(FP7 614100)

D5.4 Machine Learning for User Behaviour and Occupancy Analysis

05-07-2015 – Version 1.0

Published by the IMPReSS Consortium

Dissemination Level: Public

Project co-funded by the European Commission within the 7th Framework Programme and the Conselho Nacional de Desenvolvimento Científico e Tecnológico
Objective ICT-2013.10.2 EU-Brazil research and development Cooperation

Document control page

Document file: D5.4 Machine Learning for User Behavior
Document version: 1.0
Document owner: Eduardo Souto (UFAM)

Work package: WP5 - Data Storage, Analysis & Decision Support
Task: T5.4 Tool Support for Data Learning
Deliverable type: P

Document status: ☒ approved by the document owner for internal review
☒ approved for submission to the EC

Document history:

Version	Author(s)	Date	Summary of changes made
0.1	Wesllen Sousa	05-11-2015	First Draft
0.2	Thiago Rocha	06-09-2015	Second Draft
0.3	Eulanda Santos	06-15-2015	Second Draft Review
0.4	Lucas Lira Gomes	06-16-2015	Added the occupancy model section
0.5	Eduardo Souto	06-20-2015	Ready for internal review
1.0	Eduardo Souto	06-30-2015	Internal review comments incorporated. Final version ready for submission.

Internal review history:

Reviewed by	Date	Summary of comments
Marc Jentsch (FIT)	06/24/2015	Accepted with minor comments
Enrico Ferrera (ISMB)	06/29/2015	Accepted with minor comments

Index:

1. Executive summary

This deliverable describes scenarios to analyze user behaviour and occupancy using machine learning algorithms. Some practical scenarios that can be applied for diminishing electricity in a certain environment are proposed. All the steps to create these scenarios are covered, starting with the dataset generation until their implementation at the opera house application.

The first part of this deliverable shows the dataset that is necessary to use machine learning algorithms. It explains how a data analyst can choose, combine and extract data from the Opera House database to recognize the user activity pattern, which are used by the system to control the HVAC and Lighting system to save energy.

To develop the energy saving algorithms, first the relationship between appliances and user activities must be recognized. For this purpose, we maintain a decision tree and a vector model that describe the levels of correlations between the appliances and the activities. Based on this knowledge, we are able to create some recommendations to the users to save energy by switching off the unrequired devices during certain activities.

The second scenario is the energy consumption prediction. It uses regression algorithms to create the energy consumption models and predict them based on these models. The prediction algorithms are able to predict the energy consumptions of certain devices or places for the future time frame such as within the next hours, weeks or months. This strategy enables the data analyst or facility manager to simulate and compare energy consumptions of different devices that can save the overall energy consumptions.

The last scenario controls the HVAC system. The proposed application observes users occupation in an environment and tries to learn their habits and predict the best time to turn the HVAC system on or off to save energy and preserve the user comfort level.

These scenarios will be demonstrated using the Opera House Application that will be operated by the facility manager, and can be accessed by the visitors to retrieve information how much energy has been saved by the IMPReSS applications.

2. Introduction

2.1 Purpose, context and scope of this deliverable

The IMPReSS development platform consists of a set of technologies that help to build general-purpose applications that can access a plethora of data sources, such as information from the physical world (e.g. sensors) or even new data derived from it, as well as perform monitoring and control operations on complex systems. This is achieved through the definition of several tools and pre-defined modules that can be managed and combined in order to define a specific logic flow.

Data mining and machine learning tools are examples of algorithms designed to analyze relevant data. Consequently, these tools must be included into IMPReSS. However, in order to develop data mining and machine learning solutions to recognize patterns, it is necessary to understand deeply how different algorithms are modeled. IMPReSS focus on avoiding such a drawback by providing development tools for novice developers with no deep understanding on how the machine learning algorithms are developed. For this purpose, IMPReSS will offer a set of guided step-by-step decision support tools to allow developers to decide which data analytic technique must be used, based on the application requirements, as well as wizard and tutorial on how to use these techniques.

2.2 Background

IMPReSS is an EU-Brazil cooperation project aiming at providing a Systems Development Platform (SDP), which enables rapid and cost effective development of mixed criticality complex systems involving Internet of Things and Services (IoTS) and at the same time facilitates the interplay with users and external systems. The IMPReSS development platform will be usable for any system intended to embrace a smarter society. The demonstration and evaluation of the IMPReSS platform will focus on energy efficiency systems addressing the reduction of energy usage and CO2 footprint in public buildings, enhancing the intelligence of monitoring and control systems as well as stimulating user energy awareness.

The IMPReSS Platform consists of a set of technologies that help to build general-purpose applications accessing to a plethora of sources, such as information from the physical world, analyzing and fusing relevant data, and performing monitoring and control operations on complex system. The IMPReSS project aims at solving the complexity of System Development Platform (SDP) by providing a holistic approach that includes an Integrated Development Environment (IDE), middleware components, and a deployment tool.

3. Scenarios Overview

This section presents a solution that allows developers to implement rules for decision support systems. It shows how a dataset can be created and used by a machine learning algorithm. To create the datasets, we have extracted characteristics from our Titan database described in deliverable D5.1.2. However the same approach can be applied to any database, since IMPReSS platform can be used to deal with any database.

3.1 Dataset characteristics

Several solutions for forecast problems may be developed by using machine learning/data mining techniques provided by the IMPReSS platform. For this purpose, a historical dataset is needed in order to enable the machine learning algorithms performing forecast. The first step toward the development of intelligent applications is the definition of an objective. Then, a set of features that could represent the data distinctively must be chosen. Finally, the dataset has to be built. Once the dataset is available, it can be used as the training/test sets for several types of machine learning algorithms, such as classification, regression, cluster or association methods. Deliverable D5.3 shows the dataset format, which is required for each type of machine learning algorithm.

Typically, the features or attributes that are used represent the data samples of the overall population that are useful to support some specific decision-makings. Table 1 depicts some features from the Amazonas Theater Opera House Titan database. For instance, cluster algorithms may employ a dataset composed of data samples, such as occupation, energy consumption or even anomalies that represent a user activity.

Table 1. Description of some attributes examples that can be extracted of the titan dataset.

Feature (Attributes)	
Room	Room name
Date/time	Moments timestamp when an action occurs
Sensors (S1,S2,…,Sn)	Values of all possible sensors of the ambient (humidity, temperature, power, occupation)
Event	Assign ‘yes’ if there is event or ‘no’ when there is no event at the opera house
Peak time	Assign ‘yes’ for peak-hours, otherwise, assign ‘no’
Occupation	Assign ‘yes’ if the ambient is busy, otherwise, assign ‘no’
Amount of people	Amount of people in the room
(…)	Another attributes, if necessary
Targets (Decision attributes)	
Appliance	Value ‘0’ to turn the appliance on or ‘1’ to turn the appliance off
User Activity	User activity (Watching TV, walking, etc)
(…)	Another targets if necessary

Next sections will present some scenarios with specific datasets that were created using features depicted at Table 1. Section 4 presents a method focusing on saving energy based on user activity recognition. Section 5 presents an energy prediction method using regression algorithms. Finally, section 6 presents a method that turns the appliance on or off based on user occupation.

4. User Activity Recognition

Activities developed by users have considerable impact on the amount of energy consumed in residential environments. C. F. Lai et al. [1] claim that the energy consumption of electrical appliances is directly related to human activities. In addition, a significant amount of energy could be saved if the appliances fit according to the real users need.

In this context, this section presents a new method, called Activity-Appliance-Energy Consumption (AAEC), which can recognize users activities based on the relation between activity and appliance. The proposed method employs a decision tree algorithm to generate a model for automatically recognizing the activities carried out by the user. In addition, a ranking algorithm is used to define the importance of each appliance in relation to each activity.

4.1 Activity Recognition

The user activity recognition process is performed by processing the information collected from sensors and appliances. This process uses a decision tree algorithm to determine the category of the activities and a Euclidean distance algorithm to recognize the activity associated with each category based on the analysis of the sensors runtime.

This process consists in identifying and categorizing the activities. Category recognition is based on rules generated by a decision tree algorithm, which takes into account the object (appliance, furniture) and where the sensors are installed. Figure 1 shows an example of a generic decision tree that recognizes use activity category.

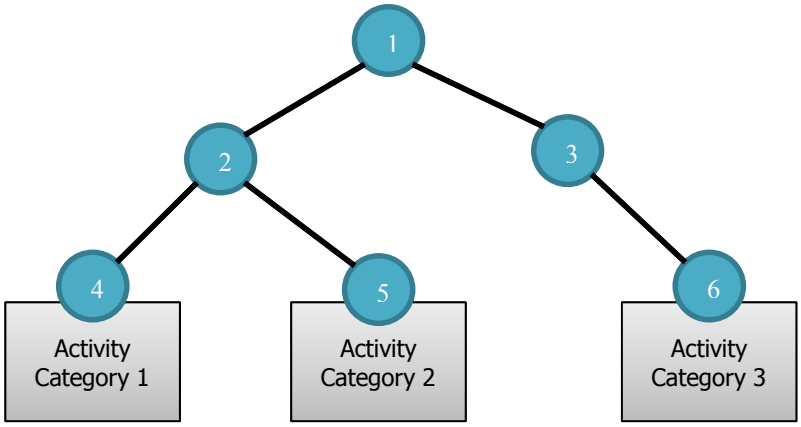

Figure 1. Example representing a generic decision tree for category activity recognition.

Tests were made based on real data obtained from an MIT project that recognizes activities in residential environments [5][6]. This database contains data from sensors and activities performed by users in two apartments for the period of two weeks. Presence Sensors were installed in different locations such as the stove, closet, washing machine, refrigerator, doors, windows, drawers, microwave, sinks, showers and sockets. The data base contains the following information: date and time of activation and deactivation intervals of the sensors, sensor location and identification of the object that has the sensor installed, the date and time of the activity and the execution interval of the activity.

The activity recognition technique is based on the analysis of the presence duration. There is a set of activities performed in a certain period of time for each category. For instance, given the category Cooking, a person may take 40 minutes to prepare a lunch and about 10 minutes to prepare a snack. Therefore, the presence duration can be compared to the average time of the activity history, as shown in Table 2.

Table 2. Example of the activities of the category 'Cooking', performed in the kitchen. The sensor runtime is closer to 'Prepare drink', since it was active for 603 seconds.

Location	Activity	Average Execution Activity (second)
Kitchen	Prepare drink	544
Kitchen	Prepare snack	470
Kitchen	Prepare coffee	898
Kitchen	Prepare dinner	1220
Kitchen	Prepare lunch	2275

A Euclidean distance measure is used in order to determine how close the detected presence is to the average time of a recorded activity. The Euclidean distance measure used [2] is defined as $|p_x - q_x|$, where p_x is the average performance of a particular activity and q_x is the time the activity remained active (how long the occupant is detected in the room for).

4.2 Relationship between activities and appliances

The relationship between activities and appliances defines how a particular appliance is important to an activity. The level of an appliance's importance in relation to an activity is defined by a vector space model of correlation, or simply the vector model. Recommendation algorithms use this relationship to identify appliances that waste energy while performing an activity.

In order to determine each appliance's level of importance related to an activity, a ranking is established based on the frequency of the vector model defined by an equation, as depicted bellow. The values obtained are normalized to adjust them in the range between 0% and 100%. The equation of normalization is given by:

$$fn(d_i) = \frac{f_{i,j}}{max(f_{i,j})}$$

Normalization $fn(d_i)$ is applied over the appliances of activity dj, where $f_{i,j}$ is the frequency of the appliance t_i and $max(f_i,f)$ is the highest frequency of all appliances from d_j. The frequency is given by:

$$f_{i,j} = \frac{M_{i,j}}{\sum_k^n M_{k,j}}$$

where $M_{i,j}$ indicates the appliance frequency t_i in activity dj and $\sum_k^n M_{k,j}$ is the total sum of the frequencies of the appliances that make up each activity.

The ranking process of appliances defines the intensity of the relationship between activities and a set of appliances. Then, for each activity A, there is a set of appliances T, with weights W. Table 3 shows an example of the ranking of appliances related to an activity called "Cleaning House".

Table 3. Example of an appliances ranking associated to an activity.

Cleaning House	Ranking (%)
Vacuum cleaner	100
Floor polisher	100
Dish washer	50
Clothes washer	50
Living room lamp	33
Radio	25
Bedroom lamp	20
Bathroom lamp	14
Area lamp	12
Kitchen lamp	9

4.3 Recommendation algorithms

Recommendation algorithms are responsible for analysing the data (correlation levels of activities and appliances), and provide suggestions to the users to control the appliances (for example, turn on or off). The appliances were split into categories in order to simplify the process. The categories are defined by groups of appliances that operate differently. These characteristics are important to distinguish the actions on each appliance to improve the system efficiency.

This categorization is able to generate more accurate solutions, since understanding the consumption patterns of each appliance category allows the developers to create more efficient algorithms for controlling the appliances. Table 4 shows the creation of four categories based on time:

Table 4. Example of electrical appliances separated by categories.

Category	**Appliances**
Continue	Refrigerator, freezer
Not timed	TV, lamp, air conditioner, radio, computer, piano
Timed	Clothes washer, microwave
Snapshots	Stove, Blender, electric lock

The category of appliances are combined with the need of a user presence (yes or no). For example, a washing machine is a timed appliance that does not require user presence. This approach creates a higher level of granularity for the classifications. Table 5 shows some possible combinations between a category and need of user presence.

Table 5. Categories of appliances for user presence.

Type	**Need User Presence**	**Category**	**Objects**
1	Yes	Not timed	Lamp, TV, Air conditioner.
2	Yes	Timed	Microwave
3	Yes	Snapshots	Blender, Mixer
4	No	Timed	Clothes washer
5	No	Not timed	Sound system
6	No	Continue	Refrigerator, freezer

Based on this categorization, the AAEC Method defines three recommendation algorithms. The first algorithm performs recommendations based on the user activities. The second algorithm makes the recommendations based on peak times. Finally, the third algorithm detects energy waste of continuous appliances if there are some changes outside the appliances factory specifications.

The algorithm starts by analysing the rank (defined by the proposed vector model) of each appliance involved with a recognized activity. If the appliances rank is less than 50%, the algorithm asks the user to see if the appliance can be turned off (lines 4-10). Preliminary experiments show that most appliances whose correlation is below 50% is weak related to the activity (see example in Table 3). Then the algorithm turns off all the appliances, which are not related to the activity and that needs the presence of a user (lines 11-15).

```
1.  Get the type 1 and 2 appliances related to recognized activity.
2.  for appliance of the activity X do
3.      rank = get rank of the appliance Y
4.      if rank > 0% and rank < 50% then
5.          if appliance is type 1 then
6.              if user to authorize turn the appliance off then
7.                  change status of the appliance
8.          if appliance is type 2 then
9.              if user to authorize turn the appliance off then
10.                 expects to complete the scheduled time to change the status of the appliance
11.   else if rank = 0% then
12.         if appliance id type 1 then
13.             turn the appliance off
14.         else if appliance is type 2 then
15.             expects to complete the scheduled time to turn the appliance off
```

Algorithm 1. Algorithm recommendation for objects of type 1 and 2.

Algorithm 2 treats appliances that do not require user presence (types 4 and 5). If the appliance is running at peak times, the algorithm makes a recommendation advising the user to utilize the appliances outside of peak times (lines 3 and 4), otherwise and if the appliance rank is less than 50%, the algorithm asks the user if the appliance can be turned off due to the weak relation with the activity (lines 5 and 6).

```
1.  Get the type 4 e 5 appliances related to recognized activity.
2.  for appliances of the type 4 and 5 do
3.      if at pick hours and appliance of turn on then
4.          recommend use out of peak hours
5.      if not at pick hours and rank <= 50 then
6.          recommend user turn the appliance off
```

Algorithm 2. Algorithm recommendation for objects types 4 and 5.

Algorithm 3 deals with of type 1, 2, 3, 4, 5 or 6. The algorithm monitors the energy consumption of appliances and compares it to the consumption of their factory specifications focusing on discovering whether or not the objects are consuming more electricity than it should. If so, the algorithm informs the user that there might be some malfunction (lines 2 and 3).

```
1. if appliances is type 1 or 2 or 3 or 4 or 5 or 6  then
2.      if consumption appliance > consumption of factory specifications then
3.          Report waste of energy to the user
```

Algorithm 3. Algorithm recommendation for objects types 3 and 6.

5. Energy Prediction

Energy prediction can help users to use appliances of a given ambient efficiently. Thus, users can organize consumption routines in order to save energy. Machine Learning Regression algorithms can be used to achieve this goal. For this, it is necessary to organize the data creating the prediction model. Next section shows how the data must be organized.

5.1 Prediction with regression algorithms

Data must be organized with a timestamp that shows when the energy consumption was collected and the energy consumption measured in *watts* or kWh. Table 6 shows an example.

Table 6. Example of dataset organization to be used for forecast energy consumption.

Date/time	Consumption (kWh)
23 Mar 2013 15:51:12 GMT	52
23 Mar 2013 15:51:12 GMT	34
(...)	(...)

Once the database is prepared, the next step involves generation and analyzes of the prediction models. The idea is to train the model using different groups of dates, such as hours, days, months or years. In this way, many models may be generated to predict consumption at different levels of granularity. For instance, when gathering the dates by day, it is possible to know the daily average energy consumption from the environment. This allows the prediction of the daily average consumption of upcoming days. The same goes for months and years.

In addition, we can use this idea of grouping dates to predict consumption per device or set of devices separately. Figure 2 shows a laptop data historical of 7 days. In this case, for example, the consumption of the next day can be foreseen.

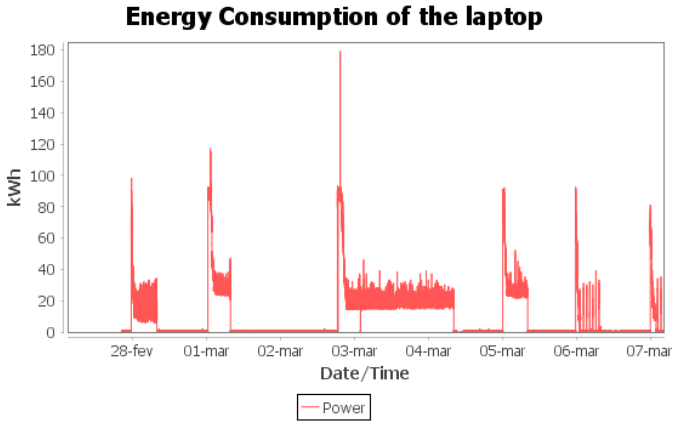

Figure 2. Example of a laptop historical consumption data of seven days.

An energy prediction example will be demonstrated in Section 7.2.

6. System HVAC and Appliance Automatic Control

Lifestyle and habits of users have a direct effect on the energy performance of dwellings and facilities. Hence, in the real environment, advanced control strategies must be adapted to user behaviors to try to keep a tradeoff between energy consumption and the user comfort [4]. In this context, this section shows a suitability predictive model of control strategy, which is based on occupancy profiles for optimizing the energy consumptions of HVAC systems.

6.1 Occupation model

People are prone to follow occupancy trends with a certain daily or weekly pattern. Usually, occupants maintain the same habit profile, however, some variations over the original model can be considered. For example, the time that someone leaves or comes back to an environment can vary randomly within two hours before or after the originally expected occupancy. At the opera house scenario, for instance, this pattern can be identified in some events. Figure 3 shows an example of an occupation time line of five days.

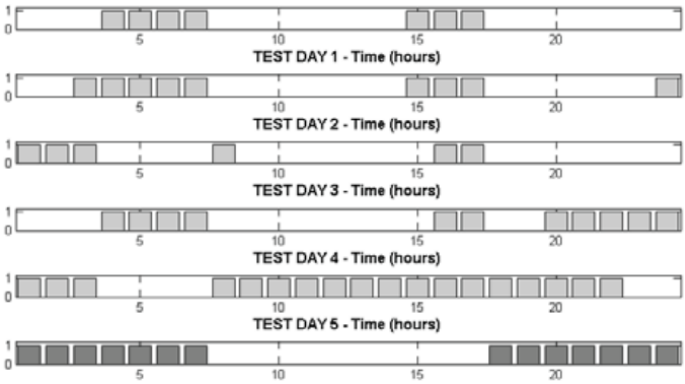

Figure 3. Day occupation example [4].

The time line represents the historical data of n-days obtained by means data collection of a certain environment. The dark square means the time period that user is in the environment and white space means the time period that the user is not in the environment. Several control strategies can be conducted in this scenario.

- **On/off control:** this controller follows a strategy of switching on devices when people arrive home and switching them off when they leave. The controller may be triggered manually or automatically through the occupancy detection.
- **Schedule control:** This type of controller establishes comfort temperatures during the expected or normal building usage time.
- **Combined control:** This controller sets the temperature to setback/setforward levels within the schedule, and goes up to comfort levels when occupancy is detected.
- **Controller based on profiles:** This controller predicts the future occupancy.

The occupancy profiles can be implemented based on cluster algorithms that triggers the on/off control. Moreover, cluster algorithms can also be used to capture pattern trends to control the environment. For example, cluster can provide additional information allowing the controller to get context awareness about user habits (e. g. to determine how reliable the obtained patterns are or how close they are to each other). The clustering information available for the context awareness appraisal includes: (a) number of patterns, (b) distance between patterns, (c) size of clusters (membership percentages), and (d) distance average and standard deviation inside clusters (regarding the cluster representative).

In the opera house scenario, we combine some attributes from the database, such as room, timestamp, occupation and event to create a prediction model, which will decide whether or not an HVAC system or appliance should be turned on or off.

7. Opera House Application

This section presents the scenarios implemented in the opera house application. First, we created a screen that automatically generates datasets in a JSON format to be used by the IMPReSS data analytic module. Then, a screen to provide energy prediction information, and a screen to automatically control the HVAC system based on user occupation.

7.1 Generate dataset

The screen shown in Figure 4 is responsible for generating datasets in a JSON format, which can be recognized by the IMPReSS analytic module. More information about the JSON format has been discussed in Deliverable 5.3. The datasets are generated by the area within a date range that can be grouped by minutes, hours, days or months. Date groups are useful to create and evaluate prediction model with different granularities in relation to time.

In this screen, it is possible to create predefined datasets or custom datasets. Predefined datasets have attributes defined during the development of the application, while the custom datasets have attributes, which can be selected by users.

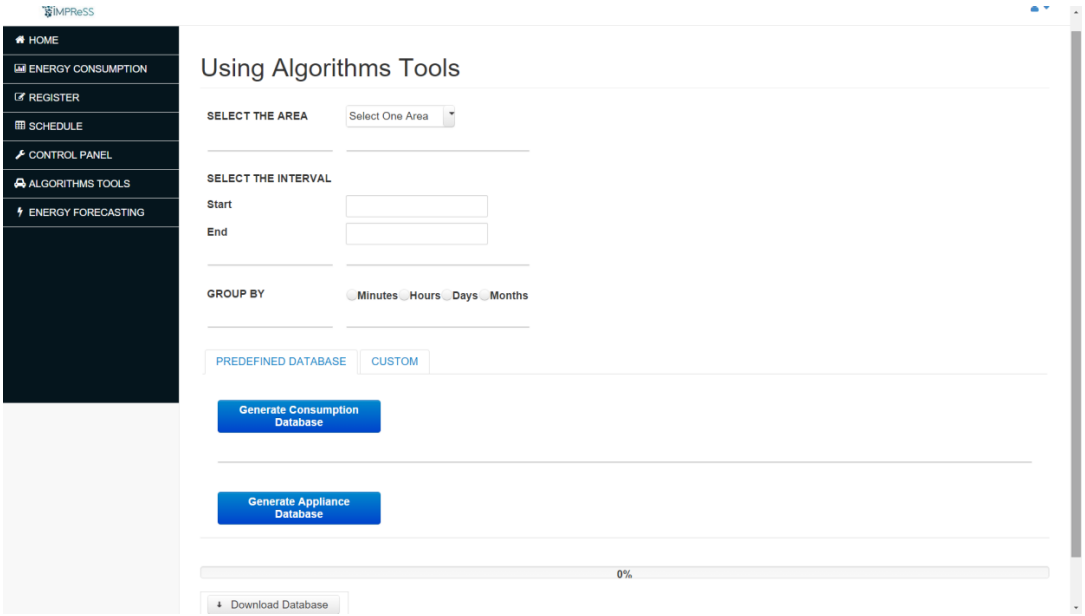

Figure 4. Dataset generator screen.

7.2 Energy prediction

The screen, depicted in Figure 5, is responsible for making predictions of energy consumption. This can be accomplished by employing any regression algorithm available at the IMPReSS data analysis module. Basically, the user chooses which dataset will be used and which date/time in the future will be used to calculate the prediction. The result is a chart that contains the history and the future values of energy consumption.

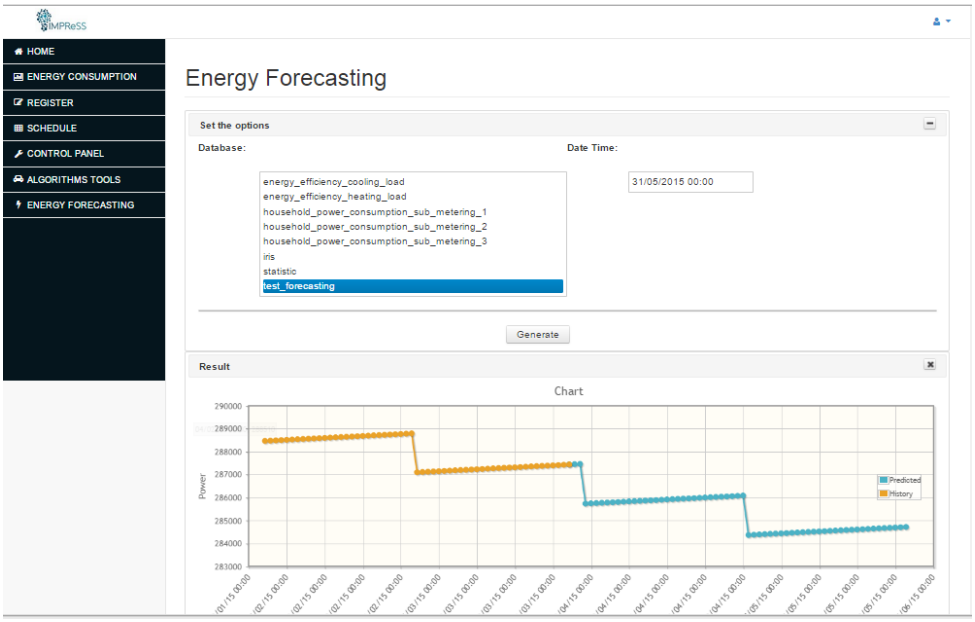

Figure 5. Energy prediction screen.

7.3 Recommendation based on occupation

The screen shown in Figure 6 is responsible for making recommendations to the users. The idea is to show the status of the HVAC system to the users in real time. Besides, the occupation model can be manually tested. To accomplish this objective, the users only have to create a dataset with room, timestamp, occupation and event attributes and then, go to the recommendation screen and choose the dataset and any classification algorithm. The result will indicate whether or not the HVAC system must be turned on.

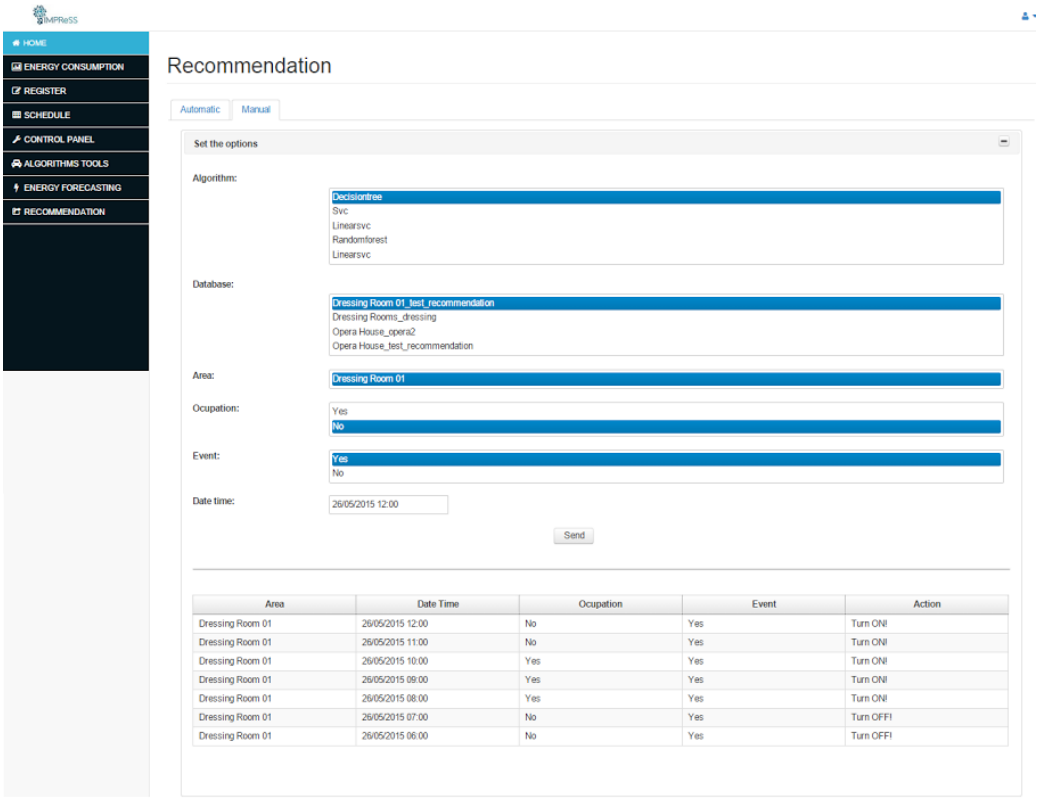

Figure 6. Recommendation HVAC system screen.

8. Enhancing Presence Detection Models

This section presents a study on augmenting the performance of predicting occupancy with PIR's presence data. This is important since data mining techniques can be employed as an intermediary step, between the collection of PIR's presence data and its usage, so that false positives and false negatives are minimized. For example, when a person is reading or when someone is outside the detection range of the PIR. These are two usual cases that automation systems based on PIRs are unable to detect the user presence and fail to provide a good user experience.
A usual workaround is to use other data sources to support decisions regarding presence in a given area. Those can be obtained from user's calendars, RFID tags, cameras and several other indirect means. However, every effort in that sense adds complexity not only to the developers of such a system, but, more importantly, to users that expect such an automation system to simply work or, at least, to adapt itself with time.
Moreover, even if other data sources are employed for automation, improving PIR's presence detection will still be beneficial, since the overall presence detection performance will increase.

8.1 Collecting Data

For improving presence detection, presence data had to be collected in the first place. For such, data from a single PIR sensor, positioned at the upper corner of a project room at the Federal University of Pernambuco (UFPE), was collected. That data entails changes of presence state in the period between 2015-05-04 and 2015-05-28.

For generating the data that was used as the ground truth (i.e. target), a program was installed in each of the five computers in the aforementioned room. The program was used for reporting if any mouse or keyboard interaction was made every thirty seconds. Later, these two datasets were correlated using the closest timestamp.

8.2 Data Pre-processing

A crucial part of data mining is feature engineering. That is, altering existent variables (i.e. features), eliminating them or creating new derived ones, in order to facilitate the learning process of a model. In that sense, five variables were created, out of a time series of presence data (i.e. 1 for has presence and 0 otherwise).

They were Hour, TimestampDiff, PeriodDay, DayOfWeek and PresenceRMS. The first four were generated so that the model could learn its users' patterns of occupancy in a temporal fashion. The last one is the RMS of the presence data. A different dataset was created for the different timeframes (i.e. 1 min, 2 min, 5 min, 10 min) used in the PresenceRMS variable. For more on then those variables, see Table 7.

Table 7. Input and Target Variables.

Name	Type	Description
Hour	C	The hour of the presence detection
TimestampDiff	N	The difference in seconds from the last presence detection
PeriodDay	C	The period of the day of the presence detection (i.e. dawn, morning, afternoon, night)
DayOfWeek	C	The day of the week of the presence detection
PresenceRMS	N	The RMS of the presence data in a given time window
HasPresence	C	The target of the prediction (i.e. 1 for has presence or 0 otherwise)

[1] * The type of the variable can be categoric (C) or numeric (N).

Last but not least, Hour, PeriodDay and DayOfWeek were replaced by equivalent dummy variables.

Model Development

For each dataset, samples were split into two parts, being 60% for training the model and 40% for testing its performance. The data destined for training was then used to train several decision trees, using the C4.5 algorithm, one for each of the aforementioned timeframes. Note that proper weights were added for the different samples (i.e. has presence or not), since the data collected, as is usual in binary decision problems, suffered from class imbalance (approximately 1:3).

8.3 Evaluation

For evaluating the models, two strategies were used. They were, ROC curves and confusion matrices. The higher the timeframes used to calculate the RMS, the better the ROC curve for the model. Or in other words, the closer the curve was to the upper left corner of the chart. However, increasing the timeframe also implies in longer response times. Therefore, there is clearly a trade-off between convenience and minimising consumption. Avoiding false negatives will please users, as lights will rarely turn off in their presence, but that also mitigate the amount of saved energy, since decisions to turn off a light, for instance, will be taken in bigger intervals. Therefore, to favour an intermediary approach, the timeframe of five minutes was chosen.

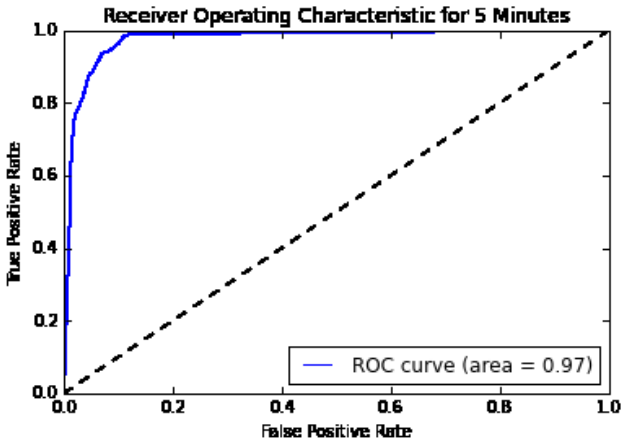

Figure 7. The ROC curve for a five minutes timeframe.

As can be seen in its ROC curve, depicted in Figure 7, the area under the ROC curve was very high, indicating that the predictor was good. However, since false positives are very unpleasant for users, the cut-off threshold of the predictor was shifted to 0.46 (scores range from 0 to 1). Being anything higher than that threshold predicted as a presence. Due to that, the number of false negatives diminished to 25, as can be seen in Table 8.

Table 8. Confusion matrix for a five minutes timeframe.

	Predicted No Presence	**Predicted Has Presence**
No Presence	1824	210
Has Presence	25	714

In addition, the success rate of predicting no presence was affected, but as already mentioned, the cut-off threshold was shifted in order to be reasonably biased towards a high true positive rate.

9. References

[1] C. F. Lai, Y.-X. Lai, L. T. Yang, and H.-C. Chao, (2012) "Integration of IoT Energy Management System with Appliance and Activity Recognition," 2012 IEEE Int. Conf. Green Comput. Commun., pp. 66–71.

[2] P. E. Danielsson, "Euclidean distance mapping," Comput. Gruphic,.s Image Processing, vol. 14, pp. 227-248, 1980.

[3] Frakes, William B., and Ricardo Baeza-Yates. (1992) "Information retrieval: data structures and algorithms".

[4] Felix I. V., Wolfgang K., Christian R. (2011). "Impact of User Habits in Smart Home Control".

[5] E. M. Tapia, S. S. Intille, and K. Larson, (2004) "Activity recognition in the home setting using simple and ubiquitous sensors", in Proceedings of PERVASIVE 2004, vol. LNCS 3001, A. Ferscha and F. Mattern, Eds. Berlin Heidelberg: Springer-Verlag, pp. 158-175.

[6] E. M. Tapia, N. Marmasse, S. S. Intille, and K. Larson, (2004) "MITes: Wireless portable sensors for studying behavior", in Proceedings of Extended Abstracts Ubicomp 2004: Ubiquitous Computing.