D4.1 FIRST 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	**31/10/2017**
Actual delivery date	**31/10/2017**
Version	**1.0**
Reviewers	**SUPSI, WVT**

History of changes

Version	Date	Comments	Main Authors
0.1	19/09/2017	Defining Table of Content	CERTH
0.4	19/09/2017	First draft of the document	CERTH, WVT
0.6	26/10/2017	First full version of the document	CERTH, WVT
0.7	30/10/2017	Quality check and review	SUPSI, WVT
1.0	31/10/2017	Final version	CERTH

Disclaimer

This document contains confidential information in the form of the enCOMPASS project findings, work and products and its use is strictly regulated by the enCOMPASS Consortium Agreement and by Contract no. 723059.

Neither the enCOMPASS Consortium nor any of its officers, employees or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein.

The contents of this document are the sole responsibility of the enCOMPASS consortium and can in no way be taken to reflect the views of the European Union.

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
ARMA	Autoregressive Moving Average
CRF	Conditional Random Field
FN	False Negative
FP	False positive
HMM	Hidden Markov Model
ID	Identification
LED	Light Emitting Diode
MCC	Matthews Correlation Coefficient
ML	Machine Learning
MQTT	Message Queue Telemetry Transport
OOB	Out-Of-Bag
PIR	Passive Infra-Red
RF	Random Forest
SVM	Support Vector Machine
TN	True Negative
TP	True Positive

EXECUTIVE SUMMARY

Deliverable D4.1 “First Activity Profiling and Matching Detector” is specified in the “amended” GA description as follows:

“Initial prototype, with documentation, of the algorithms for detecting the type of uses’ activity in different indoor conditions”.

Its major goal is to explain the initial version of the algorithms for humans’ 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 analysing 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 the sequence can lead to the human activity inference.

Deliverable D4.1 provides a description of the algorithms for the activity inference in indoor environments:

- data pre-processing algorithms;
- device state inference algorithms;
- human activity inference algorithms (next version of the deliverable).

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.

The deliverable is structured as follows:

- Section 1 is the introduction of the deliverable;
- Section 2 presents an initial overview of the state-of-the-art techniques on the activity recognition and inference algorithms;
- Section 3 presents the description of data analysis/ pre-processing;
- Section 4 provides the description of the device state detection/ recognition algorithm;
- The final two Sections contain the Conclusions and References.

1 INTRODUCTION

Knowing the true activity of occupants in a building at any given time is fundamental for the effective management of various building operation functions ranging from energy savings to security targets, especially in complex buildings with different internal kind of use. Occupant's activities within the building varies throughout the day, therefore it is difficult to characterize the different activities in different time periods. In general, activity monitoring in buildings is of high interest, since activities significantly contribute to the performance of the building. Therefore, there is a need for detailed activity knowledge.

This document is an accompanying document of the code developed within Task T4.1, and describes the algorithmic approaches that have been developed for activity detection and estimation. State-of-the-art related work is presented in the next section. A mandatory stage of the activity inference is the correct identification of the device state, which is presented in details in this document. Preliminary results about the performance of the methods developed are also provided.

Human activity can be estimated using various sources, such as occupancy sensors, as well as energy consumption. In this report, we have focused on the development of an algorithm that will be able to extract the activities based on the energy consumption.

2 BACKGROUND – STATE OF THE ART

Human activity recognition plays a significant role in human-to-human as well as in human-to-device interactions. The human ability to recognize another person's activities is one of the main subjects of study of the scientific areas of computer vision and machine learning. As a result of this research, many applications, including video surveillance systems, human-computer interaction, and robotics for human behavior characterization, require a multiple activity recognition system. In the enCOMPASS project, the activity recognition approach will utilize only information from the energy consumption in the building.

One method that has been already utilized in bibliography us Hidden Markov Models (HMMs)[1] [Kim10, Nazerfard10]. The HMM is a probabilistic model that is used for generating hidden states from given observations. Given an input sequence of observations, $(\mathrm{x}_1, \mathrm{x}_2, \dots, \mathrm{x}_t)$, the corresponding hidden state sequence, $(\mathrm{y}_1, \mathrm{y}_2, \dots, \mathrm{y}_t)$ should be found. Historical observation data can be useful, in order to estimate some parameters of the model. The Hidden Markov Model adheres to two strict assumptions:

The future state depends only on the previous one. This means, the hidden variable at time instance t, y_t, depends only on the previous hidden variable, y_{t-1}. So, the conditional probability $P\left(\frac{y_t}{y_{t-1}}\right)$ should be computed.

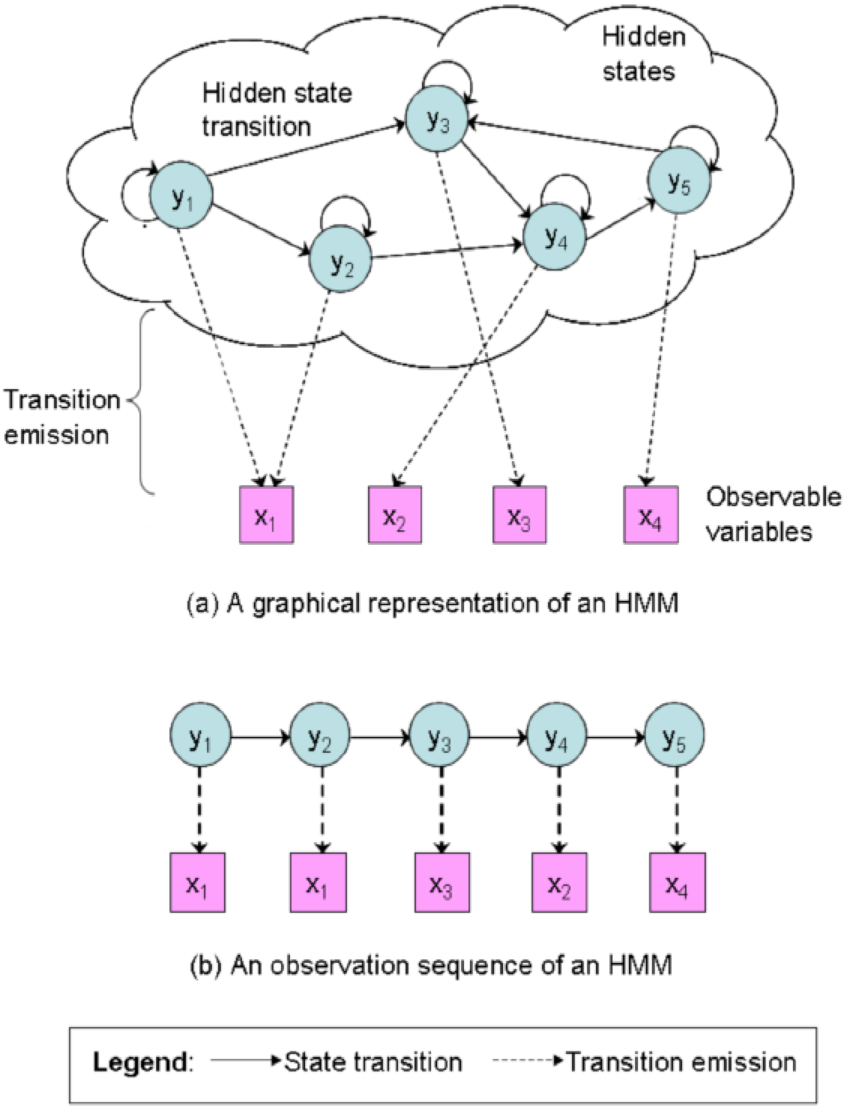
The observable variable at time t, x_t, depends only on the current hidden state y_t. So, the conditional probability $P\left(\frac{x_t}{y_t}\right)$ should be computed, which is independent from other observations or hidden states.

In order to estimate the users' activity, the hidden sequence y that maximizes the joint probability $P(x, y)$ should be found, which depends on the aforementioned conditional probabilities in the following way:

$$P(x, y) = \prod_1^T p(y_t/y_{t-1})p(x_t/y_t). \qquad (1)$$

Historical and training data are needed to find the transition probabilities from one possible hidden state to another, as well as the emission probabilities, which determine the probability of an observation, given a hidden state. An example of the description of the HMM is given in Figure 1.

[1] https://en.wikipedia.org/wiki/Hidden_Markov_model



(a) A graphical representation of an HMM

(b) An observation sequence of an HMM

Figure 1: A typical HM Model

Another method that is used for activity inference purposes is the so-called Conditional Random Fields (CRFs)[2] [Kim10, Zikos16]. They are related to HMMs but do not comply with their two strict assumptions that were mentioned before, thus being more flexible than HMMs. The CRFs are based on a conditional probability function, rather than a joint probability function. The main objective of this method is to model the conditional probability $P(Y/X)$, where Y is the set of output variables and the ones to be estimated, and X is the set of observable data. The two sets, X and Y, are linked through a feature function, $f(y_{t-1}, y_t, X, t)$. This feature function can either indicate the transition probabilities between two hidden states, y_{t-1}, y_t, or the state probabilities, i.e. the relationship between an observation and the corresponding hidden state. An expression that can represent $P(Y/X)$ is the following:

$$P(Y,X) = \frac{1}{Z(X)} exp(\textstyle\sum_i \lambda_i \sum_{t=1}^{n} f_i(y_{t-1}, y_t, X, t)), \qquad (2)$$

where $Z(X)$ is a normalization factor that refers to all of the different X observable sets [Sutton12], and ensures that $P(Y/X)$ will be a number whose minimum value is 0 and maximum value is 1. The coefficients λ_i correspond to the transition probabilities between the possible hidden states and can be obtained from

[2] https://en.wikipedia.org/wiki/Conditional_random_field

training or historical data. An example of the description of the CRF is given in Figure 2, where the coefficients λ_i are depicted in the subgraph on the left.

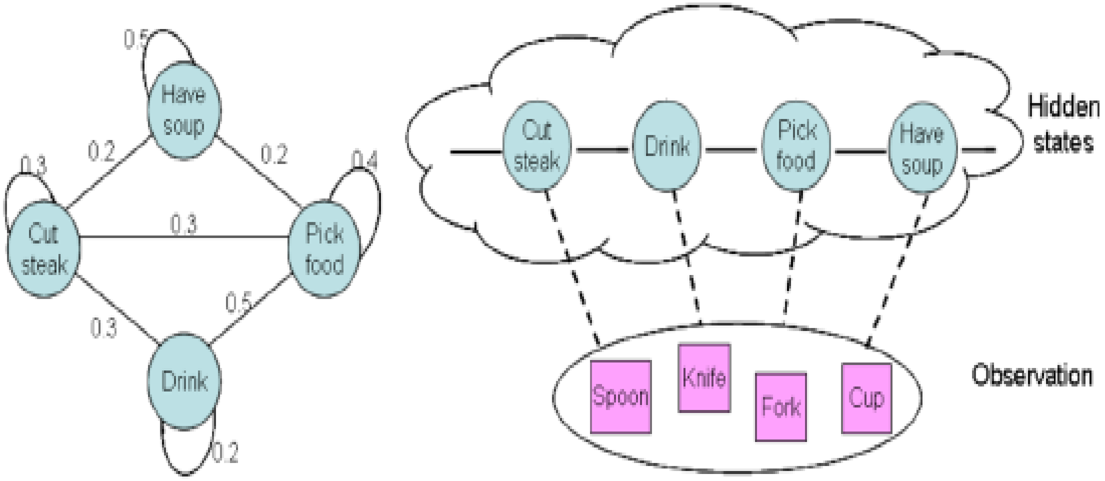


Figure 2: A typical CRF Model

Contrary to the HM Model that focuses in each pair (x_t, y_t) of observations and hidden states respectively, ignoring the previous and the next ones, the CRF Model considers the sequences of observations and hidden states (*X* and *Y*, respectively) as whole entities (In Figure 2, the observations are all in a circle, for instance). In Figure 2, one can see an example, where one eats and drinks, and based on the method that is utilized, the potential action can be inferred.

One further method that could potentially be used for human activity detection is the use of Support Vector Machines (SVMs)[3] [Landge15]. The input data is a number of observations that are represented as points in space. These observations must be separated based on some categories that correspond to the human activity that should be detected/ recognized. So, hyper-planes are used, which clearly separate the observations into groups. The equations of the hyper-planes (or lines, in case of two-dimensional space) must be computed, such that their distance from the nearest sample is maximized. This assists avoiding mistakes when classifying any sample into the correct group. A simple example in the 2D-space can be seen in Figure 3.

[3] https://en.wikipedia.org/wiki/Support_vector_machine

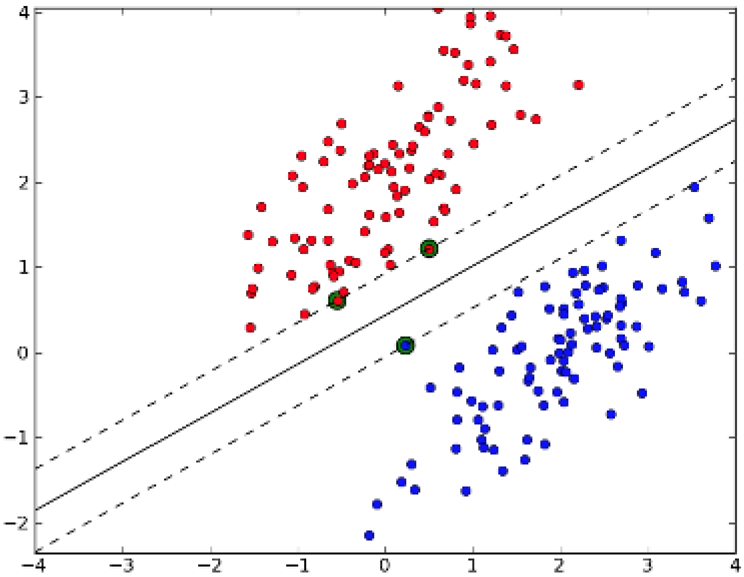


Figure 3: An example of SVMs

In Figure 3, the red and blue sample points, which are the SVMs, are divided into two groups (or classes), and the straight line is the one that best separates the data, since its distance from the intermittent lines is the maximum one. There are some cases, though, where the equation of the requested hyper-plane is not linear, because it cannot satisfy the maximum distance criterion. In these cases, kernel functions, such as a quadratic one [Landge15] are utilized. A characteristic example is the equation of a circle (see Figure 4). The SVM technique ignores data that are outliers.

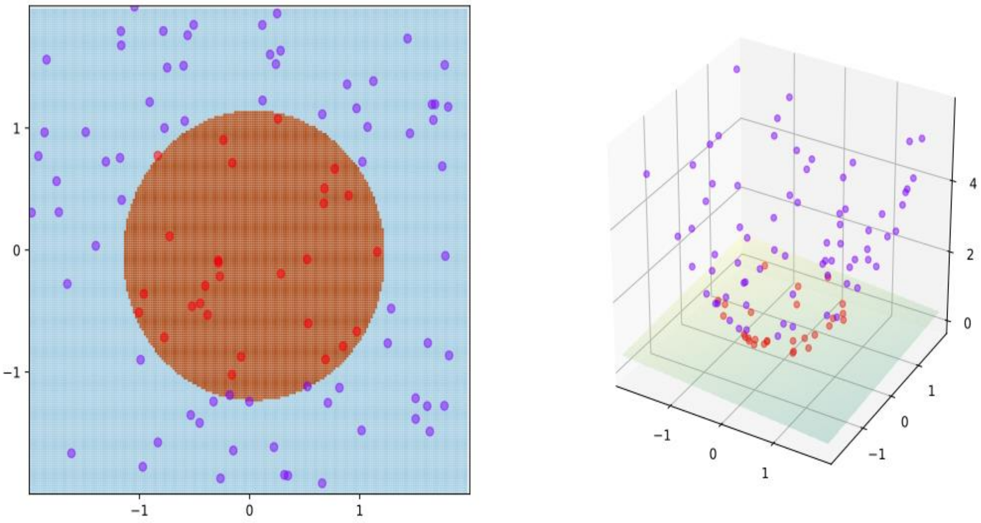


Figure 4: Kernel function $K(x,y) = x^2+y^2$

In addition, a simple use case for human activity detection is described in [Skocir16], where sensors (such as PIR) are utilized, to check some states. In [Skocir16], the pair of states that are used is the presence of a person in a room and the fact that the door is open or closed (for example, no presence and closed door are state s1, etc.). Each sequence of states (e.g. s1- s2- s3-s1) corresponds to a human activity.

Potentially, data from sensors could be exploited in favor of some or all of the aforementioned solutions, such as in [Zikos16]. Sensors are placed in specific places in each room, where the activities are considered to take place. Their on/off state is observed and stored, and it is matched to a timestamp. Taking a number of observations together as a group, and using the CRF method [Nazerfard10], a series of activities taken place at that time window can be inferred.

Moreover, the method of energy disaggregation, in order to infer an activity that happened at a specific moment [Batra15, Rao16] can also be exploited. Consumption data is collected daily per each device for a specific amount of time and as a result, the observations are divided into time periods and concluding to inferences about average or increased consumption over a specific period of time. The latter case could lead to assumptions about a user operating a device. As an example, a fridge may consume more energy than usual to restore its steady state, because at a given time its door was open.

In order to identify which device(s) is/are active, and thus detect a users' activity, one can use several methods. One of them is the use of SVMs, as mentioned above in [Landge15], but also in [Rao16], using a linear kernel, which is the simplest case when classifying data. Another method utilizes time series prediction, in order to predict future energy consumption, based on current consumption data, on past consumption data and the deviations of the latter. It is important to note that weather will affect the power consumption in a house. For this purpose, the Autoregressive Moving Average (ARMA) method is one of the most effective ones. It utilizes data collected in certain timestamps, weighted by corresponding coefficients, generally as follows:

$$x_{n+1} = w_n x_n + w_{n-1} x_{n-1} + \cdots + w_{n-k} x_{n-k}. \qquad (3)$$

The objective is to find the coefficients $w_i,\ i = n-k, \dots, n$ so that the above equation tends to zero, and thus the error is minimized.

Apart from the aforementioned, [Deshmukh15] which also deals with energy disaggregation and active device identification, proposes a method, where binary labels are assigned to devices in every sample taken (i.e. "1" when the device is on, and "0" when it is off). So, if the total number of devices is *N*, then there are *2^N* different cases. If the number of instances in the sampled data is *M*, then a matrix *A* is constructed, which is consisting of vectors that represent the states of all devices in all instances. Thus, the energy of each device can be estimated, and the devices that are active at a certain instance can be inferred.

Finally, segmentation [Xu17] could be applied in terms of the load profile (i.e. with respect to the time and the magnitude of its peaks) and its overall consumption. The objective is to classify large numbers of load profiles into representative consumption patterns. The load profile $l(t)$ is normalized as follows:

$$s(t) = \frac{l(t)}{\sum_{i=1}^{N} l(t)}. \qquad (4)$$

In the case of the overall consumption, *s(t)* is integrated over the time instances *t=1…N*:

$$I(n) = \int_0^n s(t)dt = \frac{n}{2N}(s(0) + 2s(1) + \cdots + 2s(n-1) + s(n)). \qquad (5)$$

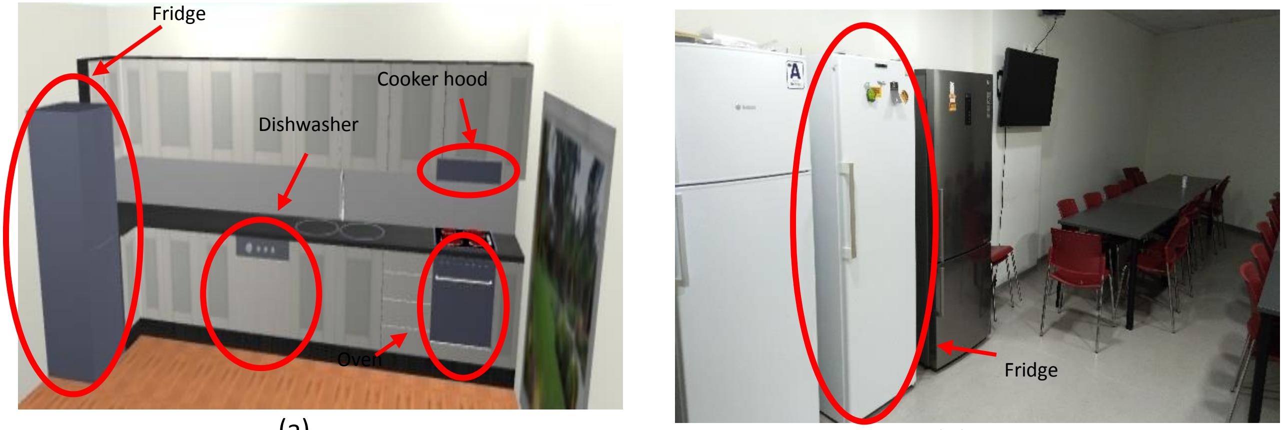
In the case of peak time, the condition illustrated below is checked:

$$\sum_{t=1}^{N}(s(t) - C_s(t))^2 \leq \theta \sum_{t=1}^{N} C_s(t)^2, \qquad (6)$$

where *$C_s(t)$* is the cluster function of *s(t)*, and *ϑ* is a threshold choice (0 ≤ θ ≤ 2).

3 Data Collection and Analysis

In order to infer the daily activities of a resident using electricity meters, one has to comprehend the operating state of an electrical appliance. Estimating the operating state of an electrical appliance within a household, based on its power consumption, requires an extensive data collection procedure. A dataset in two households over a period of 1 month has been collected. As an initial step of this approach, we focused on the power consumption of electrical appliances in a kitchen environment. From the first house (House_A) [**Figure 5**(a)], data from the oven, the cooker hood, the dishwasher, the fridge and the main consumption (includes the HVAC, lights, other appliances) of the entire floor have been collected. Regarding the second house (House_B) [**Figure 5**(b)], data from one fridge has been collected, since our goal was to check if it is possible to detect when a resident opens and closes the fridge door. This section describes the infrastructure for data collection and the pre-processing of the dataset.



(a)

(b)

Figure 5: *(a) Kitchen environment setup at CERTH' smart house, (b) Kitchen environment at CERTH/ITI's ground floor*

3.1 Data Collection Infrastructure

In this section, it is described the preparation step of the raw data, in order to be ready for the analysis. Figure 2 shows our data collection infrastructure.

A smart electricity meter in the oven and the main consumption panel of House_A has been installed in order to start acquiring the desired information. Also, the electricity consumption of selected devices (fridge, cooker hood, dishwasher and oven) has been measured via a wireless network of smart plugs that utilize the ZigBee protocol[4]. Furthermore, a special built-in module is used in order to monitor the power consumption of the electrical kitchen appliance. An aggregator application has been developed and installed. It requests the current power consumption from each module for given time steps, receives the corresponding messages, which include the measured power consumption of the connected appliance in Watts, the time stamp, the ID of the device, and then stores the data directly into the database.

[4] https://www.plugwise.com

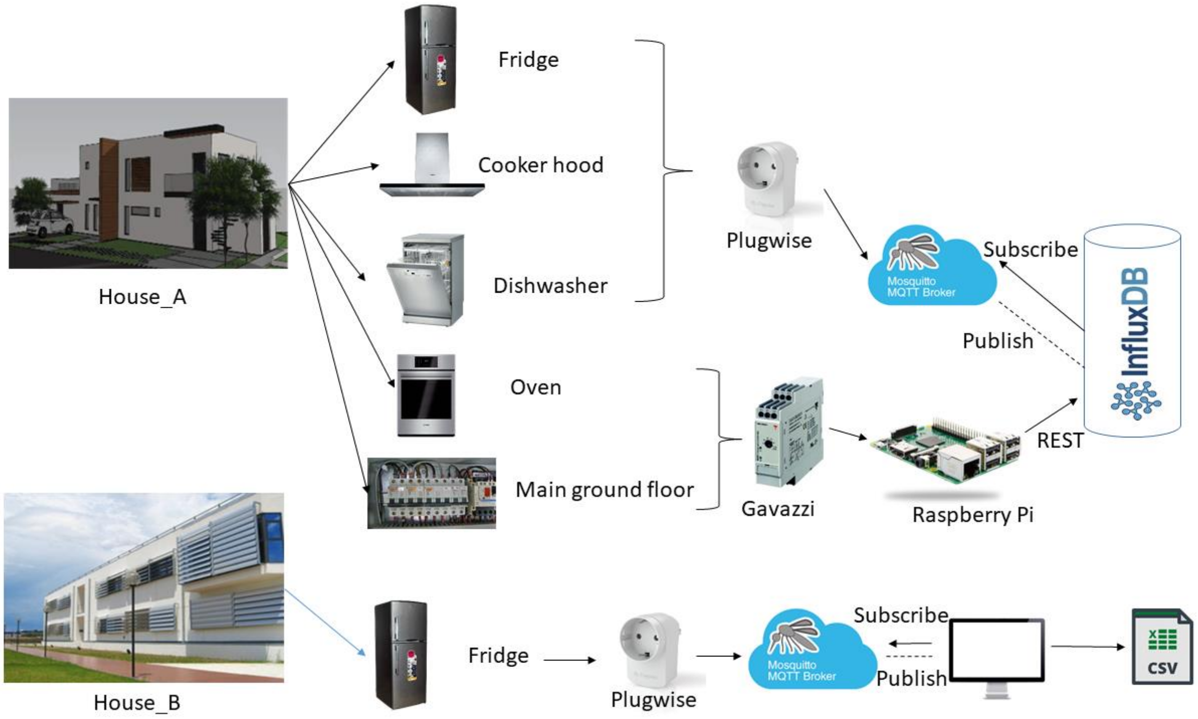


Figure 6: Data collection infrastructure

Additionally, for the second house (House_B), a similar procedure has been followed, using the smart plugs that collect the energy consumption data for the specific device under consideration.

3.2 Data Pre-Processing

Data pre-processing, is an essential step in the data mining process. After retrieving the raw data for House_A, a 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 due to technical issues with the smart plugs or the database, the sparsity of the raw data matrix should be overcome. In order to solve this problem, we filled 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 we would not get any values from the database for one day or more within the month of measurements; hence, we had to disregard these days from our dataset. Regarding House_B, we collected the electricity consumption of a fridge over a period of 10-days. The smart plug was sending data every 5-seconds, a time interval that was sufficient to detect whether someone opens and closes the door of the fridge.

The next step was to aggregate the features, consisting of electricity consumption in Watts for each of the four appliances of interest (oven, fridge, dishwasher, cooker hood). Firstly, we had to round the time (index), since there was a few millisecond delay between the "subscription" and the "publish" of the event to the MQTT broker. Secondly, the dataset regarding the "target feature" or the state of operation (ON/OFF) 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. Hence, the shape of overall dataset is 1440x4 (for each day, without taking into account the target feature).

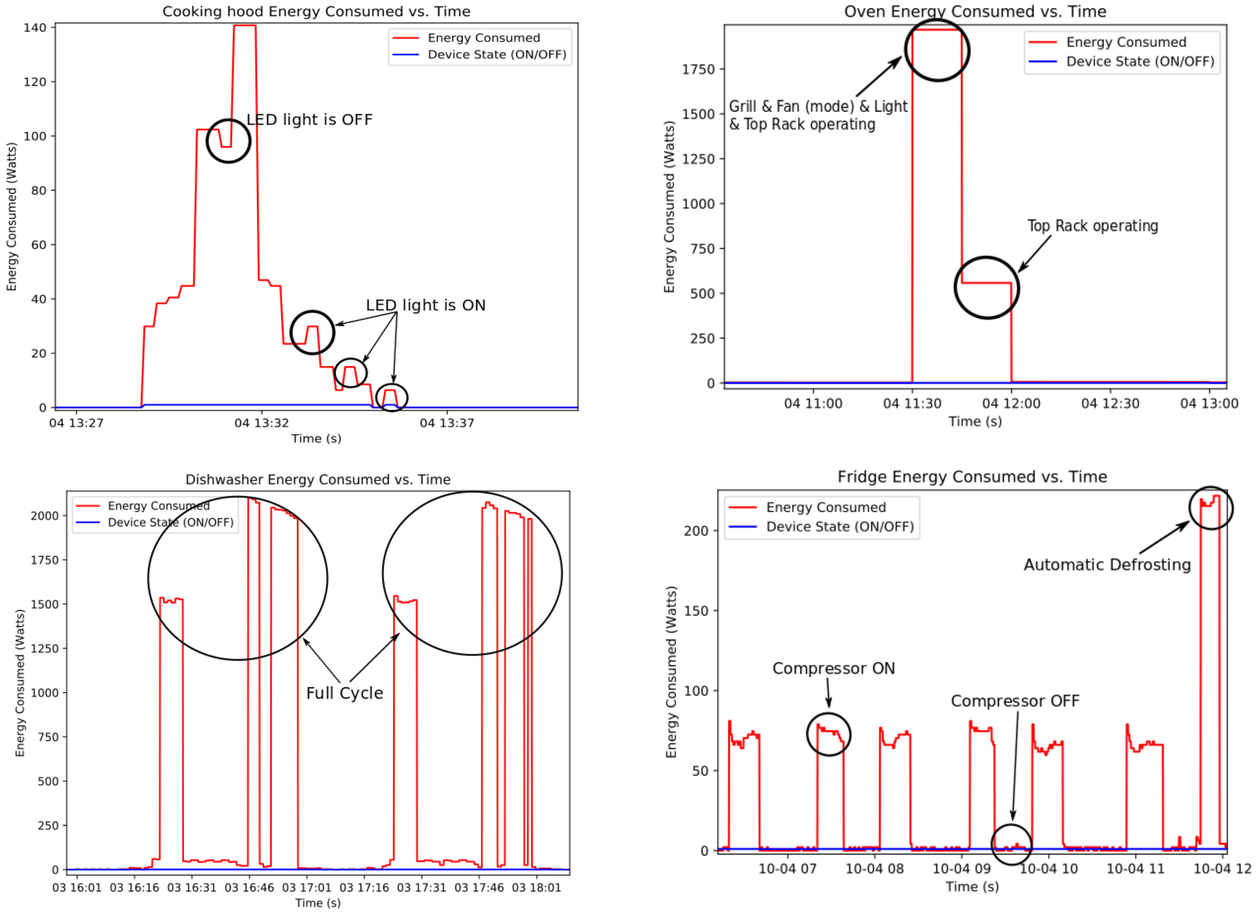


Figure 7: Power consumption pilots of selected appliances

Figure 7 shows the power consumption for the four appliances from House_A. It has been noticed that a difference in power consumption regarding the LED state of the cooker hood (measured 4 Watts) can be detected. In addition, after measuring the power consumption of the oven (smart meter sent data every 15-minutes), it should be checked if there are any “matching” times between the two devices, in order to infer the activity of cooking. Furthermore, 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 our goal was to detect the opening and closing of the door (based on the fridge light consumption). The fridge located in House_A was a state of the art machine, in terms of energy efficiency and consequently it was not possible to detect when the resident opened and closed the door, even when the data collection time was increased to 20 seconds.

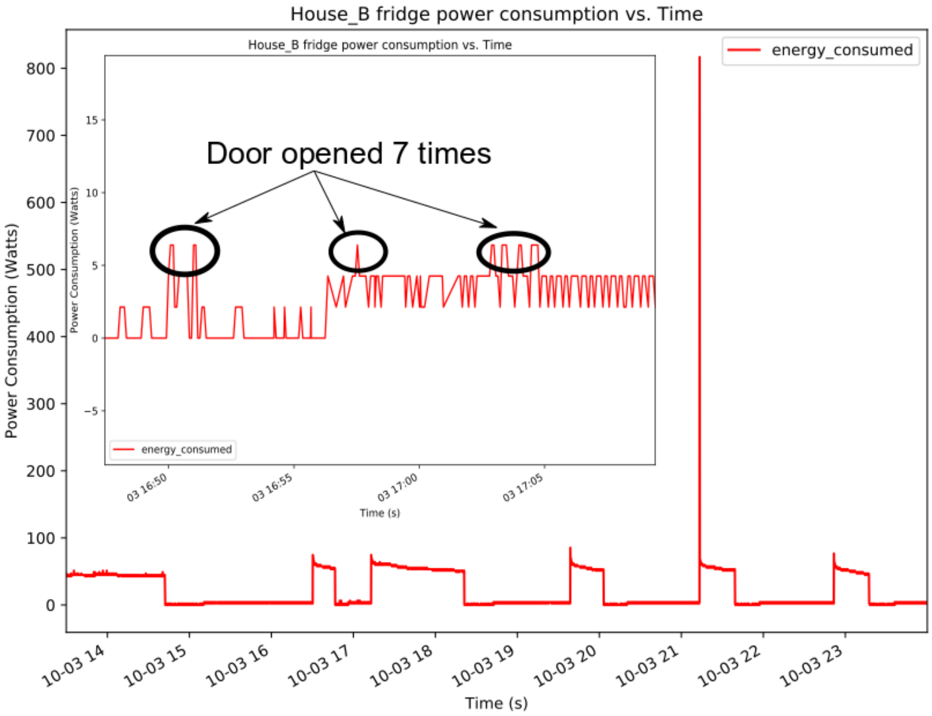


Figure 8: Power consumption of fridge from House_B

On the other hand, the fridge that is located in House_B was relatively older than the one in House_A. After sampling at 5-seconds, it is noticed that we could detect when a resident opens and closes the door of the fridge (light turned on) only when the compressor was not operating. In the opposite case, it was not possible to detect any activity, since there was no difference in the power consumption.

4 DEVICE STATE DETECTION APPROACH

In this section, it is provided a brief introduction into the field of machine learning and the classifier used as an initial approach to detect the operation state of an appliance towards user-activity context inference. Finally, our methodology and the results obtained are presented.

4.1 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 hidden structure in unlabeled data. Here, the input vectors of similar types are grouped together during training phase. *Reinforcement Learning:* Reinforcement learning allows the machine to learn its behavior based on feedback from the environment. This behavior can be learnt 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 a 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 plays 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.

4.2 RANDOM FORESTS

Random forests [Breiman 01] are the most widespread example for the concept of classification bootstrap aggregating (bagging). Bagging is used as a “meta-algorithm” in order to improve the stability and accuracy of classification algorithms. It is based on the idea of combining classifications of randomly generated training sets. In the case of random forests, the different bootstrap samples of the training dataset. The number of trees is selected to be 500 or 1000 usually. More specifically, random forests grow many classification trees. To classify a new entity with an attributes vector, put the attributes vector down each of the trees in the forest. Each tree gives a classification, and we say the tree “votes” for that class. The forest chooses the classification having the most votes (over all the trees in the forest). Each tree is growing as follows:

- If the number of entities in the training dataset is *n*, sample n entities at random but with replacement. This sample will be the training set for growing the tree.

- If the cardinality of x_i is p, a number of $\sqrt{p}$ at tributes is selected at random for each node of the tree, and the best split on these is used to split the node.
- Each tree is grown to the largest extent possible. There is no pruning.

The forest error rate (accuracy) depends on two things: (a) the correlation between any two trees in the forest. Increasing the correlation increases the forest error rate. (b) the strength of each individual tree in the forest. Increasing the strength of the individual trees decreases the forest error rate. An advantage of random forests is that they do not need k-fold cross validation. Instead, the out-of-bag (OOB) error estimate can be computed. More specifically, each tree is built using a different bootstrap sample from the original data. About one-third of the cases are left out of the bootstrap sample and not used in the construction of each tree. Put every entity left out in the construction of each tree down the tree to get a classification. In this way, a test set is obtained for each entity in about one-third of the trees. The OOB error is estimated in that test set.

4.3 Results

Four RF classifiers for each of the four appliances are implied, in order to detect their state (0: OFF, 1: ON). The feature vector has been resampled to two (2) minutes, by taking the mean value of the two previous 1-minute measurements. We selected this time, since it is sufficient to infer activities of the user for our future work. For the two-class classification scenario, in order to assess our models, we use the measures of precision, recall, accuracy, F_1 score and Matthews Correlation Coefficient (MCC), since our input data is heavily skewed towards one class (OFF states are more than the ON states), which are computed from the contents of the confusion matrix of the classification predictions (Figure 9). True positive and false positive cases are denoted as TP and FP, while true negative and false negative are denoted as TN and FN respectively.

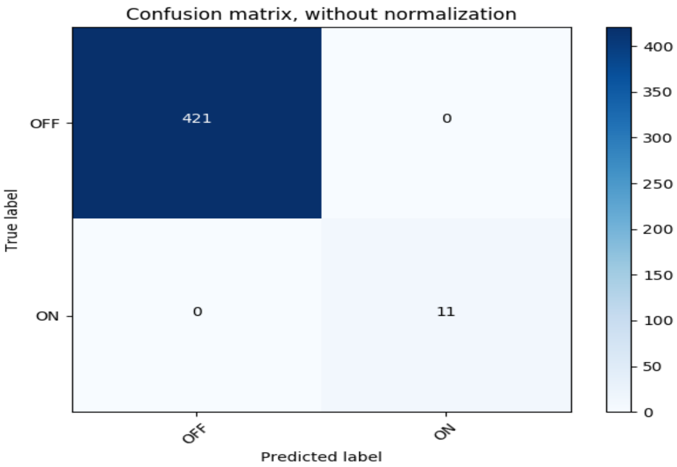


Figure 9: Confusion Matrix for the Oven

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

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

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

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

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

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{9}$$

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.

$$F_1 = 2\frac{Recision\ * Recall}{Recision + Recall} \tag{10}$$

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 MCC of a classifier is calculated as shown in Figure 9.

A 5-fold cross-validation for our experiments has been used, after splitting the dataset in a 70-30 percent for the training and testing, respectively, as shown in Figure 10.

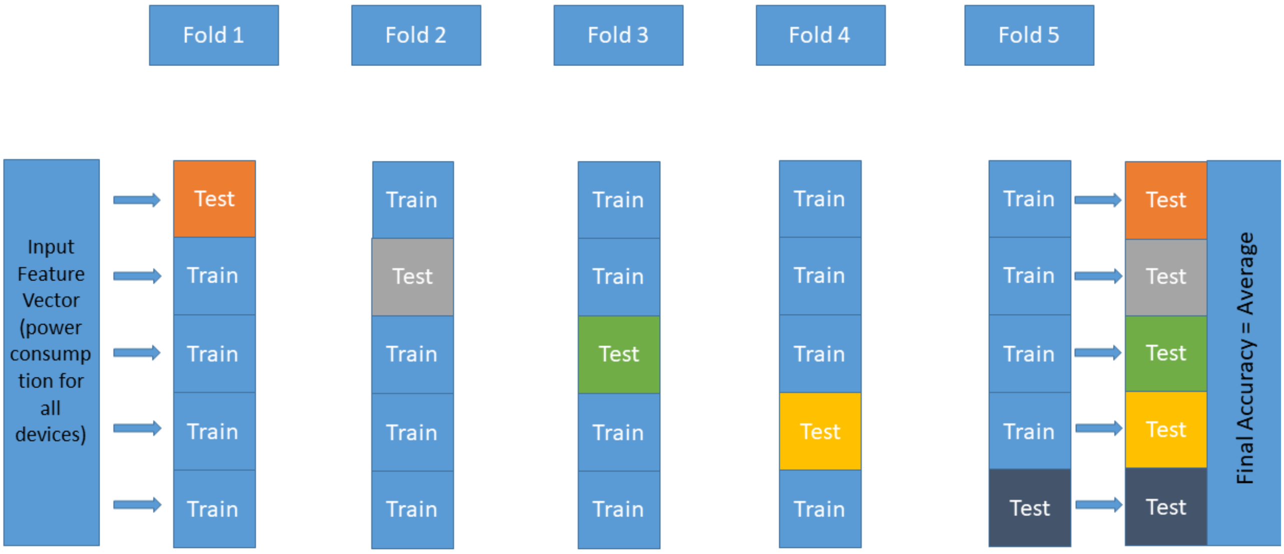


Figure 10: 5-fold cross-validation setup

Finally, Table 1 summarizes the prediction for the oven. The results for the dishwasher and the cooker hood are the same.

Table 1: *Obtained metrics for appliance state operation*

Random Forest (estimators)	PRECISION	RECALL	ACCURACY	F1-SCORE	MCC
50	100%	99%	100%	99.5%	+1
100	100%	100%	100%	100%	+1
500	100%	100%	100%	100%	+1

CONCLUSIONS AND FUTURE WORK

The present document described the methodology that will be utilized for the activity estimation. The method is based on the accuracy of the data and the correct detection of the state of each device. In this early approach, the algorithm for the correct detection of the devices' state is presented, while in the second version of the deliverable (Deliverable D4.3 "Final Activity Profiling and Matching Detector") the whole activity inference algorithm will be presented in details. Results showed that it is possible to achieve high accuracy.

The algorithms have been tested in CERTH's premises, while they will be tested, parameterized and extended with real-data from the pilot buildings. As a next step, we plan to perform experiments, in order to further improve the performance of the algorithms developed, and developing the core engine of the activity inference algorithm. Furthermore, the algorithm will be tested in real-life data, as well as with data from a long period of time. Finally, deep learning algorithms (e.g. Recurrent Neural Network) for system evaluation will be utilized.

REFERENCES

- [Batra15] N. Batra, A. Singh and K. Whitehouse, “If You Measure It, Can You Imporve It? Exploring the Value Of Energy Disaggregation”, Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments, pp. 191-200, Seoul, South Korea, Nov, 04 - 05, 2015
- [Breiman 01] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001
- [Deshmukh15] A. Deshmukh and D. Lohan, “CS446 Project: Electric Load Identification using Machine Learning”, May 8, 2015
- [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
- [Kim10] E. Kim, S. Helal and D. Cook, “Human Activity Recognition and Pattern Discovery”, IEEE Pervasive Computing, 2010, vol. 9, no. 1, doi:10.1109/MPRV.2010.7
- [Landge15] A. Landge, “Anomalous Human Activity Detection Using SVM and Active Leaning Based Approach”, International Journal of Innovative Research in Computer and Communication Engineering, vol. 3, no. 9, Sep. 2015
- [Nazerfard10] E. Nazerfard, B. Das, L.B. Holder, D.J.Cook, “Conditional Random Fields for Activity Recognition in Smart Environments”, Proceedings of the 1st ACM International Health Informatics Symposium, pp. 282-286, Arlington, Virginia, USA, Nov. 11 - 12, 2010
- [Rao16] K.M. Rao, D. Ravichandran and K. Mahesh, “Non-Intrusive Load Monitoring and Analytics for Device Prediction”, Proceedings of the International MultiConference of Engineers and Computer Scientists 2016, vol. I, IMECS 2016, March 16 - 18, 2016
- [Skocir16] P. Skocir, P. Krivic, M. Tomeljak, M. Kusek and G. Jezic, “Activity detection in smart home environment”, 20th International Conference on Knowledge Based and Intelligent Information and Engineering, Systems, Procedia Computer Science vol. 96, pp. 672 – 681, 2016
- [Sutton11] C. Sutton and A. McCallum, “An Introduction to Conditional Random Fields”, Foundations and Trends in Machine Learning, vol. 4, no. 4, pp. 267–373, 2011, DOI: 10.1561/2200000013
- [Xu17] S. Xu, E. Barbour and M.C. Gonzalez, “Household Segmentation by Load Shape and Daily Consumption”, 6th International Workshop on Urban Computing held in Conjunction with ACM KDD 2017 Conference, Halifax, Nova Scotia, Canada, August 2017
- [Zikos16] S. Zikos, A. Tsolakis, D. Meskos, A. Tryferidis, D. Tzovaras, "Conditional Random Fields - based approach for real-time building occupancy estimation with multi-sensory networks", Automation in Construction, Vol. 68, pp. 128-145, 2016, DOI: 10.1016/j.autcon.2016.05.005
- https://en.wikipedia.org/wiki/Hidden_Markov_model
- https://en.wikipedia.org/wiki/Conditional_random_field
- https://en.wikipedia.org/wiki/Support_vector_machine