2021-06

LOAD MONITORING AND ACTIVITY RECOGNITION IN SMART HOMES

FRANCO TROYA, PATRICIA

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTMENT OF ELECTRONICS
VALPARAÍSO - CHILE

IOT BASED APPROACH FOR LOAD MONITORING AND ACTIVITY RECOGNITION IN SMART HOMES

Submitted by

PATRICIA FRANCO TROYA

In partial fulfillment of the requirements for the award of the degree of
MASTER OF SCIENCE
IN ELECTRONICS ENGINEERING

THESIS SUPERVISOR	:	PhD. MOHAMED ABDELHAMID
THESIS CO-SUPERVISOR	:	PhD. JOSÉ M. MARTÍNEZ
EVALUATION COMMITTEE	:	PhD. WERNER CREIXELL
		PhD. ALEJANDRA BEGHELLI

JUNE 2021

In loving memory of Rafaela Lee Silot and Mirtha Bello Fernández . . .

ACKNOWLEDGEMENTS

Throughout my master degree studies, I have received a great deal of support and assistance from my professor and Thesis Advisor PhD. Mohamed Abdelhamid. I would really like to thank him for his insightful feedback that pushed me to improve my thinking skills and brought my research to a higher level.

This research would have been impossible without the aid and support of professor PhD. José Manuel Martínez, whose expertise was invaluable in formulating the research questions and methodology. I would also like to thank him for his devotion and guidance.

My sincere thanks to ”Universidad Técnica Federico Santa María” for giving me the opportunity to carry out my graduate studies, honoring me with the *Beca Puente para estudios de Magíster*, which allowing me to grow as a professional.

Besides, I would like to acknowledge Agencia Nacional de Investigación y Desarrollo (ANID), Proyecto Fondecyt de Iniciación en Investigación 2020, Project ID 11200178, for supporting this research.

I am deeply grateful to all the professors from Department of Electronics Engineering who contributed to my training; thanks for all their teachings and valuable support.

In addition, I would like to thank my parents and brother, whose love and support go beyond any geographical border. Thanks for being my biggest inspiration.

To the rest of my family and friends, words are not enough to express my gratitude. Thanks for always being there for me.

Last but not least, thank you to all the people Chile allowed me to meet. Without you, I could have not reached this goal, making their help invaluable.

ABSTRACT

Appliance load monitoring in smart homes has been gaining importance due to its significant advantages in achieving an energy efficient smart grid. The methods to manage such processes can be classified into hardware-based methods, including intrusive load monitoring (ILM) and software-based methods referring to non-intrusive load monitoring (NILM). ILM is based on low-end meter devices attached to home appliances in opposition to NILM techniques, in which only a single point of sensing is needed. Although ILM solutions are relatively expensive, they provide higher efficiency and reliability rather than NILMs do. Moreover, future solutions are expected to be hybrid, combining the benefits of NILM along with individual power measurement by smart plugs and smart appliances. This thesis proposes a novel ILM approach for load monitoring that aims to develop an activity recognition system based on an IoT architecture. The proposed IoT architecture consists of an appliances layer, a perception layer, a communication network layer, a middleware layer, and an application layer. The application layer consists of an appliance recognition module and activities of daily living (ADL) classification algorithm. The main function of the appliance recognition module is to label sensor data and to allow the implementation of different home applications. Three different classifier models are tested using real data from the UK-DALE dataset: feed-forward neural network (FFNN), long short-term memory (LSTM), and support vector machine (SVM). The developed ADL algorithm maps each ADL to a set of criteria depending on the appliance used. The features are extracted according to the consumption in Watt-hours and the times where appliances are switched on. In the FFNN and the LSTM networks, the accuracy is above 0.9 while being around 0.8 for the SVM network. Other experiments are performed to evaluate the classifier model using a test set. A sensitivity analysis is also carried out to study the impact of the group size on the classifier accuracy. Once results were obtained, the proposed ADL classification system was enhanced in two frameworks: a training framework and an inference framework. This is to allow a practical implementation of the system. In this regard, several modifications were made in the appliance recognition module, including the use of new data, and therefore new appliances: an electric vehicle, an oven and a microwave, from the Dataport dataset. The frameworks include graphical interfaces that

significantly facilitate its use. The dataset configuration, pre-processing and classification parameters can be easily selected and modified. In the feature extraction, inside a sliding window, statistical features of the power samples are computed. In this way, the same pre-processing can be applied in the two different datasets. A feature importance analysis can also be performed to analyze the contribution of the selected features in the models predictions. With this implementation, the real-time operation is directly related with the size of the window used.

RESUMEN

El monitoreo de carga de electrodomésticos en hogares inteligentes ha ido ganando importancia debido a sus múltiples ventajas para lograr una red eléctrica inteligente eficiente. Los métodos para administrar dichos procesos se pueden clasificar en métodos basados en hardware, conocidos como monitoreo de carga intrusivo (ILM) y los métodos basados en software que se refieren al monitoreo de carga no intrusivo (NILM). ILM se basa en dispositivos de medición de gama baja conectados a electrodomésticos en oposición a las técnicas NILM, donde solo se necesita un único punto de detección. Aunque las soluciones ILM son relativamente caras, estas brindan mayor eficiencia y confiabilidad que las soluciones NILM. Además, se espera que las soluciones futuras sean híbridas, combinando los beneficios de NILM junto a la medición de energía individual mediante enchufes y electrodomésticos inteligentes. Esta tesis propone un novedoso enfoque ILM para el monitoreo de carga que tiene como objetivo desarrollar un sistema de reconocimiento de actividad basado en una la arquitectura IoT. La arquitectura de IoT propuesta en este estudio consta de una capa de dispositivos, una capa de percepción, una capa de red de comunicación, una capa de *middleware* y una capa de aplicación. El sistema propuesto consta de un módulo de reconocimiento de electrodoméesticos y un algoritmo de clasificación de actividades diarias (ADLs). La función principal del módulo de reconocimiento de electrodomésticos es etiquetar los datos de los sensores y permitir la implementación de diferentes aplicaciones domésticas. Se prueban tres modelos de clasificadores diferentes utilizando datos reales del *dataset* UK-DALE: *feed fordward neural network* (FFNN), *long short-term memory* (LSTM) y *support vector machine* (SVM). El algoritmo de ADL desarrollado asigna cada ADL a un conjunto de criterios según el dispositivo utilizado. Las características se extraen según el consumo en Watt-hora y los *timestamps* en que estos se encienden. En las redes FFNN y LSTM, la precisión está por encima de 0,9 mientras que en la red SVM la misma es de alrededor de 0,8. Se realizan otros experimentos para evaluar el modelo de clasificador utilizando un nuevo conjunto de prueba. También se lleva a cabo un análisis de sensibilidad para estudiar el impacto del tamaño del grupo en la precisión del clasificador. Una vez que se obtuvieron los resultados, el sistema de clasificación de ADL propuesto se integró en dos *frameworks* de implementación: un *framework* de entrenamiento y otro de inferencias.

Esto con el fin de permitir una implementación práctica del sistema. Para ello, se realizaron varias modificaciones en el módulo de reconocimiento de electrodomésticos, incluyendo el uso de nuevos datos pertenecientes a nuevos dispositivos: un vehículo eléctrico, un horno y un microondas, del *dataset* Dataport. Los *frameworks* incluyen interfaces gráficas que facilitan significativamente su uso. Los parámetros de clasificación, preprocesamiento y configuración del conjunto de datos se pueden seleccionar y modificar fácilmente. En la extracción de características una ventana deslizante calcula las características estadísticas de las muestras de potencia. De esta forma, se puede aplicar el mismo preprocesamiento en los dos *datasets*. Adicionalmente se puede realizar un análisis de importancia de las características para analizar la contribución de las características seleccionadas en las predicciones de los modelos. Con los *frameworks* propuestos, la operación en tiempo real está directamente relacionada con el tamaño de la ventana.

Contents

List of Tables

List of Figures

1 | Introduction

1.1 Problem statement, Context and Motivation

Smart grid is the future electric power system, which supports bi-directional energy and information flow between consumers and service providers with improved reliability, stability, and efficiency. Internet-of-Things (IoT) technology can be used for enabling smart grid to achieve their goals in monitoring, protecting, and controlling through the incorporation of sensors, actuators, and metering devices while supporting various network functions and system automation. IoT technology has caught significant attention in various application domains such as smart buildings, healthcare systems, agriculture, smart cities and smart homes [1, 2, 3, 4].

Nowadays, the applications of smart home concepts and home energy management systems (HEMS) have been catching an increasing attention of the research community due to many advantages they offer. These technologies aim at facilitating users' operation and management of household appliances to operate automatically and optimally. Furthermore, they represent a crucial step in achieving energy efficiency. To build such management systems, it is necessary to identify and control the energy consumption of major appliances in the household which are responsible for a higher electrical consumption [5]. The home appliances are mostly used for routine housekeeping tasks such as cooking, doing laundry, or food preservation. Among these common loads, there are the washing machine, the dishwasher, the freezer, electric vehicles (EVs) and the heating, ventilating and air conditioning (HVAC). Electric vehicles are expected to play an important role in the future smart grid due to their environmental and economic benefits. However, the integration of electric vehicles in the distribution power network represents complex problems due to vehicles operation modes that adopt a bidirectional energy flow between electric vehicles and the power grid [6].

The identification of appliance usage opens the door for the implementation of a series of useful applications. Among them, demand response (DR) and load planning programs focus on analyzing individual load levels in homes or buildings. This analysis enables the possibility of identifying less efficient or malfunctioning devices and implementing the appropriate actions intended for reducing consumption. In this context, consumers become a key factor; they not only participate effectively in the sustainable smart grid system, but they can also have statistic direct feedback on real-time power consumption [7, 8]. Additional useful information could also be inferred from appliance data such as consumers' behavior patterns, including occupation, sleep patterns, and other activities. These activities are commonly known as activities of daily living (ADL), with both applications both in the energy domain and in other fields, ranging from commercial services (e.g., customer profiling and targeted marketing) and legal sector (e.g., monitoring of curfews and detection of illegal activities) to remote healthcare monitoring for elder people living alone [9].

To contribute to the development of an efficient HEMS, it is necessary to carry out a process that allows identifying and monitoring main loads in the household. The methods to manage such processes can be classified into two categories: methods based on hardware and those based on software, as shown in Figure 1.1. From bottom to up in the Figure, the task is to identify individual appliance loads through different stages: *Data acquisition*, *Feature extraction*, and *Classification*. The first stage is the process in which data is gathered using a physical device. Then, a further processing (*Feature extraction*) is performed over the acquired samples to obtain a signature that represents the appliance electrical consumption. Finally, the resultant extraction is frequently classified through machine learning (ML) models. Both methods share some common characteristics. One of them is the use of ML techniques for predicting the behavior of appliances and translating row data (e.g., current, voltage and power) into an easy and understandable form. These techniques also allow to make a deeper analysis of the electricity consumption so that it is possible to build a consumer profile, bringing privacy concerns into account [10].

On the one hand, software-based methods include measurements from only a single point of sensing (smart meter device). These methods, commonly known as non-intrusive load

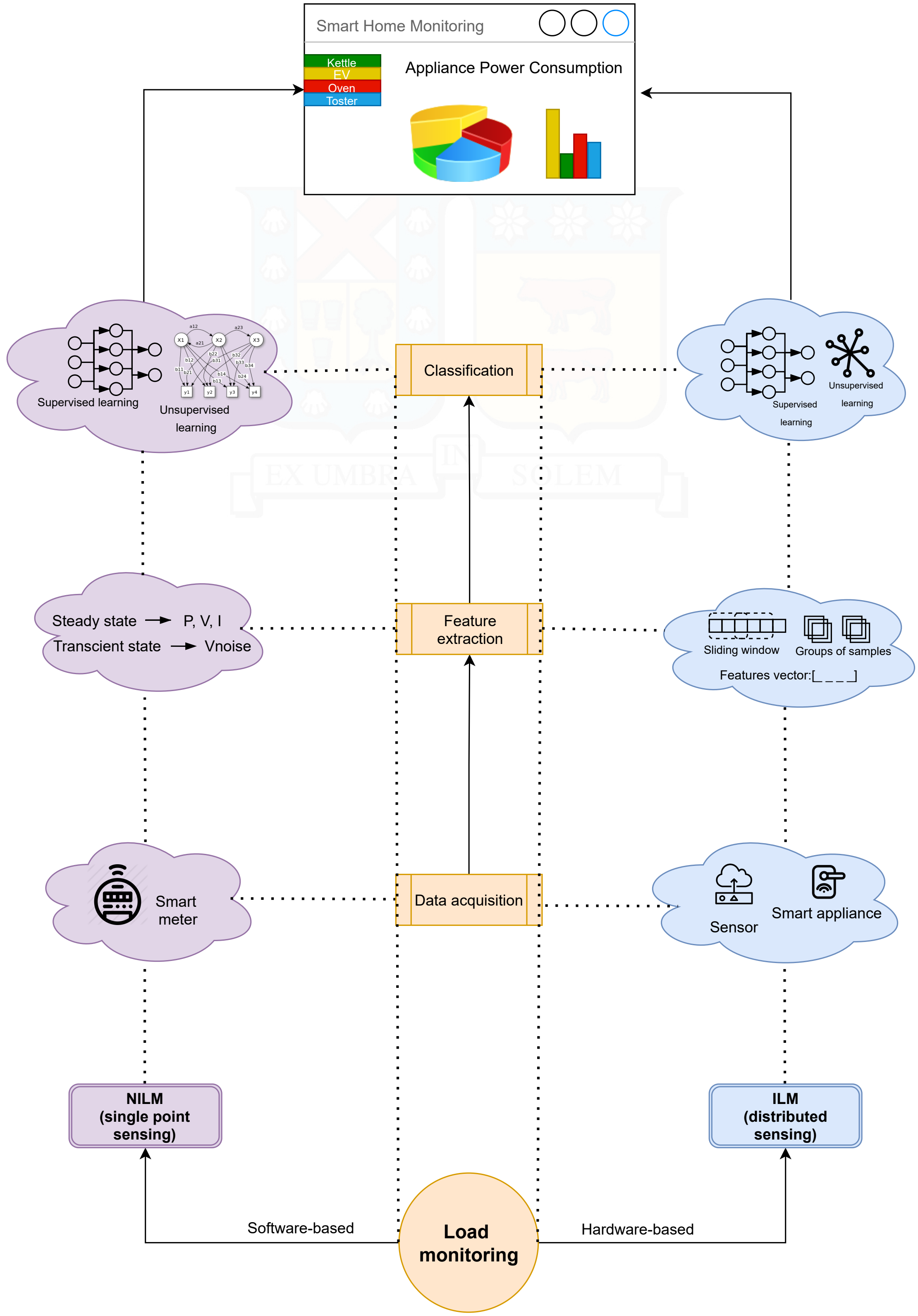


Figure 1.1: Methods of load monitoring. From bottom to up, on the left side: Software-based methods (NILM). On the right side: Hardware-based methods (ILM).

monitoring (NILM), offer an attractive solution, essentially due to their low-cost implementation since they only need a single point of detection. The feature extraction can be broadly divided into two classes, namely, steady-state and transient features. Although these solutions have caught the attention of most studies in the field for the last five years, they have shown less precision and greater difficulty for implementation in real scenarios compared with hardware-based methods. NILM algorithms are mostly based on event detection, sampling the aggregated signal captured by smart meters to obtain individual profiles of electrical appliances. The aggregated signal can be very noisy, and only a few electrical appliances could be detected, depending on the sampling frequency. Even with advanced artificial intelligence (AI) algorithms, it could be possible to monitor only a few major appliances: e.g., oven, washing machine, air-conditioner and EV [7, 8, 9]. When facing these kinds of scenarios in terms of the type of appliance used, performance remains inconclusive on different datasets [11].

On the other hand, techniques based on hardware include methods for intrusive load monitoring (ILM). This technique can be divided into two sub-categories. One refers to a model in which energy consumption profiles are obtained from the device level using sub-measurement sensors attached to appliances, also known as distributed sensing. The second is smart appliances (SA), which are devices having built-in capabilities to monitor and report their energy usage [7, 12]. The feature extraction consists of computing unique vectors using different procedures (e.g., sliding window) to be set as input of an ML-based classifier [13, 14]. Although these solutions can be relatively expensive, they provide higher efficiency and reliability compared to NILMs. Direct sensors have great potential since they have sensing and controlling operation of various devices and appliances because they can be collocated (e.g., turning off a light when an occupant leaves a room). An additional benefit is that these methods typically require a less complex solution regarding appliance recognition. The appliance recognition system assigns a label that corresponds to the device or appliance connected to the sensor. Moreover, future load monitoring techniques are expected to be hybrid, combining the benefits of NILM and individual power measurement by smart plugs, smart appliances, and HEMS [15].

Since smart appliances are not widely used due to their high market prices and interoperability issues, distributed sensing becomes an attractive solution. Distributed sensing

Table 1.1: NILM versus ILM

Aspects	ILM	NILM
Point of sensing	Multiple (distributed sensing)	One (Single-point sensing)
Communication network	Yes	No
Massive deployment	Difficult	Ease of installation
Accuracy and Reliability	High	Lower than ILM

benefits from communication technologies to allow the integration of all electrical devices. Communication networks carry control data generated by sensors attached to home appliances, carrying control commands from the home gateway to the appliances, and from the utility to the appliances registered in the home gateway [16].

In this regard, both wired and wireless communication technologies can be used. Wired technologies include coaxial cable, PLC (powerline communications), twisted pairs and optical fiber. Current in-use wireless technologies include WiFi, ZigBee, and Bluetooth. In most cases, wireless technologies are preferred due to the ease of installation, convenience, and reliability [17]. Recent approaches, see [18], describe the use of low-cost devices for communication and data processing, supporting Long-Range (LoRa) technology and connection to The Things Network to build a system that incorporates various sensors.

Taking all this information into account leads us to think about an ILM solution as an internet of things (IoT) platform used for load monitoring and its numerous applications. Among these applications, identifying ADLs is a good choice in terms of resident autonomy. One of the most common examples is that this choice has allowed older people to be nursed at home, also enabling to build up a consumer profile that can contribute to more efficient energy use. For this reason, ADLs are the most suitable to be provided as inputs for different home applications [19]. Table 1.1 summarizes the comparison of both techniques. Compared with the ILM solution, NILM uses only one sensing point (smart meter). Therefore, a communication network that allows data exchange between sensors and the home gateway is not necessary. These aspects have expanded the acceptance of a massive deployment of these NILM solutions; however, the reliability of these systems is still a challenge.

Most of the previous research works are related with NILM techniques [9, 20, 21, 22, 12, 23, 24, 25, 26]. In contrast, access to smart meter measurements is still limited and there are challenges in some countries due to regulation and implementation issues. Furthermore, high-resolution data cannot be achieved with most current commercial smart meters today having complexity in setup, data storage, and cost. On the other hand, with the advances in IoT and communication technologies, the ILM solution has become an affordable option to overcome the difficulty of implementing NILM solutions. ILM is a promising approach for the future development of residential load monitoring for different applications such as home automation, load forecasting, demand response, energy feedback, and healthcare systems. However, different requirements should be considered concerning data resolution, accuracy, real-time, and the number of appliances to be covered [15, 20].

Based on the information given above, Figure 1.2 shows a graphic description of the research fields involved in load monitoring, and how they correlate with each other. Load monitoring in the context of smart homes is part of the hardcore of an efficient smart grid, which in turns, is one of the several branches of IoT. On the other hand, load monitoring requires to identify the source of each load being monitored, from which it feeds from deep and machine learning models, both being a key part of artificial intelligence (AI). Figure 1.2 shows a clear evidence of the particular research field covered in this thesis: load monitoring, which concerns the correlation of more general areas. This study presents a novel ILM approach for load monitoring and activity recognition based on an IoT architecture in the context of smart homes.

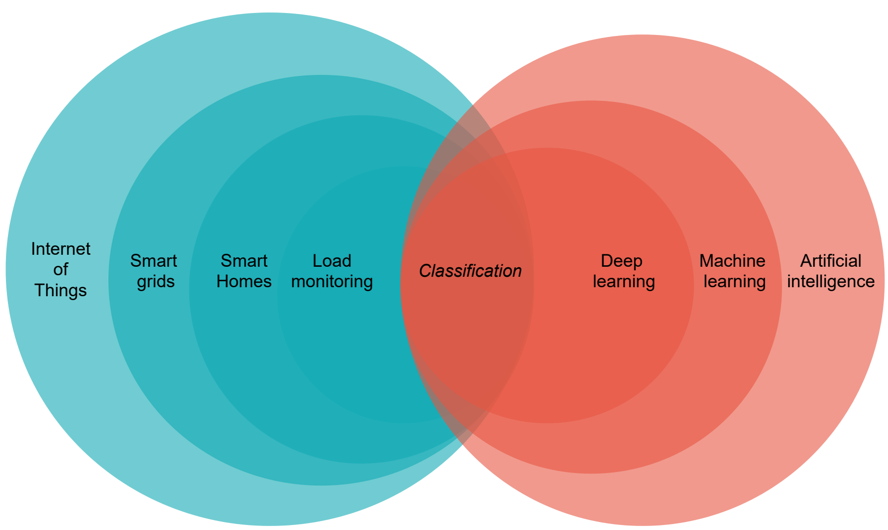


Figure 1.2: IoT and AI relation which allows the implementation of load monitoring solutions.

1.2 Hypothesis

This thesis addresses the following questions regarding load monitoring and activity recognition in the context of smart homes.

1. *Could an IoT architecture support intrusive load monitoring applications?*
   An intrusive load monitoring system consists of the deploying a set of metering devices that collect information regarding appliances' consumption. The samples gathered are usually stored and processed in a cloud server. Therefore, the communication between the metering devices and the server is a must. Since metering devices (sensors) are deployed, a communication network and a server are needed, an intrusive load monitoring system can be understood as an IoT architecture which supports ILM-related applications. This thesis proposes an IoT architecture capable of holding a framework for appliance and activity recognition in the context of smart homes. Different from former ILM approaches, this architecture allows the analysis of load monitoring problems from an IoT perspective, taking scalability and standardization concerns into account.

2. *Could an intrusive load monitoring system allow to recognize activities of daily living efficiently?*
   The analysis of appliance data provides valuable information, e.g., behavioural patterns of consumers by identifying ordinary activities (ADLs) [15]. However, this analysis has usually been performed from a non-intrusive perspective [9, 20]. This thesis describes an ADL classification system that allows the identification of ordinary activities in a simple way, based on an ILM approach. An intrusive perspective can overcome the reliability constraints which limit the implementation of load monitoring systems in practical scenarios and even in different datasets. The ADL classification system resides in the application layer of a proposed IoT architecture. To achieve this goal, an appliance recognition module identifies the target loads being monitored through machine learning techniques, and then, a set of activities of daily living are inferred from the appliances being used through and an ADL classification algorithm. The system is evaluated through the computation of different metrics such as *accuracy*, *precision*, *recall* and *F1-score*.

3. *Could the ILM system be applied in different scenarios?*

   This thesis proposes an ILM solution for appliance recognition which can include multiple appliances, e.g., electric vehicles, washing machine and dishwasher. In addition, the proposed system can be applied on different datasets, and thus to process data in different formats, i.e., with different scale and structure. Another fact to take into consideration is that the system also fits the requirements to be implemented in practical scenarios (real-time operation), being able to deal with class imbalance since not all the appliances will be used at the same time.

1.3 Objectives

General Objective:

- To design and evaluate an IoT based architecture for load monitoring and activity recognition in smart homes.

Specific Objectives:

1. To develop an IoT based architecture for load monitoring and activity recognition in smart homes.
2. To develop an appliance recognition framework to identify major loads in households. Using real data from well-known datasets, the system needs to be evaluated in order to be furtherly implemented in a practical scenario.
3. To implement a real-time activity recognition system and classify the daily living activities using the collected data from a smart home.

1.4 Methodology

The methodology to meet the objectives stated above consists of the following steps:

1. Studying the related work and solutions reported in former literature concerning IoT architectures, appliance recognition, real-time implementation, and ADL classification.

2. Defining the structure of IoT architecture, providing a detailed description of each layer, which include the physical things layer, perception layer, communication network layer, middleware layer, and application layer.
3. Selecting the target appliances; the selection must be based on its power demand and the activity that could be inferred from its usage. The main appliances are a washing machine, an oven, a dishwasher, an iron, a hairdryer, a microwave and an electric vehicle.
4. Choosing appropriate datasets that contain data from the selected appliances. This data will allow us to carry out the activity recognition system.
5. Designing the appliance recognition framework and test different machine learning models for classification.
6. Implementing the activity recognition algorithm which performs different experiments to analyze and validate results.

For Specific Objective 1:

1. Studying state-of-the-art of IoT architectures and the main layers included.
2. Proposing an IoT architecture for load monitoring and its various applications based on previous studies.

For Specific Objective 2:

1. Selecting the target appliances to be monitored.
2. Choosing an appropriate dataset which contains data from the selected appliances. These data will allow to carry out the activity recognition system.
3. Designing the appliance recognition framework and test different machine learning models for classification.
4. Performing different experiments to test the parameters sensitivity of the system.
5. Analyzing and evaluating results in terms of different classification metrics such as accuracy, precision, recall and F1-score.

For Specific Objective 3:

1. Studying related work and selecting the activities of daily living to be classified.
2. Developing and algorithm to infer activities based on the timestamp and detected appliances.
3. Analyzing the main activities during an interval of time aiming at developing a consumer profile.

1.5 General Structure of the Document

The thesis is oriented to present an IoT based approach for load monitoring and activity recognition in the context of smart homes. The document is divided into six chapters.
Chapter 1 states the research subject, giving an insight of the problem statement, and presenting the general and specific objectives, the hypothesis and methodology to follow.
Chapter 2 explains the fundamentals of ILM and presents the most relevant related work. Both Chapter 1 and Chapter 2 represent the theoretical background for the development of the thesis.
Chapter 3 describes the proposed IoT architecture and the appliance recognition module, discussing the results for this stage.
Chapter 4 shows the ADL classification algorithm and the analysis made up to build a consumer profile.
Chapter 5 proposes the framework for appliance recognition. This is in view of building a system ready to work in a real-case scenario.
Chapter 6 details the overall conclusions and future work.

2 | Related Work

The intrusive load monitoring (ILM) technique is based on measuring the electricity consumption of appliances using a low-end metering device. The applications of ILM could be implemented in both household and building contexts. The submeters or sensor nodes are typically placed close to the target appliances. The name intrusive is used since extra types of equipment, in addition to smart meters, are needed to identify electrical devices appropriately. According to [14], ILM solutions can be categorized into three groups in terms of equipment deployment granularity:

- **ILM Group 1:** This category includes submeters used to monitor household' zones or areas, measuring the consumption after the primary utility energy meter.
- **ILM Group 2:** This category groups submeters at the plug level to directly monitor appliances connected to the outlet or multi-outlet.
- **ILM Group 3:** This category consists of submeters directly embedded in the appliances or placed in a dedicated outlet (i.e., outlet for a specific appliance).

Currently, the aim of ILM is to pervasive computing (i.e., ambient intelligence, self-sensing spaces) [10]. The general procedure is shown in Figure 2.1. There are a series of devices that can be used to collect data: either smart appliances or a sensor/actuator attached to target appliances. Although smart appliances have much better performance in terms of applicability, generalizability and accuracy, their elevated commercial prices restrict their massive deployment. Hence, smart appliances remain out of the scope of this research. With the use of sensors and actuators, the main idea is to detect the appliance's electrical consumption based on their activations. Therefore, some computational procedures are necessary. First, features are extracted, and they can be used to show certain statistics, such as the maximum, minimum and mean value of an electrical measurement (e.g., power, voltage, current). Then, the statistical features are classified by using ML learning techniques. The latter process varies from supervised to unsupervised learning techniques (e.g., Support Vector Machines and K-means, respectively) [14, 13]. As the first stages of ILM involve hardware devices (sensors and actuators) and further processing is software-based, performed in a central data server (middleware), a communication network is needed to allow data exchange between sensors and the server. Middleware solutions are used to integrate and coordinate the nodes, thus achieving a real-time status and management of the household [10].

Another way used to implementing ILM, called *indirect intrusive load monitoring*, mea-

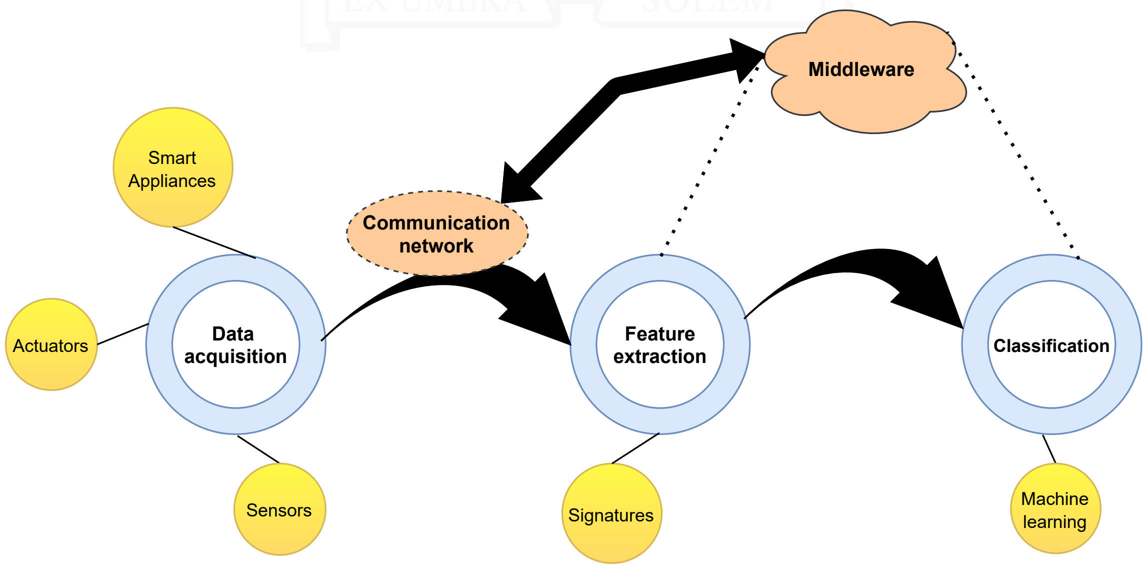


Figure 2.1: General procedure in Intrusive Load Monitoring

sures non-electrical characteristics from which each appliance's power demand is inferred. There are three forms of indirect intrusive load monitoring technique: appliance tagging, ambient sensors and conditional demand analysis. Appliance tagging is the modification of smart appliances so that a tag displays a unique signal when the appliance turns on or off. However, this solution requires a more complex processing in comparison with the one based on direct sensors [27]. Thus, it is also discarded in this current work.

Intrusive load monitoring provides accurate results and allows each individual appliance's energy consumption to be communicated to a central hub [27]. In addition to entering the house (thus the system is referred to as intrusive), the main constraint is cost. The vast majority of non-intrusive approaches [9, 28, 26, 29] claim that ILM imposes high cost and complexity of installation. However, in [11], the authors perform a cost study that keeps the cost as low as possible for every needed part. Notably, one of the most expensive parts of their platform is the transformer to power the Arduino. They use a hardware platform developed to make use of an Arduino UNO to collect energy consumption data in real time, with the help of a single CT-sensor. The cost of these physical devices (sensors) has greatly decreased in the recent years [30]. In addition, recent introduction of energy monitoring plug-in devices makes the ILM technique relatively simple to deploy to the point that in some cases it does not require any installation. These devices are ready-to-use and only need to be plugged in between an electric wall socket and the appliance's plug [31]. Moreover, in [15], it is demonstrated how the development of power electronics significantly improves the accuracy and flexibility of power control, but it greatly affects the applicability of NILM methods. Power converters not only allow the power of appliances to be continuously adjusted, but also to eliminate harmonics and compensate the reactive power. As a result, features of appliances will become indistinguishable, which is one of the main causes that make future trends in load monitoring expected to be hybrid, combining NILM, individual power measurement by smart plugs, smart appliances, and HEMS [15].

2.1 Data acquisition

Data acquisition phase obtains a load measurement at an adequate rate so that distinctive load patterns can be identified in the next stages [10]. This phase is performed to have a widespread perception of energy supply and demand by sensors from different energy sectors, also being responsible for accurate control by actuators [30].

2.1.1 Metering devices

The basis for residential load monitoring lies on metering. In essence, there are current and voltage sensors on the main power supply to measure the electricity consumption in an entire household. Additionally, as it was previously stated in this chapter, there may be sensors for specific appliances or power lines, as well as non-electrical sensors, including occupancy and environmental sensors distributed throughout the household. The measurements are either sent directly to a central data server from the sensors or are processed by the local data processing unit and then being sent to the central data server. A central data server can be located in residence receiving the data from the sub-meters; in other words, they are installed in the Home Area Network (HAN), or in the utility company which receives load data from multiple homes, i.e., the neighborhood area network (NAN) [15].

Current and voltage sensors are the most widely used electrical sensors for load monitoring. Current transformers (CTs) generally used for power measurement may not fully meet load monitoring requirements as direct current (DC) and high frequency current signatures cannot be captured. To achieve device-level measurement, dozens of current sensors are required, making it impossible to reach each individual device [15]. To overcome this constraint, the electromagnetic field sensor was proposed by the authors of [32] to indirectly obtain gross apparatus level operating states.

The energy used by household appliances can be highly dependent on environmental conditions, such as temperature, humidity, light, etc. Therefore, environmental sensors have also been introduced in load monitoring studies. This technique was previously defined as *indirect sensing*. As an example, in the BLUED residential load dataset [33], light level, sound intensity, vibration, humidity, barometric pressure, and PIR motion were measured. In the case of the PECAN Street Dataport [34], the water data was also taken into account. For some authors [15], the information on the occupation of a house is also important to save residential energy because it involves the control of the heating, ventilation and air conditioning (HVAC) system, water heater, lighting, as well as residential energy storage such as batteries and electric vehicles (EV), in case there is any. A variety of sensors can be used to detect the presence, including acoustics, lighting, camera, motion, CO2, temperature, laser beam, RFID, and humidity [35, 36]. As an economical solution, information from existing WiFi connections or HVAC sensors has also been used for presence detection [35, 37].

Recent studies, such as [15], argue that efforts to develop high frequency smart plugs to capture load signatures are not available for widespread deployment. However, in [38], the authors perceive smart plugs as a potential complimentary solution that can improve the process of efficient and low-cost load monitoring. From all the ILM solutions being used for load monitoring, smart plugs have shown a great rate of acceptance among users. The

present smart plug's technology success addresses all aspects for effective load monitoring, such as prosumers (original fusion of the words *producer* and *consumer*) being unable to make energy consumption/production changes, ensuring the security and privacy of metering data, and enabling to manage and store vast quantities of the collected data.
The physical form factor of a smart plug is an important one when it comes to show the acceptability of smart plugs by any user. First of all, it must be small, compact and compatible with traditional plugs. Secondly, it should not contain parts which are added to the hassle of installation and use. The design of commercial smart plugs available in the market is possible when putting a lot of thought into the design part, and it largely satisfies the question of aesthetic appeal and form factor. Therefore, an ideal smart plug will be a smart amalgamation of various technologies that have been routinely used independently until now [38]. A list of sensors and actuators commonly used for intrusive load monitoring are shown below.

Sensors:

- Temperature sensor
- Humidity sensor
- Air quality sensor
- Door sensor
- Prosumer sensor
- Occupancy sensor

Actuators:

- Smart socket
- Smart relay
- Smart plug
- Smart switch
- Smart valve

2.1.2 Sampling frequency

The data sampling of load monitoring can be classified into high-speed sampling and low-speed sampling. Depending on the target application, the sampling rate for electricity consumption may vary. Some authors [14, 9, 15] define a fairly high sample rate as from 1 kHz to almost 100 kHz. For higher sampling rates, the identification results are more precise, typically allowing to capture finer state transitions and eventually separating brands in the same category [14]. To monitor the electromagnetic interference generated by the switch-mode power supplies, the sampling frequency is required to be at least hundreds of kilohertz [15].

At present, most commercial devices cannot achieve high-speed sampling. In addition, the complexity of data storage, transmission and processing for high-speed sampling is significantly increased compared with low-speed sampling. Therefore, high-speed sampling is not currently considered a practical approach for large-scale solutions. The low-speed sampling rate is generally set to 1 Hz or even less. As a result, the resolution of the data drops dramatically. With low-speed sampling, transient electrical information can no longer be captured and device characteristics are more likely to overlap [15].

2.1.3 Energy Datasets

Real-world energy datasets are crucial for the development and testing of signal processing and machine learning algorithms to solve energy-related problems such as load monitoring. These datasets are the result of measurement campaigns in homes and/or industrial facilities. With special attention not to interrupt the daily routines within the monitored space so that the recorded data set resembles reality as best as possible. It is a common practice to test novel approaches across multiple data sets to demonstrate versatility as well as generalization skills [39].

Successful implementations of high-resolution residential load monitoring systems are still limited due to factors such as the setup complexity, data storage, and cost, though the measurement technology is relatively mature. The lack of real-world data is one of the major challenges in related studies. In light of this, publicly available datasets have been built by capturing and storing whole-household and circuit/device-specific load data of real houses [15].

Table 2.1 and 2.2 show a comprehensive comparative study of the available energy datasets in terms of different attributes including data acquisition granularity. This information was taken from [31].

2.2 Feature extraction

Feature extraction is a process to extract unique features or so-call signatures from the collected data [12]. As stated in [14], the feature extraction process can be broadly divided into two classes: *time-dependent* and *frequency-dependent* features. The type of

Table 2.1: Details of most commonly used Energy Datasets.

DATASET	RESOLUTION	NUMBER OF HOUSES	DURATION	FEATURES	LOCATION
REDD [40]	1 Hz, 15 kHz	6	2 weeks	p, i, v	USA
AMPds [41]	1 min	1	2 years	p, q, s, i, v	Canada
UK-DALE [42]	1 sec, 16 kHz	5	3-51 months	p, i, v	UK
DRED [43]	1 Hz	1	2 months	p	Netherlands
Dataport [34]	1 min	1000	2012 present	p	USA
GREEND [44]	1 Hz	9	1 year	p	Italy & Austria
ECO [45]	1 Hz	6	8 months	p, q	Switzerland
PLAID [46]	30 kHz	56	Summer 2013	i, v	USA
REFIT [47]	8 sec	20	2013-2015	p	UK
GREEN Grid [48]	1 min	45	2014-2018	p	New Zealand
BLUED [33]	12 kHz	1	7 days	i, v	New Zealand
SustDataED [49]	12.8 kHz	1	10 days	i, v	Portugal
iAWE [50]	1 Hz	1	73 days	p, f, Φ, i, v	India
COMBED [51]	30 sec	-	1 month	p, i, e	India
SmartCity [a] [52]	30 min	-	2010-2014	-	Australia
Smart[b] [53]	1 Hz	3	3 months	p, s	USA
Note: *i, v, p, q, s, f, e, and* Φ represent current, voltage, real power, reactive power, apparent power, frequency, energy, and phase, respectively.					

[a] refers to Smart-Grid SmartCity Customer Trial Data
[b] refers to UMass Smart Home Data Set

Table 2.2: Links of most commonly used Energy Datasets.

Link ID	Link
REDD	http://redd.csail.mit.edu/
AMPds	http://ampds.org/
UK-DALE	https://ukerc.rl.ac.uk/DC/cgi-bin/edc_search.pl/?WantComp=138
DRED	http://www.st.ewi.tudelft.nl/akshay/dred/
Dataport	https://www.pecanstreet.org/dataport/
GREEND	https://www.monergy-project.eu/?page_id=380
ECO	https://www.vs.inf.ethz.ch/res/show.html?what=eco-data
PLAID	http://plaidplug.com/
REFIT	https://pureportal.strath.ac.uk/en/datasets/refit-electrical-load-measurements-cleaned
GREEN	https://reshare.ukdataservice.ac.uk/853334/
BLUED	https://reshare.ukdataservice.ac.uk/853334/
SustDataED	https://aveiro.m-iti.org/data/
iAWE	http://energy.iiitd.edu.in/Datasets.aspx
COMBED	https://combed.github.io/
SmartCity	https://data.gov.au/dataset/ds-dga-4e21dea3-9b87-4610-94c7-15a8a77907ef/details
Smart	http://lass.cs.umass.edu/projects/smart/

information that can be inferred from sub-meters devices data has proven to be slightly related to the sampling frequency [9]. Figure 2.2 illustrates common sampling frequencies used for data acquisition in load monitoring, including the features which can be extracted from the data at each sampling frequency. Figure 2.2 is based on the one given in [9].

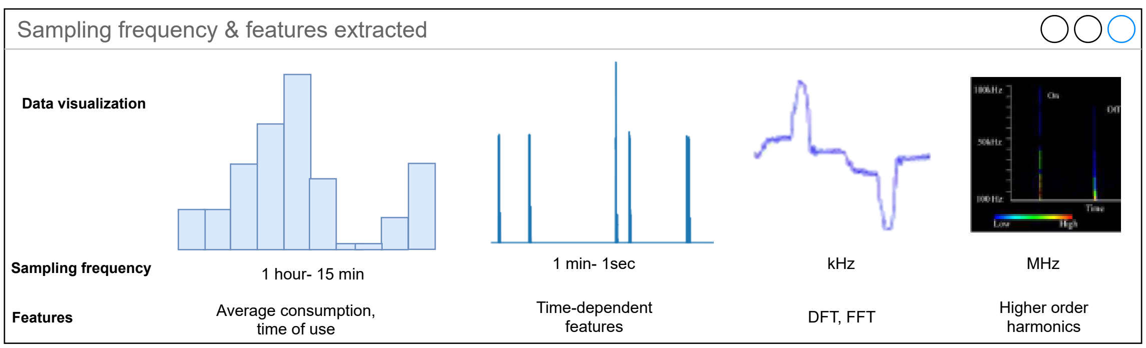


Figure 2.2: Data sampling frequency and features that can be extracted.

2.2.1 Time-dependent features

Common temporal features extracted and used in ILM are the real (active) power (P) and the reactive power (Q). Some appliances can be easily distinguished in the P-Q space depending on their resistive, capacitive and inductive characteristics. Based on these functions, others can be calculated, such as complex power and apparent power. Voltage (V) and current (I) are also very popular to be extracted as features. The voltage-current (V-I) trajectories, the root mean square, the peak values of the current, information about the current form (e.g., crest factor, the form factor or their combination) have also been proposed to characterize appliances [14].

These features provide useful information during the classification stage. Others are based on certain statistics such as counting the number of eventual occurrences of events in a period of time [14] (e.g., the number of transitions between power intervals, the number of edges, and the number of threshold crossings), the maximum, minimum or average power in a certain time interval as shown in [13].

2.2.2 Frequency-dependent features

When the frequency of acquisition is medium to high (1 kHz to 100 kHz), the frequency analysis can be used in the context of the appliance recognition tasks. Commonly, Discreet Fourier Transform (DFT) or Fast Fourier Transform (FFT) are applied. DFT is reported to be less efficient when the sampling frequency is low. The impact of FTT-based functions has been evaluated showing differences in the system performance by injecting other characteristics derived from the current in time. Using FFT on signals with a sample rate of 20 kHz, the harmonic content of the device is unique and even devices of the same category can be distinguished [14].

2.3 Classification

Classification is the task of providing a label for the data collected from each sensor. This label corresponds to the device or appliance connected to the sensor. Labeling is possible since ILM measures the energy usage of appliances separately with sensors (submeters). Therefore, appliance-level load data can be directly labeled [15].

Machine learning models, such as feed-forward neural networks (FFNN), long short-term memory networks (LSTM) networks and SVMs are preferred for labeling tasks. They allow the identification of data, thus contributing to the correct generalization of the classifier, which implies a proper performance in front of unseen data and removes the need of manually setting a label for each sensor [14, 15]. Every ML model requires to be trained using historical data in order to build the model. In some cases, like [20], data recorded through acquisition campaigns using commercial sensors or specific hardware. Since these campaigns not only imply research, but economical efforts, it cannot be implemented in most cases. As an alternative, publicly available databases like the ones presented in 2.1.3 can be used [14].

Machine learning models are categorized as either **supervised** or **unsupervised**. Both categories are discussed below.

2.3.1 Supervised Learning

Supervised learning involves learning a function which maps an input to an output based on sample input-output pairs. In classification models, the output is discrete [54, 55]. Here are some of the most commonly used classification models in load monitoring.

- **Support vector machines (SVM)** are an ML technique that bases on the structure risk minimum principle of statistical theory. It can be used for both classification and regression problems, and its main advantage lies in its working principle. It constructs a hyper-plane or set of hyper-planes in a high or infinite dimensional space. A good separation for hyper-planes implies a larger distance to the nearest training data points of any class, which is often referenced as functional margin. The larger the margin, the lower will be the generalization error of the classifier [55, 56].
  To be adapted to nonlinear data, the only requirement is to change the kernel. The kernel function aims to take input data and to transform it into the required form. Usually, default 'rbf' function is selected for implementation, which means that Radial Basis Function acts as a kernel. This is a real function whose value only depends on the distance from the origin or as an alternative on the distance to some center. SVM was used as a classifier model in [56].

- **Decision Tree:** is a tree-like flowchart, in which each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label. Growing a tree involves deciding on which features to choose and what conditions to use for splitting, along with knowing when to stop. The performance of a tree can be further increased by pruning. It involves removing the branches that make use of features having low importance. This way, complexity is reduced, and thus increasing the predictive power by reducing overfitting [55]. Decision Tree was one of the algorithms tested in [11] for classification, as well as in [57] for healthcare applications.

- **Random Forest:** is an algorithm that builds a "forest", or so-called, an ensemble of decision trees, usually trained with the "bagging" method. The bagging method, in general terms, is a combination of learning models which increases the overall results. The random forest adds additional randomness to the model, as the trees grow. Instead of searching for the most important characteristic when splitting a node, it searches for the best characteristic among a random subset of characteristics. It results in a wide diversity that generally makes a better model [55]. Random Forest achieved the highest accuracy among the different models tested in [11].

- **K-Nearest Neighbors (K-NN):** this is an algorithm that assumes the similarity between the new case/data and available cases, putting the new case into a category that is mostly similar to other available categories. It first selects the number K of the neighbors and calculates the Euclidean distance of all K neighbors. Then, it takes the K nearest neighbors according to the computed distance. Among these K neighbors, it counts the number of the data points in each category, and it finally assigns the new data points to that category for which the number of the neighbor is maximum [55]. This simple and powerful algorithm was also tested in [11] for classification.

- **Naïve Bayes** or simply "*idiot Bayes*" is an algorithm that bases on Bayes theorem to solves classification problems. It assumes that the occurrence of a certain feature is independent of the occurrence of other features. This algorithm generates a Likelihood table finding the probabilities of given features. Then, it uses Bayes theorem to calculate the posterior probability [55]. This algorithm was implemented in [58] in the context of healthcare applications.

- **Feed forward neural networks (FFNN):** a feed forward neural network or multilayer perceptron (MLP) is a machine learning model in which information flows from the input through intermediate computations to finally reach the output. There are no feedback connections, meaning that none of any output layers is fed back into itself. When determining a FFNN model configuration, no specific procedures are established to choose the number of hidden layers and neurons units. Too many parameters will conduct to overfitting, which affects model generalization and performance; on the contrary, a very simple model tends to underfit and thus, more features need to be extracted from data [54]. The number of hidden layers and neurons is directly proportional to system requirements such as computational power, time, and labeled data [9]. Such a model was used in [13] to classify a vector of ten features.

- **Long short-term memory (LSTM):** a long short-term memory network is a type of recurrent neural network (RNN) model that employs a memory cell with gated inputs, outputs, and feedback loops. Its main contribution is that it addresses the vanishing gradient problem, very common for RNNs, in which gradient information disappears or explodes, and it is propagated back through time [54]. Thus, this kind of system is reported to be better suited for time series data [26]. A LSTM network was another model tested in [11] for classification.

In supervised learning algorithms, the objective is to minimize the error that comes from the *loss function*, for each training example during the learning process. This is done by using some optimization strategies like *gradient descent*. Gradient descent is an optimization algorithm used to minimize some functions by iteratively moving in the direction of steepest descent as defined by the negative value of the gradient [54].

2.3.2 Unsupervised Learning

Unsupervised learning techniques, unlike supervised learning, are used to draw inferences and find patterns from input data without references to labeled outcomes [54, 55]. In ILM, two models have been commonly used. Both are detailed as follows:

- **Hidden Markov Models (HMMs):** implement a generative probabilistic model, in which a sequence of observable X variables is generated by a sequence of internal hidden states Z. The hidden states are not observed directly. The transitions between hidden states are supposed to have the form of a (first-order) Markov chain. They can be specified by the start probability vector π and a transition probability matrix A. The emission probability of an observable can be any distribution having parameters θ conditioned on the current hidden state. The HMM is completely determined by π, A and θ. For ILM, HMMs have been mostly used for healthcare applications. An example is the one implemented in [59].

- **K-means:** is an unsupervised clustering algorithm. The objective is to find ”K” groups (clusters) among raw data. The algorithm works iteratively to assign each "point" (the rows of the input set form a coordinate) one of the ”K” groups based on its characteristics. They are grouped in accordance with the similarity of their features (columns). The *centroids* of each group will be the coordinates of each of the K sets that will be used to label new samples. The Labels for the training dataset belong to one of the K groups already formed. These groups are defined in an organic way, which means that their position is adjusted in each iteration of the process until this algorithm converges. Once the centroids are found, they are analyzed to understand what their unique characteristics are, compared with those of the other groups. This algorithm has received less attention in the context of ILM compared with various NILM proposals. An unobtrusive model based on K-means was proposed in [55] to recognize activities of people living alone using PIR motion sensors and door sensors.

2.3.3 Evaluation metrics

For evaluating the performance of ILM algorithms, the research community has adopted for widely used performance metrics in ML-based classification models. Some of these classification metrics are redefined forms of typical machine learning metrics, including recall, precision, F1, accuracy, ROC curves, Confusion matrices, etc [54, 55].
To have a better understanding of these metrics, some basic concepts has to be considered:

- **Ground-Truth:** Predefined information that represents the expected result for the resolution of a certain problem. This information is used as a basis for comparison when algorithms are evaluated to solve the same problem [55].
- **True positive (TP):** the algorithm detects a real situation (it exists in the ground-truth and in the results of the algorithm) [55].
- **False positive (FP):** the algorithm detects an unreal situation (it exists only in algorithm results) [55].
- **True negative (TN):** element that does not appear even in the ground-truth or in the algorithm results [55].
- **False negative (FN):** a non-real situation was detected (it only present in ground-

truth) [55].

Concerning these definitions, more information can also be given to characterize the algorithms. **Precision** is defined in equation 2.1 as the relation between hit rate and algorithm errors. It is the fraction of all relevant instances divided by the instances obtained.

$$Precision = \frac{TP}{TP + FP} \tag{2.1}$$

Recall or Sensitivity, on the other hand, is defined in equation 2.2 as the fraction of relevant instances that have resulted from the total number of relevant instances. Both precision and recall are based on an understanding and measurement of relevance.

$$Recall = \frac{TP}{TP + FN} \tag{2.2}$$

F1-score is used to combine the precision and recall measurements into a single value. This is a practical metric since it makes it easier to compare the combined performance of precision and recall among various solutions. The F1 value assumes that precision and recall are equally relevant to the evaluation [55]. This metric is computed after equation 2.3.

$$F1 \quad score = 2 * (\frac{precision * recall}{precision + recall}) \tag{2.3}$$

Accuracy in classification problems is the number of correct predictions made by the model over all kinds of predictions made. This is reflected in equation 2.4. Accuracy is a good measure when classes in the data are nearly balanced. On the contrary, it should not be used as metric when the target variable classes in the data represent a majority of one class [55].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.4}$$

A **confusion matrix** is a visualization tool typically used in supervised learning (training samples tagged). Each column of the matrix represents the class instances predicted (e.g., the output of an algorithm). The row represents the true instances of a class (e.g., sample labels, known a priori) [55]. A general example is shown in Figure 2.3. In addition, the transposed of the matrix representation could be given. In this case, it is usually clarified. With the confusion matrix is easy to identify if the model cannot distinguish between two

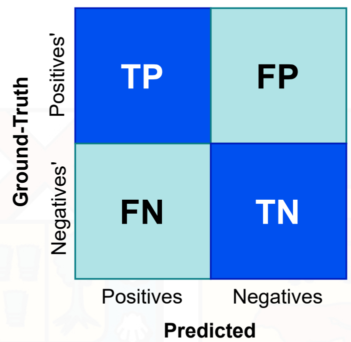


Figure 2.3: Confusion matrix representation.

or more classes. When a dataset is unbalanced, i.e., the number of samples in different classes varies a lot, the classifier error-rate is not representative of the true performance of the classifier. Therefore, a confusion matrix can also be helpful in this kind of situation. Another common visual representation is the **Receiver operating characteristic (ROC)**, or simply, the **ROC curves**. It is a graph that illustrates the performance of a classifier, in the face of variation of its threshold. To graphic an ROC curve, it is necessary to plot the **True positive rate (TPR)**, which is the same as Recall, vs. the **False positive rate (FPR)** or $1 - Recall$, at different thresholds [60]. A general representation of ROC curves is shown in Figure 2.4.

2.4 Appliances' Load Classification

George Hart, as part of his work in the early 90s [61] states that appliances can be classified according to their operational state as follows:

- **Type 1)** Devices having only two operational states (ON/OFF appliances), e.g., toaster, kettle, etc.
- **Type 2)** Multistate devices which could be represented by finite state machines (FSMs), e.g., washing machines, refrigerators, heat pumps, etc.
- **Type 3)** Continuously variable devices (CVD), referring to appliances with variable power absorption characteristics, e.g., electric drills, laptops, etc.

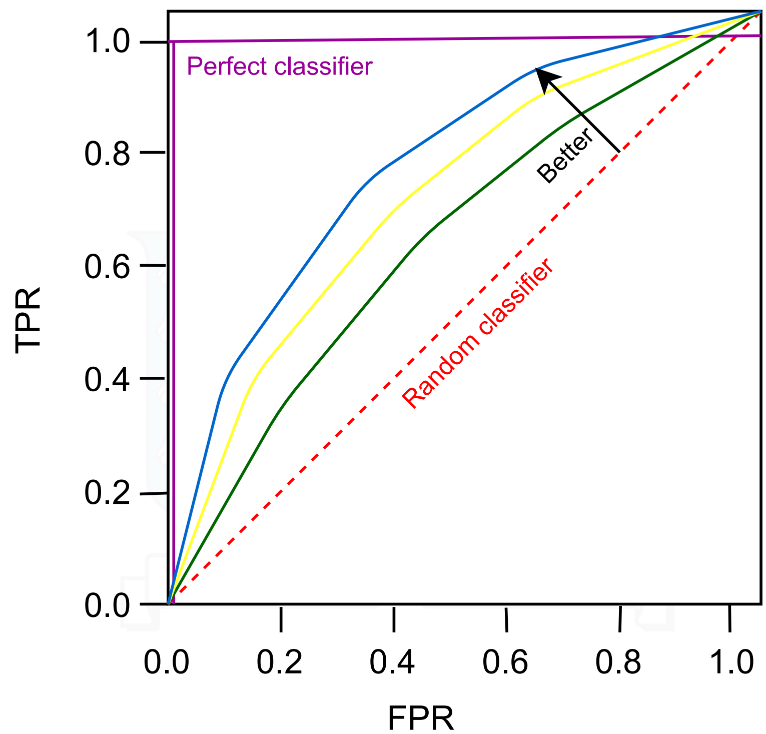


Figure 2.4: ROC curves general representation.

- **Type 4)** Permanent consumer devices, which remain active for a long period of time (weeks or days) consuming energy at a constant rate, e.g., TV receivers, telephones set, smoke detectors, etc.

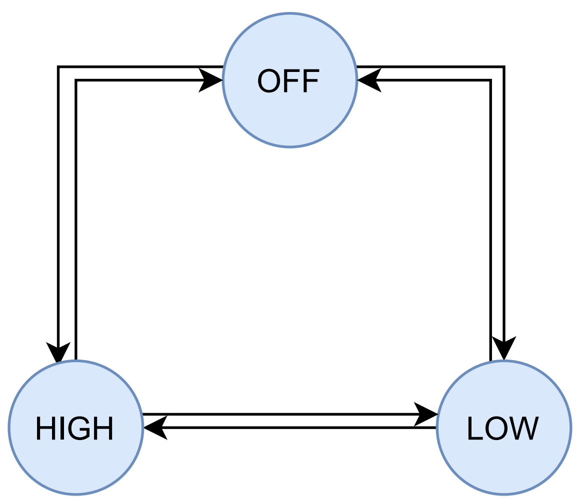


Figure 2.5: Generic model of Finite State Machines (FSM) appliances.

Continuously variable devices are considered extremely difficult to identify since they can exhibit significantly different patterns depending on their usage [11]. Based on their consumption patterns, the class of permanent consumer devices can be understood as a sub-class of ON/OFF appliances [15]. Figure 2.5 shows the operational states of ON/OFF and multi-state appliances. FSMs or multistate devices have a variant number of states, and so the consumption of each appliance. In the case represented in Figure 2.5, a generic appliance of three states is modeled. ON/OFF appliances can be understood as FSMs of two states only, meaning that the appliance is either working or not. This work processes all appliance profiles in the same manner, analyzing the load as a general and allowing that procedure can be applied to the previous four types of data. The feature extraction module that makes it possible is further explained as part of the proposed system.

2.4.1 Modeling of EV Loads

Electric vehicles become a special appliance in the household commonly referred as ”major” loads [12]. Due to the energy storage and bidirectional power transfer capability, EVs are expected to bring significant impacts on residential energy management. Therefore, the modeling of household EV load involves not only the charging behavior but also the driving behavior [15].

Generally, electric vehicle charging loads are monitored by the charging infrastructure or their own battery management system. The energy profiles of electric vehicles charging can be broken down into *gradual increase*, *constant load* and *gradual decrease stages*. Power profiles of EVs changing are usually modeled as constant loads while the duration of the gradual increase stage is short, and the gradual decrease stage only occurs under high state-of-charge conditions. To model the charging time and duration, i.e., the stochastic nature of the charging behavior, is a more complex process compared with the power profile. The upload time distribution can be directly modeled by probability density distribution, such as Gaussian distribution, Poisson distribution or non-parametric probability density estimators. Direct estimation methods usually lack sufficient information about the processes and driving behavior must also be taken into account [15].

2.5 Applications of ILM

Although there are not standard categories for intrusive load monitoring applications, literature has analyzed four main aspects: *energy usage feedback, demand response (DR), load* *forecasting and home automation* [15].

Feedback on energy usage is the most widely used ILM application. The information can be in the form of electricity bills, automated meter reading, interactive tools, or home displays of various data, including consumption comparisons and displays [15]. At this point, *local and global energy consumption understanding* play a key role. *Local energy consumption understanding* aims at providing the household with energy feedback on a single appliance. This feedback is typically given directly on the plug, on the static or mobile displays. On the other hand, *global energy consumption understanding* follows the principle of computing the relative contribution of each appliance to the global consumption through an aggregation of different metering devices [14].

In addition to information on energy usage, *appliance localization* can also be fed back to the consumer. The consumption signature can be used to identify the appliance and the position of the metering device wich provides an indication of its position. Moreover, ILM has demonstrated its potential for the indirect detection of *human activities* through electricity monitoring [14]. Daily routine are defined as *activities of daily living (ADLs)*. In order to classify ADLs, it is first necessary to recognize the appliance connected to the metering device, in a process called *appliance recognition*. This latter process, shown in 2.6, consists of assigning a label in accordance with appliance' name to the samples gathered by the metering device [9]. Figure 2.7 shows a visual representation of both *appliance recognition* and *ADL classification* concepts which show how they complement each other.

Understanding human-building interactions (HBI) at the appliance level in a smart building

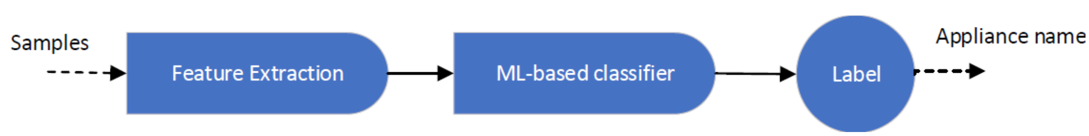


Figure 2.6: Appliance classification module architecture.

context can improve demand-supply balance efficiency, identifying the patterns of using different flexible loads (e.g., EV) in a building. In addition, to provide a statistical measurement to evaluate the benefits of engaging end- users in adaptive management of loads (e.g., engagement in DR programs) [62]. In healthcare monitoring, assistive technologies are often proprietary and tailored in specific application scenarios. By identifying ADLs, it

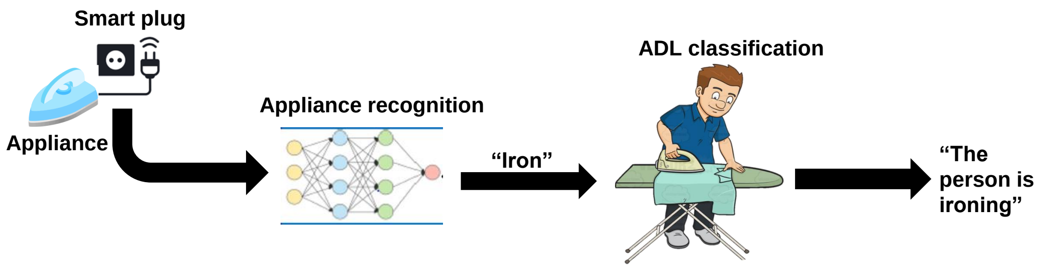


Figure 2.7: Example of appliance recognition and ADL classification.

is likely to provide early intervention for patients with a mental condition such as dementia [20] and to monitor the wellbeing of elder people living alone [63]. The concept of ADL was first described by Katz [64]. The uthors demonstrate that age-related diseases directly impact ADLs by creating ADL indices to measure the dependence level of a patient.

The appliance abnormal behavior can also be detected and fed back to the consumer. This is commonly known as *appliance monitoring* [14]. Anomalies in residential loads include faulty operating conditions and abnormalities in usage patterns that may occur due to appliance failure or user negligence [15].

Demand response (DR) is increasingly important to improve reliability, efficiency and the capacity ability to adopt an increasing penetration of renewable energy. Estimation of DR potential is generally based on load monitoring data. In intrusive load monitoring, DR applications are divided into two levels: circuit breaker level (main DR loads) and strip level (backup DR loads) on two levels Control structure DR [15].

Load forecast plays an important role in demand side management, which depends on load control data. Two main kinds of modeling have been used for these applications: black box models and models based on usage behavior. Then, the information given can be used for load forecast to optimize the DR process, or to forecast the response of loads to different price signals [15].

Finally, the *smart home automation* market is expected to have a deep growth in the near future due to increasing user preference for convenience, connectivity and demand for green and energy efficient product solutions. In this regard, ILM solutions are the foundation for the desired automation. Intelligent management systems along with IoT monitoring and control devices cooperate with each other to create a smart home system [15].

2.6 Communication technologies and Middleware

As it was previously discussed, to connect metering devices to an ILM application host, a communication network must be deployed. Examples of communication technologies include wire field network IEEE 802.3 family, power line communications, serial communication RS-232/485, wireless field network (IEEE 802.11 family, IEEE 802.15 family, mobile field network) (GSM-based 2G, CDMA-based 3G, LTE-based 4G, NR-based 5G), and low power network (NarrowBand IoT, LoRa, Sigfox) [30].

In the context of smart homes, wireless technologies have shown to be preferred over wire field technologies, due to their ease of installation, and their cost-efficient and speed capabilities [65]. The most common wireless standards for ILM solutions are detailed below.

- **WiFi:** uses a radio technology known as **IEEE 802.11**, which can transmit data over short distances using high frequencies. WiFi operates on either 2.4 GHz or 5 GHz depending on its type. Currently, there are five major types of WiFi, known as **IEEE 802.11a/b/g/n/ac**. The two most common and oldest types are **IEEE 802.11b** and **g**, which operate at a frequency of 2.4 GHz. **IEEE 802.11b** has a theoretical maximum transmission speed of about 11 Mbps, while **IEEE 802.11g** can transmit data at speeds up to 54 Mbps. **IEEE 802.11n/ac** are the newest version of the technology, and both are backwards compatible with devices running **IEEE 802.11b** or **g** [66]. **IEEE 802.11n** operates at speeds up to 450 Mbps on either 2.4 GHz or 5 GHz, or even on a single channel or two channels, while **IEEE 802.11ac** reaches the 1300 Mbps [67]. **IEEE 802.11ac** is a faster and more scalable version of **IEEE 802.11n**. It combines the wireless freedom with the capabilities of Gigabit Ethernet [67].

- **ZigBee:** a technology based on the IEEE standard, **IEEE 802.15.4**. ZigBee wireless communication technology uses the Industrial, Scientific, and Medical (ISM) band and generally adopts the 2.4 GHz band to connect with other devices as it is adopted all over the world. Its operation works through 16 channels, having a bandwidth for each of 5MHz. CSMA/CA protocol is used to avoid collisions during transmission [17, 68].

- **Z-Wave:** is the international standard for wireless interconnection of home control systems. The advantage of Z-Wave is that it works at 900 MHz instead of 2.4 GHz. The Z-Wave Alliance states that working at 900MHz provides higher performance for two reasons: less interference (operating at low frequency) and greater penetration of the waves on walls, floors and furniture (having a longer wavelength) [69].

Both **ZigBee** and **Z-Wave** are covered in [17], in which authors present a series of criteria to determine among communication protocols.

- **LoRa:** Lora is a proprietary technology owned by Semtech Corporation, which operates in the ISM band. **LoRaWan** is a network protocol that uses Lora technology which is implemented on top of the Lora physical layer. LoRaWan is in charge of specifying how communications and network architecture are carried out and is managed by the LoRa Alliance. Long range, reaching up to 20 km in open field through point-to-point communication. Operating frequencies are 915 Mhz in America, 868 MHz in Europe, 433 MHz in Asia [18]. In [18], LoRa is the chosen technology to build a system that incorporates various sensors, both personal and in the residence.

On the other hand, **Middleware** technologies act as the intermediary bridge for interaction between IoT devices and applications. There are very high requirements for this layer, such as architecture abstraction, function management, programming design and implementation, Common technologies for ILM including VM-based (MagnetOS, TinyVM), data-based (SINA, TinyDB), service based (LinkSmart, SenseWrap), Application specific (FIWARE, AutoSec), and Fog node based (EMCP, eclipse kura) [30].

2.7 Relevant Research Work

2.7.1 On Intrusive Load Monitoring

There is plenty of research work done regarding ILM. The authors in [14] presented a survey on intrusive load monitoring, which gives details about its implementation requirements. Though this paper only focuses on summarizing the main ILM techniques proposed the in literature, the authors have defined the architecture, feature extraction, and ML models

typically used for ILM applications. This work allows to envision the ILM systems as an IoT platform having more opportunities to enhance different smart home applications.
On the one hand, in [15], the authors explained how the development of power electronics significantly improves the accuracy and flexibility of power control, but it greatly affects the applicability of NILM methods. Power converters not only facilitate to adjust the power of appliances to be continuously adjusted, but also eliminate harmonics and compensate the reactive power. As a result, features extracted from appliances will become indistinguishable. Furthermore, the authors agree that future residential load monitoring is expected to be a hybrid form with the combination of NILM, individual power measurement by smart plugs, smart appliances and HEMS. The author in [10] presented a survey that establishes the basis for the development of important applications in remote and automatic intervention of energy consumption inside buildings and homes. This research work provided a theoretical background on the load monitoring methodologies, concluding that it is feasible to have fine-grained monitoring and control of appliances using ILM in smart houses to provide healthcare, convenience, entertainment, energy efficiency and security.
A survey presented in [70] leads to a fully integrated IoT based health care system, identifying the need to combine the various IoT services. These applications result in a large amount of data to be handled properly when monitoring. In that sense, cloud computing can play an important role, as it is a promising approach for efficient knowledge processing in the health sector. Another approach [71] presented an overview of sensor fusion technology and explored the relationship between sensor fusion and dense sensor networks. The multisensory approach can achieve an impressive result due to the comprehensive description of activities from sensors deployed in an indoor environment.

2.7.2 On Appliance Recognition in Smart Homes

In recent years, many proposals have emerged to describe appliance recognition in the context of smart homes [13, 72, 73, 14, 74]. In [13]; the authors presented an approach to detect and identify in-use appliances when analyzing low-frequency monitoring data gathered by meters (e.g., smart plugs) distributed in a smart home. The system implemented a supervised classification algorithm with artificial neural networks validated by using a dataset of power traces collected in real-world home settings. Since the aim was to develop an appliance recognition system, they mainly focused on the application level in the experiments. In [72], the authors proposed an electrical device identification model based on three features: energy consumption, time usage and location. The information enhanced in such features was used to train six different ML classifier models: Random Forest (RF), Bagging, LogitBoost, Decision Trees (DT), Naive Bayes and SVM. Results showed a high level of accuracy, which represented good performance of the proposed features. In that research study, the authors focused on standard techniques as the main objective was to obtain a neutral assessment of the features. Thus, no specific application such as ADL classification was performed. The authors considered the system as part of a smart grid environment. However, they only centered on the application-related issues without giving any information about the infrastructure or the IoT based architecture to support the system. A supervised learning classifier was developed in [73] for appliance classification based on its power signature. Besides building an individual appliance metering device, the objective was to create what authors called a ”load library” of appliance power signatures for training and recognition. The model employed for classification was a K-Nearest Neighbors (KNN), whose results have proved that timing of data acquisition is critical. Even though experimental results showed high accuracy, the authors did not compare the KNN with any other ML models or algorithms. A recent approach in [11] aimed to design and develop an IoT end-to-end solution to recognize electric appliances that could operate in real-time, considering low hardware cost. Three ML algorithms, K-nearest neighbors (KNN), Decision Tree (DT) and Random Forest (RF), have been implemented to classify the operating appliances. The uthors did not state any requirements regarding the instant when data collection needed to be carried out throughout the appliance's operational cycle, or the amount of data which have to be collected before classification takes place. By only using a high-resolution CT-sensor, they guaranteed cost reduction yet obtaining satisfying results. Their implementation, in a laboratory, was described as a data acquisition system that further processed the data for classification. Although they achieved a high classification accuracy, around 95%, the work did not give any details regarding ADL classification or any other application deployment.

2.7.3 On Activities of Daily Living Classification

Regarding the classification of daily living activities, the authors of [9] presented a deep learning approach based on multilayer feed-forward neural networks (FFNNs) that can identify common electrical appliances in a household from a typical SM measurement (i.e., a NILM solution). The performance of this approach was tested and validated using a publicly available UK-DALE dataset. The detected appliances were used to identify householders' activities. These activities are usually referenced as activities of daily living (ADLs). Thus, they developed an ADL classifier to provide useful information to consumers, including detailed feedback on energy usage and its main contributors, allowing the creation of itemized energy bills. Moreover, information could then be used to show opportunities for energy saving and cost reduction, thus identifying inefficient and/or malfunctioning home appliances. The proposed classification algorithm could be used as an ILM solution.

In [74], the authors presented an activity recognition and anomaly detection approach to identify daily activities in a smart home context. The system was described as a unified deep learning approach based on a Probabilistic Neural Network (PNN) classifier that processed pre-segmented activities, so there was no need of an appliance recognition system. Then, a H2O autoencoder detected anomalies within each activity class. This system may be implemented on COVID-19 scenarios, as recovery from this virus requires isolation to stop spreading the disease and to minimize the risk of transmissions. Therefore, a remote healthcare system likely to help effectively to treat patients without hospitalization.

On the other hand, previous research in ADL classification [9, 20, 63] has been based on NILM techniques. In [75], the authors structured a framework in which the daily activities were detected via a data-driven activity detection approach, using data provided by an NILM system. They aimed to estimate the personalized appliance usage for different daily activities performed by regular residents in a building. Experiments were carried out in three single-occupancy testbed apartment units, using a supervised learning model for activity recognition. The authors of [20] modeled a SVM and a random decision forest classifier using data from three test homes. The trained models were used to monitor two patients with dementia during a six-month clinical trial, undertaken in partnership with Mersey Care NHS Foundation Trust. Using data collected from electricity-based readings, the technology could accurately identify the use of individual electrical devices in the homes and detected routine behaviors of people when anomalies occur.

In a most recent work [76], the authors examined different ways in which smart energy data could be used in remote health and well-being monitoring. The authors considered three broad application domains: ambient assisted living support, population-level screening and support, and self-monitoring. This report also considered energy-health sector research synergies and opportunities to provide solutions at scale. It emphasized the potential benefits of smart energy data when supporting the health-care system, giving a complete description of the two main categories in which this research was focused on: NILM and IoT based methods (ILM).

Other approaches as [77, 78] presented a solution based on IoT, yet considering wearable sensors, such as accelerometers and smart devices, and in the case of [79], the authors proposed an intrusive approach based on computer vision techniques: a background subtraction of images, followed by 3D Convolutional Neural Networks. They used a camera to record the video and a processor that performed the task of recognition, which raised privacy concerns, and hence, a low opportunity for a massive adoption of the system. Recent applications in remote healthcare have reaffirmed the above approaches, proposing innovative solutions in this regard. The authors in [57] presented a smart home control platform which offered fully customized automatic control schemes and performed the analysis of historical records on the use of home automation devices to detect residents' behavioral patterns through IoT and machine learning, improving the comfort schemes of domestic systems. A different solution is illustrated in [58], in which the authors designed a distributed platform to monitor patient's movements and status during rehabilitation exercises. This information could be processed and analyzed remotely by the doctor appointed to the patient. Real-time monitoring of the elderly can benefit from the use of data mining algorithms, namely Support Vector Machine (SVM), from the use of data mining algorithms, namely Support Vector Machine (SVM), Gaussian Distribution of Clustered Knowledge, Multilayer Perceptron, Naive Bayes, Decision Trees, ZeroR and OneR to gain insights into data in order to detect and even predict future falls.

2.7.4 On Communication Technologies for ILM

In terms of communication technologies, a review is presented in [80] to describe the evolution of the power grid from the general conventional infrastructure to the modern smart grid (SG) and the introduction of different network architectures and technologies for communication, automation, and control. Among the different standards presented, ZigBee and Bluetooth appear to be the most popular for a HAN environment. In addition, the authors present the foundations to achieve a reliable, stable, secure, high data rate in the context of SGs. The authors in [65] proposed three communication network architectures to monitor and control distributed energy systems (DES) that included a small-scale wind turbine and a photovoltaic system. Various types of sensor nodes were used to collect different measurements, and the data transmission rate of each sensor was computed according to the sampling frequency. The topologies considered a microgrid connecting various houses and the whole system was simulated using OPNET simulator. Simulation results showed that Ethernet-based architectures performed better than WiFi and ZigBee in view of total end-to-end delay.

A comprehensive review of the HEMS literature referring to main concepts, configurations, and enabling technologies was presented in [17]. The authors provided a summary of HEMS computing trends and popular communication technologies for demand response applications, thus presenting a series of criteria to determine the choice of communication protocol. These were range of coverage, level of security, network size, latency and availability of functionality. For wireless networks, the authors explained that the cost of devices based on the open ZigBee protocol is lower than the one of devices based on the proprietary Z-Wave protocol. ZigBee tends to be used more for research purposes, while Z-Wave is preferred for commercial applications because it has a more extended range and fewer congestion issues. Finally, In [18], low-cost devices for communication and data processing, supported by Long-Range (LoRa) technology and connection to The Things Network were used to build a system that incorporated various sensors, both personal and in the residence, allowing family members, neighbors and authorized entities, including security forces, to have access to the health condition of system users and the habitability of their homes. The COVID-19 pandemic has evidenced the need for massive use of information and communication technologies, as well as an increased knowledge on the difficulties and limitations of their use. To use these kinds of technologies (such as LoRa) can benefit the development of revolutionary applications and mitigate harmful effects of a phenomenon like SARS-CoV-2 virus.

2.7.5 Summary

Based on the above discussion, it is possible to state that:

- Future trends in energy and load monitoring need to feed on IoT technologies to achieve state-of-the-art performance.
- To the best of our knowledge, this research work proposes a novel solution based on ILM for ADL classification that allows to identify the daily activities of house occupants in a simple way, which can be useful in different applications.
- Most promising results in IoT for activity recognition have been obtained in remote healthcare applications. There is a limited number of proposals in other domains such as energy consumption understanding, malfunctioning and anomaly detection.

Table 2.3 summarizes the main aspects of interest analyzed in former research works. The table highlights and compares different research domains regarding general system architecture, appliance recognition model, real-time implementation and ADL classification. The illustrated analysis can serve as a comparison which allows a better understanding and visualization of the objectives of this research work. While almost every paper has focused either on appliance recognition or real-time implementation, this research covers several aspects of overall infrastructure and applications. The design of a communication network has not been considered, since the proposed system is expected to operate at the local area network (LAN) of the target household. Therefore, the router that connects the household and the service provider acts as the household home gateway.

Table 2.3: Summary of former research work

Reference	IoT architecture	Communication technologies	Appliance Recognition	Real-time implementation	ADL classification
[9]	●[a]	●	○[b]	○	○
[11]	●	●	○	○	●
[15]	●	○	○	●	●
[20]	●	●	○	○	○
[13]	●	○	○	○	●
[72]	●	●	○	●	●
[73]	●	○	○	○	●
[14]	○	○	●	●	●
[74]	●	●	●	●	○
[10]	○	○	●	●	●
[63]	●	●	○	●	○
[75]	●	●	○	○	○
[81]	○	○	●	●	○
[77]	○	○	●	●	○
[78]	○	○	●	●	○
[79]	○	●	●	○	○
[70]	○	○	●	●	●
[82]	○	●	●	●	○
[57]	○	○	○	●	○
[58]	○	○	○	●	○
[80]	●	○	●	●	●
[65]	○	○	●	●	●
[17]	○	○	●	●	●
[18]	○	○	●	●	●
Present work	○	●	○	○	○

[a] Not considered
[b] Considered

3 | IoT architecture and Appliance Recognition

3.1 IoT architecture for Intrusive Load Monitoring

As previously mentioned in Chapter 2, regarding an ILM deployment, not only sensor nodes are needed, but a home area communication network is also required. Hence, it is feasible to analyze the architecture of such a system from an IoT perspective. Figure 3.1 proposes an architecture to implement load monitoring applications. It consists of different layers; each of them has a specific function. The result shows a system capable of monitoring the main loads in household which facilitates a series of useful applications such as the classification of activities of daily living.

- The lower part holds the physical devices layer, in which data is collected to form a data flow that is further sent to the upper layer. This layer lets energy transactions take place. The physical devices layer includes common home appliances (e.g., refrigerators, lamps, iron, microwave, oven, etc.) as well as major loads such as EVs and heating, ventilation and air conditioning (HVAC) systems.

- The second level is the perception layer responsible for data acquisition since many sensors and actuators are deployed to gather information. For this process, depending on target application, electricity consumption may vary [14]. A critical parameter to be considered is data sampling, classified into high-speed and low-speed sampling. Data sampling values above 1 kHz are considered high, increasing the complexity of data

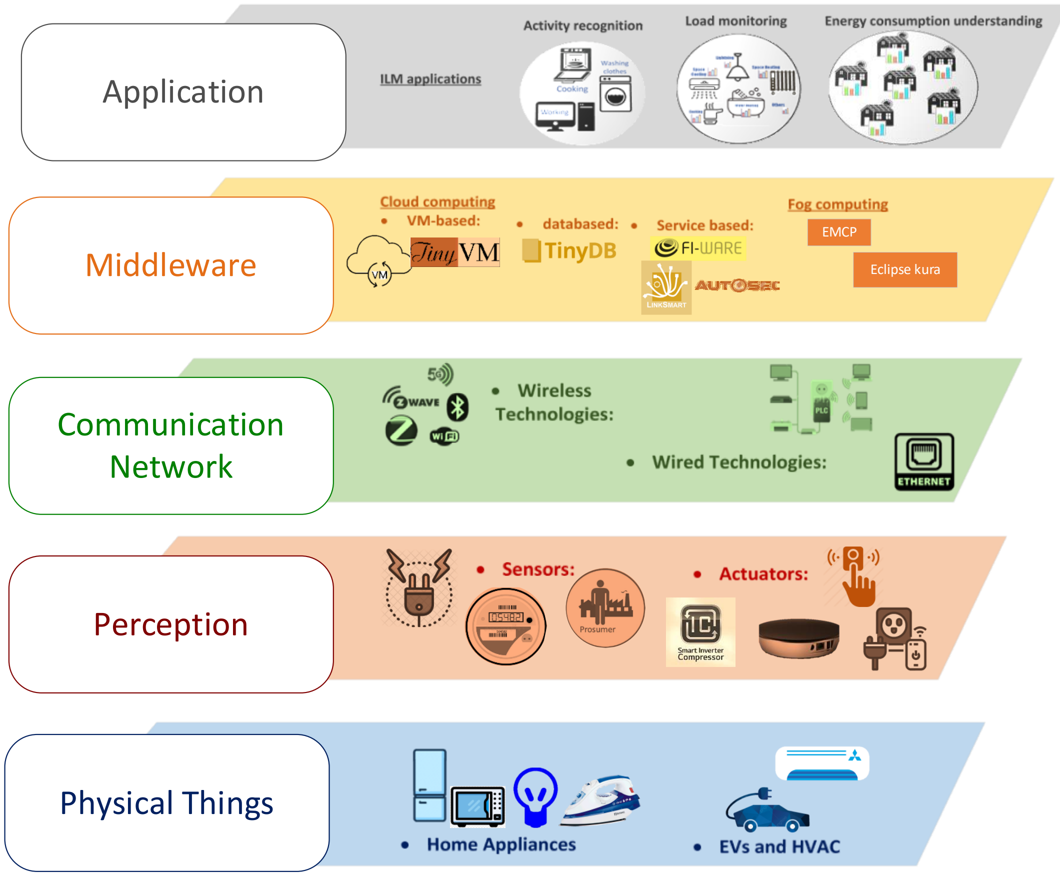


Figure 3.1: Internet of things (IoT) architecture for intrusive load monitoring (ILM).

storage, transmission and processing, compared with low-speed sampling. Therefore, high-speed sampling is often considered far from being a practical approach for large-scale applications [30]. It is possible to measure both time-dependent (active power (P), reactive power (Q), voltage (V), current (I), and V-I trajectories) and/or frequency-based features (Discrete Fourier Transform (DFT) and Fast Fourier Transform (FFT). In regard with these features, others can be obtained, such as complex power and apparent power. Frequency-based features are preferred when a high sampling rate is considered [14].

- The third layer is the communication network layer, which enables integration and communication among different devices. The technology adopted for load monitoring depends on the location of the server from which data is sent. In the context of smart homes, the communication layer is commonly referred as home area network (HAN). Within this architecture, data servers are often located at the edges of the network. Thus short-range wireless communication is preferred where the most popular standards are Bluetooth, WiFi, and Zigbee [15].

- The middleware layer mediates the interaction between IoT devices and software applications. Since computational requirements for this layer are very high, most referenced solutions appear in the domains of cloud and fog computing (e.g., VM-based: MagnetOS and TinyVM, databased: SINA and TinyDB; service-based: LinkSmart, SenseWrap, FIWARE and AutoSec, and Fog node-based: EMCP and eclipse Kura). Therefore, this layer acts as an architecture abstraction between the user interface and all deployed devices. Functional requirements, including data management, data storage, big data analysis, real-time data analysis and deep data analysis based on AI should be considered [30].

- At the top of this architecture lies the application layer. It refers to specific services dedicated to users. Thus, this layer defines a variety of applications in which ILM solutions may be performed. These solutions feed from the energy usage of appliances that are separately measured by submeters. For this, appliance-level load data need to be directly labeled [15].

This thesis focuses on the application layer since it is the hardcore of load monitoring. It is necessary to carry out a system that first can label appliances, and then this information may be applied in useful smart grid applications, e.g., activity recognition, demand response programs, load planning, etc. However, the rest of the proposed architectural layers are considered when designing the load monitoring system. Appliances are supposed to be connected to smart plugs with send the samples to a cloud server, in which the appliance recognition and further processing take place.

3.2 Appliance Recognition System

To implement many applications of ILM, first, it is necessary to identify the load connected to sensing devices. This process is commonly known as ”appliance recognition”. The logical architecture of the **appliance recognition system** is shown in Figure 3.2. A fixed-sized window of raw data, expressed in the form of: *timestamp, active power* inputs the pre-processing block. Here, data is divided into groups of samples that are different from a stand-by signal to build a vector of ten features representing distinguishing characteristics of power signatures provided by sensors. These features are described as follows:

1. Maximum power value.
2. Minimum power value.
3. Mean power of nonzero values.
4. Number of samples equal to zero.
5. Number of samples with power less than or equal to 30 W.
6. Number of samples with power between 30 and 400 W.
7. Number of samples with power between 400 and 1000 W.
8. Number of samples with power greater than 1000 W.
9. Number of power transitions greater than 1000 W.
10. Number of power transitions between 10 and 100 W.

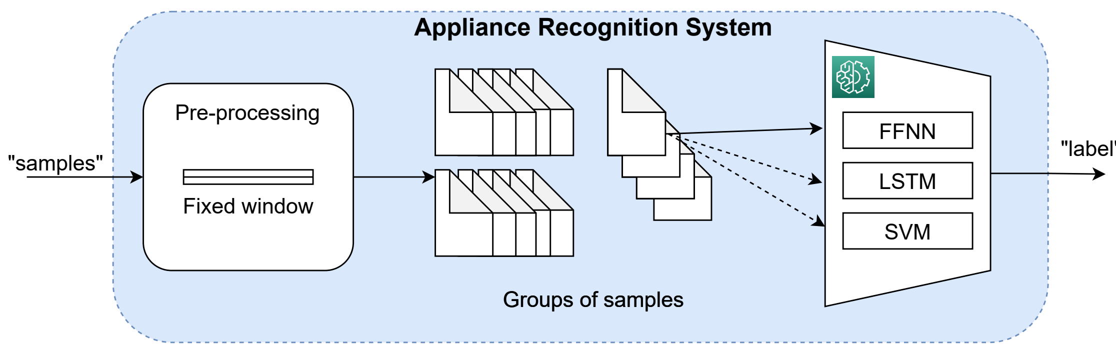


Figure 3.2: Proposed appliance recognition system.

The size of each group was set to 105 samples, and to analyze these power measurements from the first nonzero sample, as suggested in [13], allows that those devices with a long duration, such as washing machines or dishwashers, can be represented with a full-length load profile. Three different models were tested for ML-based classification, including two neural networks: an FFNN, an LSTM, and an SVM classifier was implemented instancing the SVC class provided by the `scikit-learn` library referenced in [83].
In those three cases, the model was trained with the same number of training samples in each class. After every device is identified, these labels along with the timestamps of every

sample are used to recognize ordinary activities of consumers. The specific algorithm implemented for this process will be further discussed.

3.2.1 Feed Forward Neural Network classifier

The proposed feed forward neural network (FFNN) classifier architecture is represented in Figure 3.3. It consists of 10 input neurons as the same number of features are extracted from sensor data. Then, two hidden layers of 500 and 100 neurons, respectively, alternate with a dropout layer that reduces the overfitting and achieves higher accuracy. The number of neurons in the output layer depends on the number of classes into input data that will be classified. For the system implementation, only five different appliances are considered; therefore, the output layer has only five neurons.

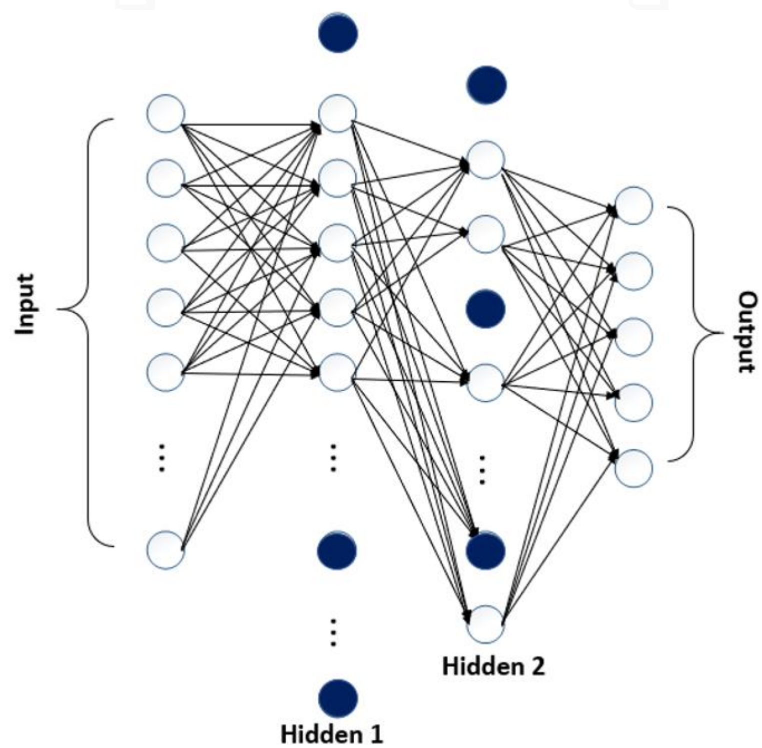


Figure 3.3: Proposed FFNN classifier architecture. The model has 10 input neurons, two hidden layers of 500 and 100 neurons, respectively and 5 output neurons. Dropout has been set with 0.5 probability. The off neurons have been represented in dark blue.

3.2.2 Long Short-Term Memory classifier

The proposed long short-term memory (LSTM) classifier architecture model can be seen in Figure 3.4. Similar to the FFNN classifier, the number of cells in the input and output layers is 10 and 5, respectively, since the feature vector length is 10 and the number of target classes is 5. Then, an intermediate dropout and another LSTM hidden layer were included to improve system generalization and to ensure capturing non-linearities of input data.

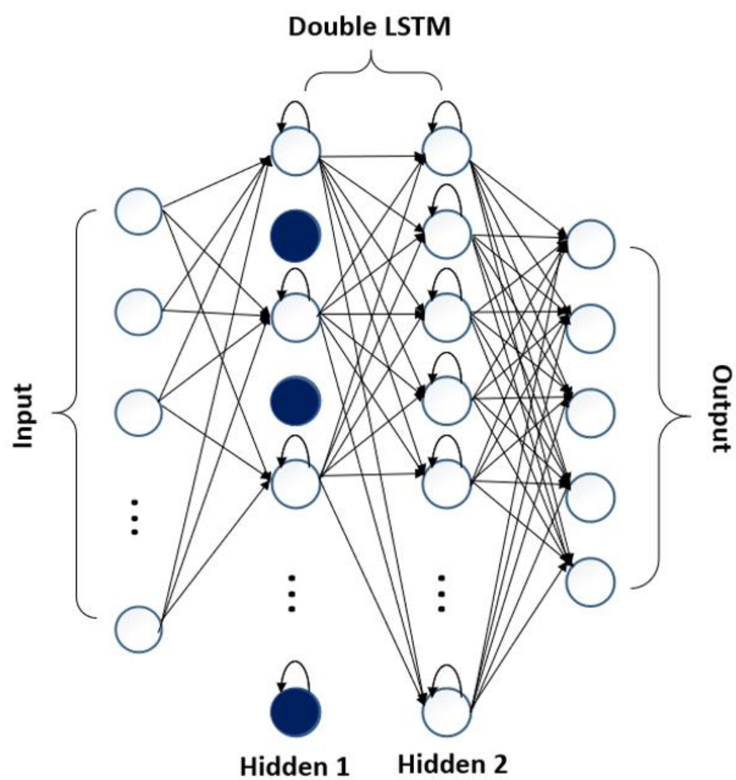


Figure 3.4: Proposed LSTM classifier architecture. The model has 10 input neurons, a double LSTM hidden layer of 200 cells each. Dropout has been set for the first hidden layer with 0.5 probability. The off neurons have been represented in dark blue.

3.2.3 Support Vector Machine classifier

The classifier was implemented by instancing the class available in [83], with the default 'rbf' as kernel function. Kernel function aims to take input data and to transform it into the required form.

3.2.4 Target appliances

Target appliances were selected considering the daily activities that could be inferred from its use. To identify various activities, a set of finite state machines (FSMs) appliances was analyzed, but the same analysis can be performed for every device in a household. The only requirement is to attach a sensor node next to each target device. Selected appliances, along with some useful metadata available in UK-DALE dataset are summarized in Table 3.1. In this work, only appliances from House 1 were used for training.

Table 3.1: Selected target appliances

APPLIANCES	TYPE	ROOM	YEAR OF PURCHASE
Hair dryer	FSM	Bedroom 1	2013
Washing machine	FSM	Utility	2007
Oven	FSM	Kitchen	2000
Dishwasher	FSM	Kitchen	2007
Iron	FSM	Bedroom 1	2006

3.3 Experimental results

This section discusses all experiments and results obtained from the designed system. First, the three ML-based classifier models are compared in terms of accuracy, precision, recall, and F1 score, and then a sensitivity analysis is performed considering different groups.

3.3.1 Experiments

3.3.1.1 UK-DALE dataset

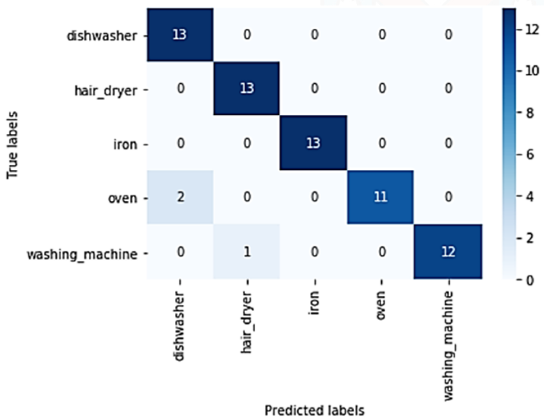
To perform every experiment, real-time data is needed. Since no customized or proper data is available, the United Kingdom-Domestic Appliance Level Electricity (UK-DALE) dataset [84] was employed. It contains aggregated and disaggregated appliance data from five houses in London, England, over several years. The dataset has two types of resolution data available: 6s and 1s. The data is stored in CSV files. The first column is an UNIX timestamp, and the rest can vary depending on the resolution data used. For the 6s data, the second column in each `CSV` file is a non-negative integer that records active power from the individual appliance. The data gathered were obtained from smart plugs attached to individual appliances [9] to measure their energy consumption. Only House 1 was considered during training, mainly due to the fact that each house in the dataset has a different number of appliances, thus identifying different daily activities, House 1 offers more possibilities. The same appliances from House 5 were used to test the performance in front of new data. The results for this experiment will be discussed later in this section.

3.3.1.2 Classifier models

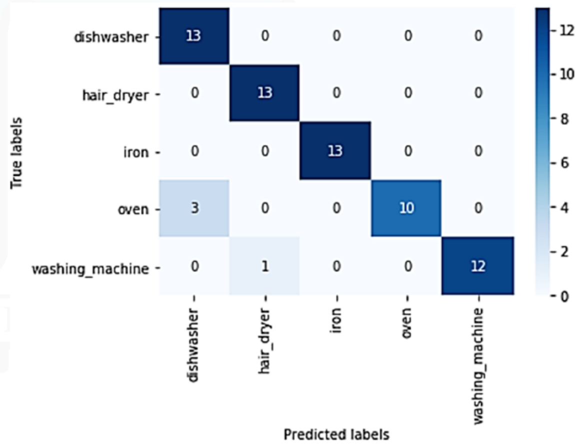
As previously mentioned, three different classifier models were employed: a vanilla neural network (FFNN), an LSTM network, and an SVM classifier. The three models were implemented in Keras running over Tensorflow. Random parameters were set to initialize the system, and depending on results, they were further readjusted to finally obtain the definitive configuration detailed in the previous section. The confusion matrix for each of the three classifiers is presented in Figure 3.5. As it can be seen, the best results are obtained with the FFNN classifier having only three misclassified samples. A higher number of incorrectly classified samples are obtained with the SVM classifier. LSTM classifier achieved a slightly lower accuracy rather than FFNN, but still, it is above 0.9 using less parameters.

Table 3.2 shows a comparison of the results obtained for each class, with the three classifiers in terms of accuracy, precision, recall, and F1-score. For these three cases, misclassifications are related mostly to hair dryer, dishwasher, and oven. For the SVM classifier, the iron obtained the best results as it returned a 100% precision, recall and F1-score. As for the LSTM, the best results regarding precision, recall and F1-score can vary depending on the model. Higher precision is obtained for the iron, washing machine, and oven; however, lower recall is attributed to the latter, achieving a 0.86957 F1-score for this appliance. In the case of the FFNN classifier, in all cases, F1-score was above 0.9. Given these results, the FFNN was stored and further used in the activity recognition module.

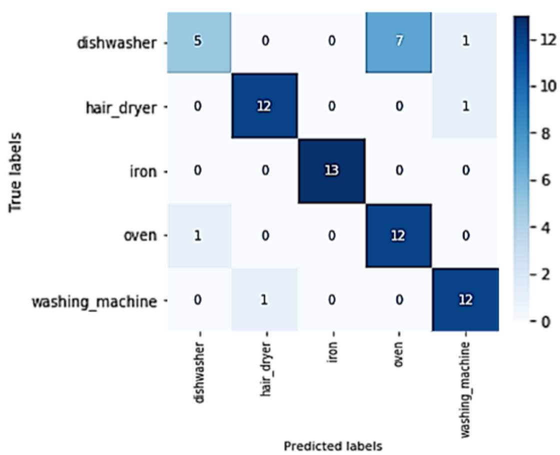
A second experiment was performed to test the model generalization in front of new data. The same five appliances were selected from a different house, in this case, House 5. The confusion matrix for this experiment is shown in Figure 3.6. As shown this figure, it is evident that there is a decrease in all metrics, meaning that if the system is implemented in different houses, it will be necessary to collect data and then retrain the classifier. In this scenario, the accuracy decreased to 0.61.



(a) Confusion matrix for FFNN classifier.



(b) Confusion matrix LSTM classifier.



(c) Confusion matrix SVM classifier.

Figure 3.5: Confusion matrix for the three implemented classifier models.

Table 3.2: Evaluation metrics for the five classes using FFNN, LSTM and SVM.

CLASSES	FFNN			LSTM			SVM		
	PRECISION	RECALL	F1-SCORE	PRECISION	RECALL	F1-SCORE	PRECISION	RECALL	F1-SCORE
Dishwasher	0.87667	1.00000	0.92857	0.81250	1.00000	0.89655	0.83333	0.38462	0.52632
Hair dryer	0.92857	1.00000	0.96296	0.92857	1.00000	0.96296	0.92308	0.92308	0.92308
Iron	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
Oven	1.00000	0.84615	0.91667	1.00000	0.76923	0.86957	0.63158	0.92308	0.75000
Washing machine	1.00000	0.92308	0.96000	1.00000	0.92308	0.96000	0.85714	0.92308	0.88889
Accuracy	0.95385			0.93846			0.83077		

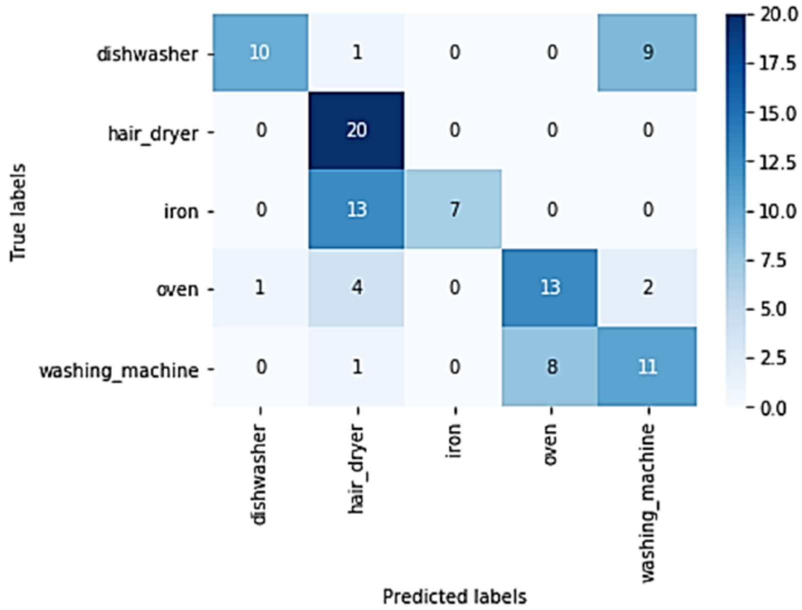


Figure 3.6: Confusion matrix for inference on House 5 of UK-DALE dataset.

3.3.2 Sensitivity analysis

A sensitivity analysis was performed to analyze the impact of the group size on the accuracy of each model. Figure 3.7 shows the results corresponding to this test. Three different values of the group size: 50, 75, and 150 were set and compared to the initial 105. From the graph in Figure 3.7, it can be seen that if the group size is decreased, the accuracy is also diminished, but more samples will be obtained for training. On the contrary, if the group size is incremented, it can either improve or decreased depending on the model employed. For the LSTM classifier, with a 150-group size, no misclassifications are obtained achieving a 100% accuracy,but with a lower amount of training samples, which could not be beneficial for its behavior in front of unseen data. In all cases, the algorithm can recognize simultaneous activities within the analyzed time window. It is also important to note that simultaneous activities can occur while neither is based on the absence of activations (i.e., *sleeping*, and *unoccupied*).

3.4 Discussion

Based on the results obtained from different experiments, the system performance is satisfactory. For a certain input window, groups of samples are gathered, and features are

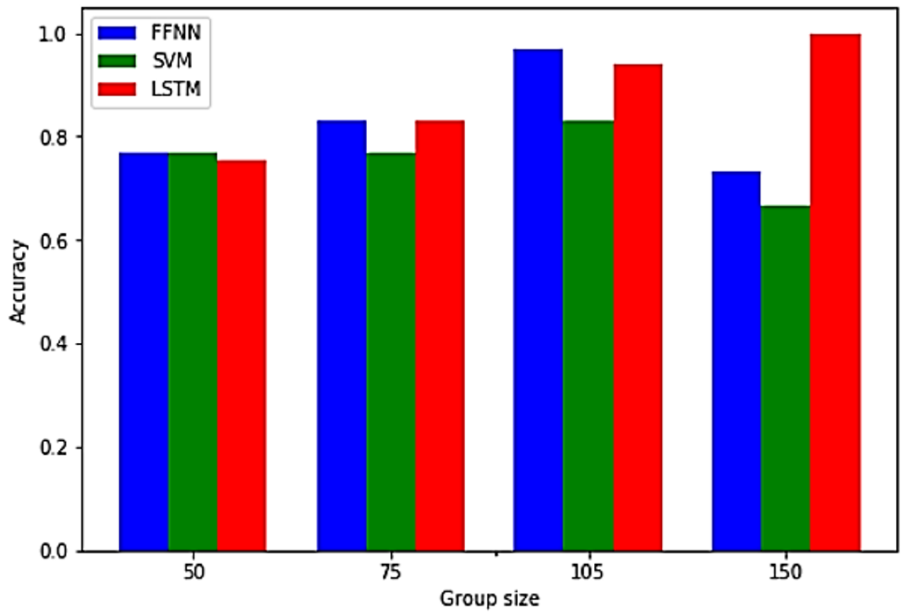


Figure 3.7: Accuracy variation in terms of group size. Four different sizes were tested: 50, 75, 105, 150.

extracted from each of them to be later labeled and associated with an activity executed by house occupants. Although the system does not work in real-time, it can identify simultaneous activities as it analyses a sequence of feature vectors independently from the timestamps. To be used in a different household, the classifier should be first retrained. This happens since houses may have the same types of electronic appliances, but their consumption can vary depending on the vendor or appliance model. Feature extraction influences the fact that different appliances can be modeled as general, not depending on the type of load (e.g., ON/OFF, FSM), which means that a TV, a kettle, or a vacuum cleaner will be analyzed in the same way. With a variation in group size, accuracy will also be different and some modifications in the model will be needed to achieve a higher score. When comparing the proposed system with the state-of-the-art techniques, in [13], among target appliances described by authors were a washing machine, an iron, a microwave oven and a dishwasher. In contrast, they run different experiments to validate the results using real-world data collected in an initial phase of a trial of the Energy@home system. The evaluation metrics in both cases (this thesis and [13]) do not offer a fair line of comparison since in the case of [13] the accuracy is represented by an average of different experiments. Still, it is possible to state that the proposed system offers competitive results in terms of accuracy compared with a similar approach that uses the same features to extract.

4 | ADL Classification

Human activity recognition can provide valuable information regarding energy consumption understating, insights about household occupants' behavior, which have several applications in energy domain and other fields, as it is shown in [9] and discussed in section 2.5.

4.1 ADL Classification System

Based on the requirements stated by the proposed IoT architecture shown in Figure 3.1, an *ADL classification System* is presented. The general ADL classification system is illustrated in Figure 4.1. Before performing ADL classification, it is necessary to recognize appliances properly. In that sense, the appliance recognition module proposed in section 3.2 is used. The machine learning classifiers assign a label to the collected appliance signatures. This is relevant since supervised learning has allowed the correct generalization in front of unseen data [19, 73]. Feature extraction block provides a vector of ten features which extract individual characteristics from each sample (e.g., the shape of the consumption profile, maximum power value and number of transitions). The ML-based block is presented as a black box, meaning that it is possible to implement one of the three ML models (FFNN, LSTM or SVM). The output of this system will be the target appliance class; in other words, it is the type of each appliance (e.g., a kettle, a boiler or a washing machine). Once labeled, an ADL classification algorithm must be deployed to infer activities based on the given label and the sample timestamp. This information is sent back to the user who can benefit from it in order to achieve energy efficiency.

Such a system allows to identify the daily activities of house occupants in a simple manner, which can be useful for various applications. In more specific details, the

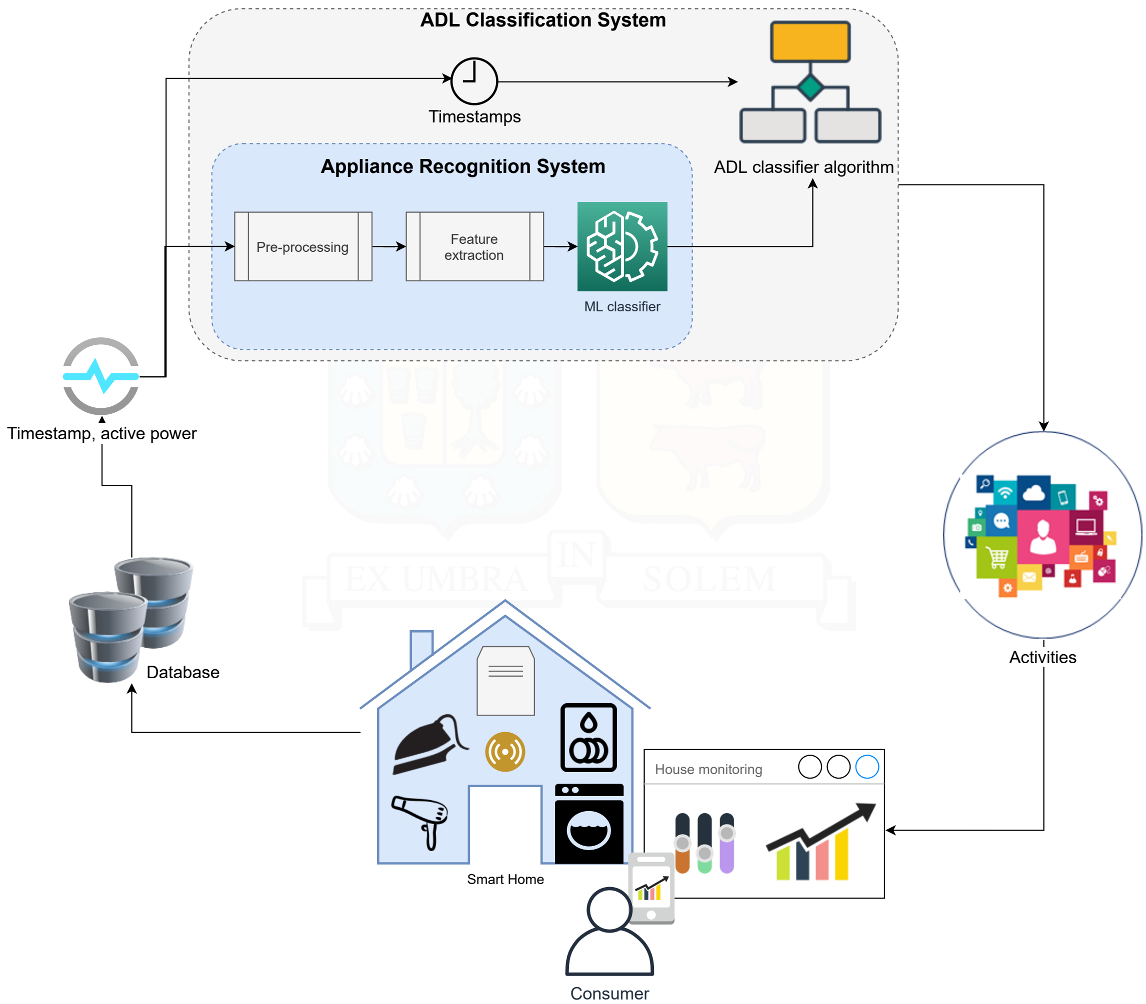


Figure 4.1: Proposed framework for load monitoring and activity recognition system.

advantages of the proposed system may be analyzed from three points of view:

- **Architecture:** As part of an IoT infrastructure, this system can be coupled with different home applications that contribute to the implementation of a sustainable smart grid and more efficient energy usage.
- **Classification:** This process requires to provide a label to each sensor. This label corresponds to the device or appliance connected to the sensor. Machine learning models, such as FFNN, LSTM networks and SVMs allow the labeling of the data and they contribute to the correct classifier generalization, which implies a proper performance regarding unseen data, also removing the need for manually set a label to each sensor.

- **Consumers:** They will be aware of their electrical consumption and activities and act accordingly to use energy efficiently.

4.1.1 ADL classification algorithm

The proposed ADL classification algorithm maps each ADL according to a set of criteria based on appliance usage: their power consumption and timestamps when they are switched on. Following this algorithm, it is possible to detect a total of eleven ADLs:

- *Washing dishes after breakfast*
- *Washing dishes after lunch*
- *Washing dishes after dinner*
- *Baking food for breakfast*
- *Baking food for lunch*
- *Baking food for dinner*
- *Ironing*
- *Drying hair*
- *Doing laundry*
- *Sleeping*
- *Unoccupied*

Sleeping and *unoccupied* ADLs are identified by an absence of detections of major appliances during hours when a householder is most likely to be "asleep" or "out of the house" during the night or day hours, respectively. Continuously variable and permanent consumer devices (low power draw appliances) are, according to [11], extremely difficult to recognize since they can exhibit different patterns significantly, depending on their usage, or they can be regarded as a sub-class of ON-OFF appliances. In the case of the proposed algorithm, it is assumed a scenario having only five FSM appliances, and therefore, no other loads are considered when there is no activity.

The pseudo-code for the classification algorithm is represented in Algorithm 1. The proposed algorithm analyzes a time-window of sensor data. After the classifier model for appliance recognition is loaded, active power and sample timestamps of every target device are ridden. Then, several groups of samples are formed and from each one of them, features are extracted. The array of features obtained is stored in a CSV file to be further inputted to the ML classifier model. Depending on the size of the time-window, there will be the total amount of groups formed. The group size can also be modified, but for training, it was set to 105 samples. From each vector of features, an activity with its corresponding date and time is returned. If no activity is detected (it could be a vector of zeros or a vector of a standby mode), then no activity will be registered. In these cases, if the timestamps indicate a night hour, then the house is classified as *sleeping*; otherwise, *unoccupied* is returned. The criteria used to classify ADLs should be customized to each individual household since it depends on target appliances, and the algorithm could be adapted to the most common daily activity.

4.2 Building a consumer profile

Figure 4.2 shows the detected activities obtained from samples of House 1 of UK-DALE dataset in three different days. The datetime shown in the x axis corresponds to the timestamps of the first sample of every 105-samples group formed. As features are extracted in groups of 105 samples, there are fewer activities detected per day. However, this analysis allows to build a profile of the householder which details its most frequent activities. For example, it is possible to state that consumers have dinner around 8:30 pm and have breakfast around 8:10 am. Therefore, the usage records of the appliance being monitored can benefit a series of useful smart grid applications like demand response and load planning. If more data are collected, then a more descriptive profile will be obtained.

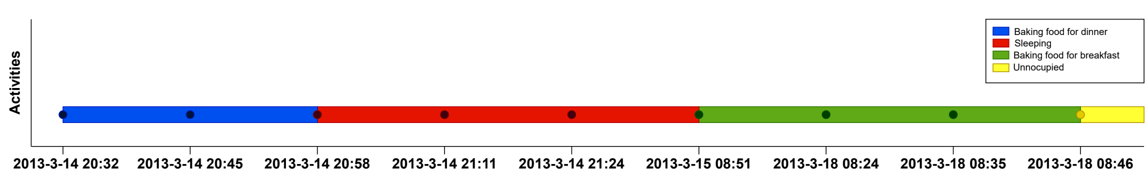


Figure 4.2: Detected activities from samples of three different days in House 1 of UK-DALE.

Algorithm 1: Activities of Daily Living Classification

```
set window_size
load classifier model
read sensor data (active power and timestamps)
for len(window_size) do
  if amount_of_samples = len(group_size) then
    feature_vector = feature_extraction(samples)
    store feature_vector and timestamps in array
  end if
end for
store array of feature vectors in inference csv file
load inference csv file
standardize inference data
for each row do
  convert time to ISO format
  if not activation detected and ISOtime = night_hours then
    activity = ”sleeping” with (probability) at (ISOtime)
  else if not activation detected and ISOtime = daily_hours then
    activity = ”unoccupied” with (probability) at (ISOtime)
  else
    prediction = model.predict(feature_vector)
    if prediction = ”dishwasher” and ISOtime = morning_hours then
      activity = ”Washing dishes after breakfast” with (probability) at (ISOtime)
    else if prediction = ”dishwasher” and ISOtime = afternoon_hours then
      activity = ”Washing dishes after lunch” with (probability) at (ISOtime)
    else if prediction = ”dishwasher” and ISOtime = evening_hours then
      activity = ”Washing dishes after dinner” with (probability) at (ISOtime)
    else if prediction = ”hair_dryer” then
      activity = ”Drying hair” with (probability) at (ISOtime)
    else if prediction = ”iron” then
      activity = ”Ironing” with (probability) at (ISOtime)
    else if prediction = ”oven” and ISOtime = morning_hours then
      activity = ”Baking food for breakfast” with (probability) at (ISOtime)
    else if prediction = ”oven” and ISOtime = afternoon_hours then
      activity = ”Baking food for lunch” with (probability) at (ISOtime)
    else if prediction = ”oven” and ISOtime = evening_hours then
      activity = ”Baking food for dinner” with (probability) at (ISOtime)
    else if prediction = ”washing_machine” then
      activity = ”Doing laundry” with (probability) at (ISOtime)
    end if
  end if
end for
return  Activity detected, Probability, Time (ISO 8601 format)
```

4.3 Discussion

With respect to the ADL classification, the proposed algorithm is similar to the one presented by authors in [9]. Both map each ADL according to a set of criteria based on appliance usage, such as power consumption and timestamps when switched on. However, authors in [9] used an NILM approach to recognize appliances; therefore, the system needs to disaggregate the smart meter power consumption signal. The two algorithms associate the lack of use of electrical appliances in a certain time-period, specifically during the night or daily hours, with the *sleeping* and *unoccupied* activities, to represent the period of time when occupants are in sleeping hours or the house is empty, respectively. In the case of [9], authors used a different algorithm per activity inferred while this thesis presents a unified proposal. To compare appliance recognition and ADL classification processes as a general, Table 4.1 describes the two processes concerning input and output of both models.

Table 4.1: Comparison of appliance recognition and ADL classification.

CLASSIFICATION		INPUT	OUTPUT
Appliance recognition	[11, 13, 72, 73] NILM Proposed ILM	Sensor data	Appliance name
ADL classification	[9, 20, 63] Proposed ILM	Appliance name	Human activity

The proposed system shows a scenario with only five appliances, and therefore, no other loads are considered when there is no activity. This criterion is only based on the absence of all appliances' activations. For a practical implementation of the system, this restriction should be modified, and more appliances need to be considered to obtain proper insights about household occupants' behavior. By testing on real data, such as UK-DALE publicly available dataset, it is proven that the proposed classification system performs accurately so that it can be implemented in a practical scenario.

5 | Frameworks for Appliance Recognition

To design a feasible system that can be implemented in a practical scenario, new data and pre-processing techniques need to be incorporated. A testbed is a proof of concept demonstration which allows to assessment of the system performance. It consists of the implementation at a short-scale in a laboratory environment. The practical deployment of a system for a real testbed evaluation presents a considerable higher degree of complexity when compared to simple simulation. Dealing with real-time data causes some new challenging modifications, especially in the pre-processing and early manipulation stages of data. Therefore, some new techniques in the pre-processing stage needed to be incorporated into the system proposed in Chapter 3. The new version becomes an easy-to-use framework for both training and inferences processes, adding a graphical interface which accelerates and facilitates its use. Next subsections will describe and analyze all the above considerations.

5.1 Why to modify the system?

As it was discussed in section 3.4, to deploy the system in a new house or to use new data, first, it is necessary to retrain it. This implies that when real data from laboratory appliances are collected, and the system might be retrained first before.

The appliance recognition module proposed in the previous chapter and illustrated in Figure 3.2 was trained with the record of five appliances in House 1 of UK-DALE dataset. This means that the module is not standardized, and if applied using a different data structure, several adjustments have to be made, especially in the feature extraction. In the UK-DALE dataset, records are structured in two columns of float values. One corresponding to the sample timestamp, and the other one is belonging to the active power. For feature extraction, the data were divided into groups and then used in the same proportion of feature vectors for each training class. This balance prevented the need to work with imbalanced classes, which would impose a great challenge in the performance of the machine learning classifier. As a result, the training was performed using as many feature vectors per class as the number of vectors in the minority class, leaving information behind. Since only the appliances activation are considered for extracting the features, in a practical scenario, it might conduct to counterproductive results. In a practical use-case, it is likely that the system has to deal with a different number of samples for each class, thus the current proposal could not handle the imbalance problem. In addition, for a practical use-case, in which the system must be retrained in front of new data, the fact of training with the whole appliance profile dividing it into fixed groups impose a restriction, since it makes the system less capable of working in real-time. Hence, there is a need to make some adjustments, forcing the system the system to be more resistant to real-time processing. Another big constraint imposed by the current pre-processing stage is the extracted features. The majority of the features depend on the scale of the data, i.e., *how many samples above 30?* This implies that the data need to be delivered in a specific format, which is not possible with some sensing devices.

On the other hand, with the advancement of technology, the smart grid is now supporting many new applications in the distribution power system. Among these applications, electric vehicles are expected to play an important role. The vehicles operation modes adopt a bidirectional energy flow between EVs and the power grid, therefore their integration into the power grid is considered a highly complex task. Many research studies have been conducted to investigate the influence of electric vehicle charging on the distribution system from power perspectives [12, 21, 31]. The impact of electric vehicle charging is expected to be significant in view of power losses, power quality, voltage deviations, harmonics and frequency shift. To overcome the new peak demand with the integration of electric vehicles, solutions include increasing generation capacity, upgrading the existing distribution system infrastructure or considering demand response techniques [85]. Demand response (DR) strategies will play an important role in load shaping in order to prevent the distribution transformer overloading, and load monitoring techniques are vital to develop efficient DR applications. In [12, 21, 31] authors propose event-detection algorithms for NILM systems based on low complexity statistical features. Although they achieve successful simulation results, the efficiency in practical scenarios has not been proved yet. As it was previously discussed, NILM techniques are rather unreliable in these situations. However, EVs are considered major loads in the context of smart homes and their monitoring has to be highly prioritized.

All in all, to make the system capable of working in nearly real-time and to include the monitoring of major loads such as EVs, a new version of the appliance recognition module is proposed. This version emerges as a framework which facilitates not only the pre-processing and training, but the inference too, becoming a flexible system which allows to apply the same principle using data from different sources.

5.2 General Description

Two new frameworks were developed: a *training framework* and an *inferences framework* with aiming at extending the appliance recognition framework presented in Chapter 3. The general description of both frameworks is given in the following subsections. Both frameworks include a graphical interface that easily facilitates to set up the system, giving the user the possibility to configure it by applying different pre-processing and features extraction techniques, as well as choosing among the classifier models.

The result is a flexible and complete system that can be scaled in the future to include more data and as well as many classifier models and pre-processing techniques as desired.

5.2.1 Training framework

Figure 5.2 shows the general composition of the training framework which was implemented in Google Colaboratory (Colab), the upper part of Figure 5.2 shows the dataset configuration. First, it is necessary to input the number of classes, the location of the dataset files, the selection of the target appliances and to choose how to input the activation threshold. If the thresholds are known, then the user (the person training the system) can manually introduce their values. On the contrary, if the manual setup is hard to obtain, thresholds will

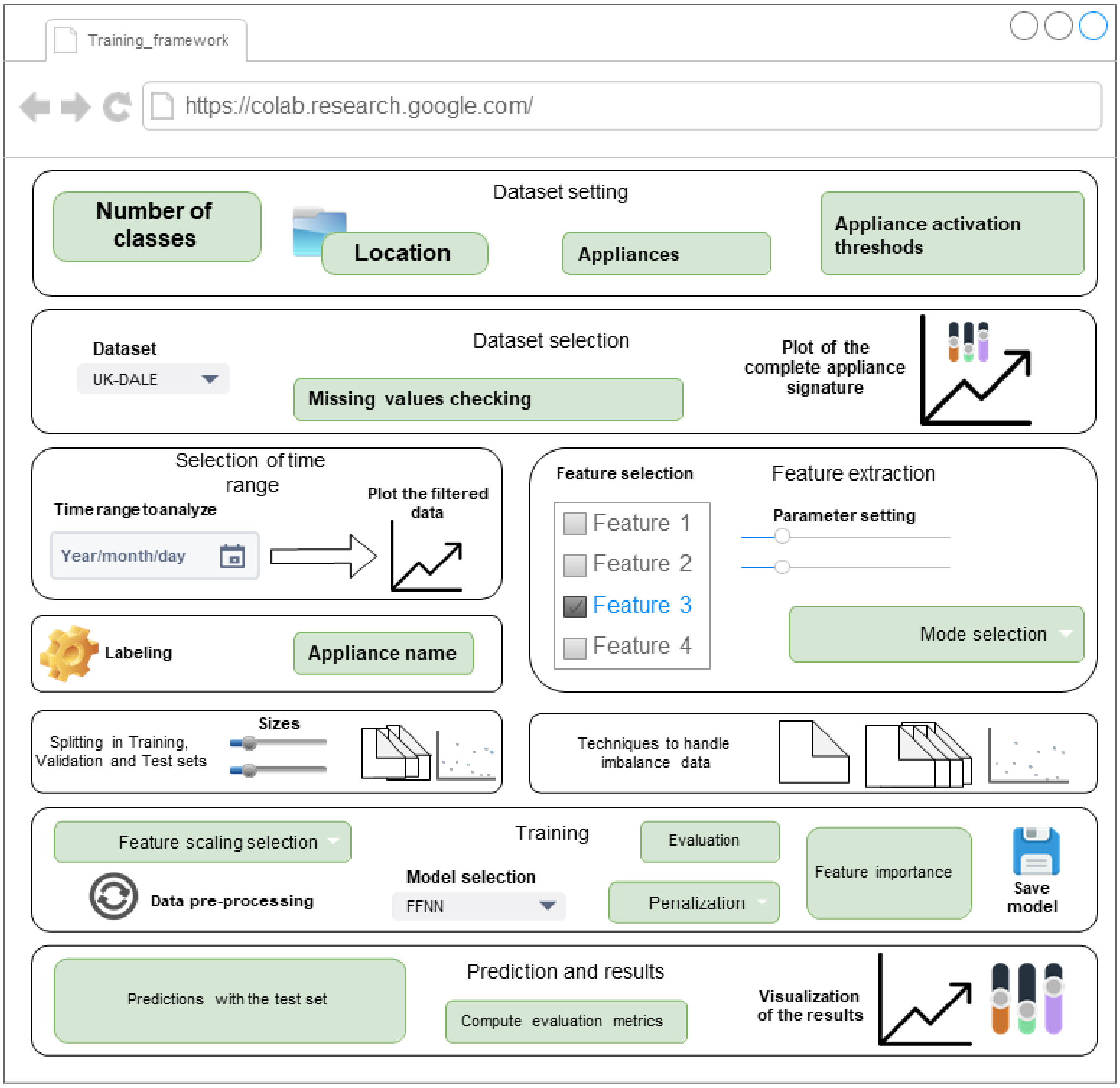


Figure 5.1: Training framework and its basic structure.

be computed automatically. In this case, the only requirement is to introduce a value that represents the limit of the neighborhood of the minimum power measurement registered by the appliances. Then, the thresholds values will be between the global minimum and the limit imposed by the user. The reason to have an activation threshold is to compute the stand-by value of every appliance. Therefore, only the activations or the values of the active power when an appliance is ON will be considered for feature extraction. To compute the activation threshold, first, the minimum is obtained; later, the algorithm looks for all power measurements between the minimum and the minimum plus the limit inputted by the user. After that, the maximum of all the filtered values is chosen as the activation threshold.

Once the initial setting is completed, the user has to choose the name of the dataset to load and optionally to check missing samples in the chosen dataset. The method to fill missing values can also be selected by the user. The possible datasets to work with will be described in the following subsections. At the end of this part, the target appliances signatures are plotted.

The middle part of Figure 5.2 represents the pre-processing and feature extraction. Once the selected dataset is uploaded and activation thresholds are computed, the user can select a time range to analyze, i.e., a subset of the profiles that will be used to extract features and train the classifier models. In the next step, a set of ten possible features can be selected to extract. These features vary from the ones proposed in subsection 3.2 in order to be applied to datasets that differ in format (e.g., scale and structure). The new features represent statistical computations that describe the appliances' profiles. They will be further detailed in section 5.4. One important aspect to remark is that in contrast with the previous version, the user can train to extract the whole set of features or either choose a subset of them. Another big difference is the way features are extracted. Instead of dividing the profiles into fixed groups, now samples will be processed inside a sliding window. This sliding window operates similarly to a 1D-convolutional layer in a convolutional neural network but without the convolution operation. Therefore, the window will have a size from which statistical features will be calculated. The size value is one of the parameters that is required to be provided by the user at this stage, along with the sliding window stride and mode. The stride parameter reveals how far the window should move at each step, and the mode describes what to do when the size of the window is larger than the number of remaining samples. For the latter, the user can choose among `padding`, `no_padding` and `dynamic`. Then, the true labels that correspond with the appliance name, must be provided for the selected appliances. A deeper analysis of all the parameters setting will be also given in section 5.4.

After feature extraction, data have to be prepared to enter the classifier model. In this case, the complete set of vectors is divided into three subsets: *training*, *validation* and *testing*. The user can set the proportion of the dataset to be included in the three subsets, for example, 80% of the feature vectors in the training set and 10% in the validation and test sets, respectively.

The bottom of Figure 5.2 is shown the training configuration. The user can choose between `standardization` and `normalization` as the feature scaling method to apply. Feature scaling is the process of converting all the features into a given range. Depending on the operation selected, the limits of this range will be established [86]. Details of both `standardization` and `normalization` will be given in section 5.4. To complete the pre-processing stage, the true labels are converted into a numerical value.

Then, the user has to decide which classifier model to train, for example, the feed-forward neural (FFNN) network, the long short-term memory (LSTM) or the support vector machine (SVM) classifier already designed in section 3.2. The only modifications introduced in this case is that the user can apply a penalization or so-called kernel regularizer to this model. In this regard, three options are available: **L1 norm**, **L2 norm** or a combination of both (**L1_L2**). A penalization can be helpful in the presence of imbalanced data. Regularizers allow to apply penalties on layer parameters during optimization. These penalties are summed into the loss function that the network optimizes [54].

Once the selected classifier is trained, the model is evaluated using the validation data. In addition, a new tool was deployed to assess the performance of the model given the chosen features. It is a process called **feature importance** and it allows us to understand how the features in our model contribute to prediction. Now, it is possible to know if a given feature has more or less relevance to the system behavior, and in the negative case, to counteract it. The best model configuration can be saved to make future inferences.

The last part of training is configured to predict with the test set. To assess the system' generalization, a set of metrics are used. These are the **precision**, **recall**, **F1-score**, **cohen's kappa coefficient** and **confusion matrices**. The cohen's kappa coefficient is a new metric with respect to the previous version and it is basically the classification accuracy normalized by the imbalance of classes in data.

5.2.2 Inference framework

The inference framework is used to test the system generalization which predicts with unseen data, similar to what is expected in a practical use-case. The general structure of the framework is shown in Figure 5.1. In the same way, as the training framework, first, it is necessary to configure the dataset to use, entering the number of classes (equivalent to the number of appliances), the appliances to use, the location of their profiles and their activation thresholds. In this case, as the same appliances are used, and their activation thresholds are computed during training, the user (i.e., the person making the predictions) only needs to input the obtained value to make inferences. Then, the user must select the dataset to analyze and optionally check missing values.

Similar to the training framework, the next step is the pre-processing and feature extraction.

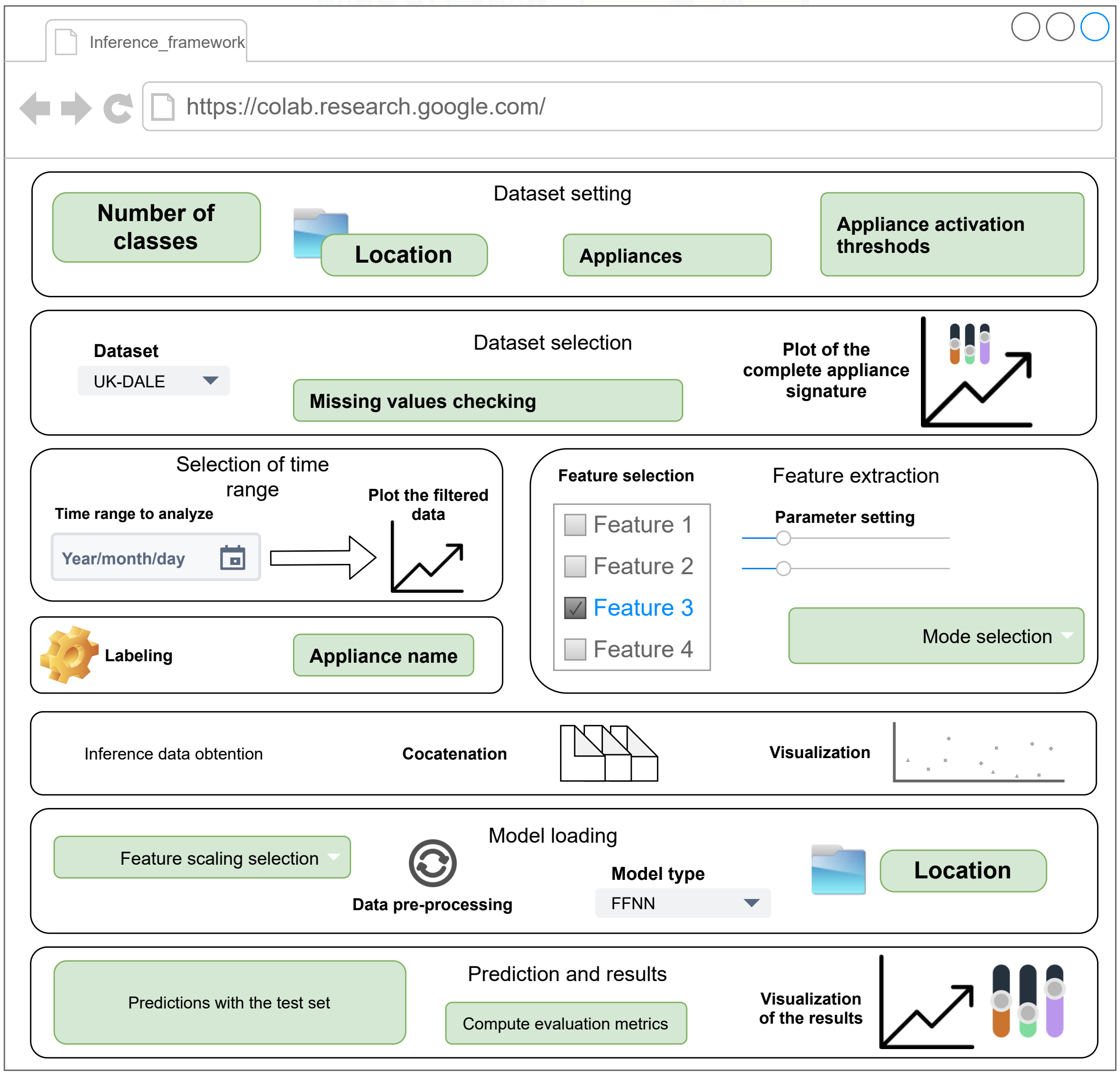


Figure 5.2: Inference framework and its basic structure.

The user needs to provide a time range to infer, it can either select the same or a different time range from training, although predicting with the same data used in training has no meaning when assessing the system generalization. Next, there are the feature selection and the parameter setting for the extraction. Obviously, this is done identically to the training. Inside a sliding window, configured with the size, stride and mode determined by the user, the set of selected features (among a group of ten) are computed. After finishing the feature extraction, the true labels need to be provided to the feature vectors.

Since there is no need to have a training or test set for inferences, the next step is to concatenate the resultant vector in a unique dataframe. Once completed, it is only required to load the model and prepare the data to input the trained classifier model (pre-processing). In that sense, the user has to provide the model's location and to select the feature scaling function to apply and as well as the model type (FFNN, LSTM and SVM).

The last step, shown at the bottom of Figure 5.1, is to perform predictions and to obtain the evaluation metrics. These are also the **precision**, **recall**, **F1-score**, **cohen's kappa coefficient** and **confusion matrices**.

5.3 Datasets

Two datasets are considered in the frameworks: the one already presented in chapter 3 UK-DALE dataset [42] and the Pecan Street Dataport [34]. Both datasets are similar as both provide individual power consumption and aggregated signal for a set of houses in a certain period of time. However, they differ in structure, in appliances included, in the scale of the measurements, in features (active power, reactive power, etc.) and in sampling frequency.

As it was previously discussed, the UK-DALE dataset involves the consumption profile of five houses in the United Kingdom (UK). It is organized in a hierarchy of file folders in which each house has its own folder, and inside, there are separated files for each appliance in the house and their aggregated power consumption. Each file is structured in two columns: one for samples' timestamps and the other one for active power measurements. The data used for training were collected at a 6 s sampling frequency, and the same house (1) and appliances (washing machine, iron, oven, hair dryer and dishwasher) data; of the former version were used for training.The general structure of the UK-DALE dataset is represented in Figure 5.3.

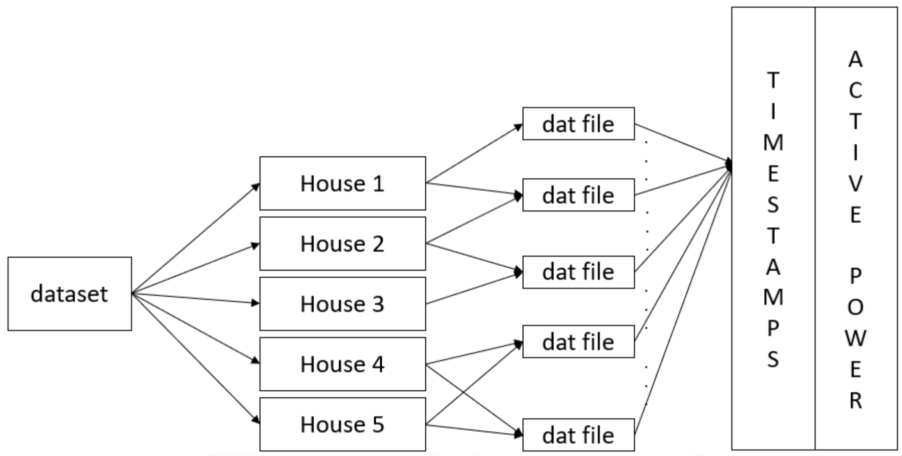


Figure 5.3: General structure of the UK-DALE dataset

In contrast, the Dataport dataset is the world's largest residential energy and water research database which has been used by a large number of researchers. Dataport contains unique, high-resolution data collected from instrumentation that Pecan Street installed in approximately 1000 homes, from which about 750 are located in Texas, 50 in Colorado, 50 in California, 100 in New York, and 50 in other states. Over the course of 2020, an additional number of 100 homes in the New York State and 100 homes in California have been added to Dataport [34]. The complete database is not freely available, some fees have to be paid depending on the amount of data a project demands. However, Pecan Street provides access to static time-series datasets (1-second energy, 1-minute energy and 15-minute energy) for 25 homes in three regions (New York, California, Austin). The provided data on New York region contains 6 months of data with 100% completeness across all intervals for individual homes, and 99% for California and Austin.

The objective pursued by including these data in the appliance recognition framework is to include the electric vehicle among the target appliances, as it is considered a major load in a smart home. Although plug-in vehicles have not been adopted at a scale, governments, utilities and automobile companies, like Tesla [87], are corroborating the opportunities that arise from reduced emissions and gasoline consumption. Apart from its many benefits, the inclusion of electric vehicles into the power grid creates along serious challenges to utilities, as this load adds stress on the power grid, which might cause voltage instabilities and blackouts [16]. The vehicles' charging consumption is assumed as the analog load introduced by connecting another house into the power grid. Therefore, load monitoring can massively contribute to avoid overload in the grid, overcoming the aforementioned challenges.

The free data files of Dataport can be heavy to process. Thus, to efficiently use the available computation resources, a subset of the dataset was used for training the proposed frameworks. The data of three appliances (electric vehicle, oven and microwave) from House 2335 of Austin's residential data was filtered in a separate `csv` file. The resultant file contains a column for the time and date of the collected samples, and another column for each appliance profile. In this case, the sampling frequency of data is 1 minute.

5.4 Pre-processing

The pre-processing stage represents all operations conceived to prepare the data to input the classifier model. Once the appliances' data are read, the feature extraction, some tactics to combat the imbalance in the data, dataset splitting and feature scaling take place. All these operations will be described below.

Optionally, when the data are loaded, the user can check the existence of missing values in the profiles. In this case, there are the available methods to fill the gaps are:

- `pad`/`ffill`: propagates last valid observation forward to next valid backfill.
- `bfill`: uses next valid observation to fill the gap.

The complete procedure in the training pre-processing stage is represented in Figure 5.4. Inside a sliding window, with size, stride and mode set by the user, a number of statistical features (10 or less) are extracted. Once the feature vectors for every target appliance profile are obtained, the user has to provide a label for each of them that corresponds with the appliance name. Then, to have a better understanding of the classes distribution, the number of vectors in each class is plotted. If the number of vectors in every class is not the same, optionally, the user can apply a set of techniques to handle the dataset imbalance. Applied these tactics or not, the next step is to split the dataset into three subsets that will be use for training: the `training`, `validation` and `test` subsets. Next, an operation to take the values to the same scale is applied to the three subsets previous to input in the classifier model. When the training is completed, the user can evaluate the impact of the selected features in the system predictions by performing a process called feature importance. All these steps will be described in the subsections below.

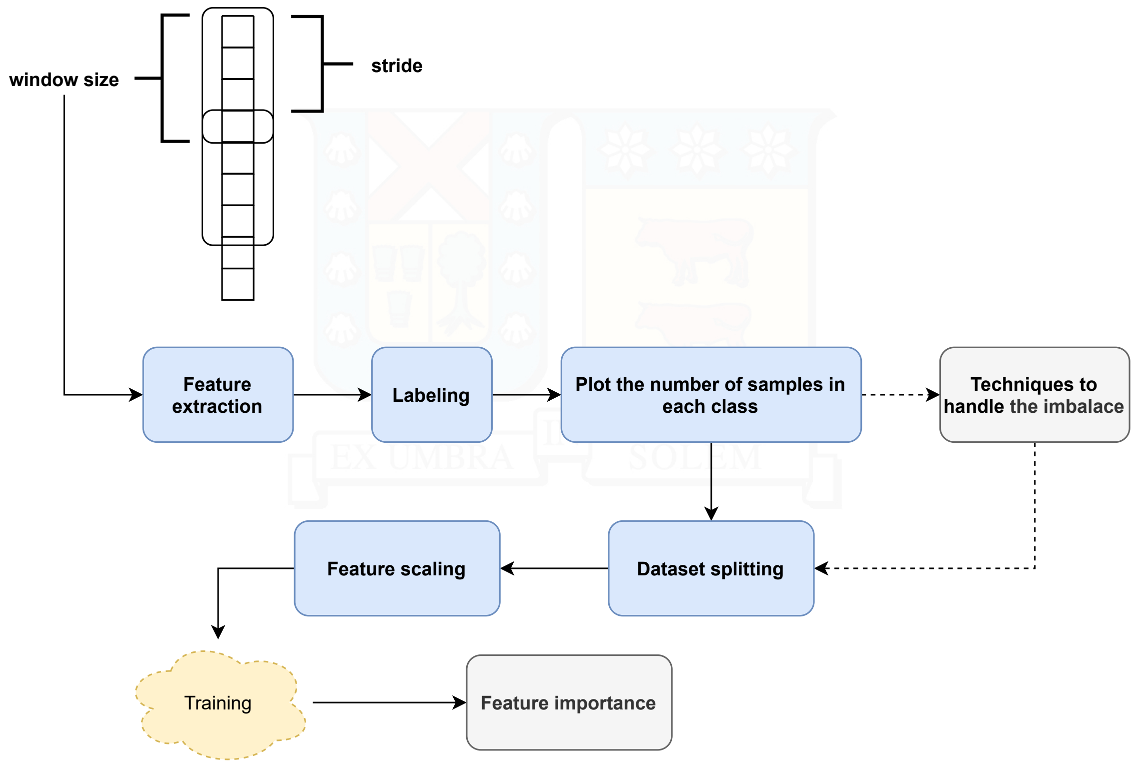


Figure 5.4: General workflow of the pre-processing stage

5.4.1 Feature extraction

A a new dataset, in addition to the UK-DALE, is considered in the appliance recognition frameworks, and both datasets are different in structure and format. The same feature vector computed in section 3.2 cannot be obtained. This is mainly due to the scale of the power measurements recorded in both datasets. However, a 10-feature vector showed to be more than enough to capture the appliance behavior and to make the classifier model distinguish one from the other. Therefore, a set of ten new features was formed. It reflects a series of statistics behind the appliance power measurements. These features are:

1. *Minimum power value* (also included in the previous section)
2. *Maximum power value* (also included in the previous section)
3. *Mean power value* (also included in the previous section): is the average or the most common value in a set of collected numbers [88].

4. *Standard deviation*: is the average amount of variability in a dataset. On average, it shows how far each score lies from the mean [88].

5. *Skewness*: is the third statistical moment and refers to a distortion or asymmetry that deviates from the symmetrical bell curve, or normal distribution, in a set of data [88].

6. *Kurtosis*: the fourth statistical moment defines how heavily the tails of a distribution differ from the tails of a normal distribution. In other words, kurtosis identifies whether the tails of a given distribution contain extreme values [88].

7. *Coefficient of variation (or relative standard deviation)*: is a statistical measurement computed as the ratio of the biased standard deviation to the mean. The metric is commonly used to compare the data dispersion between distinct series of data. Unlike the standard deviation that always has to be considered in the context of the mean of data, the coefficient of variation provides a relatively simple and quick tool to compare different data series [88].

8. *Median absolute deviation (MAD)*: computes the median over the absolute deviations from the median. It is a measurement of dispersion similar to the standard deviation but more robust to outliers. The presence of outliers does not change the value of the MAD [88].

9. *Number of values above the mean*

10. *Number of stand-by values*: based on a given activation threshold, set by the user, this feature counts number of stand-by values previous to the activation and the ones contained in the analyzed window. It serves to measure the size of the appliance activation pulses.

The flowchart of the feature extraction algorithm is given in Figure A.9. It needs to be provided with four parameters in addition to the appliance profile: **stride**, **window size**, **mode** and **activation threshold** of each appliance. The first two (stride and window size) can take integer values ranging from 1 to 50. The mode reflects what to do in the case when the window size (w) is greater than the number of remaining samples (n) in the profile. Mathematically, we know that w>n by:

$$n = length(profile) - i \tag{5.1}$$

where *length(profile)* is the size of the complete appliance profile and *i* is the position of sample being analyzed in the appliance profile. Then, three mode options are available when w>n:

- `padding`: it completes the missing values in the window by simply concatenating zeros at the end of the profiles. This option is set by *default*.
- `no_padding`: the algorithm discards the samples from which w>n. It just analyzes the value where the window fits.
- `dynamic`: it adapts the size of the window to be equal to the number of remaining samples (n). In other words, it makes w=n.

On the other hand, the activation threshold, as it was described before, can be manually configured or either computed looking for the maximum value in a range given by the minimum power measurement in the appliance profile, and a limit which depends on the value inputted by the user. Then, the selected features are computed to finally return a dataframe of feature vectors.

5.4.2 Dataset splitting

Separating data into different subsets is an important part of evaluating data in machine learning models. Typically, the data are split into a training set and testing set; most data are used for training, and a smaller portion of data is used for testing [54].
It is often used an additional set called validation set. This set is a sample of data held back from training the model which is used to give an estimate of model skill while tuning model's hyperparameters. The validation dataset is different from the test dataset that is also held back from the training of the model, but is instead used to give an unbiased estimate of the skill of the final tuned model when comparing or selecting between final models [54].
For training the proposed frameworks, the dataset is split into:

- a **training** set: a set of data used for learning, that is to fit the parameters of the

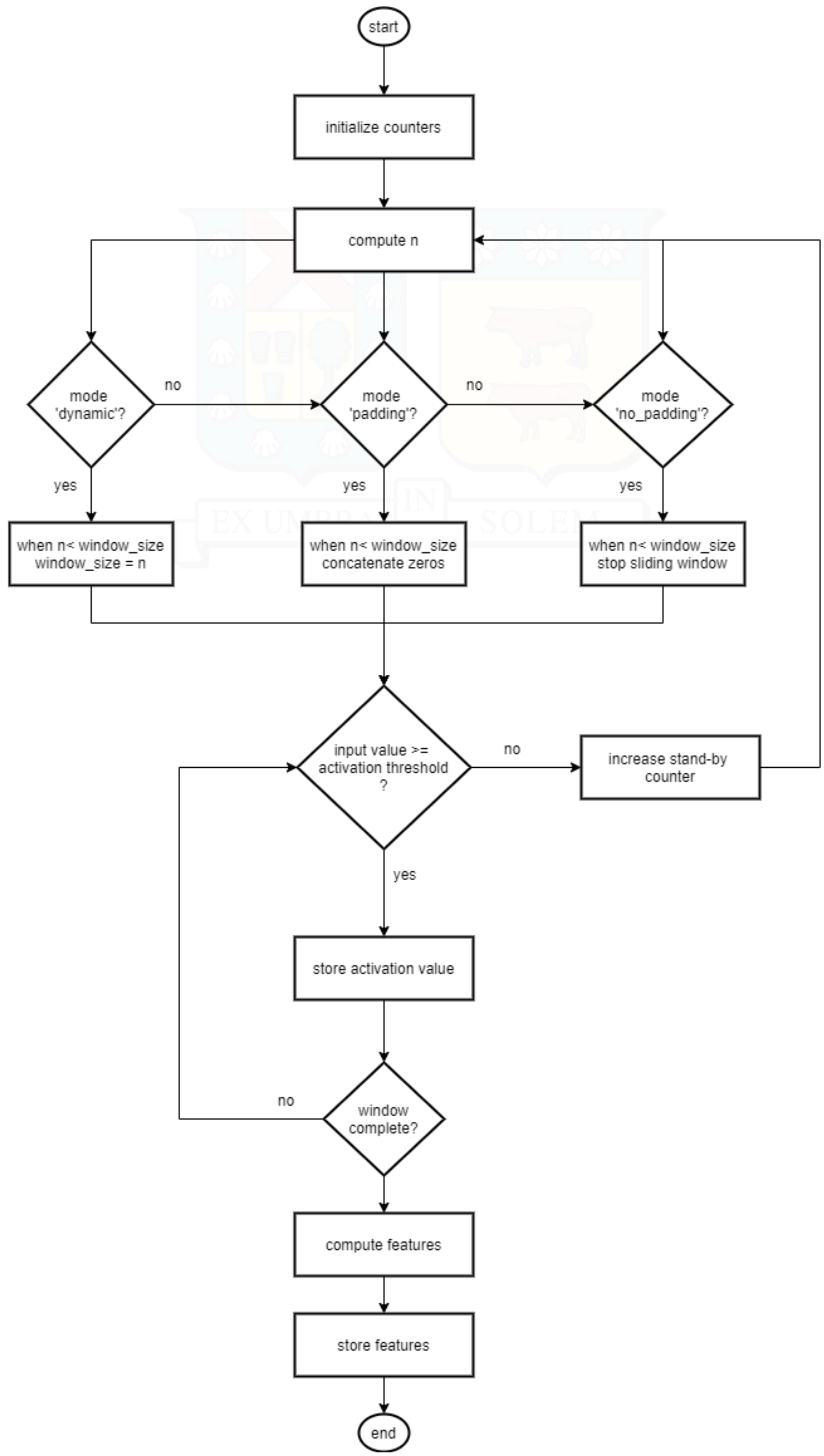


Figure 5.5: Flowchart of the feature extractor algorithm.

classifier.

- a **validation** set: a set of data used to tune the parameters of a classifier, for example, to choose the number of hidden units in a neural network.
- a **test** set: a set of data used only to assess the performance of a fully-specified classifier.

The proportion of data for each subset is determined by the user. In addition, the validation set is useful in the feature selection. This impact on the selected features will be further discussed in the next subsections.

5.4.3 Techniques to handle class imbalance

Since the proposed appliance recognition frameworks only consider the activations (i.e., power measurements when the appliances are on), and the activation pulses can have different time duration. Then, the resultant number of feature vector in each class is different. Hence, we are in the presence of imbalanced data to train.
Data imbalance is a challenging problem which has not got clear solution yet. However, some techniques can help to reduce the impact of this disproportion in the dataset. It is important to highlight that all these tactics are only applied to the training set. The techniques implemented in the proposed training framework are:

- *Drop duplicates*: it executes a method that removes all the duplicated feature vectors in the training set. It can serve as an undersampling procedure to reduce the number of elements in the majority classes (i.e., the classes with a higher number of vectors).
- *Oversampling*: these are synthetically generated samples making use of the Synthetic Minority Over-sampling Technique (SMOTE) algorithm [89]. This algorithm, as its name indicates, generates synthetic samples for the minority classes. It focuses on the feature space to generate new instances with the help of interpolation between the positive instances that lie together.
- *Undersampling*: it reduces the number of samples in the minority class following a certain strategy (i.e., the proportion of samples in each class). In the case of the proposed framework, the strategy has to be included in system as a dictionary. The

implemented algorithm takes advantage of the `RandomUnderSampler` class available in `scikit-learn` [83].

- *Penalized models*: penalized classification causes an additional cost of the model for making classification mistakes on the minority class during training. These penalties can bias the model to pay more attention to the minority class [54]. In the proposed training framework, penalization is performed by setting regularizers to the classifier models. The penalties are added to the loss function that the network optimizes. Three different regularizers are available. The three may be transferred to a layer as a string identifier:
  1. **L1 norm:** a regularizer that applies a L1 regularization penalty. The user needs to provide the coefficient to use.
  2. **L2 norm:** a regularizer that applies a L2 regularization penalty. The user needs to provide the coefficient to use.
  3. **L1_L2 norm:** a regularizer that applies a combination of both L1 and L2 regularization penalties. The user needs to provide the coefficient to use.
- *Cohen's Kappa coefficient*: Accuracy can be misleading when working with an imbalanced dataset. It is the case in which accuracy is high (such as 90%), but the accuracy is only reflecting the underlying class distribution. Including a new metric, such as cohen's kappa coefficient can help describe better the behavior of the system. Cohen's kappa coefficient represents the classification accuracy normalized by the imbalance of the classes in the data. It is based on comparing the concordance observed in a set of data, with respect to what could occur by mere chance [90].

5.4.4 Feature scaling

Feature scaling is a technique to standardize the independent features present in data in a fixed range. It is performed during the data pre-processing to handle highly varying magnitudes or values or units. If feature scaling is not done, then the machine learning model tends to weigh greater values, and to consider smaller values as the lower values, regardless of the unit of the values [54].

The two available techniques in the proposed frameworks are: `standardization` and `normalization`. Standardization takes the values to be centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation. It is computed using the Z-score:

$$Z = \frac{x - \mu}{\sigma} \tag{5.2}$$

where x is the raw score, μ is the mean, and σ is the standard deviation. Normalization is a scaling technique in which values are shifted and rescaled so that they range between 0 and 1 applies [54]. It is implemented by applying the Min-Max scaler formula:

$$X' = \frac{X - min(X)}{max(X) - min(x)} \tag{5.3}$$

Here, $max(X)$ and $min(X)$ are the maximum and the minimum values of the feature, respectively.

5.4.5 Feature Importance

Feature importance reflects the relative contributions of features to predictions made by a model. They are a set of techniques that assign a score to input features based on how useful they are at predicting a target variable. Feature importance scores play an important role in a predictive modeling project, providing an insight into data, an insight into the model, and the basis for dimensionality reduction and feature selection which can improve the efficiency and effectiveness of a predictive model on the problem [91].

In the proposed training framework, a feature's importance is calculated as the difference between the baseline score and the average score obtained by permuting the corresponding column of the test set. If the difference is small, then the model is insensitive to permutations of the feature, thus its importance is low. Conversely, if the difference is large, then the feature's importance is high. The parameter `permutations`, set by the user, controls the number of permutations per feature. More permutations imply better estimates, at the cost of computation time. The user can decide which score metric to use among a set of four possibles:

- `cohens_kappa_score`: computes the cohen's kappa coefficient.
- `accuracy_score`: uses the classification accuracy.

- `balanced_accuracy_score`: defined as the average of recall obtained on each class.
- `f1_score`: computes the classification F1-score.

5.5 Classifier models

In the case of the classifiers, the same model developed in section 3.2: The feed-forward neural network, the long short-term memory and the support vector machine. The main differences lie in the inputs and outputs of the system, in addition to the regularization criteria that the user might choose to apply. The input size is now determined by the number of features selected by the user. The outputs, on the other hand, correspond to the number of classes being classified. If the regularization checkbox is selected, then a penalization will be applied to the chosen classifier. The penalization methods to apply were described in subsection 5.4.3. On the contrary, if regularization is unchecked, then `kernel_regularizer=None` will be pass to the model. This means that no penalization will be applied.

5.6 Evaluation and results

Several tests were performed in order to proof the reliability of the proposed frameworks using both datasets. The first step was to set the training parameters. A group of them, e.g., the activation threshold, the test set size and the classifier's hyperparameters (learning rate, batch size and epochs) remained unvaried among the experiments. Others, like the stride value and window size, were gradually incremented to study their influence on the system performance.

5.6.1 Parameters setting

In Figure 5.2, from top to bottom, it is first needed to determine the dataset-related parameters. Appliances activation thresholds were manually configured, thus, the remaining parameter to set in this part is the number of classes. Appliance activation thresholds were established after analyzing the data of all the appliances in each dataset, and observing their stand-by pulses.The resultant values are shown in Table 5.1. The number of classes corresponds to the name of appliances selected in both datasets, i.e., 5 for UK-DALE and 3 for Dataport datasets.

Table 5.1: Manual set activation thresholds and selected time range to use for training in both dataset: UK-DALE and Dataport.

DATASET	SELECTED TIME RANGE	APPLIANCES	ACTIVATION THRESHOLD
UK-DALE	4/4/2015-4/18/2015	washing machine	0
		oven	3
		iron	1
		hair dryer	1
		dishwasher	1
Dataport	1/1/2018-1/8/2018	electric vehicle	0.001
		oven	0.004
		microwave	0.004

When checking for missing values, the method for filling the gaps was `ffill`. On the other hand, the time range to extract the features was determined by analyzing the appliances signatures and searching for an interval in which all selected appliances had activations. This interval needed to be as short as possible, since in a possible practical use-case, this selection implies the minimum amount of time required for collecting data, but at the same time, the classifiers need enough data for training. The selected ranges are also shown in Table 5.1. A total of two weeks and one week data for UK-DALE and Dataport, respectively, were used for training.

Regarding the feature extractor parameters, the window size and stride were gradually increased to assess the system behavior through several metrics. For this analysis, all the features were selected in `padding` mode. However, some experiments used a different value of the last two, to study their influence on the model's predictions.

The rest of the pre-processing and training parameters: test size, feature scaling method, coefficients of regularizers and number of permutations in the feature importance remained unvaried over the experiments and its values are summarized in Table 5.2. The classifiers hyperparameters and initial setting are shown in Table 5.3.

5.6.2 Analysis of the influence of the feature extractor parameters

Figures 5.6 and 5.7 show the results obtained with the two datasets after varying the feature extractor parameters, stride and window size, to analyze their influence in the FFNN

Table 5.2: Feature extraction parameters.

Parameter	Value
Test size	10%
Feature scaling method	`standardization`
mode	`padding`
Baseline score	`cohen_kappa_score`
Permutations	100

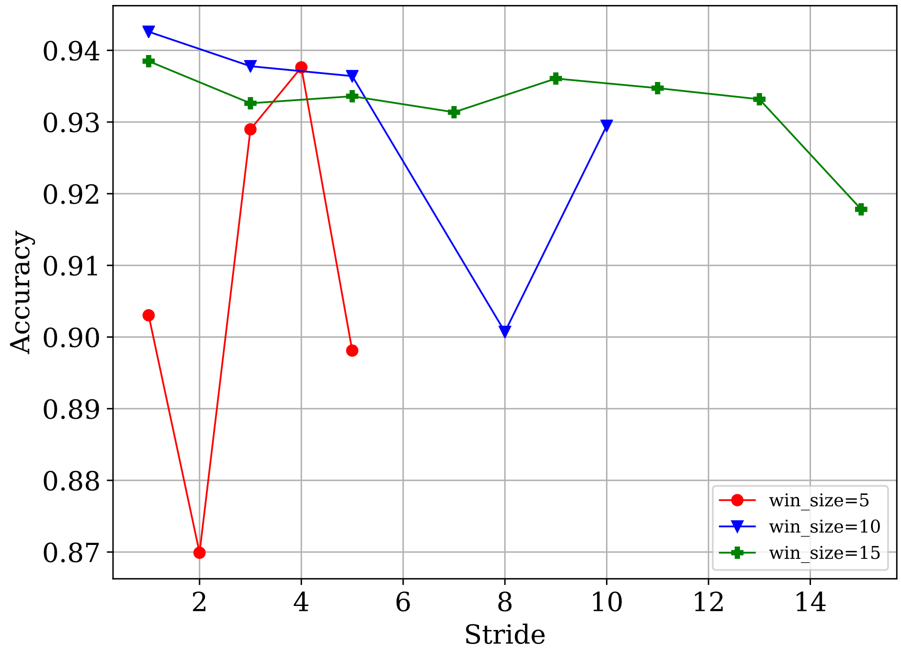
Table 5.3: Training hyperparameters.

Hyperparameter	Value
Epochs	100
Batch size	10
Optimizer	`Adam`
Regularizer	`l1_l2`
L1_norm coefficient	0.00001
L2_norm coefficient	0.0001

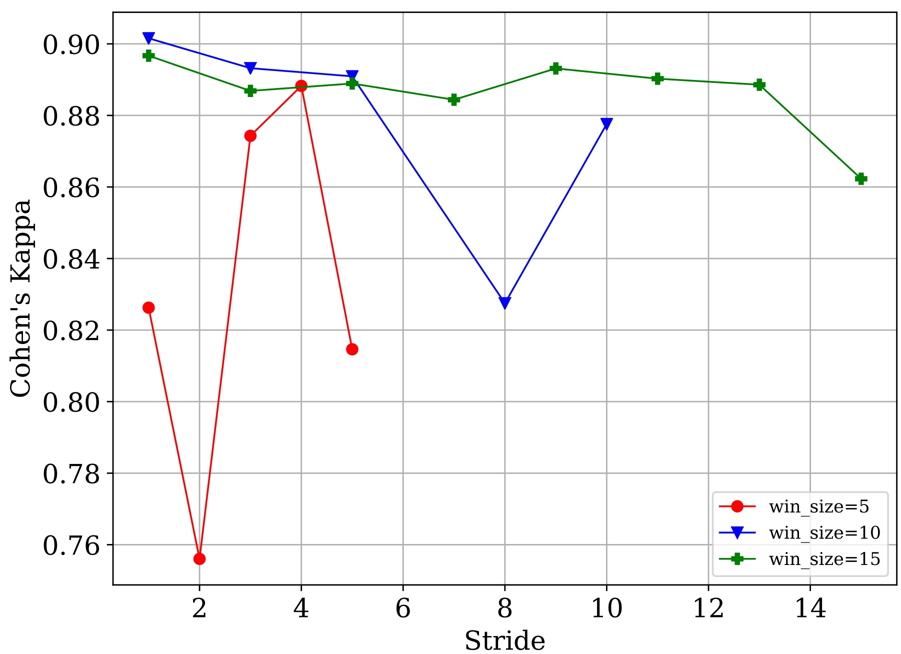
model, the model with the best results in Chapter 3. The metrics to evaluate the model's performance are the accuracy in the case of Figure 5.6a and Figure 5.7a, and the cohen's kappa coefficient, in the case of Figure 5.6b and Figure 5.7b. Since we are facing imbalance data, both metrics need special attention.

Accuracy and cohen's kappa are obtained from three different values of window size: 5, 10 and 15, by gradually increasing the stride. With the UK-DALE, represented in Figure 5.6, for a low value of window size, such as 5, the accuracy seems to have a random behavior ranging from 0.86 with a stride of 2 to 0.93 with a stride of 4. For high values of window size, the accuracy is higher for low values of stride. The best performance was obtained for a value of 10 in window size and stride of 1. In the Dataport dataset, the accuracy reaches 0.99, showing a very competitive and solid performance. In this case, the best result in this case is obtained from a window size of 15 and a stride of 13.

In terms of cohen's kappa, the values obtained from the UK-DALE dataset are, in most cases, around 0.8. The highest values, and therefore the best performance for this dataset is obtained from a window size of 10 and a stride of 1. For the Dataport dataset, the

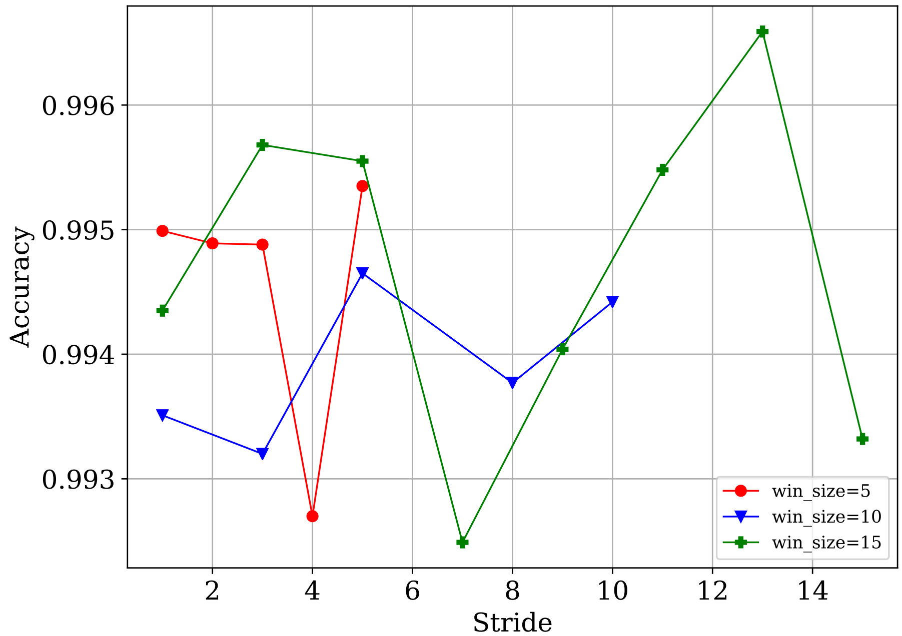


(a) Accuracy obtained for different values of stride and window size using UK-DALE data.

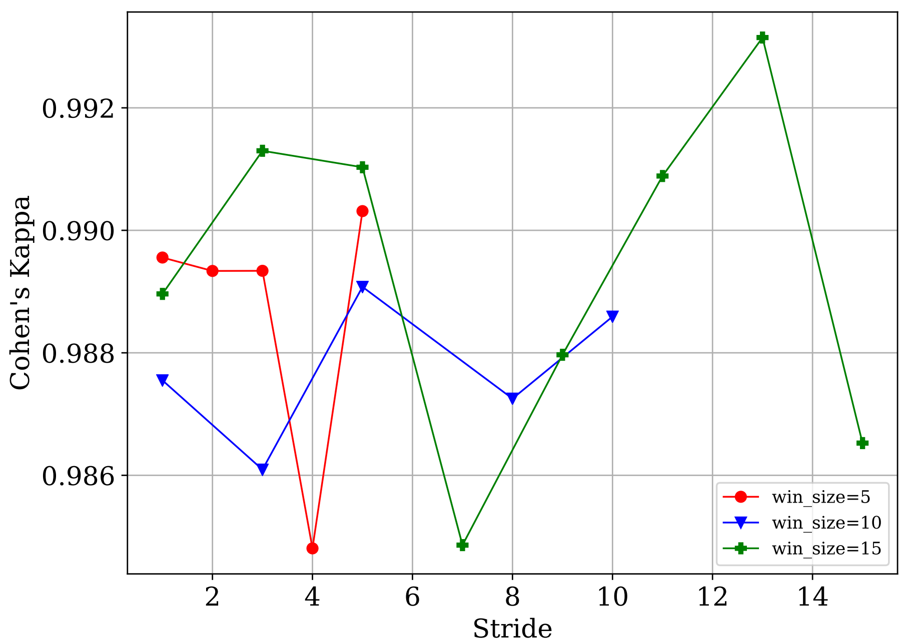


(b) Cohen's Kappa coefficient obtained for different values of stride and window size with UK-DALE data.

Figure 5.6: Resultant metrics after varying the stride and window size with UK-DALE data.

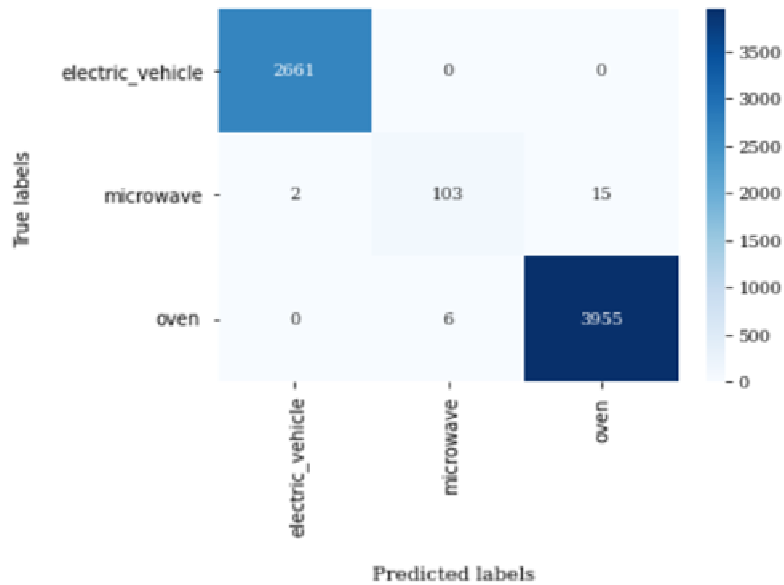


(a) Accuracy obtained for different values of stride and window size with Dataport data.

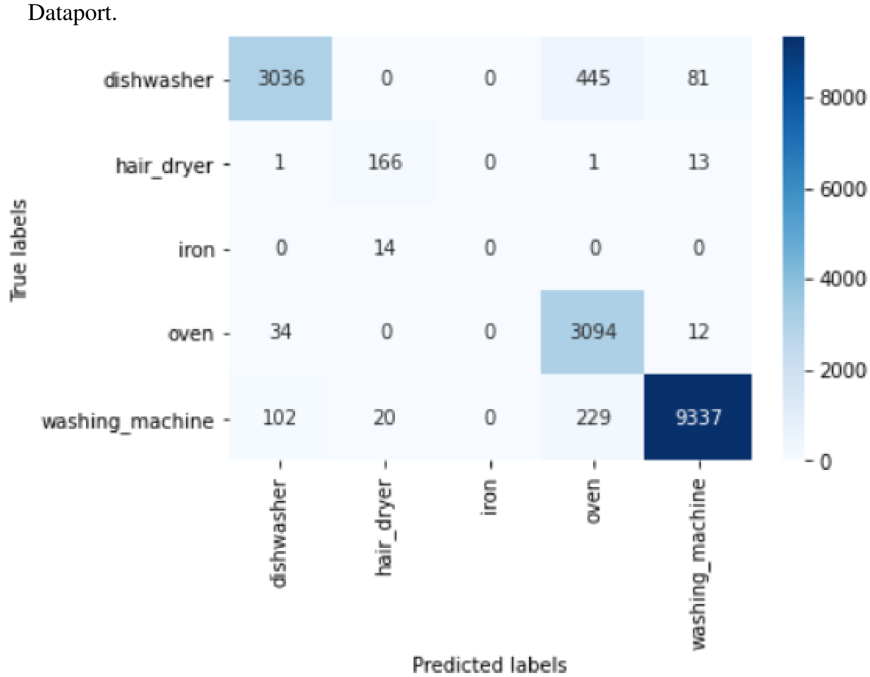


(b) Cohen's Kappa coefficient obtained for different values of stride and window size with Dataport data.

Figure 5.7: Resultant metrics after varying the stride and window size with UK-DALE data.



(a) Confusion matrix for FFNN classifier with the best configuration parameters in Dataport.



(b) Confusion matrix for FFNN classifier with the best configuration parameters in UK-DALE

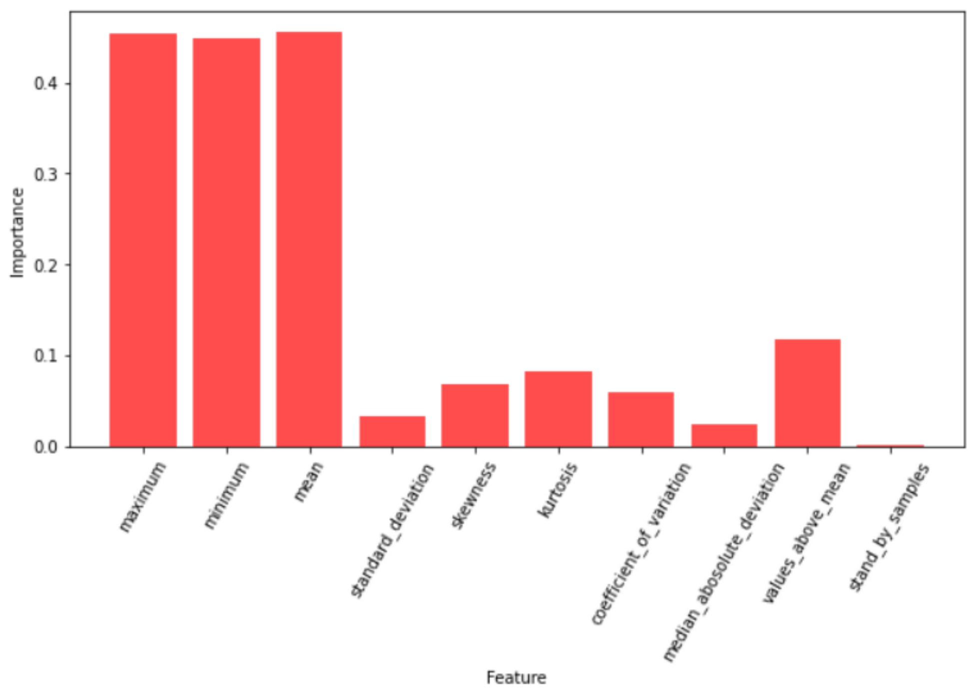
Figure 5.8: Confusion matrices obtained for the best parameter configuration using both datasets: UK-DALE and Dataport.

kappa coefficient is, in most cases, around 0.98, achieving a maximum of 0.993148.
The difference in performance in both datasets could rely on the number of classes (5 in UK-DALE and only 3 for Dataport) and the number of features extracted from each class. In different configurations of the parameters, there is a lower imbalance in the data of Dataport dataset.
All the above experiments were ran using all possible features and the model which gave the best results in the previous version: the FFNN. The mode used in all cases was the default `padding` and the selected regularizer was `l1_l2`.
The confusion matrices obtained from the best parameter configuration in both datasets are shown in Figure 5.8. In the Dataport, all the outliers are related to the microwave, 71 features vector were incorrectly classified as an oven. Both appliances have a pretty similar behavior in terms of consumption. One of the most probable causes of these misclassifications may be related to the features selected (all of them in this case).
With the UK-DALE dataset, the most visible problems are related to the dishwasher. The vast majority of misclassifications in this class are labeled as either an oven or a washing machine. More attention should be paid to the minority class, the iron, in which 100% of their feature vectors were incorrectly classified. This could be highly related to the features selected. Therefore, an analysis of the importance of the selected features may be helpful to overcome this issue.

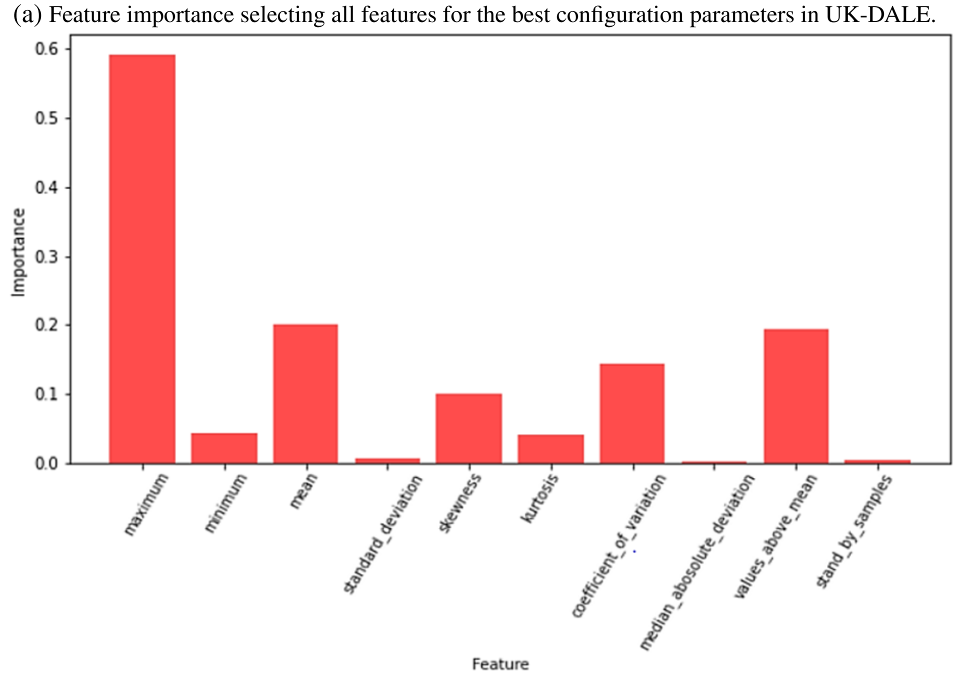
5.6.3 Feature importance analysis

Analyzing the feature importance those cases in which the best result was achieved in terms of accuracy and coheńs kappa coefficient, Figures 5.9a and 5.9b show the importance obtained when the complete set of features were selected for training using both datasets.

Retraining with only relevant features in both cases, the results show an improvement comparing with Figure 5.8b and Figure 5.8a for the UK-DALE and Dataport, respectively. In the case of UK-DALE, more samples in the minority class are correctly classified when training with the most relevant features. For the Dataport dataset, improvements are evidence in the majority classes. Figure 5.10 shows the confusion matrices obtained after retraining with only relevant features. Figure 5.10a shows the one obtained from the UK-DALE data, and Figure 5.10b from Dataport data. This analysis reveals that depending

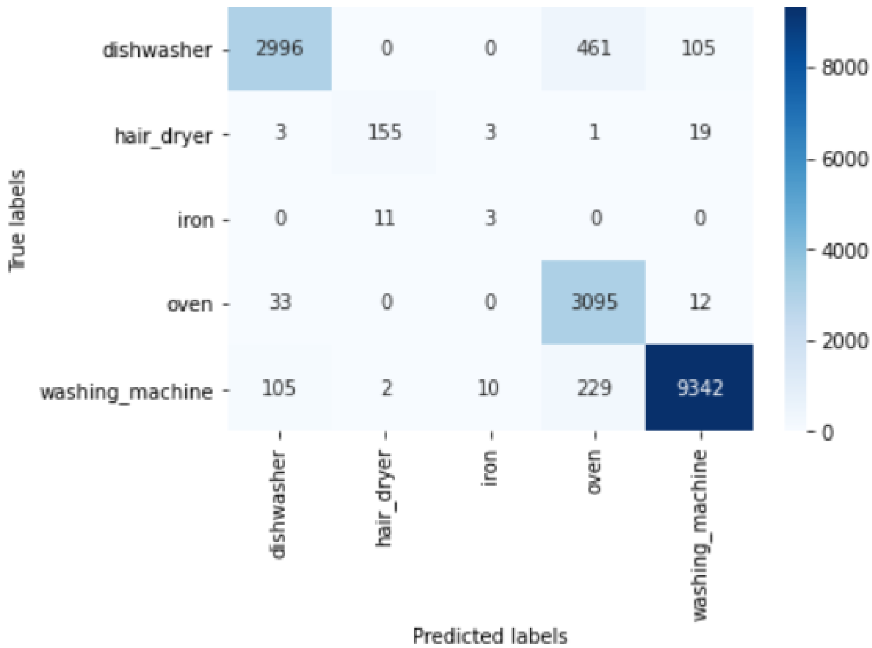


(a) Feature importance selecting all features for the best configuration parameters in UK-DALE.

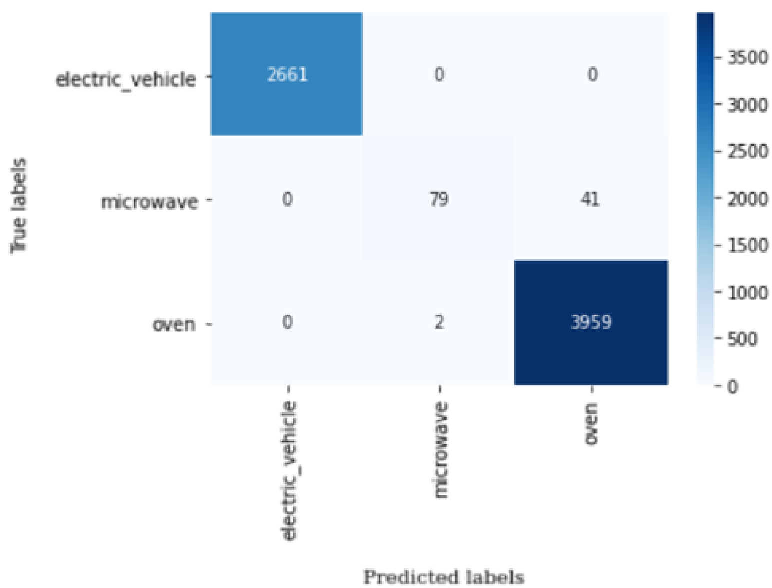


(b) Feature importance selecting all features for the best configuration parameters in Dataport.

Figure 5.9: Feature importance selecting the complete set of features (10) with the best configuration parameters in both datasets: UK-DALE and Dataport.



(a) Confusion matrix for FFNN classifier with the best configuration parameters in UK-DALE after removing less relevant features (only the number of stand-by samples, in this case).



(b) Confusion matrix for FFNN classifier with the best configuration parameters in Dataport after removing less relevant features (standard deviation, the mean absolute deviation and the stand-by values, in this case).

Figure 5.10: Consufion matrices obtained for the best parameter configuration after computing the feature importance using both datasets: UK-DALE and Dataport.

on the data, the selected features will give better results or not, i.e., the performance of the features varies depending on the data. The selection must be specific for the different datasets.

5.6.4 Predicting with the three classifier models

After the analysis performed on the feature extractor parameters, the combination which gave the best results was used to predict the rest of the classifier models. This is in view of testing the performance of the classifiers in the implementation frameworks for appliance recognition. Table 5.4 summarizes the results obtained from this experiment using both datasets. It is evident in Table 5.4 that in both cases (UK-DALE and Dataport) the best performing model involving Stride=1 Window size=10 for UK-DALE and Stride=13 and Window size=15 for Dataport, represents the feed-forward neural network. However, repeating the analysis shown in Figure 5.6 for the LSTM model, the best parameter configuration turned out to be: Stride=8 and Window size=10, giving an accuracy of 0.94602 and a cohen's kappa coefficient of 0.906754. Both metrics are pretty closed to the values obtained from the best parameter configuration for the FFNN; therefore, it is possible to state that the best performance of the machine learning classifiers directly depends on the feature extractor parameter configuration.

Special attention should be paid to the minority class in UK-DALE (an iron) as none of the 14 samples in the test set was correctly classified. This is most likely due to the problems that entail the fact of working with imbalanced data. Support vector machine is very sensitive to class imbalance [92]. Hence, a similar analysis as the one performed in section 5.6.3 is required to improve the FFNN model's behavior.

With the Dataport dataset, the support vector machine gives significantly poor results in comparison with the rest of the models. Thus, it is not recommended to choose the SVM as the preferred classifier model in this occasion.

5.6.5 Predicting with new data

To test the behavior of the model in front of new data, the Inference framework was configured to predict using data from a new house. In this case, the experiment was carried out in House 5 of UK-DALE, as in Chapter 3. The parameters set were the same as in

Table 5.4: Evaluation metrics for FFNN, LSTM and SVM, computed in both datasets and using the best configuration parameters: Stride(1) Widow size(10) for UK-DALE and Stride(13) and window size(15).

DATASETS	CLASSES	FFNN			LSTM			SVM		
		PRECISION	RECALL	F1-SCORE	PRECISION	RECALL	F1-SCORE	PRECISION	RECALL	F1-SCORE
UK-Dale	washing machine	0.98877	0.96377	0.97611	0.97655	0.96728	0.97189	0.98832	0.95231	0.96998
	oven	0.82091	0.98535	0.89564	0.81929	0.98185	0.89323	0.81427	0.98153	0.89011
	iron	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
	hair dryer	0.83000	0.91713	0.87139	0.83838	0.91713	0.87599	0.86224	0.93370	0.89655
	dishwasher	0.95682	0.85233	0.90156	0.96532	0.82061	0.88710	0.92414	0.84812	0.88450
	ACCURACY	0.94260			0.93717			0.93446		
	COHEN'S KAPPA	0.901569			0.891624			0.888067		
Dataport	electric vehicle	0.99925	1.00000	0.99962	0.99905	1.00000	0.99962	0.51222	1.00000	0.67744
	oven	0.99622	0.99849	0.99735	0.9148	0.99899	0.99522	0.99397	0.99924	0.99660
	microwave	0.94495	0.85833	0.89956	0.95455	0.70000	0.80769	1.00000	0.355739	0.52659
	ACCURACY	0.99659			0.99407			0.75863		
	COHEN'S KAPPA	0.993148			0.988040			0.648251		

training. The window size and the stride are 10 and 1, respectively. The selected time range to analyze was also determined by looking for a period of time with activations of every target appliance. The resultant range is from September 1st, 2014 to September 8th, 2014. The confusion matrix obtained from this prediction is shown in Figure5.11. As it can be seen, the model behaves poorly in front of unseen data having a very low accuracy of 0.46241. Therefore, as in Chapter 3, in order to apply this system in a new house, first, the classifier models have to be retrained.

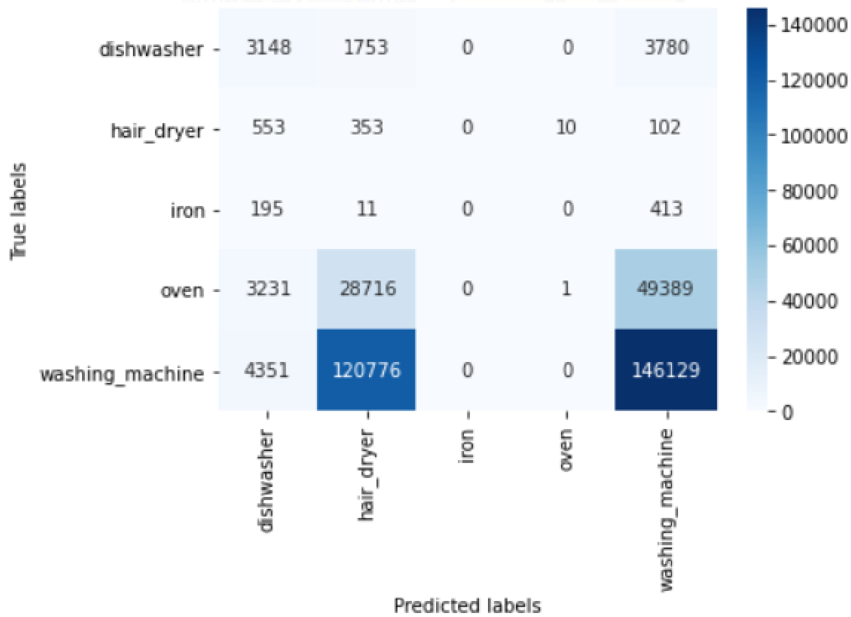


Figure 5.11: Confusion matrix obtained after applying the trained model in House 5 of UK-DALE.

A second prediction was made, but this time using new data from House 1 of UK-DALE dataset and having the same configuration of feature extraction parameters (Stride+1 and Window size=10). In this case, the system achieved an accuracy of 0.93385, which is not far from the 0.9426 obtaining from training. The confusion matrix for this experiment is shown in Figure 5.12.

5.7 Discussion

In this chapter, two frameworks were developed aiming at presenting a tool that facilitates a testbed implementation of the ADL classification System. To be successful in a practical scenario, the system presented in Chaper 3 needed some modifications. Especially, in the feature extraction process, the system presented in Chapter 3 is not capable of working

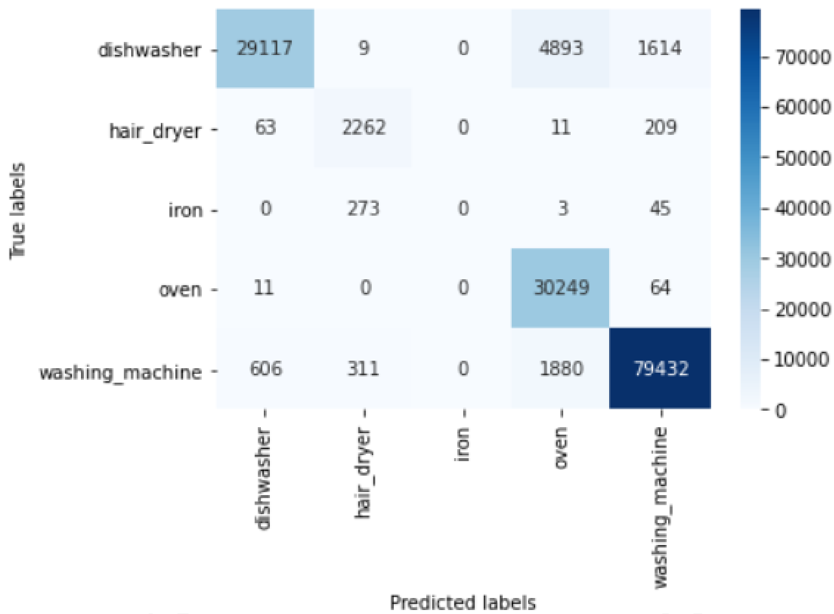


Figure 5.12: Confusion matrix obtained after applying the trained model in House 1 of UK-DALE using a different time range from training.

in real-time. Now, with new considerations, the feature extraction relies on a sliding window. Depending on the size and stride programmed by a user, the system will be closer or not to real-time operation. For example, if the sampling frequency is 6 seconds (as in UK-DALE), and the window size is 10, this means that it is necessary to wait a minute to gather all the samples in the window and extract the features. If the sampling frequency is 1 minute (case of Dataport) and the window size is 15, this means that 15 minutes are necessary to collect the numbers of samples considered in the window, thus, it will be further from real-time. Therefore, although 10 and 15 are the sizes which gave the best results in UK-DALE and Dataport, respectively, the analysis for choosing the values of these parameters for a practical operation has to consider their relation with real-time necessarily. The sensitivity analysis on both, the stride and window size, causes to determine these parameters efficiently.

On the other hand, new data were introduced, including new appliances as an electric vehicle or a microwave. Although the oven is a target appliance in both datasets, they behave differently. Consequently, this ratifies the fact that even if two appliances are of the same type, consumption may differ significantly. The microwave and the oven exhibit pretty similar signatures, however, the classifier is able to distinguish between them accurately.

With regard to predictions using unseen data, to apply the system in a different house, the classifier models still need to be retrained. This is due to low accuracy and poor generalization shown in the experiments performed in House 5 of UK-DALE. When predicting new data from the same house but in a different time range, the accuracy does not vary much from training.

Although there are some challenges that need to be overcome, the proposed frameworks facilitates the application of the same principle (pre-processing and classification) to data organized in different structures, introducing certain standardization in this process.

6 | Conclusions and Future Work

Smart homes aim to facilitate the operation and management of household appliances so that it can be operated automatically and optimally. With the identification of appliance, a series of smart grid applications could be carried out, such as demand response and load planning. This work presented an IoT based approach for load monitoring applications. The IoT architecture can support distributed sensing and ADL classification in smart homes. In a future perspective, this structure can also include smart meter data to provide a more efficient load monitoring system, allowing the integration of applications such as demand response, energy consumption understanding and load planning. The same principle can be applied not only in the context of smart homes, but also in buildings too, thus contributing to achieve a more efficient smart grid.

This work proposed an ADL classification system that combines state-of-the-art solutions among its different modules. ILM is based on low-end meter devices attached to home appliances in opposition to NILM techniques, in which only a single point of sensing is needed. Machine learning models are applied in the appliance recognition module, specifically, three different models were tested using the UK-DALE dataset: FFNN, LSTM and SVM. Accuracy was above 0.9 for FFNN and LSTM classifiers, and around 0.8 for SVM. Once appliances are recognized, the ADL classification algorithm infers an activity based on the appliance label obtained and on the timestamps of the samples. To test the performance of the classifier in front of new data, the system was applied for a different house in the same dataset, notably decreasing its accuracy. Results suggest that before having the system in full operation, it might be necessary to retrain the classifier using the new data. Another experiment was performed to analyze the impact of the group size on the ML classifier accuracy. These groups gather a fixed number of samples from which appliances are identified. If the group size decreases or increases, the same behavior can be expected for the accuracy, apart from the LSTM model which increments the accuracy when the group size is bigger.

With regard to the ADL classification algorithm, future work will explore the ways of redefining the '*sleeping*' and '*unoccupied*' activities, directly related in this thesis to the lack of activity. In practice, people interact with low power appliances, which are harder to identify and are wrongly associated with unoccupied appliances. For this reason, this part requires much more exploration and discussion. Another future improvement of the ADL classification algorithm relies on evaluating the possibilities to include other types of metering devices (such as motion sensors) to compute useful metrics like house occupancy. In addition, the inclusion of smart energy data, ambient assisted living technologies, load monitoring and classification of daily activities provide significant opportunities to improve the quality of life of consumers. Future work will study and evaluate the possibilities to enrich its current implementation in medical practice.

Although the ADL classification system successfully proofs the concept of recognizing activities through an intrusive approach of load monitoring to allow a practical implementation of this system in a laboratory environment (Testbed), some modifications need to be carried out. In this regard, two new frameworks were presented for appliance recognition: a Training framework and an Inference framework. In addition to bringing an easy-to-use tool to the user training or predicting the system through the use of a graphical interface, it allowed to incorporate some major loads, such as an electric vehicle and a microwave, in the monitoring system. The available data with these two new appliances, Dataport dataset, show a notable difference in structure with regard to UK-DALE dataset. Therefore, new statistic features were proposed in order to apply the same pre-processing principle in both cases. The proposed frameworks allow the user to select and test specific parameters related with dataset configuration, feature extraction and classifier model setting. Feature extraction relies on a sliding window. Depending on the size and stride programmed by a user, this system will be able to operate in real-time. A sensitivity analysis on stride and window size was performed aiming to find the values that gave higher accuracy. This metric was about 0.99 and 0.94 for the best configuration parameters when evaluating this system using both datasets. Another aspect to remark is the analysis on feature importance.

The user not only has the possibility to select which feature to extract, but they can also carry out an analysis to quantify the influence of selected features in the models prediction. The main limitation is still the behavior of the classifier models in front of new data, which still shows low accuracy. This means that to apply the system in a new house, it has to be retrained first. However, for new data of the same house, the performance is stable with regard to the training process. Future work will try to overcome this generalizing limitation by studying new classifier models and pre-processing techniques which improve the performance of this system.

6.1 Contributions

- P. Franco, J.M. Martínez, Y.C. Kim and M.A Ahmed, "IoT Based Approach for Load Monitoring and Activity Recognition in Smart Homes", in IEEE Access, vol. 9, pp. 45325 - 45339, March 2021, doi: 10.1109/ACCESS.2021.3067029.
- P. Franco, J.M. Martínez, Y.C. Kim and M.A Ahmed, "A Framework for IoT based Load Monitoring in Smart Homes", in IEEE Access, *under revision*.

Bibliography

[1] Yasin Kabalci, Ersan Kabalci, Sanjeevikumar Padmanaban, Jens Bo Holm-Nielsen, and Frede Blaabjerg. Internet of Things Applications as Energy Internet in Smart Grids and Smart Environments. *Electronics*, 8(9):972, August 2019.

[2] Sen Tan, Yanpeng Wu, Peilin Xie, Josep M. Guerrero, Juan C. Vasquez, and Abdullah Abusorrah. New Challenges in the Design of Microgrid Systems: Communication Networks, Cyberattacks, and Resilience. *IEEE Electrification Magazine*, 8(4):98–106, December 2020.

[3] Ying Wu, Yanpeng Wu, Josep M. Guerrero, Juan C. Vasquez, Emilio J. Palacios-Garcia, and Jiao Li. Convergence and Interoperability for the Energy Internet: From Ubiquitous Connection to Distributed Automation. *IEEE Industrial Electronics Magazine*, 14(4):91–105, December 2020.

[4] Joaquim Leitao, Paulo Gil, Bernardete Ribeiro, and Alberto Cardoso. A Survey on Home Energy Management. *IEEE Access*, 8:5699–5722, 2020.

[5] Bs Hartono, Sri Paryanto Mursid, and Sapto Prajogo. Review: Home energy management system in a Smart Grid scheme to improve reliability of power systems. *IOP Conference Series: Earth and Environmental Science*, 105:012081, January 2018.

[6] Chunhua Liu, K. T. Chau, Diyun Wu, and Shuang Gao. Opportunities and Challenges of Vehicle-to-Home, Vehicle-to-Vehicle, and Vehicle-to-Grid Technologies. *Proceedings of the IEEE*, 101(11):2409–2427, November 2013.

[7] Attique Ur Rehman, Tek Tjing Lie, Brice Vallès, and Shafiqur Tito. *Low Complexity Event Detection Algorithm for Non- Intrusive Load Monitoring Systems*. May 2018. Pages: 751.

[8] Attique Ur Rehman, Tek Tjing Lie, Brice Valles, and Shafiqur Rahman Tito. Event-Detection Algorithms for Low Sampling Nonintrusive Load Monitoring Systems Based on Low Complexity Statistical Features. *IEEE Transactions on Instrumentation and Measurement*, 69(3):751–759, March 2020.

[9] Michael A. Devlin and Barry P. Hayes. Non-Intrusive Load Monitoring and Classification of Activities of Daily Living Using Residential Smart Meter Data. *IEEE Transactions on Consumer Electronics*, 65(3):339–348, August 2019.

[10] Danilo Burbano. Intrusive and Non-Intrusive Load Monitoring (A Survey) Inference and Learning Approach. (1):9, 2015.

[11] Radu-Casian Mihailescu, David Hurtig, and Charlie Olsson. End-to-end anytime solution for appliance recognition based on high-resolution current sensing with few-shot learning. *Internet of Things*, 11:100263, September 2020.

[12] Attique Ur Rehman, Tek Tjing Lie, Brice Valles, and Shafiqur Rahman Tito. Event-Detection Algorithms for Low Sampling Nonintrusive Load Monitoring Systems Based on Low Complexity Statistical Features. *IEEE Transactions on Instrumentation and Measurement*, 69(3):751–759, March 2020.

[13] Francesca Paradiso, Federica Paganelli, Antonio Luchetta, Dino Giuli, and Pino Castrogiovanni. ANN-based appliance recognition from low-frequency energy monitoring data. In *2013 IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pages 1–6, Madrid, June 2013. IEEE.

[14] Antonio Ridi, Christophe Gisler, and Jean Hennebert. A Survey on Intrusive Load Monitoring for Appliance Recognition. In *2014 22nd International Conference on Pattern Recognition*, pages 3702–3707, Stockholm, Sweden, August 2014. IEEE.

[15] Xinmei Yuan, Peng Han, Yao Duan, Rosemary E. Alden, Vandana Rallabandi, and Dan M. Ionel. Residential Electrical Load Monitoring and Modeling – State of the Art and Future Trends for Smart Homes and Grids. *Electric Power Components and Systems*, pages 1–19, November 2020.

[16] F. Bouhafs, M. Mackay, and M. Merabti. Links to the Future: Communication Requirements and Challenges in the Smart Grid. *IEEE Power and Energy Magazine*, 10(1):24–32, January 2012.

[17] U. Zafar, S. Bayhan, and A. Sanfilippo. Home Energy Management System Concepts, Configurations, and Technologies for the Smart Grid. *IEEE Access*, 8:119271–119286, 2020. Conference Name: IEEE Access.

[18] José Paulo Lousado and Sandra Antunes. Monitoring and Support for Elderly People Using LoRa Communication Technologies: IoT Concepts and Applications. *Future Internet*, 12(11):206, November 2020.

[19] Nadia Oukrich. *Daily Human Activity Recognition in Smart Home based on Feature Selection, Neural Network and Load Signature of Appliances*. Theses, Université Mohamed V ; Ecole Mohammadia d'Ingénieurs - Université Mohammed V de Rabat - Maroc, April 2019.

[20] Carl Chalmers, Paul Fergus, Casimiro Adays Curbelo Montanez, Sudip Sikdar, Freya Ball, and Bryony Kendall. Detecting Activities of Daily Living and Routine Behaviours in Dementia Patients Living Alone Using Smart Meter Load Disaggregation. *IEEE Transactions on Emerging Topics in Computing*, pages 1–1, 2020.

[21] Attique Ur Rehman, Tek Tjing Lie, Brice Vallès, and Shafiqur Tito. *Low Complexity Event Detection Algorithm for Non- Intrusive Load Monitoring Systems*. May 2018. Pages: 751.

[22] Attique Ur Rehman, Shafiqur Rahman Tito, Pieter Nieuwoudt, Gulshad Imran, Tek Tjing Lie, Brice Valles, and Waseem Ahmad. Applications of Non-Intrusive Load Monitoring Towards Smart and Sustainable Power Grids: A System Perspective. In

2019 29th Australasian Universities Power Engineering Conference (AUPEC), pages 1–6, Nadi, Fiji, November 2019. IEEE.

[23] Daisy H. Green, Steven R. Shaw, Peter Lindahl, Thomas J. Kane, John S. Donnal, and Steven B. Leeb. A MultiScale Framework for Nonintrusive Load Identification. *IEEE Transactions on Industrial Informatics*, 16(2):992–1002, February 2020.

[24] M. Gaur and A. Majumdar. Disaggregating Transform Learning for Non-Intrusive Load Monitoring. *IEEE Access*, 6:46256–46265, 2018. Conference Name: IEEE Access.

[25] Michele DIncecco, Stefano Squartini, and Mingjun Zhong. Transfer Learning for Non-Intrusive Load Monitoring. *arXiv:1902.08835 [cs, stat]*, September 2019. arXiv: 1902.08835.

[26] Jack Kelly and William Knottenbelt. Neural NILM: Deep Neural Networks Applied to Energy Disaggregation. *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments - BuildSys '15*, pages 55–64, 2015. arXiv: 1507.06594.

[27] E. J. Aladesanmi and K. A. Folly. Overview of non-intrusive load monitoring and identification techniques. *IFAC-PapersOnLine*, 48(30):415–420, January 2015.

[28] Jack Kelly, Nipun Batra, Oliver Parson, Haimonti Dutta, William Knottenbelt, Alex Rogers, Amarjeet Singh, and Mani Srivastava. NILMTK v0.2: a non-intrusive load monitoring toolkit for large scale data sets: demo abstract. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings - BuildSys '14*, pages 182–183, Memphis, Tennessee, 2014. ACM Press.

[29] Mohammad Mahmudur Rahman Khan, Abu Bakr Siddique, and Shadman Sakib. Non-Intrusive Electrical Appliances Monitoring and Classification using K-Nearest Neighbors. page 5.

[30] Ying Wu, Yanpeng Wu, Josep M. Guerrero, and Juan C. Vasquez. Digitalization and decentralization driving transactive energy Internet: Key technologies and infrastructures. *International Journal of Electrical Power & Energy Systems*, 126:106593, March 2021.

[31] Attique Ur Rehman. *Load Disaggregation: Towards Energy Efficient Systems*. Thesis, Auckland University of Technology, 2021. Accepted: 2021-02-16T21:18:35Z.

[32] Suman Giri, Mario Bergés, and Anthony Rowe. Towards automated appliance recognition using an EMF sensor in NILM platforms. *Advanced Engineering Informatics*, 27(4):477–485, October 2013.

[33] K. Anderson, A. Ocneanu, Diego Benitez, Derrick Carlson, A. Rowe, and M. Berges. BLUED: A fully labeled public dataset for event-based non-intrusive load monitoring research. *Proceedings of the 2nd KDD Workshop on Data Mining Applications in Sustainability (SustKDD)*, pages 1–5, January 2012.

[34] Dataport | Residential Data.

[35] Jose A. Oliveira-Lima, Ramiro Morais, J. F. Martins, Adrian Florea, and Celson Lima. Load forecast on intelligent buildings based on temporary occupancy monitoring. *Energy and Buildings*, 116:512–521, March 2016.

[36] Omid Ardakanian, Arka Bhattacharya, and David Culler. Non-intrusive occupancy monitoring for energy conservation in commercial buildings. *Energy and Buildings*, 179:311–323, November 2018.

[37] Han Zou, Hao Jiang, Jianfei Yang, Lihua Xie, and Costas Spanos. Non-intrusive occupancy sensing in commercial buildings. *Energy and Buildings*, 154:633–643, November 2017.

[38] Nagender Kumar Suryadevara and Gyan Ranjan Biswal. Smart Plugs: Paradigms and Applications in the Smart City-and-Smart Grid. *Energies*, 12(10):1957, May 2019.

[39] Christoph Klemenjak, Stephen Makonin, and Wilfried Elmenreich. Towards Comparability in Non-Intrusive Load Monitoring: On Data and Performance Evaluation. In *2020 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5, Washington, DC, USA, February 2020. IEEE.

[40] J Zico Kolter and Matthew J Johnson. REDD: A Public Data Set for Energy Disaggregation Research. page 6.

[41] Stephen Makonin, Fred Popowich, Lyn Bartram, Bob Gill, and Ivan Bajic. *AMPds: A Public Dataset for Load Disaggregation and Eco-Feedback Research*. August 2013. Journal Abbreviation: Electrical Power and Energy Conference, the Annual (EPEC) Pages: 6 Publication Title: Electrical Power and Energy Conference, the Annual (EPEC).

[42] UK Domestic Appliance-Level Electricity (UK-DALE) dataset.

[43] Akshay S.N. Uttama Nambi, Antonio Reyes Lua, and Venkatesha R. Prasad. LocED: Location-aware Energy Disaggregation Framework. In *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, pages 45–54, Seoul South Korea, November 2015. ACM.

[44] Andrea Monacchi, Dominik Egarter, Wilfried Elmenreich, Salvatore D'Alessandro, and Andrea M. Tonello. GREEND: An energy consumption dataset of households in Italy and Austria. In *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 511–516, Venice, Italy, November 2014. IEEE.

[45] Christian Beckel, Wilhelm Kleiminger, Romano Cicchetti, Thorsten Staake, and Silvia Santini. The ECO data set and the performance of non-intrusive load monitoring algorithms. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, pages 80–89, Memphis Tennessee, November 2014. ACM.

[46] Jingkun Gao, Suman Giri, Emre Can Kara, and Mario Bergés. PLAID: a public dataset of high-resoultion electrical appliance measurements for load identification research: demo abstract. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, pages 198–199, Memphis Tennessee, November 2014. ACM.

[47] David Murray, Lina Stankovic, and Vladimir Stankovic. An electrical load measurements dataset of United Kingdom households from a two-year longitudinal study. *Scientific Data*, 4(1):160122, December 2017.

[48] Ben Anderson, David Eyers, Rebecca Ford, Diana Giraldo Ocampo, Rana Peniamina, Janet Stephenson, Kiti Suomalainen, Lara Wilcocks, and Michael Jack. New Zealand GREEN Grid Household Electricity Demand Study 2014-2018, September 2018. Publisher: UK Data Service.

[49] Miguel Ribeiro, Lucas Pereira, Filipe Quintal, and Nuno Nunes. SustDataED: A Public Dataset for Electric Energy Disaggregation Research. In *Proceedings of ICT for Sustainability 2016*, Amsterdam, the Netherlands, 2016. Atlantis Press.

[50] Nipun Batra, Manoj Gulati, Amarjeet Singh, and Mani B. Srivastava. It's Different: Insights into home energy consumption in India. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings - BuildSys'13*, pages 1–8, Roma, Italy, 2013. ACM Press.

[51] Nipun Batra, Oliver Parson, Mario Berges, Amarjeet Singh, and Alex Rogers. A comparison of non-intrusive load monitoring methods for commercial and residential buildings. *arXiv:1408.6595 [cs]*, August 2014. arXiv: 1408.6595.

[52] Smart-Grid Smart-City Customer Trial Data | Datasets | data.gov.au - beta.

[53] Sean Barker, Aditya Mishra, David Irwin, Emmanuel Cecchet, and Prashant Shenoy. Smart*: An Open Data Set and Tools for Enabling Research in Sustainable Homes. page 6.

[54] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[55] Andriy Burkov. "All models are wrong, but some are useful." — George Box. page 152.

[56] İlker Güven and Fuat Şimşir. Demand forecasting with color parameter in retail apparel industry using artificial neural networks (ANN) and support vector machines (SVM) methods. *Computers & Industrial Engineering*, 147:106678, September 2020.

[57] Josimar Reyes-Campos, Giner Alor-Hernández, Isaac Machorro-Cano, José Oscar Olmedo-Aguirre, José Luis Sánchez-Cervantes, and Lisbeth Rodríguez-Mazahua. Discovery of Resident Behavior Patterns Using Machine Learning Techniques and IoT Paradigm. *Mathematics*, 9(3):219, January 2021. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

[58] Héctor Sánchez San Blas, André Sales Mendes, Francisco García Encinas, Luís Augusto Silva, and Gabriel Villarubia González. A Multi-Agent System for Data Fusion Techniques Applied to the Internet of Things Enabling Physical Rehabilitation Monitoring. *Applied Sciences*, 11(1):331, January 2021. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

[59] Akriti Verma, Adnan Anwar, M. A. Parvez Mahmud, Mohiuddin Ahmed, and Abbas Kouzani. A Comprehensive Review on the NILM Algorithms for Energy Disaggregation. *arXiv:2102.12578 [cs, eess]*, April 2021. arXiv: 2102.12578.

[60] Jonathan E Fieldsend, Richard M Everson, J E Fieldsend, and R M Everson. Visualisation of multi-class ROC surfaces. page 8.

[61] G. W. Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891, December 1992. Conference Name: Proceedings of the IEEE.

[62] Milad Afzalan and Farrokh Jazizadeh. A Machine Learning Framework to Infer Time-of-Use of Flexible Loads: Resident Behavior Learning for Demand Response. *IEEE Access*, 8:111718–111730, 2020.

[63] José M. Alcalá, Jesús Ureña, Álvaro Hernández, and David Gualda. Assessing Human Activity in Elderly People Using Non-Intrusive Load Monitoring. *Sensors (Basel, Switzerland)*, 17(2), February 2017.

[64] Sidney Katz, Amasa B. Ford, Roland W. Moskowitz, Beverly A. Jackson, and Marjorie W. Jaffe. Studies of Illness in the Aged: The Index of ADL: A Standardized Measure of Biological and Psychosocial Function. *JAMA*, 185(12):914–919, September 1963.

[65] M N Mohammed, S F Desyansah, S Al-Zubaidi, and E Yusuf. An internet of things-based smart homes and healthcare monitoring and management system: Review. *Journal of Physics: Conference Series*, 1450:012079, February 2020.

[66] Emad Felemban. A Comparative Study of IEEE 802.11 family Protocols. page 11, 2020.

[67] 802.11ac: The Fifth Generation of Wi-Fi Technical White Paper. page 20, 2018.

[68] T. Watteyne. 4 - Lower-power wireless mesh networks for machine-to-machine communications using the IEEE802.15.4 standard. In Carles Antón-Haro and Mischa Dohler, editors, *Machine-to-machine (M2M) Communications*, pages 63–77. Woodhead Publishing, Oxford, January 2015.

[69] A. R. Deshmukh Vaishnavi V. Kale. SURVEY PAPER ON Z- WAVE TECHNOLOGY. volume Vol. 7 - Issue 5 (May-2020). JETIR, May 2020. ISSN: 2349-5162 Issue: Vol. 7 - Issue 5 (May-2020).

[70] R Alekya, Neelima Devi Boddeti, K Salomi Monica, Dr R Prabha, Dr V Venkatesh, and Kevin Ashton. IoT based Smart Healthcare Monitoring Systems: A Literature Review. *Clinical Medicine*, 7(11):9, 2020.

[71] S. Deep, X. Zheng, C. Karmakar, D. Yu, L. G. C. Hamey, and J. Jin. A Survey on Anomalous Behavior Detection for Elderly Care Using Dense-Sensing Networks. *IEEE Communications Surveys Tutorials*, 22(1):352–370, 2020. Conference Name: IEEE Communications Surveys Tutorials.

[72] Andrea Tundis, Ali Faizan, and Max Mühlhäuser. A Feature-Based Model for the Identification of Electrical Devices in Smart Environments. *Sensors*, 19(11):2611, January 2019. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

[73] J R Cuñado and N B Linsangan. A Supervised Learning Approach to Appliance Classification Based on Power Consumption Traces Analysis. *IOP Conference Series: Materials Science and Engineering*, 517:012011, April 2019.

[74] Labiba Gillani Fahad and Syed Fahad Tahir. Activity recognition and anomaly detection in smart homes. *Neurocomputing*, 423:362–372, January 2021.

[75] Simin Ahmadi-Karvigh, Burcin Becerik-Gerber, and Lucio Soibelman. A framework for allocating personalized appliance-level disaggregated electricity consumption to daily activities. *Energy and Buildings*, 111:337–350, January 2016.

[76] Jon Paxman, Matt James, Dr Enrico Costanza, and Julia Manning. Smart Future of Healthcare. page 76.

[77] Deepti Sehrawat and Nasib Singh Gill. IoT Based Human Activity Recognition System Using Smart Sensors. *Advances in Science, Technology and Engineering Systems Journal*, 5(4):516–522, August 2020.

[78] T. Perumal, Y. L. Chui, M. A. B. Ahmadon, and S. Yamaguchi. IoT based activity recognition among smart home residents. In *2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)*, pages 1–2, October 2017.

[79] Ashwin Geet D'Sa and B. G. Prasad. An IoT Based Framework For Activity Recognition Using Deep Learning Technique. *arXiv:1906.07247 [cs, stat]*, June 2019. arXiv: 1906.07247.

[80] Naeem Raza, Muhammad Qasim Akbar, Aized Amin Soofi, and Samia Akbar. Study of Smart Grid Communication Network Architectures and Technologies. *Journal of Computer and Communications*, 07(03):19–29, 2019.

[81] Jon Paxman. Smart Future of Healthcare - 20/20health.

[82] Samundra Deep, Xi Zheng, Chandan Karmakar, Dongjin Yu, Leonard G. C. Hamey, and Jiong Jin. A Survey on Anomalous Behavior Detection for Elderly Care Using Dense-Sensing Networks. *IEEE Communications Surveys & Tutorials*, 22(1):352–370, 2020.

[83] sklearn.svm.SVC — scikit-learn 0.24.0 documentation.

[84] Jack Kelly and William Knottenbelt. The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. *Scientific Data*, 2(1):150007, December 2015.

[85] Pavel Brandstetter, Jan Vanek, and Tomas Verner. Electric vehicle energy consumption monitoring. In *Proceedings of the 2014 15th International Scientific Conference on Electric Power Engineering (EPE)*, pages 589–592, Brno-Bystrc, Czech Republic, May 2014. IEEE.

[86] Kwang Gi Kim. Book Review: Deep Learning. *Healthcare Informatics Research*, 22(4):351, 2016.

[87] Electric Cars, Solar & Clean Energy.

[88] Ramalingam Shanmugam and Rajan Chattamvelli. *Statistics for scientists and engineers*. John Wiley & Sons, Inc, Hoboken, New Jersey, 2015.

[89] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, June 2002. arXiv: 1106.1813.

[90] Frederic P. Miller, Agnes F. Vandome, and McBrewster John. *Cohen's Kappa*. VDM Publishing, September 2010. Google-Books-ID: szq2cQAACAAJ.

[91] Brandon Butcher and Brian J. Smith. Feature Engineering and Selection: A Practical Approach for Predictive Models: by Max Kuhn and Kjell Johnson. Boca Raton, FL: Chapman & Hall/CRC Press, 2019, xv + 297 pp., ISBN: 978-1-13-807922-9. *The American Statistician*, 74(3):308–309, July 2020.

[92] Benjamin X Wang and Nathalie Japkowicz. Boosting Support Vector Machines for Imbalanced Data Sets. page 10.

A | Appendix

Target appliances' signatures:

From UK-DALE

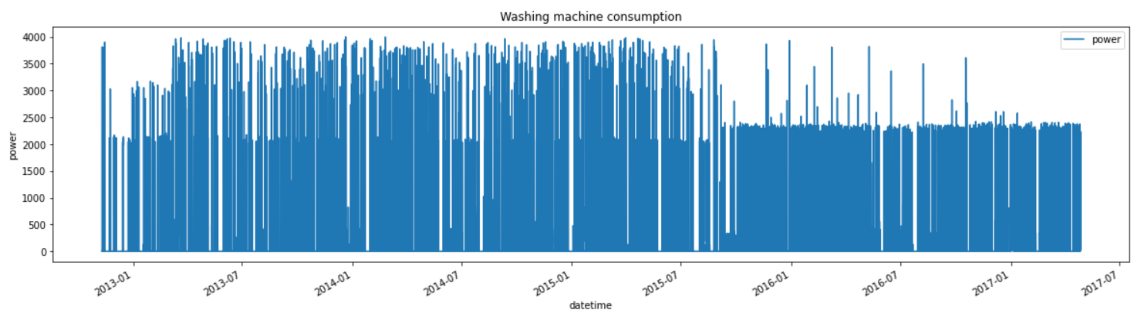


Figure A.1: Power signature of the washing machine.

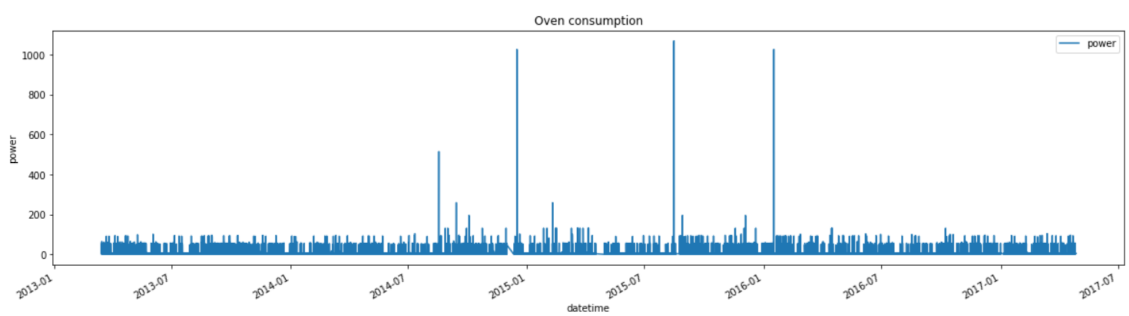


Figure A.2: Power signature of the oven (UK-DALE).

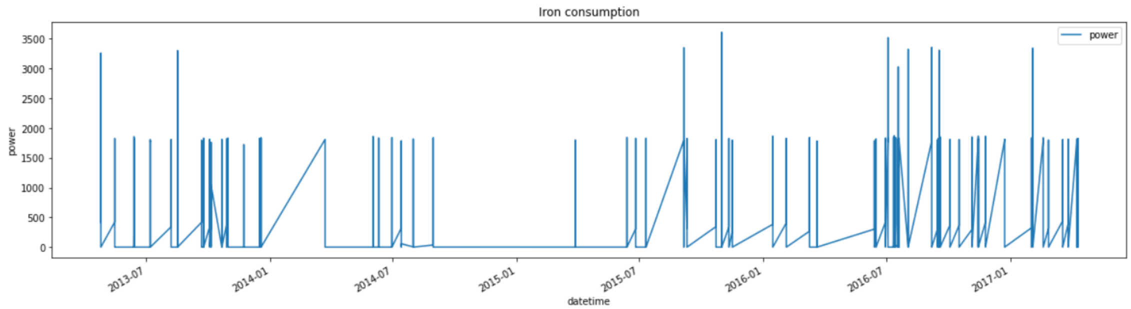


Figure A.3: Power signature of the iron.

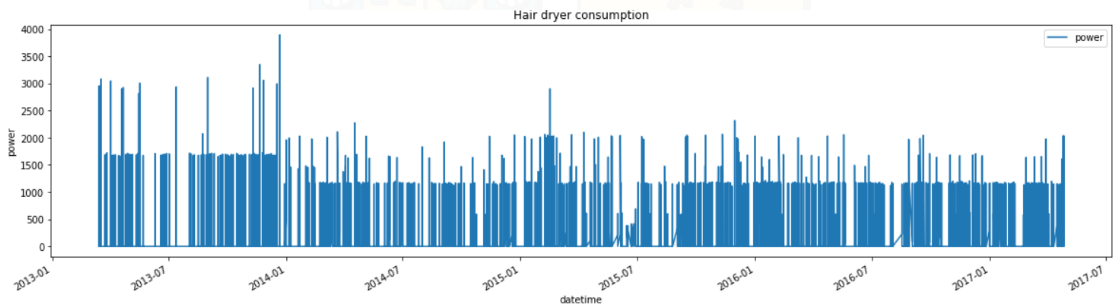


Figure A.4: Power signature of the hair dryer.

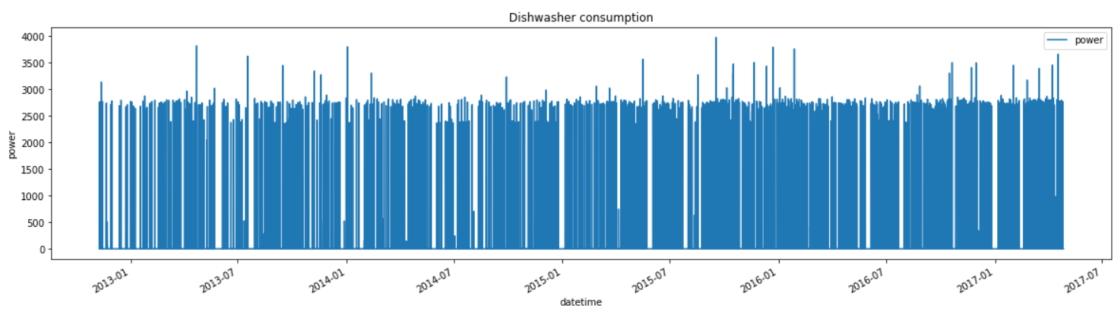


Figure A.5: Power signature of the dishwasher.

From Dataport:

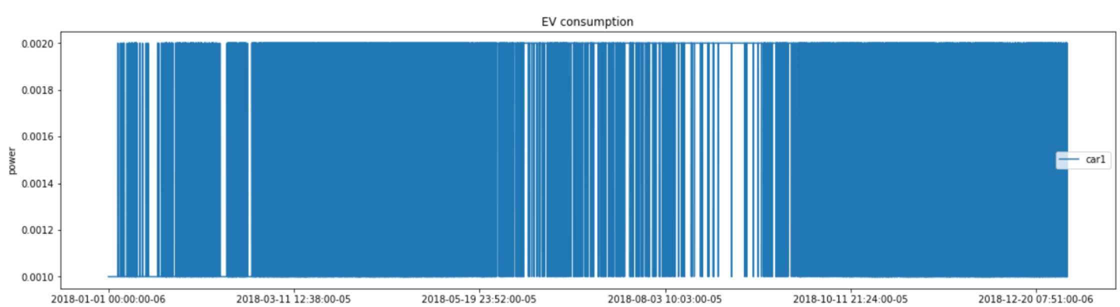


Figure A.6: Power signature of the electric vehicle.

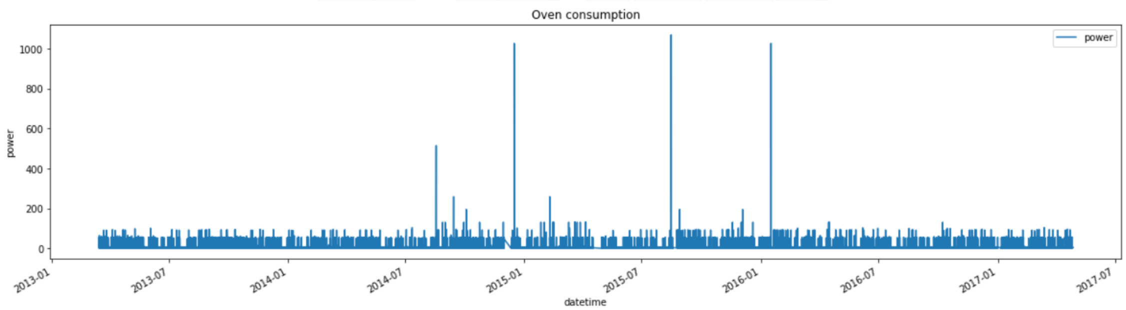


Figure A.7: Power signature of the oven (Dataport).

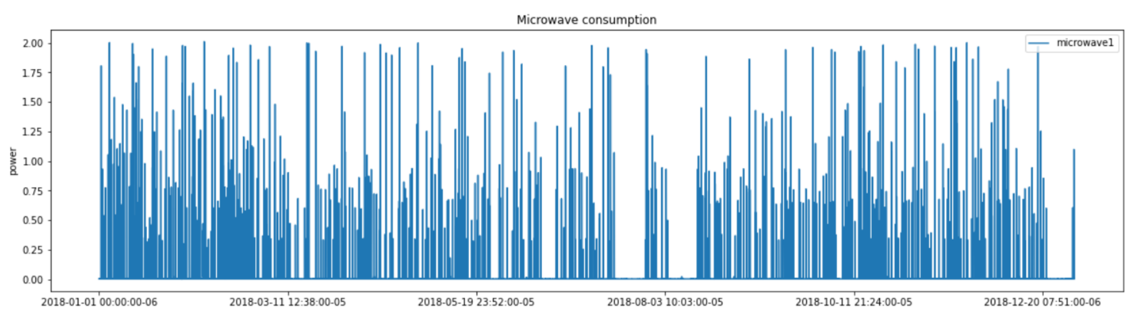


Figure A.8: Power signature of the microwave.

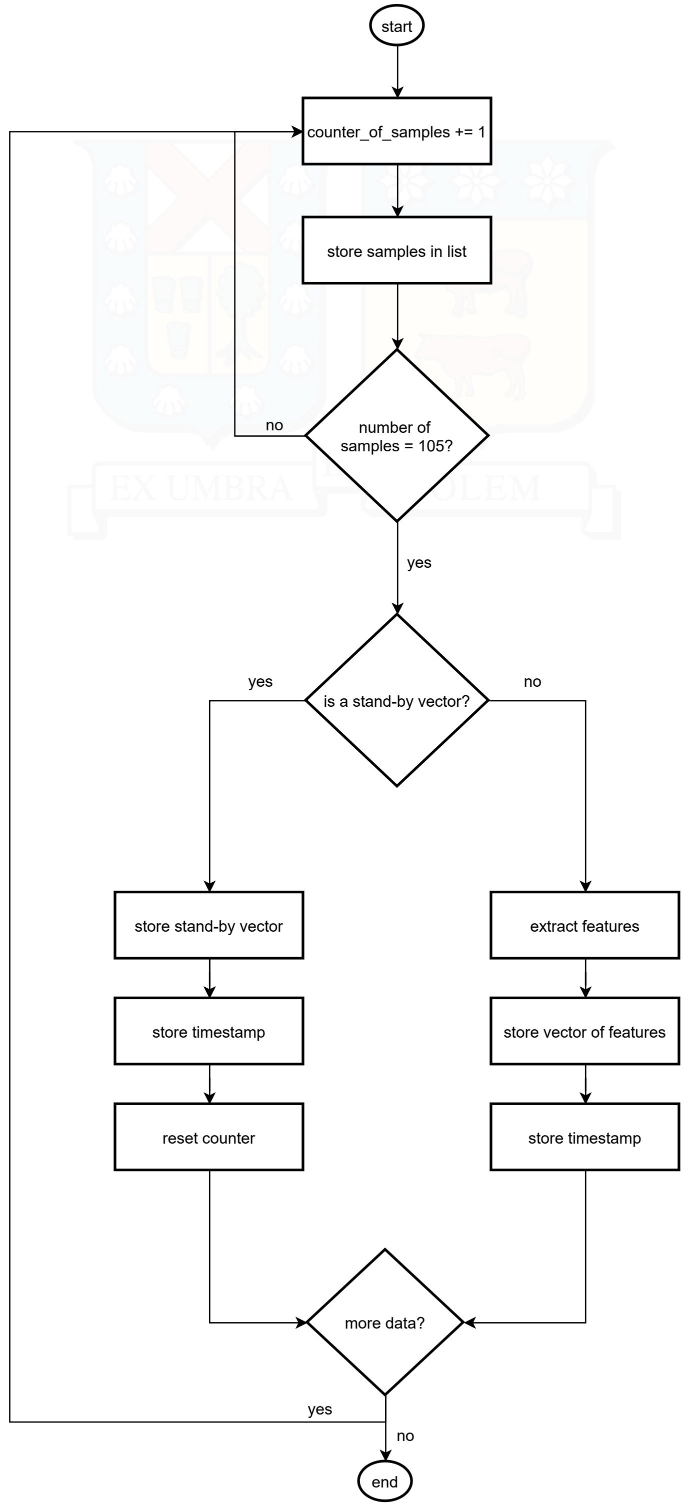


Figure A.9: Flowchart of the first feature extractor algorithm.