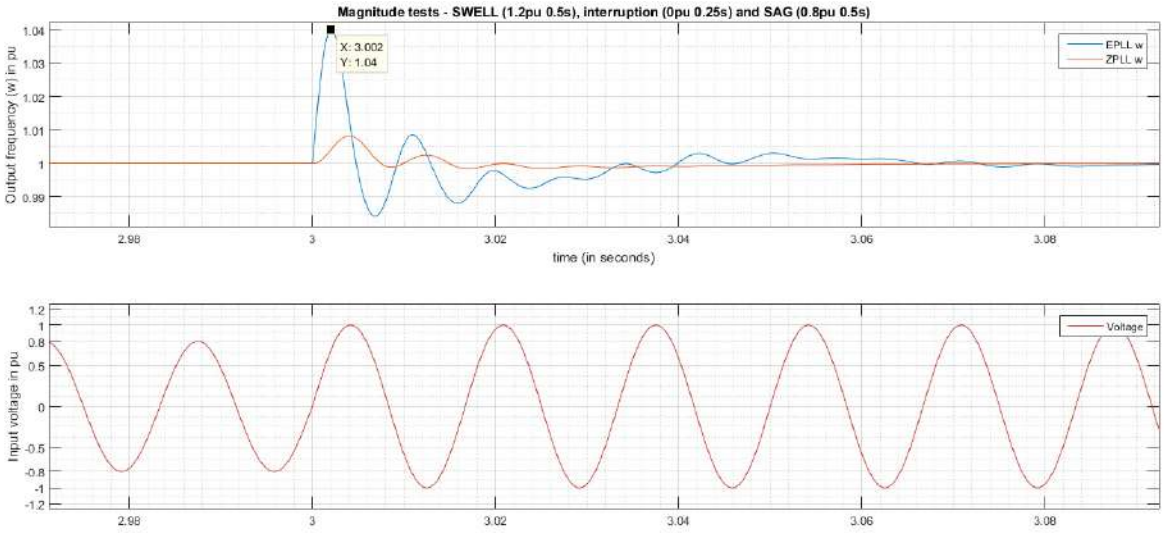

Figure 38 - SAG Out (0.8 to 1.0pu) - Zoom of frequency output (in pu) of EPLL and ZPLL

The results of the maximum error of ω measurement and maximum settling time (in ms) of each PLL are presented in Table 12. It is important to highlight that there was a minimum difference between the PLL´s control loops, with acceptable results for both algorithms. In the steady state, both tested algorithms presented zero offset, locking frequency perfectly, without the presence of harmonics at input.

Table 12 – PLL Test (1) Results for voltage step events

	Events					
Measurement	***SWELL In***	***SWELL Out***	***Interruption In***	***Interruption Out***	***SAG In***	***SAG Out***
EPLL ω Max Error	3.9%	4.1%	22.5%	35.4%	4.1%	4%
ZPLL ω Max Error	1%	1.3%	8.2%	18.2%	1.1%	0.8%
EPLL / ZPLL Error ratio	3.9X	3.15X	2.74X	1.94X	3.72X	5X
Maximum measured settling time						
EPLL	270ms		**ZPLL**	120ms		

4.3. Voltage magnitude step response test with harmonics

The second test intended to discover the behavior of the PLL´s in the presence of voltage steps, with in-phase harmonic components mixed with the 60Hz at the input of each PLL. The harmonic magnitude ratios for each individual in-phase contributions are shown in Figure 39. The 2nd harmonic has 1%, the 3rd harmonic 7%, the 4th harmonic 1%, the 5th harmonic 5%, the 6th harmonic 1%, the 7th harmonic 3%, the 8th harmonic 1%, the 9th harmonic 3%, the 10th harmonic 1% and the 11th harmonic has 3%. The displayed values were assigned heuristically, based on PRODIST, as previously shown in Table 8.

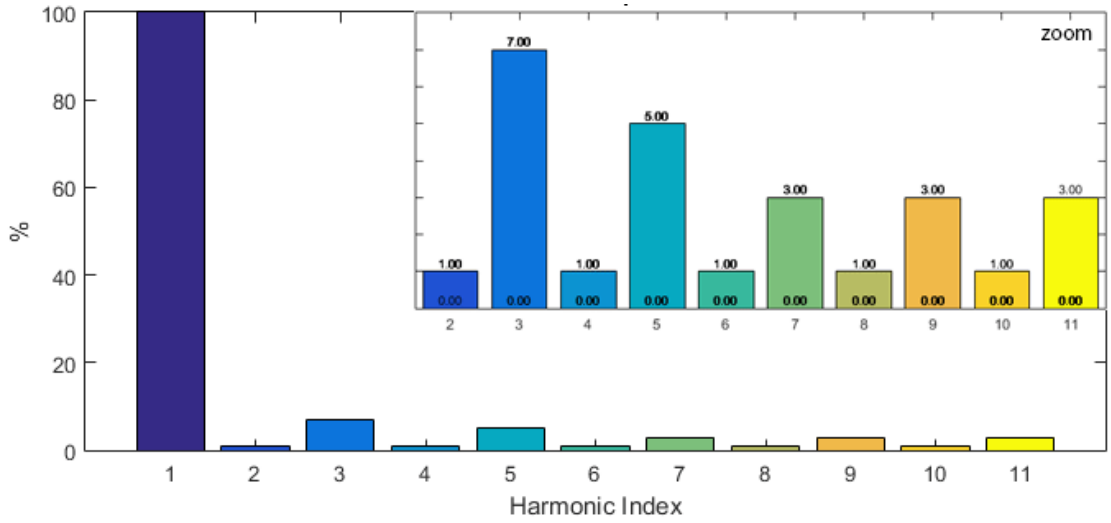

Figure 39 – Harmonic components for PLL setup test 2

The test for the voltage steps with harmonics follows same time distribution as in Section 4.2. It started with magnitudes from 0s to 0.75s of 1.0pu, followed by a SWELL of 1.2pu for 0.5s, and then returned to 1.0pu, finishing at 1.75s. Thereafter, the system exhibited a short interruption (0.0pu) of 0.25s and then restored the supply voltage to 1.0pu for 0.5s. The next step had a SAG of 0.8pu magnitude for 1s, and then restored supply to 1.0pu at 3.5s for up to 4.0s, for the total duration of the simulated test. The complete setup for the magnitude steps is shown Table 13 and in Figure 40.

Table 13 – PLL Test (1) Setup for magnitude of voltage steps

Time interval	Voltage input	Frequency	Harmonics
0.00 ~ 0.75s	1.0 pu	60Hz	*2nd harmonic - 1%* *3rd harmonic - 7%* *4th harmonic - 1%* *5th harmonic - 5%* *6th harmonic - 1%* *7th harmonic - 3%* *8th harmonic - 1%* *9th harmonic - 3%* *10th harmonic - 1%* *11th harmonic - 3%*
0.75 ~ 1.25s	1.2 pu	60Hz	
1.25 ~ 1.75s	1.0 pu	60Hz	
1.75 ~ 2.00s	0.0 pu	60Hz	
2.00 ~ 2.50s	1.0 pu	60Hz	
2.50 ~ 3.50s	0.8 pu	60Hz	
3.50 ~ 4.00s	1.0 pu	60Hz	

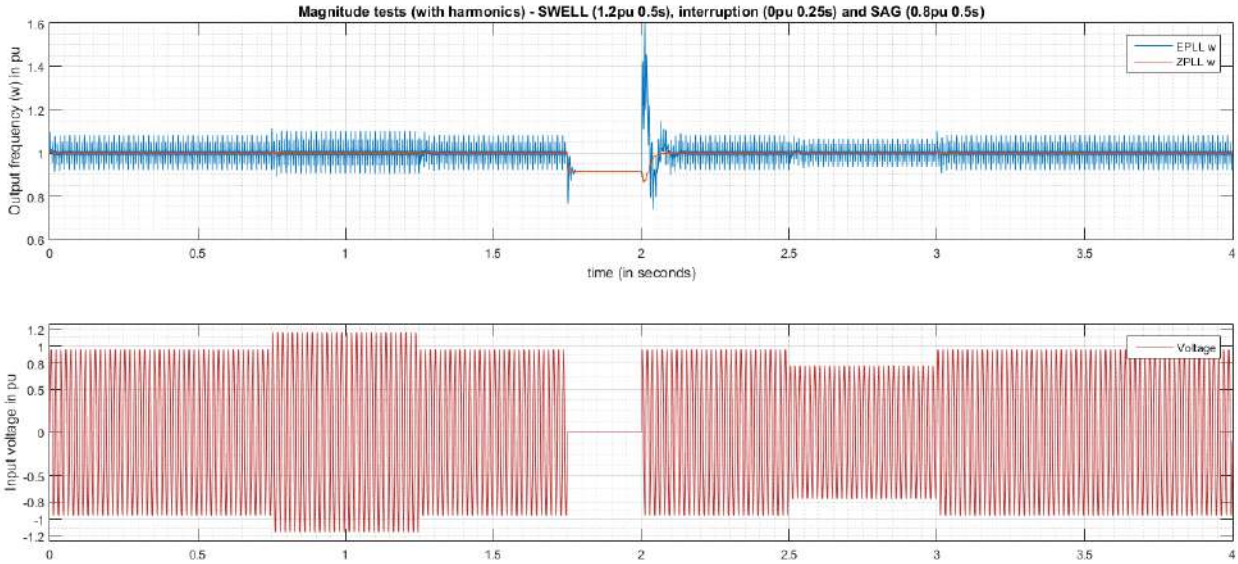

Figure 40 - PLL Tests - Full view of voltage input (in pu) and frequency output (in pu) of EPLL and ZPLL with harmonics

The sequence of the disturbance test zoom is displayed for SWELL in (Figure 41), SWELL out (Figure 42), Interruption in (Figure 43), Interruption out (Figure 44), SAG in (Figure 45) and SAG out (Figure 46), for all steps of the developed test.

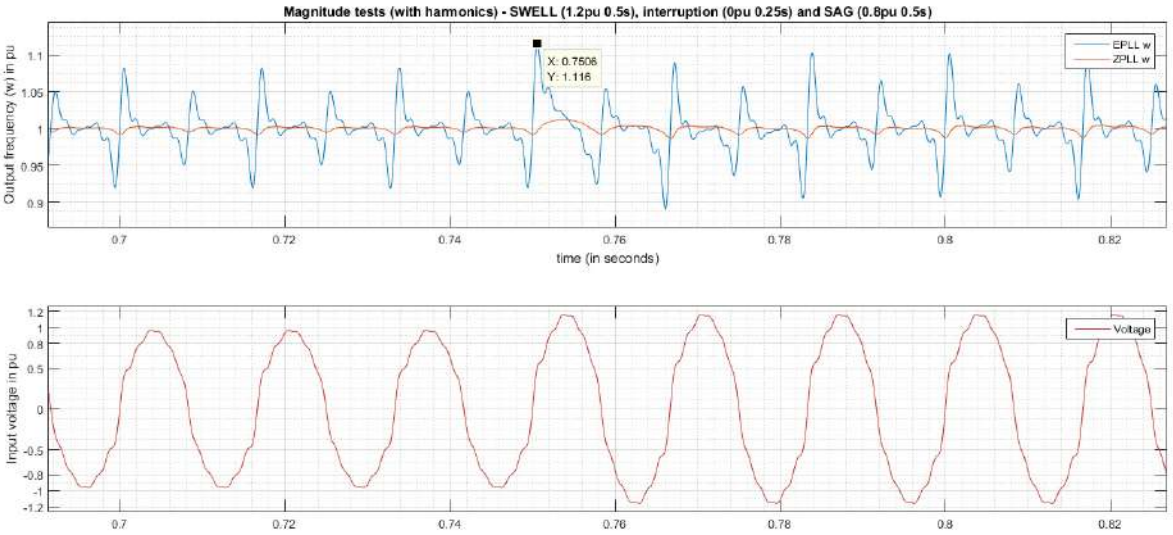

Figure 41 - SWELL In (1.0 to 1.2pu) - Zoom of frequency output (in pu) of EPLL and ZPLL with harmonics

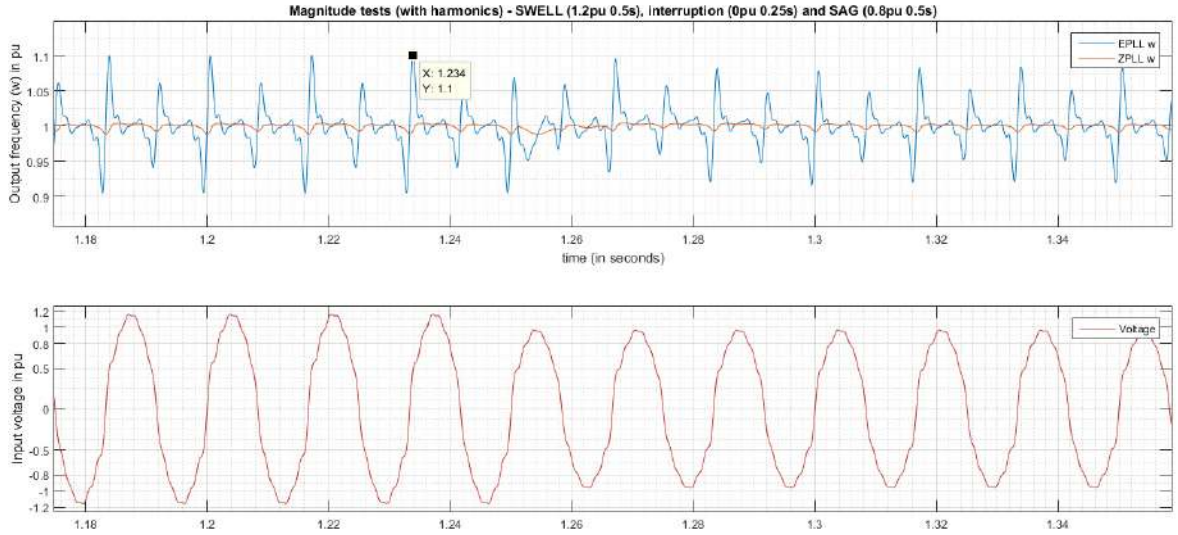

Figure 42- SWELL Out (1.2 to 1.0pu) - Zoom of frequency output (in pu) of EPLL and ZPLL with harmonics

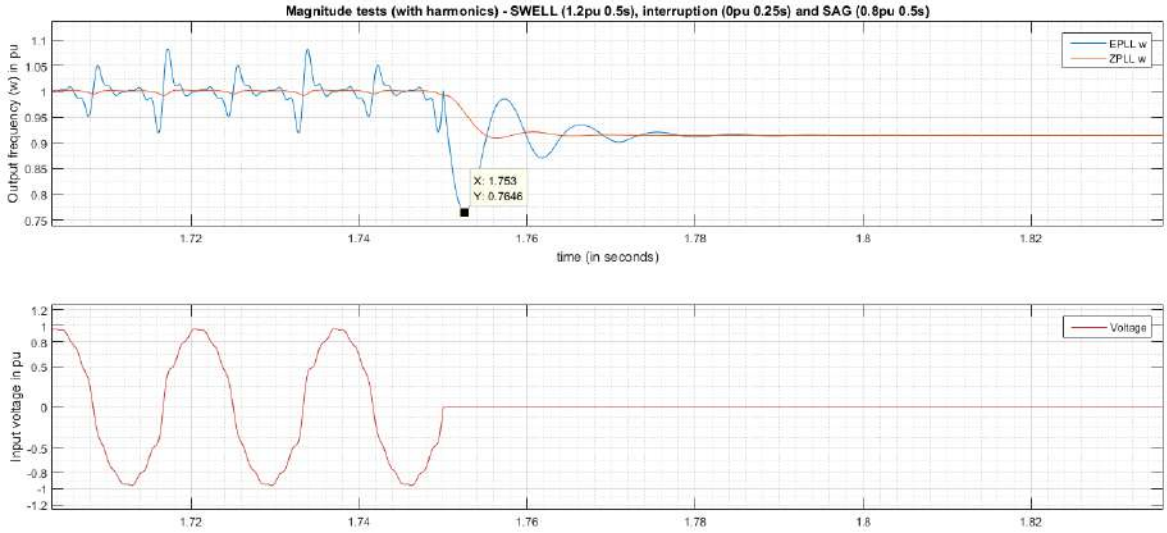

Figure 43 - Interruption In (1.0 to 0.0 pu) - Zoom of frequency output (in pu) of EPLL and ZPLL with harmonics

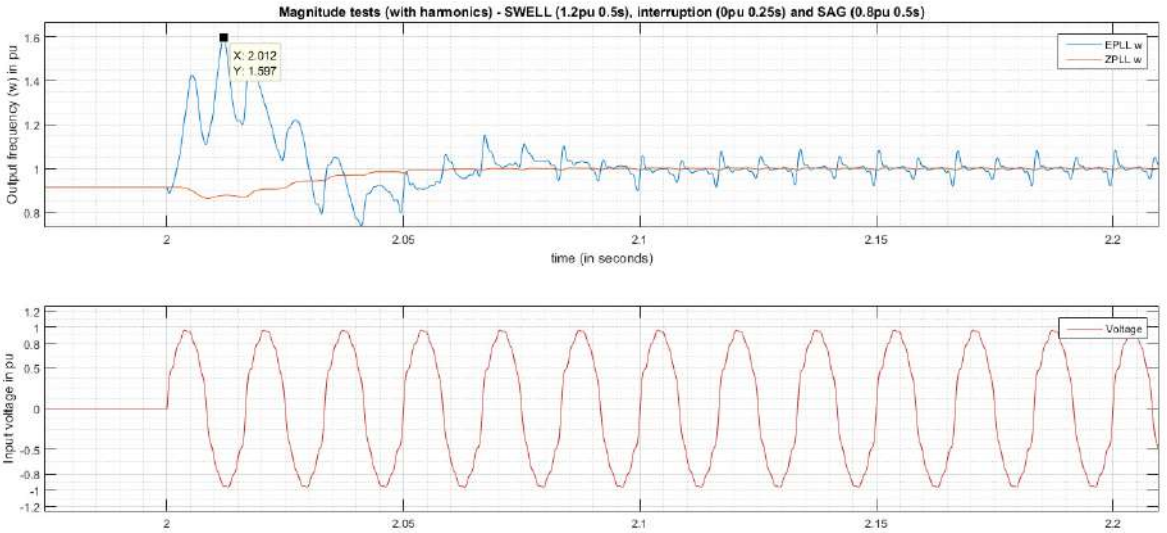

Figure 44 - Interruption Out (0.0 to 1.0pu) - Zoom of frequency output (in pu) of EPLL and ZPLL with harmonics

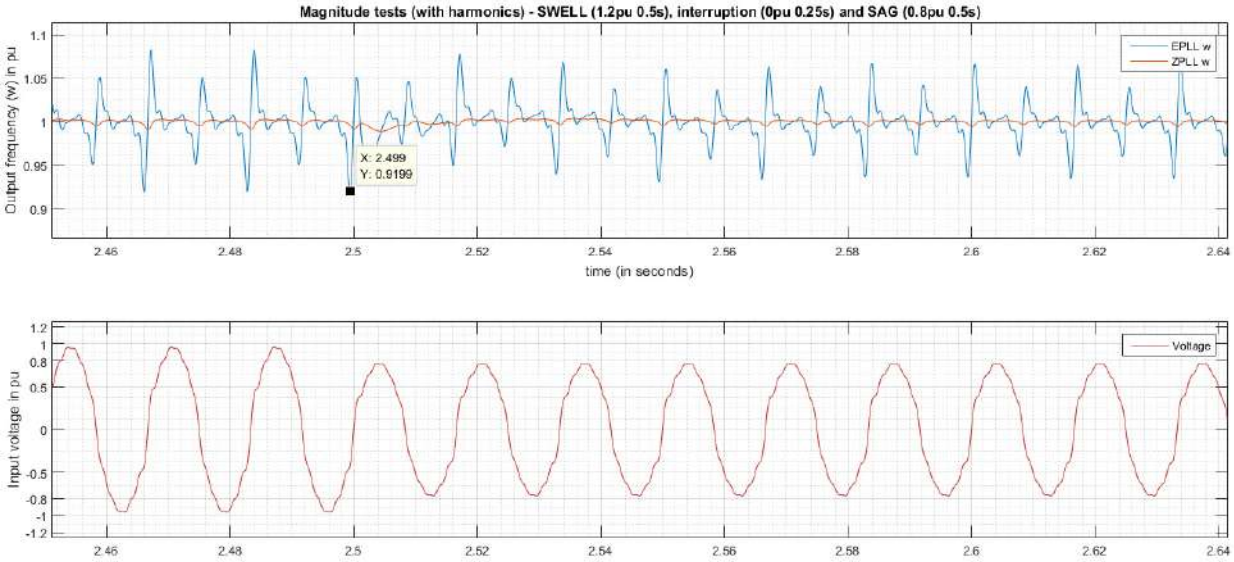

Figure 45- SAG In (1.0 to 0.8pu) - Zoom of frequency output (in pu) of EPLL and ZPLL with harmonics

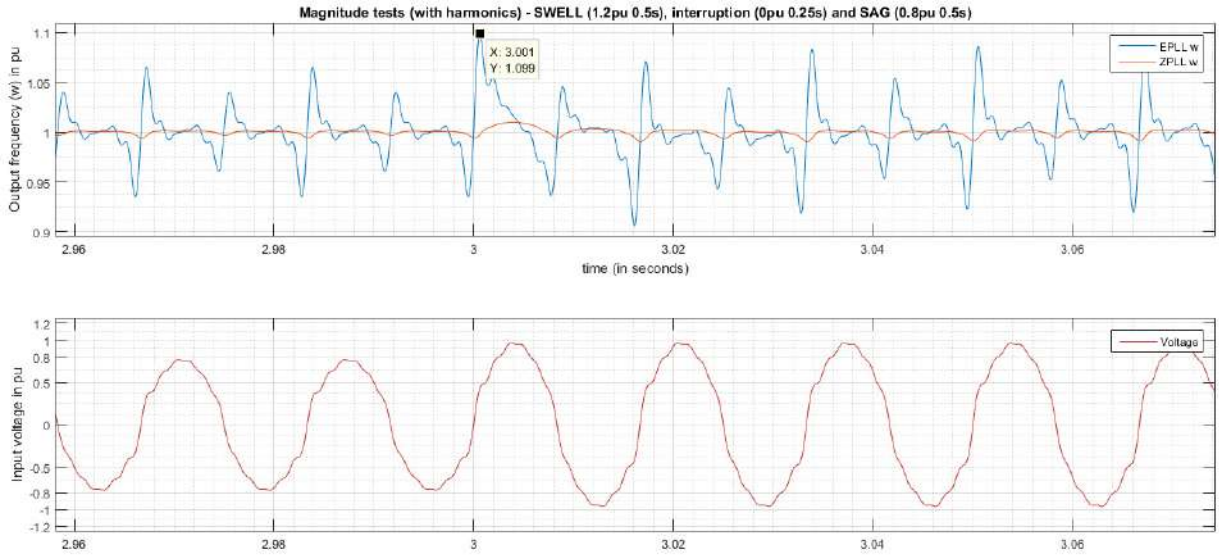

Figure 46 - SAG Out (0.8 to 1.0pu) - Zoom of frequency output (in pu) of EPLL and ZPLL with harmonics

The results are shown in Table 14 for both a maximum error of ω locking and the maximum settling time (in ms) of each PLL. It is important to highlight the significant difference between the PLL´s control loops. In a steady state, the algorithms presented a variation of ±10% for EPLL and less than 1% for ZPLL, offset around the fundamental frequency by the presence of harmonics.

The voltage from the energy distribution lines usually featured harmonics, induced by non-linear loads and the absence of active filters over distribution lines. Thus, the performance of the ZPLL presents advantages for the mitigation of these instabilities in steady-state ω tracking.

Table 14 - PLL Test (2) Results for voltage step events with harmonics

	Events					
Measurement	***SWELL In***	***SWELL Out***	***Interruption In***	***Interruption Out***	***SAG In***	***SAG Out***
EPLL ω Max Error	11.6%	11%	11%	59.7%	9.8%	9.9%
ZPLL ω Max Error	4.0%	3.98%	3.95%	38%	3.65%	3.7%
EPLL / ZPLL Error ratio	2.9X	2.76X	2.78X	1.57X	2.68X	2.67X
Maximum measured settling time						
EPLL	120ms		**ZPLL**	70ms		

4.4.Frequency magnitude step response test

The third test aimed to observe the behavior of the PLL´s in the presence of frequency steps. The test started with 1.0s of 1.0pu magnitude (60Hz), followed by a step to 1.1pu (66Hz) for 0.5s, and then returned to 1.0pu (60Hz) at 1.5s. At this time, system continued for 1.0s up to 2.5s, when frequency dropped to 0.9pu (54Hz) for 0.5s, and was then restored to 1.0pu (60Hz), from 3.0s up to 4.0s, for the total duration of the simulated test. The complete setup for the magnitude of the steps is shown in Table 15 and illustrated in Figure 47.

Table 15 - PLL Test (3) Results for frequency step events

Time interval	Voltage input	Frequency	Harmonics
0.00 ~ 1.0s	1.0 pu	60Hz	none
1.0 ~ 1.5s	1.0 pu	66Hz	
1.5 ~ 2.5s	1.0 pu	60Hz	
2.5 ~ 3.0s	1.0 pu	54Hz	
3.0 ~ 4.0s	1.0 pu	60Hz	

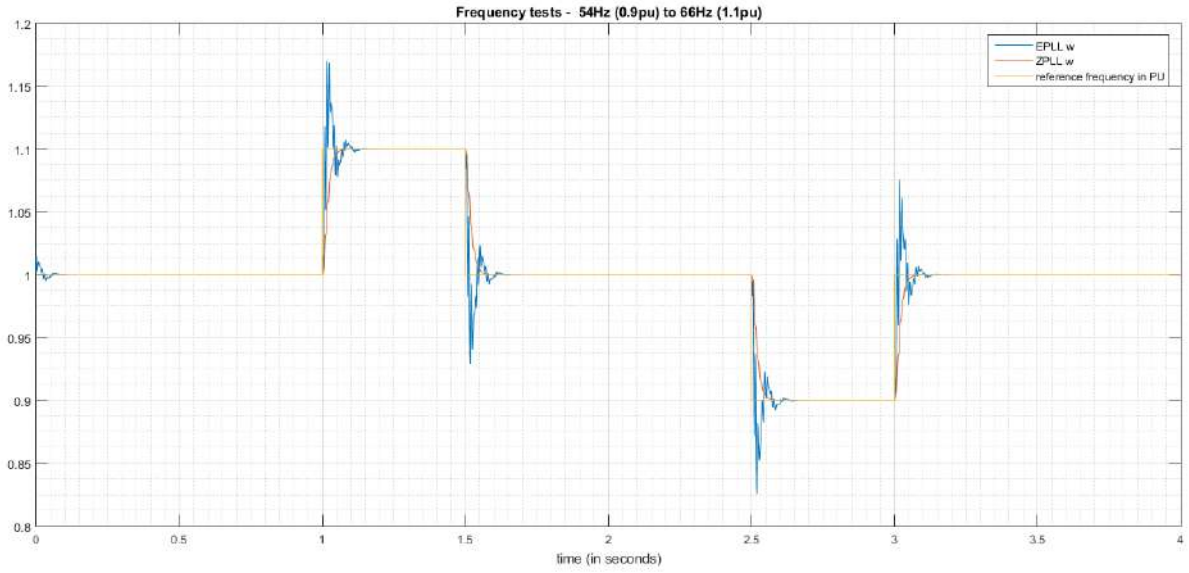

Figure 47 - PLL Tests - Full view of frequency (pu) and frequency tracking of EPLL and ZPLL

At full view, and, more obviously, at zoom view, Figure 48 shows that the test demonstrated the superiority of ZPLL over EPLL, for tracking frequency steps. ZPLL presented no overshoot, in contrast of 6.5% from EPLL, and had a faster response time (approximately 60ms to recover against approximately 160ms from EPLL) for reestablishing the new frequency and tracking it correctly.

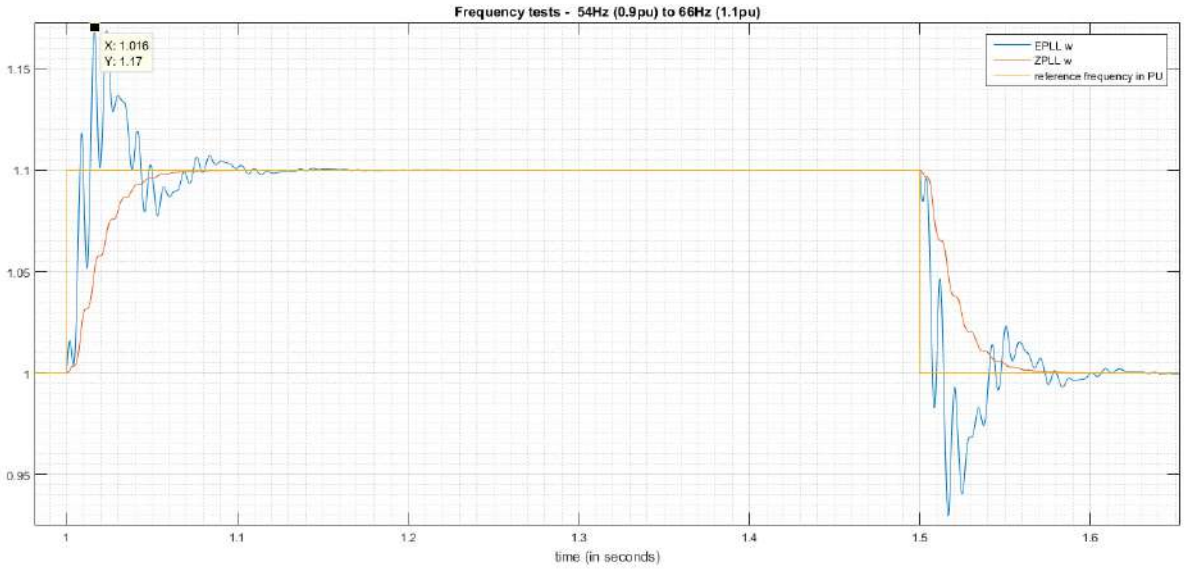

Figure 48 – Zoom view of frequency (pu) and frequency tracking (pu) of EPLL and ZPLL

It is important to take conditions such as this into account for testing purposes, when attempting to guarantee the stability of PLL´s control loops, because frequency steps are difficult to replicate in a grid connected environment, because of the natural stability of the synchronous generators used by energy providers in their grids. However, it is altogether possible that this situation could occur within certain environments; for example, inside microgrids or with a huge *virtual power plant* (VPP) with intermittent/ renewable power supplies, or with an *uninterruptible power supply* (UPS) connected to that environment.

4.5.Frequency magnitude step response test with harmonics

The fourth test aimed to investigate the behavior of the PLL´s in the presence of frequency steps and harmonics, mixed with the 54Hz, 60Hz and 66Hz steps of the test. The observed individual in-phase harmonic contribution content has already been presented in Figure 39, with values based on PRODIST, and also previously shown in Table 8.

As expected from previous tests, in Section 4.4, this test demonstrated the superiority of ZPLL over EPLL, for tracking frequency steps, in the observed scenario. Moreover this was true in a steady-state and not only transient states. The results are shown in Figure 49, as a full test view. Figure 50 shows the zoom between 2.0s and

3.0s. ZPLL showed no overshoot, in contrast of an overshoot of 12.5% for EPLL, and once again, ZPLL had a faster response time for reestablishing and correctly tracking the new frequency.

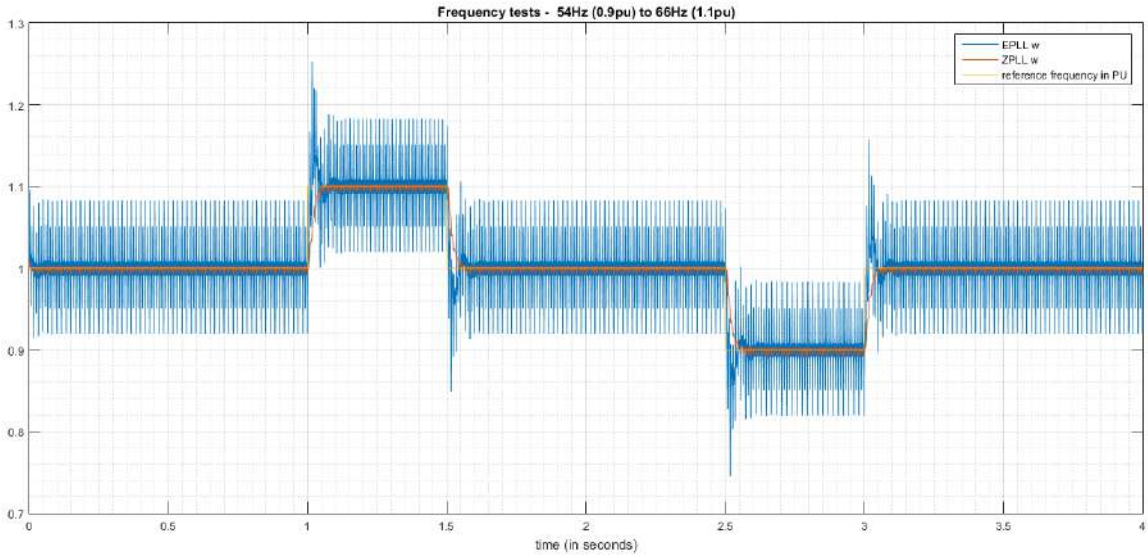

Figure 49 - PLL Tests - Full view of frequency (pu) and frequency tracking (pu) of EPLL and ZPLL with harmonics

Because of the results achieved in this PLL study, we decided to use **ZPLL** as the main PLL for frequency synchronization, and for delivering synchronized datasets to the next step of the proposed method.

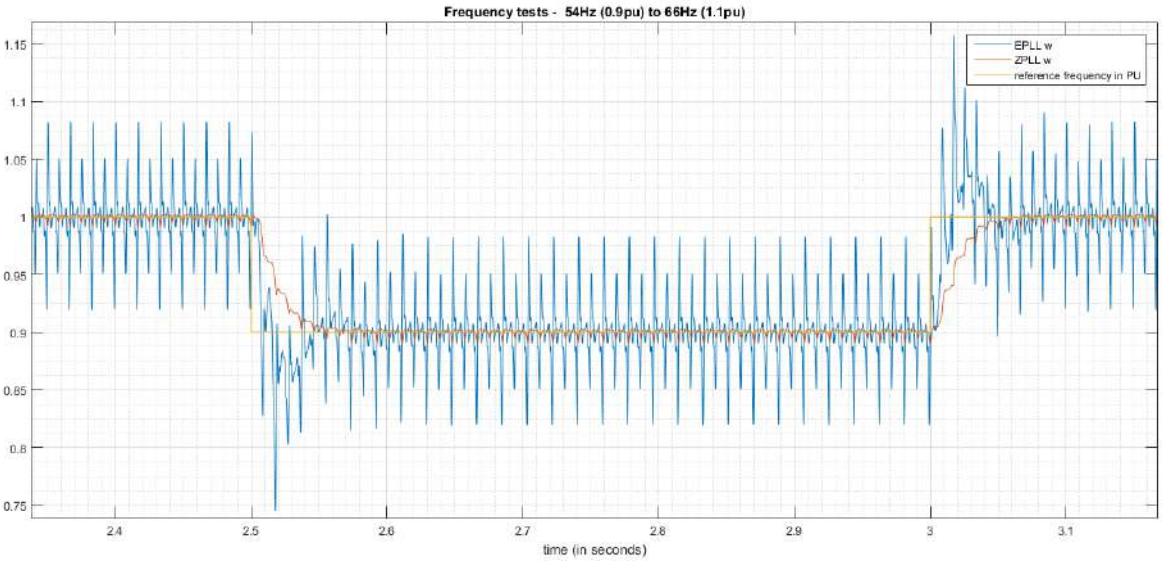

Figure 50 – Zoom view of frequency (pu) and frequency tracking (pu) of EPLL and ZPLL

4.6.Resampling waveforms

This thesis sampled approximately 100 hours of data from appliances at 10 kHz, as the initial step of the BRAD dataset, shown in Figure 51. After using ZPLL to track the frequency and phase of individual appliances' voltage waveforms, including all 94 appliances, the PLL algorithm had evaluated 8,596,755 cycles of 60Hz.

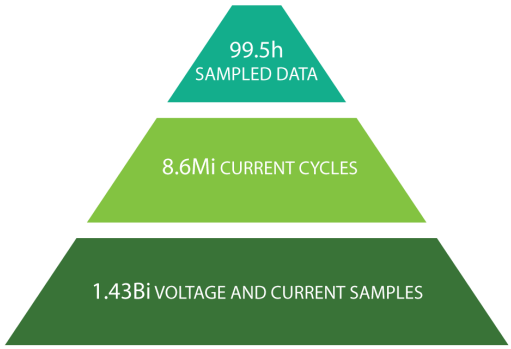

Figure 51 – Sampled data numbers

It is known that, by ratio between sampling rate (10 kHz) and voltage of grid rate (approximately 60Hz), not every cycle of sampled voltage and current will display the same number of samples. From the BRAD dataset, approximately 64.2% of the total measured 60Hz cycles contained 167 samples. Because the majority of cycles contained 167 samples, we proposed resampling the remaining 3,076,667 cycles, which did not match the 167 samples, in order to cohere them to other 167 samples 60Hz cycles.

The proposed resampling process allowed all voltage and current cycles from the different appliances to have the same number of samples. As previously stated (in Section 3.2), this condition permits the creation of synthetic mixture scenarios by adding individual appliances' current channels as an aggregate, because the mixture of two or more currents, flowing into a node, could be represented by the sum of each current separately.

The resampling was made using a *resample* function from MATLAB software (MATHWORKS, 2016). This performs an FIR filtering process, including anti-aliasing, normalization of the results, and then implements a rate change, by a desired value. Our initial tests of the waveform processing evaluated some voltage and current cycles and, visually, those cycles didn't show any distortion in waveform, including edge effects.

However, a visual check of signals alone, does not guarantee that all the processes of resampling will maintain the original waveforms and their features integrally. Thus, we proposed a method to investigate all 3,076,667 cycles, which did not match the 167 points, automatically, by measuring the variation of generated waveforms and comparing their shapes before and after the resample processing. Thus, if the measured variation was below an acceptable value, the feature extraction from the resampled waveform assured a preservation of the original information.

Power Spectrum Density (PSD) describes how the power of a signal is distributed over frequency. The nature of the signal can often be determined from information contained in the measured spectrum. Consequently, evaluating an estimation of PSD, before and after resampling, for both voltage and current, can provide important information about changes in waveforms.

To accomplish that, we proposed a method to evaluate significant changes in studied signals automatically by computing the PSD. The steps are described as:

i. For each appliance K, use the PLL algorithm and enumerate each cycle (N) of 60Hz, as shown in Figure 52.

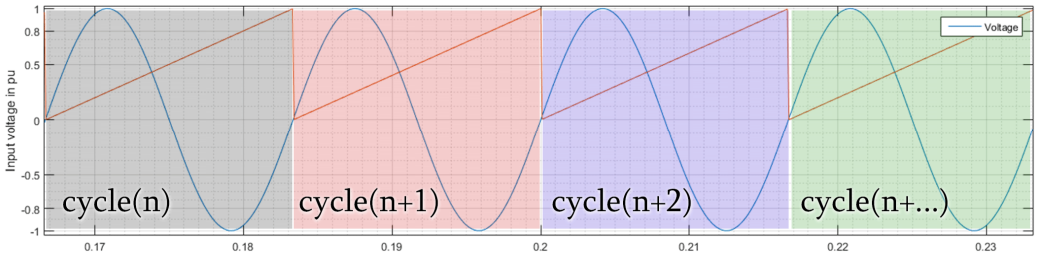

Figure 52 – Each cycle of voltage and current from the appliance samples is enumerated after the synching procedure

ii. For each cycle index (N) from the original **sampled** channels of voltage and current, compute the **sum** of the spectrum for each signal:

$$SUM_{sampled}(K,N) = \{sum(PSD(cycle(K,N))^{voltage}_{sampled}) | sum(PSD(cycle(K,N))^{current}_{sampled})\};$$

iii. For each cycle index (N) from the **resampled** channels of voltage and current, compute the **sum** of the spectrum for each signal:

$$SUM_{resampled}(K,N) = \{sum(PSD(cycle(K,N))^{voltage}_{resampled}) | sum(PSD(cycle(K,N))^{current}_{resampled})\};$$

iv. For each evaluated **sum**, compute the **ratio** between signals **before** and **after** resampling:
$PSD_{ratio}(K,N) = SUM_{resampled}(K,N) \div SUM_{sampled}(K,N)$;

v. For each evaluated voltage and current channel in $PSD_{ratio}(K,N)$, compute the **mean** value of all ratios, obtained from the cycle index (N), for each appliance K:
$PSD_{mean}(K) = mean(PSD_{ratio}(K,N),N)$;

vi. For each evaluated voltage and current channel in $PSD_{ratio}(K,N)$, compute the **standard deviation** value of all the ratios, obtained from the cycle index (N), for each appliance K:
$PSD_{std}(K) = std(PSD_{ratio}(K,N),N)$;

The **mean** ($PSD_{mean}(K)$) and **standard deviation** ($PSD_{std}(K)$) values**,** obtained for each appliance K, contained important information about the changes in the power spectrum that originated in the resample process. The $PSD_{std}(K)$ measured how far the different cycles were from the **mean**. It is important to mention that completely different signals can exhibit similar values for the **mean** and **standard deviation** of the PSD, which could have misled the proposed comparison. However, our proposed analysis related two signals that were, by assumption, from the same source and, therefore, intrinsically high correlated, allowing this comparison to proceed.

According to an article from NiPy (NITIME, 2009), naive estimations of power spectrum, based on Fourier Transform DFT/FFT algorithms, suffer from several problems (Teukolsky, et al., 2007). Welch´s periodogram (Welch, 1967) and Multi-taper spectral estimation (MTSE) (Percival, et al., 1993) are both algorithms that are able to infer the PSD more precisely, with MTSE being less error-prone than Welch´s algorithm (NITIME, 2009).

Therefore, we computed all appliances' waveforms PSD´s using MTSE in order to evaluate the proposed method. The results of $PSD_{mean}(K)$ for the voltage and current channels, of each appliance K, are illustrated in Figure 53 and Figure 54, respectively. Similarly, $PSD_{std}(K)$ for the voltage and current channels, of each appliance K, are shown in Figure 55 and Figure 56.

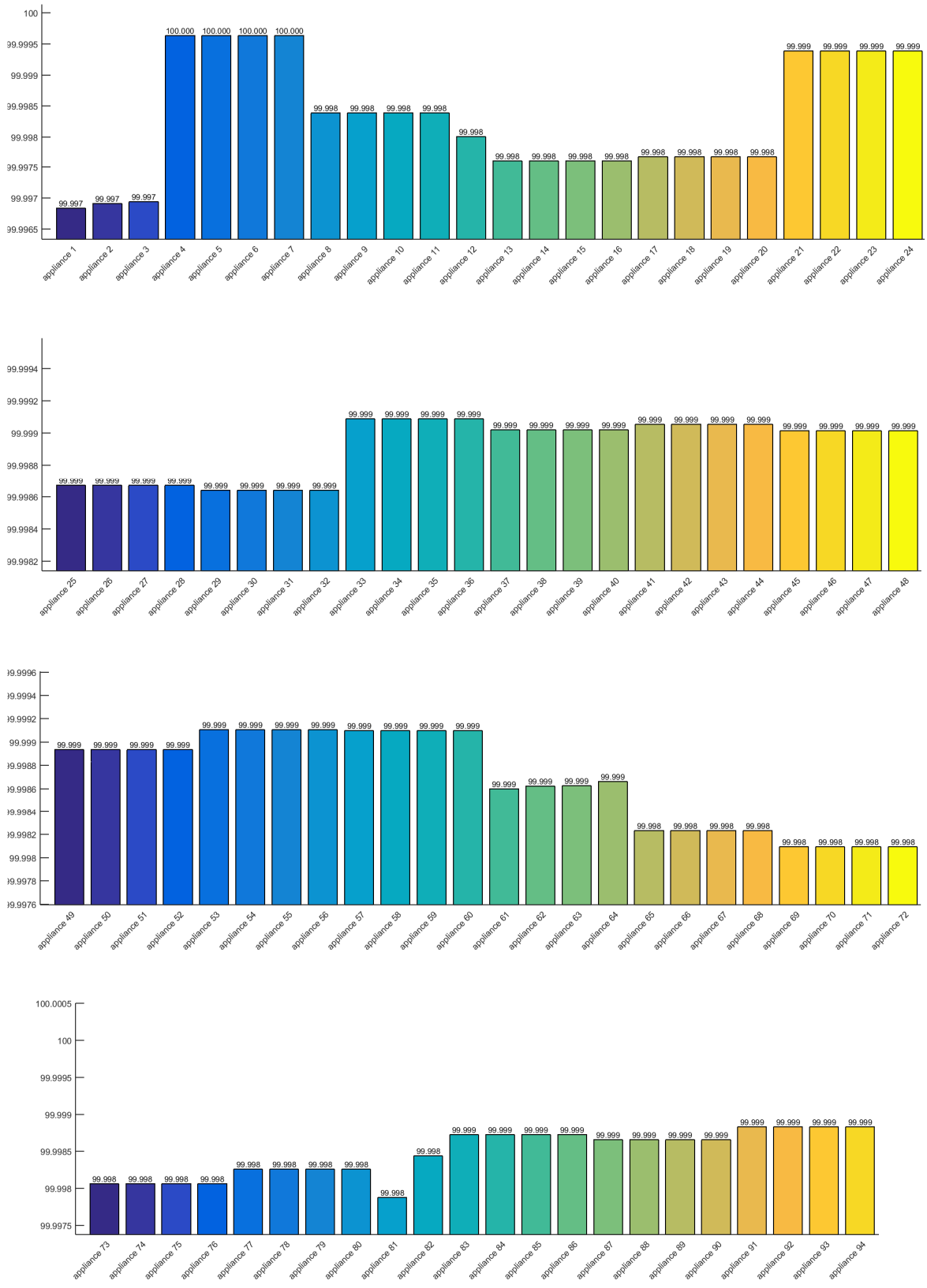

Figure 53 – Voltage $PSD_{mean}(K)$ values for each appliance K

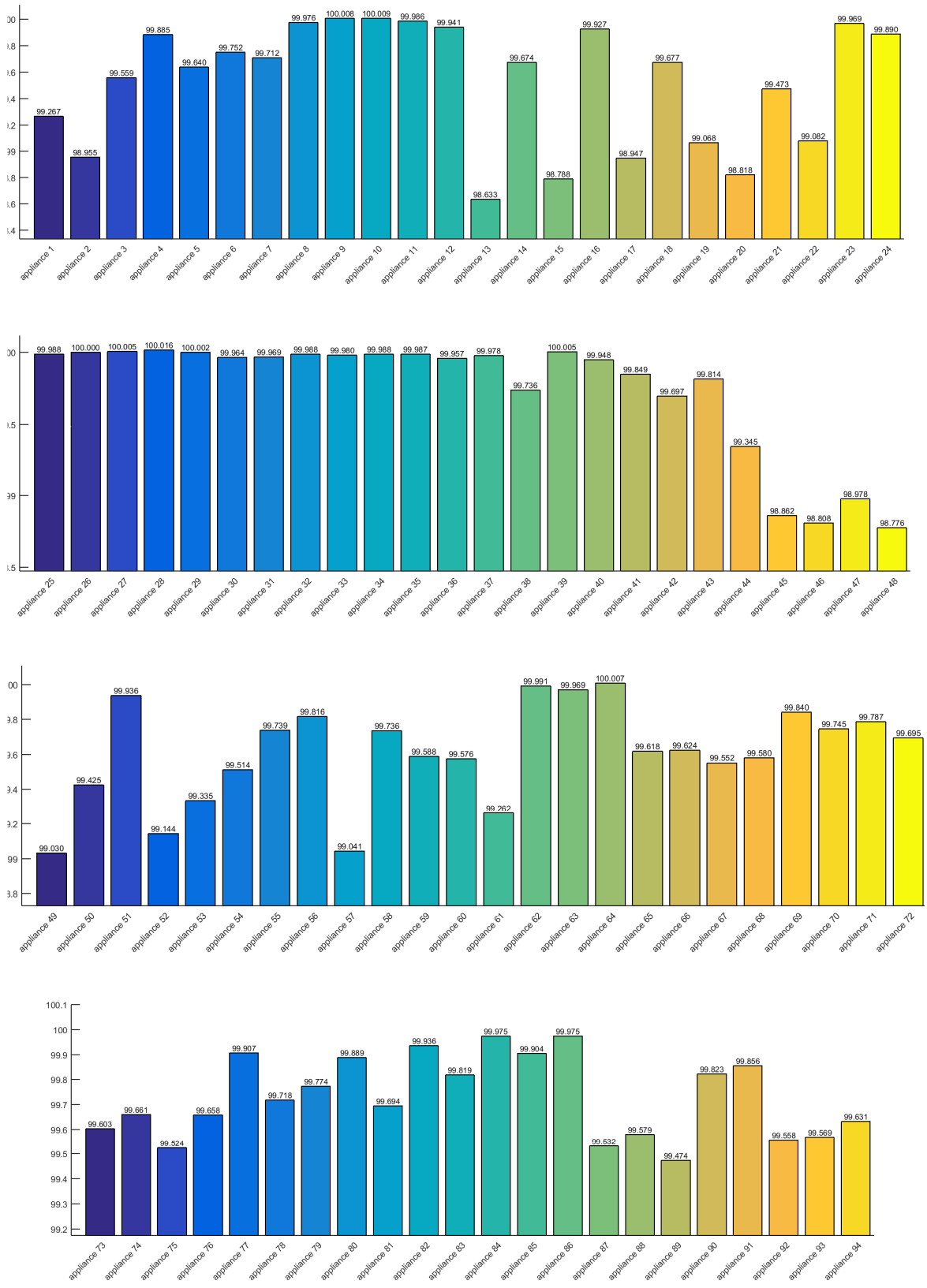

Figure 54 – Current $PSD_{mean}(K)$ values for each appliance K

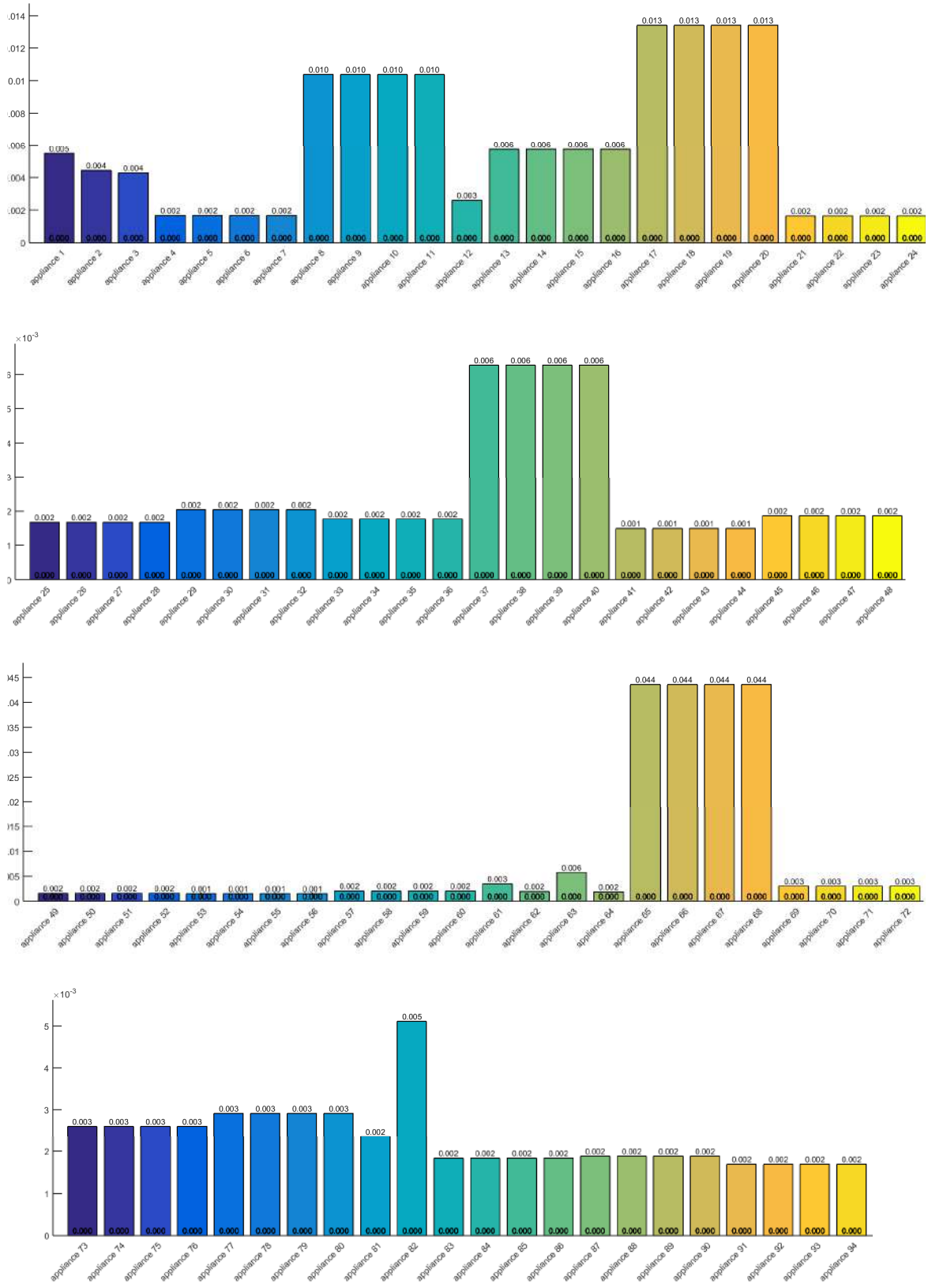

Figure 55 – Voltage $PSD_{std}(K)$ values for each appliance K

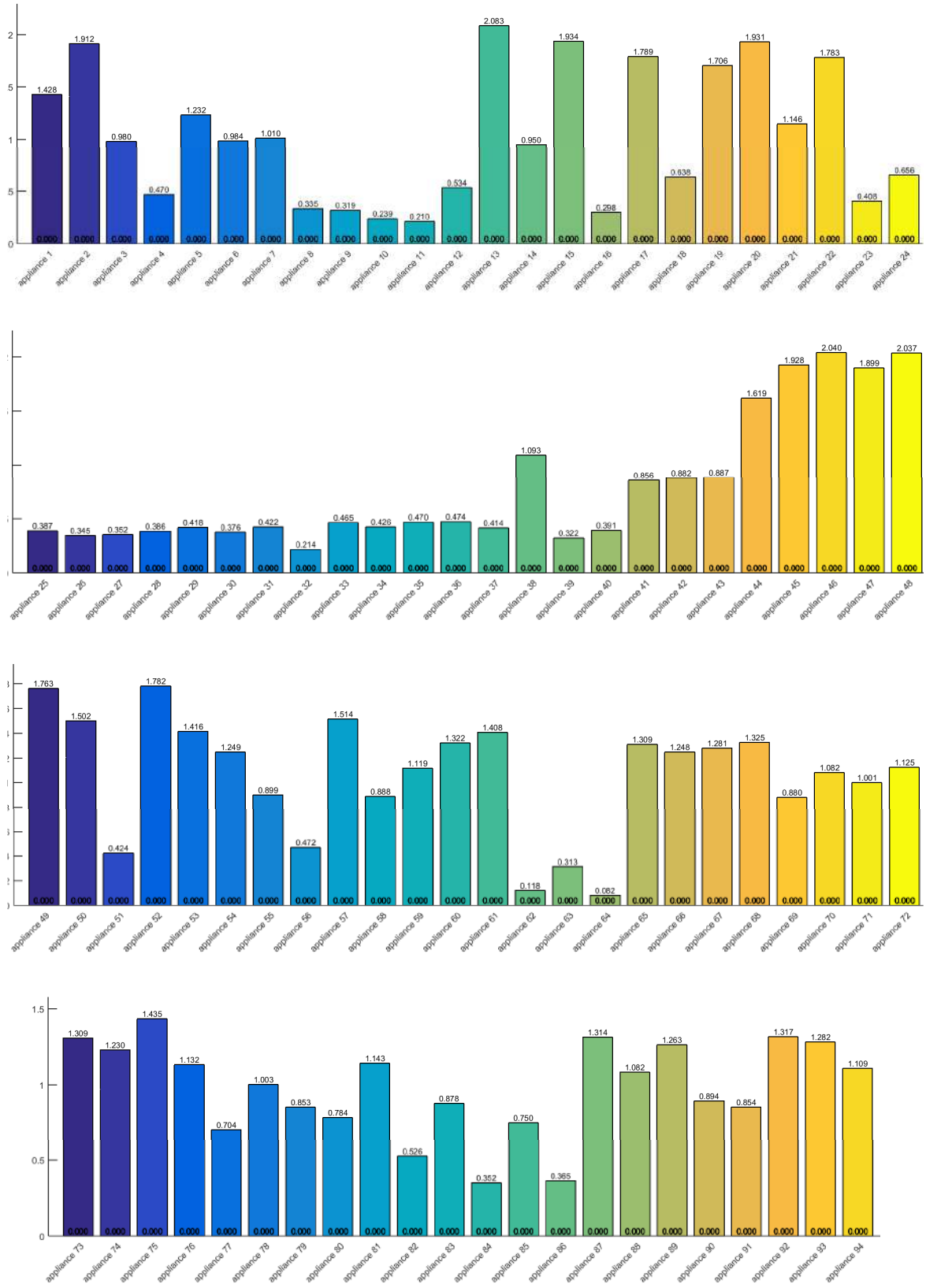

Figure 56 – Current $PSD_{std}(K)$ values for each appliance K

When we reviewed the results for $PSD_{mean}(K)$ in the voltage and current channels, the majority of values were near 99~100%, denoting similar PSD **mean** values for both the original and the resampled waveforms. Otherwise, results obtained from **standard deviation**, suggested that error spread in the PSD, caused by the resampling, were low (less than ~2%). This means that most of the spectrum was preserved. Consequently, the resample was applied to all the sampled data, for both voltage and current waveforms, enabling us to combine the current signals from different appliances into a single mixture stream.

4.7.Synthetic dataset results

In order to ensure different aggregate scenarios, with similar non-intrusive monitoring complexity levels, we proposed initially selecting ten appliances at a time, from the 94 individual acquired appliances, without repetitions. These rules forced each singular scenario to have different appliances, thus avoiding duplicates. Furthermore, we wanted each appliance included, to be inside at least ten different mixture scenarios, for training purposes.

This meant that the number of potential combinatorial analysis for 94 appliances, at 10 by 10, and without repetitions, was nearly 9.04 x 10^{12}. Without using a brute force algorithm, we used a combinatorial optimization "branch and cut" algorithm, which was described by Mitchell in his work (Mitchell, 1999). In this way it was possible to use only 93 different combinations, to meet the combinatorial requirements.

To resolve the problem of different waveform lengths, smaller waveforms were repeated up to the length of the larger ones, and the final samples were clipped if the size did not match. In order to display the waveforms in each scenario, we highlighted one example from the combinatorial mixture files. The randomly selected appliances were:

1 – A **15 inch TV** (waveform displayed in Figure 57),

2 – A **fridge** (waveform displayed in Figure 58),

3 – **The charger for an electric drill** (waveform displayed in Figure 59),

4 – A **food processor** (waveform displayed in Figure 60),

5 – A **15W yellow fluorescent light** (waveform displayed in Figure 61),

6 – **A 18W yellow fluorescent light** (waveform displayed in Figure 62),

7 – A 48W hoverboard charger (waveform displayed in Figure 63),

8 – **The lighter for a stove** (waveform displayed in Figure 64),

9 – **A vacuum Cleaner** (waveform displayed in Figure 65),

10 – A **7W LED light** (waveform displayed in Figure 66).

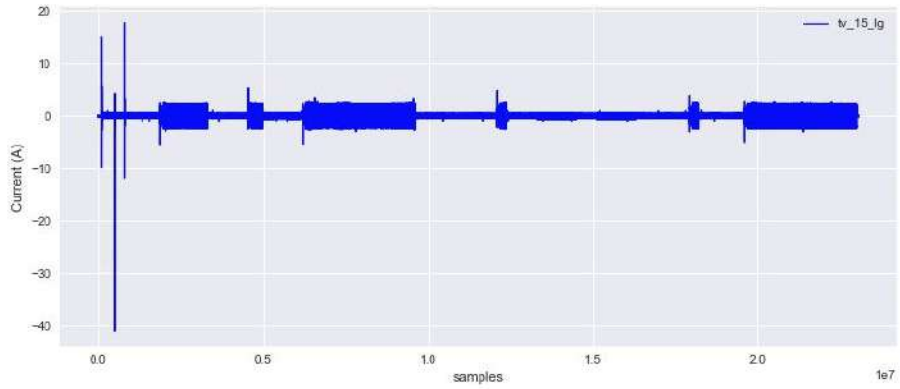

Figure 57 – Appliance 1 - Base current waveform envelope of a TV with 23.066.040 samples

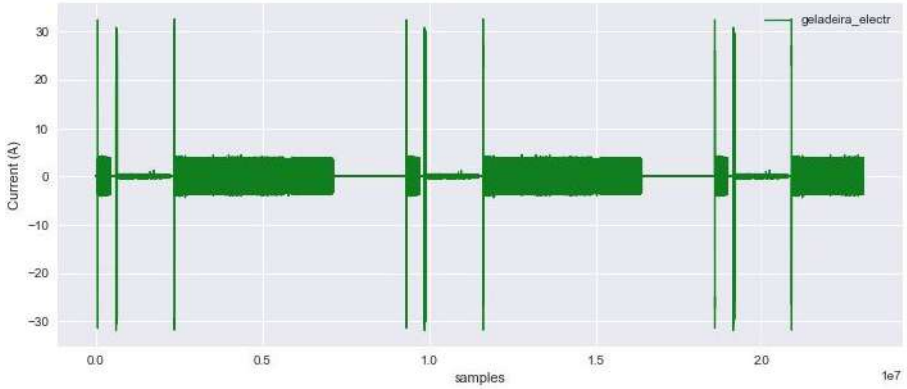

Figure 58 – Appliance 2 - Current waveform envelope of a fridge with 3 (last clipped) repetitions to achieve base current samples of Figure 57.

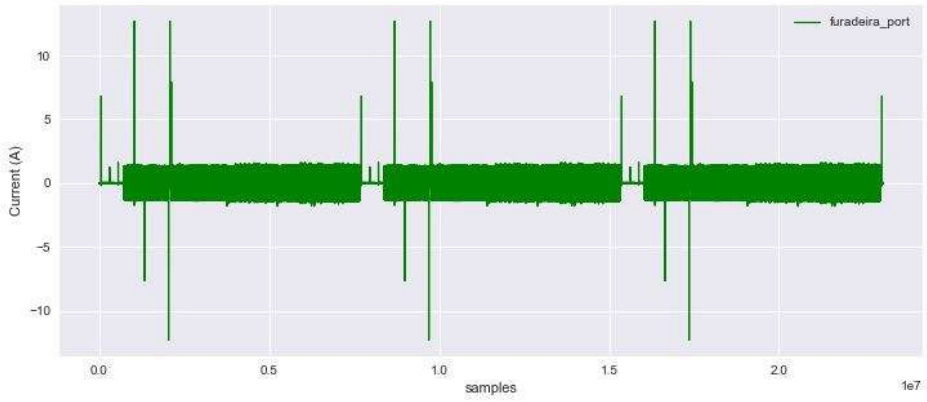

Figure 59 - Appliance 3 - Current waveform envelope of the charger for and electric drill with 4 (last clipped) repetitions to achieve base current samples of Figure 57.

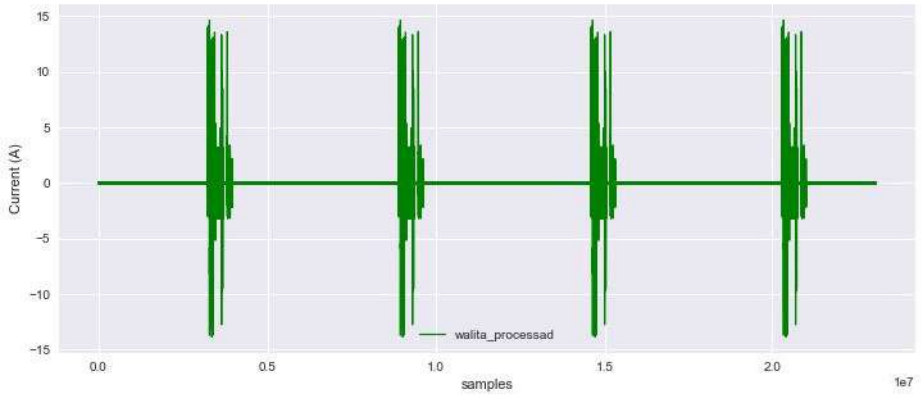

Figure 60 - Appliance 4 - Current waveform envelope of a food processor with 5 (last clipped) repetitions to achieve base current samples of Figure 57.

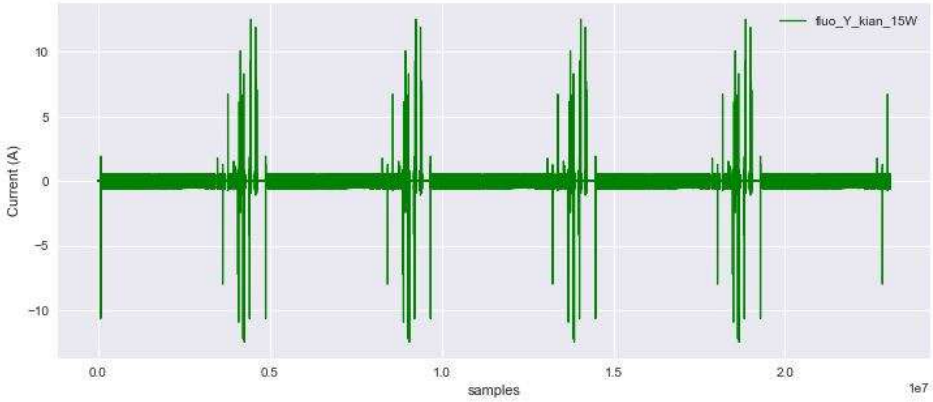

Figure 61 – Appliance 5 - Current waveform envelope of a 15W yellow fluorescent light with 5 (last clipped) repetitions to achieve base current samples of Figure 57.

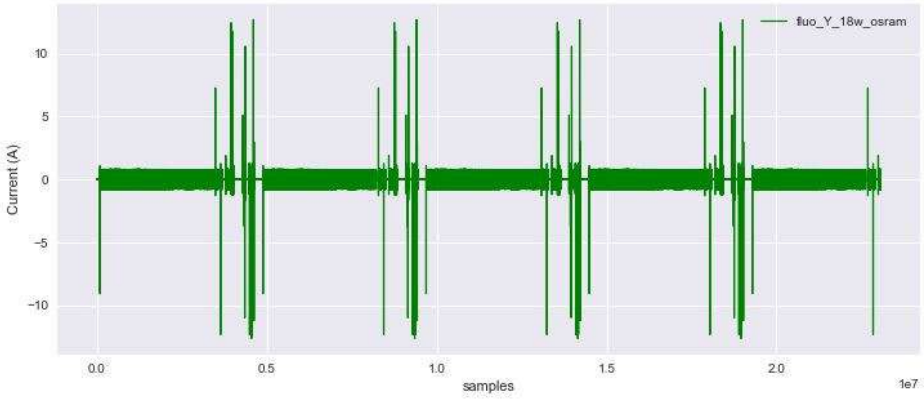

Figure 62 - Appliance 6 - Current waveform envelope of a 18W yellow fluorescent light with 5 (last clipped) repetitions to achieve base current samples of Figure 57.

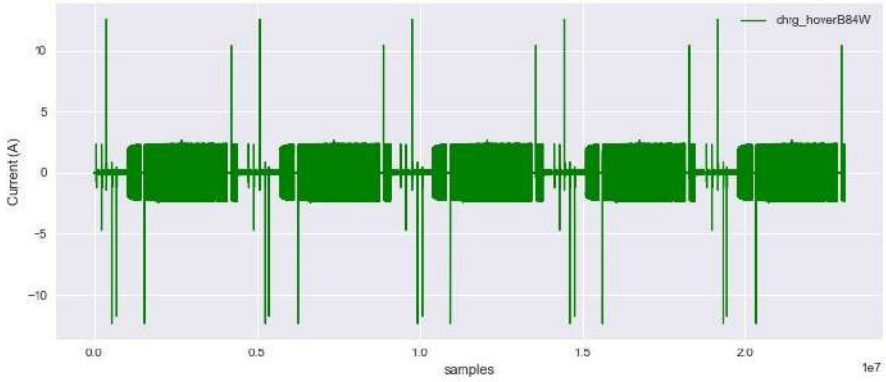

Figure 63 - Appliance 7 - Current waveform envelope of a 48W hover board charger with 5 (last clipped) repetitions to achieve base current samples of Figure 57.

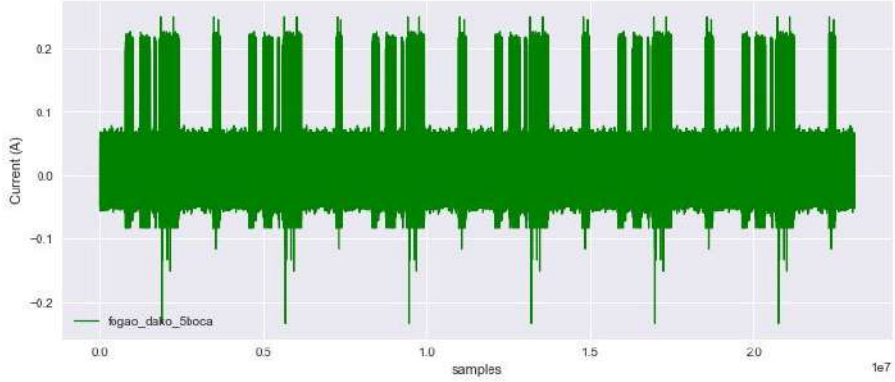

Figure 64 - Appliance 8 - Current waveform envelope of a Lighter of a stove with 7 (last clipped) repetitions to achieve base current samples of Figure 57.

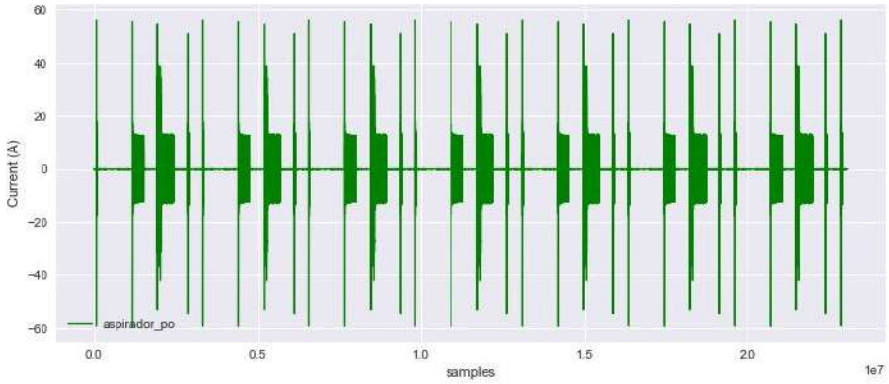

Figure 65 - Appliance 9 - Current waveform envelope of a Vacuum Cleaner with 8 (last clipped) repetitions to achieve base current samples of Figure 57.

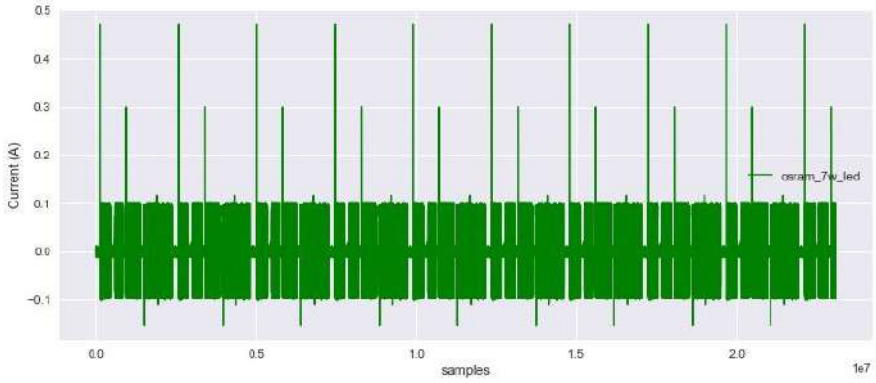

Figure 66 - Appliance 10 - Current waveform envelope of a LED 7W light with 10 (last clipped) repetitions to achieve base current samples of Figure 57.

Therefore, the waveform displayed in Figure 67, plotted a resultant synthetic aggregate of 10 non-repeating appliances, selected to be used in the training process of a machine learning algorithm and, thus, feature extraction. As mentioned in Section 3.2, the grid voltage channel was defined by using the longest sampled individual appliance voltage channel, as illustrated in blue in Figure 68.

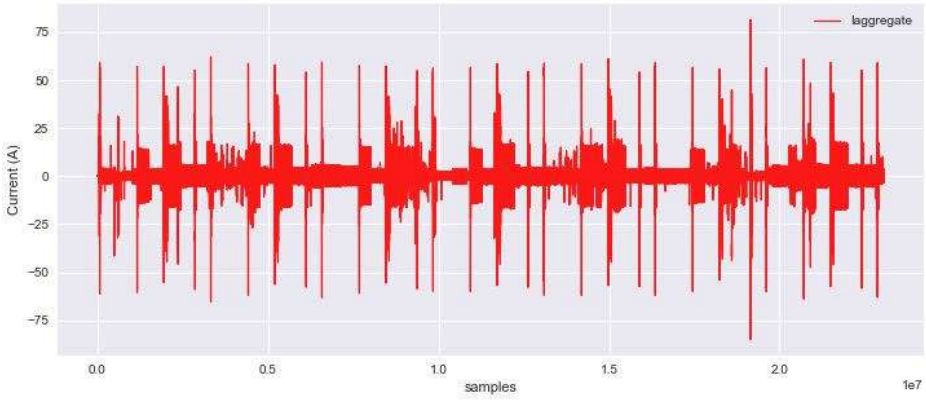

Figure 67 – Current waveform envelope result of synthetic aggregate of appliances set

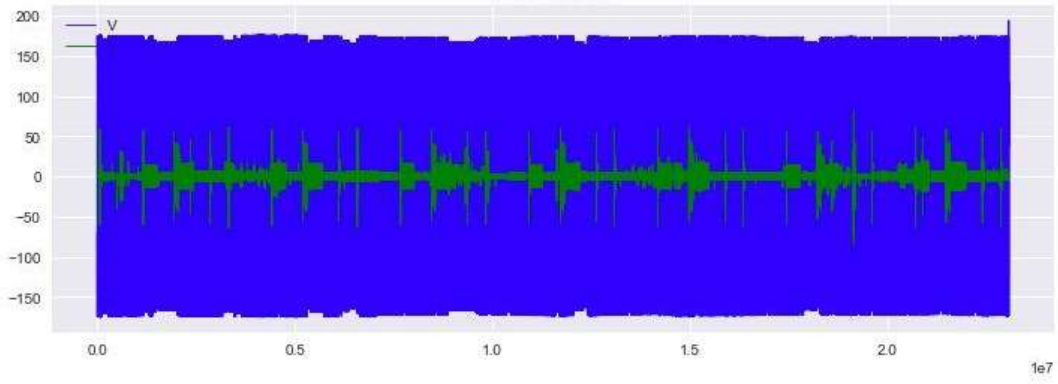

Figure 68 - Current and Voltage waveforms envelope results of synthetic aggregate of appliances set

The resample process simulated a synchronized sampling frequency of 10,020Hz (167 x 60Hz) with a complete sinusoid inside 167 points. It is important to note that there is a phase error, derived from the sampling rate, of approximately 2.1º, which is inherent to the involved frequencies. However, at this time, this thesis will not cover this associated problem.

5. Algorithms' Parameterization

As previously described in Chapter 3, classifiers need quantized outputs to fit their models correctly. For this reason, we used an unsupervised clustering algorithm for the quantization of active power (P) with performance and scaling for massive datasets. The algorithm was d*ensity-based spatial clustering of applications with noise* (DBSCAN) (Ester, et al., 1996), which was implemented by the *Hierarchical Density-Based Spatial Clustering of Applications with Noise* (HDBSCAN) library (McInnes, et al., 2017).

In order to develop feature selection, we compared sets of features, using three different algorithms: (i) Mutual Information (Ross, 2014); (ii) Multi-class AdaBoost (Zhu, et al., 2006) and, (iii) Extra-Trees (Geurts, et al., 2006). For each appliance and each feature extraction rate, those estimators calculated a different optimal set of features.

In addition, we used three different NILM classification algorithms to measure the performance of sets of features and feature extraction rates: Support Vector Machines (Campbell C., 2011) with SVC (Support Vector Classification); Logistic Regression (Freedman, 2009) and a Deep Learning technique, using TensorFlow v1.5 (GOOGLE, 2018) with KERAS high-level neural networks API (Chollet, et al., 2015).

5.1. HDBSCAN parameters

The unsupervised clustering algorithm, HDBSCAN, was forced to contain two (2) to six (6) cluster bins. This meant that all the appliances were forced to have no more than six, and never less than two, distinctive power states. To force the algorithm to discover clusters between these limits, some of the algorithm's parameters had to be changed dynamically, depending on the size of the data. Thus, we created dynamic parameters, K1 and K2 that were adjusted, depending of the results of the first HDBSCAN.

The initial value of K1 was parameterized at 0.018 and K2 at 0.038. After the first run of the HDBSCAN algorithm, if the number of clusters was between two and six, the adjustment was not required and clustering process was complete. However, if the number of clusters was greater than six, K2 was readjusted to its last value plus 0.004 and K1 was readjusted to its last value plus 0.002. Alternatively, if the number of

clusters was smaller than two, K2 was readjusted to its last value minus 0.004 and K1 was readjusted to its last value minus 0.002. The valid interval for each K1 and K2 variable was [0.001,0.9].

In order to find the correct parameterization for '**min_cluster_size**' and '**min_samples**' (the inputs of HDBSCAN that were responsible for discovering the number of clusters), we used the number of samples from each appliance (N), and multiplied it by K1 for '**min_cluster_size**' and by K2 for '**min_samples**'. All the parameterization was done heuristically, by adjusting the best parameters that worked well for the 94 studied appliances. The complete parameterization that was used in this study, for HDBSCAN, is listed in Table 16. The specific *chebyshev* metric and *boruvka* with *kdtree* algorithm operation is detailed in the HDBSCAN library web page (McInnes, et al., 2017).

Table 16 – HDBSCAN algorithm parameterization

Parameter	Description	Selected Value
min_cluster_size	The minimum size of clusters.	**min(N*K1;10)**
min_samples	The number of samples in a neighborhood for a point to be considered a core point.	**min(N*K2;5)**
metric	The metric to use when calculating distance between instances in a feature array.	**chebyshev**
alpha	A distance scaling parameter as used in robust single linkage.	**0.06**
algorithm	Select which algorithm to use.	**boruvka_kdtree**
leaf_size	The number of points in a leaf node of the tree.	**5**
core_dist_n_jobs	Number of parallel jobs to run in core distance computations. below -1: (n_cpus + 1 + core_dist_n_jobs) are used.	**-1**
cluster_selection_method	The method used to select clusters from the condensed tree. Excess of Mass (eom) algorithm to find the most persistent clusters.	**eom**
allow_single_cluster	Allow to produce a single cluster.	**True**

We also verified the magnitude of clustering errors, in active power, using HDBSCAN library. Adequate results were obtained, with quantized values almost overlapping the measured values, as shown in Figure 69. The majority of measured errors were lower than 1.0 x 10^{-12} Wh, for all the individual measurements.

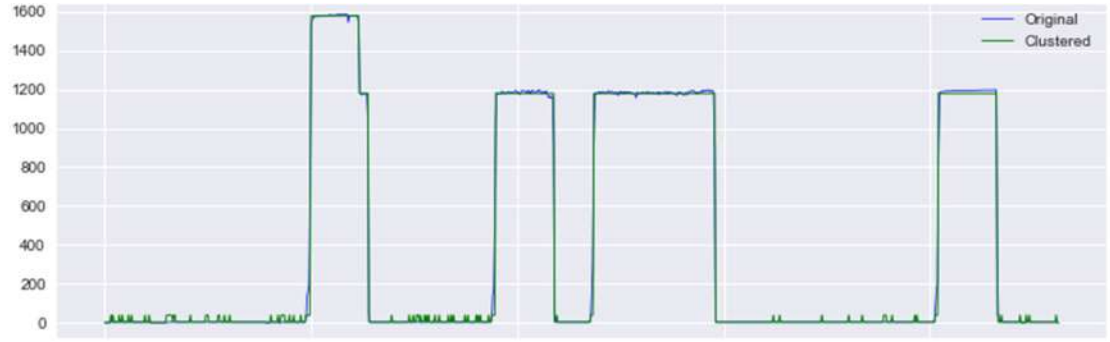

Figure 69 – Microwave oven clustered and original active power sample with 1s (60 cycles) feature measurement with HDBSCAN

However, because of the algorithm's implementation, it was not possible to compute processing for appliances' waveforms with more than 500,000 data points, on the super computer used by this study, as it consumed more than 128GB of RAM and 90GB of swap file, as shown in Figure 70. Thus, some files (18) were discarded from the processing (from 6674 files). Expunged datasets were: the washing machine, up to 6 cycles; the fridge, up to 4 cycles; the drinking fountain, up to 6 cycles; the dishwasher, 1 cycle; and the water heater, for 1 cycle.

Figure 70 – RAM and SWAP file memory limit achieved during clustering process

5.2.Feature Selection

In order to determine which minimum set of electrical features, obtained from the household aggregate data, maintained equivalent levels in NILM metrics for most of appliances, we evaluated algorithms to measure feature correlation from the input to output, called 'feature importance', and expunged all irrelevant data from the main set.

We evaluated three different algorithms: (i) Mutual Information (Ross, 2014); (ii) Multi-class AdaBoost (Zhu, et al., 2006) and, (iii) Extra-Trees (Geurts, et al., 2006),

(previously described in Section 3.5) to discover the correlation between input features and quantized active power (P), in order to assign a selected feature set. For each algorithm, this study used a greedy criterion that considered all features with a greater importance than a specified parameter. A classification algorithm was evaluated for each feature set, in order to validate whether there was a significant loss in metric scores. The feature set with the minimum number of features was then selected for posterior analysis, a process that this thesis called *Feature Ranking*. The ranking drove our choice of proposed method, based on this context and this sample universe of appliances, to find the most relevant and prevalent features to resolve the NILM problem.

5.2.1. Mutual Information parameters

The Mutual Information (MI) (Ross, 2014) algorithm, measures the correlation between variables, based on entropy estimation from k-nearest neighbors, as described in Scikit-Learn API (Pedregosa, et al., 2011). The algorithm needs to fit MI model, according to given training data; therefore, this we used the 30 features array as input; and, as the output of model, we used active power (P). The model fit used individual acquired data from each appliance. The parameterization used in the algorithm is described in Table 17.

Table 17 – Parameters used in Mutual Information algorithm

Parameter	Description	Selected Value
n_neighbors	Number of neighbors to use for MI estimation for continuous variables	**30**
random_state	The seed of the pseudo random number generator for adding small noise to continuous variables in order to remove repeated values.	**None**
featureImportanteLimit	The metric used to remove non-important features. Heuristically defined.	**0.2**

As an example; using a specific appliance, with a defined feature extraction rate, processing datasets with Mutual Information output a feature importance list as illustrated in Figure 71, and as described numerically in Table 18. Graphically, it is possible to identify 12 taller bins, over features P, I_{rms}, S, Q, CF, h1 (Fourier spectrum

magnitude of fundamental frequency), h2, h3, h4, h5, h7, and h9 (Fourier spectrum magnitude of 2nd, 3rd, 4th, 5th, 7th and 9th harmonics). It should be noted that, for each appliance and feature extraction rate studied, this algorithm evaluated a different feature importance ranking.

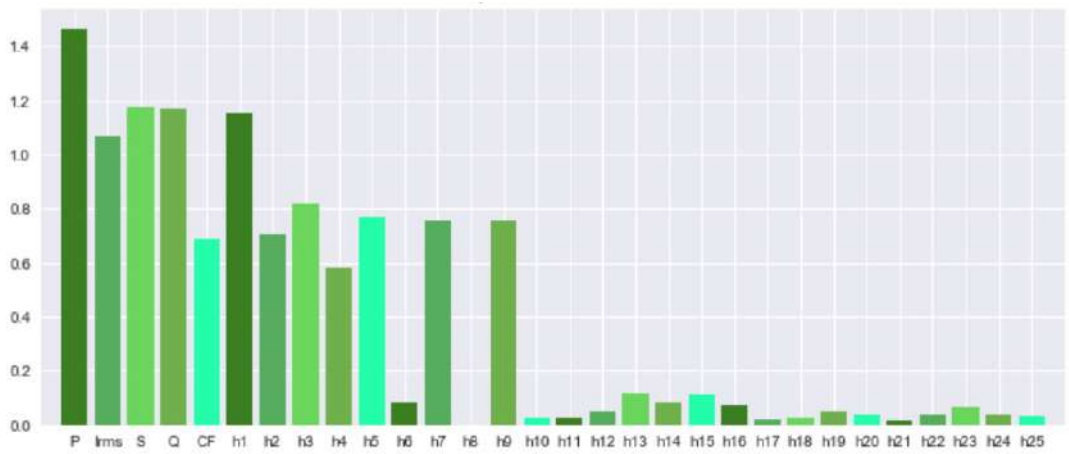

Figure 71 – Example of a Feature Importance list by Mutual Information

Samples from ten different appliances were tested initially, to parameterize safe limits so that uncorrelated features might be expunged. Therefore, for this algorithm, this study heuristically parameterized 0.2 as the minimum importance limit to determine the size of a feature set.

Table 18 – Feature importance ranking by Mutual Information

Feature	Importance	Feature	Importance	Feature	Importance
P	1.465079	**CF**	0.690280	**h24**	0.040969
S	1.176086	**h4**	0.579836	**h22**	0.038349
Q	1.168983	**h13**	0.118458	**h20**	0.036345
h1	1.153246	**h15**	0.113563	**h25**	0.032411
Irms	1.068231	**h6**	0.084500	**h18**	0.029952
h3	0.819415	**h14**	0.082526	**h10**	0.029296
h5	0.769135	**h16**	0.072623	**h11**	0.027822
h7	0.759107	**h23**	0.066127	**h17**	0.019246
h9	0.759053	**h19**	0.051066	**h21**	0.017410
h2	0.705037	**h12**	0.049898	**h8**	0.000000

5.2.2. Multi-class AdaBoost parameters

The Multi-class AdaBoost (Zhu, et al., 2006) algorithm is a meta-estimator that fits a classifier to the original dataset and additional copies. The weights of the incorrectly classified instances are adjusted to focus subsequent classifiers on those incorrect classifications, theoretically the more complex cases, in order to fit the model. Its implementation is based on AdaBoost SAMME.R (Zhu, et al., 2006). The parameterization that was used in the algorithm is described in Table 19.

Table 19 – Parameters used in Multi-class AdaBoost algorithm

Parameter	Description	Selected Value
n_estimators	The maximum number of estimators at which boosting is terminated	**600**
learning_rate	Learning rate shrinks the contribution of each classifier	**1**
base_estimator	The base estimator from which boosted ensemble is built.	**Decision Tree**
algorithm	SAMME.R or SAMME boosting algorithm	**SAMME.R**
featureImportanteLimit	The metric used to remove non-important features. Heuristically defined.	**0.001**

As an example; from a specific appliance, with a defined feature extraction rate, processing datasets with Multi-class AdaBoost output a feature importance list as illustrated in Figure 72, and as described numerically in Table 20. Graphically, it is possible to identify 4 bins, over features P, I_{rms}, Q and h5 (Fourier spectrum magnitude of 5^{th} harmonic). It should be noted that, for each appliance and feature extraction rate studied, this algorithm also evaluated a different feature importance ranking.

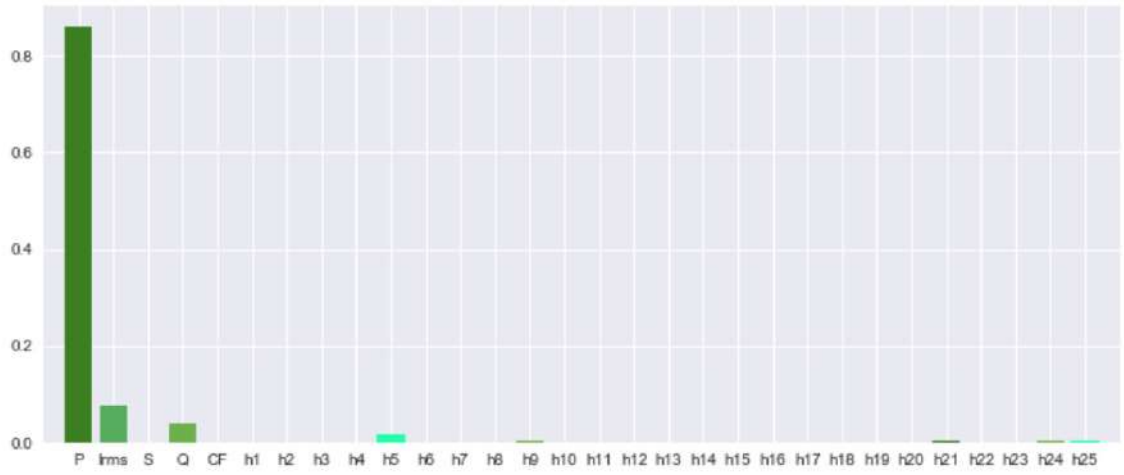

Figure 72 – Example of a Feature Importance by AdaBoost

Samples from ten different appliances were tested initially, to parameterize safe limits so that uncorrelated features might be expunged. Therefore, for this algorithm, this study heuristically parameterized 0.001 as the minimum importance limit to determine the size of a feature set.

Table 20 – Feature importance ranking by AdaBoost SAMME.R

Feature	Importance	Feature	Importance	Feature	Importance
P	0.862223	**h1**	0.000000	**h13**	0.000000
Irms	0.077405	**h2**	0.000000	**h14**	0.000000
S	0.035278	**h3**	0.000000	**h15**	0.000000
h5	0.015731	**h4**	0.000000	**h16**	0.000000
Q	0.002885	**h6**	0.000000	**h18**	0.000000
h24	0.002841	**h7**	0.000000	**h19**	0.000000
h21	0.002181	**h25**	0.000000	**h20**	0.000000
h17	0.001454	**h9**	0.000000	**h22**	0.000000
h8	0.000000	**h11**	0.000000	**h23**	0.000000
CF	0.000000	**h12**	0.000000	**h10**	0.000000

5.2.3. Extra-Trees parameters

Extremely randomized trees, or Extra-Trees (Geurts, et al., 2006), is part of the tree-based ensemble methods, as described in Scikit-Learn API (Pedregosa, et al., 2011). The parameterization that was used in the algorithm is described in Table 21.

Table 21 – Parameters used in Extra-Trees algorithm

Parameter	Description	Selected Value
n_estimators	The number of trees in the forest.	**600**
max_depth	The maximum depth of the tree.	**None**
min_samples_split	The minimum number of samples required to split an internal node.	**2**
min_samples_leaf	The minimum number of samples required to be at a leaf node.	**1**
featureImportanteLimit	The metric used to remove non-important features. Heuristically defined.	**0.01**

As an example, from a specific appliance, with a defined feature extraction rate, processing datasets with Extra-Trees output a feature importance list as illustrated in Figure 73, and as described numerically in Table 22. Graphically, is possible to identify 11 taller bins, over features P, Irms, S, Q, CF, h1 (Fourier spectrum magnitude of fundamental frequency), h2, h3, h5, h7, and h9 (Fourier spectrum magnitude of 2^{nd}, 3^{rd}, 5^{th}, 7^{th} and 9^{th} harmonics). It should be noted that, that for each appliance and feature extraction rate studied, this algorithm also evaluated a different feature importance ranking.

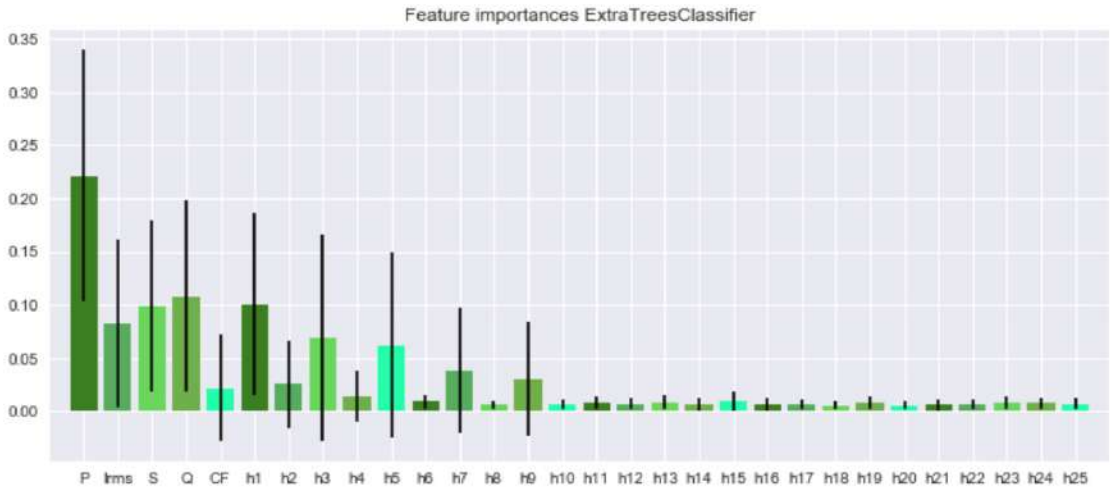

Figure 73 – Example of a Feature Importance given by Extra-Trees

Samples from ten different appliances were tested initially, to parameterize safe limits so that uncorrelated features might be expunged. Therefore, for this algorithm, for this algorithm, this study heuristically parameterized 0.01 as the minimum importance limit to determine the size of a feature set.

Table 22 – Feature importance ranking by Extra-Trees

Feature	Importance	Feature	Importance	Feature	Importance
P	0.221036	**CF**	0.021795	**h16**	0.006819
Q	0.108412	**h4**	0.014238	**h12**	0.006768
h1	0.100638	**h15**	0.009446	**h14**	0.006622
S	0.098660	**h6**	0.008836	**h10**	0.006378
Irms	0.082065	**h13**	0.008685	**h22**	0.006368
h3	0.068902	**h11**	0.008181	**h17**	0.006134
h5	0.062321	**h23**	0.008097	**h8**	0.006132
h7	0.038320	**h19**	0.007772	**h21**	0.005925
h9	0.030587	**h24**	0.007355	**h18**	0.005700
h2	0.025310	**h25**	0.007112	**h20**	0.005387

5.3. Classification methods

As described in Section 3.6, we proposed evaluating NILM problem solving with three different classification algorithms, each with its particular traits: (i) Logistic Regression, described by Freedman (Freedman, 2009) as a statistical method; (ii) Support Vector Machines with SVC (Support Vector Classification), described by Campbell (Campbell C., 2011) as a non-probabilistic classifier with non-linear classification capabilities; and, (3) a developed non-linear classifier using Deep Learning techniques, with TensorFlow v1.5 (GOOGLE, 2018) and KERAS high-level neural networks API (Chollet, et al., 2015).

Each algorithm required extensive parameterization before it began to converge. The final parameters that were used for each method were presented in Sections 5.3.1, 5.3.2 and 0. The models were trained using thirty chosen features as input, and the quantized active power (P) was the output of the classifiers. The training process used 90% of the synthetic dataset as training data and 10% was kept aside as validation data (unseen data) for the metrics' evaluation. The appliance and the feature extraction rate were chosen randomly to display examples of the classification results of the tested algorithms. The appliance example, "Desktop PC", displayed four different quantized power states, at the desired feature extraction rate.

We preprocessed the data for each appliance before using it in the training. For the BRAD dataset, where individual appliances were sampled, we established **offset** and **gain** values for each extracted feature, for each appliance. Therefore, each **offset** and **gain** value were computed to provide *zero mean* and scaling to *unit variance*, thus forcing all input to have the same order and to be centered on zero.

5.3.1. Logistic Regression parameters

The selection of parameters, listed in Table 23, was tuned to a Logistic Regression implementation that was built on Scikit-Learn API (Pedregosa, et al., 2011).

Table 23 – Parameters used in Logistic Regression classifier

Parameter	Description	Selected Value
penalty	Used to specify the norm used in the penalization.	**L2**
tol	Tolerance for stopping criteria.	**1e-4**
C	Inverse of regularization strength.	**1e-4**
solver	Algorithm to use in the optimization problem.	**lbfgs**
max_iter	Maximum number of iterations taken for the solvers to converge.	**5000**
multi_class	The loss minimized is the multinomial loss fit across the entire probability distribution	**multinomial**
warm_start	Reuse the solution of the previous call to fit as initialization.	**True**

Figure 74 shows a Logistic Regression classifier model prediction. It is possible to identify the algorithm's predictions of active power (P), on the y axis, inside a mixture of ten other appliances (displayed in scale in Figure 75) over samples on the x axis.

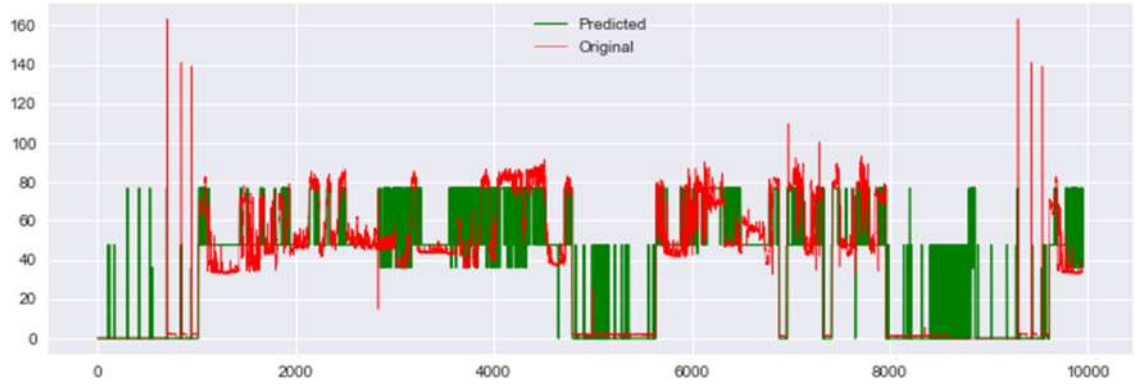

Figure 74 – Logistic Regression predictions' zoom for a Desktop PC

These results were obtained using a heuristic recurring approach for training the same model, ten times, and by setting the **warm_start** parameter to *True*. This option reuses the previous model as the initial condition of next model, which showed the best results for most of the appliances.

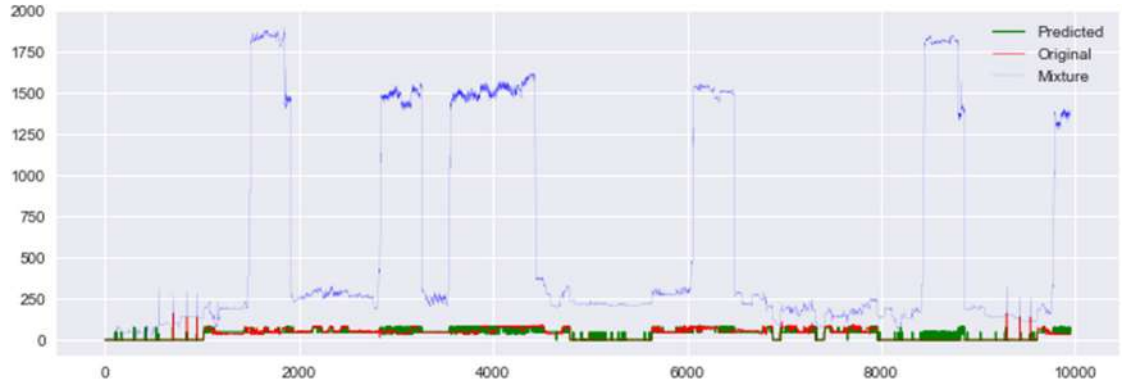

Figure 75 – Logistic Regression predictions for a Desktop PC inside a mixture

5.3.2. Support Vector Machines parameters

The selection of parameters, listed in Table 24, was tuned to a Support Vector Machines implementation based on LIBSVM (Chang, et al., 2013). The classifier, shown, is difficult to scale when a dataset is larger than 50,000 points, hence, (and as previously described in Section 2.8) fit (training) and predict times were also collected to allow a further comparison using sets of classification algorithms and appliances mixture datasets.

The parameterization required extensive fine-tuning to return adequate results, showing more sensitivity in the parameters to optimize than Logistic Regression did. However, the outcomes were promising, and it recovered disaggregated quantized active power accurately in most assorted mixture scenarios.

Table 24 – Parameters used in SVM classifier

Parameter	Description	Selected Value
svm_type	Set the type of SVM.	**0 (C-SVC)**
kernel_type	Set type of kernel function.	**2 (radial basis function)**
degree	Set degree in kernel function.	**3**
cost	Set the parameter C of C-SVC.	**1.0**
cachesize	Set cache memory size in MB.	**8192**
epsilon	Set tolerance of termination criterion.	**0.002**

Figure 76 shows an SVC model prediction. It is possible to identify the algorithm's predictions of active power (P), on the y axis, inside a mixture of ten other appliances (displayed in scale in Figure 77) over samples on the x axis.

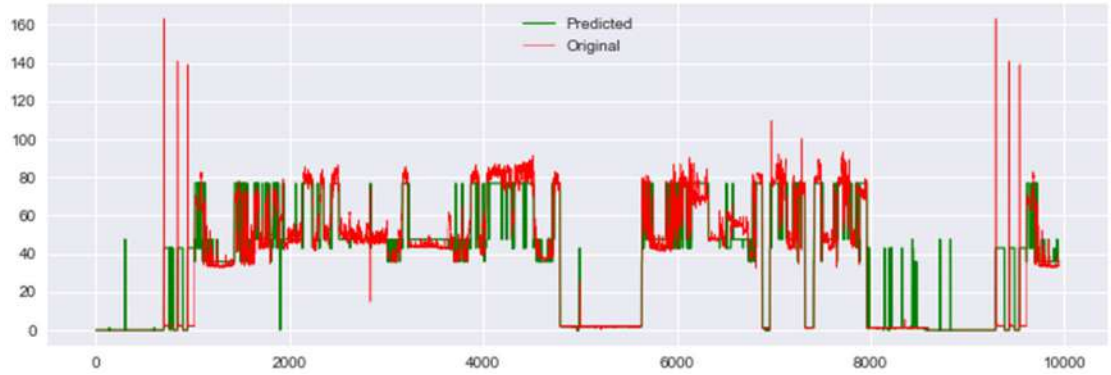

Figure 76 – Support Vector Machine predictions zoom for a Desktop PC

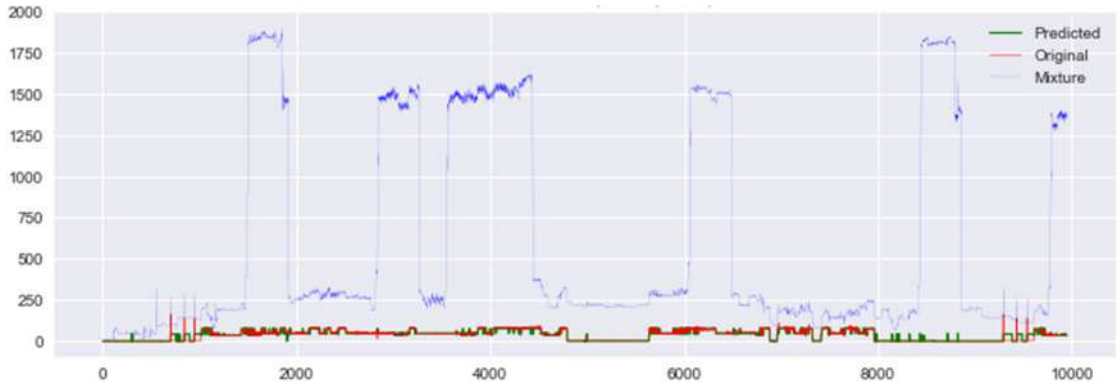

Figure 77 - Support Vector Machine predictions for a Desktop PC inside a mixture

5.3.3. Deep Learning architecture

The proposed classifier was built on TensorFlow v1.5 API (GOOGLE, 2018) and KERAS high-level neural networks API (Chollet, et al., 2015). Actual Deep Leaning development exhibits advanced layers such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). These layers constitute an advantage, when the input data is not randomly presented, but is organized in order, such as in a sliding window approach, or as a 'curriculum learning' strategy (Zaremba, et al., 2014).

However, the evaluated implementations of Logistic Regression and SVM classifiers were not able to scale the massive amounts of windowed data to fit their models. Therefore, the input data for all the classifiers were standardized to reduce the input to **one random sample of a feature set** at a time: simultaneously reducing the improvements of those advanced layers. Thus, the proposed architecture of Deep Learning used only 'Dense' fully connected layers, to seek a fair comparison of the algorithm's performance.

As previously stated in Section 3.6, no scientific method exists to describe the "how to" for managing the size of each layer, or the number of required layers, not which activations, initializers and optimizers or regularization methods your Deep Neural Network will require. For this reason our developed model was designed heuristically by validating related architectures, by comparing similar complexity problems, and thereafter, by recursively verifying the networks' convergence to a subsample of datasets. The developed architecture is listed in Table 25.

Figure 78, shows a Deep Learning model prediction. It is possible to identify the algorithm's predictions of active power (P), on the y axis, inside a mixture of ten other appliances (displayed in scale in Figure 79) over samples on the x axis.

The presented parameterization of selected models was tested on mixture datasets in order to compare the performance of each algorithm over the feature extraction rate periods and between themselves.

Table 25 – Architecture used by proposed Deep Learning classifier

Layer	Description	Size
Dense	Input	**Size of feature set**
Dense	Linear	**128**
Batch Normalization	Used only during training.	**128**
Dense	ReLU	**256**
Dense	ReLU	**512**
Dropout	Used only during training	**0.5**
Dense	ReLU	**512**
Dense	ReLU	**256**
Dense	ReLU	**128**
Dense	Softmax	**Size of quantization clusters**

The results of the simulations are presented in Chapter 6.

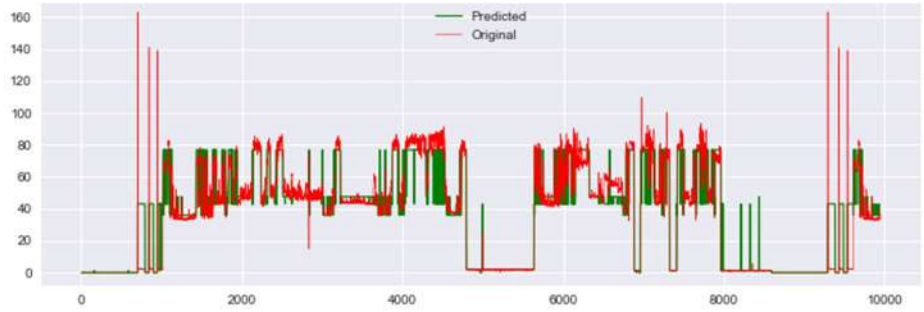

Figure 78 – Deep learning classifier predictions zoom for a Desktop PC

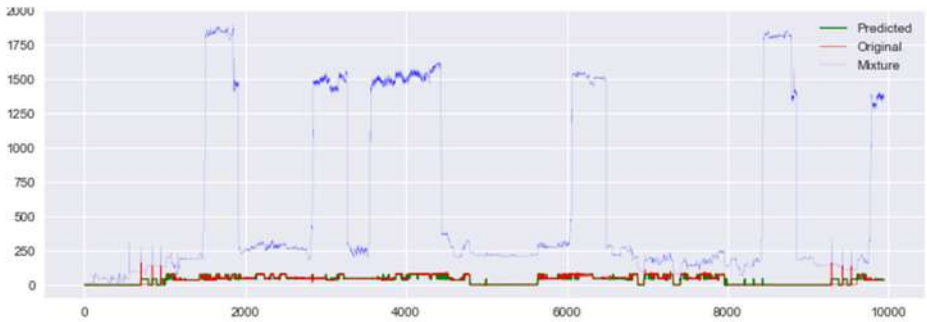

Figure 79 – Deep learning classifier predictions for a Desktop PC inside a mixture

6.Simulations and Results

This Section presents the practical results of this study. As mentioned earlier (in Chapter 3), huge combinations of datasets, features, feature extraction rates, metrics, and classifier tests repeated to be able to understand the problem completely.

The simulation process generated more than 300GB of data, including the clustering results, the feature selection sets, a range of feature extraction rates, the ground truth data, all trained models and multiple mixture streams. The synthetic mixtures used an HDF5 file format (THE HDF GROUP, 1997) as their default file format for this study, which performs well for Big Data storage and for data compression.

A summary of the evaluated simulations gives: ninety-four appliances; three NILM classification methods; four feature sets, used for feature selection; seventy-one different feature extraction rates, ten distinctive mixtures per appliance, including the combination of ten non-repeated appliances, and eight distinct metrics scores. These are all illustrated in Figure 80.

Figure 80 – Numbers involved in simulations

As previously described, the metrics selected for evaluation NILM performance were:

I. **Accuracy (A)**
II. **F-measure / F-score (F1)**
III. **Precision (P)**
IV. **Recall (R)**
V. **Total energy correctly assigned (TECA)**
VI. **Normalized error in assigned power (NEAP)**
VII. **Fit Time (training)**
VIII. **Predict Time**

The appliances' ground truth data used 6,656 quantized active power (P) single files, containing the information from each of the ninety-four appliances over the

seventy-one feature extraction rate ranges. For the mixture files, 66,030 individual pairs of appliance-mixture files were created, containing mixture and ground truth data for all seventy-one unique feature extraction rates to be tested. Each one of the three NILM algorithms computed training and predictions four times; once for each appliance-mixture pair, which resulted in 762,360 total training and predictions steps, with eight computed metrics.

As part of the first analysis of the proposed methodology, we opted to study a subset of the appliances' universe, containing fifteen devices, and to evaluate metrics in only three of those ten generated mixture files, in order to present results. The appliances were chosen from the set, empirically, with the aim of selecting each one from a different category, including: (1) an X-Box One; (2) an LCD monitor; (3) a notebook; (4) an air conditioner; (5) a fridge; (6) a washing machine; (7) a microwave oven; (8) a hair dryer; (9) a 60W halogen bulb; (10) a 23W fluorescent bulb; (11) a 10W LED bulb; (12) a 50 inch plasma TV; (13) a ceiling fan with three speeds; (14) a desktop computer; and (15) a clothes iron.

This produced 115,020 training-predictions steps, which evaluated the selected eight metrics, and represented about 15.1% of the full training steps, as illustrated in Figure 81. Nevertheless, the time for training and prediction phases took more than three months of 100% workload (when possible) to complete the tasks in our machine learning computer (MLC).

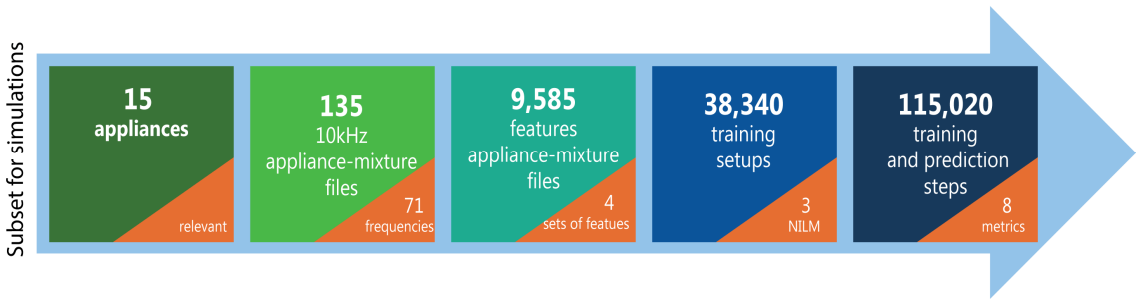

Figure 81 – Selected subset of data for simulations

The MLC configuration was a Dual Intel Xeon E5-2698v4 CPU (forty processors with eighty total threads); 128GB of RAM; 6TB RAID Storage, including 2TB of RAID SDD and 4x NVidia GTX-1080 with 8GB of RAM each (total of 32GB). The results for each appliance's simulations are presented in the next Section.

6.1. Detailed analysis of one appliance

The first analysis to be evaluated is the sensitivity of each one of the eight analyzed metrics, over the feature extraction rates. Figure 22 has already illustrated this as a scheme to find the *minimum feature extraction rate,* and Section 3.4, described how the proposed method used the full set of thirty features, to train and predict load disaggregation with the three available synthetic mixture datasets. For each *best metric score*, which we expected to be at the *maximum feature extraction rate* (defined as a number of 60Hz cycles), we proposed to establish an *acceptable metric score* value as a 5% set point below the maximum, whereas the feature extraction period (displayed in cycles) advanced. We chose the X-Box One as the appliance to examine for this detailed analysis. The graphs and tables from the analysis of the other fourteen appliances are displayed in Appendix C.

Therefore, for the appliance, the X-Box One, the evaluated metrics, which were obtained from the models, over the feature extraction rate, are illustrated in Figure 82. They perfectly fit the predictions of Figure 22, which displays a drop in performance whereas the number of *feature extraction rates* (in cycles) rises. Each prediction used three different mixture datasets; so, for each evaluated model, three curves were adjusted to show the spread of metrics scores, using error bar graphs. The filled curves, one for each NILM classifier, in Figure 82, show the worst case prediction results.

In the first step of the analysis, this study extracted useful information from Figure 82, and created a summary for the X-Box's metric scores. Table 26 summarizes all the extracted information, for each trained model (Deep Learning, SVM and Logistic Regression), and for each metric, using the following column headers:

(I) - "***max%***" - This column represents the *best metric score*, which was expected (but not required) to be at the *maximum feature extraction rate.* This column displays the highest score (higher is better) value (between 0 and 100%) for each metric;

(II) - "***min%***" - This column shows the *best metric score*, which was expected (but not required) to be at the *maximum feature extraction rate.* This column displays the lowest score (lower is better) value (between 0 and 100%) for each metric;

(III) - *"**-5%**"* - This column shows the *desired metric score*, defined as the **5%** cut-off point **below** the highest *best metric score* (displayed in "***max%***"). This column shows the **minimum desired metric value** (between 1 and 95%) of each metric for each trained model and feature set. It is used in the metrics: **Accuracy (A), F-measure (F1), Precision (P), Recall (R),** and **Total energy correctly assigned (TECA)**.

(IV) - *"**+5%**"* - This column shows the *desired metric score*, defined as the **5%** cut-off point **above** the lowest *best metric score* (displayed in "***min%***"). This column shows the **maximum desired value** (between 5 and 100%) of each metric for each trained model and feature set. It is used in the **Normalized error in assigned power (NEAP)** metric;

(V) - *"**cycle**"* - This column displays a number that represents the *minimum feature extraction rate* (in number of 60Hz cycles), i.e., the point on the **x** *axis* where the *desired metric score,* shown in *"**+5%**"* or *"**-5%**"*, is obtained; and,

(VI) - *"**feasible time cycle**"* – As shown in Figure 82, specifically at "Time to Train Model" and "Time do Predict Model" graphs, the compute time decreases exponentially, whereas the *feature extraction rate* (displayed in cycles) rises. Therefore, this column displays the minimum number of cycles for *feature extraction rate*`, whereas the algorithm run time presents scalability, that is, it is lower than 10% of the maximum compute time. It is used in **Fit Time** and **Predict Time** metrics.

Table 26 – X-Box One / Metrics summary

	Accuracy			F-Measure			Precision			Recall		
with Full Features	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*
Deep Learning	95	90	35	92	87	20	93	88	16	92	87	25
SVM	75	70	10	70	65	3	92	87	3	65	60	4
Logistic Regression	60	55	15	42	37	2	52	47	2	42	37	5

	TECA			NEAP			Train Time	Predict Time
with Full Features	*max%*	*-5%*	*cycle*	*min%*	*+5%*	*cycle*	*feasible time cycle*	*feasible time cycle*
Deep Learning	97	92	70	5	10	90	4	2
SVM	94	89	7	5	10	90	5	10
Logistic Regression	85	80	10	25	30	17	2	1

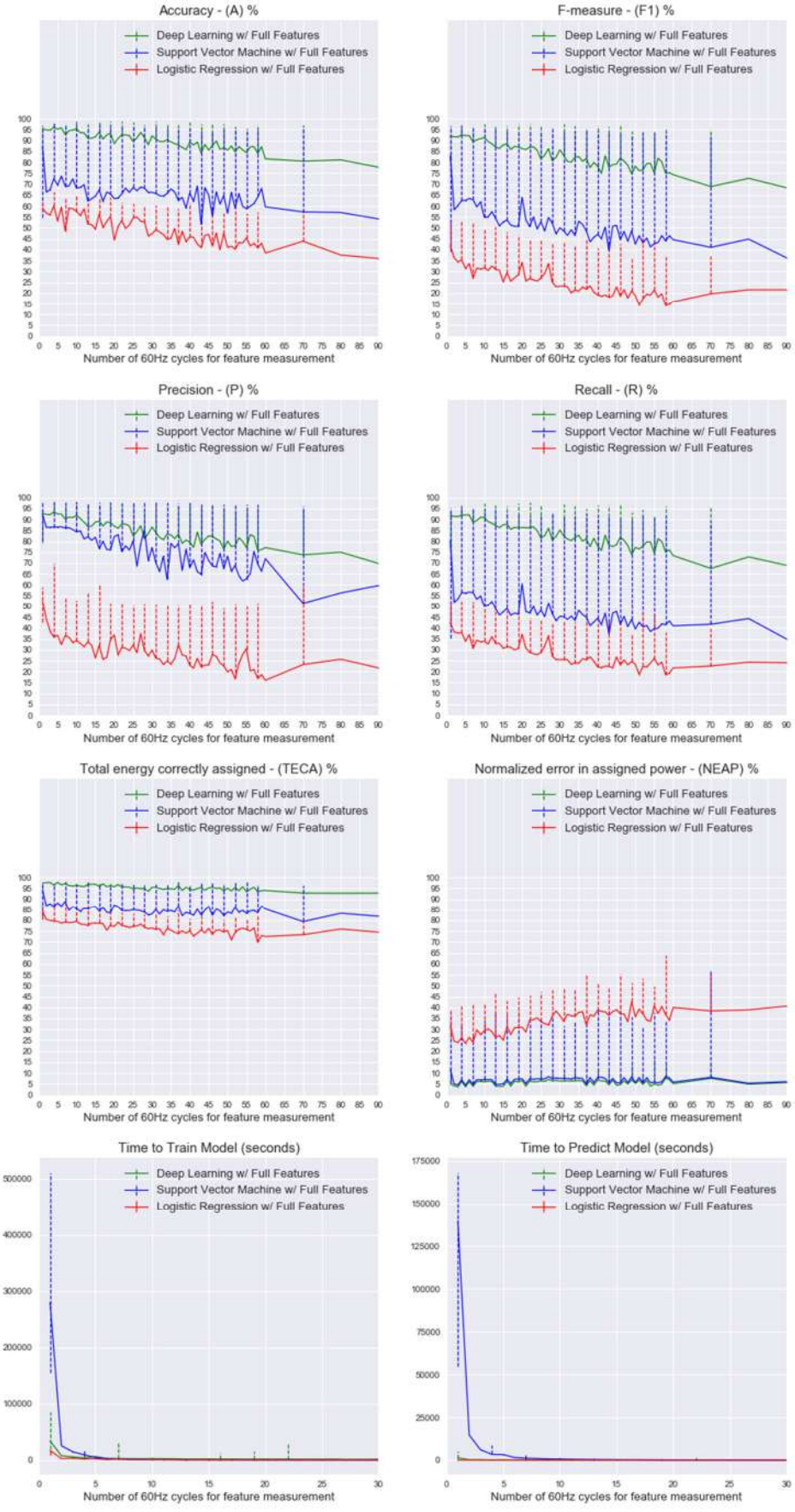

Figure 82 – X-Box One / NILM Methods comparison with full features set

The second step of the proposed method was to analyze the summary. In our studied scenario and mixtures, and using our algorithm implementations and parameterization as described in Chapter 5, the *Deep Learning* technique presented the highest scores for the worst case scenario, at column *"**cycle**"* in evaluated metrics: **16 cycles**, against 3 from SVM and 2 from Logistic Regression. It also presented the highest scores for the worst case scenario at column *"**-5%**"*, representing the *desired metric score* for classification metrics' scores: 87% for Deep Learning, against 60% from SVM and 37% for Logistic Regression; The same method also gave the major score for the worst case scenario, at column *"**-5%**"* for metric **Total Energy Correctly assigned (TECA)**: 92%, against 89% from SVM and 80% from Logistic Regression. It also displayed the lowest score at column *"**+5%**"* for metric **Normalized error in assigned power (NEAP)**: 10%, tying with SVM and bettering Logistic Regression, with 30%. Thus, after an evaluation of the performance of the NILM classification algorithms, it is clear that the *Deep Learning* method was the most successful classifier that was applied in the evaluated X-Box One scenario.

The third step of the proposed method was to test the feature sets, using the Mutual Information (Ross, 2014), Multi-class AdaBoost (Zhu, et al., 2006) and, Extra-Trees (Geurts, et al., 2006) algorithms as previously described and parameterized in Chapter 5. The metrics for each feature set were evaluated using the *Deep Learning* method, as shown in Figure 83. AdaBoost displayed a significant loss of performance, with high variance along *feature extraction rates*. Mutual Information and Extra-Trees displayed similar results over the complete set of features. In the final step of the detailed analysis, we compared results and created a **summary** for posterior feature selection and feature extraction rate analysis.

The Extra-Trees algorithm evaluated 18 features, representing a data reduction of close to 40% (compared to the full set). Mutual Information assigned too much feature importance to features, and therefore displayed 28 features. AdaBoost evaluated 10 features, which represented almost a 67% reduction. However, AdaBoost results could not be used because of performance loss.

The complete list of algorithms and their detailed feature sets are shown in Table 27. Extra-Trees presented the lowest number of features, with acceptable scores in metrics. A summary of the selected solution for feature selection, using Extra-Trees, and the

feature extraction rates, using the results obtained from Deep Learning's lowest "cycle" score (16 cycles), for the studied appliance, are summarized in Table 28.

Table 27 – X-Box One / Summary of Feature Selection

Deep Learning	**Feature List**																														
16 cycles	*Features*						*Fourier Harmonics*																								data
Method	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	Size	% red.
Full set	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	30	
Extra-Trees	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓		✓		✓		✓				✓		✓		✓		✓		✓	18	40%
M. Inform.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓	28	7%
AdaBoost	✓	✓	✓			✓	✓	✓										✓		✓		✓							✓		10	67%

Table 28 – X-Box One / Summary of complete solution

Solution	Method	Feature Selection Method	Num. Features	F. Extraction Rate
	Deep Learning	Extra-Trees	18	16 cycles

Selected Features	*Features*						*Fourier Harmonics*																							
	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓		✓		✓		✓				✓		✓		✓		✓		✓

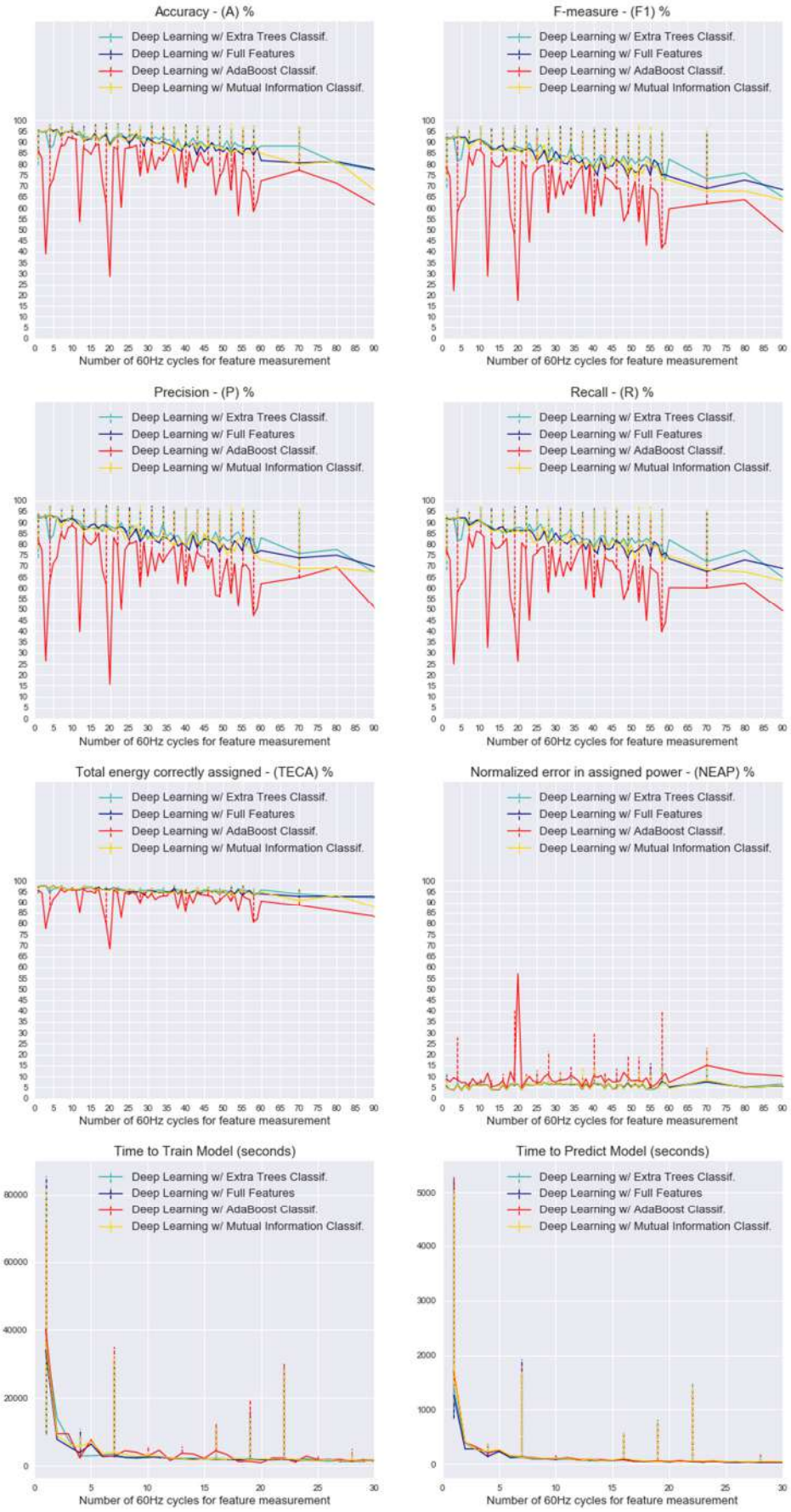

Figure 83 – X-Box One / Feature Selection methods comparison

Using a similar analysis, as above, over our studied universe of 15 appliances, we present an evaluated summary of all figures and tables in Table 29.

Table 29 – Summary of appliances tables and figures

Appliance	NILM methods comparison figure	Metrics Summary table	Feature Selection methods comparison figure	Summary of feature selection table	Summary of complete solution table
Notebook	Figure 93	Table 36	Figure 94	Table 37	Table 38
LCD Monitor	Figure 95	Table 39	Figure 96	Table 40	Table 41
Air Conditioner	Figure 97	Table 42	Figure 98	Table 43	Table 44
Fridge	Figure 99	Table 45	Figure 100	Table 46	Table 47
Washing machine	Figure 101	Table 48	Figure 102	Table 49	Table 50
Microwave oven	Figure 103	Table 51	Figure 104	Table 52	Table 53
Hair dryer	Figure 105	Table 54	Figure 106	Table 55	Table 56
Halogen bulb	Figure 107	Table 57	Figure 108, Figure 109	Table 58	Table 59
Fluorescent bulb	Figure 110	Table 60	Figure 111	Table 61	Table 62
LED bulb	Figure 112	Table 63	Figure 113	Table 64	Table 65
Plasma TV	Figure 114	Table 66	Figure 115	Table 67	Table 68
Ceiling fan	Figure 116	Table 69	Figure 117	Table 70	Table 71
Desktop PC	Figure 118	Table 72	Figure 119	Table 73	Table 74
Clothes Iron	Figure 120	Table 75	Figure 121	Table 76	Table 77

6.2. Analysis of feature extraction rate

A general analysis of our studied sampled universe of appliances revealed important insights and information about NILM hardware implementations. Using the data, evaluated in Section 6.1, this Section focuses on an analysis related of the feature extraction rate. The first step we proposed is to create a summary of the minimum feature extraction rate (defined as a number of 60Hz cycles) found for each studied appliance. The resulting compilation is shown in Table 30.

Table 30 – Summary of feature extraction rates

Appliance	Feature extraction rate (in number of 60Hz cycles) FER_{cycles}
Clothes Iron	27
Desktop PC	53
Ceiling fan	20
Plasma TV	23
LED bulb	52
Fluorescent bulb	17
Halogen bulb	35
Hair dryer	60
Microwave oven	25
Washing machine	42
Fridge	17
Air conditioner	**16**
X-Box One	**16**
LCD Monitor	70
Notebook	100

The air conditioner and X-Box One displayed $\min(FER_{cycles})$ as **16**. This means that, for each 16 sampled cycles of 60Hz, the feature extraction algorithms needed to evaluate feature measurements and to store the information. As previously explained in Chapter 3, **lower** values of FER_{cycles} means a **higher throughput** of the data and, consequently, a higher rate of feature measurements.

This result (16 cycles) indicates a period of 266.67ms, or 3.75Hz, for the feature extraction rate. This rate is related to a minimum drop of 5% in the evaluated metric. Given our previous premises, conservatively, this rate reduces the *feature extraction rate* by 16 times. Choosing larger values of FER_{cycles} can reduce the number of times each measurement needs to be evaluated and sent over telecommunication networks for storage and off-line computation. Therefore, as a second proposed step, it is important to identify the central tendency measure for the *feature extraction rates* of the studied appliances. This will indicate the value that we could potentially choose, without overly affecting the performance of obtained metric scores. If we apply an ascending sort to the FER_{cycles} array, the resultant is:

$$FER_{cycles}{}^{sorted} = [16,16,17,17,20,23,25,\mathbf{27},35,42,52,53,60,70,100]$$

Thus, the evaluated **median** value of the array, over the listed appliances, targets to a rate of 27 cycles (or a period of 450ms). The use of this median *feature extraction rate* value (27 cycles) reduced the data transfer to 68.7%, compared to the conservative choice (16 cycles). Thus, as the third step of the analysis, half of the listed appliances needed to be re-evaluated, in order to identify whether a change in feature extraction rate could lead to a huge and unwanted loss of metrics.

The affected appliances were reevaluated for 450ms (from the figures shown in Appendix C) to verify whether a variation in the metrics scores could be significant for our results. After reanalyzing the appliances, shown in Table 31, it was clear that the majority of variations were insignificant, particularly in TECA and NEAP metrics, with a variation in scores of lower than 1%. These results allowed us to make a feature extraction rate change, inside our studied sample universe of appliances, to 27 cycles (450ms).

Table 31 – Appliance variation in metrics after feature extraction rate change

Appliance	FER (cycles)	Accuracy	F-measure	Precision	Recall	TECA	NEAP
Ceiling fan	New 27 (A)	87%	81%	83%	82%	97%	2%
	Old 20 (B)	90%	85%	86%	86%	97%	2%
	Variation% (B-A)	**3%**	**4%**	**3%**	**4%**	**0%**	**0%**
Plasma TV	New 27 (A)	90%	85%	86%	85%	94%	8%
	Old 23 (B)	92%	90%	90%	90%	95%	7%
	Variation% (B-A)	**2%**	**5%**	**4%**	**5%**	**1%**	**-1%**
Fluorescent bulb	New 27 (A)	84%	75%	80%	75%	97%	4%
	Old 17 (B)	87%	82%	83%	81%	97%	3%
	Variation% (B-A)	**3%**	**7%**	**3%**	**6%**	**0%**	**-1%**
Microwave oven	New 27 (A)	95%	78%	91%	75%	98%	2%
	Old 25 (B)	95%	80%	91%	76%	98%	2%
	Variation% (B-A)	**0%**	**2%**	**0%**	**1%**	**0%**	**0%**
Fridge	New 27 (A)	85%	83%	84%	85%	90%	20%
	Old 17 (B)	90%	88%	87%	85%	90%	17%
	Variation% (B-A)	**5%**	**5%**	**3%**	**0%**	**0%**	**-3%**
Air conditioner	New 27 (A)	95%	90%	85%	85%	95%	10%
	Old 16 (B)	96%	90%	92%	90%	95%	10%
	Variation% (B-A)	**1%**	**0%**	**7%**	**5%**	**0%**	**0%**
X-Box One	New 27 (A)	90%	85%	85%	85%	95%	6%
	Old 16 (B)	92%	86%	87%	87%	96%	5%
	Variation% (B-A)	**2%**	**1%**	**2%**	**2%**	**1%**	**-1%**

6.3. Analysis of feature selection

This section focuses on the analysis that concerns the feature selection, after evaluating the data in Section 6.1. As the first step, this study proposed creating a summary of the previous feature sets found in the simulations. This compilation is shown in Table 32. The results were an attempt to elucidate a feature's importance for load disaggregation and for NILM problems, in general. The objective was to construct a *Feature Ranking*, based on the number of times that each appliance appears in the list. So, for each appliance, the features were placed in columns and, for each appearance, the count was increased.

Table 32 – Features' summary for studied appliances

	Feature List																													
	Features						*Fourier Harmonics*																							
Appliance	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Clothes Iron	✓	✓	✓	✓	✓	✓		✓		✓		✓		✓		✓				✓		✓				✓				
Desktop PC	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ceiling fan	✓	✓	✓	✓	✓	✓		✓		✓		✓		✓						✓										
Plasma TV	✓	✓	✓	✓				✓	✓				✓	✓		✓		✓			✓	✓	✓	✓	✓					
LED bulb	✓	✓	✓	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓				✓		✓		✓
Fluorescent bulb	✓	✓	✓	✓	✓	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓
Halogen bulb	✓	✓	✓	✓	✓	✓		✓		✓				✓						✓										
Hair dryer	✓	✓	✓	✓	✓			✓						✓	✓															
Microwave oven		✓	✓		✓	✓			✓											✓						✓				
Washing machine	✓	✓	✓	✓	✓	✓	✓	✓																	✓			✓		
Fridge	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓		✓		✓										
Air conditioner	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓		✓	✓		✓						✓		✓		✓
X-Box One	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓		✓		✓		✓				✓		✓		✓		✓		✓
LCD Monitor	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓		✓	✓	✓	✓	✓		✓		✓
Notebook	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓		✓	✓			✓			✓						✓		

Therefore, as the second step, we evaluated a *Feature Ranking*, based on the number of times (*appearance score*) that each individual feature appeared inside each selection. The ranking is shown in Table 33, and includes other columns related to the individual and cumulative percentages over the entire selection. Individual percentages are calculated by dividing the number of times the appliance appeared by the total number of appearances. The cumulative percentages are calculated by summing all the individual appliances' percentages as the ranking decreases.

Little attention was given to other features, compared to active power (P) and reactive power (Q). The majority of previous NILM researches assumed that active power (P) is, by far, the most important feature for load disaggregation. It is clear that, when measuring each appliance individually, all necessary information about energy consumption and the status of the appliance could be directly obtained from P.

However, for disaggregation purposes, could active power (P) be a less relevant feature than others could? According to our studied universe of sampled appliances and evaluated algorithms, *root mean squared current* (I_{rms}) and *apparent power* (S) displayed a higher appearance score than P. Even the *reactive power* (Q) and *3rd harmonic of Fourier* displayed the same score as P. Therefore, it is certainly, a relevant feature for measuring energy, but for disaggregation purposes, in our studied universe, it appeared to have similar relevance as others features.

Table 33 – Feature ranking sorted by appearance in studied universe of appliances

Feature	*Appearance score*	*Individual %*	*Cumulative %*
I_{rms}	15	6,47%	6,47%
S	15	6,47%	12,93%
3rd	14	6,03%	18,97%
P	14	6,03%	25,00%
Q	14	6,03%	31,03%
CF	13	5,60%	36,64%
F	13	5,60%	42,24%
9th	12	5,17%	47,41%
5th	11	4,74%	52,16%
7th	10	4,31%	56,47%
15th	10	4,31%	60,78%
11th	9	3,88%	64,66%
13th	8	3,45%	68,10%
17th	8	3,45%	71,55%
21st	8	3,45%	75,00%
---------- *50% of features*			
23th	8	3,45%	78,45%
2nd	6	2,59%	81,03%
4th	6	2,59%	83,62%
25th	6	2,59%	86,21%
10th	5	2,16%	88,36%
19th	5	2,16%	90,52%
8th	4	1,72%	92,24%
20th	4	1,72%	93,97%
6th	3	1,29%	95,26%
18th	3	1,29%	96,55%
12th	2	0,86%	97,41%
14th	2	0,86%	98,28%
16th	2	0,86%	99,14%
22nd	1	0,43%	99,57%
24th	1	0,43%	100,00%

Table 33 displays the 30 evaluated features in a ranking, by the *appearance score*, of our study. Therefore, if we select features using the ranking, we could adjust the data reduction by using these parameters as a premise. Our study defined a premise where 50% was adequate for data reduction, for this initial analysis; however, other premises, reducing or increasing features, could be studied in future. If we look for the cumulative percentages for example, the **first half of the features** comprises 75% of the appearances in the list. Thus, using our defined *feature ranking*, this study proposes the following analysis of this specific 15 feature set, representing 50% of data reduction:

$$S_{50\%features} = [I_{rms}, S, 3^{rd}, P, Q, CF, F, 9^{th}, 5^{th}, 7^{th}, 15^{th}, 11^{th}, 13^{th}, 17^{th}, 21^{st}];$$

to check whether if the contained features were enough to maintain metrics at a predefined feature extraction rate.

Therefore, as the third step of analysis, we created and trained new models for the *Deep Leaning* classifier, using the $S_{50\%features}$ array as their input. Notwithstanding that, all the appliances were reevaluated in a comprehensive comparison over the models trained by the complete set of features (full set, with 30 features). The comparison results for the eight metric scores, using an average of three tested scenarios, are displayed in Figure 84. The comparison results for the worst case scenario are displayed in Figure 85. A more detailed view, with zoom, of the Figures 84 and 85, can be examined in Appendix C, at Figures 122, 123, 124, 125, 126, 127, 128 and 129.

The appliance comparison shows that, in just a few cases for some appliances, the full set scored better than the chosen 15 features set. Most of metric scores were similar, or even better, when the feature ranking set of 15 features was used. Additionally, the full set required longer training and predicting times in most of the studied cases. Therefore, feature selection with a 50% data reduction premise, based on our defined *feature ranking*, was proven to maintain metric scores close to the maximum rates obtained by the Deep Learning classifier. Countless combinations of features, even more restrictive, based on our *feature ranking*, could be tested in future; however, at this time, it is clear that the main contribution presented for the feature selection is not our *feature ranking*, but the method to create a *feature ranking* based on a studied appliance universe. Thus, we considered the $S_{50\%features}$ adequate for our analysis of feature selection.

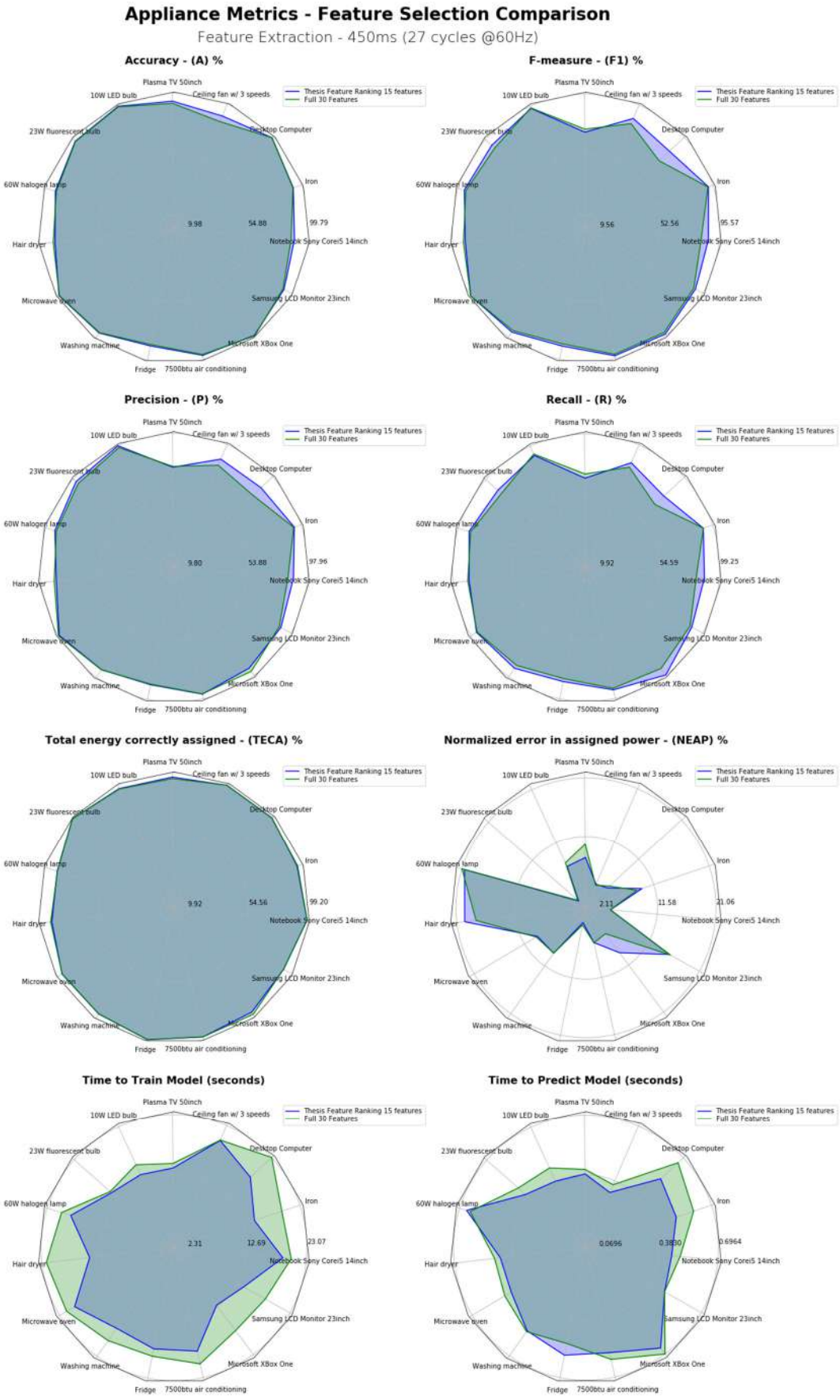

Figure 84 – Feature selection comparison for appliances at 450ms extraction (average results)

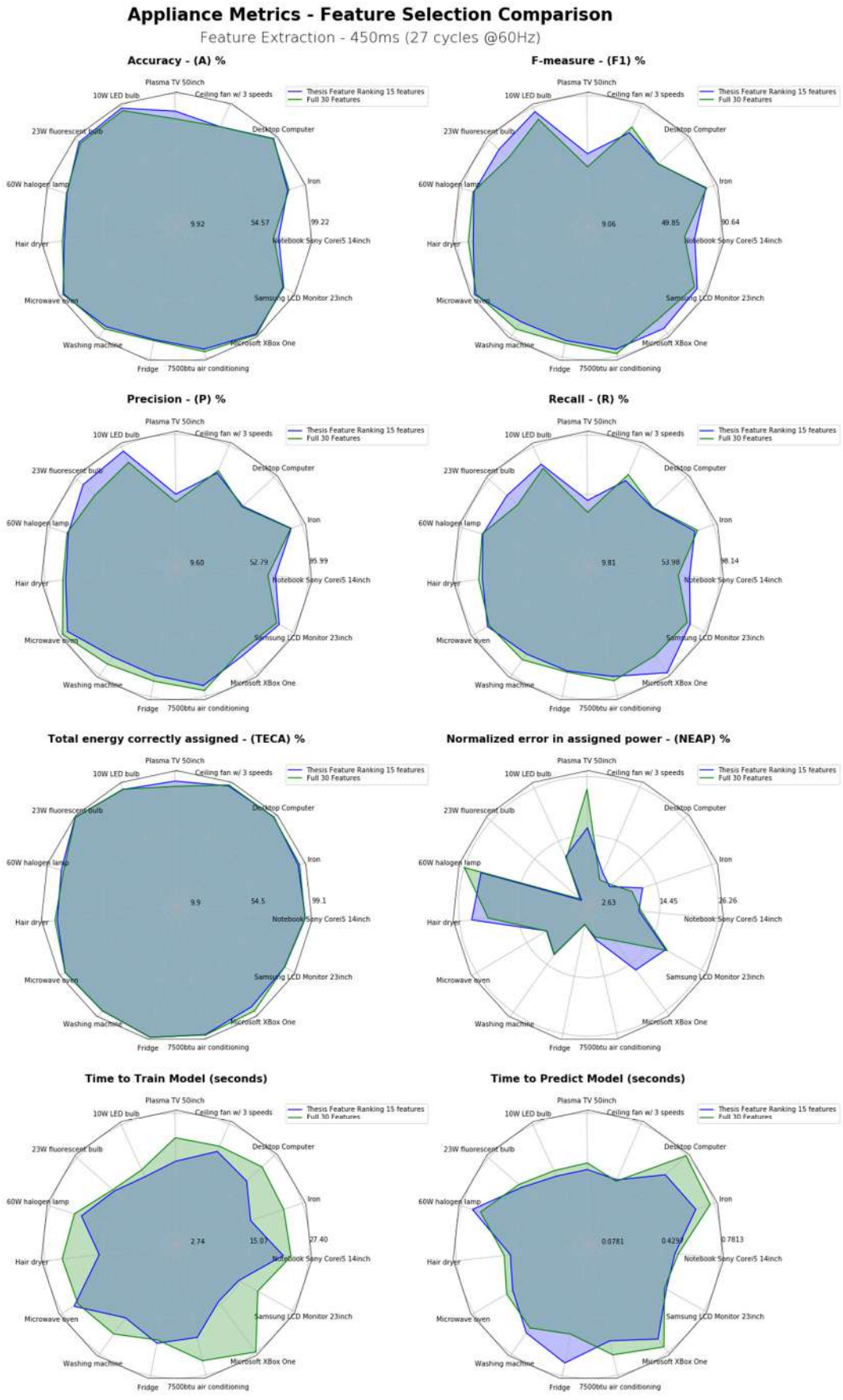

Figure 85 – Feature selection comparison for appliances at 450ms extraction (worst results)

6.4. Analysis of a real-life dataset

The final analysis that we proposed was to train models with synthetic mixture streams and to predict those models using a real-life sampled mixture datasets. In order to maintain similar levels of dataset complexity, we sampled datasets with exactly 10 appliances being activated at random times, to simulate a home. The appliances in the real mixture were: (1) a 23W Fluorescent bulb; (2) a hair dryer; (3) a 9W fluorescent bulb; (9) a 40W halogen bulb; (5) a notebook; (6) a 42 inch LED TV; (7) a 100W halogen bulb; (8) an Apple MacMini computer (9) a 23 inch LED monitor; (10) an X-Box One.

The only appliances of which the trained model had any previous knowledge were the 23W fluorescent bulb, the hair dryer and the X-Box One. Therefore, the sampled mixture dataset presented 7 unseen appliances. The two appliances that we selected empirically, for an evaluation of the predictions, were (1) the 23W fluorescent bulb and (2) the hair dryer. Both appliances were selected because of their displayed average best and worst values, respectively, from the metric scores after the analysis of Section 6.3. As seen in Figure 86, the predictions results of disaggregation were promising for the 23W Fluorescent bulb, and for the hair dryer, in Figure 87 .

The 23W fluorescent bulb displayed almost flawless predictions, losing performance only in an **on** state, because of oscillations in power consumption. The hair dryer was more difficult to assign an active power status correctly, because of the quantization process. The clustering algorithm found only two clusters for this appliance; however, visually it contains at least five quantization steps, which contributed to the reduced number of visible power states.

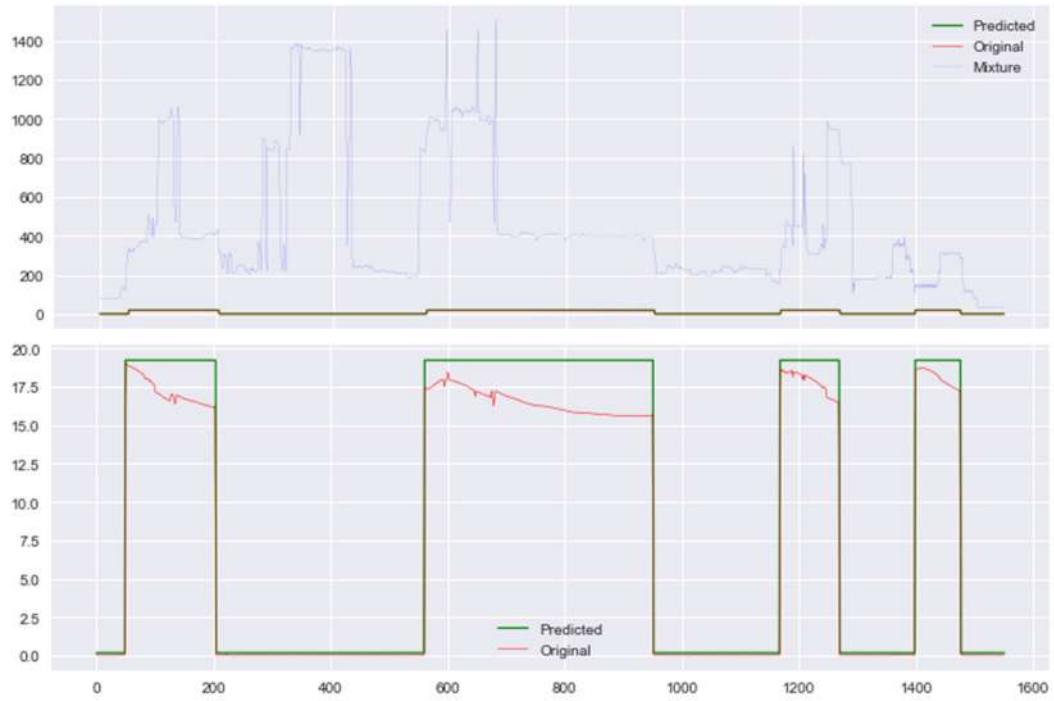

Figure 86 - 23W Fluorescent bulb active power (P) predictions inside the mixture (top) and with zoom (bottom)

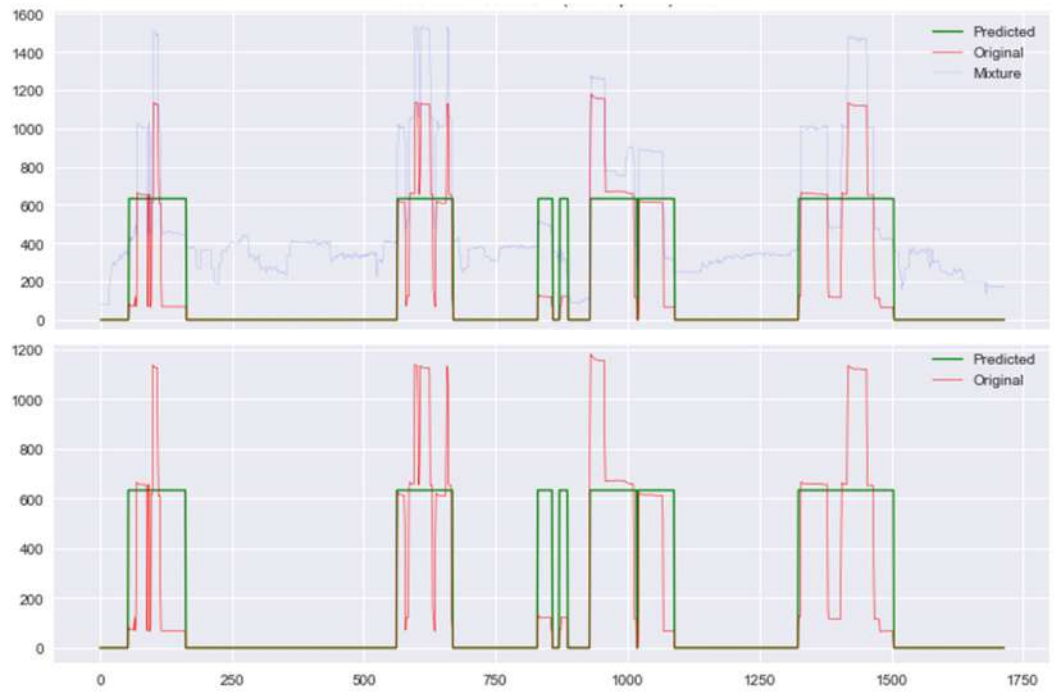

Figure 87 – Hair dryer active power (P) predictions inside the mixture (top) and with zoom (bottom)

7. Conclusions

The development of this thesis and its research involved knowledge about power electronics and its associated theories and practice: Some of the included elements were the testing of PLLs for synchronizing voltage waveform; analog and digital signal processing, with the development of a low-cost analog anti-aliasing filter and the analysis using power spectral density; Fourier harmonics to generate synthetic datasets and feature extraction; the hardware and firmware development of an operational, low-cost smart meter; the machine learning implementations of NILM disaggregation methods, space quantization (clustering), feature selection and feature extraction rate analysis; and, the development of an innovative Deep Learning technique that would be able to disaggregate different kinds of electrical loads efficiently.

All the simulations involved a great deal of time to carry our training and make predictions, because of the large number of potential combinations of the different machine learning methods, the feature selection methods and the comprehensive frequency selection range. The training and prediction time, using a fully loaded computing capacity of 40 Intel XEON CPUs and 4 GTX-1080 GPUs, took almost three months of non-stop twenty-four hour computation.

All the models, training data, datasets and images generated more than 300GB of data, which was compressed, for posterior analysis. The human review of this generated data was a long and time-consuming step to complete. More than 10,000 graphs of clustering results, mixture results, training results, prediction results and feature selection results were generated that together took more than more six months' work. This included an intensive selection of data, and models and rework on parameterization and architecture. In addition all the machine learning systems had to be retrained before any actual outcomes were achieved.

In some ways, the results that were obtained from the developed methodology for feature selection broke the paradigm of using exclusively active power (P) as a main feature for disaggregation purposes, which is an important contribution to the field of NILM study. The results for feature ranking can also be considered a major result, as

they showed feature significance based on the number of appearances, during the development of our methodology.

Our feature extraction rate analysis also revealed that 27 cycles of 60Hz are adequate for features measurement, and allow resolution balancing whilst maintaining metrics at acceptable performance levels, while also reducing the need for feature measurement extraction, transmission and storage by 27 times.

Our development of a methodology for synthetic mixture generation was a relevant contribution for current and future NILM development and, in addition, our training with synthetic datasets showed great results, even when used on real-sampled mixtures load disaggregation tests.

7.1.Future work

The solutions presented in this study are merely a first initial step for NILM development. We believe that merging different models that have been trained on different datasets will certainly improve performance for unseen environments, and will allow the creation of an ensemble to disaggregate general homes or offices. Deep neural networks could also be trained, based on appliance kind and not for a specific appliance; thus, reducing the number of networks to be trained and creating an artificial intelligent architecture that is more generalist and more scalable.

The methodology we presented, for creating synthetic datasets, does not limit the maximum number of appliances per mixture. Our research used a limit of ten appliances per mixture, merely for the purposes of controlling the environment's complexity, and to provide similar levels of feature extraction. There is a still an unknown maximum number of appliances, that are limited by noise's being added to mixture from each appliance, something that does not occur when sampling aggregated appliances at the same time. These noise limits could be part of future study, or, alternatively, research could seek a method to filter noise with signal processing.

Finally, as concerns Deep Neural Networks; more advanced activations, such as SELU´s, Scaled Exponential Linear Units (Klambauer, et al., 2017), could offer

improvements in performance. The use of Convolutional Neural Networks, Recurrent Neural Networks, Spatial Transform Network, Curriculum Learn and a strategy using Reinforcement Leaning could potentially raise performance in metric scores and further reduce the real necessity for feature extraction at the defined rate.

However, we believe that the analysis of power consumption of the machine learning algorithms, for disaggregation purposes, is an important issue to evaluate. It remains unclear whether more power would be consumed, by computing training/predictions, than might potentially be saved, by using the results of those computations. The transfer of energy savings, from homes to cloud servers, may become reality, which would mean that the complexity of the algorithms required to fit and predict the data should not surpass the potential for saving energy. Therefore, future NILM research should highlight the fine-tuning of energy efficient algorithms.

Bibliography

ACEEE Advanced Metering Initiatives and Residential Feedback Programs: A Meta-Review for Household Electricity-Saving Opportunities [Online]. - 05 2010. - https://www.smartgrid.gov/document/advanced_metering_initiatives_and_residential_feedback_programs_meta_review_household_ele_0.

Adabi A. [et al.] Status and challenges of residential and industrial non-intrusive load monitoring [Conference] // Technologies for Sustainability (SusTech), 2015 IEEE Conference on. - 2015. - pp. 181-188. - DOI: 10.1109/SusTech.2015.7314344.

Alcala Jose, Parson Oliver and Rogers Alex Detecting Anomalies in Activities of Daily Living of Elderly Residents via Energy Disaggregation and Cox Processes [Conference] // Proceedings of the 2Nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments. - New York, NY, USA : ACM, 2015. - pp. 225-234. - ISBN: 978-1-4503-3981-0 DOI: 10.1145/2821650.2821654.

Alejandro Lisa [et al.] Global Market for Smart Electricity Meters: Government Policies Driving Strong Growth [Online] // U.S. International Trade Commission. - OFFICE OF INDUSTRIES WORKING PAPER, 06 2014. - 03 09, 2016. - https://www.usitc.gov/publications/332/id-037smart_meters_final.pdf.

Allison Paul What's So Special About Logit? \textbar Statistical Horizons // What's So Special About Logit? \textbar Statistical Horizons. - 2015.

AMAZON AMAZON SANDISK 32 GB [Online] // AMAZON. - 2016. - 03 15, 2016. - http://www.amazon.com/SanDisk-microSDHC-Standard-Packaging-SDSQUNC-032G-GN6MA/dp/B010Q57T02/ref=pd_sim_107_5?ie=UTF8&dpID=41f7MricNhL&dpSrc=sims&preST=_AC_UL160_SR160%2C160_&refRID=1N4XVJB2CJFJGAANXWKA.

Anderson Kyle [et al.] BLUED: A Fully Labeled Public Dataset for Event-Based Non-Intrusive Load Monitoring Research [Conference] // Proceedings of the 2nd KDD Workshop on Data Mining Applications in Sustainability (SustKDD). - Beijing, China : [s.n.], 2012.

ANEEL Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional – PRODIST Modulo 8 - rev9 [Report]. - 2018.

Armel Carrie K. [et al.] Is disaggregation the holy grail of energy efficiency? The case of electricity [Journal] // Energy Policy. - jan 2013. - 0 : Vol. 52. - pp. 213-234. - ISSN: 0301-4215.

Armel Carrie K. [et al.] Is disaggregation the holy grail of energy efficiency? The case of electricity [Journal] // Energy Policy. - jan 2013. - 0 : Vol. 52. - pp. 213-234. - ISSN: 0301-4215.

Barker Sean [et al.] Smart*: An Open Data Set and Tools for Enabling [Article] // SustKDD, August, 2012. - 2012.

Basseville Michèle On-line detection of jumps in mean [Book Section] // Detection of Abrupt Changes in Signals and Dynamical Systems. - [s.l.] : Springer Berlin Heidelberg, 1986. - Vol. 77.

Basu K. [et al.] Nonintrusive Load Monitoring: A Temporal Multilabel Classification Approach [Journal] // Industrial Informatics, IEEE Transactions on. - 2015. - 1 : Vol. 11. - pp. 262-270. - ISSN: 1551-3203 DOI: 10.1109/TII.2014.2361288.

Batra Nipun [et al.] A comparison of non-intrusive load monitoring methods for commercial and residential buildings [Journal] // arXiv:1408.6595. - 2014.

Batra Nipun [et al.] It's Different: Insights into Home Energy Consumption in India [Conference] // Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings. - New York, NY, USA : ACM, 2013. - pp. 3:1--3:8. - ISBN: 978-1-4503-2431-1 DOI: 10.1145/2528282.2528293.

Batra Nipun [et al.] NILMTK: an open source toolkit for non-intrusive load monitoring [Conference] // Proceedings of the 5th international conference on Future energy systems. - Cambridge, United Kingdom : ACM, 2014. - pp. 265-276. - DOI: 10.1145/2602044.2602051.

Berges M. [et al.] User-Centered Nonintrusive Electricity Load Monitoring for Residential Buildings [Journal] // J. Comput. Civ. Eng.. - [s.l.] : American Society of Civil Engineers, may 2011. - 6 : Vol. 25. - pp. 471-480. - ISSN: 0887-3801 DOI: 10.1061/(ASCE)CP.1943-5487.0000108.

Berges Mario [et al.] Learning Systems for Electric Consumption of Buildings [Article] // Computing in Civil Engineering. - 2009.

Bergman D. C. [et al.] Distributed non-intrusive load monitoring [Conference] // Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES. - 2011. - pp. 1-8.

Berkson Joseph Application of the Logistic Function to Bio-Assay [Journal] // Journal of the American Statistical Association. - 1944. - 227 : Vol. 39. - pp. 357-365. - ISSN: 0162-1459 DOI: 10.2307/2280041.

Bermingham M. L. [et al.] Application of high-dimensional feature selection: evaluation for genomic prediction in man [Journal] // Scientific Reports. - may 2015. - Vol. 5. - p. 10312. - ISSN: 2045-2322 DOI: 10.1038/srep10312.

BLUETOOTH Adopted specifications [Online] // Bluetooth. - SIG, 2016. - 03 15, 2016. - https://www.bluetooth.com/specifications/adopted-specifications.

Boctor S. Design of a third-order single amplifier filter [Article] // IEEE Transactions on Circuits and Systems. - 1975. - Vol. 22.

Bonfigli R. [et al.] Unsupervised algorithms for non-intrusive load monitoring: An up-to-date overview [Conference] // Environment and Electrical Engineering (EEEIC), 2015 IEEE 15th International Conference on. - 2015. - pp. 1175-1180.

Bottou L. and Bousquet O. The Tradeoffs of Large Scale Learning [Book Section] // Advances in Neural Information / book auth. Platt J., Koller D. and Singer Y.. - 2008.

Bouhouras A. S. [et al.] Load signatures improvement through the determination of a spectral distribution coefficient for load identification [Conference] // European Energy Market (EEM), 2012 9th International Conference on the. - 2012. - pp. 1-6.

Breiman Leo Arcing classifier (with discussion and a rejoinder by the author) [Article] // Ann. Statist. - 1998. - 3. - Vol. 26.

Campbell C. Ying Y. Learning with Support Vector Machines [Book]. - 2011.

Cang Shuang and Yu Hongnian Mutual information based input feature selection for classification problems [Journal] // Decision Support Systems. - dec 2012. - 1 : Vol. 54. - pp. 691-698. - ISSN: 0167-9236 DOI: 10.1016/j.dss.2012.08.014.

Cardoso J.-F. Blind signal separation: statistical principles [Journal] // Proceedings of the IEEE. - oct 1998. - 10 : Vol. 86. - pp. 2009-2025. - ISSN: 00189219 DOI: 10.1109/5.720250.

CERNE Desperdício de energia atinge R$ 61,7 bi em três anos [Online]. - 05 24, 2017. - http://cerne.org.br/desperdicio-de-energia-atinge-r-617-bi-em-tres-anos/.

Chang Chih-Chung and Lin Chih-Jen LIBSVM: A library for support vector machines [Journal] // ACM Transactions on Intelligent Systems and Technology. - apr 2013. - 3 : Vol. 2. - pp. 1-27. - ISSN: 21576904 DOI: 10.1145/1961189.1961199.

Chang Hsueh-Hsien [et al.] Power-Spectrum-Based Wavelet Transform for Nonintrusive Demand Monitoring and Load Identification [Journal] // IEEE Transactions on Industry Applications. - 2014. - 3 : Vol. 50. - pp. 2081-2089. - ISSN: 0093-9994.

Chen Kun-Long, Chang Hsueh-Hsien and Chen Nanming A new transient feature extraction method of power signatures for Nonintrusive Load Monitoring Systems [Conference] // Applied Measurements for Power Systems (AMPS), 2013 IEEE International Workshop on. - 2013. - pp. 79-84.

Chen Y. C. [et al.] Detecting users' behaviors based on nonintrusive load monitoring technologies [Conference] // Networking, Sensing and Control (ICNSC), 2013 10th IEEE International Conference on. - 2013. - pp. 804-809.

Chollet F. and others Keras [Online]. - 2015. - 01 02, 2018. - https://github.com/keras-team/keras.

Clayton David, Hills Michael and Lindsey J K Statistical Models in Epidemiology [Journal]. - 1993. - p. 5.

Cole A. I. and Albicki A. Data extraction for effective non-intrusive identification of residential power loads [Conference] // Instrumentation and Measurement Technology Conference, 1998. IMTC/98. Conference Proceedings. IEEE. - 1998. - Vol. 2. - pp. 812--815 vol.2. - ISSN: 1091-5281.

DEEPMIND DISCOVER MORE ABOUT ALPHAGO [Online] // DeepMind AlphaGo Zero. - 10 18, 2017. - 01 20, 2018. - https://deepmind.com/research/alphago/.

DIGI-KEY World's Largest Selection [Online] // Digi-Key Electronics. - Digi-Key, 2016. - 03 15, 2016. - http://www.digikey.com/.

Dinesh C. [et al.] Residential Appliance Identification Based on Spectral Information of Low Frequency Smart Meter Measurements // Residential Appliance Identification Based on Spectral Information of Low Frequency Smart Meter Measurements. - 2015. - Vol. PP. - pp. 1-12. - ISSN: 1949-3053.

Diniz Paulo S. R., Silva Eduardo A. B. da and Netto Sergio L. Digital Signal Processing: System Analysis and Design [Book]. - [s.l.] : Cambridge University Press, 2010. - Vol. 2.

Du Liang [et al.] Electric Load Classification by Binary Voltage 2013;Current Trajectory Mapping [Journal] // IEEE Transactions on Smart Grid. - 2015. - 1 : Vol. 7. - pp. 358-365. - ISSN: 1949-3053.

Ducange P., Marcelloni F. and Antonelli M. A Novel Approach Based on Finite-State Machines with Fuzzy Transitions for Nonintrusive Home Appliance Monitoring [Journal] // IEEE Transactions on Industrial Informatics. - 2014. - 2 : Vol. 10. - pp. 1185-1197. - ISSN: 1551-3203.

Egan James P Signal detection theory and ROC-analysis [Book]. - [s.l.] : New York : Academic Press, 1975.

EIA Annual Energy Outlook 2010 [Online] // Energy Information Administration. - U.S Department of Energy, 2010. - 03 03, 2016. - http://www.eia.gov/oiaf/archive/aeo10/index.html.

EPRI Residential Electricity Use Feedback: A Research Synthesis and Economic Framework [Online]. - 02 27, 2009. - http://www.epri.com/abstracts/Pages/ProductAbstract.aspx?ProductId=000000000001016844.

Ester M. [et al.] A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise [Article] // Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining. - Portland : [s.n.], 1996.

FCC Code of Federal Regulations (CFR) annual edition [Online] // ELECTRONIC CODE OF FEDERAL REGULATIONS. - US Government Publishing Office, 03 08, 2016. - 03 10, 2016. - http://www.ecfr.gov/cgi-bin/text-idx?c=ecfr&tpl=/ecfrbrowse/Title47/47cfrv1_02.tpl.

Feng C. Y. [et al.] Tracing of energy consumption by using harmonic current [Conference] // Research and Development (SCOReD), 2013 IEEE Student Conference on. - 2013. - pp. 444-449.

Freedman A. D. Statistical Models: Theory and Practice [Book Section] / book auth. Freedman David A.. - [s.l.] : Cambridge University Press, 2009.

Freund Yoav and Schapire Robert E A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting [Journal] // Journal of Computer and System Sciences. - aug 1997. - 1 : Vol. 55. - pp. 119-139. - ISSN: 00220000 DOI: 10.1006/jcss.1997.1504.

Gao Jingkun [et al.] PLAID: A Public Dataset of High-resolution Electrical Appliance Measurements for Load Identification Research: Demo Abstract [Conference] // Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings. - New York, NY, USA : ACM, 2014. - pp. 198-199. - ISBN: 978-1-4503-3144-9 DOI: 10.1145/2674061.2675032.

Gao Z. [et al.] Anti-aliasing Filter Circuit Design for Active Power Filter [Conference] // Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2010 2nd International Conference on. - 2010. - Vol. 1. - pp. 147-150. - DOI: 10.1109/IHMSC.2010.44.

Geurts P., Ernst D. and Wehenkel L. Extremely randomized trees [Journal]. - [s.l.] : Kluwer Academic Publishers, 2006.

Gillingham Kenneth, Rapson David and Wagner Gernot The Rebound Effect and Energy Efficiency Policy [Article] // Resources For The Future. - Washington, DC : [s.n.], 2014.

Giri Suman and Berges Mario An energy estimation framework for event-based methods in Non-Intrusive Load Monitoring [Article] // Energy Conversion and Management. - [s.l.] : Elsevier, 2015.

Giri Suman, Lai Po-Hsiang and Berges Mario Novel Techniques For ON and OFF states detection of appliances for Power Estimation in Non-Intrusive Load Monitoring [Article] // The 30th International Symposium on Automation and Robotics in Construction and Mining. - Montreal : [s.n.], 2013.

Glorot X. and Bengio Y. Understanding the difficulty of training deep feedforward neural networks [Article]. - Quebec : [s.n.], 2010.

GOOGLE TensorFlow - Open-Source Machine LEarning Framework for everyone. - 2018.

GTM RESEARCH The U.S. Smart Meter Market [Journal]. - [s.l.] : Lacey, 2013.

Guyon Isabelle [et al.] Gene Selection for Cancer Classification using Support Vector Machines [Journal] // Machine Learning. - jan 2002. - 1-3 : Vol. 46. - pp. 389-422. - ISSN: 0885-6125, 1573-0565 DOI: 10.1023/A:1012487302797.

Hart G.W. Nonintrusive appliance load monitoring [Journal] // Proceedings of the IEEE. - 1992. - 12 : Vol. 80. - pp. 1870-1891. - ISSN: 0018-9219 DOI: 10.1109/5.192069.

Ho Tin Kam Random decision forests [Conference] // Proceedings of 3rd International Conference on Document Analysis and Recognition. - 1995. - Vol. 1. - pp. 278--282 vol.1. - DOI: 10.1109/ICDAR.1995.598994.

Huijskens Thomas Mutual information-based feature selection // Mutual information-based feature selection. - oct 2017.

IBM Deep Blue [Online] // IBM100 Icons of Progress. - 1996. - 01 20, 2018. - http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/deepblue/.

IEA World Energy Outlook 2011 [Online] // International Energy Agency. - 2011. - 03 03, 2016. - https://www.iea.org/publications/freepublications/publication/WEO2011_WEB.pdf.

IEEE 802.11 IEEE 802.11™ The Working Group for WLAN Standards [Online] // IEEE 802.11™. - 2012. - Fevereiro 08, 2012. - http://www.ieee802.org/11/.

Ioffe S. and Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift [Article] // CoRR. - 2015.

Iwayemi A. and Zhou C. SARAA: Semi-Supervised Learning for Automated Residential Appliance Annotation // SARAA: Semi-Supervised Learning for Automated Residential Appliance Annotation. - 2015. - Vol. PP. - pp. 1-8. - ISSN: 1949-3053.

Joachims Thorsten Text categorization with Support Vector Machines: Learning with many relevant features [Conference] // Machine Learning: ECML-98. - [s.l.] : Springer, Berlin, Heidelberg, 1998. - pp. 137-142. - ISBN: 978-3-540-64417-0 978-3-540-69781-7 DOI: 10.1007/BFb0026683.

Kamat S. P. Fuzzy logic based pattern recognition technique for non-intrusive load monitoring [Conference] // TENCON 2004. 2004 IEEE Region 10 Conference. - 2004. - Vol. C. - pp. 528--530 Vol. 3.

Karimi-Ghatemani M. and Iravani M.R. A method for synchronization of power electronic converters in polluted and variable-frequency enviroments [Journal]. - 2004.

Kelly Jack and Knottenbelt William J. Neural NILM: Deep Neural Networks Applied to Energy Disaggregation [Journal] // CoRR. - 2015. - Vol. abs/1507.06594.

Kelly Jack and Knottenbelt William The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes [Journal]. - 2015. - 150007 : Vol. 2. - DOI: 10.1038/sdata.2015.7.

Kelly Jack and Knottenbelt, William J. Metadata for Energy Disaggregation [Journal] // CoRR. - 2014. - Vol. abs/1403.5946.

Kim Hyungsul [et al.] Unsupervised Disaggregation of Low Frequency Power Measurements [Article] // SDMSIAM. - 2011.

Kingma D. P. and Ba J. Adam: A Method for Stochastic Optimization [Article]. - 2014.

Klambauer G. [et al.] Self-Normalizing Neural Networks [Article] // CoRR. - 2017.

Kleiminger Wilhelm, Beckel Christian and Santini Silvia Household Occupancy Monitoring Using Electricity Meters [Conference] // Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2015). - Osaka, Japan : [s.n.], 2015.

Klein P. [et al.] Analysis of fingerprints of electric appliances as starting point for an appliance characteristics catalog [Conference] // Industrial Electronics Society, IECON 2014 - 40th Annual Conference of the IEEE. - 2014. - pp. 3517-3521.

Kleinbaum David G. and Klein Mitchel Maximum Likelihood Techniques: An Overview [Book Section] // Logistic Regression. - [s.l.] : Springer, New York, NY, 2010. - ISBN: 978-1-4419-1741-6 978-1-4419-1742-3 DOI: 10.1007/978-1-4419-1742-3_4.

Kohs Greg AlphaGo. - RO*CO FILMS, 2017.

Kolter J. Zico [et al.] Reference Energy Disaggregation Data Set (REDD) [Article]. - 2011.

Kondo D. [et al.] Cost-benefit analysis of Cloud Computing versus desktop grids [Conference] // Parallel Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on. - 2009. - pp. 1-12. - ISSN: 1530-2075 DOI: 10.1109/IPDPS.2009.5160911.

Koza J.R. [et al.] Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming [Book Section] // Artificial Intelligence in Design '96 / book auth. Gero J.S. and Sudweeks F.. - [s.l.] : Springer, Dordrecht, 1996.

Kozachenko L F and Leonenko N N Sample Estimate of the Entropy of a Random Vector [Journal]. - 1987. - p. 9.

Laughman C. [et al.] Power signature analysis [Journal] // IEEE Power and Energy Magazine. - 2003. - 2 : Vol. 1. - pp. 56-63. - ISSN: 1540-7977.

Lawson Ed [et al.] Decision forests for machine learning classification of large, noisy seafloor feature sets [Journal] // Computers \& Geosciences. - feb 2017. - Vol. 99. - pp. 116-124. - ISSN: 0098-3004 DOI: 10.1016/j.cageo.2016.10.013.

Leeuwen D. A. Van and Brummer N. Channel-dependent GMM and Multi-class Logistic Regression models for language recognition [Conference] // 2006 IEEE Odyssey - The Speaker and Language Recognition Workshop. - 2006. - pp. 1-8. - DOI: 10.1109/ODYSSEY.2006.248094.

Liao J. [et al.] Non-intrusive appliance load monitoring using low-resolution smart meter data [Conference] // Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on. - 2014. - pp. 535-540.

Lin Y. H., Tsai M. S. and Chen C. S. Applications of fuzzy classification with fuzzy c-means clustering and optimization strategies for load identification in NILM systems [Conference] // Fuzzy Systems (FUZZ), 2011 IEEE International Conference on. - 2011. - pp. 859-866. - ISSN: 1098-7584.

Lin Yu-Hsiu and Tsai Men-Shen Non-Intrusive Load Monitoring by Novel Neuro-Fuzzy Classification Considering Uncertainties [Journal] // Smart Grid, IEEE Transactions on. - 2014. - 5 : Vol. 5. - pp. 2376-2384. - ISSN: 1949-3053 DOI: 10.1109/TSG.2014.2314738.

Liu Y. and Chen M. A review of nonintrusive load monitoring and its application in commercial building [Conference] // Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2014 IEEE 4th Annual International Conference on. - 2014. - pp. 623-629.

Lovins Amory The Negawatt Revolution - Solving the CO2 Problem [Online]. - Green Energy Conference, 1989. - 03 03, 2016. - http://www.ccnr.org/amory.html.

Luo Dong [et al.] Monitoring HVAC equipment electrical loads from a centralized location - Methods and field test results [Article] // ASHRAE TRANSACTIONS 108. - 2002. - 841-857.

Maasoumy Mehdi [et al.] BERDS - BERkeley EneRgy Disaggregation Data Set [Article]. - 2013.

Maccone C. Telecommunications, KLT and Relativity [Book]. - Colorado Springs, CO, USA : IPI Press, 1994.

Makonin S. [et al.] AMPds: A public dataset for load disaggregation and eco-feedback research [Conference] // Electrical Power & Energy Conference (EPEC), 2013 IEEE. - 2013. - pp. 1-6.

Marianne R. Knut H. S. How energy efficiency fails in the building industry [Article] // sciencedirect. - 2009. - ISSN 0301-4215 : Vol. 37.

MATHWORKS The Language of Technical Computing [Online] // Matlab. - MathWorks, 2016. - 03 16, 2016. - http://www.mathworks.com/products/matlab/.

McInnes L., Healy J. and Astels S. hdbscan: Hierarchical density based clustering [Article] // The Journal of Open Source Software. - 2017.

MIT Technology Review Deep Learning [Online] // MIT Technology Review. - 2013. - 01 02, 2018. - https://www.technologyreview.com/s/513696/deep-learning/.

Mitchell J. E. Branch-and-Cut Algorithms for Combinatorial Optimization Problems [Journal]. - 1999. - p. 19.

MOMYPI ModMyPi [Online] // ModMyPi. - ModMyPi, 2016. - 03 14, 2016. - http://www.modmypi.com/raspberry-pi/?ref=rs.

Monacchi Andrea and Egarter, Dominik and Elmenreich, Wilfried and D'Alessandro, Salvatore and Tonello, Andrea M. GREEND: An Energy Consumption Dataset of Households in Italy and Austria [Journal] // CoRR. - 2014. - Vol. abs/1405.3100.

Müller Meinard Dynamic Time Warping [Book Section] // Information Retrieval for Music and Motion. - [s.l.] : Springer Berlin Heidelberg, 2007.

Nair V. and Hinton G. E. Rectified Linear Units Improve Restricted Boltzmann Machines [Article]. - 2010.

Nascimento P.P.M Applications of deep learning techniques on NILM [Online]. - 04 2016. - 02 05, 2018. - http://pee.ufrj.br/teses/textocompleto/2016041101.pdf.

NITIME Multi-taper spectral estimation [Online] // Nitime: time-series analysis for neuroscience. - 2009. - 01 02, 2018. - http://nipy.org/nitime/examples/multi_taper_spectral_estimation.html.

NUMPY NumPy [Online]. - 2017. - 01 20, 2018. - http://www.numpy.org/.

OXFORD Definition of overfitting in English [Online]. - 2018. - 01 02, 2018. - https://en.oxforddictionaries.com/definition/overfitting.

Parson Oliver Unsupervised Training Methods for Non-intrusive Appliance Load Monitoring from Smart Meter Data [Book]. - University of Southampton : [s.n.], 2014. - Vol. Thesis for the degree of Doctor of Philosophy.

Patel Shwetak N. [et al.] At the flick of a switch: detecting and classifying unique electrical events on the residential power line [Conference] // Proceedings of the 9th international conference on Ubiquitous computing. - Innsbruck, Austria : Springer-Verlag, 2007. - pp. 271-288.

Pattem S. Unsupervised Disaggregation for Non-intrusive Load Monitoring [Conference] // Machine Learning and Applications (ICMLA), 2012 11th International Conference on. - 2012. - Vol. 2. - pp. 515-520.

PECAN STREET Dataport [Online] // Pecan Street. - 2016. - 03 12, 2016. - https://dataport.pecanstreet.org/.

Pedregosa F. [et al.] Scikit-learn: Machine Learning in Python [Article] // Journal of Machine Learning Research. - 2011.

Percival D. B. and Walden A. T. Spectral Analysis for Physical Applications: Multitaper and Conventional Univariate Techniques [Article]. - Cambridge : Cambridge University Press, 1993.

Pereira L. and Nunes N. J. Semi-automatic labeling for public non-intrusive load monitoring datasets [Conference] // Sustainable Internet and ICT for Sustainability (SustainIT), 2015. - 2015. - pp. 1-4.

Pereira L. and Nunes N. J. Towards systematic performance evaluation of non-intrusive load monitoring algorithms and systems [Conference] // Sustainable Internet and ICT for Sustainability (SustainIT), 2015. - 2015. - pp. 1-3.

Pereira Lucas [et al.] SustData: A Public Dataset for ICT4S Electric Energy Research [Article] // 2nd International Conference on ICT for Sustainability (ICT4S 2014). - 2014.

Pereira Lucas [et al.] The design of a hardware-software platform for long-term energy eco-feedback research [Conference] // Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems. - Copenhagen, Denmark : ACM, 2012. - pp. 221-230. - DOI: 10.1145/2305484.2305521.

Pereira Lucas, Nunes Nuno and Berges M. SURF and SURF-PI: A File Format and API for Non-Intrusive Load Monitoring Public Datasets [Conference] // International Conference on Future Energy Systems (e-Energy \textquoteright14) [short paper]. - Cambridge, UK : ACM, 2014. - DOI: 10.1145/2602044.2602078.

Prasad R. S. and Semwal S. A simplified new procedure for identification of appliances in smart meter applications [Conference] // Systems Conference (SysCon), 2013 IEEE International. - 2013. - pp. 339-344.

PROCEL Avaliação do mercado de eficiência energética no Brasil - Ano base 2005 - Classe residencial [Online] // Procel Info. - Procel, 2005. - 03 16, 2016. - http://www.procelinfo.com.br/main.asp?View=%7B5A08CAF0-06D1-4FFE-B335-95D83F8DFB98%7D&Team=&params=itemID=%7B528C16AF-3ECF-4898-A599-D9251D2D4654%7D%3B&UIPartUID=%7B05734935-6950-4E3F-A182-629352E9EB18%7D.

Quintal Filipe [et al.] SINAIS: home consumption package: a low-cost eco-feedback energy-monitoring research platform [Conference] // Proceedings of the 8th ACM Conference on Designing Interactive Systems. - Aarhus, Denmark : ACM, 2010. - pp. 419-421. - DOI: 10.1145/1858171.1858252.

RASPBERRY Raspberry Pi [Online] // Raspberry. - 2016. - 03 14, 2016. - https://www.raspberrypi.org/.

Reinhardt Andreas [et al.] On the Accuracy of Appliance Identification Based on Distributed Load Metering Data [Conference] // Proceedings of the 2nd IFIP Conference on Sustainable Internet and ICT for Sustainability (SustainIT). - 2012. - pp. 1-9.

Ridi A., Gisler C. and Hennebert J. ACS-F2 2014; A new database of appliance consumption signatures [Conference] // Soft Computing and Pattern Recognition (SoCPaR), 2014 6th International Conference of. - 2014. - pp. 145-150.

Ridi A., Gisler C. and Hennebert J. Appliance and state recognition using Hidden Markov Models [Conference] // Data Science and Advanced Analytics (DSAA), 2014 International Conference on. - 2014. - pp. 270-276.

Ross B. C. Mutual Information between Discrete and Continuous Data Sets [Journal]. - 2014.

Sadeghianpourhamami N. [et al.] Comprehensive feature selection for appliance classification in NILM [Journal] // Energy and Buildings. - sep 2017. - Vol. 151. - pp. 98-106. - ISSN: 0378-7788 DOI: 10.1016/j.enbuild.2017.06.042.

SCYPY SciPy.org [Online]. - 2018. - 01 20, 2018. - https://scipy.org/.

SDGROUP SD Card Association [Online] // SD Specifications Physical Layer Simplified Specification. - SD Group, 01 22, 2013. - 4.10. - 03 15, 2016. - https://www.sdcard.org/downloads/pls/pdf/part1_410.pdf.

Shulga Dima 5 Reasons Logistic Regression should be the first thing you learn when becoming a Data Scientist. - apr 2018.

Silapachote Piyanuch, Karuppiah Deepak R. and Hanson Allen R. Feature Selection Using Adaboost for Face Expression Recognition [Report] : Tech. rep. / MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER SCIENCE, MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER SCIENCE. - 2005.

Smith Denson [et al.] An efficient distributed protein disorder prediction with pasted samples [Journal] // Computers \& Electrical Engineering. - jan 2018. - Vol. 65. - pp. 342-356. - ISSN: 0045-7906 DOI: 10.1016/j.compeleceng.2017.08.002.

Srivastava N. [et al.] Dropout: A Simple Way to Prevent Neural Networks from Overfitting [Article] // Journal of Machine Learning Research. - 2014.

T. A. Brasil Modelagem matemática de algoritmos phase-locked loop baseados em controladores proporcionais integrais e proposta de um novo algoritmo baseado em estratégia fuzzy. - Rio de Janeiro : [s.n.], 2013.

Tay Francis E. H and Cao Lijuan Application of support vector machines in financial time series forecasting [Journal] // Omega. - aug 2001. - 4 : Vol. 29. - pp. 309-317. - ISSN: 0305-0483 DOI: 10.1016/S0305-0483(01)00026-3.

Teukolsky S.A., Vetterling W.T. and Flannery B.P. Numerical Recipes: The Art of Scientific Computing [Book Section]. - Cambridge : Cambridge University Press., 2007. - Vol. 3rd Ed.

THE ECONOMIST Negawatt hour [Online]. - 03 1, 2014. - 03 03, 2016. - http://www.economist.com/news/business/21597922-energy-conservation-business-booming-negawatt-hour.

THE HDF GROUP The HDF Group. Hierarchical Data Format, version 5, 1997-2018. - 1997.

TI CC2650 SimpleLink multi-standard 2.4 GHz ultra-low power wireless MCU [Online] // TI. - Texas Instruments, 2016. - 03 15, 2016. - http://www.ti.com/product/CC2650.

TI Texas Instruments [Online] // Texas Instruments. - Texas Instruments, 2016. - 03 14, 2016. - http://www.ti.com/.

Trung K. N. [et al.] Using FPGA for real time power monitoring in a NIALM system [Conference] // Industrial Electronics (ISIE), 2013 IEEE International Symposium on. - 2013. - pp. 1-6. - ISSN: 2163-5137 DOI: 10.1109/ISIE.2013.6563850.

UCSB University of California [Online] // ECE145B/218B Lecture Notes. - ECE Department - University of California, 2011. - 01 21, 2018. - http://www.ece.ucsb.edu/~long/ece145b/PLL_intro_FMD_FS.pdf.

Vanhoucke V. [et al.] Rethinking the Inception Architecture for Computer Vision [Article] // CoRR. - 2015.

Vergara Jorge R. and Estevez Pablo A. A review of feature selection methods based on mutual information [Journal] // Neural Computing and Applications. - jan 2014. - 1 :

Vol. 24. - pp. 175-186. - ISSN: 0941-0643, 1433-3058 DOI: 10.1007/s00521-013-1368-0.

VEZ-BR Segundo relatório, o Brasil será o sexto no mundo em investimento [Online] // Veículos de Emissão Zero. - 2012. - 03 09, 2016. - http://www.vezdobrasil.com.br/segundo-relatorio-o-brasil-sera-o-sexto-no-mundo-em-investimento/.

Wang Ruihu AdaBoost for Feature Selection, Classification and Its Relation with SVM, A Review [Journal] // Physics Procedia. - jan 2012. - Vol. 25. - pp. 800-807. - ISSN: 1875-3892 DOI: 10.1016/j.phpro.2012.03.160.

Weiss M. [et al.] Leveraging smart meter data to recognize home appliances [Conference] // Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on. - 2012. - pp. 190-197.

Welch P.D. The use of the fast fourier transform for the estimation of power spectra: a method based on time averaging over short modified periodograms [Article] // IEEE Transactions on Audio and Electroacoustics. - 1967.

Wichakool W. [et al.] Smart Metering of Variable Power Loads [Journal] // IEEE Transactions on Smart Grid. - 2015. - 1 : Vol. 6. - pp. 189-198. - ISSN: 1949-3053.

YOKOGAWA [Online]. - 2015. - 01 20, 2018. - https://tmi.yokogawa.com/br/solutions/products/data-acquisition-equipment/high-speed-data-acquisition/dl850edl850ev-scopecorder/.

Zadeh Lotfi A. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes [Journal] // Systems, Man and Cybernetics, IEEE Transactions on. - 1973. - 1 : Vols. SMC-3. - pp. 28-44. - ISSN: 0018-9472 DOI: 10.1109/TSMC.1973.5408575.

Zaremba W. and Sutskever I. Learning to Execute [Article] // CoRR. - 2014.

Zeifman M. and Roth K. Nonintrusive appliance load monitoring: Review and outlook [Journal] // IEEE Transactions on Consumer Electronics. - 2011. - 1 : Vol. 57. - pp. 76-84. - ISSN: 0098-3063.

Zeifman M. and Roth K. Viterbi algorithm with sparse transitions (VAST) for nonintrusive load monitoring [Conference] // Computational Intelligence Applications In Smart Grid (CIASG), 2011 IEEE Symposium on. - 2011. - pp. 1-8.

Zeifman Michael, Akers Craig and Roth Kurt Nonintrusive appliance load monitoring (NIALM) for energy control in residential buildings [Article] // Fraunhofer Center for Sustainable Energy Systems. - 2011.

Zhang Min-Ling and Zhou Zhi-Hua ML-KNN: A lazy learning approach to multi-label learning [Article] // Pattern Recognition. - [s.l.] : Elsevier, 2007.

Zhang Min-Ling and Zhou Zhi-Hua Multi-Label Neural Networks with Applications to Functional Genomics and Text Categorization [Article] // IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING. - [s.l.] : Elsevier, 2006.

Zhou Xi [et al.] HMM-Based Acoustic Event Detection with AdaBoost Feature Selection [Book Section] // Multimodal Technologies for Perception of Humans. - [s.l.] : Springer, Berlin, Heidelberg, 2007. - ISBN: 978-3-540-68584-5 978-3-540-68585-2 DOI: 10.1007/978-3-540-68585-2_33.

Zhu J. [et al.] Multi-class AdaBoost [Journal]. - 2006.

Ziarani A.K. and Konrad A. A method of extraction of nonstationary sinusoids, Signal Processing [Journal]. - 2004.

Zimmermann Jean-Paul [et al.] Household Electricity Survey: A study of domestic electrical [Article]. - 2012.

Zoha Ahmed [et al.] Non-Intrusive Load Monitoring Approaches for Disaggregated Energy Sensing: A Survey [Journal] // Sensors. - 2012. - 12 : Vol. 12. - p. 16838. - ISSN: 1424-8220 DOI: 10.3390/s121216838.

Zou Tongyuan [et al.] Polarimetric SAR Image Classification Using Multifeatures Combination and Extremely Randomized Clustering Forests [Journal] // EURASIP Journal on Advances in Signal Processing. - dec 2009. - 1 : Vol. 2010. - p. 465612. - ISSN: 1687-6180 DOI: 10.1155/2010/465612.

Appendix A

BRAD Dataset

Overview

One of the challenges in NILM research and development is the lack of publicly available data. In order to contribute to further research in NILM, the dataset generated during the development of this work has been made freely available.

The Brazilian Appliance Dataset (BRAD) contains measurements from grid voltage and current channels of 94 individual Brazilian appliances. The data were acquired using Yokogawa DL850EV ScopeCorder at a sample rate of 10 kHz.

In the following sections, the BRAD file format and the script used to generate the dataset are described. More information on downloading, using and contributing to BRAD can be found at http://brad.caldeira.tech/.

File format

Each BRAD file contains data from one measurement of one individual appliance. The files are in HDF5 format (https://www.hdfgroup.org/solutions/hdf5) and contain an HDF5 dataset object with two dimensions: voltage and current.

The voltage is the first dimension of the dataset (index 0) and is measured in Volts (V). The current is the second dimension (index 1) and is measured in Amperes (A). The HDF5 data type is 32-bit floating point. Besides the data from the actual samples, some metadata is associated with each dataset in the HDF5 file, as described in the Table 34.

Table 34 - Dataset metadata

Key	Description
type	Type of appliance
manufacturer	Manufacturer of the appliance
model	Model of the appliance
sampling_rate	Sampling rate of the meter, in Hertz
grid_voltage	Grid voltage, in Volts (rms)
voltage_hash	Hash of the elements of the voltage dimension concatenated*
current_hash	Hash of the elements of the current dimension concatenated*

* The hashes are calculated using the xxHash algorithm (http://cyan4973.github.io/xxHash/) and written in the metadata fields in hexadecimal format with capital letters.

Finally, the names of the files follow the pattern below:

{voltage_hash}_{current_hash}.h5

Sampling rate specifications

Future contributions to the BRAD should follow the sampling specification detailed in Table 35.

Table 35 – Sampling rate specifications

Sampling rate	Minimum 10 kHz
Voltage	Measured in Volts; minimum resolution of 0,1 V
Current	Measured in Amperes with values no greater than 400 A; minimum resolution of 0,1 A
ADC resolution	Minimum 12 bits for analog to digital conversion

Dataset generation script

Step	Action
1	Connect the meter and the appliance to the outlet
2	Connect the voltage and current probes
3	Turn on the appliance and the meter, but without recording
4	Check whether the voltage and current ranges are within the meter range (scale)
5	Check whether the voltage and current signals are in phase
6	Check whether the current signal does not saturate in each state of the appliance
7	Turn off the appliance
8	Start recording
9	Turn on the appliance
10	Plug and unplug the appliance power source 3 times
11	Activate all appliance power states (example for a TV appliance: change channels, increase and decrease volume, set standby mode)
12	Turn off the appliance
13	Finish recording
14	Convert meter data to BRAD file format, as specified above

Appendix B

A cost-effective smart meter

Armel and her colleagues (Armel, et al., 2013) presented a business case that calculated the cost of saved kWh, using power monitoring solutions, compared to the cost of a kWh provided by the utilities. Even the worst case scenario showed four times less cost per kWh for the energy efficiency solutions, including software-based disaggregation, which makes it highly viable for commercialization.

Under Armel's premises (Armel, et al., 2013) (10% of 1064kWh per household saved monthly), customers can save from USD 4.79 to USD 10.11 per month. If we take a payback time of twenty-four months, as the premise for users' disposition to go forward to an NILM platform, for energy saving, even worst case scenario (USD 4.79 saved per month) solution could not cost more than around USD 115.

With a price target defined, development tries to find the balance between **power** and **cost**. This requires presuppositions about the technical requirements of the desired application and platform. Thus, as prior requisite, research will evaluate energy distribution sector to find number of simultaneous sampling channels needed.

In the majority of European countries, power supply is provided to homeowners as 50Hz, $230V_{rms}$. United States residential buildings commonly use two phases with 60Hz and $120V_{rms}$. In Brazil this depends on the region and the electricity utilities: In southeast and southern Brazil, distribution can be up to a three-phase supply, with a phase offset of 120^{o} between each other and magnitude of $127V_{rms}$; in northern Brazil the power supply is $220V_{rms}$.

Therefore, to achieve the minimum prerequisites for a three phase NILM monitor, that satisfies Brazil's requirements, the meter needs to be able to sample at least **five channels** simultaneously: **two voltage** channels (V_{AB}, V_{BC}), with the desired three voltage measurement (for zero sequence or neutral problem verifications), and **three current** channels (I_A, I_B, I_C). In order to link time-stamped data, the meter would also require a *Real Time Clock* (RTC).

This thesis adopted a 16 bit word-length as its premise for signal resolution. Within the feature extraction rate and the features set, defined in Sections 6.2 and 6.3, respectively, in one hour, the system will generate:

- 30 features x 2 bytes (quantization) x 8 k (number of 450ms in 1h) = 480kB (of raw data);
- 8 bytes (date and time) for each epoch time (450ms) = 64kB

Thus, 544kB of data per hour is required to store the results, for disaggregation purposes. Thus, for a minimum 90 days of raw data in mass storage, and without overwriting any previous record in memory, the system will require at least 1.175GB. This includes only raw information data of measurements and doesn't evaluate space to program code, calibration, and configuration or parameterization.

Available platforms

In 2012, Pereira (Pereira, et al., 2012) presented the design of a platform using a "Netbook" as hardware for acquiring and processing data. The proposed idea was for more than just the purposes of NILM and included an "Eco-Feedback Research", which used a built-in camera and microphone to sensor human activity.

The proposed sampling system used an audio input TRS plug to capture 2 channels, voltage and current, via the computer's sound card. The platform proposal (Pereira, et al., 2012) also used Wi-Fi (IEEE 802.11) for communication platform, which outperformed desired throughput with rates up to 600Mbps in 2.4GHz spectrum (IEEE 802.11, 2012), and, mass storage probably exceeds minimum requirement with nowadays hard drives sizes. Unfortunately, the sound cards that are embedded in "Netbooks" today, do not provides the minimum number of sampling channels for metering three-phase systems; they only have two sampling channels available in their sound cards.

The cost of a "Netbook", as an NILM meter platform, is also prohibitive to mass adoption in most of houses, especially in developing countries, such as Brazil. As stated by Armel (Armel, et al., 2013), payback will possibly be long enough to overtake consumers' expectations.

Trung and his colleagues (Trung, et al., 2013) introduced the use of *Field Programmable Gate Array* (FPGA) to disaggregate loads. Their proposed architecture, with a market price estimated at €150, gathered data up to 500 kHz and processed the entire algorithm in parallel mode. Notwithstanding this, system is not yet commercially available and it is not suitable for mass adoption because of its high target price.

In contrast, commercially available embedded platforms, such as Raspberry Pi3 (RASPBERRY, 2016), embed a 1.2GHz quad-core ARM CPU, Wi-Fi and Bluetooth wireless communication technologies, with a 32GB microSD card, and cost USD 55 in online stores (MOMYPI, 2016). Unfortunately, the number of channels that are available in Raspberry soundcards do not meet the minimum requisites for a three-phase metering, either.

In our quest to meet all the desired specifications and price target, we searched for a more embedded and customizable solution such as the *System on Chip* (SoC) from Texas Instruments (TI, 2016). The SoC CC2650 (TI CC2650, 2016), provides wireless communication (multi-protocol standard for Bluetooth and Zigbee) with a Cortex-M0 dedicated processor, a main 32-bit ARM Cortex-M3 (48 MHz), an extra processor called *Sensor Controller* as auxiliary ultralow-power processing unit. This microcontroller also integrates an internal temperature sensor, a *Real Time Clock*, 8 analog input channels with a 200 kS/s rate, SPI, I^2C and UART communication ports. The price for a single chip, buying it directly from TI, for 1,000 units is USD 3.30 per unit (TI CC2650, 2016).

In order to manufacture ten complete NILM meters (including microcontrollers, communication circuits, power supply, voltage and current isolated transducers, analog filtering, signal conditioning, circuit board, resistors, capacitors, inductors, microSD card holders, case, plugs, extra components, soldering and assembly services), we calculate that the unit cost is around USD 120, using a famous electronic parts seller (DIGI-KEY, 2016). It is important to mention the value of each unit decreases exponentially as production scale grows.

To meet the specifications for mass storage, the platform will require an external device, using SPI communication, for recording raw data. Flash memories cards, such as microSD cards, are affordable solutions for embedded mass storage. Therefore, the complete solution, including a USD 10 32GB micro SD card (AMAZON, 2016), which

was bigger than need to storage 90 days of data, and had a target cost of USD 130. This is close to the previously established price and meets all the requirements for an NILM meter.

Platform structure

Internally, NILM meters are divided in five different functional blocks, as shown in Figure 88 – a power supply unit (A), with an embedded AC-DC converter to $3.3V_{DC;}$ analog front-end (B) with components to do signal conditioning of voltage and current to meet microcontroller ADC specifications ($3.3V_{pp}$ signal excursion with $1.65V_{offset}$); (C) anti-aliasing filter (-30dB @ 5 kHz by Nyquist-Shannon sampling theorem for sampling rates of 10 kHz). Complimentary, research provided development of analog anti-aliasing filter.

The platform also used a microSD as mass storage (D), implemented through SPI mode (SDGROUP, 2013); and used Bluetooth (E) as a wireless communication standard. SPI mode is a secondary communication protocol offered by Flash SD devices and contains a subset of the SD Memory Card protocol commands.

The SPI implementation offered by microSD cards is capable of using an off-the-shelf host, which allows the card to transfer data directly to embedded microcontroller SPI ports. A disadvantage is that SPI mode has a lower performance when compared to SD's native mode.

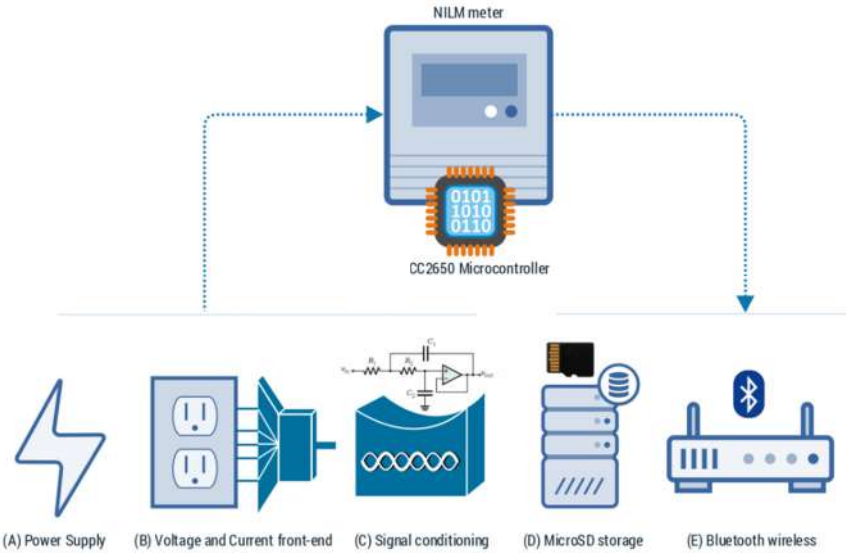

Figure 88 - NILM meter main components (A) Internal power supply unit (B) Voltage and current transducers (C) Filtering and signal conditioning circuits (D) MicroSD mass storage (E) Bluetooth smart 4.1 wireless and TI CC2650 microcontroller

SPI Bus protocol

The SPI mode uses a *four-wire* serial bus to communicate with a microcontroller host: clock (SCK); Master-Out Slave-In (MOSI); Master-In Slave-Out (MISO); and Slave Select (SS). Its protocol is Master-Slave oriented and was initially developed by Motorola. The SPI protocol for microSD cards consists of a command, a response and, when necessary, the transfer of data-blocks and tokens. Command size is fixed at a 6-byte length, and replies for each command can assume an assorted byte length, as described in specifications (SDGROUP, 2013).

SPI protocol needs an initialization method. To enter into SPI mode, the card needs at least 74 cycles of the clock with the SS line in high (logic level 1). The procedure must be executed with a low frequency operation rate (usually 100 kHz as clock). After initialization, the card will wait for the CMD0 command to reset and will then enter into *SPI Operation Mode*. Faster clocks could now be configured to obtain higher transfer rates.

After the initialization process, the flash memory card goes into *Data Transfer Mode*. The block length for modern SDHC and SDXC cards is fixed at 512 bytes. Sending and receiving data blocks on SD cards can be made in a single transfer mode, where just 512 bytes are transferred, or, in multiple block mode, where the host defines the number of blocks to be transferred and sends all the data serialized.

Modern SD cards are optimized to use multiple block transfer mode, where pre-erase command (ACMD23) can be made in order to optimize the *write* throughput, and *read* process can achieve specifications of sustained data rates.

Anti-aliasing filter

It is necessary that the low-pass anti-aliasing filter precede the analog-to-digital converter stage, to limit the bandwidth of the signal to be digitalized; thus, avoiding aliasing. Theory is broadly studied in book "*Digital Signal Processing: System Analysis and Design*" (Diniz, et al., 2010).

The analog-to-digital converter embedded hardware in the microcontroller allows a maximum signal excursion of $3.3V_{pp}$, and negative voltage (from the microcontroller ground reference) is prohibited. The grid voltages and currents and must be measured by isolated transducers and signal conditioned to meet electrical specifications.

Our initial project for an anti-aliasing filter followed the proposed methodology of Gao and his colleagues (Gao, et al., 2010): A low-pass filter, with stop-band frequency of 5kHz (attenuation of -30dB) indicated a fourth-order Butterworth filter, realized by Multiple-Feedback or Sallen-key circuits. However, in order to implement an Nth-order filter, the project requires $\frac{N}{2}$ stages. Thus, 2 stages were required for implementation of anti-aliasing filter, employing 2 operational amplifiers for each channel.

Nevertheless, a topology from Boctor (Boctor, 1975) allowed the implementation of a 3rd order Inverse Chebyshev filter, with improved pass-band and similar stop-band as designed using 4th order Butterworth filter. Figure 89 shows that a 3rd order Boctor topology has similar performance of a 4th order Butterworth filter, although with reduced cost.

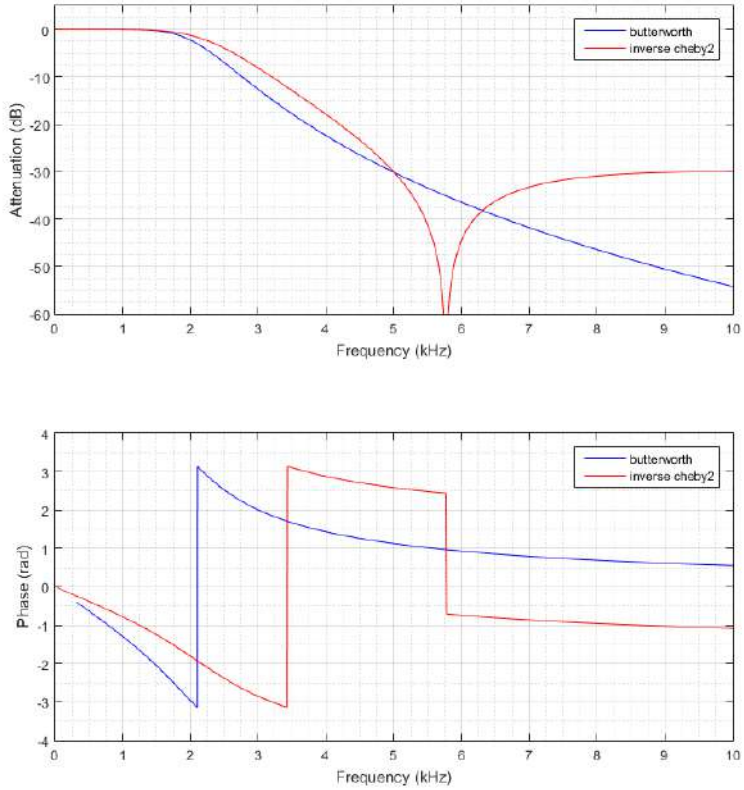

Figure 89 – Magnitude gain (top) and Phase response (bottom) comparison of 4th order Butterworth and 3rd order Inverse Chebyshev low-pass anti-aliasing filter

Smart meter development

The initial development of an NILM meter used large SMD components and delivered modularity for a comprehensive test environment, enabling a solely hand-made platform without requiring a machine to do welding. Three different boards were designed: a main motherboard, a voltage-current measurement daughterboard, and a dual-current measurement daughterboard.

The motherboard was developed to accommodate a microcontroller, a power supply, wireless components, general purpose input and output such as buttons and LED´s, an OLED display, an anti-aliasing filter, a microSD card, and up to four coupled daughterboards. The voltage-current daughterboard was designed to measure one voltage channel and one current channel, performing isolation and signal conditioning. The dual-current daughterboard was designed to measure 2 isolated currents with signal conditioning. Projected first in 3D and assembled in LEMT Laboratory, the complete NILM meter configuration is shown in Figure 90 (a) – with 3D CAD project and (b),

with the motherboard displayed in green, the voltage-current daughterboard in red and the dual-current in dark blue.

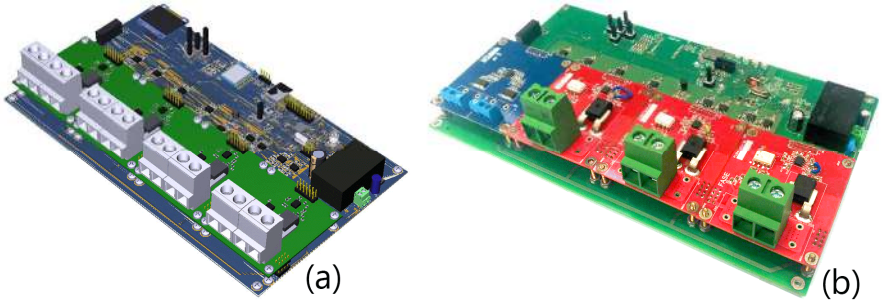

Figure 90 – (a) 3D Project of smart meter and (b) version of developed prototype

Our firmware development also included a Bluetooth low energy stack, using TI-RTOS (a real-time operating system with a preemptive kernel task scheduler and drivers for some hardware).

The CC2650 is a multi-core system, thus, the development used a concurrent computing paradigm. The main processor (Cortex-M3) was responsible for the feature measurements: processing the sampled data and storage of computed features on the microSD card, running SPI protocol, as shown in Figure 91. TI-RTOS provided a driver for SPI hardware using embedded µDMA (micro-direct memory access), therefore releasing the CPU to realize tasks and other computations while data was transferred, in parallel, to mass storage.

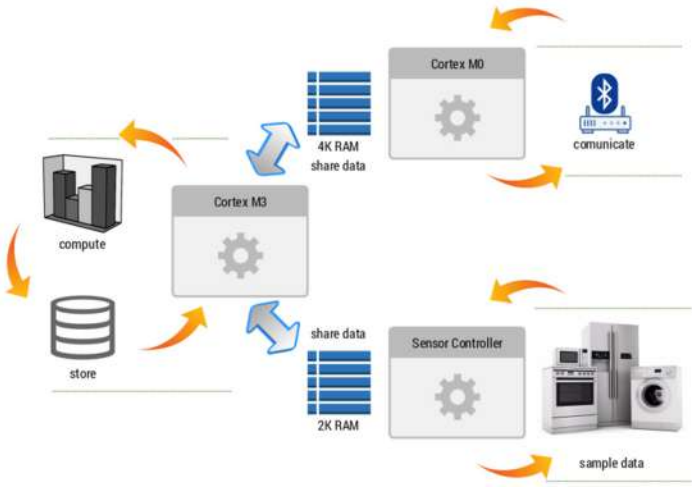

Figure 91 - Multi-core processing scheme

The *Sensor Controller* CPU assumed sampling data for each individual channel, using a double-buffer strategy to wake up the main CPU when one buffer was entirely filled. Distinct memory buffers were chosen to avoid *Readers-Writers* problem.

The NILM meter actually evaluated the 15 electrical features, defined by this thesis, using a feature extraction rate of 450ms (27 cycles). Therefore, future measurements from developed smart meter could be used in research and development of future NILM studies. A miniaturized version of the smart meter, for an automated machine welding process, was also developed for home and office use, and is compatible with DIN rail EN50022, similar to that used to mount circuit breakers, as shown in Figure 92.

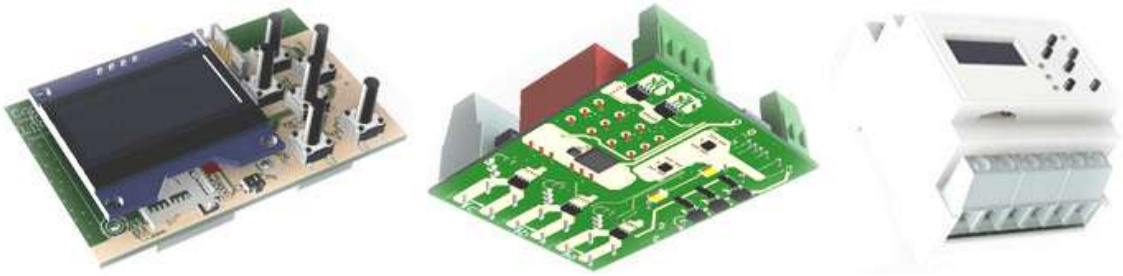

Figure 92 – 3D project of miniaturized version of NILM smart-meter

More information on downloading, using and contributing NILM meter can be found at http://smartmeter.caldeira.tech/.

Appendix C

Appliances analysis

Notebook analysis

Table 36 – Notebook / Metrics summary

	Accuracy			F-Measure			Precision			Recall		
with Full Features	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*
Deep Learning	99	94	135	99	94	130	99	94	140	99	94	130
SVM	99	94	58	93	88	58	99	94	70	92	87	58
Logistic Regression	94	89	3	53	48	3	70	65	10	50	45	3

	TECA			NEAP			Train Time	Predict Time
with Full Features	*max%*	*-5%*	*cycle*	*min%*	*+5%*	*cycle*	*feasible time cycle*	*feasible time cycle*
Deep Learning	97	92	300	6	11	100	10	3
SVM	96	91	300	7	12	45	5	16
Logistic Regression	94	89	300	17	22	230	2	1

Table 37 – Notebook / Summary of Feature Selection

Deep Learning	**Feature List**																														
90 cycles	*Features*						*Fourier Harmonics*																								data
Method	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	Size	% red.
Full set	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	30	
Extra-Trees	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓		✓	✓			✓			✓						✓			17	43%
M. Inform.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	30	0%
AdaBoost	✓		✓		✓	✓													✓												5	83%

Table 38 – Notebook / Summary of complete solution

Solution	Method	Feature Selection Method	Num. Features	F. Extraction Rate
	Deep Learning	Extra-Trees	17	90 cycles

Selected Features	*Features*						*Fourier Harmonics*																							
	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓		✓	✓			✓			✓						✓		

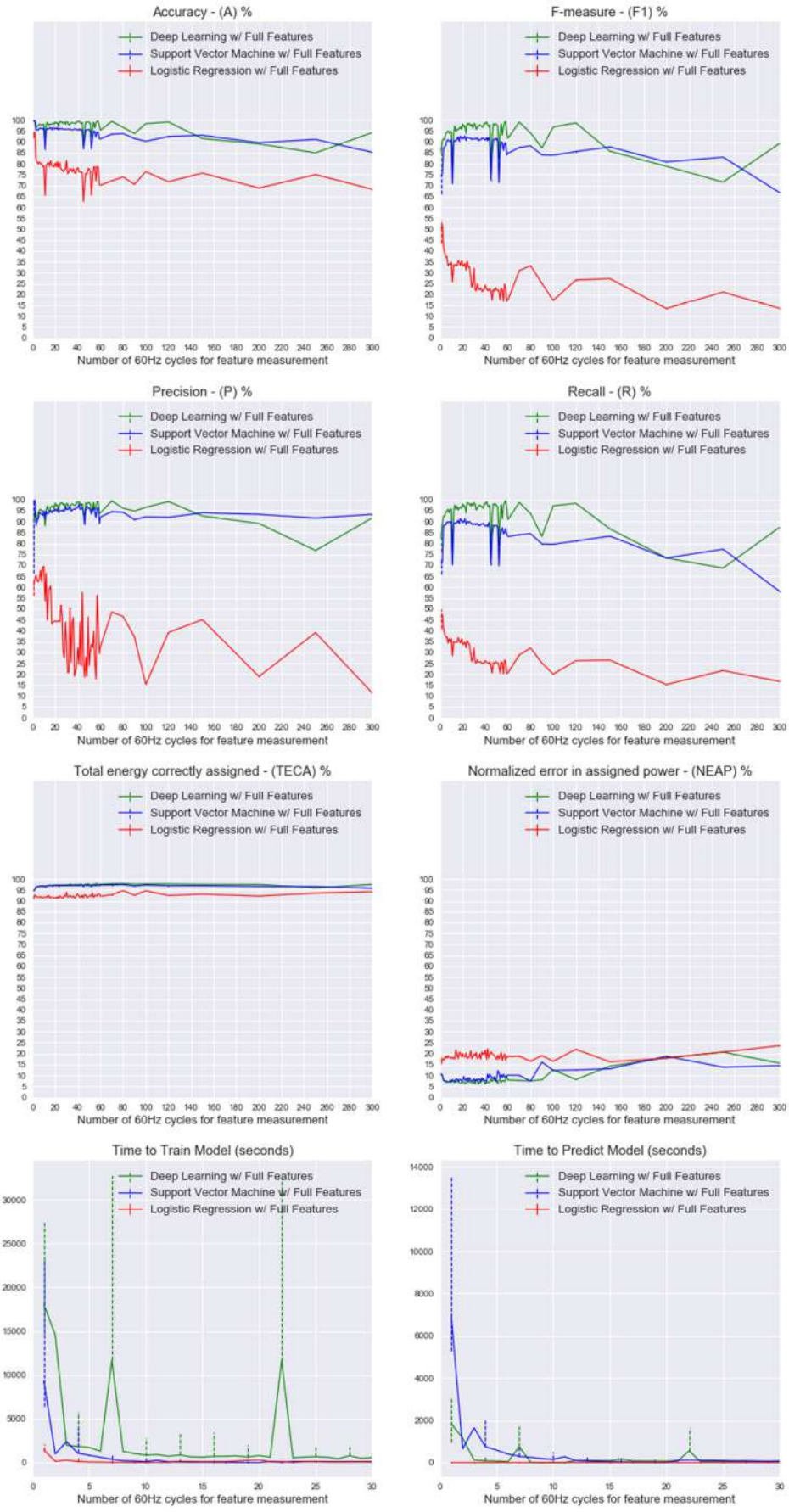

Figure 93 – Notebook / NILM Methods comparison with full features set

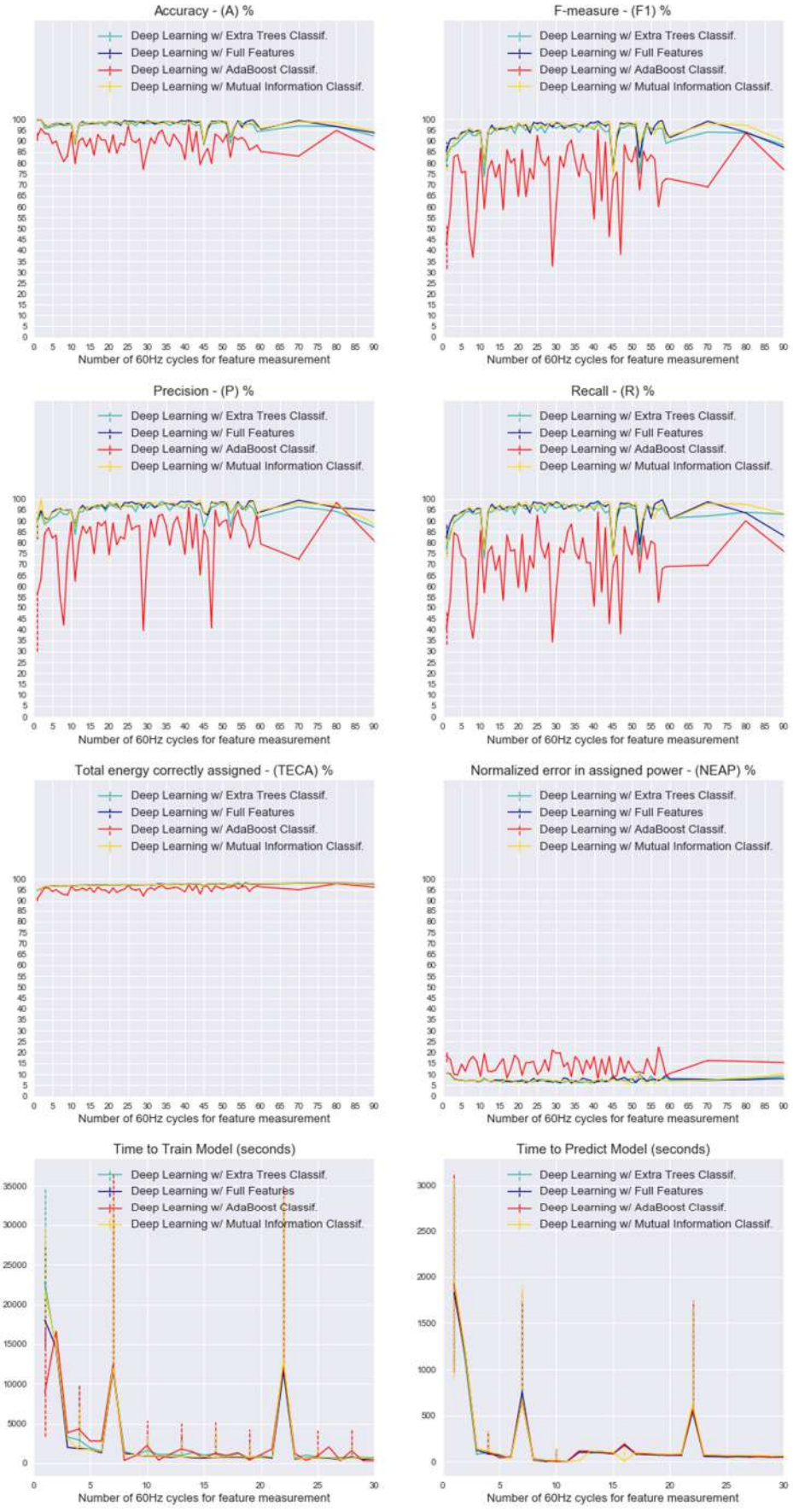

Figure 94 – Notebook / Deep Learning Feature Selection methods comparison

LCD Monitor analysis

Table 39 – LCD Monitor / Metrics summary

	Accuracy			F-Measure			Precision			Recall		
with Full Features	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*
Deep Learning	97	92	250	93	88	300	97	92	250	96	91	250
SVM	90	85	60	90	85	20	97	92	100	90	85	100
Logistic Regression	85	80	4	50	45	3	68	63	4	45	40	6

	TECA			NEAP			Train Time	Predict Time
with Full Features	*max%*	*-5%*	*cycle*	*min%*	*+5%*	*cycle*	*feasible time cycle*	*feasible time cycle*
Deep Learning	99	94	300	3	8	70	30	2
SVM	99	94	300	3	8	35	7	10
Logistic Regression	94	89	200	20	25	20	5	1

Table 40 – LCD Monitor / Summary of Feature Selection

Deep Learning	**Feature List**																														
40 cycles	*Features*						*Fourier Harmonics*																								data
Method	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	Size	% red.
Full set	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	30	
Extra-Trees	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓		✓	✓	✓	✓	✓		✓		✓	25	17%
M. Inform.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		29	3%
AdaBoost	✓					✓	✓	✓			✓		✓		✓	✓				✓		✓		✓							11	63%

Table 41 – LCD Monitor / Summary of complete solution

Solution	Method	Feature Selection Method	Num. Features	F. Extraction Rate
	Deep Learning	Extra-Trees	25	40 cycles

Selected Features	*Features*						*Fourier Harmonics*																							
	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓		✓	✓	✓	✓	✓		✓		✓

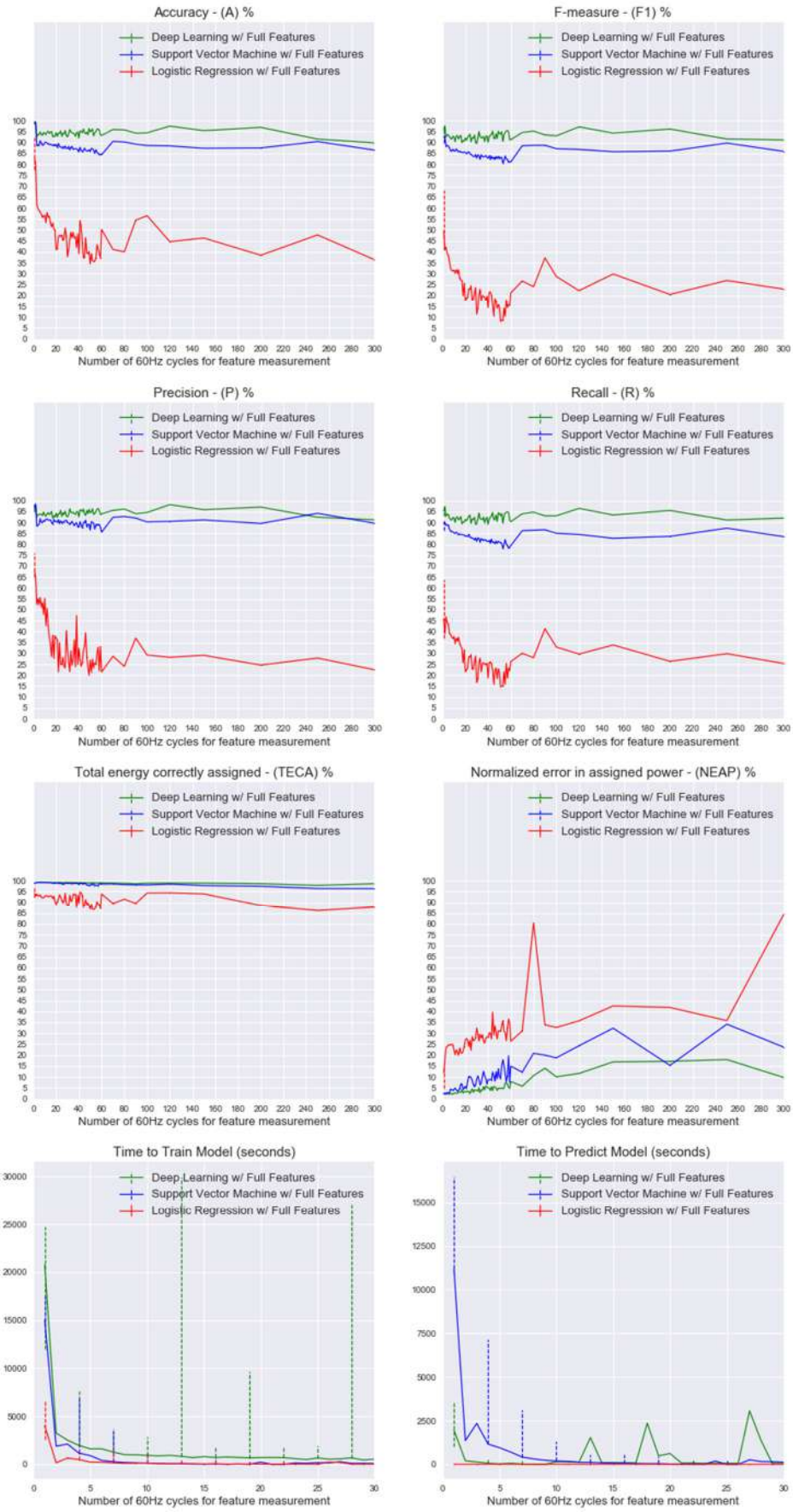

Figure 95 – LCD Monitor / NILM Methods comparison with full features set

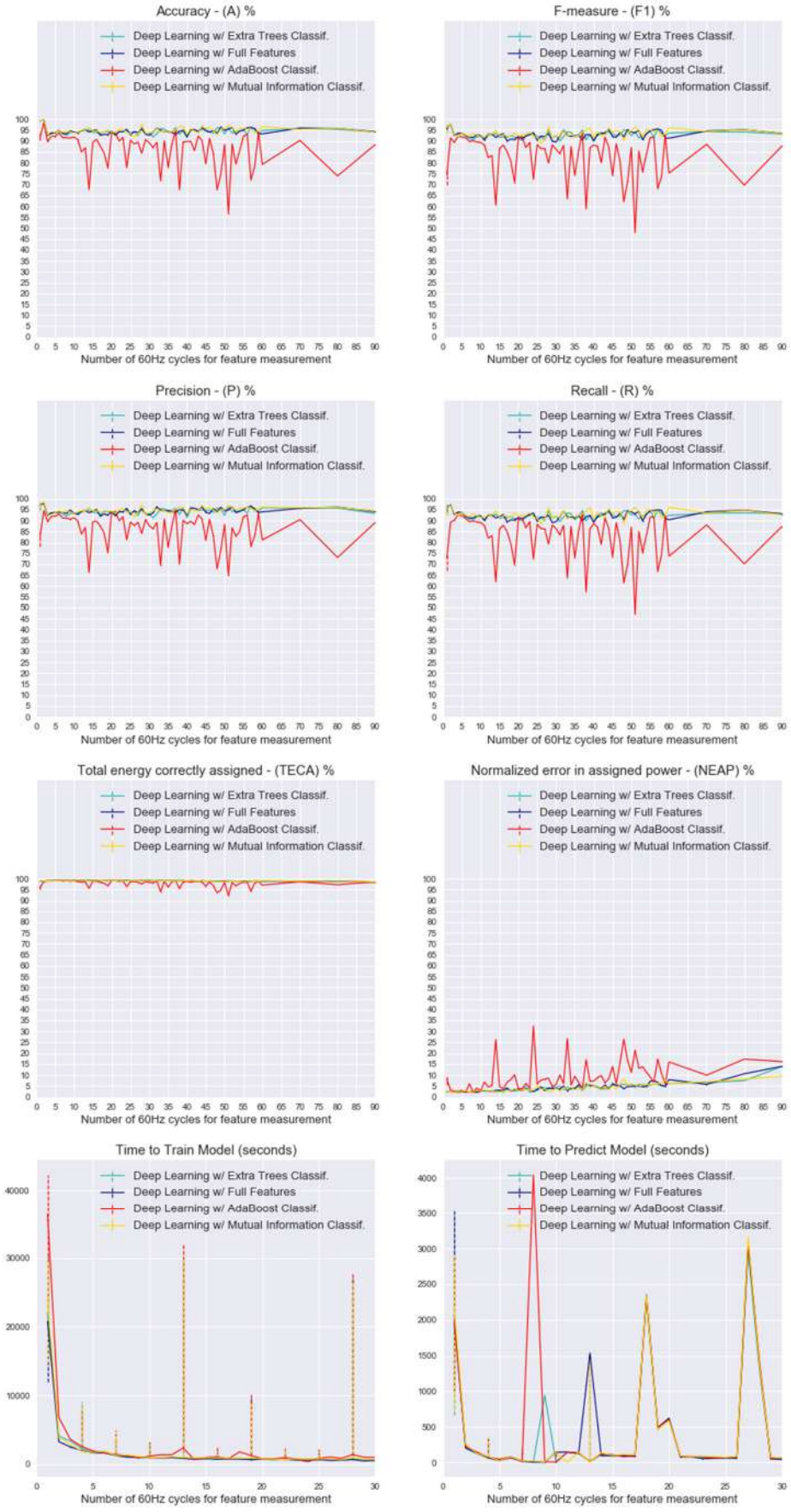

Figure 96 – LCD Monitor / Feature Selection methods comparison

Air conditioner analysis

Table 42 – Air conditioner / Metrics summary

	Accuracy			F-Measure			Precision			Recall		
with Full Features	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*
Deep Learning	97	92	65	95	90	25	96	91	25	95	90	16
SVM	96	91	12	89	84	7	95	90	5	85	80	6
Logistic Regression	80	75	5	50	45	2	65	60	2	45	40	3

	TECA			NEAP			Train Time	Predict Time
with Full Features	*max%*	*-5%*	*cycle*	*min%*	*+5%*	*cycle*	*feasible time cycle*	*feasible time cycle*
Deep Learning	96	91	90	8	13	90	15	7
SVM	96	91	70	8	13	70	9	15
Logistic Regression	93	88	5	12	17	13	3	1

Table 43 – Air conditioner / Summary of Feature Selection

Deep Learning	**Feature List**																														
16 cycles	*Features*						*Fourier Harmonics*																								data
Method	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	Size	% red.
Full set	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	30	
Extra-Trees	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓		✓	✓		✓						✓		✓		✓	19	37%
M. Inform.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	30	0%
AdaBoost	✓	✓	✓			✓	✓	✓																							6	80%

Table 44 – Air conditioner / Summary of complete solution

Solution	Method	Feature Selection Method	Num. Features	F. Extraction Rate
	Deep Learning	Extra-Trees	19	16 cycles

Selected Features	*Features*						*Fourier Harmonics*																							
	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓		✓	✓		✓						✓		✓		✓

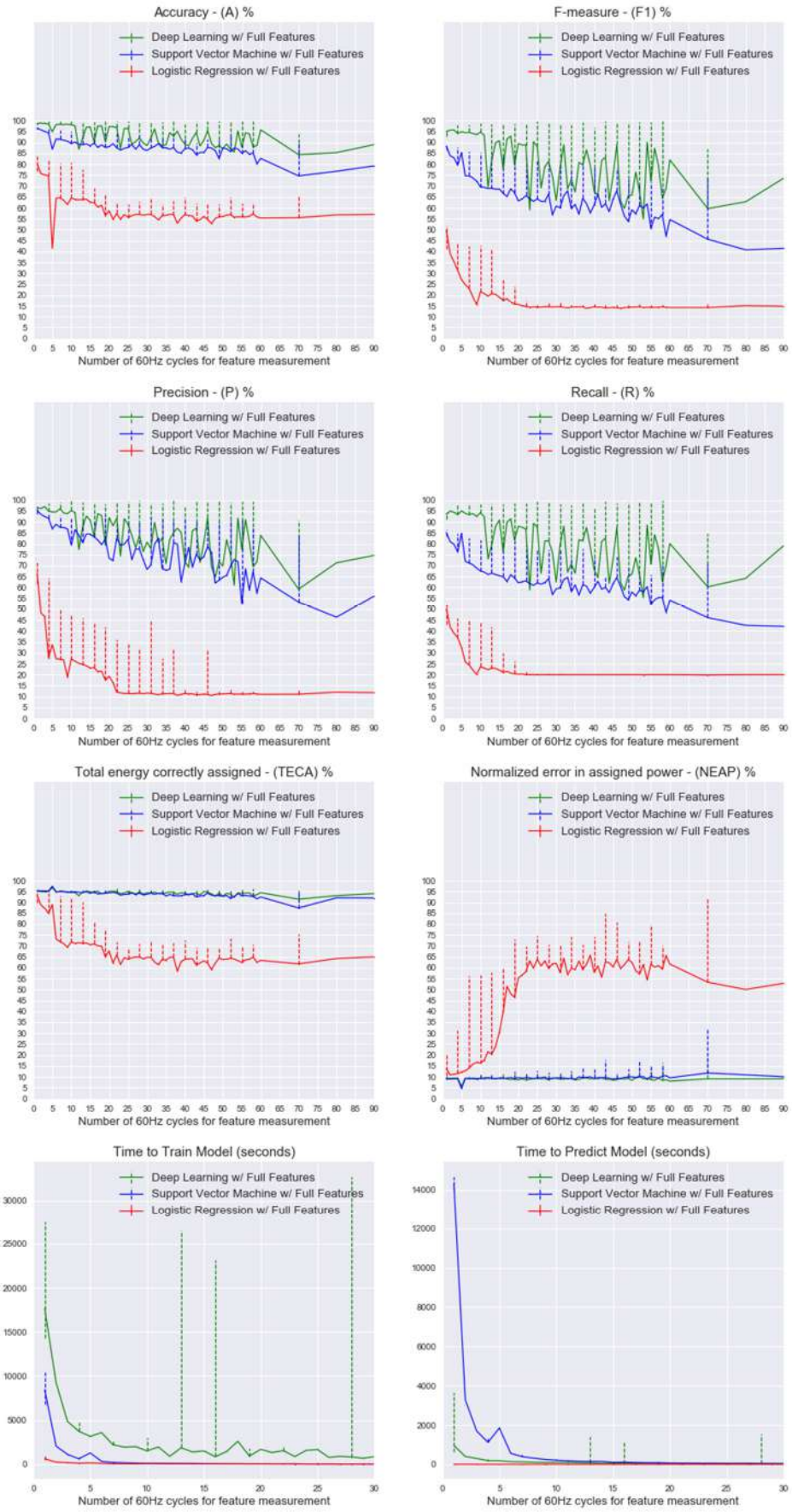

Figure 97 – Air conditioner / NILM Methods comparison with full features set

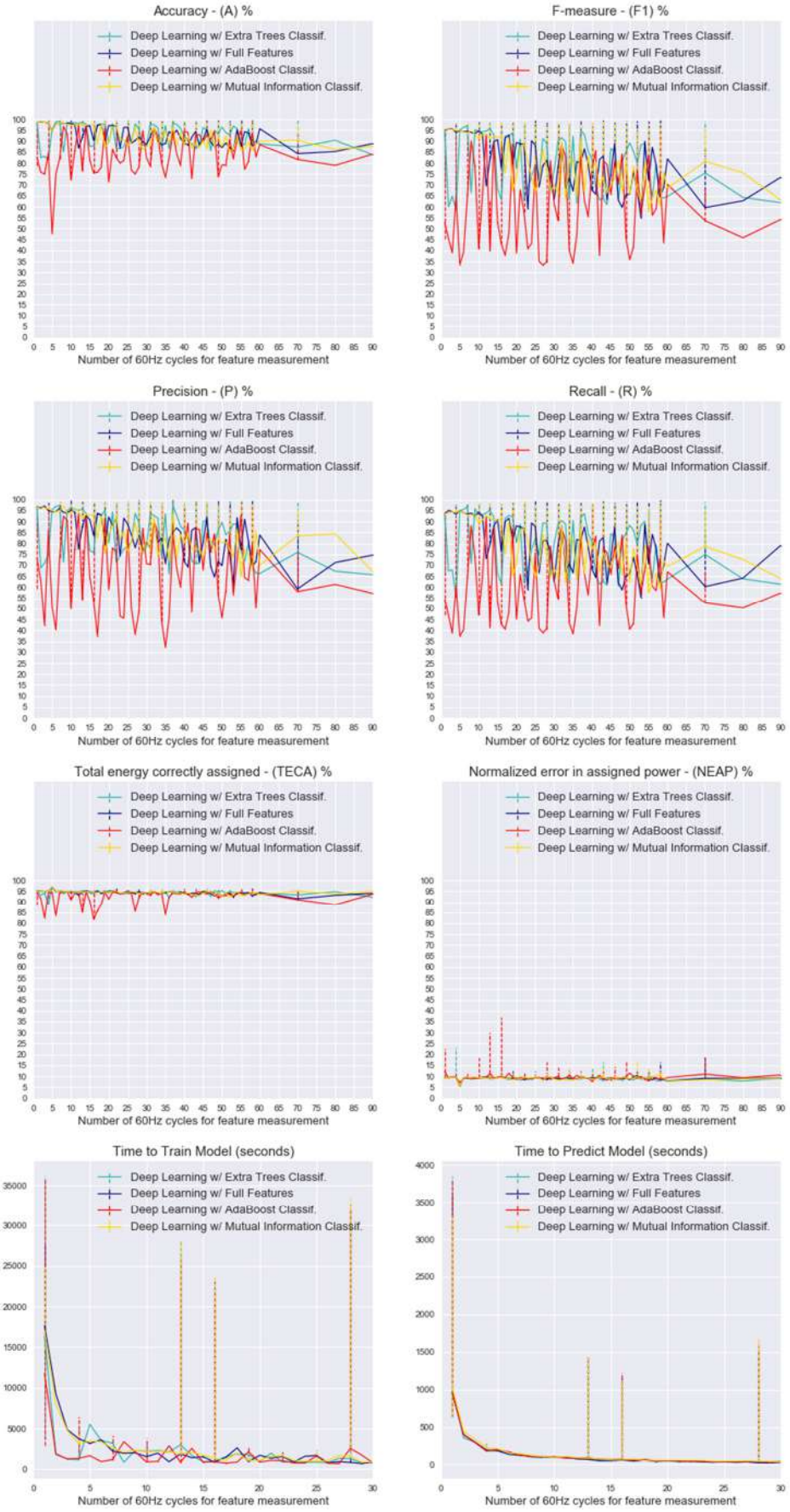

Figure 98 – Air conditioner / Feature Selection methods comparison

Fridge analysis

Table 45 – Fridge / Metrics summary

	Accuracy			F-Measure			Precision			Recall		
with Full Features	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*
Deep Learning	96	91	17	90	85	22	95	90	17	90	85	23
SVM	97	92	17	85	80	14	95	90	5	90	85	3
Logistic Regression	80	75	4	75	70	3	80	75	2	70	65	3

	TECA			NEAP			Train Time	Predict Time
with Full Features	*max%*	*-5%*	*cycle*	*min%*	*+5%*	*cycle*	*feasible time cycle*	*feasible time cycle*
Deep Learning	95	90	55	10	15	17	10	5
SVM	95	90	15	10	15	17	5	13
Logistic Regression	80	75	5	25	30	6	3	1

Table 46 – Fridge / Summary of Feature Selection

Deep Learning	**Feature List**																													
17 cycles	*Features*						*Fourier Harmonics*																							data
Method	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	Size	% red.
Full set	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	30	
Extra-Trees	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓		✓		✓											16	47%
M. Inform.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓	26	13%
AdaBoost	✓				✓	✓	✓	✓	✓									✓		✓										✓	9	70%

Table 47 – Fridge / Summary of complete solution

Solution	Method	Feature Selection Method	Num. Features	F. Extraction Rate
	Deep Learning	Extra-Trees	16	17 cycles

Selected Features	*Features*						*Fourier Harmonics*																							
	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓		✓		✓										

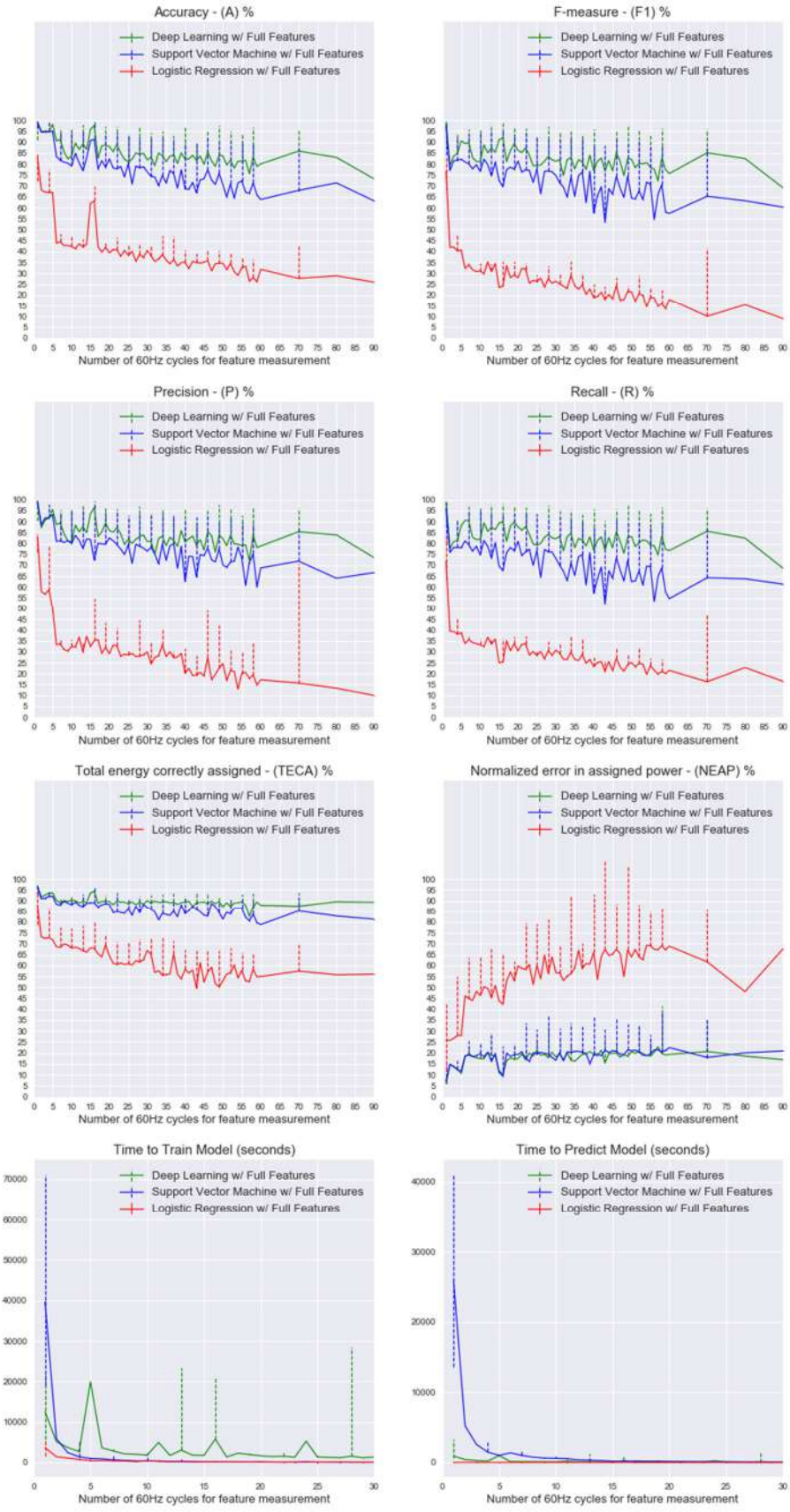

Figure 99 – Fridge / NILM Methods comparison with full features set

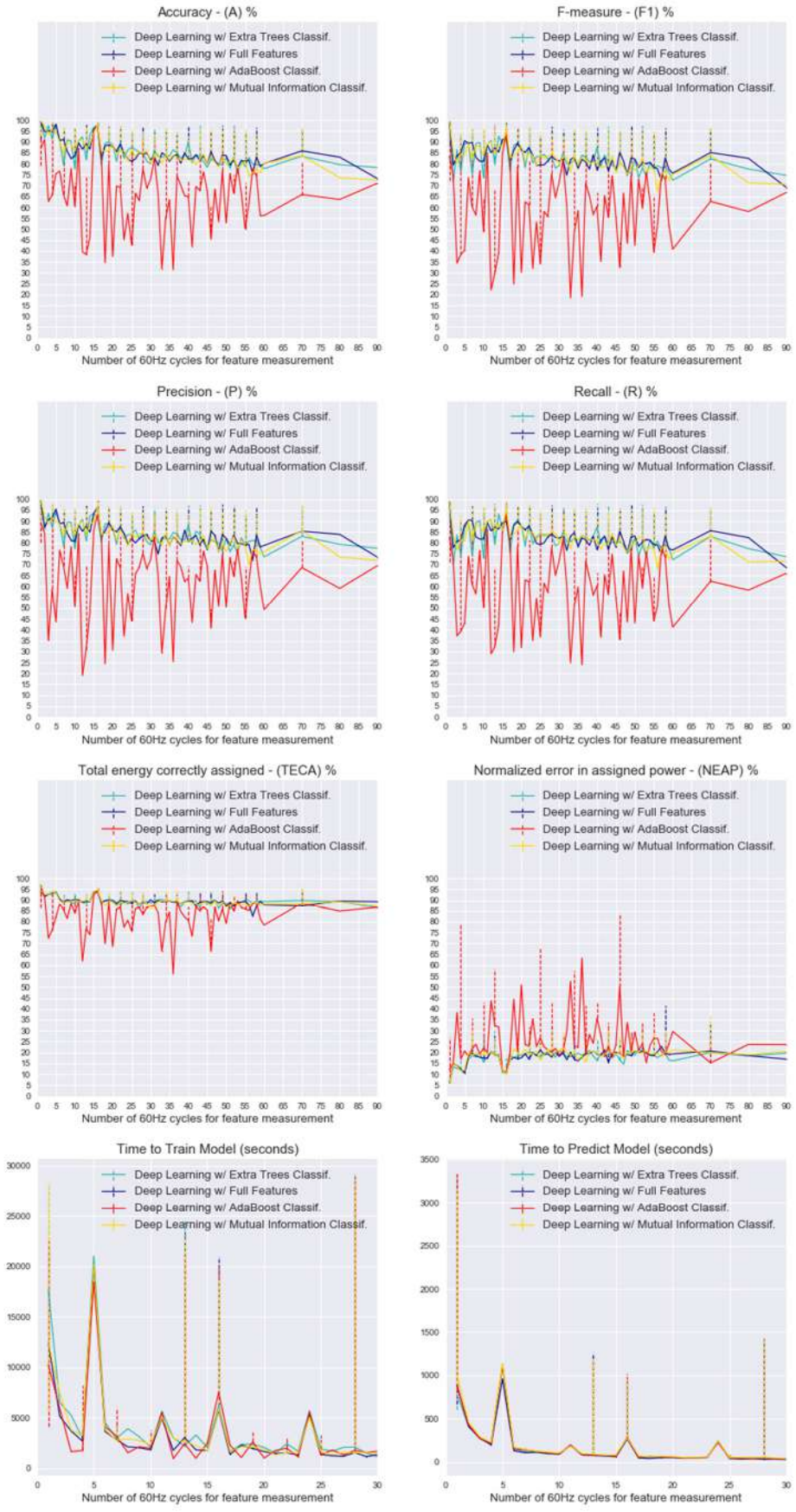

Figure 100 – Fridge / Feature Selection methods comparison

Washing machine analysis

Table 48 – Washing machine / Metrics summary

	Accuracy			F-Measure			Precision			Recall		
with Full Features	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*
Deep Learning	97	92	120	97	92	120	97	92	120	96	91	120
SVM	95	90	18	90	85	43	95	90	10	90	85	42
Logistic Regression	55	50	17	45	40	10	55	50	10	45	40	8

	TECA			NEAP			Train Time	Predict Time
with Full Features	*max%*	*-5%*	*cycle*	*min%*	*+5%*	*cycle*	*feasible time cycle*	*feasible time cycle*
Deep Learning	94	89	60	12	17	42	5	4
SVM	94	89	60	12	17	3	5	10
Logistic Regression	80	75	8	40	45	2	2	1

Table 49 – Washing machine / Summary of Feature Selection

Deep Learning	**Feature List**																													
42 cycles	*Features*						*Fourier Harmonics*																							data
Method	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	Size	% red.
Full set	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	30	
Extra-Trees	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓											18	40%
M. Inform.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓									21	30%
AdaBoost	✓	✓	✓	✓	✓	✓	✓	✓																	✓			✓			10	67%

Table 50 – Washing machine / Summary of complete solution

Solution	Method	Feature Selection Method	Num. Features	F. Extraction Rate
	Deep Learning	AdaBoost	10	42 cycles

Selected Features	*Features*						*Fourier Harmonics*																							
	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	✓	✓	✓	✓	✓	✓	✓	✓																	✓			✓		

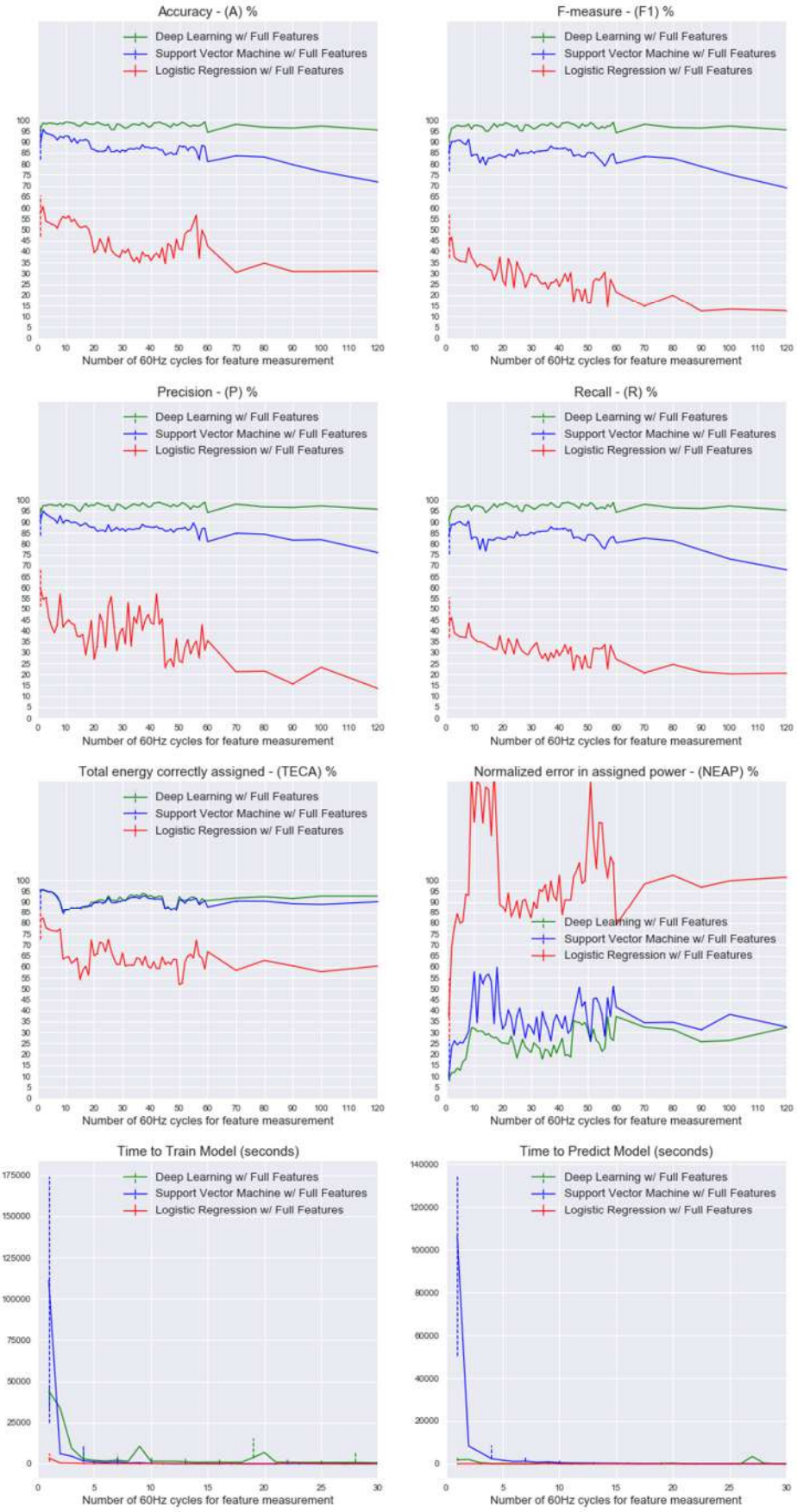

Figure 101 – Washing machine / NILM Methods comparison with full features set

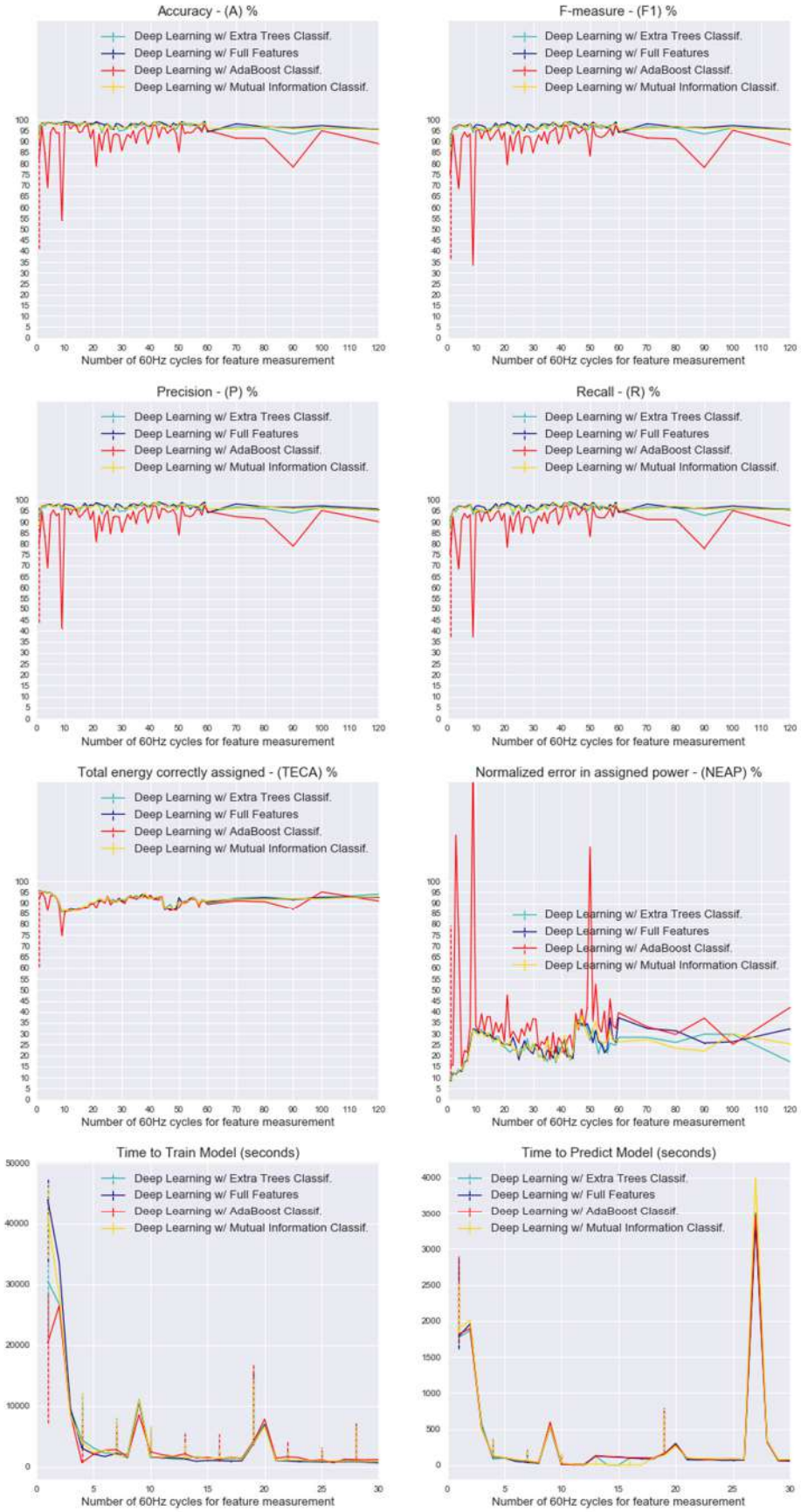

Figure 102 – Washing machine / Feature Selection methods comparison

Microwave oven analysis

Table 51 – Microwave oven / Metrics summary

	Accuracy			F-Measure			Precision			Recall		
with Full Features	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*
Deep Learning	97	92	42	95	90	25	95	90	42	93	88	35
SVM	94	89	42	74	69	5	85	80	15	70	65	15
Logistic Regression	90	85	15	45	40	3	50	45	5	45	40	3

	TECA			NEAP			Train Time	Predict Time
with Full Features	*max%*	*-5%*	*cycle*	*min%*	*+5%*	*cycle*	*feasible time cycle*	*feasible time cycle*
Deep Learning	98	93	90	3	8	90	12	5
SVM	98	93	90	3	8	90	5	10
Logistic Regression	95	90	25	7	12	22	1	1

Table 52 – Microwave oven / Summary of Feature Selection

Deep Learning	**Feature List**																													
25 cycles	*Features*						*Fourier Harmonics*																						data	
Method	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	Size	% red.
Full set	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	30	
Extra-Trees	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	27	10%
M. Inform.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	28	7%
AdaBoost		✓	✓		✓	✓	✓		✓											✓						✓					8	73%

Table 53 – Microwave oven / Summary of complete solution

Solution	Method	Feature Selection Method	Num. Features	F. Extraction Rate
	Deep Learning	AdaBoost	8	25 cycles

Selected Features	*Features*						*Fourier Harmonics*																							
	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
		✓	✓		✓	✓	✓		✓											✓						✓				

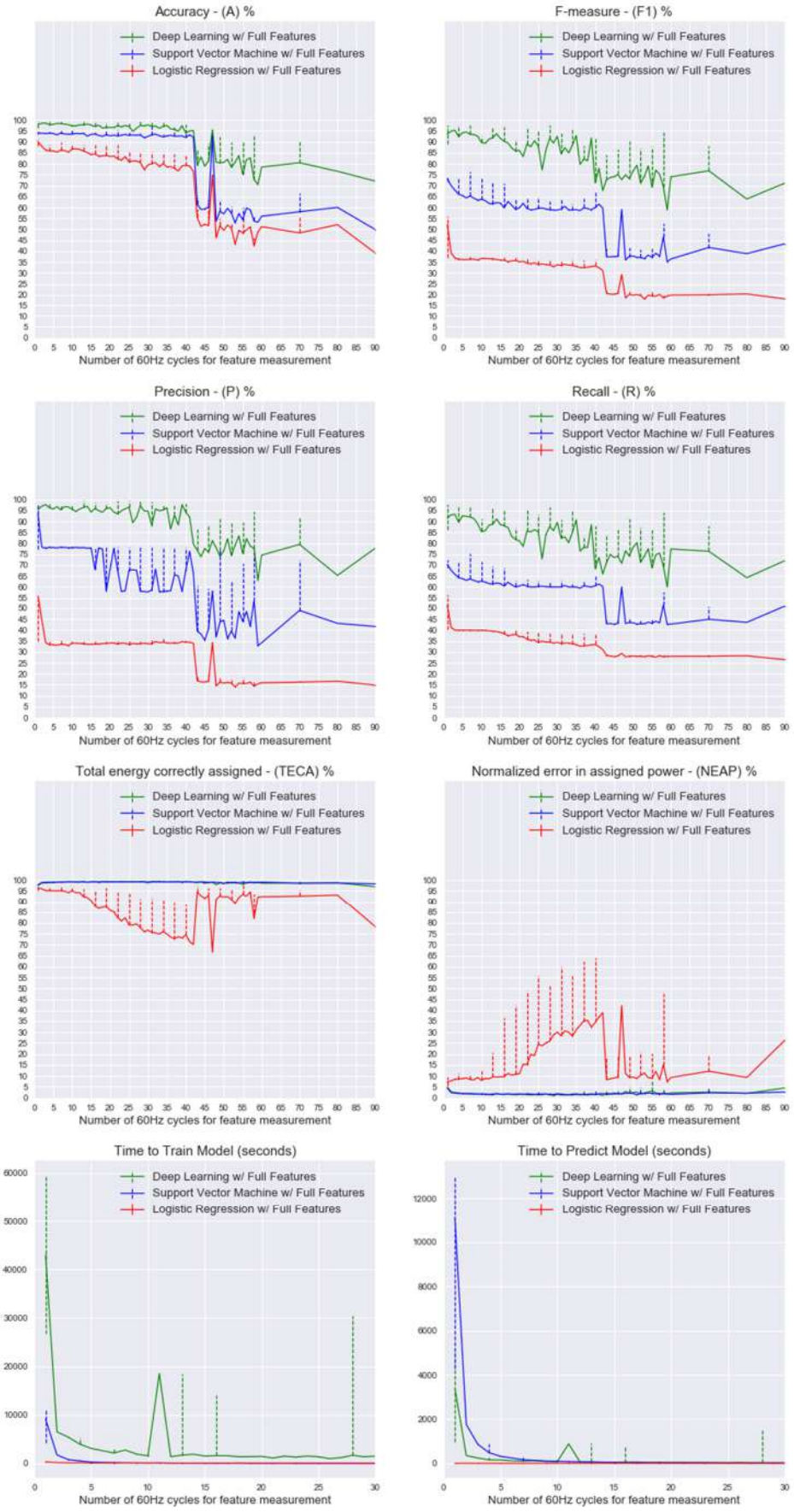

Figure 103 – Microwave oven / NILM Methods comparison with full features set

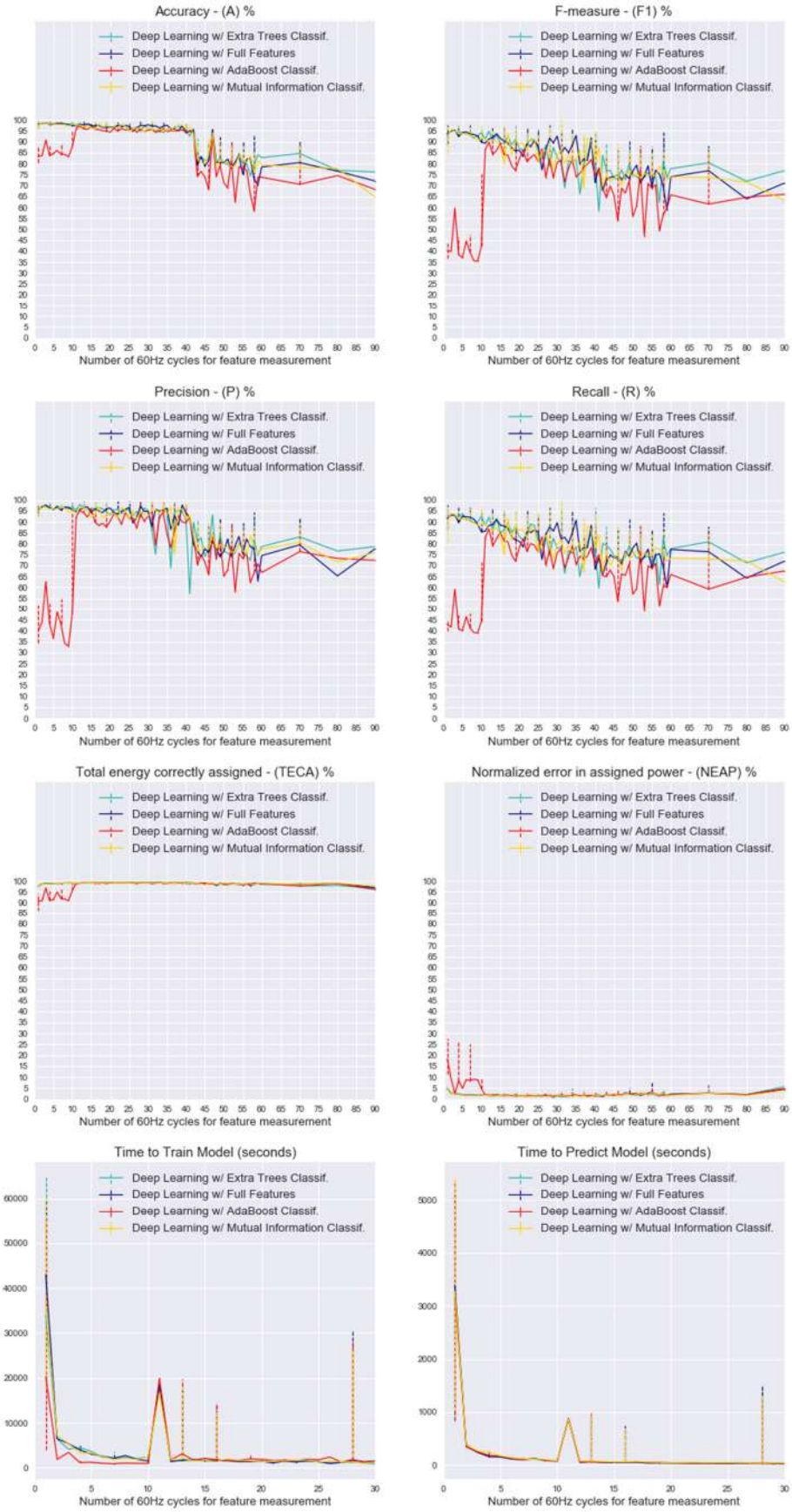

Figure 104 – Microwave oven / Feature Selection methods comparison

Hair dryer analysis

Table 54 – Hair dryer / Metrics summary

	Accuracy			F-Measure			Precision			Recall		
with Full Features	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*
Deep Learning	99	94	120	95	90	120	98	93	120	95	90	100
SVM	99	94	40	94	89	36	99	94	37	93	88	25
Logistic Regression	85	80	5	50	45	3	55	50	4	45	40	2

	TECA			NEAP			Train Time	Predict Time
with Full Features	*max%*	*-5%*	*cycle*	*min%*	*+5%*	*cycle*	*feasible time cycle*	*feasible time cycle*
Deep Learning	97	92	70	5	10	60	12	10
SVM	97	92	70	5	10	28	5	15
Logistic Regression	80	75	3	40	45	2	2	1

Table 55 – Hair dryer / Summary of Feature Selection

Deep Learning	**Feature List**																															
60 cycles	*Features*						*Fourier Harmonics*																									data
Method	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	Size	% red.
Full set	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	30	
Extra-Trees	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓		✓	25	17%
M. Inform.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓		✓		✓	26	13%
AdaBoost	✓	✓	✓	✓	✓	✓		✓						✓	✓																9	70%

Table 56 – Hair dryer / Summary of complete solution

Solution	Method		Feature Selection Method	Num. Features	F. Extraction Rate
	Deep Learning		AdaBoost	9	60 cycles

Selected Features	*Features*						*Fourier Harmonics*																							
	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	✓	✓	✓	✓	✓	✓		✓						✓	✓															

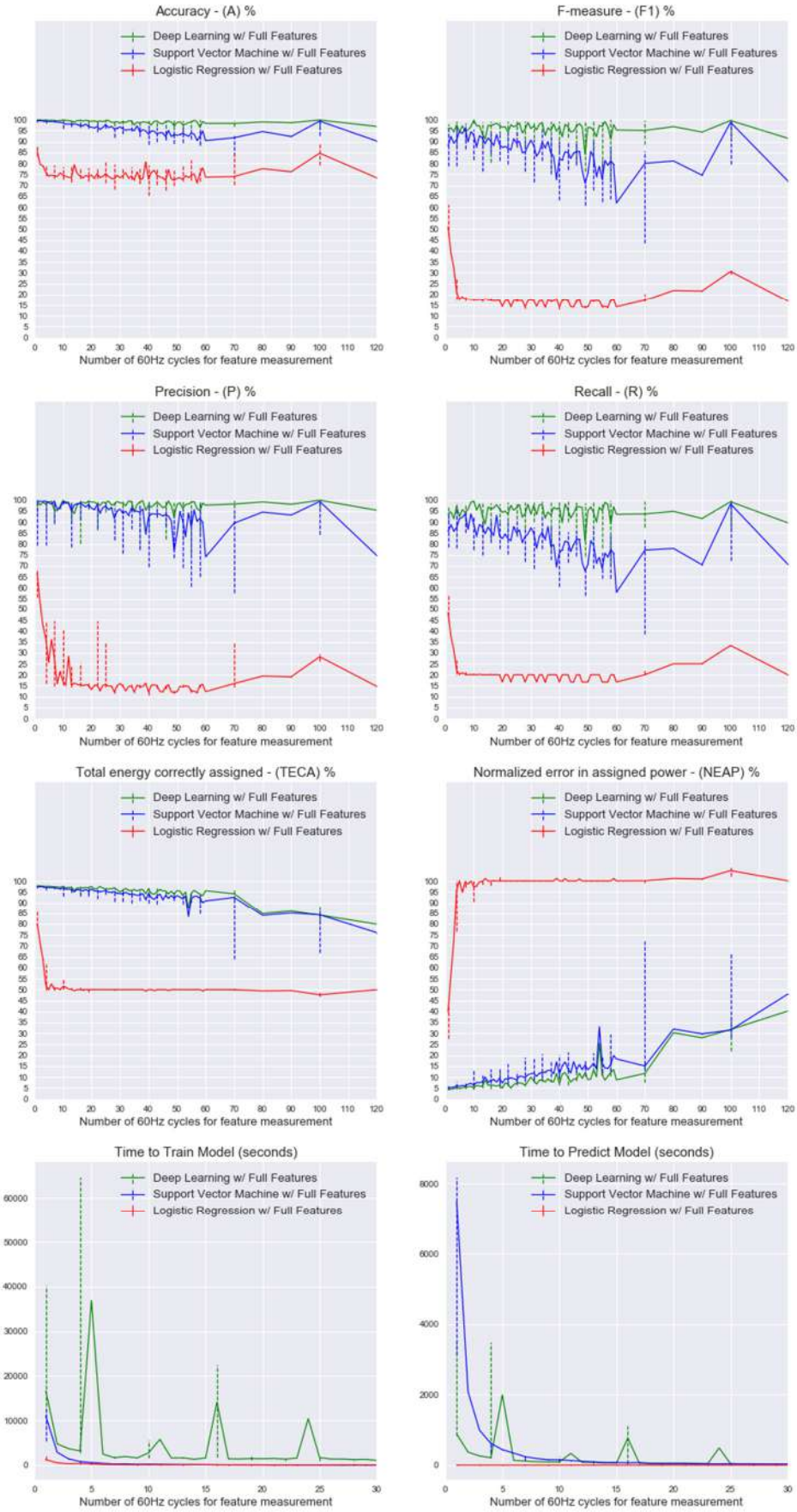

Figure 105 – Hair dryer / NILM Methods comparison with full features set

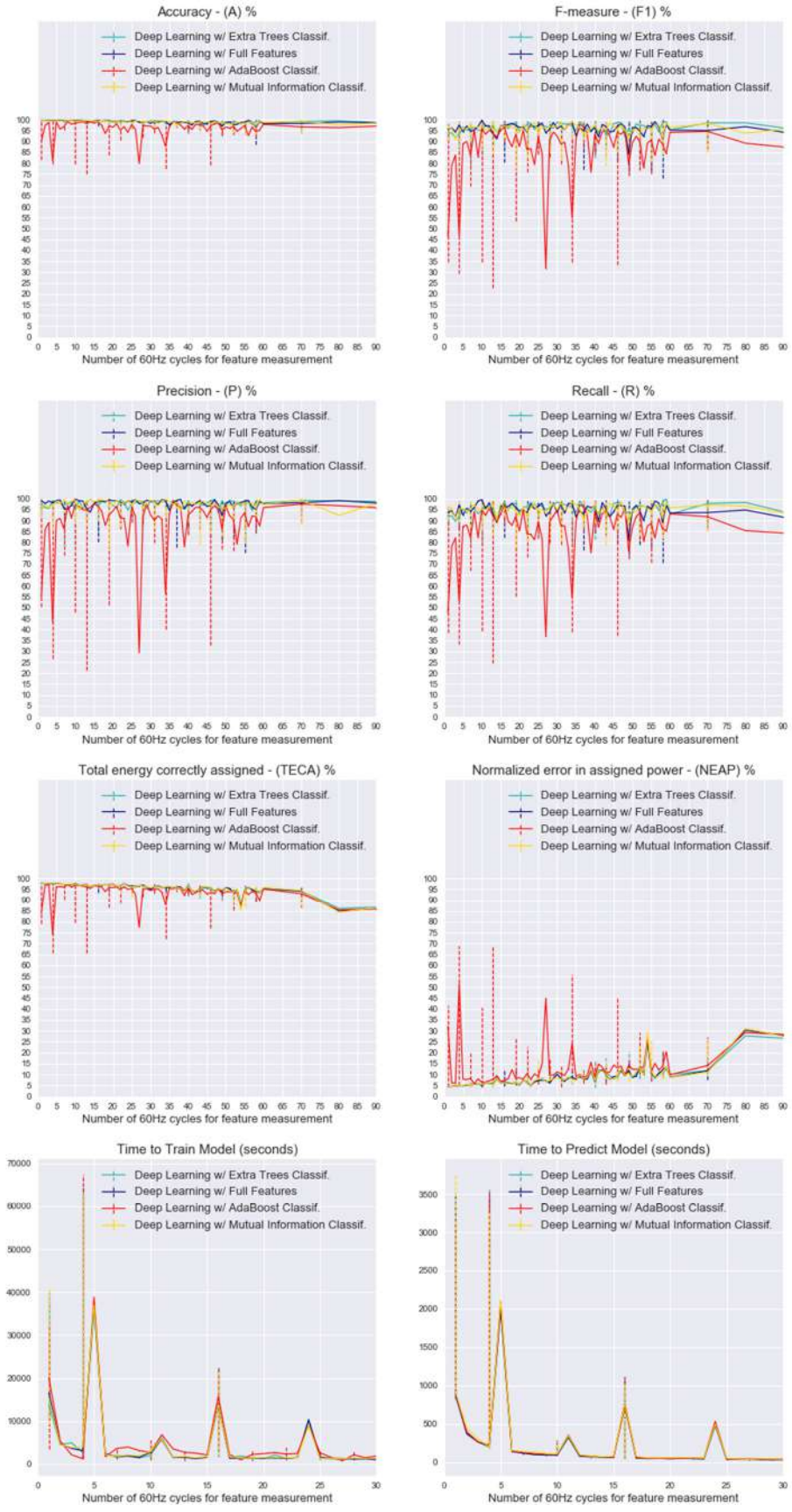

Figure 106 – Hair dryer / Feature Selection methods comparison

Halogen bulb analysis

Table 57 – Halogen bulb / Metrics summary

	Accuracy			F-Measure			Precision			Recall		
with Full Features	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*
Deep Learning	97	92	35	80	75	30	85	80	55	75	70	55
SVM	97	92	35	80	75	40	85	80	50	75	70	54
Logistic Regression	80	75	55	35	30	30	55	50	7	35	30	55

	TECA			NEAP			Train Time	Predict Time
with Full Features	*max%*	*-5%*	*cycle*	*min%*	*+5%*	*cycle*	*feasible time cycle*	*feasible time cycle*
Deep Learning	97	92	75	3	8	80	13	3
SVM	97	92	75	6	11	80	15	15
Logistic Regression	88	83	60	22	27	60	4	1

Table 58 – Halogen bulb / Summary of Feature Selection

SVM	**Feature List**																																
35 cycles	*Features*						*Fourier Harmonics*																										data
Method	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	Size	% red.
Full set	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	30	
Extra-Trees	✓	✓	✓	✓	✓	✓		✓		✓				✓						✓											10	67%
M. Inform.	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓		✓				✓		✓		✓				✓		✓			16	47%
AdaBoost		✓	✓			✓		✓																							4	87%

Table 59 – Halogen bulb / Summary of complete solution

Solution	Method	Feature Selection Method	Num. Features	F. Extraction Rate
	SVM	Extra-Trees	10	35 cycles

Selected Features	*Features*						*Fourier Harmonics*																							
	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	✓	✓	✓	✓	✓	✓		✓		✓				✓						✓										

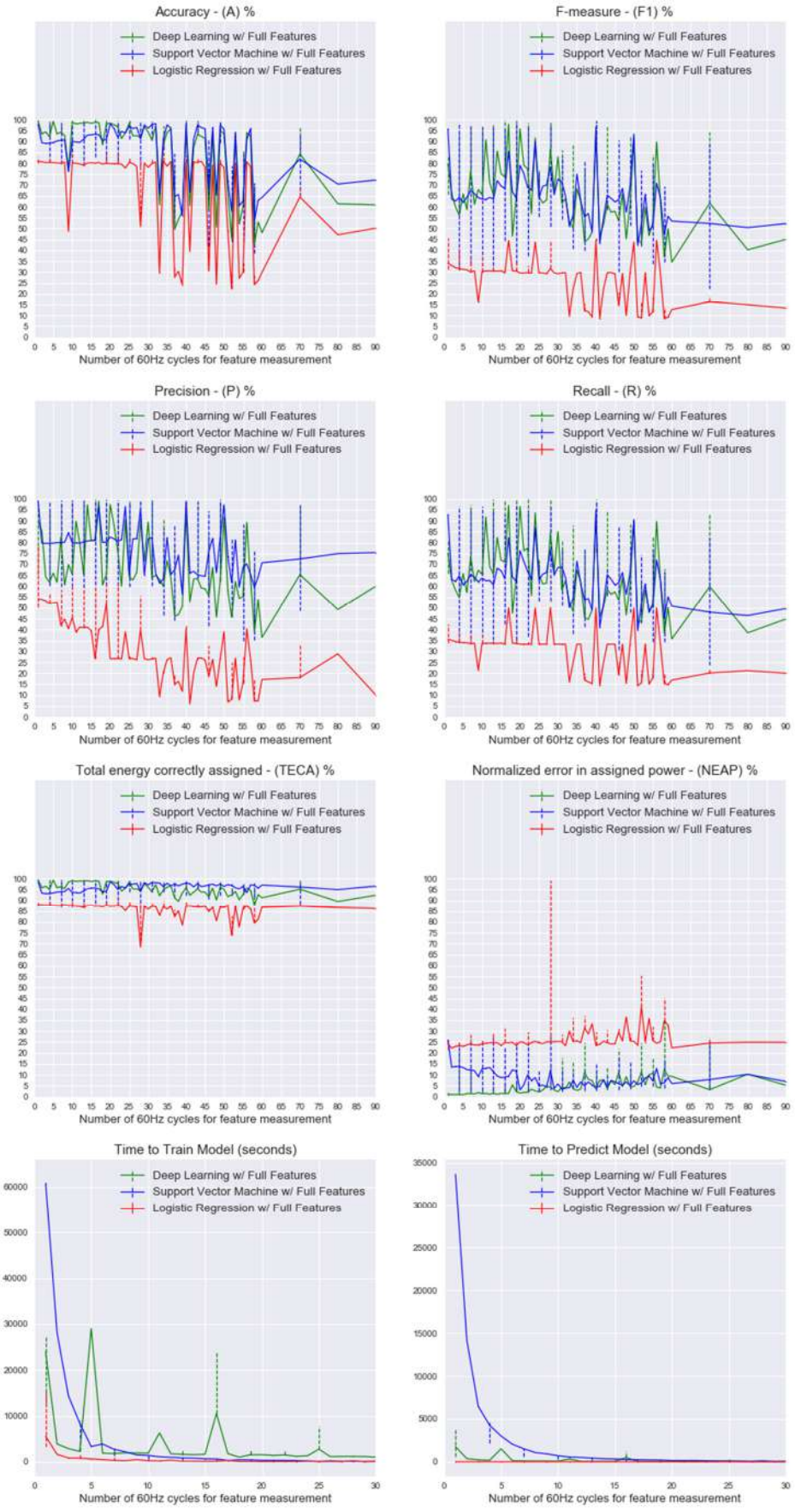

Figure 107 – Halogen bulb / NILM Methods comparison with full features set

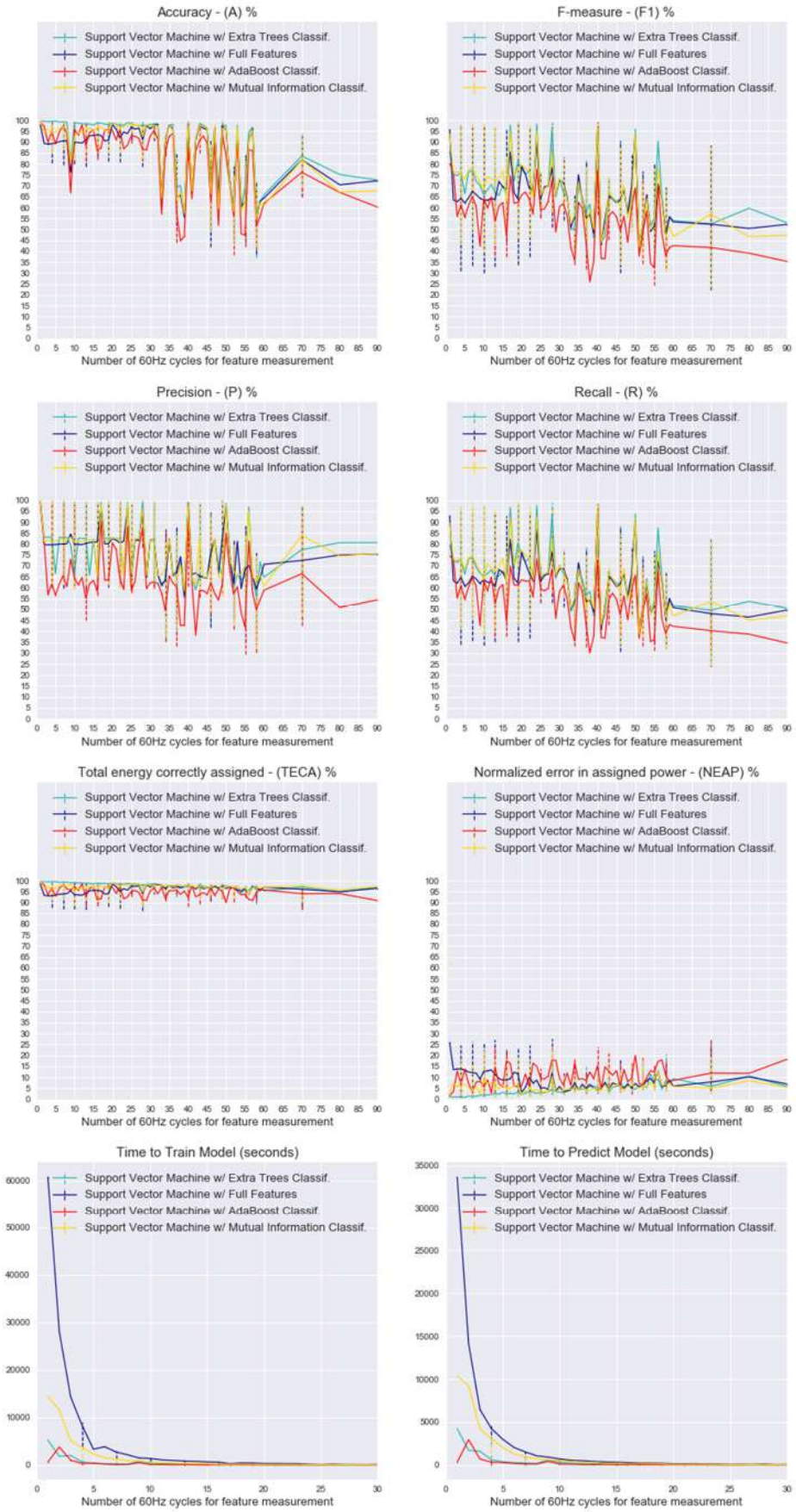

Figure 108 – Halogen bulb / SVM - Feature Selection methods comparison

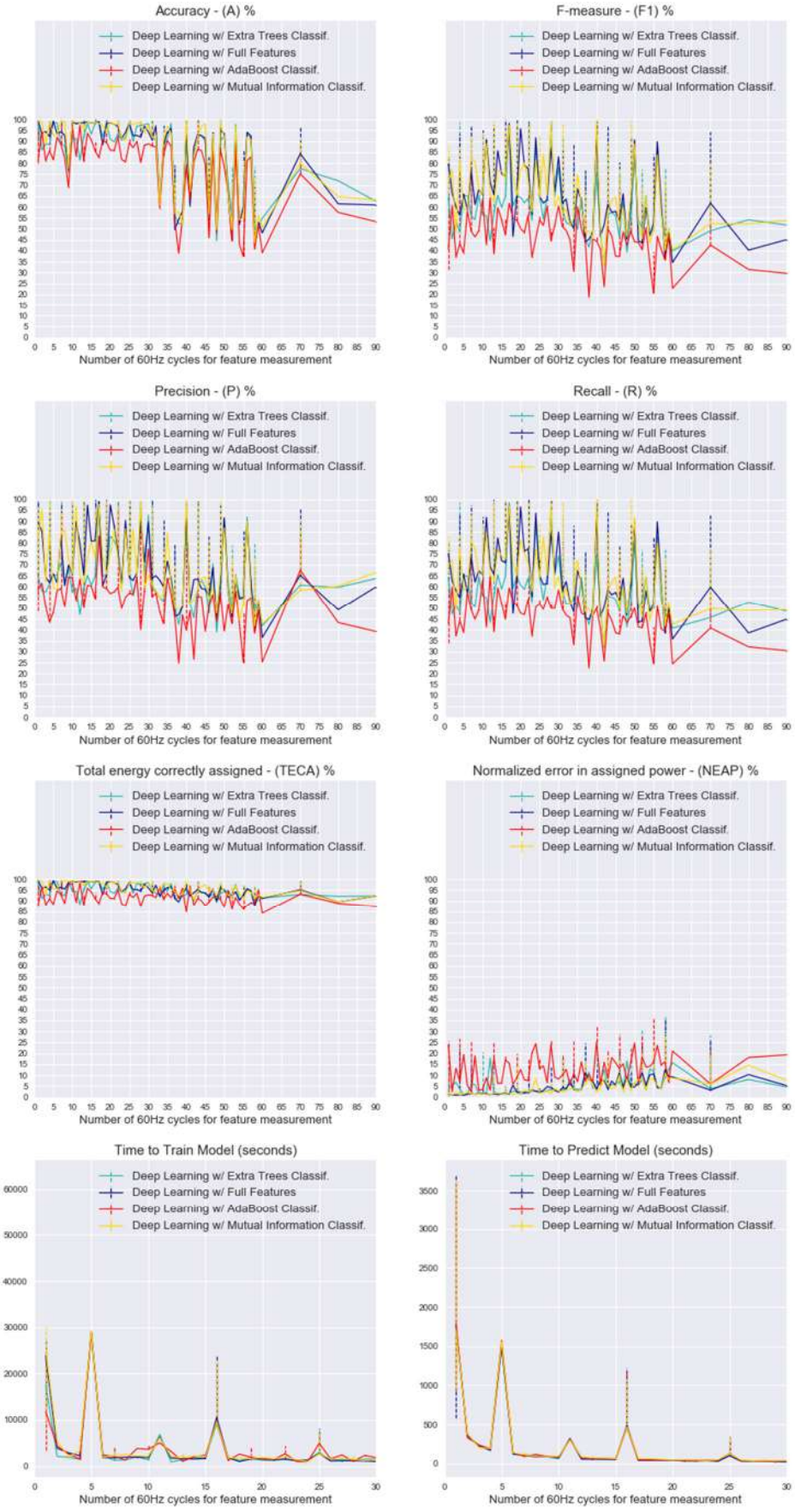

Figure 109 – Halogen bulb / Deep Learning - Feature Selection methods comparison

Fluorescent bulb analysis

Table 60 – Fluorescent bulb / Metrics summary

	Accuracy			F-Measure			Precision			Recall		
with Full Features	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*
Deep Learning	92	87	17	84	79	22	85	80	25	82	77	30
SVM	90	85	13	75	70	15	88	83	14	70	65	18
Logistic Regression	68	63	5	45	40	5	49	44	4	45	40	4

	TECA			NEAP			Train Time	Predict Time
with Full Features	*max%*	*-5%*	*cycle*	*min%*	*+5%*	*cycle*	*feasible time cycle*	*feasible time cycle*
Deep Learning	99	94	90	2	7	60	15	3
SVM	99	94	70	2	7	35	5	11
Logistic Regression	93	88	13	10	15	10	3	1

Table 61 – Fluorescent bulb / Summary of Feature Selection

Deep Learning	**Feature List**																													
17 cycles	*Features*						*Fourier Harmonics*																							data
Method	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	Size	% red.
Full set	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	30	
Extra-Trees	✓	✓	✓	✓	✓	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	18	40%
M. Inform.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	30	0%
AdaBoost	✓	✓	✓			✓		✓			✓				✓	✓				✓	✓		✓	✓							12	60%

Table 62 – Fluorescent bulb / Summary of complete solution

Solution	Method	Feature Selection Method	Num. Features	F. Extraction Rate
	Deep Learning	Extra-Trees	18	17 cycles

Selected Features	*Features*						*Fourier Harmonics*																							
	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	✓	✓	✓	✓	✓	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓

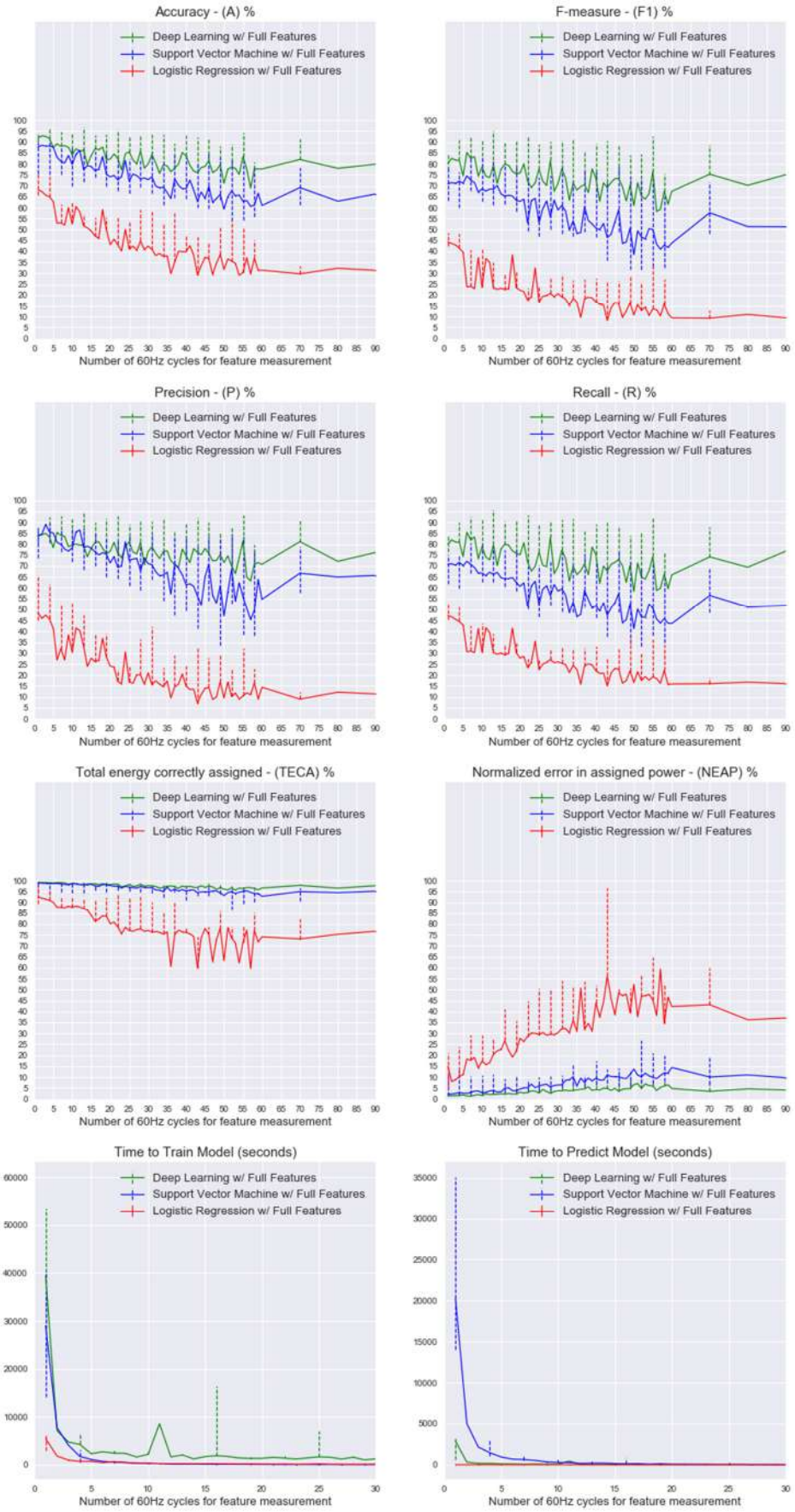

Figure 110 - Fluorescent bulb / NILM Methods comparison with full features set

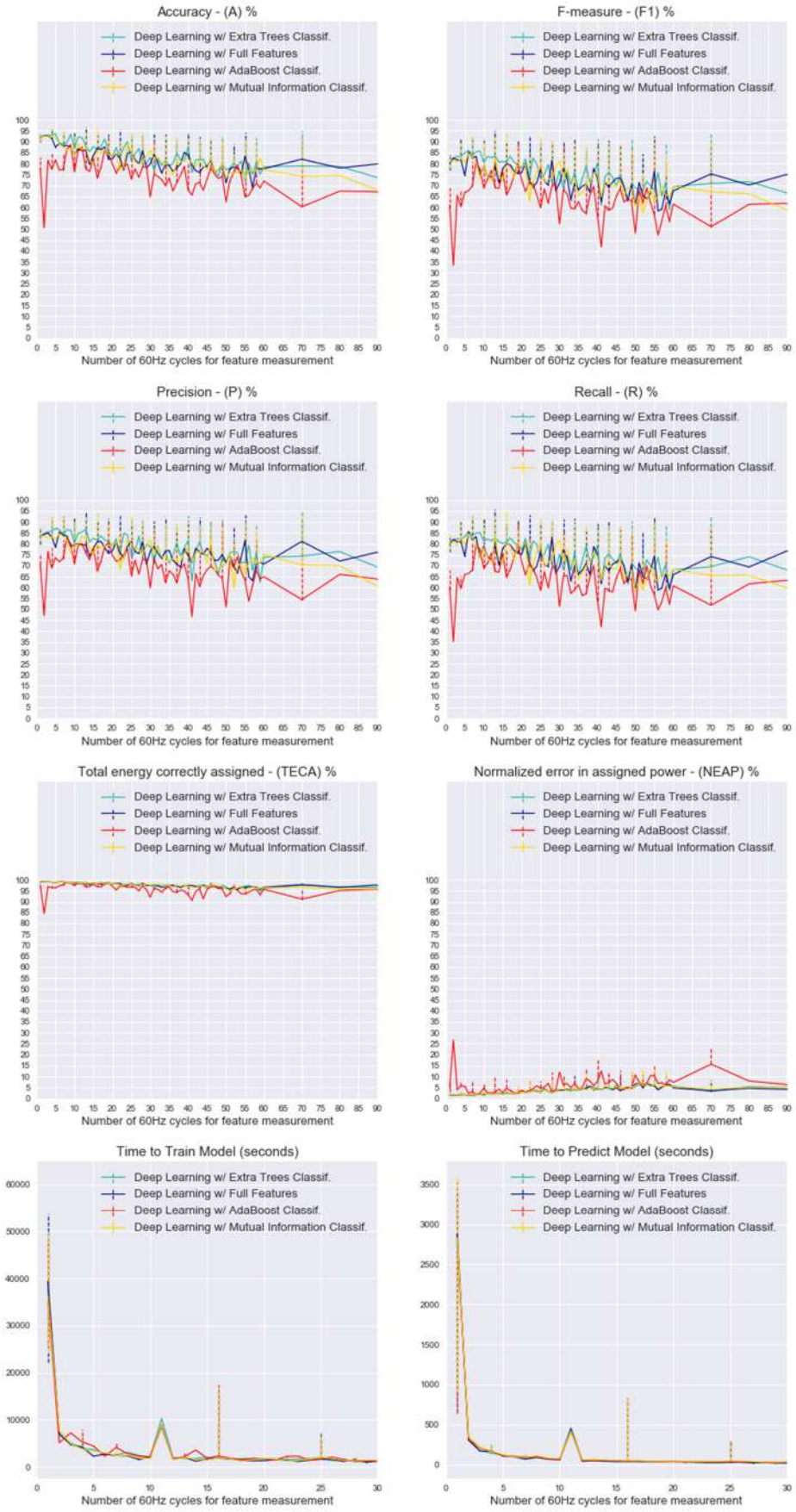

Figure 111 - Fluorescent bulb / Feature Selection methods comparison

LED bulb analysis

Table 63 - LED bulb / Metrics summary

	Accuracy			F-Measure			Precision			Recall		
with Full Features	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*
Deep Learning	99	94	52	70	65	52	70	65	60	70	65	55
SVM	96	91	46	65	60	52	65	60	60	63	58	55
Logistic Regression	85	80	5	50	45	4	55	50	13	53	48	3

	TECA			NEAP			Train Time	Predict Time
with Full Features	*max%*	*-5%*	*cycle*	*min%*	*+5%*	*cycle*	*feasible time cycle*	*feasible time cycle*
Deep Learning	98	93	56	2	7	60	2	2
SVM	97	92	75	3	8	47	5	10
Logistic Regression	89	84	6	21	26	15	2	1

Table 64 - LED bulb / Summary of Feature Selection

Deep Learning	**Feature List**																																
52 cycles	*Features*						*Fourier Harmonics*																										data
Method	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	Size	% red.
Full set	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	30	
Extra-Trees	✓	✓	✓	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓				✓		✓		✓	16	47%
M. Inform.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	28	7%
AdaBoost	✓							✓																							2	93%

Table 65 - LED bulb / Summary of complete solution

Solution	Method	Feature Selection Method	Num. Features	F. Extraction Rate
	Deep Learning	Extra-Trees	16	52 cycles

Selected Features	*Features*						*Fourier Harmonics*																							
	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	✓	✓	✓	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓				✓		✓		✓

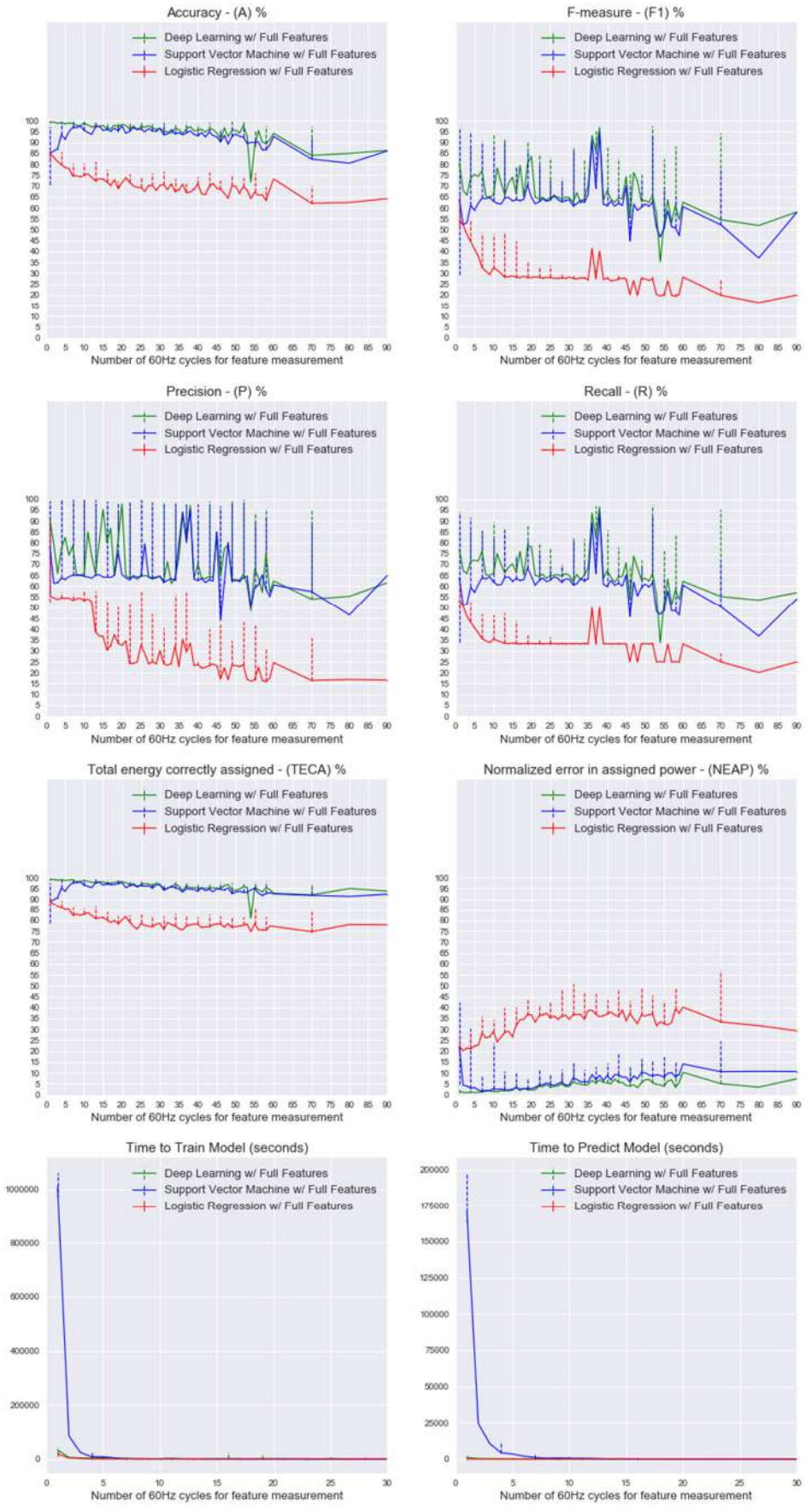

Figure 112 - LED bulb / NILM Methods comparison with full features set

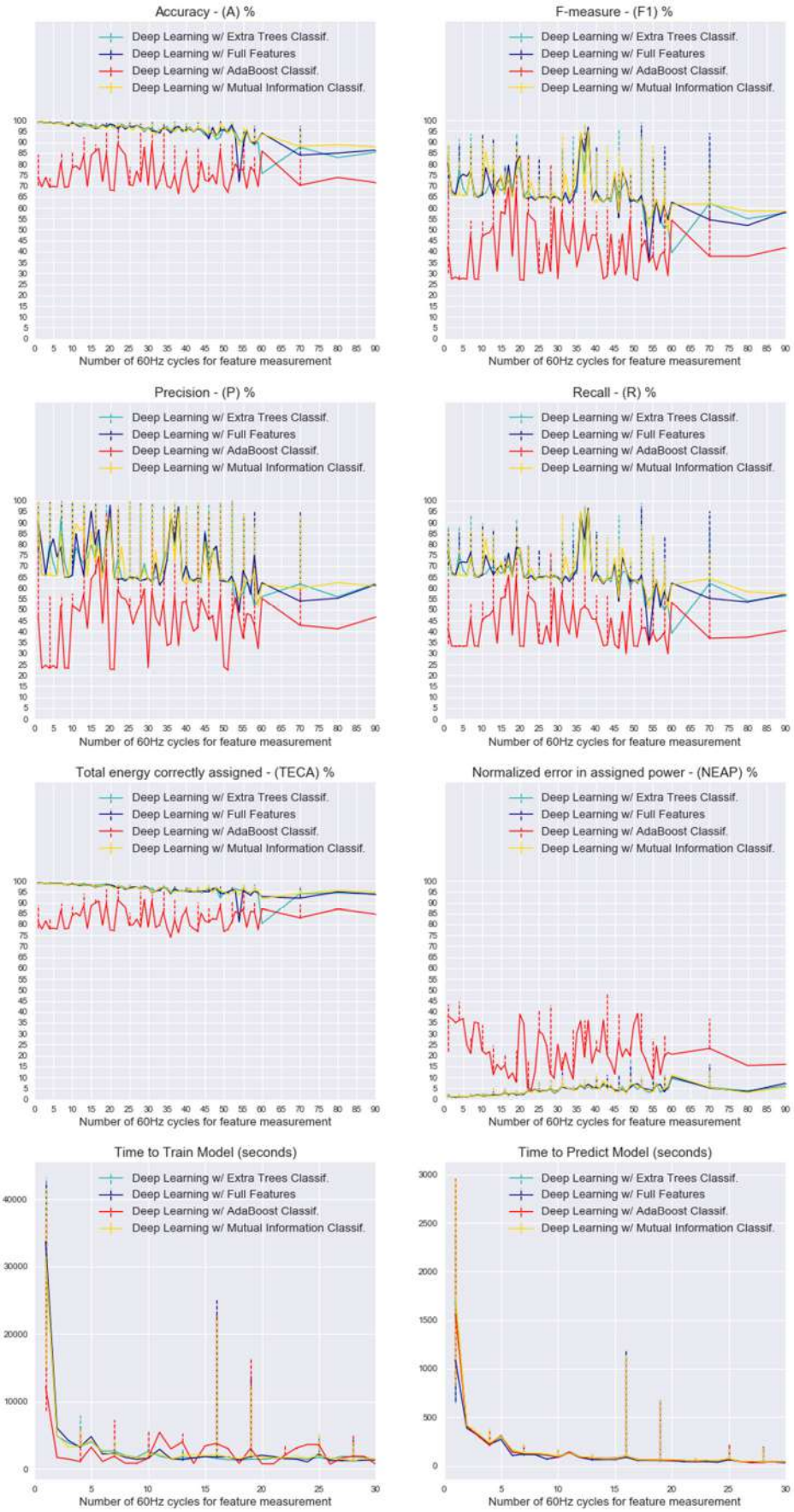

Figure 113 - LED bulb / Feature Selection methods comparison

Plasma TV analysis

Table 66 – Plasma TV / Metrics summary

	Accuracy			F-Measure			Precision			Recall		
with Full Features	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*
Deep Learning	97	92	45	97	92	23	96	91	27	97	92	45
SVM	96	91	23	96	91	22	95	90	22	96	91	22
Logistic Regression	68	63	3	53	48	3	60	55	4	55	50	2

	TECA			NEAP			Train Time	Predict Time
with Full Features	*max%*	*-5%*	*cycle*	*min%*	*+5%*	*cycle*	*feasible time cycle*	*feasible time cycle*
Deep Learning	96	91	90	8	13	90	7	2
SVM	96	91	90	8	13	90	5	7
Logistic Regression	91	86	6	17	22	4	2	1

Table 67 – Plasma TV / Summary of Feature Selection

Deep Learning	**Feature List**																													
23 cycles	*Features*						*Fourier Harmonics*																							data
Method	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	Size	% red.
Full set	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	30	
Extra-Trees	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓		✓		✓		✓		✓		✓		✓		✓	21	30%
M. Inform.	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	27	10%
AdaBoost	✓	✓	✓	✓				✓	✓				✓	✓		✓		✓			✓	✓	✓	✓	✓						15	50%

Table 68 – Plasma TV / Summary of complete solution

Solution	Method	Feature Selection Method	Num. Features	F. Extraction Rate
	Deep Learning	AdaBoost	15	23 cycles

Selected Features	*Features*						*Fourier Harmonics*																							
	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	✓	✓	✓	✓				✓	✓				✓	✓		✓		✓			✓	✓	✓	✓	✓					

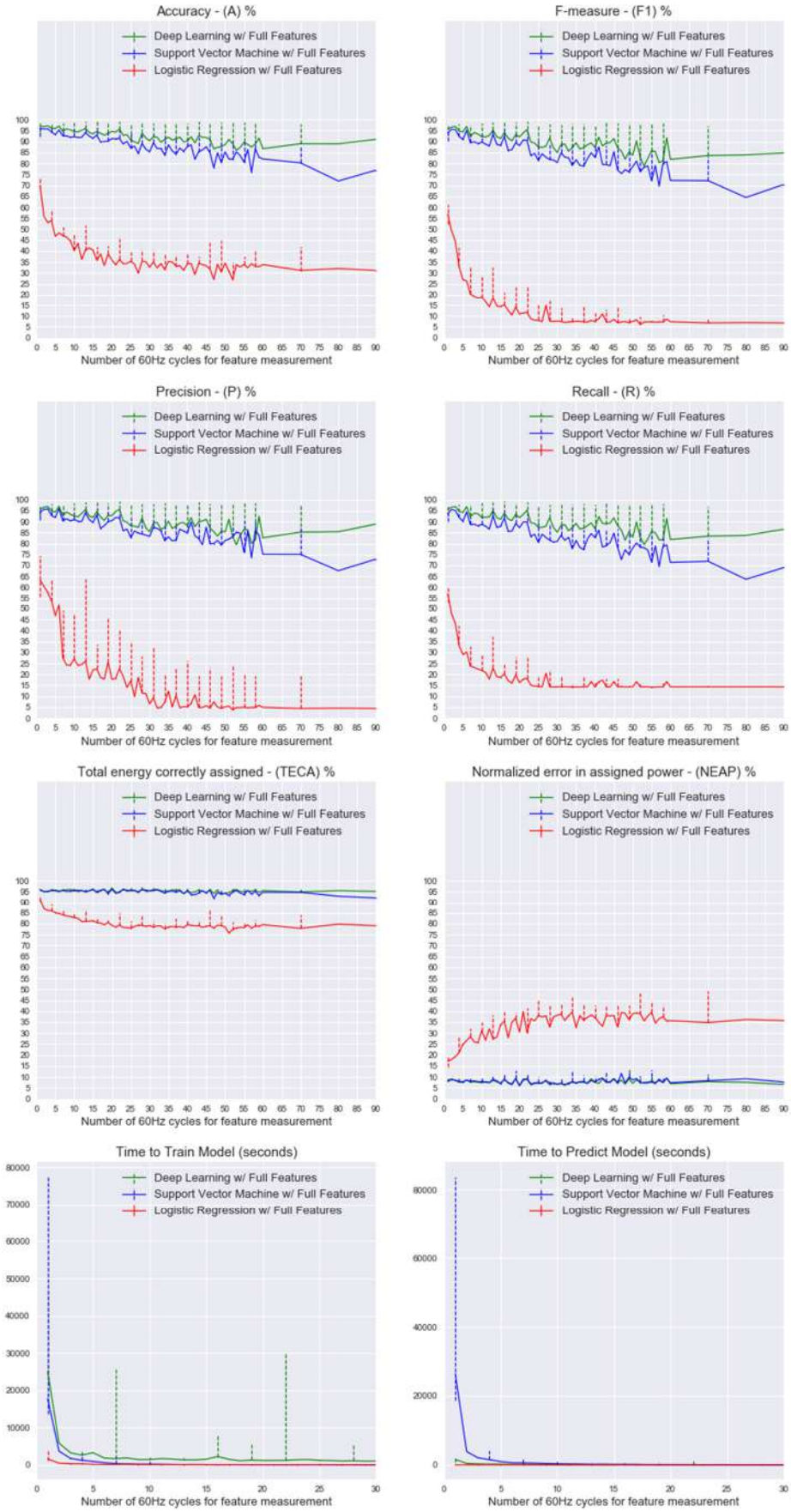

Figure 114 – Plasma TV / NILM Methods comparison with full features set

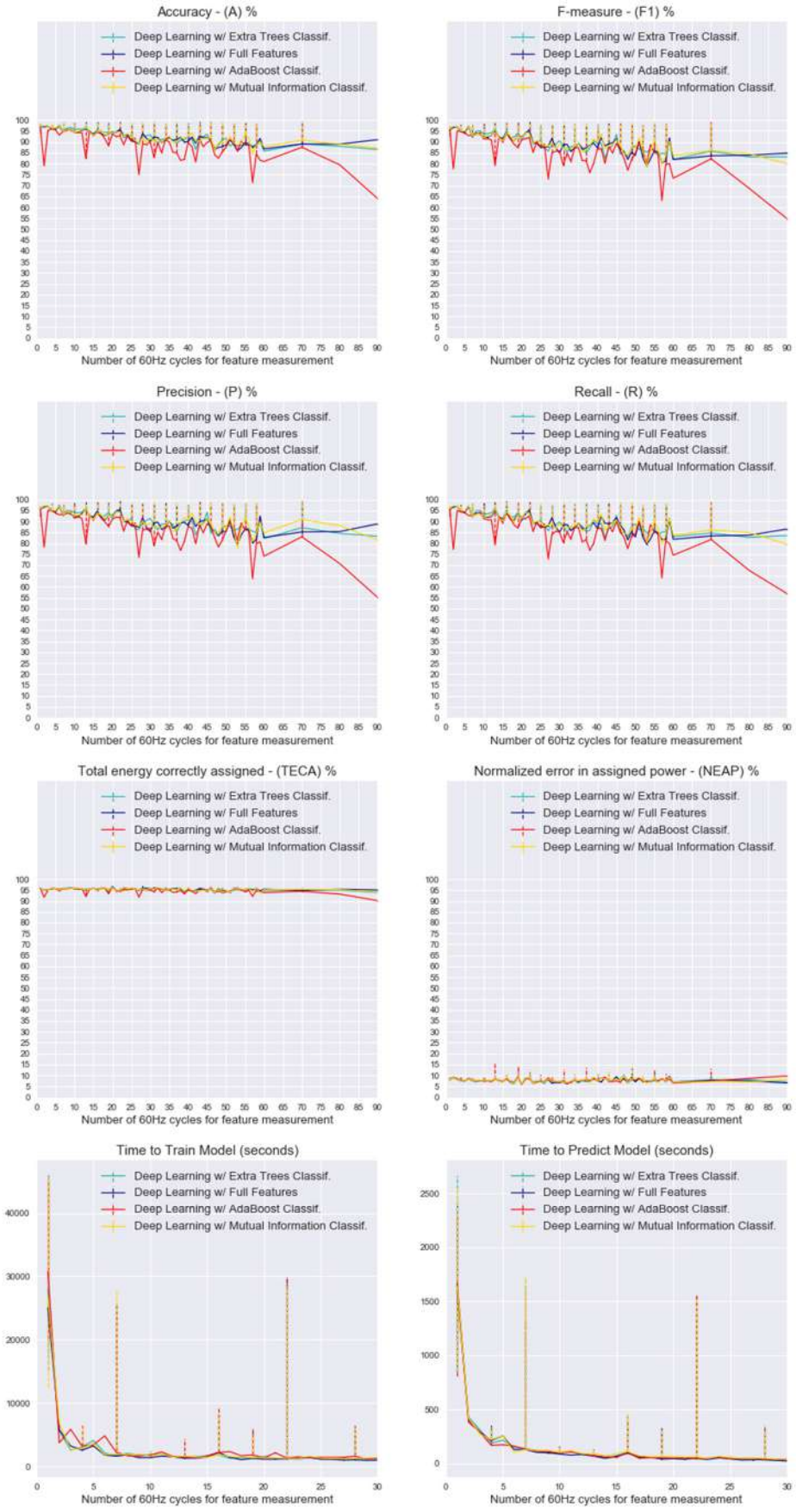

Figure 115 – Plasma TV / Feature Selection methods comparison

Ceiling fan analysis

Table 69 – Ceiling fan / Metrics summary

	Accuracy			F-Measure			Precision			Recall		
with Full Features	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*
Deep Learning	93	88	20	91	86	23	92	87	24	91	86	20
SVM	65	60	3	50	45	2	80	75	8	55	50	3
Logistic Regression	50	45	7	45	40	2	50	45	7	45	40	2

	TECA			NEAP			Train Time	Predict Time
with Full Features	*max%*	*-5%*	*cycle*	*min%*	*+5%*	*cycle*	*feasible time cycle*	*feasible time cycle*
Deep Learning	99	94	80	2	7	80	6	5
SVM	92	87	90	4	9	90	6	13
Logistic Regression	92	87	35	17	22	15	3	1

Table 70 – Ceiling fan / Summary of Feature Selection

Deep Learning	**Feature List**																													
20 cycles	*Features*						*Fourier Harmonics*																							data
Method	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	Size	% red.
Full set	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	30	
Extra-Trees	✓	✓	✓	✓	✓	✓		✓		✓		✓		✓						✓											11	63%
M. Inform.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓						✓											13	57%
AdaBoost	✓	✓	✓	✓						✓											✓			✓				✓			8	73%

Table 71 – Ceiling fan / Summary of complete solution

Solution	Method	Feature Selection Method	Num. Features	F. Extraction Rate
	Deep Learning	Extra-Trees	11	20 cycles

Selected Features	*Features*						*Fourier Harmonics*																							
	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	✓	✓	✓	✓	✓	✓		✓		✓		✓		✓						✓										

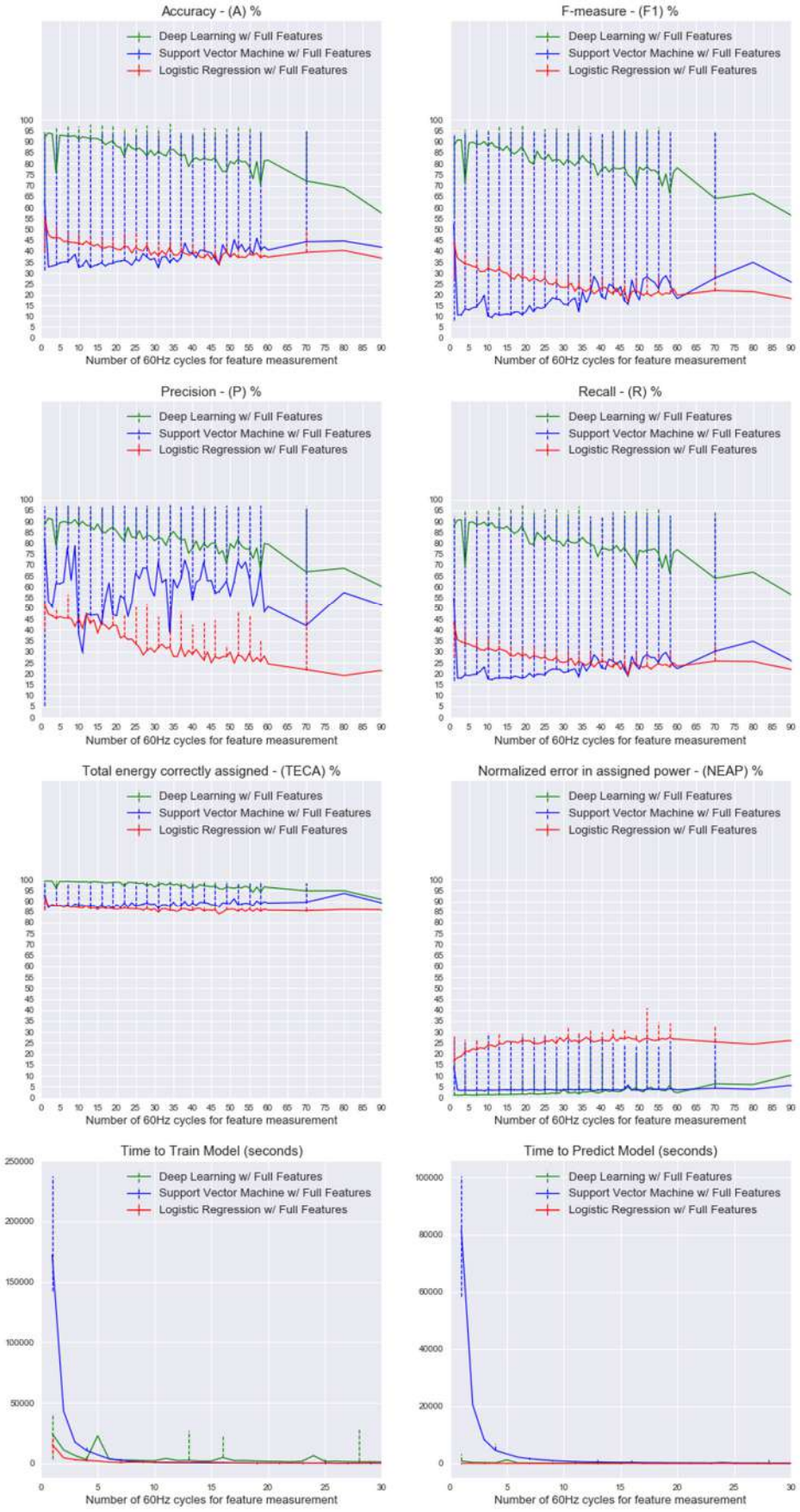

Figure 116 – Ceiling fan / NILM Methods comparison with full features set

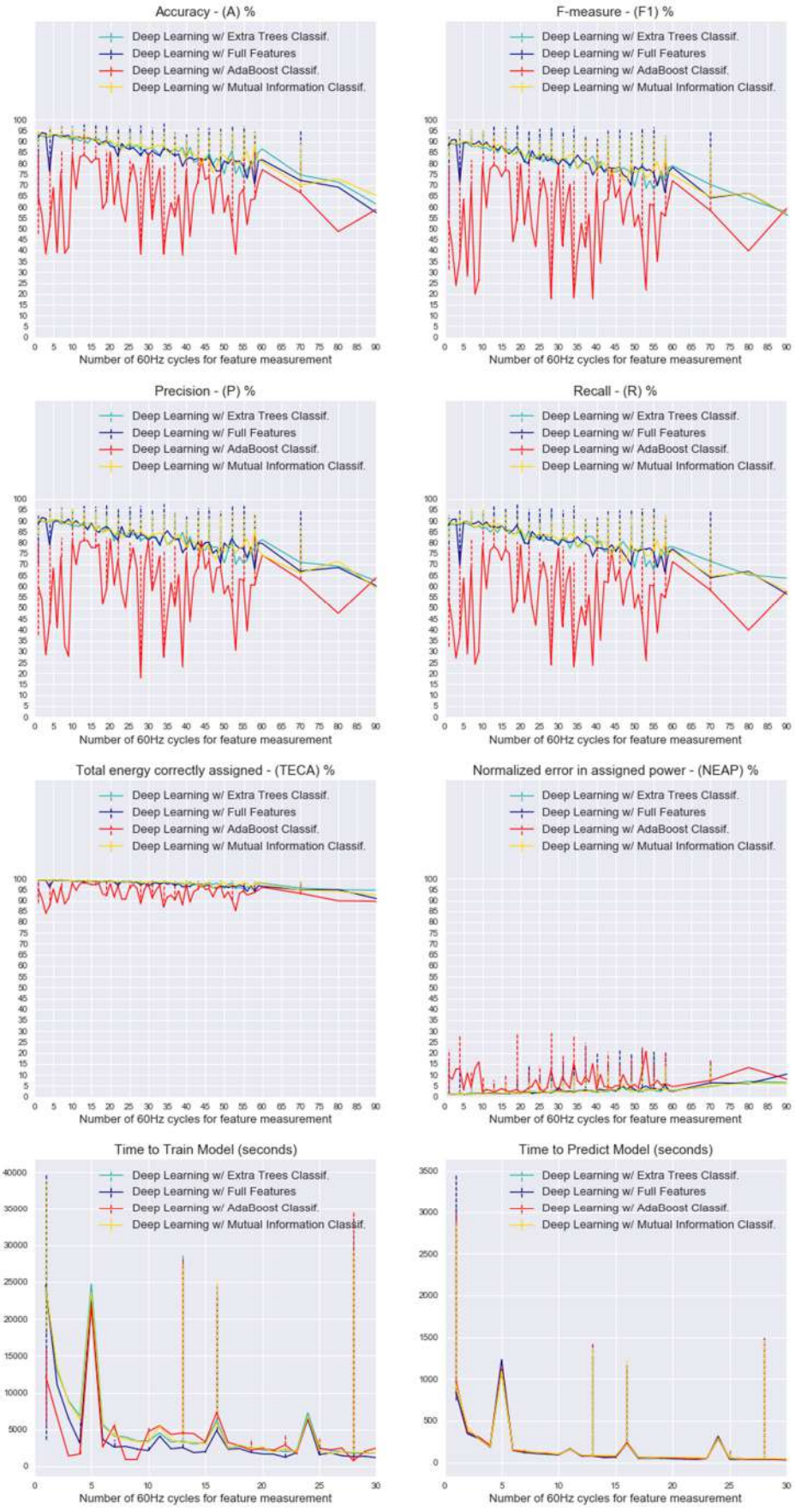

Figure 117 – Ceiling fan / Feature Selection methods comparison

Desktop PC analysis

Table 72 – Desktop PC / Metrics summary

	Accuracy			F-Measure			Precision			Recall		
with Full Features	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*
Deep Learning	95	90	60	92	87	53	92	87	55	92	87	55
SVM	93	88	40	90	85	20	90	85	41	90	85	14
Logistic Regression	70	65	5	50	45	2	65	60	2	55	50	2

	TECA			NEAP			Train Time	Predict Time
with Full Features	*max%*	*-5%*	*cycle*	*min%*	*+5%*	*cycle*	*feasible time cycle*	*feasible time cycle*
Deep Learning	93	88	90	14	19	90	6	4
SVM	93	88	90	14	19	90	3	10
Logistic Regression	94	89	6	13	18	10	2	1

Table 73 – Desktop PC / Summary of Feature Selection

Deep Learning	**Feature List**																													
53 cycles	*Features*						*Fourier Harmonics*																							data
Method	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	Size	% red.
Full set	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	30	
Extra-Trees	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	28	7%
M. Inform.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	30	0%
AdaBoost	✓	✓	✓															✓													4	87%

Table 74 – Desktop PC / Summary of complete solution

Solution	Method	Feature Selection Method	Num. Features	F. Extraction Rate
	Deep Learning	Extra-Trees	28	53 cycles

Selected Features	*Features*						*Fourier Harmonics*																							
	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

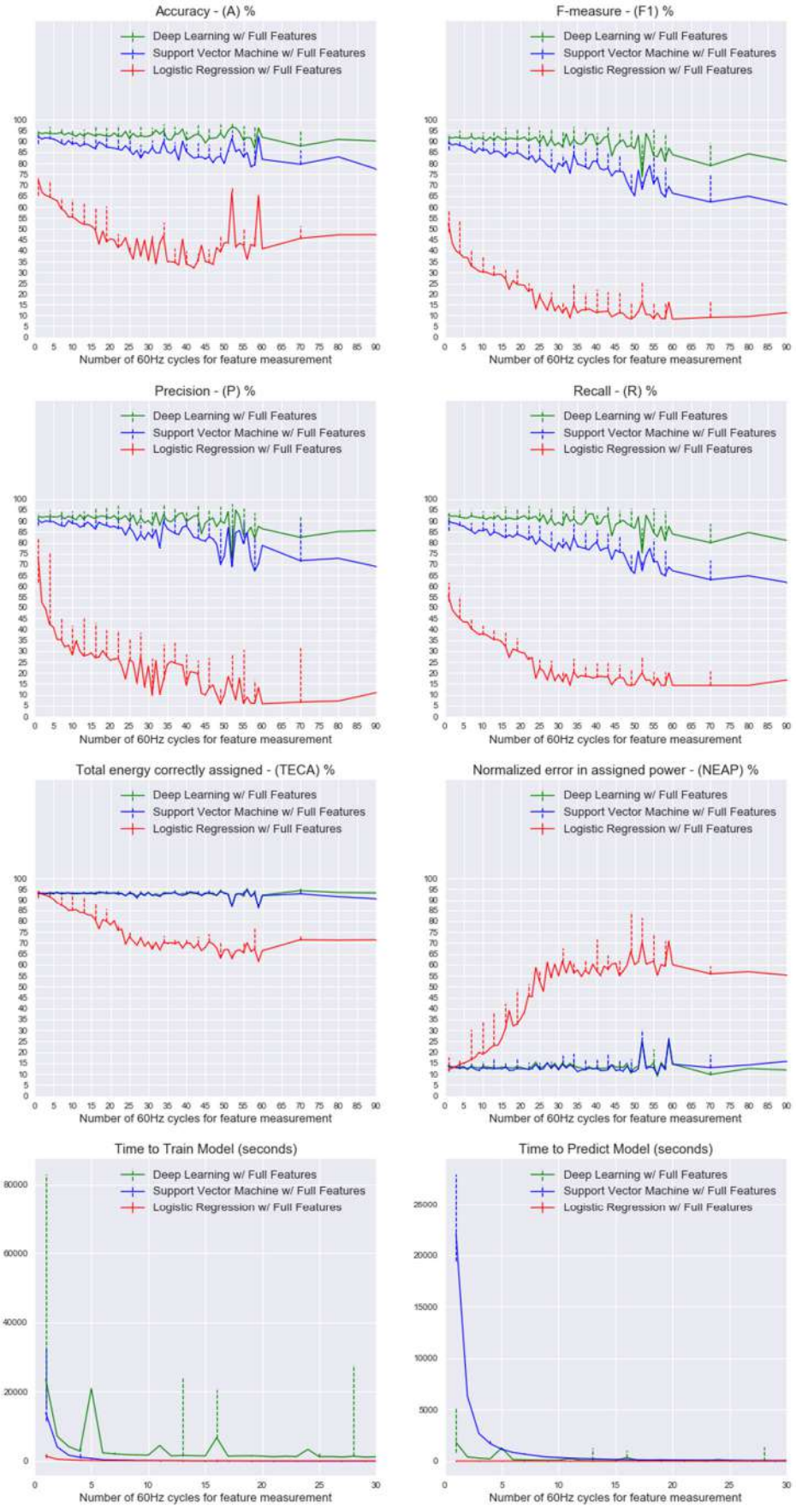

Figure 118 – Desktop PC / Feature Selection methods comparison

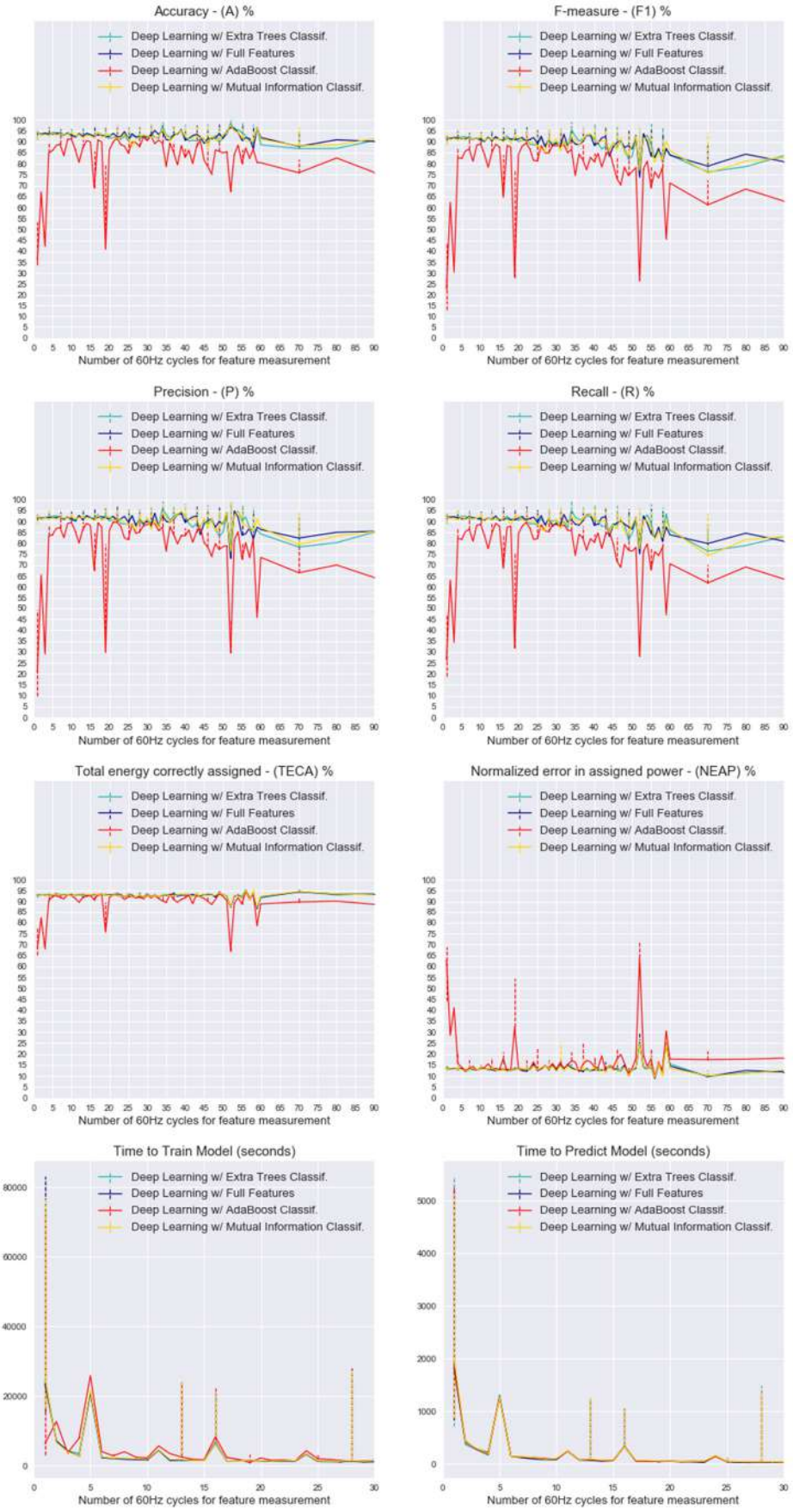

Figure 119 – Desktop PC / Feature Selection methods comparison

Clothes iron analysis

Table 75 – Clothes iron / Metrics summary

	Accuracy			F-Measure			Precision			Recall		
with Full Features	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*	*max%*	*-5%*	*cycle*
Deep Learning	100	95	90	99	94	90	99	94	90	99	94	90
SVM	100	95	90	99	94	90	99	94	90	99	94	80
Logistic Regression	98	93	90	64	59	7	80	75	23	63	58	5

	TECA			NEAP			Train Time	Predict Time
with Full Features	*max%*	*-5%*	*cycle*	*min%*	*+5%*	*cycle*	*feasible time cycle*	*feasible time cycle*
Deep Learning	98	93	90	7	12	27	5	4
SVM	98	93	90	7	12	15	4	4
Logistic Regression	95	90	5	20	25	90	2	1

Table 76 – Clothes iron / Summary of Feature Selection

Deep Learning	**Feature List**																															
27 cycles	*Features*						*Fourier Harmonics*																									data
Method	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	Size	% red.
Full set	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	30	
Extra-Trees	✓	✓	✓	✓	✓	✓		✓		✓		✓		✓		✓				✓		✓				✓					14	53%
M. Inform.	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓			18	40%
AdaBoost		✓	✓		✓																										3	90%

Table 77 – Clothes iron / Summary of complete solution

Solution	Method	Feature Selection Method	Num. Features	F. Extraction Rate
	Deep Learning	Extra-Trees	14	27 cycles

Selected Features	*Features*						*Fourier Harmonics*																							
	P	I	S	Q	CF	F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	✓	✓	✓	✓	✓	✓		✓		✓		✓		✓		✓				✓		✓				✓				

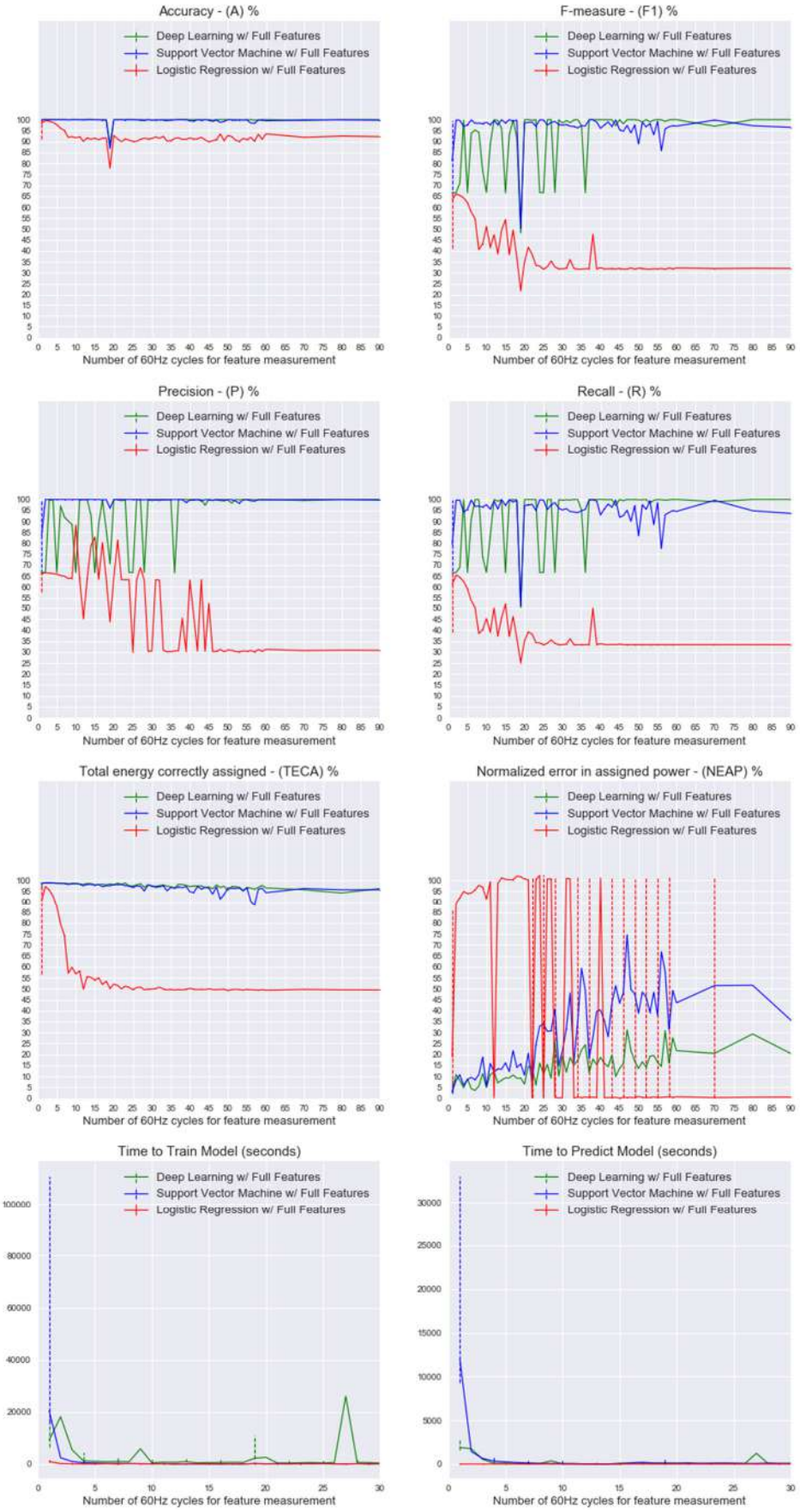

Figure 120 – Clothes iron / Feature Selection methods comparison

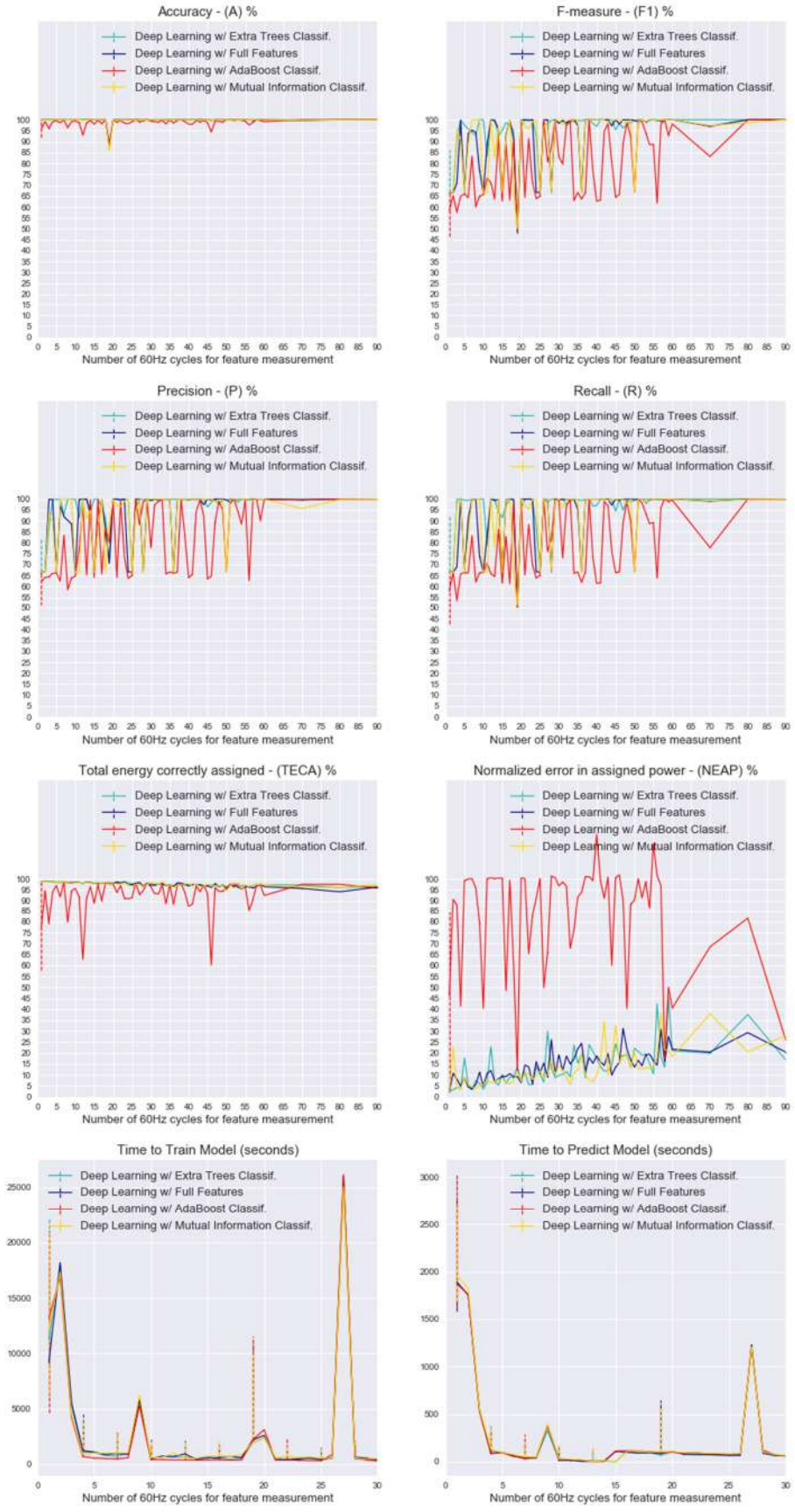

Figure 121 – Clothes iron / Feature Selection methods comparison

Zoom of Feature selection comparison

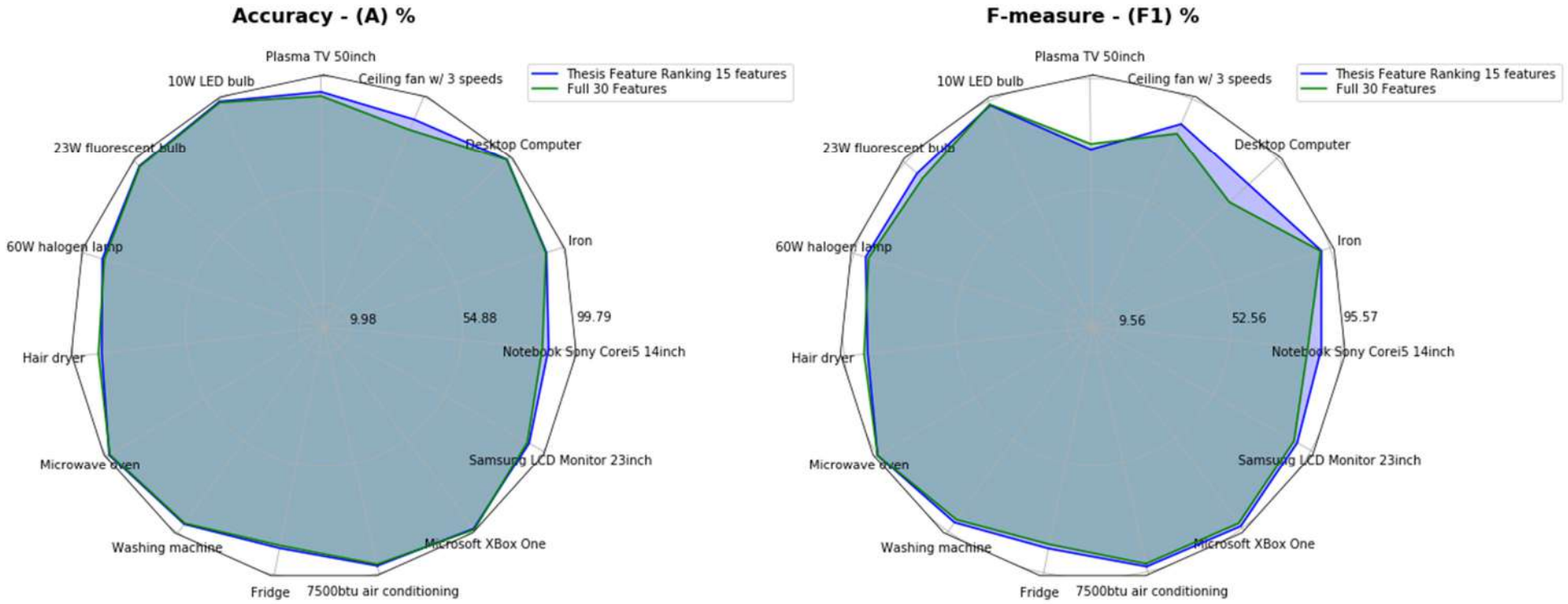

Figure 122 – Zoom of Accuracy and F-Measure from Figure 84

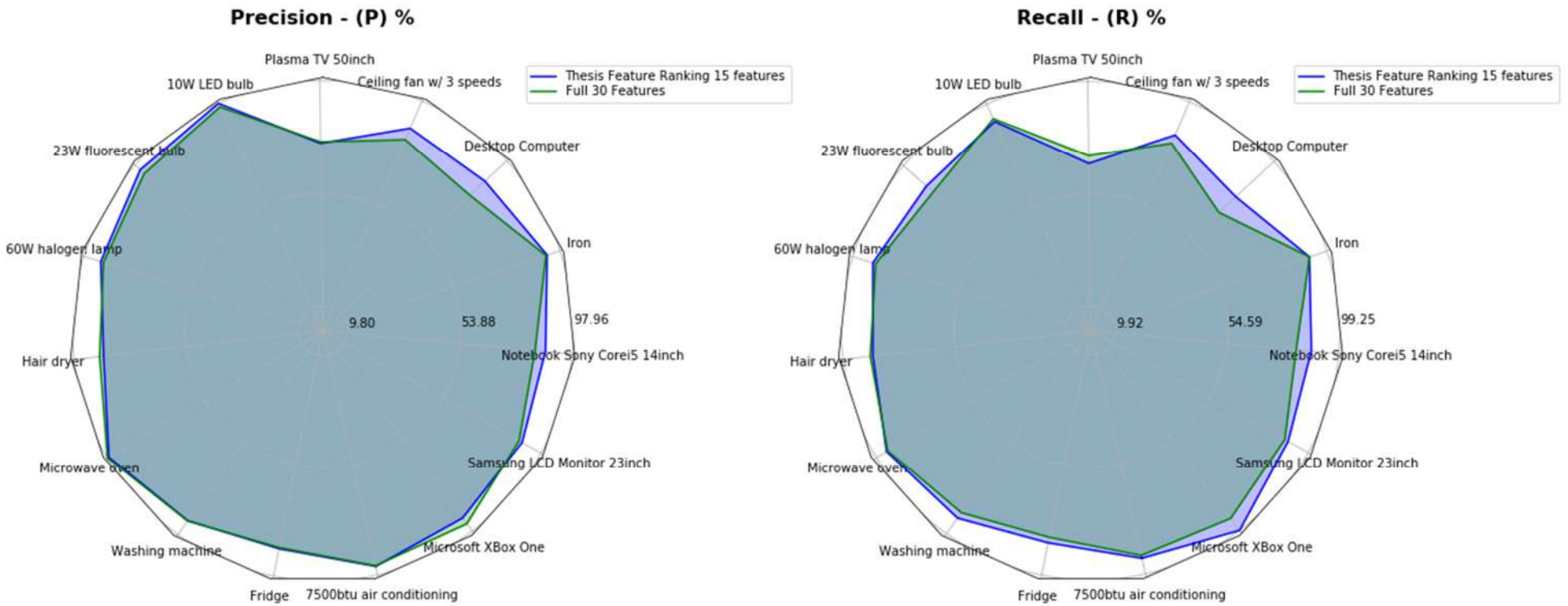

Figure 123 – Zoom of Precision and Recall from Figure 84

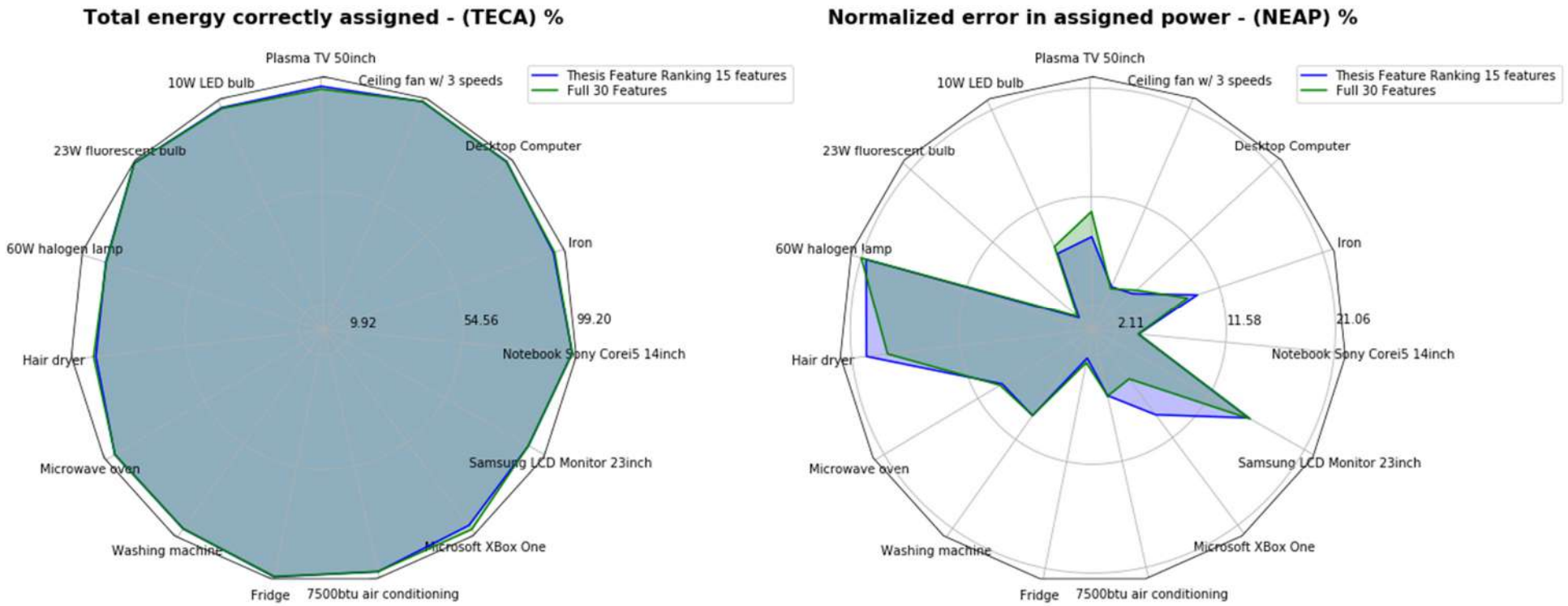

Figure 124 – Zoom of TECA and NEAP from Figure 84

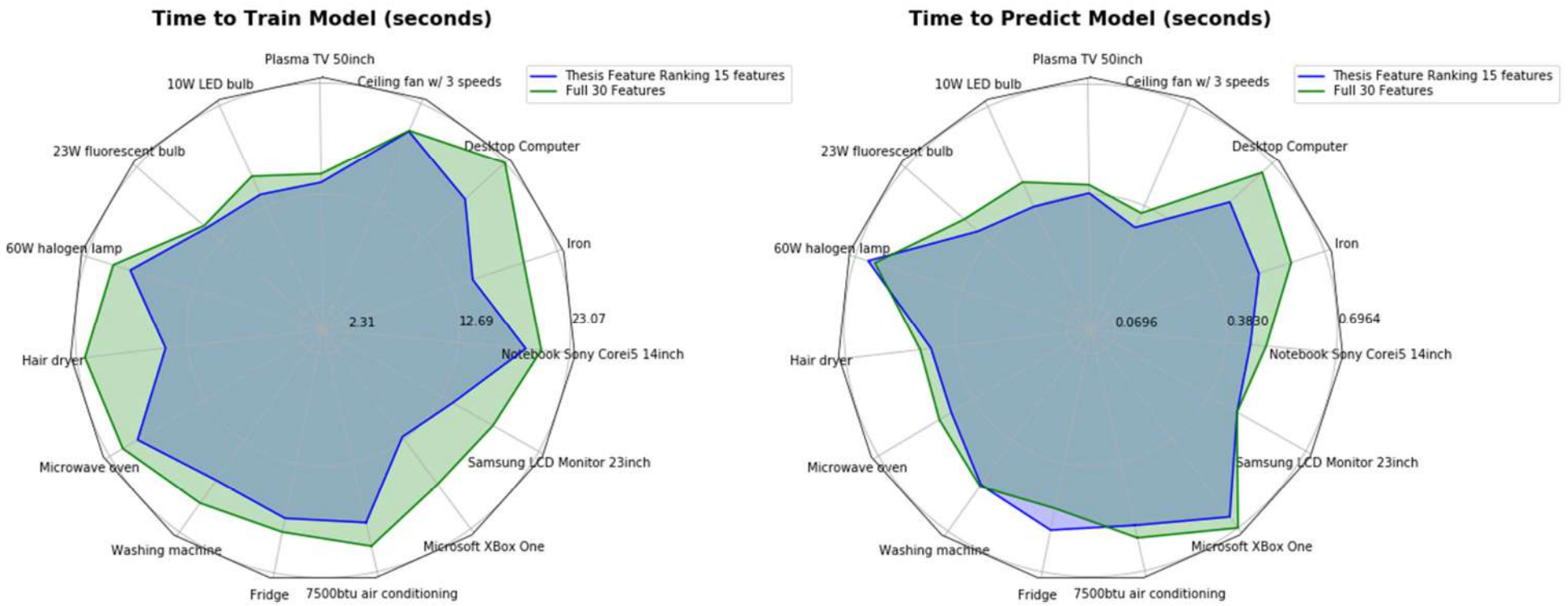

Figure 125 – Zoom of *Time to Train* and *Time to Predict* from Figure 84

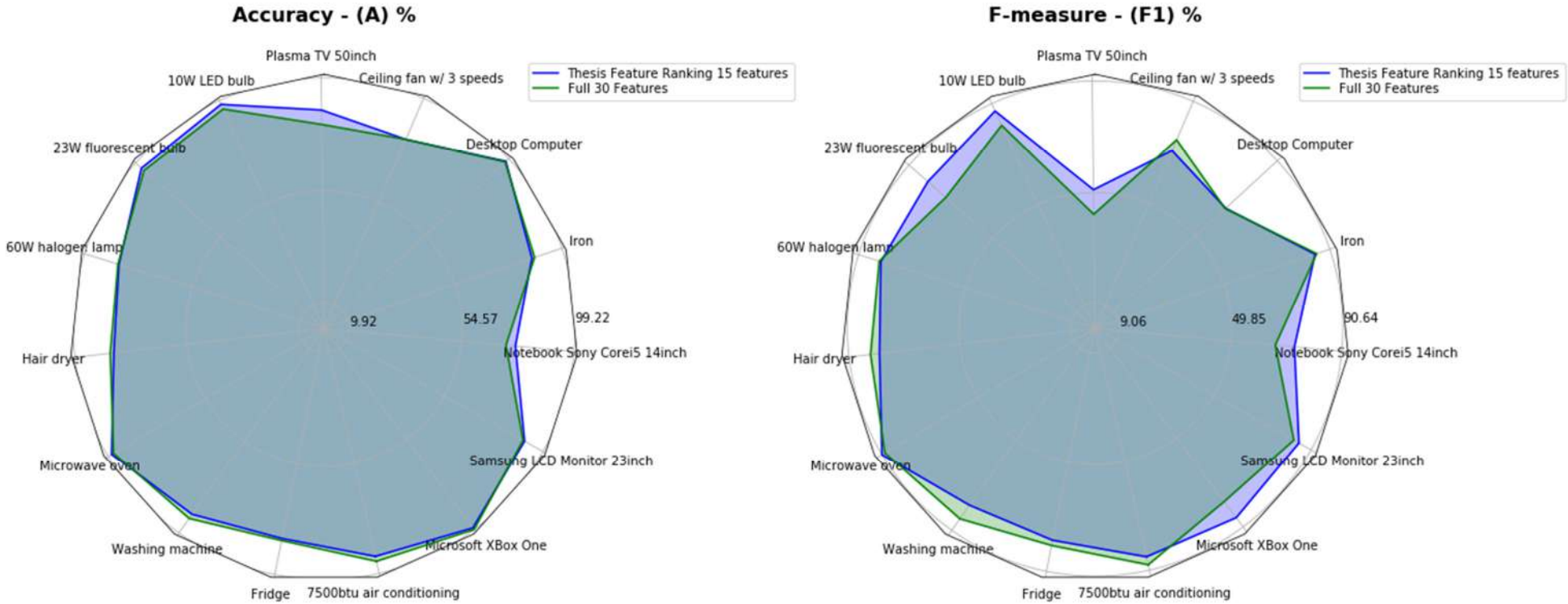

Figure 126 – Zoom of Accuracy and F-Measure from Figure 85

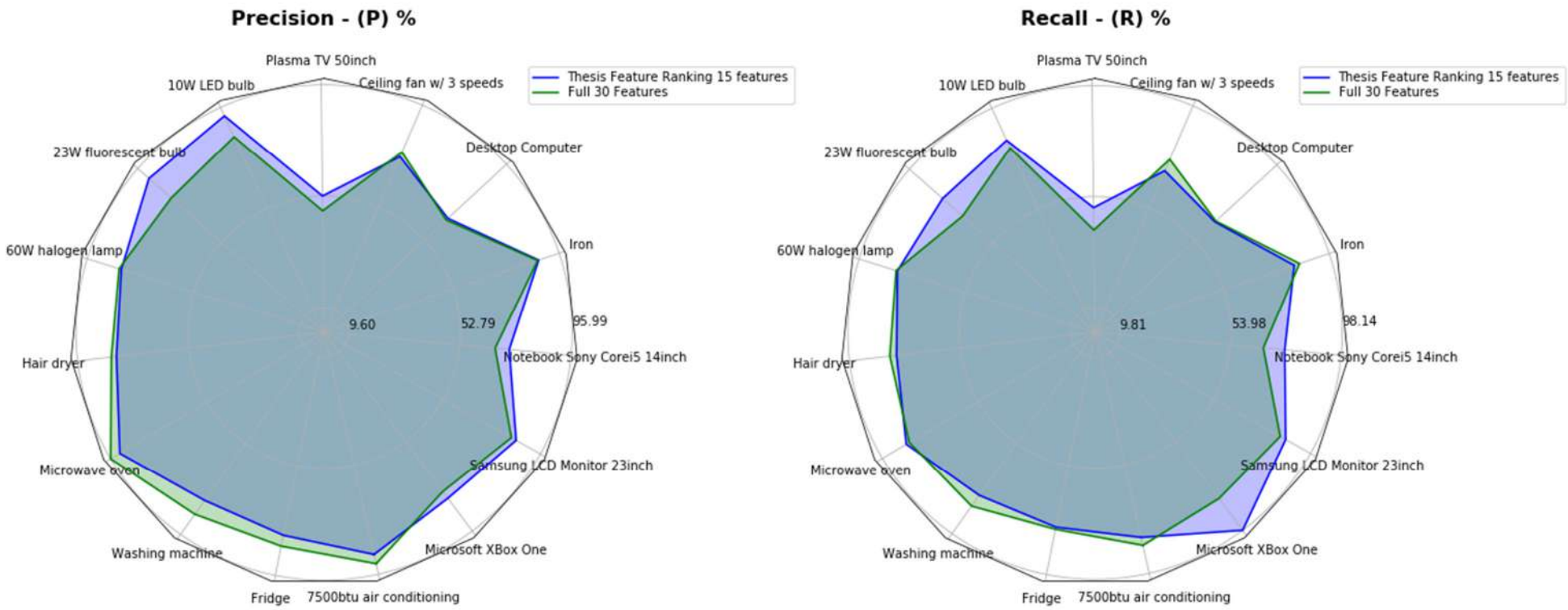

Figure 127 – Zoom of Precision and Recall from Figure 85

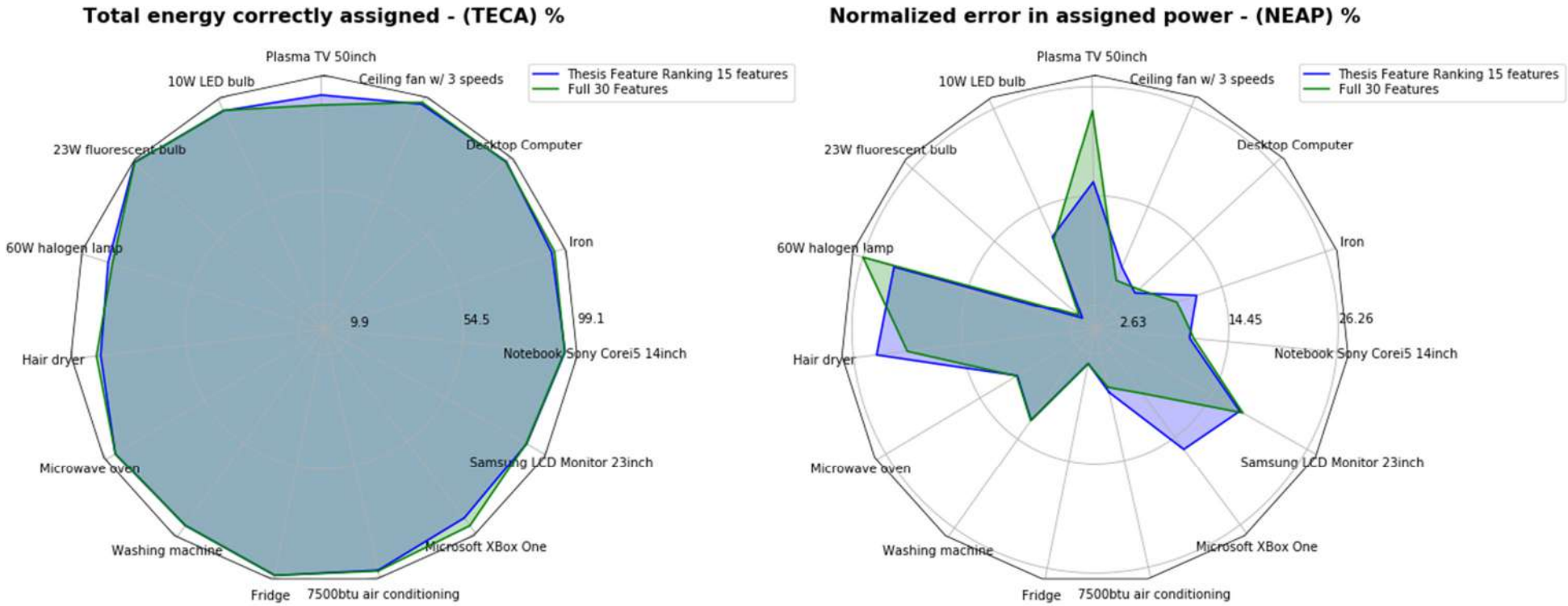

Figure 128 – Zoom of TECA and NEAP from Figure 85

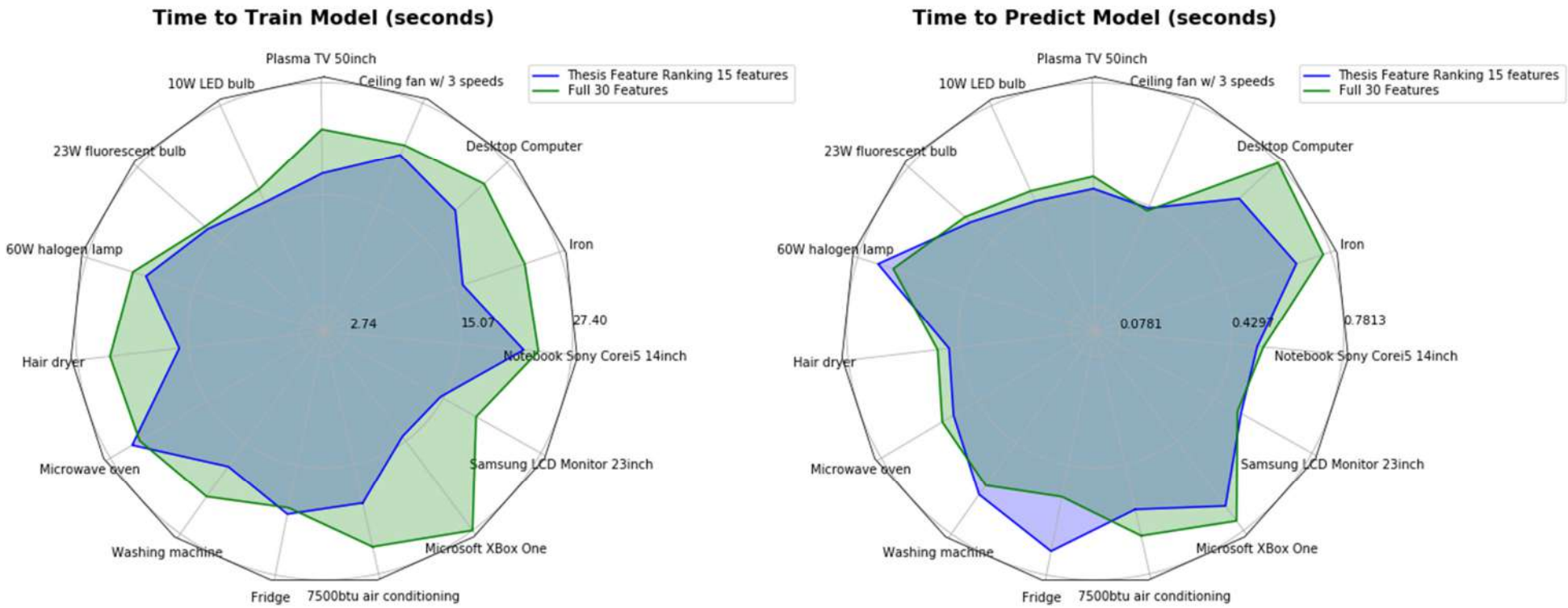

Figure 129 – Zoom of *Time to Train* and *Time to Predict* from Figure 85