D4.3 FINAL ACTIVITY PROFILING AND MATCHING DETECTOR

Demonstrator

Project title	**Collaborative Recommendations and Adaptive Control for Personalised Energy Saving**
Project acronym	**enCOMPASS**
Project call	**EE-07-2016-2017 Behavioural change toward energy efficiency through ICT**
Work Package	**WP4**
Lead Partner	**CERTH**
Contributing Partner(s)	**WVT**
Security classification	**Public**
Contractual delivery date	**31/12/2018**
Actual delivery date	**31/12/2018**
Version	**1.0**
Reviewers	**POLIMI, SUPSI, WVT**

History of changes

Version	Date	Comments	Main Authors
0.1	01/12/2018	Defining Table of Content	CERTH
0.4	10/12/2018	First draft of the document	CERTH, WVT
0.6	21/12/2018	First full version of the document	CERTH, WVT
0.7	27/12/2018	Quality check and review	SUPSI, WVT
0.8	27/12/2018	Quality check and review	PMI
1.0	28/12/2018	Final version	CERTH

Disclaimer

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 723059.

Table of Contents

TABLE OF FIGURES

LIST OF DEFINITIONS, ACRONYMS AND ABBREVIATIONS

Abbreviation	Definition
FN	False Negative
FP	False positive
ID	Identification
MCC	Matthews Correlation Coefficient
ML	Machine Learning
MQTT	Message Queue Telemetry Transport
RF	Random Forest
SVM	Support Vector Machine
TN	True Negative
TP	True Positive

EXECUTIVE SUMMARY

Deliverable D4.3 “Final Activity Profiling and Matching Detector” is a **demonstrator deliverable**, specified in the “amended” GA description as follows:

“Final activity profiling and matching detector. Second prototype, with documentation, of the algorithms for detecting the type and context of users’ activity in different indoor conditions, from exogeneous and endogenous variables and adaptable based on runtime feedback, validated in the pilot experiments”.

This document is an accompanying document of the code developed for D4.3 within Task T4.1, and describes the algorithmic approaches that have been developed for activity estimation and recognition. The State-of-the-art related work was presented in Section 2 at Deliverable D4.1 “First Activity Profiling and Matching Detector.

The code of the software deliverable is available in the following software repository:

ssh://18.184.32.200:22/var/git/ encompass-activity-detector.git

Access is granted upon request.

Its major goal is to explain the algorithms for human activity inference in indoor environments developed within enCOMPASS project.

Building activities detection/ recognition is an essential task for building analysis. It is highly related to the building energy consumption and performance. The activity inference can be performed by analyzing the energy consumption of the building, as well as the energy consumption and state of each device in it. The analysis of the energy consumption can lead to the estimation of the device state, which in turn can lead to the human activity inference.

Deliverable D4.3 provides a description of the algorithms for the activity inference in indoor environments:

- Data pre-processing;
- Device state inference algorithms;
- Human activity inference algorithms.

The main dependencies with other deliverables are as follows:

- Deliverable D3.1 “Datasets with Context Data and Energy Consumption Data”: This deliverable contains the specification of each one of the energy consumption historical data set, which will be collected by the utility companies, as well as the building owners of the enCOMPASS pilots.
- Deliverable D3.3 “First Energy Disaggregation Algorithms” and D3.5 “Final Energy Disaggregation Algorithms”: These deliverables contains the algorithms which will disaggregate and provide information about the energy consumption of individual devices and appliances in buildings, especially in households, where the acquired information will be mainly based on the central building energy consumption.
- Deliverable D3.4 “Final user tracking algorithms”: This deliverable contains the algorithms for the occupancy inference in indoor environments.

The deliverable is structured as follows:

- Section 1 is the introduction of the deliverable;

- Section 2 presents the data used for the activity recognition and inference algorithm;
- Section 3 provides the description of the activity inference algorithm;
- Section 4 gives the experimental results of the activity inference algorithms on the project datasets;
- Section 5 presents a demonstration of the algorithm;
- The final two Sections contain the Conclusions and References.

1 INTRODUCTION

Being aware of the activity of people in real time is very important for security, surveillance and even for understanding behavioural patterns. Nowadays, the true activity of occupants is gaining importance, as activity recognition is associated with appliance consumption, therefore possible energy saving when creating building management systems. Electric consumption of the appliances has a big proportion of total electric consumption worldwide. Towards this end, many researchers try to recognize activity from electric consumption in order to use these results to try to alter the wasteful energy habits. In general, being able to recognise the activity of residents is significant for the performance of a building.

1.1 PURPOSE AND SCOPE

Indoor activity recognition is the process that aims at estimating the action of a resident or group of residents inside the house or within a specific room of the house. The procedure includes many steps, which may differ across selected methods. Regardless of the approach used to observe the action of a person, the first step is to decide which type of data will be used and how it is going to be collected. Moreover, the data must be gathered daily in order to be able to analyze everyday patterns and come to a qualitative conclusion.

The motion sensor data collection and analysis that is used to detect a person’s presence or absence, is primarily used as the main indicator of the occupant's action. Occupancy is significant for the position of a person inside a building. Undoubtedly, one can assume the absence of an occupant by the continuous lack of motion sensing within a given period. Contrarily, the existence of motion detection signals that the occupant changed position. After taking into consideration the energy consumption within a residence, this movement can be ultimately related to distinct activities.

Total residence consumption data that is broken to its components indicates the load of energy consumption disaggregated to the level of the individual appliances. These disaggregated data links appliance usage with individual activity. Substantially, we can recognize from the disaggregation analysis which appliance is used and for how long it is used, therefore we can recognize the type of activity that is directly associated with this appliance function.

The objective is to exploit the combination of Occupancy inference along with the disaggregation data in order to recognize the activity of an occupant. The obtained information can be used for further purposes like the provision of more personalized hints based on occupants activities and consumption.

1.2 DELIVERABLE STRUCTURE

This document is an accompanying document of the code developed for D4.3 within Task T4.1, and describes the algorithmic approaches that have been developed for activity estimation and recognition. The State-of-the-art related work was presented in Section 2 at Deliverable D4.1 “First Activity Profiling and Matching Detector.

To begin with, Section 2 presents the way data was collected and analysed for the consumption disaggregation. Moreover, the next section introduces the basic components and features of the Activity Inference engine. Consequently, the final section discusses the experimental set, along with the results that are demonstrated based on a ground truth.

2 DATA COLLECTION AND ANALYSIS

In order to infer the daily activities of a resident using electricity meters, one has to disaggregate the total energy consumed and comprehend the operating state of the electrical appliances. The dataset that was used for the energy disaggregation training is one of the most used datasets for this purpose, REDD dataset (Figure 1). It contains power consumption data from real homes, for the whole house as well as for each device separately. More specifically, it contains 6 homes' data that were recorded for several weeks, while each house contains a different number of devices. Only 4 devices were used, as they were the only ones that were found to be common to the energy disaggregation data of the enCOMPASS users which are calculated on a daily basis. The data is intended to be used for the prediction of the consumption of each device separately, utilizing the whole-house signal. In the case of the current project, REDD dataset was used in order to create models that disaggregate the energy for each one of the users' devices, at every 15 minutes timestamp. Disaggregation of energy is the first step of the activity inference algorithm. The next step includes the estimation of the operating state of an electrical appliance within a household, based on its power consumption. This task requires extensive data collection. A dataset of one household over a period of 1 month has been collected from CERTH's smart house. As an initial step of this approach, we focused on the power consumption of electrical appliances in a kitchen environment. Data from the oven, the dishwasher, the washing machine, the fridge and the main consumption (includes the HVAC, lights, other appliances) of the entire floor have been collected.

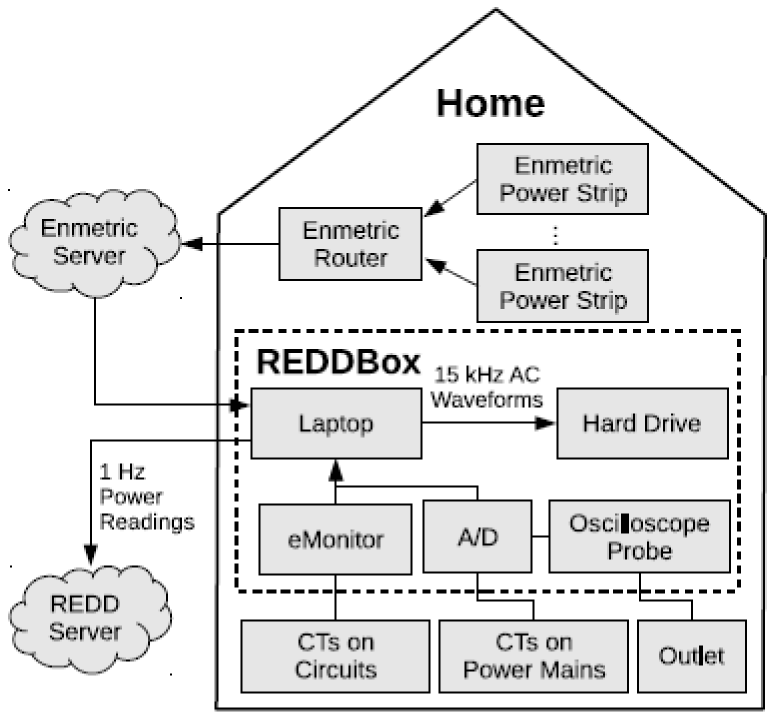

Figure 1: Data collection infrastructure

2.1 DATA COLLECTION INFRASTRUCTURE FOR DISAGGREGATION

In this section, we describe the preparation steps of the raw data, in order to make them ready for the analysis.

Regarding the REDD dataset, the electricity consumption data from the plugs were measured using a wireless plug monitoring system manufactured by Enmetric. As shown in Figure 2, the system consists of many power strips. Each strip has four outlets that are monitored independently and a router. The router is connected to the home's internet connection and handles all the data received from the home's wireless devices. All of the information of the low-frequency power data described above, is finally sent to a central server at the rate of 1 Hz.

Data collection for a whole circuit requires a more sophisticated solution, so the eMonitor (Figure 3) device, which is manufactured by Powerhouse Dynamics, is used. eMonitor involves current transformers connected to all of the individual circuits in the circuit breaker panel. The power consumption is sent to the central server using the monitor API, transmitting one reading per second, at the highest rate.

Figure 2: Enmetric plug monitoring system

Figure 3: Powerhouse Dynamics eMonitor device

Concerning the data collected for the device state detection, a smart electricity meter in the oven and main consumption devices of the smart house have been installed and are used for the acquisition of the desired information.

The electricity consumption of selected devices (fridge, dishwasher, oven, washing machine) has been measured via a wireless network of smart plugs that utilize ZigBee protocol (Figure 4).

Furthermore, the power consumption of the electrical kitchen appliances is monitored with a special built-in module. An aggregator application has been developed and installed. It requests the current power consumption from each module for given time steps and receives the corresponding messages including the measured power consumption of the connected appliance in Watts, the time stamp, the ID of the device, and stores the data directly into the database.

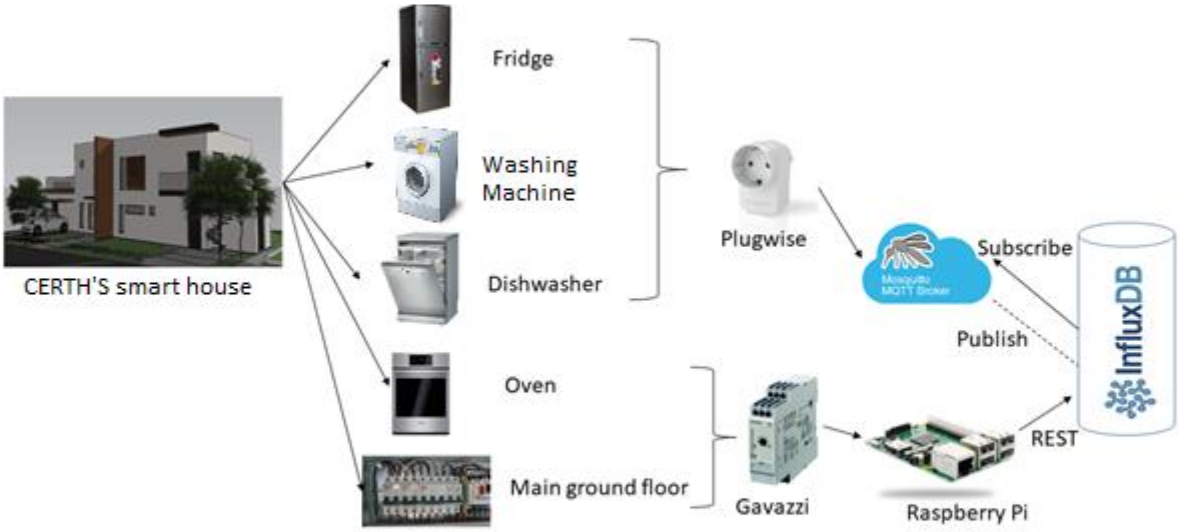

Figure 4: CERTH's Smart House data collection infrastructure

2.2 DATA PROCESSING FOR MACHINE STATE DETECTION

Data pre-processing is an essential step in the inference process. Data need to be transformed in an appropriate format in order to be processed from the algorithms and build the predictive models.

The REDD dataset is a well-structured dataset without inconsistences, so the pre-processing task in this case was relatively simple. The labels of each device were initially split, and finally merged with the respective consumption readings and timestamps for each device. In addition to that, timestamps had to be converted from unix time to a more human readable format. The goal was to divide the dataset into training and test set according to 'days', which is easier to perceive than 'seconds', used by unix time.

After retrieving the raw data for the smart house, a pre-processing step was performed in order to create the final aggregated dataset, which includes events per 1-minute intervals of all the measured features. It is worth mentioning that the raw data matrix was sparse duo to issues that emerged from the smart plugs functioning or the database. This problem was addressed by filling the missing values with the mode of the values of the last 15 minutes, until a new value was sent to the database. However, there were cases that no values were available on the database for one or more days, during the month of the measurements; hence, these days were disregarded from the dataset. The next step involved the aggregation of the features, consisting of power consumption in Watts for each of the four appliances (oven, fridge, dishwasher, washing machine). Firstly, the time (index) was rounded, due to a small time delay between the "subscription" and the "publish" of the event to the MQTT broker. Secondly, the state of operation (ON/OFF) of each device was manually labeled. The fridge was considered to be always "ON", even when the compressor was not operating. The rest of the devices were labeled as "OFF" (0) when the reading of the sensor was between 0 and 2.1348 Watts (a value around 2 Watts was considered as a 0 from the manufacturer) and "ON" (1) when the reading of the sensor was greater than 3 Watts.

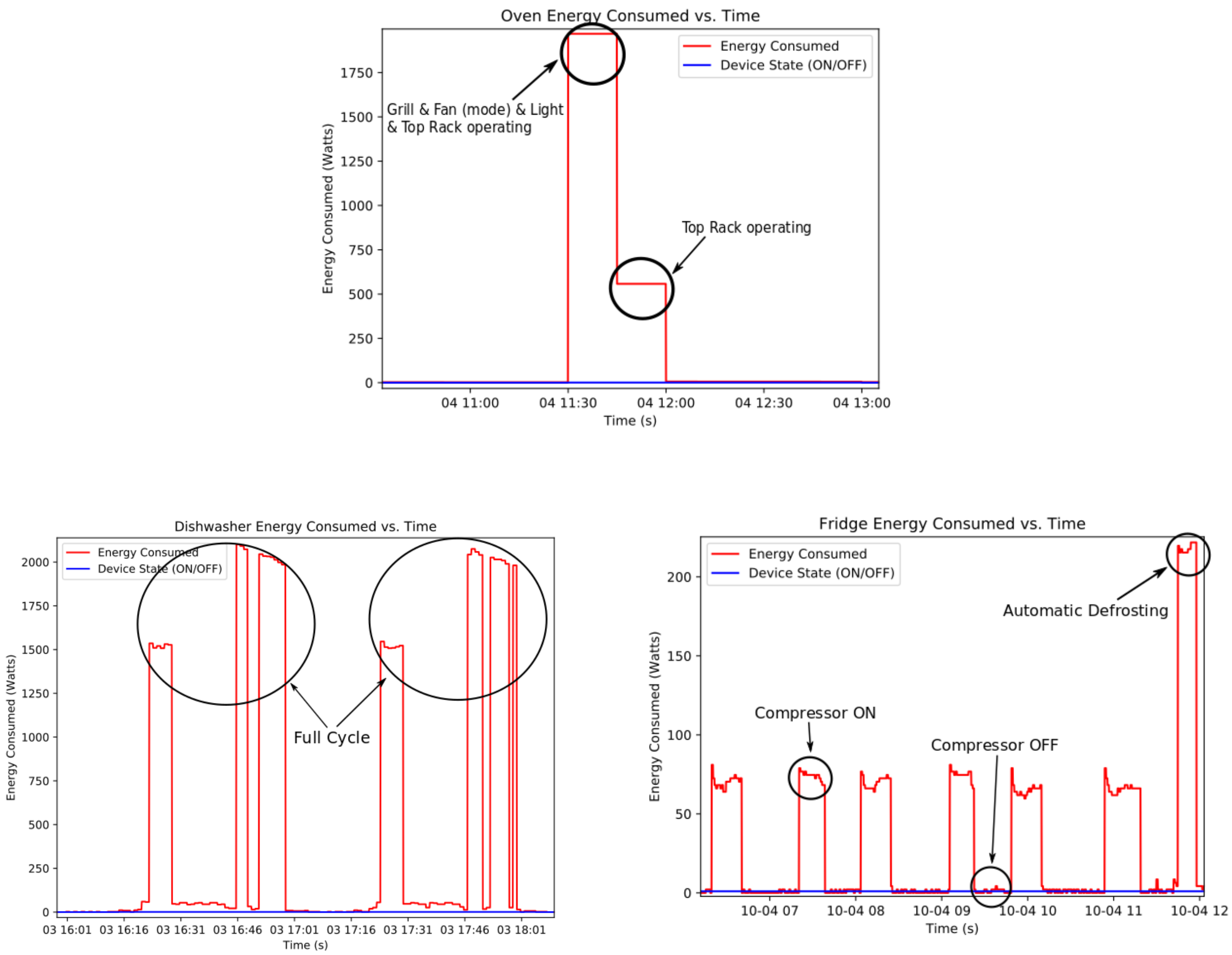

Figure 5: Power consumption pilots of selected appliances

Figure 5, shows the power consumption for three appliances from CERTH smart house. The operation of the dishwasher is periodic and therefore quite trivial to infer the activity of washing the dishes. The most challenging appliance was the fridge, since the fridge located in smart house was a state of the art machine, in terms of energy efficiency.

2.3 Data Collection For EnCompass

The activity inference algorithm is implemented for the enCOMPASS pilots, using data obtained from the databases of SES, SHF and WVT utilities. The data that are used belong to the following tables of the three pilot databases, which have the same logical schema:

- Occupancy inference: This table provides information regarding the home's occupancy, every 15-minutes timestamps. Motion sensors in combination with other features such as energy consumption are used to predict the presence of the occupants in the house. This knowledge is critical when predicting the activity of the occupants, as it can inform us for the absence of the occupants which can't be predicted by the use of consumption data.

Table 1: Occupancy Inferece Table

Timestamp	Occupancy
00:00	0
00:15	0
00:30	1
…	…
23:00	1
23:45	0

- Meter consumption: The total consumption of each house is stored on this table on 15-minutes timestamps. The measurements are obtained from smart meters which are installed in all of the enCOMPASS houses. Total consumption is later disaggregated by the algorithm to the consumption of each device separately. The algorithm aims at predicting the functioning state of each device that is used to make the final the final inference about the user's activity.

Table 2: Meter consumption table

Timestamp	consumption
00:00	1.927
00:15	0.130
00:30	3.3.76
…	…
23:00	0.531
23:45	1.235

- Disaggregation data: This table contains the disaggregated consumption of each house on a daily basis. It is filled by a service that runs daily on all of the pilots. The daily disaggregation data are used combined with the predictive models in order to obtain a more accurate result at the 15-minute disaggregation of the total power consumption.

Table 3: Dissagregation data table.

Date	Oven	Dishwasher	Washing Machine	Fridge
18-12-2018	2.4	2	1.9	2.4
19-12-2018	2.4	2	1.8	1.9
20-12-2018	v	0.8	0	2.4
…	…	…	…	…
30-12-2018	2.4	2.0	1.7	2.4
31-12-2018	2.4	2.0	2.0	2.4

- User profile: This table contains the information provided by the user on the questionnaire answered when entering the pilot. In the case of the activity inference algorithm it is used in order to find out if the user owns the devices that are utilized for the activity prediction. This way some certain activities are excluded from the users than do not own the corresponding devices (e.g. washing machine -> washing clothes).

Table 4: User profile Table

user_oid	electric_oven	Dishwasher	Washing Machine	Fridge
1	0	1	0	0
2	1	0	1	0
3	1	0	0	0
…	…	…	…	…
99	0	0	1	1
100	0	0	0	1

3 ACTIVITY INFERENCE

In this section, we provide a brief introduction into the field of machine learning and to the classifier used to detect the operation state of an appliance towards user-activity context inference. Next, the basic patterns of Occupancy Inference are presented and the prediction of whether an appliance is on operational mode or not based on consumption disaggregation. Finally, the results that are obtained by the application of the Algorithm are presented.

Iinitially Data is collected from the enCOMPASS Databases. Specifically, information from Occupancy Inference and meter consumption is gathered. Certain activities can be directly recognized by Occupancy Inference as described at Section 3.2. For the other activities the disaggregation Algorithm is applied using information from meter consumption table as mentioned at Section 2.1. Afterwards the Device state Algorithm is implemented as presented in Section 2.2, in order to implement the activity recognition.

The overall function of the Activity Inference is shown in Section 4.3.

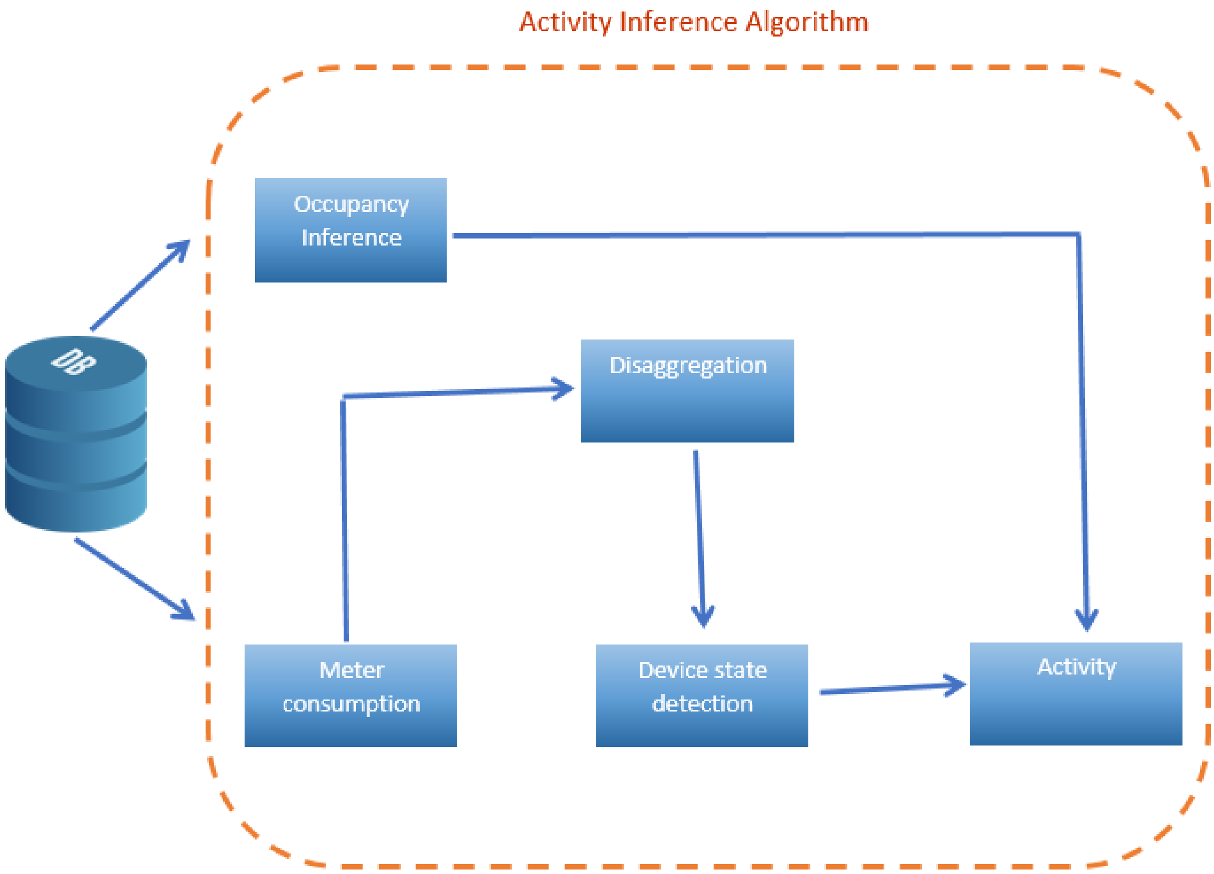

Figure 6: Activity Inference Algorithm functionality

3.1 MACHINE LEARNING AND CLASSIFICATION

Machine Learning is a technology for mining knowledge from data. A major focus of machine learning research is the automatic recognition of complex patterns and the intelligent decisions based on that data. ML is a scientific discipline that is concerned with the design and development of algorithms that allow computers to evolve behaviors based on data. The most common data mining tasks are Supervised, Unsupervised and Reinforcement Learning.

Supervised Learning: Supervised Learning is the type of learning that takes place when the training instances are labeled with the correct result, which gives feedback about how learning is progressing. In this case, the classes to which the training samples belong are known beforehand. *Unsupervised Learning:* In unsupervised learning, there is not any desired output, so no error signal is generated. It refers to the problem of trying to find a hidden structure in unlabeled data. Here, the input vectors of similar types are grouped together during the training phase. *Reinforcement Learning:* Reinforcement learning allows the machine to learn its behavior based on feedback from the environment. This behavior can be learned finally, or keep on adapting as time goes by. This automated learning scheme implies that there is little need for a supervisor who knows about the domain of application. The ML algorithm that it is used in this work so far is Random Forest (RF), which is explained in the following section.

Classification is a data mining function that assigns items in a collection to target categories or classes. Classification is a supervised learning in which individual item of data set is categorized to different groups based on prior knowledge. The characteristics of data play the important role in the performance of classifier depends [Kalousis04]. Classification is one of the most frequently studied problems by Data Mining and machine learning (ML) researchers. Classification derives a function or model, which determines the class of an object based on its attributes. A set of objects is given as the training set. A classification function or model is constructed by analyzing the relationship between the attributes and the classes of the objects in the training set. This function or model can then classify future objects. This helps us develop a better understanding of the classes of the objects in the database.

3.1.1 Decision tree regressor

A decision tree regressor is a model that breaks the data set into partitions while a tree is developed. A decision tree is a tree where each of the nodes represents an attribute and each of the branches represent a result, as shown in Figure 7. The model recursively splits the data set while each partition fits to a prediction.

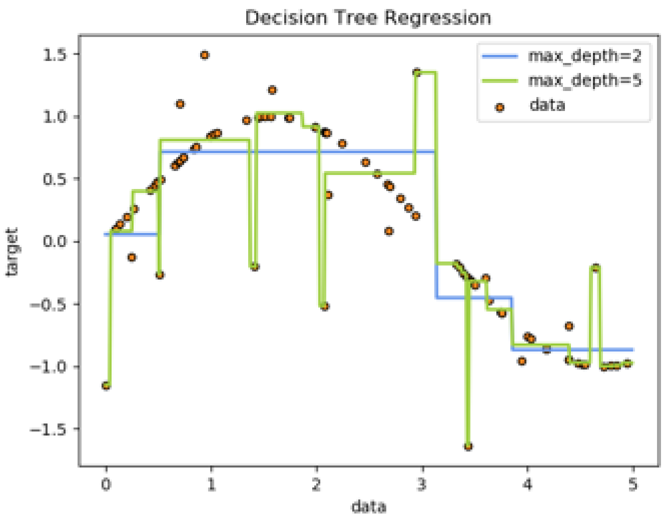

Figure 7: Decision tree

Decision tree regression is similar to decision tree classification (Figure 8)[1], however it uses Mean Squared Error or similar metrics instead of cross-entropy or gini impurity to determine splits.

3.1.2 Linear Support Vector Machines

Linear Support Vector Machines (LSVMs) classifier are one of the most distinguished classification and regression algorithms [Boser92]. Linear SVMs perform a linear classification based on Kernel functions, as shown in Figure 8.

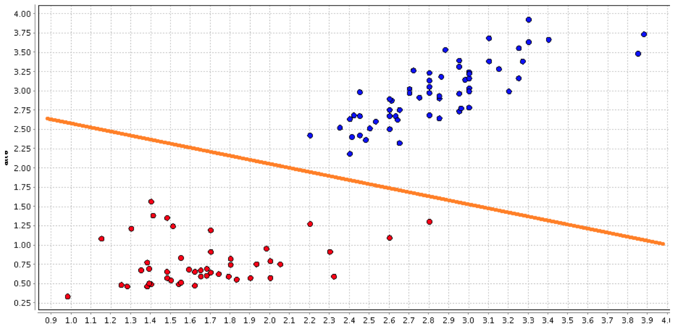

Figure 8: Linear Support Vector Machine

3.2 Occupancy and Energy Disaggregation

Activity recognition is only applied for household users. There are two main factors for this application:

- Household users are the only users that have a user_ profile table. In this table, there are entries of appliances (AC, electric_oven, microwave, electric_hot_plates, electric_kettle, coffee_machine, vacuum_cleaner, dishwasher, washing_machine_existence, washing_machine_energy_info, washing_machine_shared, tumble_dryer_existence, tumble_dryer_shared, dehumidifier, fridge, freezer, tv_set, hi_fi, desktop_computer, laptop_computer, gaming_set) with the indication if the user owns each or not (0: does not have this device, 1: have this device).
- To run the disaggregation Algorithm there must be information for the ownership or not of electric devices, by extension consumption data per device.

To begin with, each user is checked for being a "household user" from the user_profile table from the existence of electric appliances. No appliances state that the user is a "public user" or a "school user" and as a result, the activity Inference Algorithm is not applied for this type of user.

Occupancy inference is an Algorithm, which indicates if there is a movement, or not within the room the sensor is installed. This type of information is provided from the occupancy_inference table where 1 indicates that there is motion and 0 indicates that there is no motion at a this quarter of time. The information from the sensor is provided periodically every quarter. Each day 96 indications of motion are provided. This mix of 96 zeros and ones gives specific patterns during the day. There are some motifs within these formations that can be correlated with specific activities.

The first pattern that can be associated with an activity is the sequence of 96 zeros during a day. This sort of data indicates that no motion is detected from the sensor installed during a 24-hour period; as a result, the resident is not in the house. As shown in Figure 9, this activity is clarified as "Absent". The same activity

[1] https://scikit-learn.org/stable/auto_examples/tree/plot_tree_regression.html#sphx-glr-auto-examples-tree-plot-tree-regression-py

is recognized while the resident is not within the room or house, as a result there are continuous zeros at the occupancy_inference table.

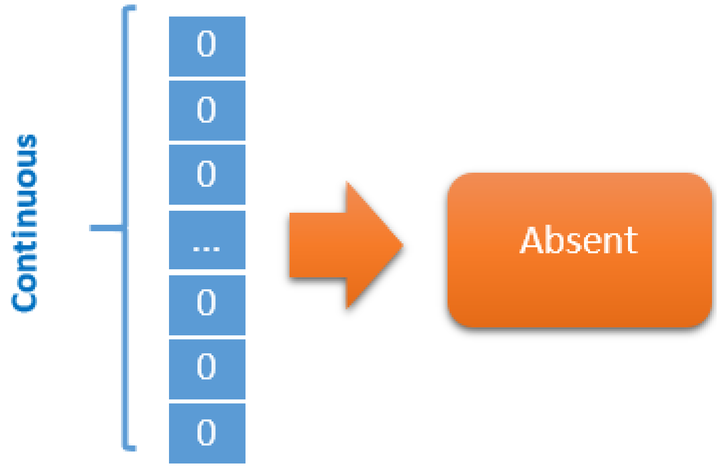

Figure 9: Absence of resident all day.

Another pattern of zeros that can be correlated with a specific activity, is sleeping. A sequence of continuous zeros from 22:00 to 8:30 is indicating that the resident is sleeping. This duration of time depends on the daily routine of each household user's habits. We can recognize that a person is sleeping from the repetition of zeros during the night as shown in Figure 10.

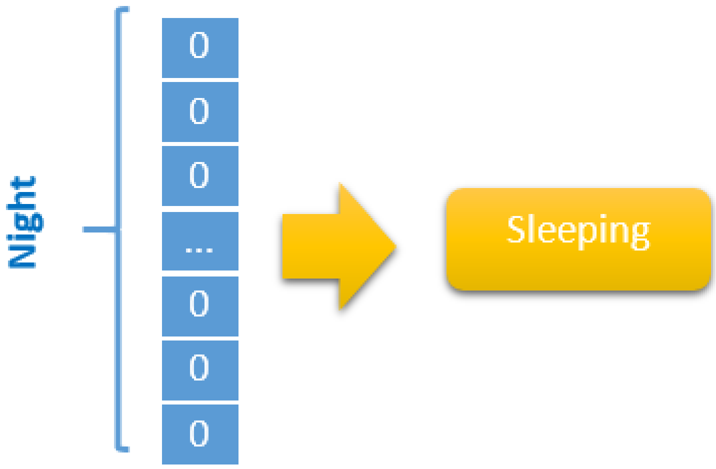

Figure 10: Resident Sleeping

The person is considered to be sleeping even though the sensor receives a motion for a timestamp or more during the night as long as the most signals are zero. As concluded from the research of Tamara et al [Tamara et al 10], in the homes of eight people, they got up at least on time per night. Figure 11, Figure 12 and Figure 13 represent examples of various sleeping patterns collected randomly from SES household users.

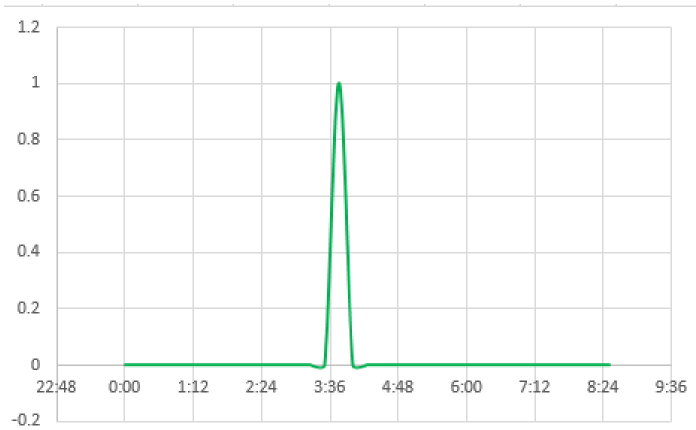

Figure 11: Sleeping pattern, the sensor shows a motion for one timestamp.

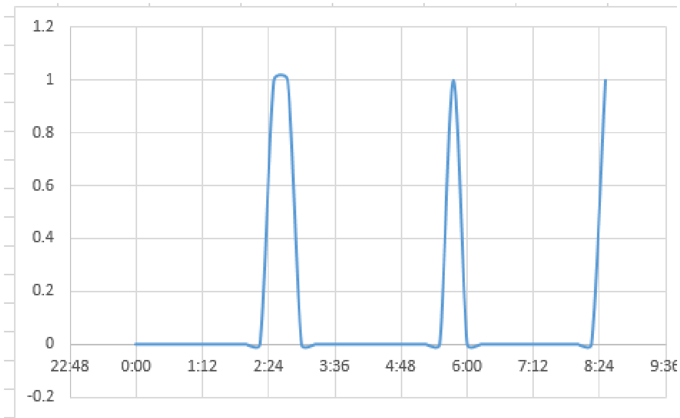

Figure 12: Sleeping pattern, the sensor shows a motion for three timestamps.

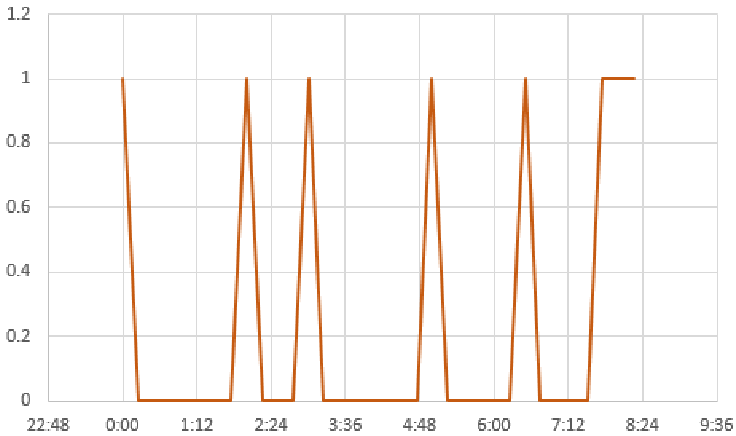

Figure 13: Sleeping pattern, the sensor shows a motion for six timestamps.

On the other hand, a sequence of continuous ones after a period of sleep or absence indicates that this person is awake. In this instance, the household resident is doing an activity which needs to be recognized from disaggregation consumption data, as shown in Figure 14.

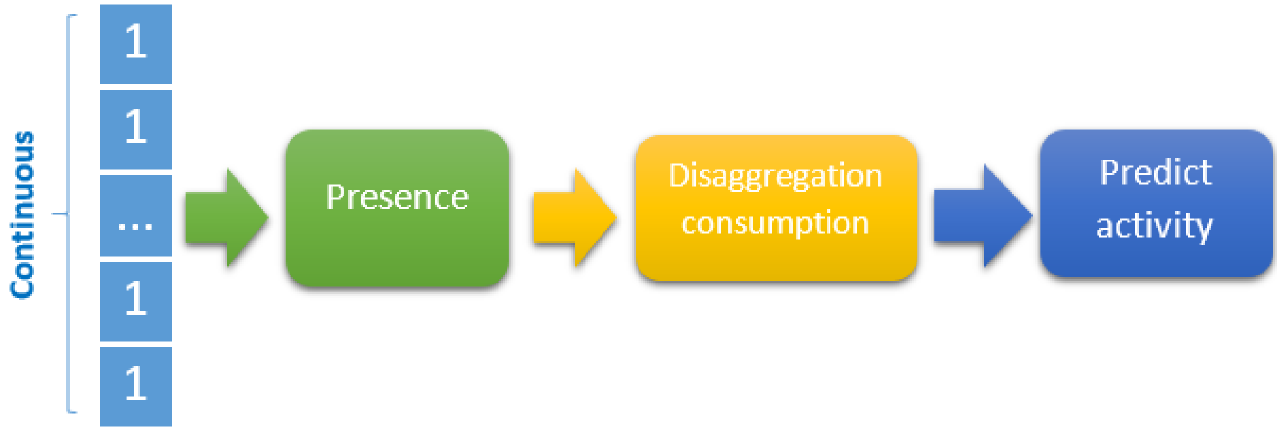

Figure 14: Presence pattern, the sensor shows a motion.

The process described above uses only occupancy data and is able to detect two activities: “Sleeping” and “Absent”. The timestamps that do not match either of those cases proceed to the energy disaggregation process. The models for each device are trained on the REDD disaggregation dataset using Decision Tree Regression. A small part of the dataset is not included in the training process and it is later used as a testing set being evaluated with MSE and MAE. The criterion under which the Decision Tree Regressor is parametrized, is the minimization of the MSE. The devices that are found to be common on the REDD dataset and on the enCOMPASS users’ homes are: electric oven, dishwasher, washing machine and fridge. The rest of the consumption is marked as “other devices”. The models developed are finally implemented on the enCOMPASS users’ homes and the total energy of each 15 minutes timestamp is disaggregated. Since there is already a daily disaggregation of the energy available, it is also taken into consideration at the calculation of the consumption values of each device.

When the power consumption is disaggregated for 96 timestamps (one day), the algorithms proceed to the device state detection stage. The prediction that needs to be done for the four devices is if they are on a functioning state (on) or on an idle state (off) at every 15-minute timestamp. The predictive models used at this stage have been trained on CERTH’s smart house devices. Those classification models are used in order to make a binary prediction regarding the on/off state of each device. Then, an overlapping time window of 3 timestamps goes across the whole vector of the daily consumption values, calculating the probability that each device is on or off at every timestamp. The models’ classification result combined with the probability give us the final prediction for the device state. Table 5 shows the final form of the device state table after running the algorithm for one user and one day.

Table 5: Device state table.

Timestamp	Oven	Dishwasher	Washing Machine	Fridge
00:00	0	1	0	0
00:15	1	0	1	0
00:30	1	0	0	0
…	…	…	…	…
23:00	0	0	1	1
23:45	0	0	0	1

The state of the devices is used for the inference of the following activities: “Cooking”, “Washing Dishes”, “Washing Clothes”. For the purposes of the current approach, the fridge is not semantically connected to any of the inferred activities so its state detection is not finally used. The consumption disaggregation of more kitchen devices, combined with the fridge, could lead to the inference of more activities commonly found in the literature (e.g. preparing breakfast) [Hyuncheol Seo16].

4 EXPERIMENTAL SETUP AND RESULTS

In order to implement the Activity Inference Algorithm data was collected from all the three pilot databases (SHF, SES and WVT) and for each database the experiment was established and activity recognition was performed. Initially users are distinguished to household, public users and school users as shown in Table 6. The Algorithm is implemented for 80 users from the SES Database, 129 users from the SHF Database and 154 users from the WVT Database.

Table 6: Number of users per Database

Database	Number of users	Household users	public users & school users
SES	95	80	15
SHF	157	129	28
WVT	173	154	19

4.1 PILOT DATA ANALYSIS

For each user, Occupancy data is collected from the occupancy_inference table. The patterns of occupancy differentiate from user to user depending on their daily routine. Examples of these patterns are presented in Figure 15, Figure 16 and Figure 17, where various daily programs can be recognised.

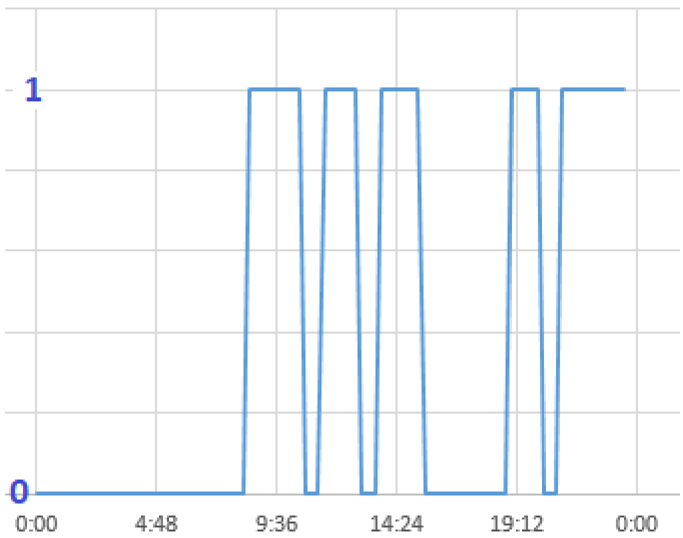

Figure 15: Household SES user occupancy pattern

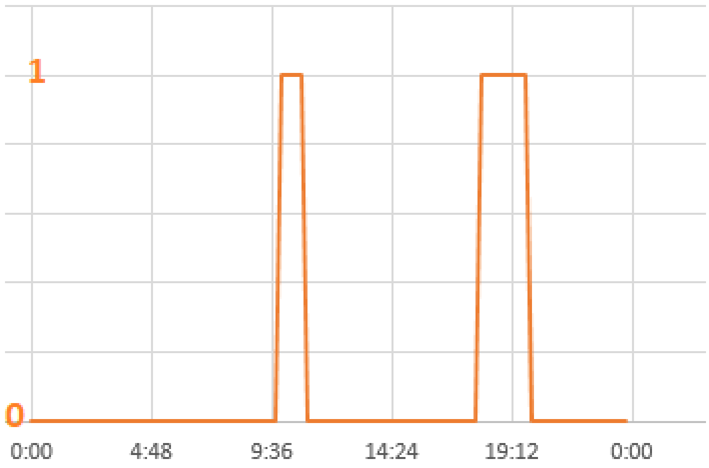

Figure 16: Household SHF user occupancy pattern

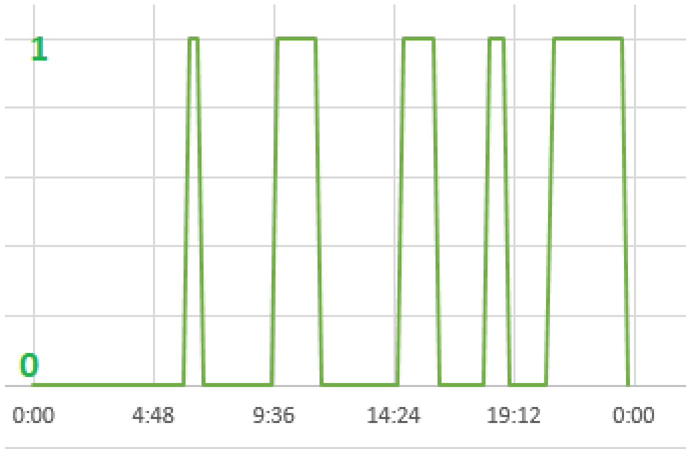

Figure 17: Household WVT user occupancy pattern

The Activity Algorithms is applied for three (3) devices electric oven, washing clothes machine and dishwasher. Not all the users own all the three electric appliances. Each user needs to be an owner of each of these appliances in order to disaggregate the total energy consumption to these components. In case a device exists on the user_profile table then the Algorithm for disaggregation is applied for this specific device. As shown in

Figure 18, all household SES users' own an electric oven, 77 out of 80 household SES users own a dishwasher and 68 out of 80 household SES users own a washing machine. As shown in Figure 19, 122 out of 129 household SHF users own an electric oven, 118 out of 129 household SHF users own a dishwasher and 121 out of 129 household SHF users' own a washing machine. Finally, as shown in Figure 20, 120 out of 154 household WVT users own an electric oven, 120 out of 154 household WVT users own a dishwasher and 127 out of 154 household WVT users own a washing machine.

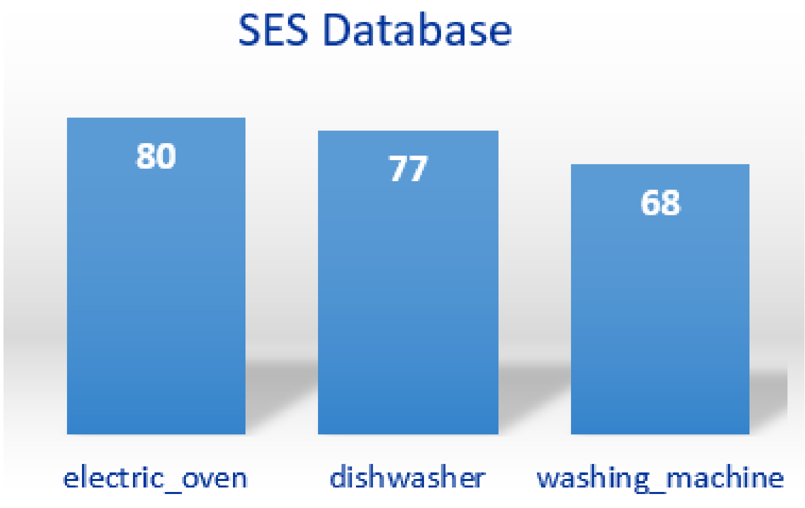

Figure 18: SES household users’ number of owners per electric device.

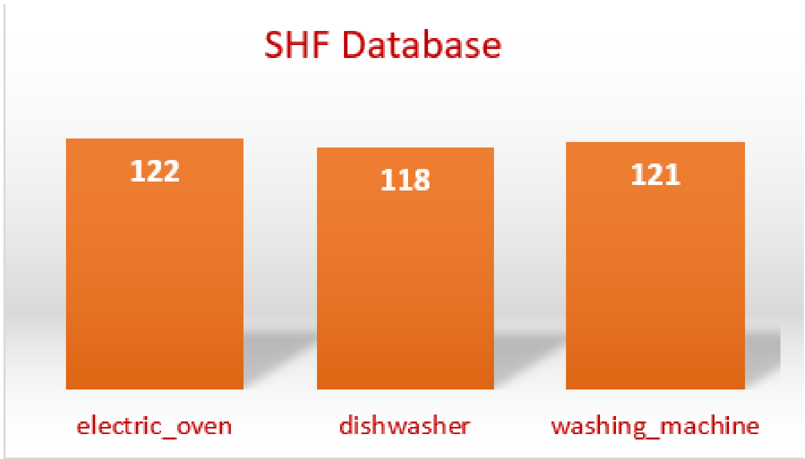

Figure 19: SHF household users’ number of owners per electric device.

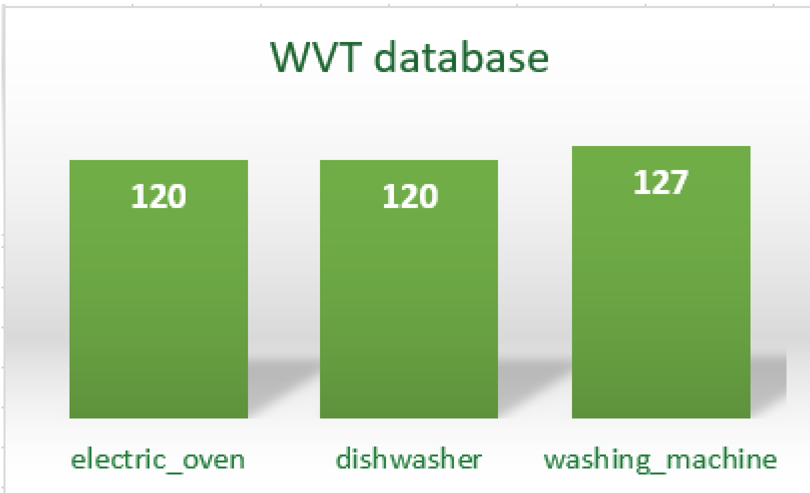

Figure 20: WVT household users’ number of owners per electric device.

Each user in the Database is associated with a smart meter, which transmits consumption data every quarter of time. This daily information for a user from SES Database (96 timestamps) is as shown in Figure 21. The increase of energy consumption indicates that an appliance is turned on that period.

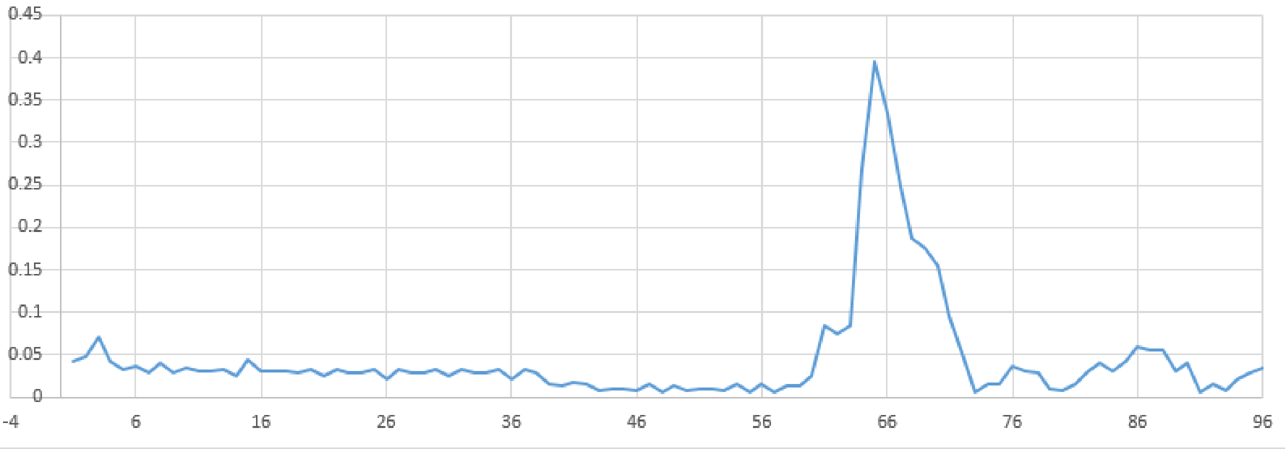

Figure 21: SES user daily consumption.

The patterns formed by the consumption data of a random sample of four household users are variegated, as shown in Figure 22. Each user has a different consumption motif; therefore, for each user for a specific day, data consumption data is collected.

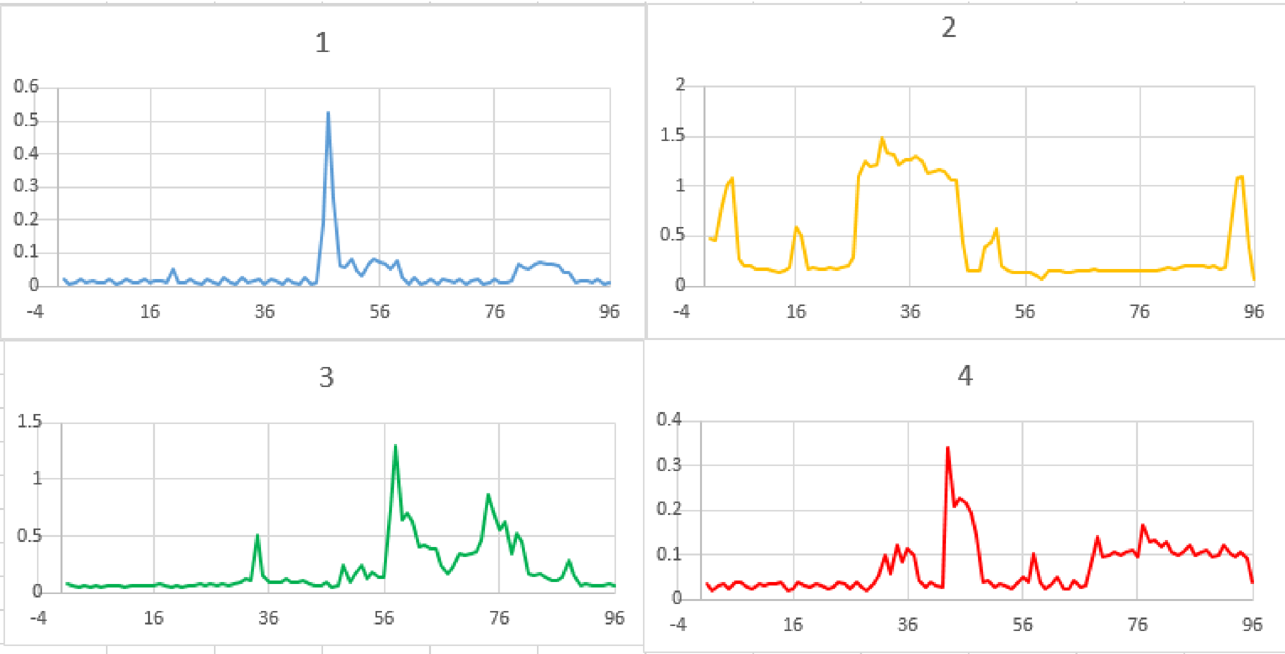

Figure 22: Random sample of daily consumption.

The disaggregation of total consumption data is implemented as mentioned above and is collected and validated with total disaggregation data from disaggregation_data table. An example of the partition of each appliance total disaggregation consumption is shown in Figure 23. The other partition includes all the other type of devices like lights, television, PCs that their consumption is not disaggregated to its components.

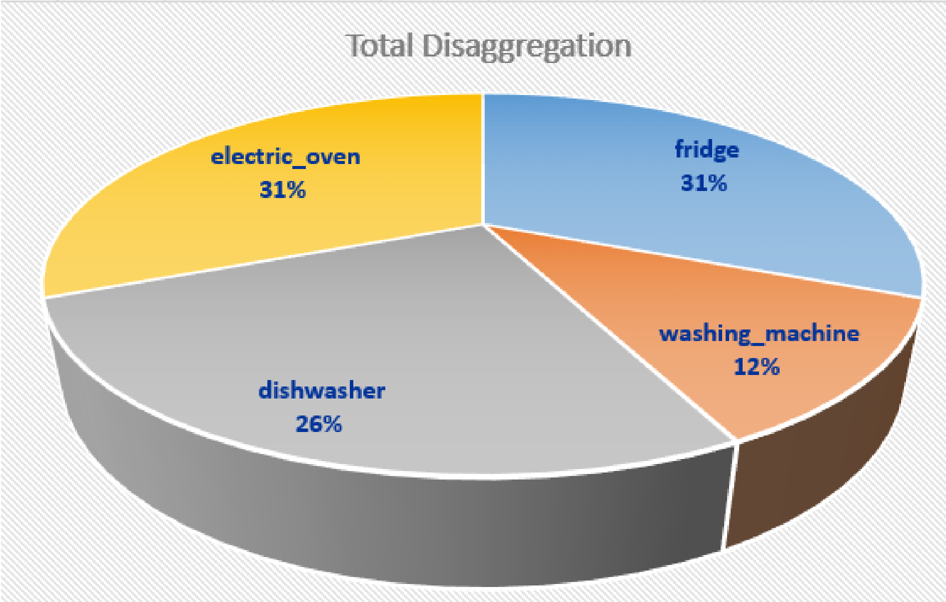

Figure 23: Example of Consumption ddisaggregation per device.

We disaggregate the energy consumption daily for the fridge, washing machine, dishwasher and the electric oven. Observing the average disaggregation of a day per Database (SES, SHF and WVT), as presented at Table 7, we may conclude that those four devices play a big part of energy waste in everyday use. As a result, to be able to predict the activities that are directly correlated with these, may be the first step towards behavioral change toward energy efficiency.

Table 7: Daily average disaggregation data

Database	fridge consumption	washing machine consumption	dishwasher consumption	electric oven consumption	Other appliances consumption
SES	11,90 %	3,72 %	7,73 %	7,73 %	68,92 %
SHF	24,98 %	5,58 %	2,68 %	10,05 %	43,29 %
WVT	25,27 %	6,61 %	12,68 %	13,25 %	42,19 %

To predict the activity of a user two factors are taken into consideration. Occupancy at a specific timestamp and consumption at the same timestamp. During the period that an occupant is active, the energy consumption is raising. On the contrary, while the resident is idle the consumption is in low levels. This correlation of Occupancy and consumption can be observed in Figure 24, collected randomly form a SES user. Meanwhile, there still could be a boost at the energy consumption, although the occupant is inactive due to the reason than he turned on an appliance and he left the room for more than a quarter or time, or even he went to bed.

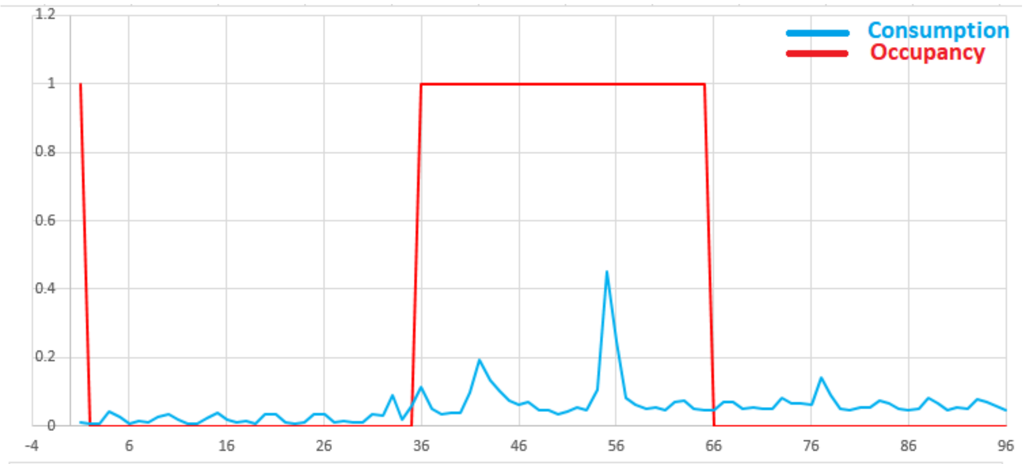

Figure 24: Correlation of Consumption and Occupancy

The objective of the Activity Inference Algorithm is to distinguish the alternation of the patterns of both Occupancy and Energy Consumption and recognize the activity, as shown in Figure 25. From 0:15 to 8:30 (timestamp 1 to 33) the activity recognition should be “sleeping” or “absent”, from 8:45 to 16:00 (timestamp 34 to 65) should be “doing an activity” and from 16:15 to 23:45 (timestamp 66 to 96) the activity recognition should be “sleeping “or “absent”.

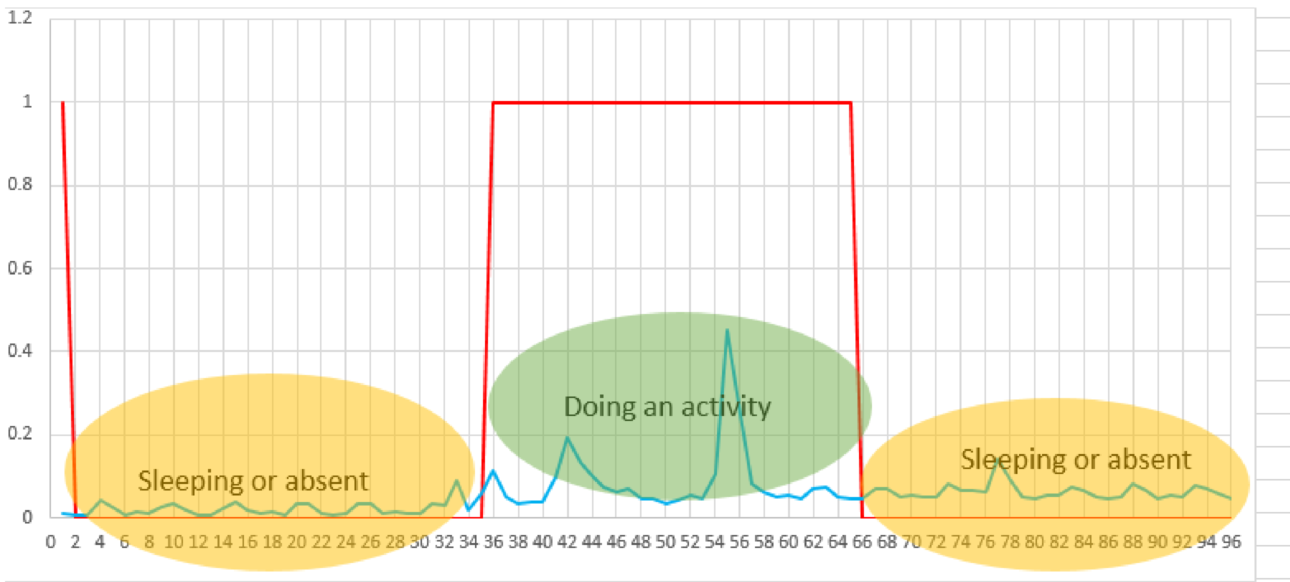

Figure 25: Activity recognition from Occupancy and Consumption chart.

Finally, during the period in which the user is active, so he is doing an activity, this activity is recognized from the Algorithm. The activities associated to the on/off state of the three devices, washing machine, dishwasher and electric oven, are *washing clothes*, *washing dishes* and *cooking*. The activities associated with no occupancy and low levels of consumption are *absent* and *sleeping*. In case the occupancy is 1 and consumption is zero, we assume there is a meter transmission error and there is an activity, but we are not able to predict it because we cannot compute disaggregation data and that is “Other activity”. Moreover, in case there is Occupancy and high levels of consumption, but the disaggregation result is zero for all the three devices, then there is an activity which is not directly correlated with the three devices and that is “Other activity”.

To sum up, the Activity Inference Algorithm is presented in Figure 26 and five (6) activities are predicted (Absent, cooking, washing dishes, washing machine, Other activity, Sleeping).

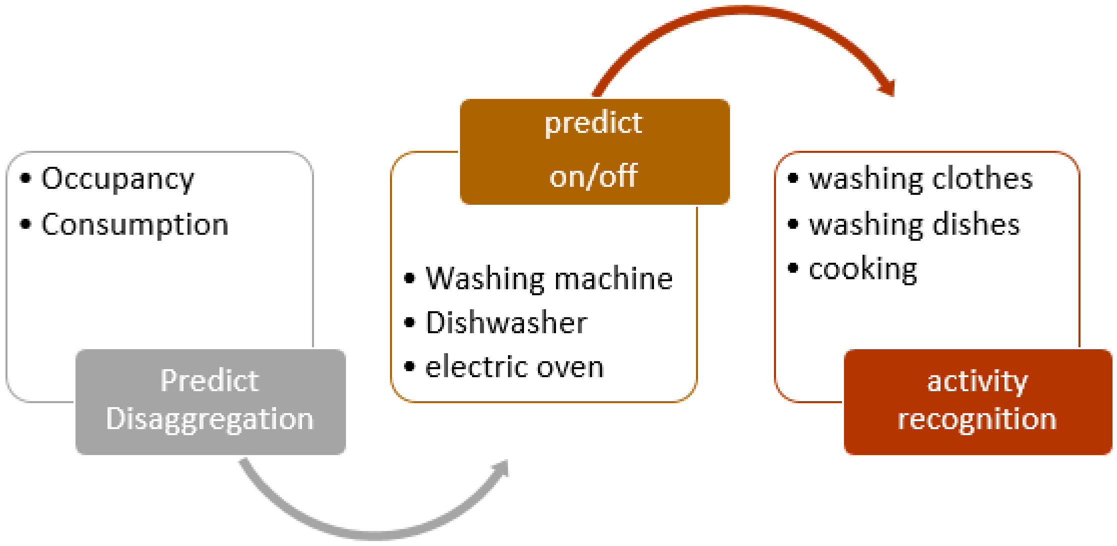

Figure 26: Activity Inference Algorithm.

4.1 Disaggregation Results

The models that were used for the disaggregation of the devices' consumption were trained on REDD dataset. One model was created for every device. Training set, validation set and test set contained 10 days, 6 days and 1 day of data respectively. The models are trained using the Decision Tree Regression algorithm. The algorithm runs for many different cases of "min_samples_split" and finally the optimal is selected after evaluating with Mean Squared Error and Mean Absolute Error. The disaggregation results for the dishwasher device are shown in Figure 27. The model is trained and tested on House 1 of the low frequency REDD Dataset.

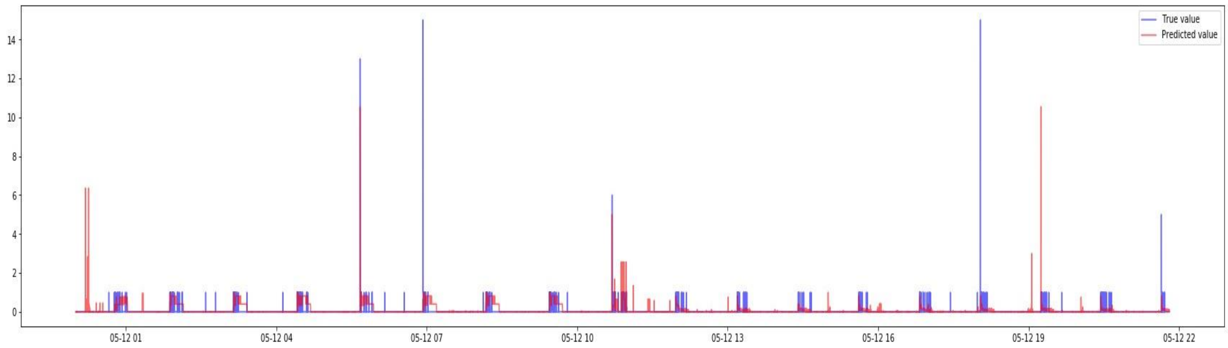

Figure 27: Real and predicted dishwasher disaggregated consumption

4.2 Device State detection results

A model was built for each one of the devices that were used at the activity inference algorithm. Three models were built in total (oven, dishwasher, washing machine) using Linear Support Vector Machine classifiers. 70% of the available data were used for training and the rest 30% were used for testing the models. The low dimensionality of the data, combined with the low complexity of the linear kernel, have as a result a model which is trained fast. Actually, there is no need for a more complex model, since the detection of the devices' on/off state is not a challenging task. The manual labeling of the devices' state is done according to some rules that seem to be easily recognized from the learning algorithm.

The general form of a confusion matrix, as well as the confusion matrices of the models of each device, are presented below. Confusion matrix provides deeper insights on the classification process, and is often used on unbalanced datasets where metrics like accuracy can be misleading. It reports the number of false positives, false negatives, true positives, and true negatives.

- True Positive (TP): Instances that are correctly classified as positive.
- False Positive (FP): Instances that are falsely classified as positive.
- True Neagative (TN): Instances that are correctly classified as negative.
- False Negative (FN): Instances that are falsely classified as negative.

Table 8 : Confusion matrix

		Actual	
		Device ON	Device OFF
Predicted	Device ON	**TP**	**FP**
	Device OFF	**FN**	**TN**

Table 9: Oven state detection confusion matrix

		Actual	
		Oven ON	Oven OFF
Predicted	Oven ON	485	0
	Oven OFF	0	9876

Table 10: Dishwasher state detection confusion matrix

		Actual	
		Dishwasher ON	Dishwasher OFF
Predicted	Dishwasher ON	53	0
	Dishwasher OFF	0	10308

Table 11: Washing machine state detection confusion matrix

		Actual	
		Washing Machine ON	Washing Machine OFF
Predicted	Washing Machine ON	21	0
	Washing Machine OFF	0	10340

The device state detection dataset is an imbalanced dataset, as in most cases the devices are not working (OFF state). For this reason, our model is evaluated by some more complex metrics that derive from the confusion matrices.

Precision is the ratio of predicted true positive cases to the sum of true positives and false positives and is given by the equation:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

Recall is the proportion of the true positive cases to the sum of true positives and false negatives and is given by the equation:

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

Accuracy is the fraction of the total number of predictions that were correct.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3}$$

Precision or recall are not enough metrics to describe the classifier's efficiency. Therefore, F_1 score is calculated as a combination of these two metrics. It is defined as twice the harmonic mean of precision and recall, and is the metric we will be most referring to.

$$F_1 = 2\frac{Recision * Recall}{Recision + Recall} \tag{4}$$

Matthews Correlation Coefficient (MCC) is a measure of quality of binary classification. A perfect prediction is represented by a coefficient of +1. On the other hand, a value of −1 indicates that no single instance was classified correctly. A coefficient of 0 represents a classification, which is no better than a random guess. The results from the device state on/off are presented at Table 12.

Table 12: Metrics describing classifier's efficiency

	Accuracy	Precision	Recall	F-score	Matthews Correlation
Oven	1.0	1.0	1.0	1.0	1.0
Dishwasher	1.0	1.0	1.0	1.0	1.0
Washing Machine	1.0	1.0	1.0	1.0	1.0

4.3 Activity Inference testing results

The Algorithm was applied for two days for a WVT household user. This user provided us the daily routine for each quarter for 2 days. For Day 1 the predicted activities 74 out of 96 timestamps were correct and for Day 2 82 out of 96 were correct, as seen in Table 13.

Table 13: Actual and predicted activity for Day 1 and Day 2

DAY 1		
TIME	ACTUAL ACTIVITY	PREDICTED ACTIVITY
0:00	Sleeping	Sleeping
0:15	Sleeping	Sleeping
0:30	Sleeping	Sleeping
0:45	Sleeping	Sleeping
1:00	Sleeping	Sleeping
1:15	Sleeping	Sleeping
1:30	Sleeping	Sleeping
1:45	Sleeping	Sleeping
2:00	Sleeping	Sleeping
2:15	Sleeping	Sleeping
2:30	Sleeping	Sleeping
2:45	Sleeping	Sleeping
3:00	Sleeping	Sleeping
3:15	Sleeping	Sleeping
3:30	Sleeping	Sleeping

3:45	Sleeping	Sleeping
4:00	Sleeping	Sleeping
4:15	Sleeping	Sleeping
4:30	Sleeping	Sleeping
4:45	Sleeping	Sleeping
5:00	Sleeping	Sleeping
5:15	Sleeping	Sleeping
5:30	Sleeping	Sleeping
5:45	Sleeping	Sleeping
6:00	Sleeping	Sleeping
6:15	Sleeping	Sleeping
6:30	Sleeping	Sleeping
6:45	Sleeping	Sleeping
7:00	Sleeping	Sleeping
7:15	Other activity	Sleeping
7:30	Other activity	Other activity
7:45	Other activity	Washing dishes
8:00	Other activity	Washing dishes
8:15	Other activity	Washing dishes
8:30	Other activity	Other activity
8:45	Other activity	Other activity
9:00	Other activity	Other activity
9:15	Other activity	Other activity
9:30	Other activity	Other activity
9:45	Other activity	Other activity
10:00	Other activity	Other activity
10:15	Other activity	Other activity
10:30	Other activity	Other activity
10:45	Other activity	Other activity
11:00	Other activity	Cooking
11:15	Other activity	Cooking
11:30	Other activity	Cooking
11:45	Other activity	Cooking
12:00	Other activity	Cooking
12:15	Other activity	Other activity
12:30	cooking	Cooking
12:45	cooking	Other activity
13:00	cooking	Cooking
13:15	cooking	Other activity
13:30	cooking	Cooking
13:45	Other activity	Cooking
14:00	Other activity	Other activity
14:15	Other activity	Other activity
14:30	Other activity	Other activity

14:45	Other activity	Other activity
15:00	Other activity	Other activity
15:15	Other activity	Other activity
15:30	Other activity	Other activity
15:45	Other activity	Other activity
16:00	washing clothes	Other activity
16:15	washing clothes	washing clothes
16:30	washing clothes	washing clothes
16:45	washing clothes	washing clothes
17:00	washing clothes	washing clothes
17:15	washing clothes	Other activity
17:30	washing clothes	washing clothes
17:45	washing clothes	washing clothes
18:00	washing clothes	Other activity
18:15	washing clothes	Other activity
18:30	Other activity	Other activity
18:45	Other activity	Other activity
19:00	Other activity	Other activity
19:15	Other activity	Other activity
19:30	Other activity	Other activity
19:45	Other activity	Other activity
20:00	Other activity	Other activity
20:15	Other activity	Other activity
20:30	Other activity	Other activity
20:45	Other activity	Other activity
21:00	Other activity	Other activity
21:15	Other activity	Other activity
21:30	Other activity	Other activity
21:45	Other activity	Other activity
22:00	Other activity	Other activity
22:15	Other activity	Other activity
22:30	Other activity	Other activity
22:45	Other activity	Other activity
23:00	Other activity	Other activity
23:15	Other activity	Other activity
23:30	Other activity	Other activity
23:45	Other activity	Other activity
DAY 2		
TIME	ACTUAL ACTIVITY	PREDICTED ACTIVITY
0:00	Sleeping	Sleeping
0:15	Sleeping	Sleeping
0:30	Sleeping	Sleeping

0:45	Sleeping	Sleeping
1:00	Sleeping	Sleeping
1:15	Sleeping	Sleeping
1:30	Sleeping	Sleeping
1:45	Sleeping	Sleeping
2:00	Sleeping	Sleeping
2:15	Sleeping	Sleeping
2:30	Sleeping	Sleeping
2:45	Sleeping	Sleeping
3:00	Sleeping	Sleeping
3:15	Sleeping	Sleeping
3:30	Sleeping	Sleeping
3:45	Sleeping	Sleeping
4:00	Sleeping	Sleeping
4:15	Sleeping	Sleeping
4:30	Sleeping	Sleeping
4:45	Sleeping	Sleeping
5:00	Sleeping	Sleeping
5:15	Sleeping	Sleeping
5:30	Sleeping	Sleeping
5:45	Sleeping	Sleeping
6:00	Sleeping	Sleeping
6:15	Sleeping	Sleeping
6:30	Sleeping	Sleeping
6:45	Sleeping	Sleeping
7:00	Sleeping	Sleeping
7:15	Other activity	Sleeping
7:30	Other activity	Washing dishes
7:45	Other activity	Washing dishes
8:00	Other activity	Washing dishes
8:15	Other activity	Washing dishes
8:30	Other activity	Other activity
8:45	Other activity	Other activity
9:00	Other activity	Other activity
9:15	Other activity	Other activity
9:30	Other activity	Other activity
9:45	Other activity	Other activity
10:00	Other activity	Other activity
10:15	Other activity	Other activity
10:30	Other activity	Other activity
10:45	Other activity	Other activity
11:00	Other activity	Other activity
11:15	Other activity	Other activity
11:30	Other activity	Other activity

11:45	Other activity	Other activity
12:00	Cooking	Other activity
12:15	Cooking	Cooking
12:30	Cooking	Cooking
12:45	Cooking	Cooking
13:00	Cooking	Cooking
13:15	Cooking	Cooking
13:30	Cooking	Other activity
13:45	Other activity	Other activity
14:00	Other activity	Other activity
14:15	Other activity	Other activity
14:30	Other activity	Other activity
14:45	Other activity	Other activity
15:00	Other activity	Other activity
15:15	Other activity	Other activity
15:30	Other activity	Other activity
15:45	Other activity	Other activity
16:00	Other activity	Other activity
16:15	Other activity	Other activity
16:30	Other activity	Other activity
16:45	Other activity	Other activity
17:00	Other activity	Other activity
17:15	Other activity	Other activity
17:30	Other activity	Other activity
17:45	Other activity	Other activity
18:00	Other activity	Other activity
18:15	Other activity	Other activity
18:30	Other activity	Other activity
18:45	Other activity	Other activity
19:00	Other activity	Other activity
19:15	Other activity	Other activity
19:30	Other activity	Other activity
19:45	Other activity	Other activity
20:00	Cooking	Other activity
20:15	Cooking	Other activity
20:30	Cooking	Other activity
20:45	Cooking	Other activity
21:00	Cooking	Other activity
21:15	Other activity	Other activity
21:30	Other activity	Other activity
21:45	Other activity	Other activity
22:00	Cooking	Other activity
22:15	Cooking	Cooking
22:30	Cooking	Other activity

22:45	Cooking	Cooking
23:00	Cooking	Other activity
23:15	Other activity	Other activity
23:30	Other activity	Other activity
23:45	Other activity	Other activity

Below, we present the general form of the confusion matrices, as well as the confusion matrices of the models of each activity. The false positives, false negatives, true positives, and true negatives are indicated as:

- True Positive (TP): activities that are correctly classified as positive.
- False Positive (FP): activities that are falsely classified as positive.
- True Neagative (TN): activities that are correctly classified as negative.
- False Negative (FN): activities that are falsely classified as negative.

The results are presented at Table 14, Table 15, Table 16, Table 17, and Table 18.

Table 14: Sleeping detection confusion matrix

Day 1		Actual		Day 2		Actual	
		sleeping	not sleeping			sleeping	not sleeping
predicted	sleeping	29	1	predicted	sleeping	29	1
	not sleeping	0	66		not sleeping	0	66

Table 15: Other Activity detection confusion matrix

Day 1		Actual		Day 2		Actual	
		Other activity	not Other activity			Other activity	not Other activity
predicted	Other activity	42	6	predicted	Other activity	45	10
	not Other activity	10	38		Not Other activity	5	36

Table 16: Cooking detection confusion matrix

Day 1		Actual		Day 2		Actual	
		Cooking	not Cooking			Cooking	not Cooking
predicted	Cooking	3	6	predicted	Cooking	7	0
	not Cooking	3	84		not Cooking	5	84

Table 17: Cooking detection confusion matrix

Day 1		Actual	
		Washing Clothes	not Washing Clothes
predicted	Washing Clothes	6	0
	not Washing Clothes	4	86

Table 18: Metrics describing classifier's efficiency

Day 1				
	Accuracy	Precision	Recall	F-score
Sleeping	0.99	0.97	1.00	0.98
Cooking	0.91	0.33	0.50	0.40
Other	0.83	0.88	0.81	0.84
Washing Clothes	0.96	1.00	0.60	0.75
Day 2				
	Accuracy	Precision	Recall	F-score
Sleeping	0.99	0.97	1.0	0.98
Cooking	0.95	1.00	0.58	0.74
Other	0.84	0.82	0.90	0.86

5 ALGORITHM USER INTERFACE

In this section, a simple user interface of the activity detector is displayed, which has been established for demonstration purposes.

Initially, the user is requested to enter certain information, which is the day, month and year we need to detect the activity. Moreover, he needs to provide the Database abbreviations for which the recognition will establish, as shown in Figure 28.

```
Enter day : 18
Enter month: 12
Enter year: 2018
selected date is: 18 12 2018
Enter Database Instance (SES,SHF,WVT): WVT
```

Figure 28: Initial user imports

There is a message displayed that connection to the given Database established. Eventually, the end-user must select the smart meter oid he wants, as shown in Figure 29

```
Succesfully connected to  wvt  DB ...

Application started!
====================================

Enter smart_meter_oid:
```

Figure 29: Smart meter oid selection

Finally, the activities results per timestamp are provided, as shown in Figure 30.

```
Printing activity for user  221
2018-12-18 00:00:00     Sleeping
2018-12-18 00:15:00     Sleeping
2018-12-18 00:30:00     Sleeping
2018-12-18 00:45:00     Sleeping
2018-12-18 01:00:00     Sleeping
2018-12-18 01:15:00     Sleeping
2018-12-18 01:30:00     Sleeping
2018-12-18 01:45:00     Sleeping
2018-12-18 02:00:00     Sleeping
2018-12-18 02:15:00     Sleeping
2018-12-18 02:30:00     Sleeping
2018-12-18 02:45:00     Sleeping
2018-12-18 03:00:00     Sleeping
2018-12-18 03:15:00     Sleeping
2018-12-18 03:30:00     Sleeping
2018-12-18 03:45:00     Sleeping
2018-12-18 04:00:00     Sleeping
2018-12-18 04:15:00     Sleeping
2018-12-18 04:30:00     Sleeping
2018-12-18 04:45:00     Sleeping
2018-12-18 05:00:00     Sleeping
```

```
2018-12-18 05:15:00     Sleeping
2018-12-18 05:30:00     Sleeping
2018-12-18 05:45:00     Sleeping
2018-12-18 06:00:00     Sleeping
2018-12-18 06:15:00     Sleeping
2018-12-18 06:30:00     Sleeping
2018-12-18 06:45:00     Sleeping
2018-12-18 07:00:00     Sleeping
2018-12-18 07:15:00     Washing dishes
2018-12-18 07:30:00     Washing dishes
2018-12-18 07:45:00     Washing dishes
2018-12-18 08:00:00     Washing dishes
2018-12-18 08:15:00     Other activity
2018-12-18 08:30:00     Other activity
2018-12-18 08:30:00     Washing dishes
2018-12-18 08:45:00     Other activity
2018-12-18 08:45:00     Washing dishes
2018-12-18 09:00:00     Other activity
2018-12-18 09:00:00     Washing dishes
2018-12-18 09:15:00     Other activity
```

```
2018-12-18 09:30:00     Other activity
2018-12-18 09:45:00     Other activity
2018-12-18 10:00:00     Other activity
2018-12-18 10:15:00     Other activity
2018-12-18 10:15:00     Washing dishes
2018-12-18 10:30:00     Other activity
2018-12-18 10:45:00     Other activity
2018-12-18 11:00:00     Other activity

....

2018-12-18 23:00:00     Other activity
2018-12-18 23:00:00     Sleeping
2018-12-18 23:15:00     Other activity
2018-12-18 23:15:00     Sleeping
2018-12-18 23:15:00     Washing dishes
2018-12-18 23:30:00     Other activity
2018-12-18 23:30:00     Sleeping
2018-12-18 23:30:00     Washing clothes
2018-12-18 23:45:00     Sleeping
Ending application... press any button
```

Figure 30: Printing of results

CONCLUSIONS

The present document describes the algorithms developed for the indoor activity inference. The method is based on the occupancy and the disaggregation data, moreover from the detection of the state of each of the devices correlated directly to an activity. The whole activity inference algorithm was presented in details.

For the algorithm application, data from all the pilots was used along with data from a household user. The algorithm was parameterized and extended with real-data from the pilot buildings and finally tested at a real house. Finally, the algorithm has been applied to all enCOMPASS pilot users.

REFERENCES

- [Breiman 01] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
- [Boser92] B.E. Boser, I.M. Guyon, V.N. Vapnik, “A training algorithm for optimal margin classifiers”, Proceedings of the Fifth Annual Workshop on Computational Learning Theory (ACM’92), pp. 144–152, 1992.
- [Hyuncheol Seo16] Seo, HyunCheol & Kim, Jungwoo & Hong, Won-Hwa. (2016). A Study on Power Consumption of Household Appliances for Life Conduct Activity of Single Household in University Town. Indian Journal of Science and Technology. 9. 10.17485/ijst/2016/v9i46/107201.
- [Kalousis04] A. Kalousis, J. Gama, and M. Hilario, “On data and algorithms: Understanding inductive performance,” Machine Learning, vol. 54, no. 3, pp. 275–312, 2004
- [Tamara et al 10] Hayes, T. L., Riley, T., Pavel, M., & Kaye, J. A. (2010). Estimation of rest-activity patterns using motion sensors. Conference proceedings : Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference, 2010, 2147-50.
- https://en.wikipedia.org/wiki/Hidden_Markov_model
- https://en.wikipedia.org/wiki/Conditional_random_field
- https://en.wikipedia.org/wiki/Support_vector_machine
- https://scikit-learn.org/stable/auto_examples/tree/plot_tree_regression.html#sphx-glr-auto-examples-tree-plot-tree-regression-py