D4.4 FINAL USER BEHAVIOR MODEL AND RECOMMENDER

Project title	**Collaborative Recommendations and Adaptive Control for Personalised Energy Saving**
Project acronym	**enCOMPASS**
Project call	**EE-07-2016-2017 Behavioral change toward energy efficiency through ICT**
Work Package	**WP4**
Lead Partner	**GRA**
Contributing Partner(s)	**EIPCM**
Security classification	**PU (Public)**
Contractual delivery date	**31/08/2019**
Actual delivery date	**30/08/2019**
Version	**1.0**
Reviewers	**PMI, SUPSI**

History of changes

Version	Date	Comments	Main Authors
0.1	12/06/2019	Structure of the document	Krisztina Tarjányi, Eszter Somos, Balázs Hidasi
0.2	12/08/2019	Introduction, theory, description of the recommendation engine and demo	Eszter Somos, Krisztina Tarjányi, Balázs Hidasi, Gábor Vadász
0.3	12/08/2019	Behavioral change state clustering	Marta Chelminska
0.4	13/08/2019	Review of the document in relation to Notification Engine	Luigi Caldararu
0.5	16/05/2019	Quality check	Piero Fraternali
0.6	21/08/2019	Review of the document (structure, other notes)	Marta Chelminska, Jasminko Novak
0.7	26/08/2019	Restructuring, some additional amendments	Eszter Somos, Krisztina Tarjányi
0.8	28/08/2019	Review of the restructured document	Marta Chelminska
0.9	29/08/2019	Review of the document	Andrea-Emilio Rizzioli
0.10	29/08/2019	Review of the document	Jasminko Novak
1.0	30/08/2019	Finalization of the document	Eszter Somos, Krisztina Tarjányi, Gabor Vadasz

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

EXECUTIVE SUMMARY

The present deliverable is defined in the Grant Agreement document as:

"Validated prototype, with documentation, of the models and algorithms for determining the current status of the users and the most suitable actions to recommend based on the adaptive context detection, validated in the pilot experiments."

This document describes both the final recommendation engine and its demonstrator tool. Following the overview of the RE, a theoretical introduction, the logic of the recommendation algorithm, the user model and the functionalities of the demonstrator are described.

This document accompanies the D4.4 deliverable, which is a demonstrator.

Section 6 of this document explains how to use the online version of the demonstrator.

The demonstrator can be reached at http://212.32.240.74:5000/

with basic http authentication.

username: enCOMPASS

password: sWX8JW9Des

1 INTRODUCTION

1.1 THE RE AND THE DEMONSTRATOR

This chapter describes the final recommendation logic, the main principles for creation of the final user model and the demonstrator tool, which is able to present the performance of the recommendation engine (RE) in an easily understandable way. The logic of the RE and demonstrator is based on the first documentation (D4.2. First user model and recommender). The final version of the recommender was produced by fine tuning the user model and the background algorithms with results of further research, data and experiences from the pilots.

As promised in the Grant Agreement, the demonstrator app uses the final prototype of the RE. Partial developments of the RE were deployed on the pilots during the project.

It is important to understand that there are differences between the RE and the demonstrator tool shown in this document. The demo was built to show the logic of the RE in a clear way. The main differences between the two are the following:

1. The RE works separately on the three pilots based on the same principles. It uses data from the different pilots, groups users and reclassifies them, it compares them to each other and by mathematical methods (described further) calculates which is the most appropriate recommendation for them. The app does not show the way of calculation and its stages are not visible, but just the result of the process - the recommendation for the given user.

2. The demo is based on real data from the Swiss pilot. It uses data from real users (such as questionnaire data) and a slightly modified data set for consumption and sensor data. To augment the periods of complete data (i.e. when all the type of data was available) a data set was created for the Swiss pilot, which is used in the background of the demo. Two time periods were chosen for demonstration purposes: one during the heating period (2019/01/01 - 2019/01/22) and one during the cooling period (2019/06/17 - 2019/07/07). Periods with high percentage of complete data was chosen. In cases where some data was not available, data from the same user but another time period was used. This method is only applied in the demonstrator tool, not in the RE. Only users with questionnaire data are included in the demonstrator.

3. The demonstrator offers the possibility to select different inputs (different users and dates) and shows the recommendations offered to them and the time of the notification. The demonstrator also shows the status of the user model, the type of the cluster to which the user belongs on the selected date and some features on which the message selection was based on.

1.2 OVERVIEW OF THE RE WITHIN ENCOMPASS PLATFORM

In this section we introduce the aims of the RE in the enCOMPASS platform in order to provide an understanding of the framework in which RE and the demonstrator was built.

In line with the Grant Agreement, the RE was created within the framework of WP4 with the objective to help consumers improve their energy-related behavior towards more energy-efficient patterns. The RE is a very important part of the system as recommendations support saving actions.

The approach of the recommender engine includes inferring user behavior and context using the sensor and consumption data, user profiles and potentially other information available. We also added a clustering feature to the recommender, which helps to find similar users and user patterns, and a collaborative recommendation feature to include users' reactions to recommendations received in the past. These features are referring to the tasks described in WP4: Task 4.2 Context-aware user and building modelling (M6 – M24) and Task 4.3. Collaborative recommender for energy saving (M9 – M34).

After analyzing the users, the recommender selects recommendations from a static database based on predefined rules: the RE, where the recommendation algorithm is matched with the context-awareness system, detects certain user behavior patterns (e.g. inefficient energy use) and recommends corresponding recommendation(s) when this behavior is observed multiple times. The detection of energy wasting behavior is indirect (i.e. inferred). Recommendations are accompanied by adaptive incentive messages based on the user type as described in *D5.3 First visualization and feedback interfaces and behavioral game* and in *D5.4. Final visualization and feedback interfaces.*

Recommendations become visible in the recommendation queue of the user and push notifications are sent at the appropriate time. In the interface of the app, personalised recommendations are listed on the "Just for you" tab,while general messages for the users are available for the users on the "Tips" tab.

Figure 1.-4. show examples of the "Tips" tab view (*Figure 1-2*) and of the "Just for you" tab view (*Figure 3-4*) in the application:

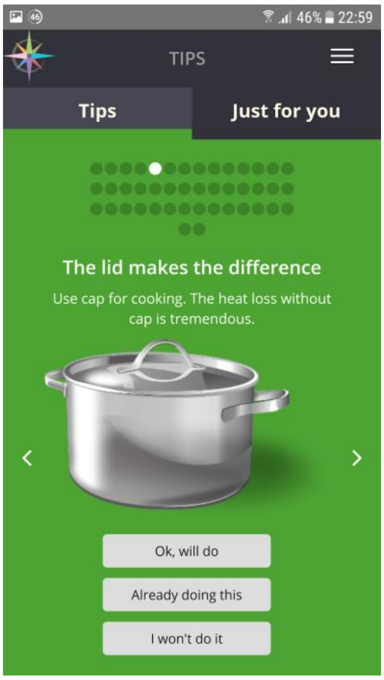

Figure 1: Example for "Tips"

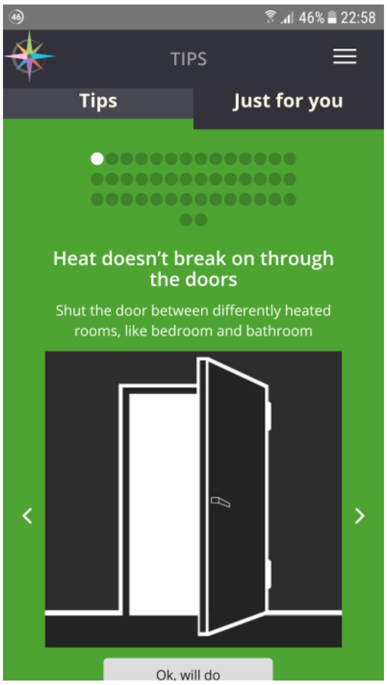

Figure 2: Example for "Tips"

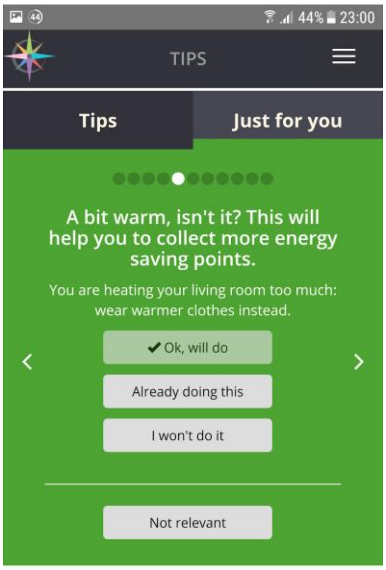

Figure 3: Example for "Just for you"

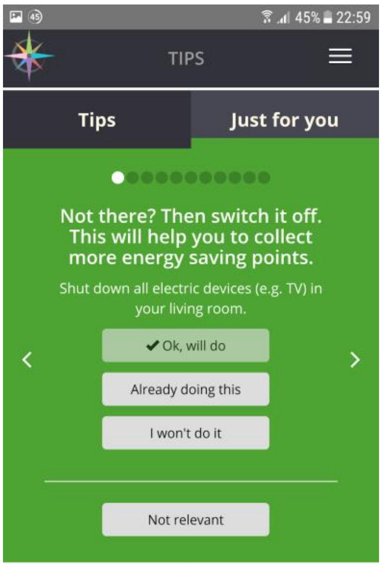

Figure 4: Example for "Just for you"

1.3 INPUT DATA FOR THE RECOMMENDATION ENGINE

In the following two sections we introduce the list of the input data used for our purposes and we describe methods for analysing them.

List of input data

The RE ingests the following data available in the enCOMPASS Platform:

- consumption and sensor data from the pilots
  - 96 measurements per day per user (both the consumption meter and the sensors record data every 15 minutes)
  - In the three pilots there are different sensors, therefore the RE considers:
    - in Germany: luminance, motion, temperature, humidity
    - in Switzerland: luminance, motion, temperature, humidity
    - in Greece: temperature, humidity
- user profile data
  - Originally collected data, used to infer different features about the users and the buildings in which they live:
    - information about residents: number of adults, kids and pets
    - features of the house: number of rooms, type of dwelling (e.g. single family house, apartment) and heating source (electricity, oil, gas, wood)
    - information about devices in the household: presence or absence of several device types
    - responses to the first and intermediate questionnaires
- information from the different components of the enCOMPASS platform
  - Disaggregation results (from the Disaggregation Engine elaborated within th task of *T3.4. Disaggregation of energy use*)

- Thermal and visual comfort results (from Comfort Inference Engine - developed in *T3.3. Indoor climate detection*)
- Activity Inference (from Activity Inference Engine developed in *T4.1. Privacy-preserving user activity type profiling and matching*)[1]

- user actions on the Platform including user responses to RE suggestions, the so called feedbacks. In the enCOMPASS application users are given the possibility to provide feedback about the energy saving tips. There are four versions for response: I will do it/I am already doing this/I won't do it/Not relevant. Users were able to provide multiple feedbacks for one message. Some users gave an amount of feedbacks that questions their validity. To ensure that only intentional and well thought feedback is used, data from users giving more than 1 feedback for more than half of the recommendations were excluded from the input data. Also feedback for a message from a user giving more than 4 feedback for that message were also excluded.

Information flow of the RE can be seen on *Figure 5*, where the list of available data and also the external processes which help us to process information can be seen.

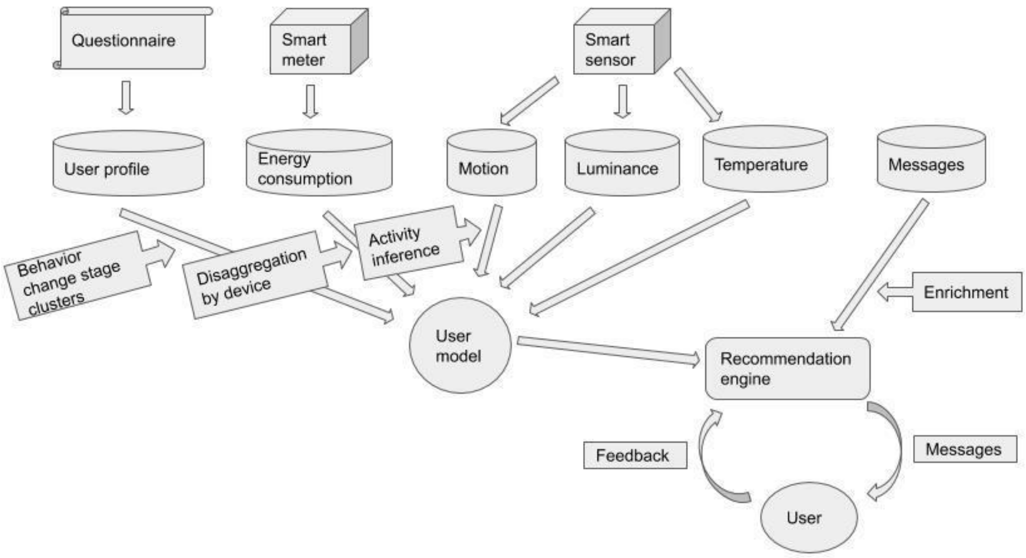

Figure 5: Information flow of the RE

1.4 OVERALL STRUCTURE OF THE RE

The RE supports the enCOMPASS platform in its endeavor to change energy consumption habits of the users. While the platform is a viable tool without the RE as well, using this component amplifies the platform's effect by personalizing the communication with the users.

There are two main components inside the RE: a user model and selection of the best message for the given user.

[1] After the review of the results of occupancy inference and activity inference we have chosen to use activity inference, as it has bigger information content.

First, a user model is built for each user. The creation of this user model is:

- recurrent: every time the RE runs, the user model is updated. This way the personalization follows the change in the user behavior and the change in the context (e.g. time of year);
- using many inputs: inputs from multiple components of the enCOMPASS platform are included in the user model: user questionnaires, smart meter measurements and outputs of other components as it was described above;
- using multiple algorithms: different statistical models (like clustering and time series analysis) are used.

As the second step, the RE selects messages targeted at each user and it also calculates a personalized timing for the notification. The message selection is based on the user model and uses a collaborative method. As the users interact with the enCOMPASS application by giving feedback on the recommendations, the message selection process can be based on more data and become more accurate (e.g. if users give feedback based on which the given recommendation is not relevant for them, the RE engine will provide this recommendation to the users within the same cluster with a lower weight, thus with lower probability).

In terms of structuring and technical elaboration we shall emphasize the following: the RE backend is deployed as a SaaS like service, and thus it is not an integral part of the enCOMPASS platform. Its installation and monitoring are provided by Gravity on their own server.
The recommendation backend in high level is an API-less service. At this abstraction level this is the is only communication between enCOMPASS platform and RE. RE takes input in the form of an uploaded directory of database exports from the enCOMPASS platform. RE calculates recommendations for the next day and then sends the next day's recommendations to the enCOMPASS platform in the form of an AP.
Internally the recommendation backend limits the number of runs of these tasks to once a day in order to prevent recalculation and resending the data (data import is done in every case).
The data ingestion and process of the RE is started by checking if data upload happened from the enCOMPASS Platform. By checking it every minute, it prevents large delays in the orchestration between this and the next component.
For better handling of debugging sessions and issue solving in enCOMPASS platform orchestration, if the recommendations are already calculated and sent to the platform on a given day, calling the RE on that day again from orchestration will not stop the orchestration chain, but only log a warning, and call the next element in the chain.

As a next step in the orchestration the Notification Engine (NE) sends out notifications from the notification queue. As the NE handles also other notifications (i.e. gamification, system notifications, recommendations), the final number and time when the recommendation is sent out is defined by the rules of the NE (the users receive notifications 2 or 5 times per week depending on their previously set preference). Therefore, the proposed and the actual timing of a notification might differ.

2 ALGORITHMS AND STRATEGIES USED IN THE RECOMMENDATION ENGINE

While creating the RE, we have reviewed and used several algorithms. First, we had to examine the collected data and create groups/clusters for them. Subsequently, we needed to identify mechanisms allowing us to select the best recommendation for a user or for a user group.

In the following we briefly describe machine learning algorithms used for the generation of recommendations: clustering, time series analysis and collaborative recommendations.

2.1 MACHINE LEARNING FOR RECOMMENDATION GENERATION - INTRODUCTION

Machine learning is a scientific field using algorithms and statistical models for solving specific problems without explicit instructions. Machine learning models are trained to recognize patterns in the data and modify the output accordingly. For example, for an image recognition algorithm not using machine learning, explicit rules of what should be categorized as a specific object (e.g. a cat) need to be set, while for a model using machine learning, these rules are not defined in advance. The model uses the input data for modifying its inner state.

There are several relatively well separated categories of machine learning tasks. If the input and the output of the problem are both given to the algorithm, we speak about supervised learning. In this case, the algorithms use the outputs (also known as labels) to optimize their inner models. Once these inner models have been trained, the algorithms can be used to produce output using only input data. For example, in image recognition using supervised learning, the initial dataset needs to contain labels for the images (for example, indicating if the image contains a cat or not).

In contrast, if the output is not available for a dataset, unsupervised learning algorithms should be applied. These algorithms aim to find patterns and structure in a dataset. An example of unsupervised learning algorithms is clustering. The goal of clustering algorithms is to split data points into well separated groups. This method is often used in business settings for recognizing different customer groups in the user base.

In the following, the theoretical background of the three main algorithm classes used in the RE is detailed. These algorithms are all part of the machine learning domain, but in the realization of these algorithms inside the RE component, only time-series analysis is used in full depth. For clustering and collaborative filtering, a non-machine learning version is implemented due to the requirements of this specific task and the size of the dataset.

2.1.1 Clustering

The goal of any clustering method is to divide a set of objects into groups. In the end, the objects in the same group should be more similar to each other than to those in other groups. Defining similarity is not unequivocal as the objects can have (and in most cases do have) multiple features, with some of them not correlating with each other.

To illustrate, how different features can impact clustering, let's look at *Figure 6* that shows the importance of religion and the GDP per capita for different countries. It is relatively clear that there is a group of countries where the importance of religion is quite high while the GDP is low (like in Ethiopia and Uganda) and another group where religion is not that important, but the GDP is very high (like in Germany or Canada). But there are also many countries in between (like Turkey) or with very different characteristics (like United States). If the religion of the country is also considered to cluster countries, the grouping becomes even less obvious as countries at the very far ends of the chart can have the same religion (like Germany and Ethiopia) while countries next to each other can have different religions (like Vietnam and Ukraine).

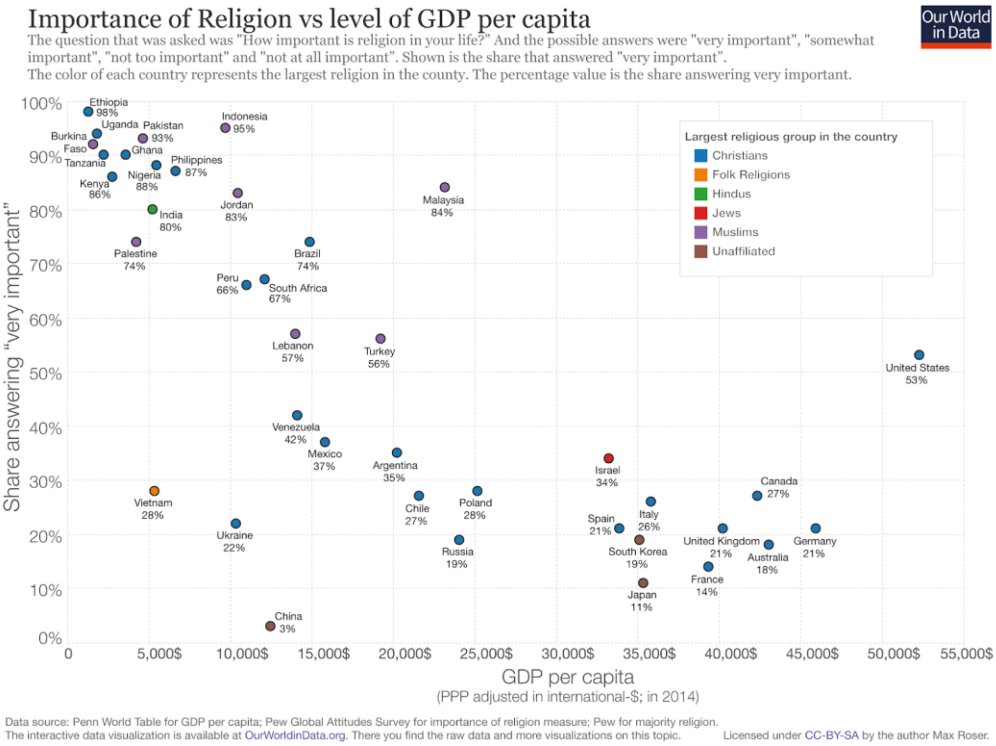

Figure 6: Importance of Religion vs level of GDP/capita (example for clustering)

In the RE, the users of the enCOMPASS application are the objects of clustering. The goal is to create groups of users with similar features. By finding an optimal grouping of users based on their features, recommendations can be personalized using collaborative methods (see 2.1.) based on the behavior of other members of the same cluster. In addition, the phrasing of recommendations can be personalized to address users in different behavioural change stages.

There are multiple algorithms used for solving clustering problems. One of the most widely used is the K-means algorithm. For this algorithm, the number of clusters needs to be defined in advance. The algorithm selects in iterative steps centres for each cluster and then assigns each data point to the centre closest to it. More formally, if c_i is the collection of centroids in set C, then each data point x is assigned to a cluster based on

$$\underset{c_i \in C}{\arg\min}\ dist(c_i, x)^2$$

After that, the coordinates of each centre are updated to the average of the data points within the corresponding cluster:

$$c_i = \frac{1}{|S_i|}\sum_{x_i \in S_i} x_i$$

The iterative steps of centre coordinate update - data point assignment is stopped when there is only very little change in the cluster membership of the data points.

An optimized version of the K-means algorithm is called K-means++. While the original algorithm selects the initial cluster centres randomly, the K-means++ algorithm takes into account the density of the data points in the multidimensional space. Essentially, it selects the initial centres relatively far from each other. This helps to find an optimal solution faster and with higher probability compared to the K-means version.

Another method of solving a clustering problem is called hierarchical clustering. When applying hierarchical clustering, initially each observation is treated as a separate cluster. In each step the two clusters closest to each other are merged. For finding the closest clusters multiple methods can be used. For example, the distance of the closest or the farthest members of two clusters can be considered or calculated cluster centres can be used for distance calculation. At the end of the cluster merging steps all observations fall into one cluster. The path of merging steps is displayed in a special format called the dendrogram. Looking at the dendrogram, one can decide which step should be considered as the final one. The merges made after this step won't be taken into account, creating the number of clusters defined at this final step (*Figure 7*).

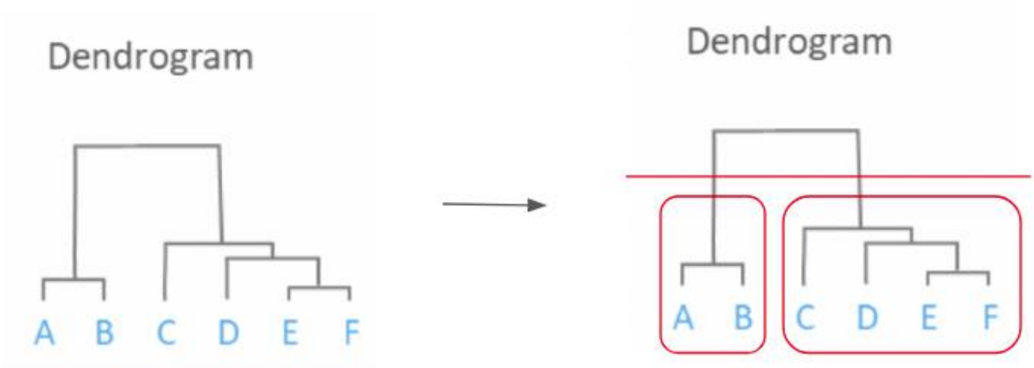

Figure 7: An example of a dendrogram chart and the final step creating two clusters

The recommendation group clustering (which is based on user profile and consumption data) inside the RE is based on the K-means clustering theory while the behaviour change state clustering (where we examine the behavior of the users, described more in 3.1.3 and 3.2.2.). is based. on the hierarchical method. For the recommendation group clustering the centres for the clusters were defined based on domain knowledge and the result of prior data analysis.

2.1.2 Time series analysis

Time series analysis is a frequently used subdomain of machine learning. All methods of time series analysis make use of the specificity of this data type: data points were collected at time intervals making the data ordered and equally spaced.

One question about time series is how to measure their similarity. In cases when the frequency and the amplitude of two time series is the same, calculating similarity is straightforward. But in most real life data (e.g. stock data, speech recognition data) that is not the case. The main problem is that series may vary in speed and can have accelerations within them (*Figure 8*)[2].

[2] All figures are from Elena Tsiporkova: Dynamic Time Warping Algorithm [PowerPoint slides]. Retrieved from http://www.mathcs.emory.edu/~lxiong/cs730_s13/share/slides/searching_sigkdd2012_DTW.pdf?source=post_page

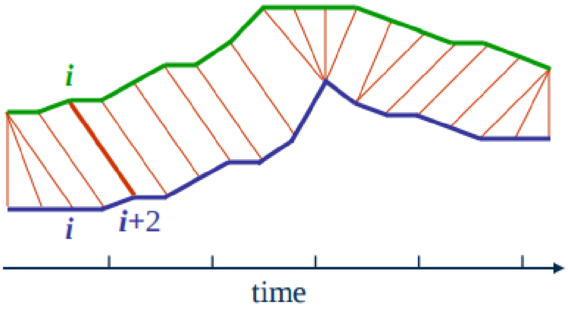

Figure 8: Examples of time series similarity

In the RE time series analysis is used to determine similarity between energy consumption of different households. In case of energy consumption time series, it is also important to explore similar patterns even if the starting point or the pace of energy use is not the same between two households. For example, two households with bursts of high energy consumption in the morning and in the evening should have high similarity, even if people in one household wake up earlier in the morning, get home earlier in the evening and use electric devices in different order compared to the other household.

One of the algorithms capable of detecting similarity between time series of real-life data is called dynamic time warping (Keogh & Ratanamahatana, 2005). It works as follows: starting with one of the series it takes all data points one by one. For each point it calculates the Euclidean distance from all points in the other series and chooses the one with the smallest value. The same procedure is then performed for the second series. The sum of these minimal distances determines the similarity between the two time series.

Formally, if the points of two series from time t=1 to time t=n are ($i_1...i_n$) and ($j_1...j_n$), then first a matrix is calculated filled with the Euclidean distances of each point of the two series. From this matrix an optimal path is found starting at t=1 and ending at t=n by dynamic algorithm: $f(k, l) = d(i_k, j_l) + \min(f(k-1, l), f(k, l-1), f(k-1, l-1))$ where $d(i_k, j_l)$ is the distance between points (i_k ,j,) (*Figure 9*).

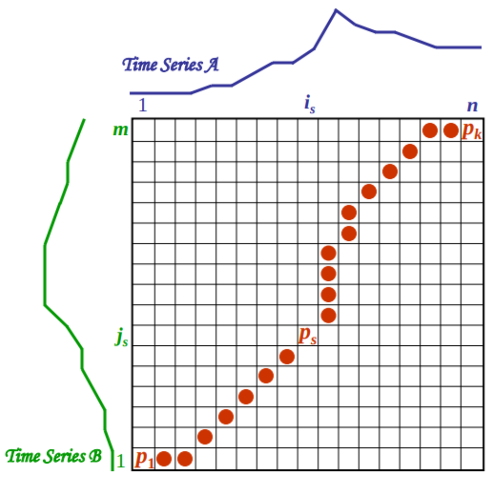

Figure 9: Example of distance matrix with minimal distance

In an optimized version of the dynamic time warping algorithm, instead of calculating all possible distances, some restrictions are applied when selecting the point with the smallest distance. They are useful for two reasons: the algorithm takes less time to find the similarity score and similarity can be measured more precisely. The restrictions applied are the following:

- monotonicity (Figure 10): the time indices cannot go back in time, thus avoiding matching one feature of one of the time series multiple times for the other series. Formally is-1 <= is and js-1 <= js.

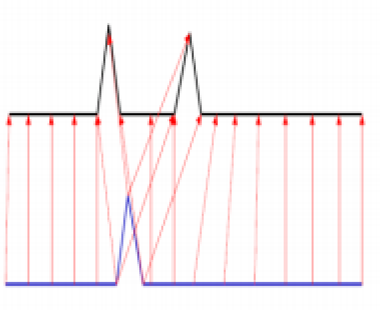

Figure 10

- continuity (*Figure 11*): there can't be time indices not matched for either series, thus avoiding skipping features. Formally i_s - i_{s-1} <= 1 and j_s - j_{s-1}<= 1.

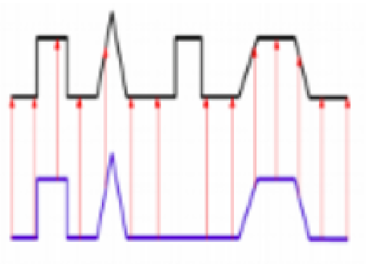

Figure 11

- boundary conditions (*Figure 12*): the matching process has to start at the first point of the series and finish at the last, thus the beginning or the end of the series can't be cut off. Formally i_1 = 1 and i_k = n and j_1 = 1 and j_k = n.

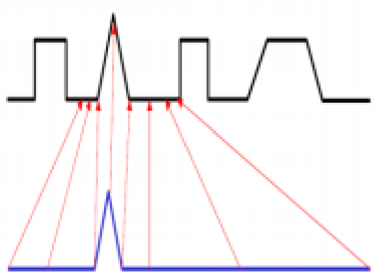

Figure 12

- warping window (*Figure 13*): a point selected from the other series as a match can't be too far away from the original point, thus avoiding counting only few similar features. Formally $|i_s - j_s|$ <= r where r < w, w being the window parameter.

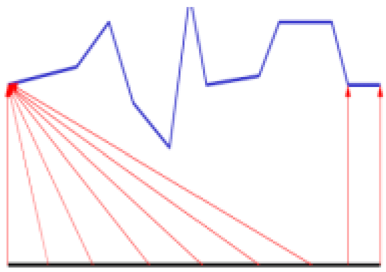

Figure 13

- slope constrain (*Figure 14*): the optimal path in the distance matrix can't have long phases of going only in one direction, thus avoiding short parts from one series from being matched with a long path from the other series. Formally $(j_{sp} - j_{s0})/(i_{sp} - i_{s0}) <= p$ and $(i_{sq} - i_{s0})/(j_{sq} - j_{s0}) <= q$ where p and q are slope parameters.

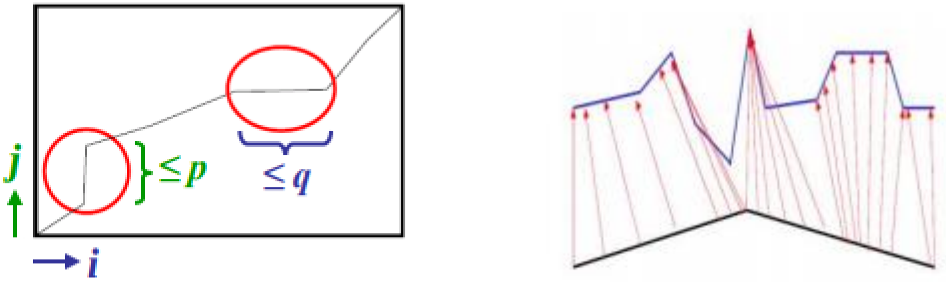

Figure 14

2.1.3 Collaborative recommendations

Broadly speaking, there are two main types of recommendation strategies: content-based recommendations and collaborative recommendations. In both cases the goal is to find items that are interesting for a given user.
The content-based approach uses metadata of the items to calculate similarities between them. If a user has interacted with some items, it can be assumed that he or she will be interested in items with similar content. For example, two action movies can be more similar to each other than to a romantic comedy. If there are multiple action movies in a user history, a content-based algorithm will select further action movies for that user.
In contrast, the collaborative method uses the behavior of the users to determine similarity between items. The underlying assumption is that items that are interesting for similar users can be seen as similar, even if their content looks different. For example, if a user watched five action movies, a film watched by other users with similar history can be recommended to him, even if it is a romantic one.
The RE in the enCOMPASS platform uses collaborative methods. The assumption is the same as described above: looking at what similar users liked and disliked is a very good indicator of what the current user will like and dislike. Our RE defines similar users by looking at the users with the same recommendation group cluster membership and uses the feedback provided by these users for the collaborative calculation.
One of the algorithms used for collaborative recommendations is the nearest neighbor method. For each item we assign an array. The arrayis the length of the users and it contains the feedback given to the user for the item. For example, if five users rated movies in a database the vector for each movie would consist of these ratings. If two users gave 5 stars, two 4 stars and one 1 star for the movie Matrix, it's array will look like: matrix = [5,5,4,4,1]. In real life most of the values will be 0, representing that the user did not give a rating for the item. This method can also be used without explicit ratings. For example in a web shop for every item the vector contains 1 for users who bought the given item and 0 for the ones who did not. Having a user-length array for each item, the cosine similarity between any two items can be calculated as:

$$\frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2} \sqrt{\sum\limits_{i=1}^{n} B_i^2}},$$

The similarity score is determined by the interaction of the users with the items (see example in the *Table 1* below. If a user interacted with an item, the corresponding cell is filled with 1. Otherwise the cell is empty). For a given user, items that are similar to the ones he or she has already interacted with can be recommended.

	User 1	User 2	User 3	User 4
Item A				
Item B		1	1	1
Item C		1		1
Item D	1		1	1
Item E				
Item F		1	1	

Table 1: Example for selecting recommendation

In this example, for recommending an item to User 1, the item most similar to item D (the one that User 1 has already interacted with) can be selected. The item similarity for item D (using the formula described above) is:

- item A - item D: 0
- item B - item D: 0.66
- item C - item D: 0.41
- item E - item D: 0
- item F - item D: 0.41

Thus the best recommendation for User 1 is Item B..

3 RECOMMENDER ALGORITHM

3.1 MESSAGES

Three types of messages are included in the recommendable message set: technically feasible tips, general recommendations and recommendations specific to clusters based on household features. In addition, the recommendations specific to clusters based on household features are enriched with different variations connected to the behavioral change state clusters.

Each week, one message from each message type can be selected for each user. Most messages have recommendation criteria that have to be true for the given user in the given time-period. Also, messages that have already been delivered to the user are excluded from the recommendable ones. Therefore, it is possible that not every user receives one of each type in each week.

3.1.1 Technically feasible tips

The first group of tips, which we refer to as “technically feasible tips” are the ones connected directly to measured conditions of the household. Although these tips are technically feasible, these are the most action oriented, hence the name of the tips group. The tips in this group are triggered by a well described user behavior pattern and aim to change that specific behavior. For each tip, there are criteria that must be present in the user data at least once in the time period considered. For example, for the tip recommending using window shades ("Let's enjoy a bit of shade! To keep your living room cool, shade the windows when you are not at home."), the criteria

is that in the previous weeks there were periods of at least 60 minutes when nobody was present in the living room, but the luminance was higher than 60 lux during daytime. This message can be only recommended in the summertime.

3.1.2 General recommendations

General recommendations encourage energy saving by describing good practices that are easy to follow. They are often connected to the usage of one specific device. Generally, all users can receive them except if the recommendation is strictly about a device does not present in the household or during a season not compatible with the recommendation (e.g. air conditioning recommendation in winter). An example of such a recommendation is: "Leave the fridge in peace! Leave the fridge open as short as possible. Decide first what to put in or take from the fridge. For example, sort shopping first while unpacking, then put in all cooled groceries in one go.”

3.1.3 Recommendations specific to recommendation group clusters

Cluster specific recommendations are aimed specifically at the way of life described by the cluster of each center. These are the most personalized messages helping to save energy in a way that best suits a particular household. An example of such a recommendation is: "Your turn... to save energy! Combine energy-saving with family time! There are more and more energy-related games on the market. Play with your children and discover together the importance of energy savings."

In addition, these recommendations have been enriched with different variations connected to the behavioral change stage clusters to address the need to apply appropriate incentive mechanisms or persuasive technologies depending on users’ current behavioral change stage. This should encourage individuals’ advancement in behavioral change stages and prevent them from slipping back (as suggested by the socio-technical behavior change model by Koroleva et al., 2019).

To design incentive elements of the recommendations addressing specific behavioral change stages in a theoretically sound way, we have followed the approach recommended by Fogg (2009) in his behavior model for persuasive design. The model stipulates that for a person to perform a desired behavior, three factors need to be present simultaneously: motivation, ability and trigger. First, individuals who are lacking motivation to behave in a certain way will not do so, even if performing the action is within their ability. Similarly, a person will not perform a desired behavior if he or she is highly motivated but not able to do so, e.g. because of the lack of necessary skills or resources. Finally, even if the motivation and ability are sufficiently high, one needs an appropriate trigger to perform a behavior – an element of any form prompting a person to act. In enCOMPASS, personalized recommendations are designed to fulfil the trigger function.

Fogg (2009) provides recommendations about how to design triggers aimed at specific individuals characterized by different levels of motivation and ability. He suggests addressing persons who lack motivation by triggers that he refers to as "sparks", which should include an appropriate motivational element. If the missing factor for activation of behavior change is a person's ability, Fogg suggests using a "facilitator" trigger, which should emphasize the simplicity of the considered behavior. Finally, Fogg suggests addressing people with high levels of both motivation and ability with triggers that he refers to as "signals" and which have the role of reminders simply bringing one's attention to the intended behavior.

One can argue that the groups addressed by Fogg in his suggested design of triggers correspond to the three of the behavioral change stages outlined by Koroleva et al. (2019) which are used for behavioral clustering of the enCOMPASS users: pre-conte mplation, contemplation and action (described in more detail in section 3.2.2). For the purpose of the RE, we argue that in terms of motivation and ability levels, the most characteristic feature of the pre-contemplation stage is low level of motivation to change behavior and the most characteristic feature of the contemplation stage is low level of ability to change behavior. Following this logic, the action cluster is characterised by high levels of both motivation and ability to change behavior.Following this simplified mapping, we have followed Fogg's guidelines to personalize recommendations for specific user groups. We have created three versions of each recommendation intended for different behavioral clusters. These differentiated recommendations share the same core message referring to an energy saving action, but their phrasing is intended to address specific behavioral characteristics of a given user group. Accordingly, the versions of recommendations intended for the *pre-contemplation* cluster contain motivational elements. The users of the *contemplation* cluster, in turn, are addressed with a version, which emphasises the simplicity of performing a recommended behavior. Finally, the recommendations aimed at the *action* cluster have been phrased as reminders or casual suggestions. The example in *Table 2* illustrates this approach.

Addressed user group	**Recommendation**	**Explanation**
Generic	A big house uses more energy. Use shades during the summer to prevent warming your house.	-
Pre-contemplation	Lots of energy are needed to cool down a big house using A/C. To <u>enjoy</u> cool temperature in your house in the Summer, close the shades of your windows during the hot days! During the night, open up all the windows to take in the cooler air.	The element of <u>anticipated pleasure</u> is intended to increase users' motivation.
Contemplation	One needs a lot of energy to cool down a big house using A/C or forced ventilation. You can <u>easily</u> avoid it! To prevent your house from excessive warming up on hot summer days, simply draw the windows' shades during the day.	The emphasis on <u>simplicity</u> is intended to increase users' perception of ability.
Action	One needs a lot of energy to cool down a big house using A/C or forced ventilation. <u>Don't forget</u> to close the shades of your windows before leaving the house on hot summer days!	The recommendation phrased as a <u>reminder</u> is intended to raise users' attention to the desired behavior.

Table 2. Example of a generic recommendation personalised to address users in each of the behavioral change stages.

3.2 USER MODEL

For each user a user model is calculated every week. The user model is based on the inputs described in the chapter 1.3. The user model has four major parts, all serving different purposes: Recommendation group cluster membership, Cluster membership based on behavioral change stage, Optimal time for notifications, Lighting, heating and humidity patterns (*Figure 15*).

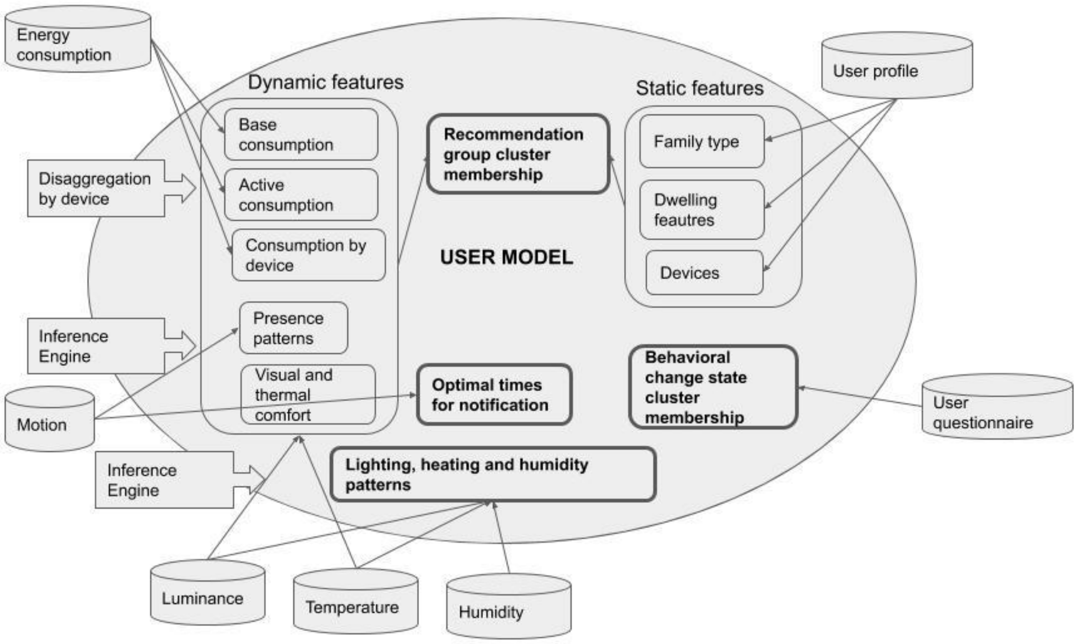

Figure 15. User model

3.2.1 Recommendation group cluster membership

The most complex part of the user model is the recommendation group cluster membership. Each user is assigned to one recommendation group cluster every week. The membership can change in the next week for the same user, but at any time a user can belong to only one cluster. The recommendation group cluster membership is used in two ways:

-it determines the recommendable message set for cluster specific recommendations.

-it determines the input for the collaborative message selection.

The principle behind the RE is that there is no objective ordering of the recommendable items. The usefulness of each item is subjective and there can be a big variety in users' preferences. Also, preference of the same user can change in time as the contextual and psychological conditions change. Thus, a good recommendation is personalized for the current state of the current user. One way to personalize recommendations is to find groups among users who share similar measurable features. We can assume that the subjective usefulness of different items is more similar within these groups compared to other groups. Therefore, the purpose of creating recommendation group clusters inside the enCOMPASS platform is to create groups for collaborative score calculation. Using the collaborative score enables the RE to find recommendations that can be personally useful for a user.

As mentioned above, user preferences can only be modelled if current context is included. For this reason, the features used in our clustering include dynamic ones that are based on the users' behavior in the close past. This way, the model can follow the change in users' lifestyle while also accounting for the stable features.

Data: feature engineering

In deciding how to use the input data two goals were considered: on the one hand, we want to see a complex picture about users capturing as many aspects of their way of life as possible. On the other hand, data needs to be aggregated and the most telling features should be selected. If all data was included in the model without aggregation, the resulting cluster structure would consist of too many groups each including only a few users.

As the clusters are used for recommendation selection, while creating the groups we also considered that clusters should be connectable to the recommendations of the enCOMPASS platform. This means that features used for cluster creation were selected to be easily interpretable and to include motives mentioned in the recommendations (e.g. type of energy source).

As mentioned above, both static and dynamic (based on temporary behavior) features are used in the model.

Static features

These features are based on the data in the user profile

-family type:

aggregate feature using the no_of_adults_older_than_16 and no_of_kids_younger_than_16 columns from the database. The most salient feature of the household is captured. Values can be: children (no_of_kids_younger_than_16>0), no child (no_of_kids_younger_than_16=0), 1 adult, 1-2 adults.

-house type:

raw feature from user profile.

-number of rooms:

two categories are used: 1-3 rooms, 4-5 rooms.

-heating type:

raw feature from user profile.

-heating source:

raw feature from user profile.

-lighting type:

only used if lighting type is traditional (incandescent light bulb).

-devices in the household:

raw feature from user profile.

Dynamic features

These features are aggregated based on data from the previous 3 weeks.

-energy consumption:

uses the energy consumption data coming from the smart meters. Two types of energy consumption are distinguished: base consumption is calculated from the measured consumption values at 03:00, 03:15, 03:30, 03:45 AM whereas active consumption is calculated from time periods when the user is neither sleeping nor absent based on the output of the Activity Inference component. All users' consumption types are compared to each other, and thus we got categorization based on it: Households in the lower 40% consumption group are categorized as 'low', households in the upper 40% categorized as 'high' and the remaining households are categorized as 'average' consumption separately for base and active consumption.

-disaggregated consumption:

uses the output of the Disaggregation Engine component. Separate features are calculated for the following devices: washing machine, tumble dryer, dishwasher, electric oven, heat pump. For each device mean consumption is used and device-specific categories are calculated: households in the lower 40% are categorized as 'low', households in the upper 40% categorized as 'high' and the remaining households are categorized as 'average' separately for each device.

-presence pattern:

uses the output of Activity Inference component. For this feature, each user can have multiple patterns at the same time. The possible patterns are:

- away on weekends: there was at least one weekend when the user was absent more than 50% of the time.
- away at work time: the user was absent more than 50% of the time between 8AM and 7PM on workdays
- home during the day: the user was not absent on between 8AM and 7PM more than 50% of the time.
- home during afternoons: the user was not absent between 2PM and 7PM more than 50% of the time.
- home most of the time: the user was not absent between 8AM and 11PM more than 50% of the time.

-comfort level:

uses the output of the Comfort level inference component. Visual and thermal comfort are separate features. The mean comfort level for the period is calculated for each user. If the mean is above 0, the user is categorized as 'high', otherwise the user is categorised as 'low' for the given feature.

Clustering: method

The theoretical background of clustering methods is described in the chapter 2. As the purpose of the clusters is to support message selection in the RE, we needed the clusters to be easily interpretable and connectable to recommendable tips. We decided to use a stable cluster model in the engine. This means that the cluster centers were fixed while the cluster membership of each user is recalculated every week. The cluster centers were created using the aggregated features described above and based on data analysis and expert knowledge..

Clustering: results

We have defined 13 cluster centers. Each cluster was given a name for easier interpretation and identification. While these names can't capture all the features of the cluster centers, all features defined are equally important. For the centers, the most salient features were defined, while the other features can be present or not without changing the closeness to the cluster center For example, the presence of some devices can be important for a cluster center, while the presence of other devices are not taken into account. E.g. for 'Device lover freelancer', cluster center “the presence of a laptop” is defined while the presence of an AC is not taken into account.

The cluster centers are shown in *Tables 3* and *4*.

Cluster center/Static features	**family type**	**house type**	**No. of rooms**	**heating type**	**heating source**	**lightning type**	**devices**
Family in a big house	children		4-5				AC, dishwasher
Family in a small house	children		1-3	no heat pump			no AC, no washing machine
Energy aware family	children		1-3				no AC, no washing machine
Working couple	1-2 adults		1-3				laptop, TV
Energy aware with low intensity lighting				heat pump			heat pump, no AC
Device lover freelancer	no child						TV, hi-fi, desktop, laptop, gaming-set
IT child	children						TV, hi-fi, desktop, laptop, gaming-set
Home alone	1 adult	apartment/independent house	1-3				
Oil radiator, not energy focused			4-5	radiator	oil		
Average working person							water boiler
Not electric heating in a big house			4-5		not electric		no heat pump
20th century guy						traditional	no heat pump, no water boiler

E-car owner							electric car

Table 3: Cluster centers defined based on static features

Cluster centre/Dynamic features	**energy consumption**	**disaggregated consumption**	**presence pattern**	**comfort level**
Family in a big house	high active	high dishwasher, high washing machine	away on weekend	
Family in a small house	low active			
Energy aware family	low active			
Working couple	high active	low dishwasher, low electric oven	away at work time	
Energy aware with low intensity lighting		high heat pump		high thermal, low visual
Device lover freelancer	high active	high tumble dryer, high dishwasher	home during the day	
IT child	high active		home during afternoon	
Home alone	high active, low base		home most times	
Oil radiator, not energy focused	high base			
Average working person	average active, average base		away at work time	
Not electric heating in a big house	low base			
20th century guy	high active			low visual
E-car owner				

Table 4: Cluster centres defined based on dynamic features

3.2.2 Cluster membership based on behavioral change stages

Behavioral change state cluster membership is stable through the pilot period. Each user receives one membership that is not recalculated. This membership determines the variation of the cluster specific recommendation.
Users without questionnaire data are not assigned to behavioral change state clusters and receive the general version of the cluster specific recommendations.

The users have been clustered into groups corresponding to three of the stages of behavioral change described by Koroleva et al. (2019) in the socio-technical behavior change process model: pre-contemplation, contemplation and action. This has been done based on the surveys conducted in the initial phase of the pilots, specifically a selection of variables measuring users' antecedents of behavior change towards energy saving.

The socio-technical behavior change process model described by Koroleva et al. (2019) in the context of energy saving distinguishes four stages of behavioral change: pre-contemplation, contemplation, action and monitoring. Persons belonging to the *pre-contemplation* stage are characterized by low awareness of the need to change and in turn by the lack of intention to change (for example to take energy-saving measures). The next stage, *contemplation*, is comprised of individuals who are aware of the necessity to change their behavior but do not do it due to situational factors, e.g. lack of knowledge of how to save energy. Further up on the behavior change ladder are persons in the *action* and *monitoring* stages. Both of these groups are characterized by high levels of awareness of the need to change, knowledge of how to do so and consequently – intention to save energy. What distinguishes the two stages are their actual energy-saving actions: while the individuals in the *action* stage take their first actual steps to save energy, in the *monitoring* stage, the focus lies on persistence of the desired behavior and preventing the persons from slipping back to previous habits.

Although there is no straightforward procedure to classify pilot users into particular behavioral change stages, we have used a set of constructs measured in the survey that closely correspond to features characterizing each stage. These are the following:

- **Ascription of responsibility** measuring one's sense of responsibility for energy-related problems**,**
- **Personal norm** measuring one's obligation to save energy,
- **Self-efficacy** measuring one's sense of ability to save energy**,**
- **Behavioral intentions to save energy** measuring one's intention to save energy.

One can intuitively characterize the behavioral change stages using the above constructs. In our mapping, the *pre-contemplation* stage is characterized by generally low levels of all four constructs, in particular of ascription of responsibility and personal norm. In the case of the contemplation stage, while ascription of responsibility and personal norm are somehow higher, the self-efficacy remains too low for the users to take action. Finally, in the case of the action and monitoring stages, the levels of all four constructs – in particular of behavioral intentions to save energy – are high. Due to the similarity in operationalisation of these two behavioral stages, in our data analysis, they have been considered jointly and for simplicity referred to as action. *Table 5* illustrates schematically how the measured constructs map to each of the analysed behavioral change stages.

Behavioral Change Stage	**Personal norm**	**Ascription of responsibility**	**Self-efficacy**	**Behavioral intention to save energy**
Pre-contemplation	Low	Low	Low	Low
Contemplation	High	High	Low	High
Action + monitoring	High	High	High	High

Table 5. Mapping of behavioral constructs to behavioral change stages.

The values of the four constructs described above have been calculated as averages of several validated measurement instruments obtained in the survey conducted for all the pilots in the initial phase of their duration. This necessarily limited the analysis to the users who have completed the first survey. We estimated users' ascription of responsibility using four items measured on a 1-5 Likert scale in line with Steg et al. (2005), e.g. *I am jointly responsible for the energy problems*. To assess personal norm, we have used five items measured on a 1-7 Likert scale in line with Steg et al. (2005), e.g. *I feel morally obligated to save energy*. Self-efficacy reflecting perceived behavioral control was estimated using three items on a 1-7 Likert scale based on Thøgersen and Grønhøj (2010), e.g. *I believe that I'm able to avoid all unnecessary electricity consumption in my home*. To assess users' behavioral intention to save energy, we have asked them to evaluate on a Likert scale 1-7 three statements in line with Ajzen (1991), e.g. *I will try to save a substantial amount of energy in the next three months*.

Several users who completed the survey did not provide answers to all these questions. In cases where less than half of questions forming a given construct has been left unanswered, we have imputed the missing values using averages of the addressed questions. In other cases, we have dropped the observations from the analysed sample.

We have grouped users into behavioral change stages based on their score in terms of the measured constructs using cluster analysis. We have performed the analysis separately for each of the pilots, in each case using the Euclidean distance as dissimilarity measure and the Ward's method for hierarchical clustering.

In all the three pilots, it was possible to cluster users with respect to the behavioral constructs. While in the cases of Germany and Greece the correspondence of results to the behavior change theory was relatively clear, in the case of Switzerland, the interpretation was somewhat less straightforward. Nevertheless, the groupings were clear enough for the purposes of the recommender.

Germany

The algorithm applied to the German pilot data has grouped the total of 90 users who had completed the first survey into four clusters. The levels of the measured constructs in each of the resulting user clusters are illustrated in *Figure 16* and summarized quantitatively in *Table 6* (in Appendix). The analysis of particular clusters reveals that they closely correspond to the behavioral change stages.

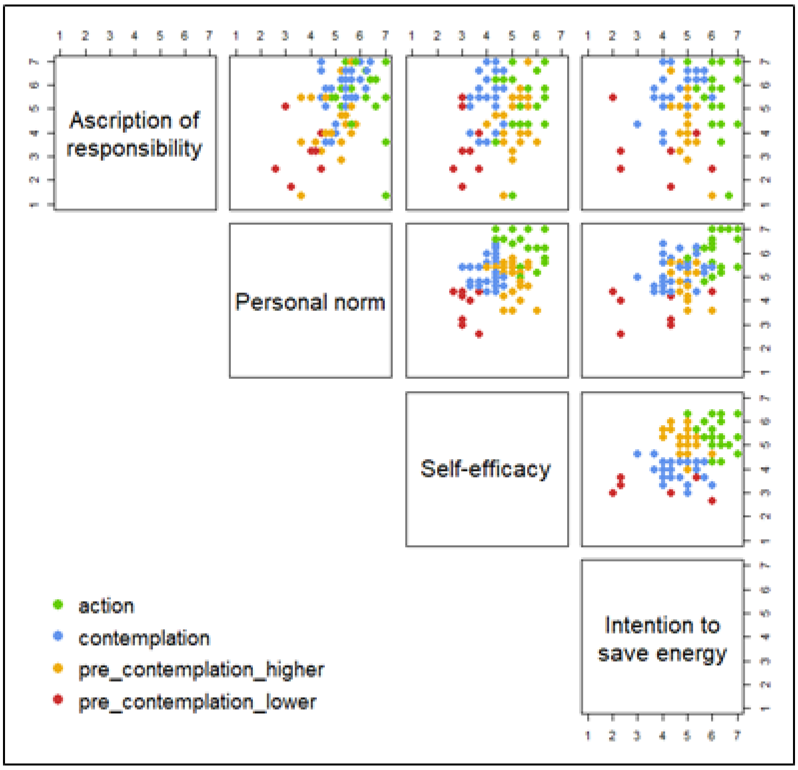

Figure 16: Clustering results - German pilot.

Note: in the graph, the variable ascription of responsibility is rescaled from its original scale 1-5 to the scale of the remaining variables (1-7) to facilitate easy comparison.

Two of the clusters (marked in Figure 16 with red and orange) share features particularly characteristic for the *pre-contemplation stage* namely relatively low values of constructs ascription of responsibility and personal norm. While the first cluster (*pre_contemplation_lower*, 8 users) is characterized by the lowest average values of all four considered constructs, in the second one (*pre_contemplation_higher*, 27 users), the average values of the examined variables are somewhat higher. This relates in particular to the construct self-efficacy with its mean above the pilot average. Nevertheless, due to low values of ascription of responsibility and personal norm, for the purpose of constructing the recommender, the two groups have been merged into a single *pre_contemplation* cluster with a total of 35 users.

The next cluster (marked blue, 31 users) is characterized by somewhat higher values of most of the examined constructs. Most notably, compared to both *pre_contemplation* groups, the users in this cluster have high average ascription of responsibility and personal norm. At the same time, they appear to have particularly low level of self-efficacy. Therefore, we have interpreted this cluster as representing the *contemplation* stage.

Due to high values of all considered constructs, one can easily notice that the last group of users (green, 24 households) corresponds closest to the *action* stage. The persons belonging to this cluster declare high ascription of responsibility and personal norm. They also appear to have high level of self-efficacy and express intention to save energy.

Greece

The results of cluster analysis performed on the 108 users of the Greek pilot for whom survey data was available are comparable to the German case. *Figure 17* depicts the four resulting clusters and *Table 7* (in Appendix) summarizes their main characteristics.

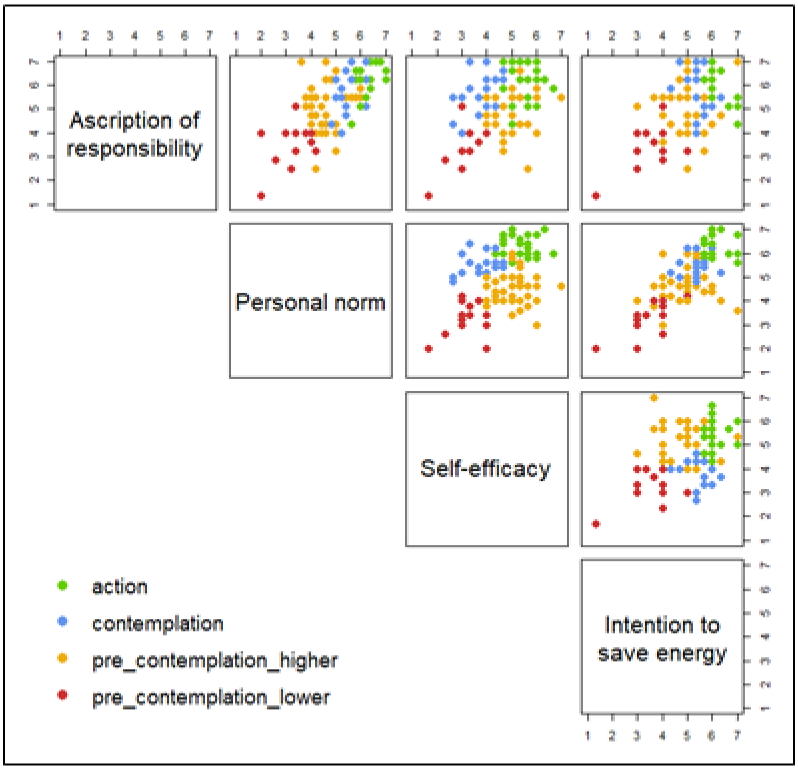

Figure 17: Clustering results - Greek pilot.

Note: in the graph, the variable ascription of responsibility is rescaled from its original scale 1-5 to the scale of the remaining variables (1-7) to facilitate easy comparison.

Like in the case of Germany, one can distinguish two clusters, which for the purpose of the recommender can be jointly interpreted as the *pre_contemplation* stage of behavioral change (marked red and orange in *Figure 17*, 64 users in total). The first of them (*pre_contemplation_lower*, 22 respondents) contains users who – on average – have very low values of all four constructs. While the second one (*pre_contemplation_higher*, 42 respondents) appears to be generally higher on the spectrum of the four variables, its average ascription of responsibility and personal norm remain relatively low in comparison to the remaining users in the pilot. This supports merging the two clusters.

The persons classified into the next cluster (blue, 18 users) are characterized by somewhat higher values of most of the constructs: ascription of responsibility, personal norm and intention to save energy. This might be an indication that they are more advanced in the behavioral change process than the *pre_contemplation* group of respondents. On the other hand, they have particularly low average level of self-efficacy, which led to interpreting this cluster as the *contemplation* stage.

Last but not least, the remaining cluster (green, 26 users) corresponds clearly to the *action* stage of the behavioral change process. With the values of all four considered constructs above the pilot average, the users in this group outperform the remaining participants in terms of all analysed dimensions.

Switzerland

In the case of Switzerland, we could also identify clusters corresponding to the behavioral constructs. However, their interpretation in line with the socio-technical behavioral change process model was not as straightforward as in the German and the Greek case. *Figure 18* depicts the obtained clustering results and *Table 8* (in Appendix) provides an overview of a selection of quantitative metrics.

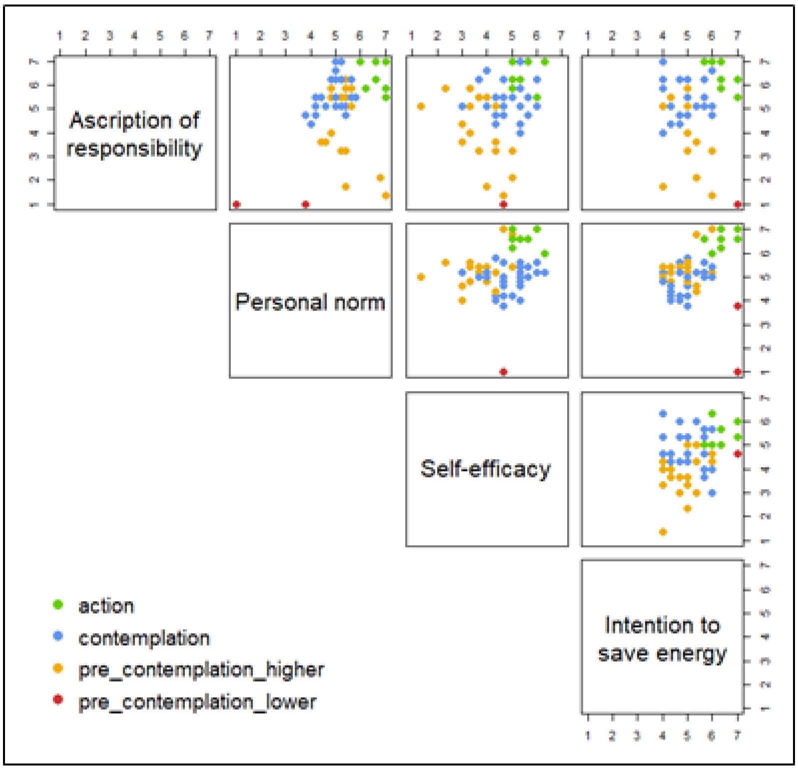

Figure 18. Clustering results - Swiss pilot.

Note: in the graph, the variable ascription of responsibility is rescaled from its original scale 1-5 to the scale of the remaining variables (1-7) to facilitate easy comparison.

The cluster comprised of users characterized by relatively high values of all four constructs was easiest to interpret. This relatively small group (green in *Figure 18*, 9 users) can be quite clearly distinguished as having higher values of constructs personal norm and self-efficacy than the remaining clusters. In terms of ascription of responsibility and intention to save energy, the individuals in this group are also in the upper spectrum of the values of the whole pilot. Therefore, this cluster has been identified as corresponding to the *action* stage.

Two other clusters (blue and orange) are more intertwined – their levels of personal norm and intention to save energy have very similar values (both slightly below or close to pilot averages). However, the 20 users of the orange cluster appear to have significantly lower ascription of responsibility and self-efficacy, which has convinced us to interpret them in terms of the behavioral change theory as belonging to the *pre-contemplation* stage.

The values of these variables in the other cluster (blue, 29 users) are higher but still below those in the group identified as the *action* stage. For that reason, we have interpreted it as representing users who are in the *contemplation* stage. This interpretation, however, is not unambiguous due to the cluster's relatively high values of self-efficacy, which – in line with the socio-technical behavior change process model – one would expect to be low. On the other hand, the self-efficacy of the users in this cluster remains on average below that of the users in the action stage, which indicates that there is still room for improvement in that respect.

Finally, the clustering algorithm has classified two users marked red in *Figure 18* as a single cluster without assigning them to any of the larger groups. Due to the values that they take, which in the framework of the behavior change model appear contradictory (e.g. very low levels of ascription of responsibility and personal norm accompanied with a very high level of intention to save), one could discard them as outliers. However, for the purpose of the recommender, we merge them with the *pre-contemplation* cluster on the grounds of particularly low levels of ascription of responsibility and personal norm.

3.2.3 Optimal time for notifications

The User model of the RE has another feature, which is the identification of the optimal type for notification.

Each message is enriched with a notification time category. For each category, a personalized optimal notification time is calculated for each user every week. This calculation uses the data from the motion sensor in the household. This part of the user model is used for personalizing the time of notification, however, as described in 1.4 this feature is not taken into consideration by the NE.

3.2.4 Lighting, heating and humidity patterns

Another information for the recommender, which is user for the User model are data coming from the smart meter sensors. This information is used for checking if the criteria of a technically feasible tip were present in the previous time period. Patterns present in the user data are stored in the user model. These patterns are recalculated every week.

3.3 Recommender logic

3.3.1 Calculating recommendation group clusters

The RE uses a custom cluster assessment algorithm for assigning the users into the recommendation group clusters. As described above, the clustering is based on aggregated (static and dynamic) features and the cluster centers are fixed. The custom algorithm is responsible for assigning the users around these centers in a way that the users within one group are more similar to each other and to the cluster center than to users in other groups and other cluster centers.
For each day at least 85% of data should be available for each user. Users without enough data are placed to general recommendation group cluster.

The steps of the algorithm are described here and shown in *Figure 19*.

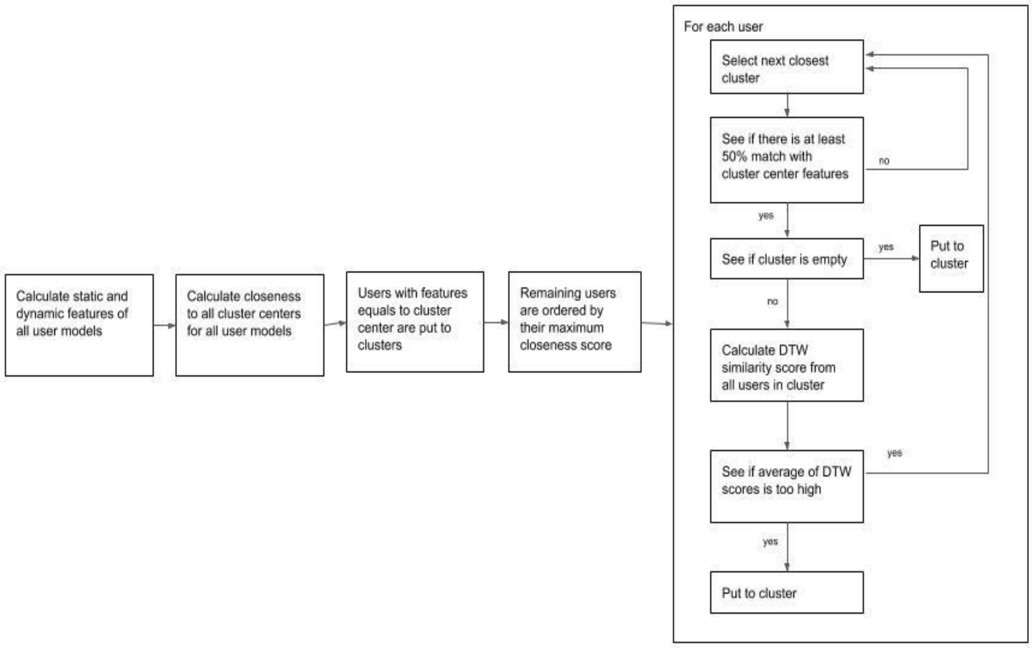

Figure 19: steps of RE computation

First, static features are taken from the user model and dynamic features are recalculated for the users considering data from the previous time period. This results in a vector of zeros and ones in the length of all possible feature values (1-n) for each user. f_{kj} is the feature j for user k. The value of k_{kj} is 1 if the feature is true for user k.

$f_{kj} = \{0,1\}$
$u_k = (f_{k1}, f_{k2} \dots f_{k\,n})$

Then the closeness to all cluster centers is calculated for all user models. The closeness score is calculated by summing the elements of the user vector that are present in the cluster center. For each cluster center c, a vector is assigned containing ones for the defining features and zeros for the not restricted features. The closeness score (C_{ik}) is the dot product of the cluster center vector (c_i) and the user feature vector (u_k).

$c_{lj} = \{0,1\}$
$c_l = (c_{l1}\ c_{l2} \dots c_{ln})$

$C_{lk} = u_k * c_l = \sum_{i=1}^{n} u_{ki} * c_{li} = u_{k1}*c_{l1} + u_{k2}*c_{l2} + \dots + u_{kn}*c_{ln}$

Also, for all cluster centers, the percentage of the features that are present in the user vector - is calculated.

$P_{lk} = C_{lk}*100/(c_{l1} + c_{l2} + \dots + c_{ln})$

Users having a perfect match (100% matching features) for a cluster are assigned to that cluster. The remaining users are ordered by their maximum closeness score. Going in this order, the algorithm described below assigns each user to a cluster.

If cluster match is below 50%, the algorithm proceeds to the cluster with the next highest cluster closeness score. If at least half of the features match, and the cluster is currently empty the user is assigned to this cluster. If the cluster is not empty, - a pariwise dynamic time warping similarity score (DTW) is calculated between the current user's energy consumption time series data and the energy consumption time series of the users in the cluster. If the average of these DTW scores is above 25 (meaning that the user's energy consumption pattern is very different from the cluster members', see examples in *Figure 20-21*), the algorithm starts from the beginning for the cluster with the next highest score. If the average DTW score is below 26, the user is assigned to the current cluster.

This figure shows a case when the average DTW score = 66 (the user is not assigned to the cluster). On the axis the time is shown (one datapoint for every 15 minutes), on the y axis the energy consumption is shown (kWh).

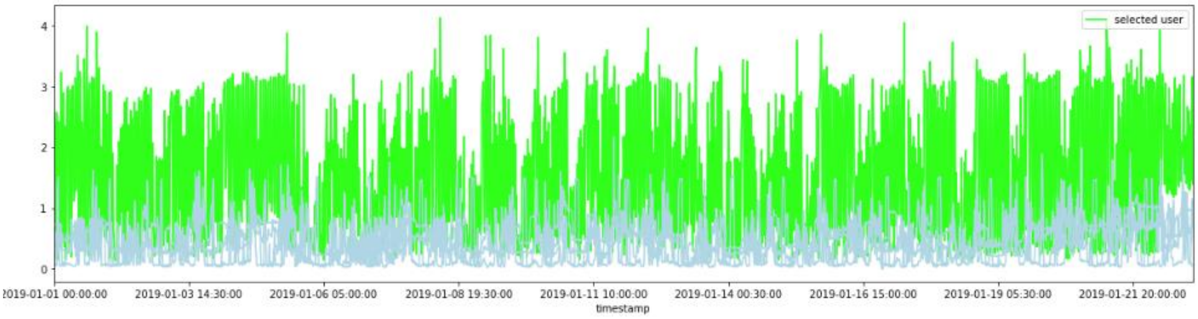

Figure 20: Example for high DTW score

This figure shows a case when the average DTW score = 21 (the user is put to the cluster)

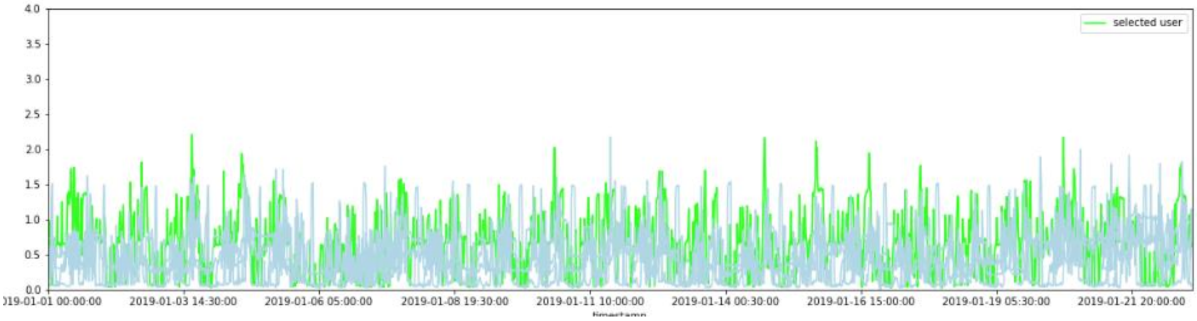

Figure 21: Example for low DTW score

3.3.2 Selecting and timing messages

This section describes the process of selecting and timing messages. Message selection process is different for the different types of tips (technically feasible, general, cluster specific) and uses different parts of the user model.

3.3.2.1 Selecting technically feasible tips

Figure 22 depicts the process of selecting technically feasible tips.

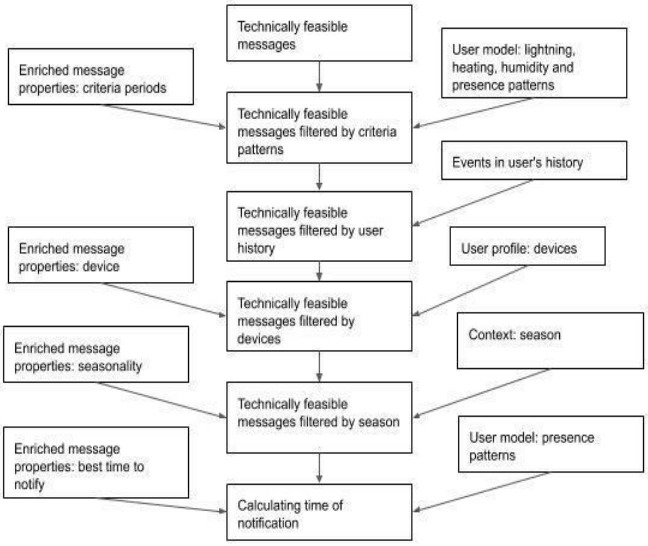

Figure 22: Process of selecting technically feasible tips.

For users and for all items (j is in the range of 1 - N (N=16 for technically feasible tips), the recommender calculates the score p_{ij} which has a starting value of 1.

$p^1_{ij} = 1$

For messages that have been already recommended to the user, the score is set to 0.

$$\begin{cases} item\ in\ user's\ history\ p_{ij}^2 = 0 \\ item\ not\ in\ user's\ history\ p_{ij}^2 = p_{ij}^1 \end{cases}$$

Messages about unowned devices or off season are filtered out.

$$\begin{cases} unowned\ device\ p_{ij}^3 = 0 \\ owned\ device\ or\ not\ connected\ to\ device\ p_{ij}^3 = p_{ij}^2 \end{cases}$$

Messages that are off season are filtered out.

$$\begin{cases} off\ season\ p_{ij}^5 = 0 \\ on\ season\ or\ not\ connected\ to\ season\ p_{ij}^5 = p_{ij}^4 \end{cases}$$

Messages with criteria not matching sensor patterns in user model are filtered out.

$$\begin{cases} not\ matching\ criteria\ p_{ij}^6 = 0 \\ matching\ criteria\ p_{ij}^6 = p_{ij}^5 \end{cases}$$

For making the recommendations less deterministic and to select from items with the same score, a random noise r_{ij} from a uniform distribution between 0.001 and 0.009 is added to each item.

$r_{ij} \sim U(0.001, 0.009)$

$p^7_{ij} = p^6_{ij} + r_{ij}$

For each day at least 85% of data should be available for each user. Users without enough data don't receive technically feasible recommendations.

3.3.2.2 Selecting general recommendations

Figure 23 depicts the process of selecting general recommendations.

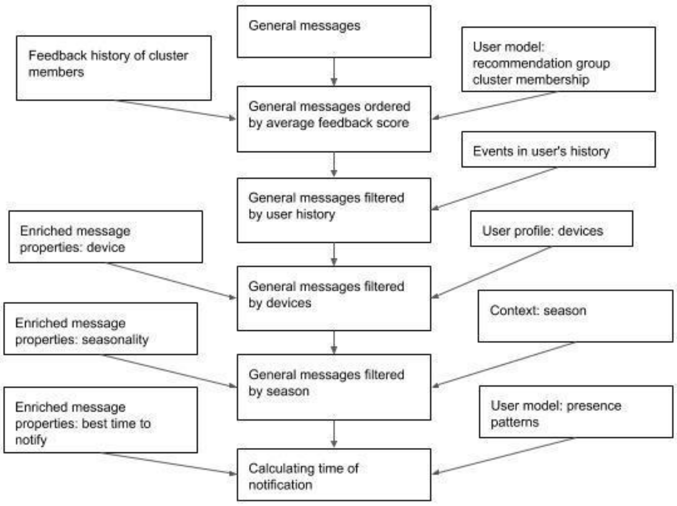

Figure 23: Process of selecting general recommendations

For users and for all items (j is in the range of 1 - N (N=71 for general tips) the recommender calculates the collaborative score. For this, all the previous feedback is collected from the user's cluster ($feedback_{im}$, m is in the range of 1 - M the number of feedbacks for tip j from cluster i) and the mean is calculated for all items.

$$p^1_{ij} = feedback_{1j} + feedabck_{2j} + ... + feedback_{mj}/M$$

For messages that have been already recommended to the user, the score is set to 0.

$$\begin{cases} item\ in\ user's\ history\ \ p^2_{ij} = 0 \\ item\ not\ in\ user's\ history\ \ p^2_{ij} = p^1_{ij} \end{cases}$$

Messages about unowned devices or off season are filtered out.

$$\begin{cases} unowned\ \ device\ p^3_{ij} = 0 \\ owned\ \ device\ or\ not\ connected\ to\ device\ p^3_{ij} = p^2_{ij} \end{cases}$$

Messages that are off season are filtered out.

$$\begin{cases} off\ \ season\ \ p^5_{ij} = 0 \\ on\ season\ or\ not\ connected\ to\ season\ \ p^5_{ij} = p^4_{ij} \end{cases}$$

3.3.2.3 Selecting cluster specific recommendations

Figure 24 depicts the process of selecting cluster specific recommendations.

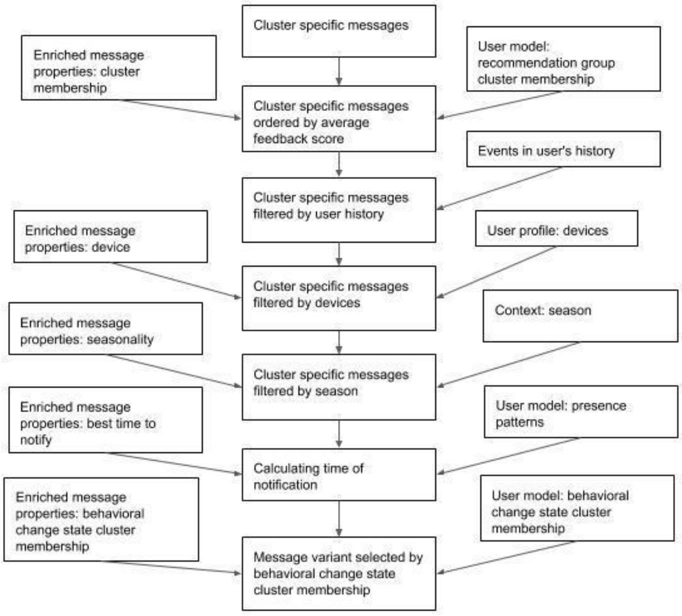

Figure 24: process of selecting cluster specific recommendations

For users and for all items (j is in the range of 1 - N (N=31 for cluster specific recommendations) the recommendations assigned to the user's cluster are selected. For those recommendations the recommender calculates the collaborative score. For doing that all the previous feedback is collected from the user's cluster ($feedback_{im}$, m is in the range of 1 - M the number of feedbacks for recommendation j from cluster i) and the mean is calculated for all items.

p^1_{ij} = $feedback_{1j}$ + $feedabck_{2j}$ + …. + $feedback_{mj}$/M

For messages that have been already recommended to the user, the score is set to 0.

$$\begin{cases} item\ in\ user's\ history\ \ p_{ij}^{2}=0 \\ item\ not\ in\ user's\ history\ \ p_{ij}^{2}=p_{ij}^{1} \end{cases}$$

Messages about unowned devices or off season are filtered out.

$$\begin{cases} unowned\ \ device\ p_{ij}^{3}=0 \\ owned\ \ device\ or\ not\ connected\ to\ device\ p_{ij}^{3}=p_{ij}^{2} \end{cases}$$

Messages that are off season are filtered out.

$$\begin{cases} off\ \ season\ p_{ij}^{4}=0 \\ on\ season\ or\ not\ connected\ to\ season\ p_{ij}^{4}=p_{ij}^{3} \end{cases}$$

Message variation matching the behavioral change state cluster membership of the user is selected.

3.3.2.4 Selecting messages and calculating the best time to send out recommendations[3]

For every week messages m from each message type with the highest scores are selected for each user. Let j be the index of messages ordered by their score for each user, i the identifier for the user and p the score for the given message.

$p_{ij} : p_{ij} \geq p_{ij-1}$

$m_{ik} \in \{p_{i1}, p_{i2} \dots p_{ik}\}$

For calculating the time of notification for each selected message 4 time categories were defined. For each message a time category t_j was chosen:

T = {'morning','evening','weekday_arrive_home','weekend_usually_home'}

$t_j \in T$

For messages where the time category feature is weekday_arrive_home a weekday is randomly chosen from the next 7 days. Where the time category feature is weekend_usually_home a weekend is randomly chosen from the next 7 days. Otherwise any other day is chosen from the next 7 days.

Next, the hour of sending the notification is calculated. For each user and for each time category, a time interval I_{it} was calculated and stored in the user model:

$H = \{9,10,11,12,13,14,15,16,17,18,19,20,21,22\}$

$I_{it} \subseteq H$

For each message, an hour h_{ij} was chosen randomly from the time interval for the given time category for the given user.

$h_{ij} \in I_{it}$

4 FUNCTIONALITIES OF THE DEMONSTRATOR

The purpose of the demonstrator app is to illustrate the working of the RE. As mentioned in chapter 1, inner states of the RE are shown in the demonstrator that are hidden in the platform. Also, the dataset of the demonstrator is offline and not changing, making the recommendation logic re-playable.

In some points the RE inside the enCOMPASS platform and the demonstration works differently. From a structural point of view, the demonstrator is not connected to the platform in production. While inside the platform the RE runs continuously digesting new data every day, the demonstrator app only simulates the running of two days: one in wintertime (2019/01/23) and one in summertime (2019/07/08). Re-running these days in the demonstrator doesn't interact with the enCOMPASS platform. For the demonstrator, data from 2019-01-01 to 2019-01-22 and from 2019-06-17 to 2019-07-07 was used from all data sources.

[3] It is an important feature of the RE, however, it works as described here only in the demo version. The reason is the following: in the real enCOMPASS platform we needed to harmonise different aims as we have messages from the different components (i.e. gamification engine, inference engine) and the users have preference for notification frequency. Therefore, we needed to elaborate a common notification rule. In this rule RE's messages are attached to the notification queue with high priority.

In the next paragraphs the main features of the demonstrator are described.

- Clustering of users (recommendation group clusters)

The demo app starts with the recommendation group clustering of users into the recommendation group clusters. While in the RE this is a "black box" process, the demonstrator shows steps of the clustering algorithm. Also, the cluster membership of each user can be seen for both time periods.

- Household selection by user profile information

Energy consumption and sensor data together with the output of different components of the enCOMPASS platform and responses from initial questionnaires for 54 households in Switzerland are used in the demo app. The features calculated from these sources together with the cluster membership information are shown in the demo. Any household can be selected to simulate the recommendations it would get by clicking on it.

- Simulating recommendation

The demo app simulates the recommendation process. Two periods can be chosen for the simulation: one during the heating season and one during the cooling season. For each period, the recommendations and feedback from the preceding time is taken into account.

- Presenting the result of the recommendation and the user model

For each period, messages that would be sent to the user in the following week are shown. For each message, the date when it starts to show on the user's tab and the hour when the notification about the message would be sent out day are presented. Thus, the optimal notification time is visible in the demo version.

Logs for the recommendation selection process are also shown. These are calculated during the recommendation process but only presented for the demo. During the real recommendation period, these will be only used by the algorithms, but won't be shown anywhere.

5 EXAMPLE RESULTS

In the next section we describe different examples of the results of the RE. For a better understanding, we show some cases which do appear in the demo as well.

5.1 RECOMMENDATION GROUP CLUSTER

5.1.1 Changing cluster

As described previously, dynamic features are included in the recommendation group cluster selection. This way the change in behavioral patterns can be followed and the RE considers the current state of the household. If the energy consumption patterns change, the household can be assigned to a new cluster.
Let's look at household 86 in the demo application. In the first period (winter 2019) it was assigned to 'Device lover freelancer cluster'. Apart from the matching static features, the household also showed consumption

patterns in line with the cluster center: high active consumption in general and according to the Disaggregation engine component, high device consumption for dishwasher and tumble dryer.
In the second period of the demonstrator (summer 2019) these patterns changed: the general active consumption and the dishwasher consumption changed to category low and the tumble dryer consumption changed to average. At the same time the average thermal and visual comfort level changed to high for thermal and low for visual. These patterns together made the household closer to the 'Energy aware with low intensity' cluster center.
As can be seen in the demonstrator app, the recommendation group cluster membership of this household is different in the two periods.

5.1.2 Collaborative score

Collaborative scoring is one of the main features of the RE. The best way for selecting messages which are personally useful for a household is to look at what recommendations did other, similar household find useful. For example, let's look at household 146 in the second period of the demonstrator tool. It was assigned to the 'Device lover freelancer' recommendation group cluster. For selecting a general message, feedback from other members of that cluster was used. *Figure 25* shows the collaborative score for all messages for the current cluster. The lower the collaborative score is, the better the average feedback from other users (recommendations without score were not rated by other users). As can be seen in the demonstrator, the algorithm selected the recommendation with the lowest collaborative score for this household:
"Standby is an energy leech, Turn off standby mode!" (oid 101).
Looking at the static features (a lot of electric devices) and dynamic features (high active consumption) of this household, it can be seen that the selected message can be especially useful for them.

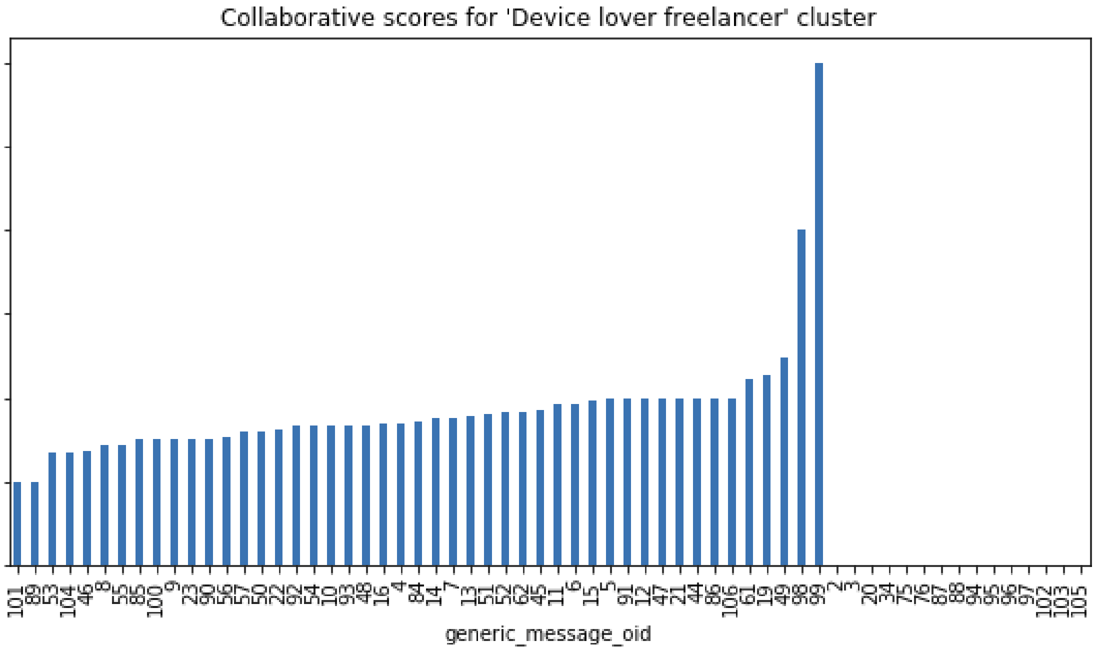

Figure 25: Collaborative scores

5.2 TECHNICALLY FEASIBLE TIPS

Technically feasible tips refer to measurable features within the household, or activity of the user. In this terms these recommendations are all enriched with behavioral pattern criteria. To examine if these criteria were true in the examined period, data from the smart sensors are analysed. The recommendations are intended to change specific behaviors with very actionable insights. For example, household 81 received this technically feasible tip during the summer period:
"Let's enjoy a bit of shade! To keep your living room cool, shade the windows when you are not at home."
The criteria for this message is that in the previous weeks there were periods of at least 60 minutes when nobody was present in the living room, but the luminance was higher than 60 lux during daytime. This message can be only recommended in summertime.
As can be seen in the demonstrator, in the household of user 81 there were 19 periods when this criteria was true.

5.3 RECOMMENDATION GROUP CLUSTER SPECIFIC RECOMMENDATIONS

As third version for the recommendations we choose behavior cluster specific recommendations for the users. As it is seen in the next two examples beside behavioral clusters, other features are also taken into consideration. Cluster specific recommendations encourage energy saving behavior in a way that suits the household the most. An example for pre-contemplation stage user, who is family oriented:

Household 127 was assigned to the 'Family in a big house' recommendation group cluster in the second time period of the demonstrator tool. In terms of the behavioral change stage clusters, they were assigned to the pre_contamplation group. As can be seen in the demonstrator, in the second period the following cluster-specific recommendation was selected for them:
"Your turn... to save energy! Combine energy-saving with family time! There are more and more energy-related games on the market. Play with your children and discover together the importance of energy savings."
This recommendation captures the family status of this household and the phrasing is targeted for their behavioral change state.Another example of cluster specific recommendation (contemplation phase) can be seen for household 142 in the first period. The household was assigned to the 'Family in a small house' recommendation group cluster and the contemplation behavioral change stage cluster. The cluster-specific message selected for it encourages the installation of a heat pump in a way that is anticipated to motivate them best based on the psychological features of contemplation cluster:
"Generate energy yourself. A one-off investment for lasting savings! Consider installing a heat pump combined with solar panel and storage battery. You can become energy self-sufficient."

6 HOW TO USE THE DEMONSTRATOR (USER GUIDE)

The demonstrator can be reached at http://212.32.240.74:5000/ with basic http authentication using the following log-in credentials:

username: enCOMPASS
password: sWX8JW9Des

After opening the app, first, a time period needs to be selected. Winter period refers to a simulation for date 2019-01-23 and summer period refers to a simulation for date 2019-07-08. The data for the winter period calculation is from the time between 2019-01-01 and 2019-01-22 and the data for the summer period calculation is from the time between 2019-06-17 and 2019-07-07.

Figure 26: Time period selcetion in the demonstrator

After period selection one needs to click on "Execute clustering" to start the clustering process.

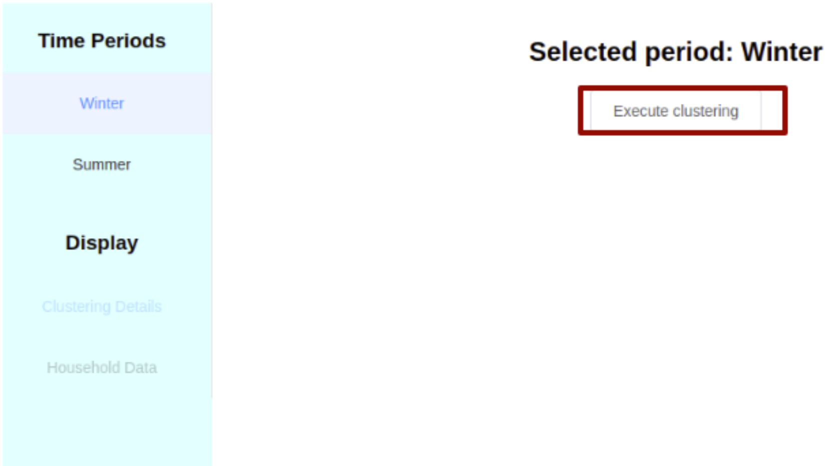

Figure 27: Execute clustering in the demonstrator

The process takes some time. When it's finished, the logging appears and the buttons "Clustering Details" and "Household Data" become active. Clicking on the "Clustering Details" triggers the tab with the log displayed below.

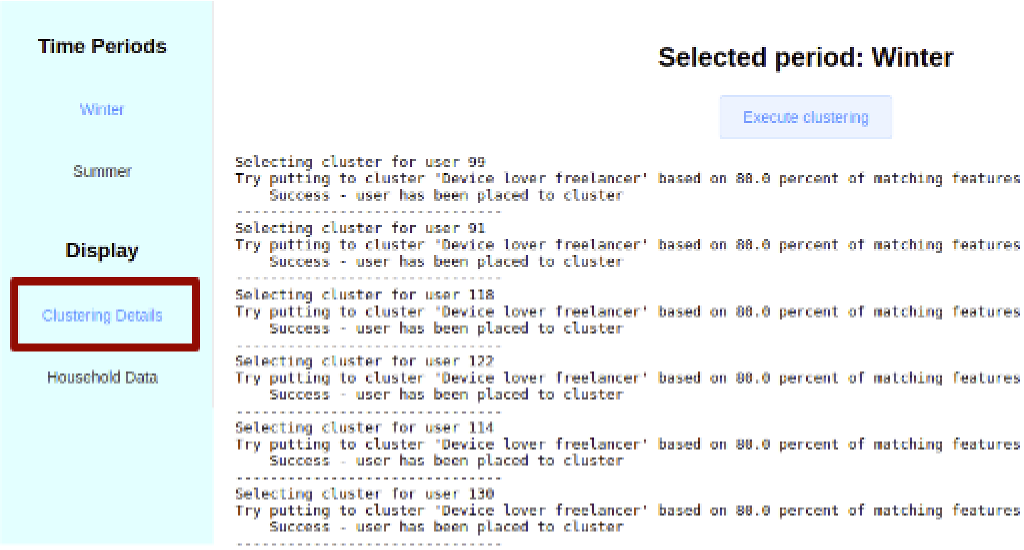

Figure 28: Clusterinf details in the demonstrator

Clicking on the "Household Data" changes the tab to a view where all households are shown. First, all households are shown. By choosing a recommendation group cluster from the "Filter for cluster" menu, the households only for the chosen cluster are shown. In each row the aggregated features after feature engineering for the recommendation group clustering and the result of the behavioral change starter clustering are shown. To view all features, the slider at the bottom of the page can be used.

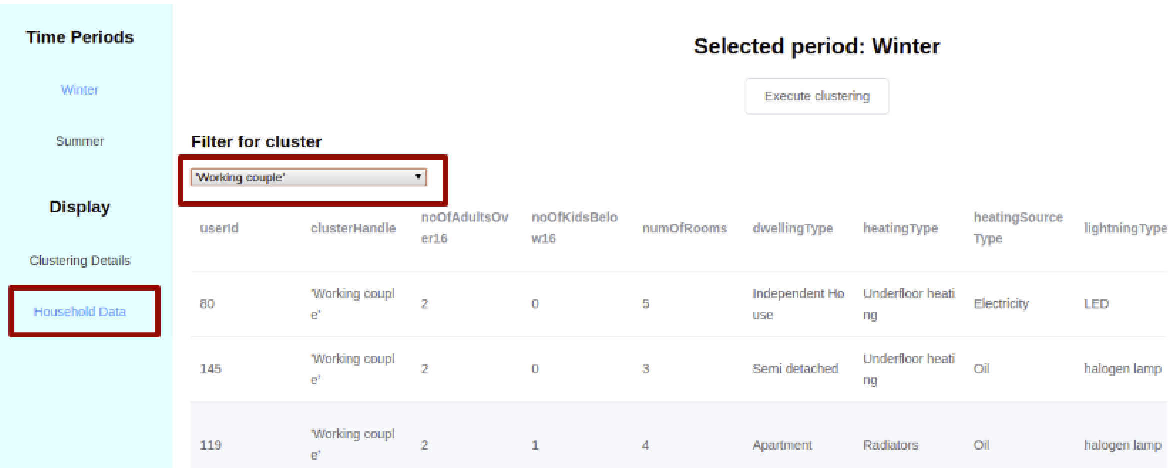

Figure 29: Filter for clusters in the demonstrator

By clicking on a row in the "Household Data" tab, a pop-up window appears.

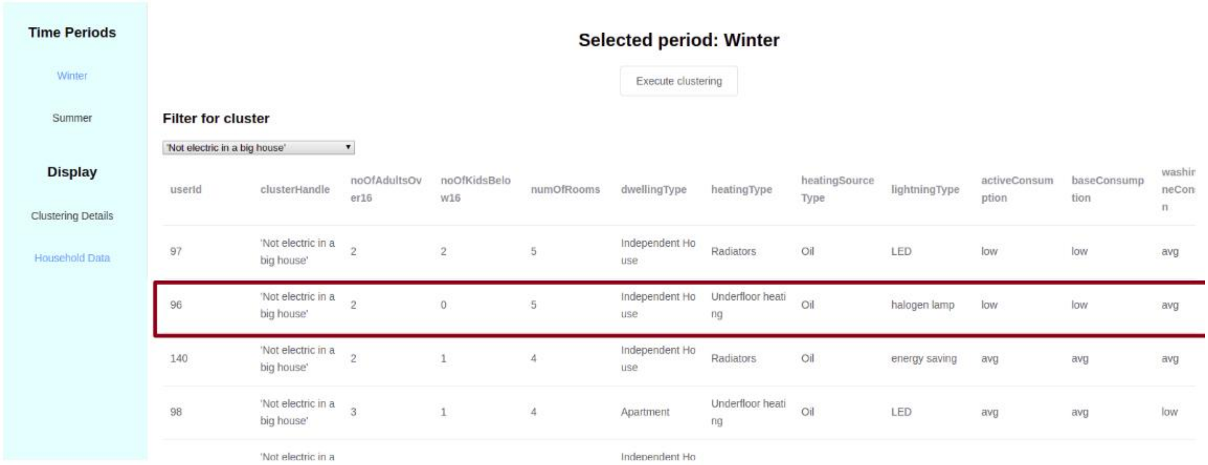

Figure 30: Hosusehold data in the demonstrator

In this window the recommendations selected by the RE for the given user can be seen in a table. Next to the title and the description of the recommendation, the personalized notification time and the type of the message is also shown. To the right, the logging of recommendation selection process is shown.

Figure 31: Pop-up household data in the demonstrator

By clicking outside the pop-up window, the app changes back to the "Household Data" tab and a new user, a new cluster or a new period can be selected.

7 REFERENCES

Ajzen, I. (1991), The theory of planned behavior, Organizational behavior and human decision processes, 50(2), pp. 179-211.

Fogg, B.J. (2009), A behavior model for persuasive design. Proceedings of the 4th international Conference on Persuasive Technology, ACM.

Keogh, E., & Ratanamahatana, C. A. (2005). Exact indexing of dynamic time warping. Knowledge and information systems, 7(3), 358-386.

Koroleva, K., Mehlenhorst, M., Novak, J., Herrera Gonzalez, S.L., Fraternali, P., Emilio-Rizzoli, A. (nd), Designing a holistic socio-technical behavior change systems for energy saving, Energy Informatics, Accepted for publication.

Steg, L., Lieke D., Wokje A. (2005), Factors influencing the acceptability of energy policies: A test of VBN theory, Journal of environmental psychology 25(4), pp. 415-425.

Thøgersen, J., Grønhøj, A. (2010), Electricity saving in households – a social cognitive approach, Energy Policy, 38(12), pp. 7732–7743, https://doi.org/10.1016/j.enpol.2010.08.025.

Tsiporkova, Elena : Dynamic Time Warping Algorithm [PowerPoint slides]. Retrieved from http://www.mathcs.emory.edu/~lxiong/cs730_s13/share/slides/searching_sigkdd2012_DTW.pdf?source=post_page

8 APPENDIX

Table 6: Characteristics of the clusters in the German pilot

Cluster	No. of users	Share of the pilot	Metric	Ascription of responsibility	Personal norm	Self-efficacy	Intention to save energy
Pre-contemplation-lower	8	8.89 %	Median:	2.500	4.100	3.000	4.333
			Mean:	2.656	3.775	3.167	3.875
Pre-contemplation-higher	27	30.00 %	Median:	3.500	5.200	5.333	5.000
			Mean:	3.444	4.956	5.160	4.852
Contemplation	31	34.44 %	Median:	4.000	5.200	4.333	4.333
			Mean:	4.097	5.195	4.043	4.559
Action	24	26.67 %	Median:	4.125	6.200	5.333	6.000
			Mean:	4.021	6.075	5.472	6.125
German pilot overall	*90*	*100.00 %*	Median:	*4.000*	*5.400*	*4.667*	*5.000*
			Mean:	*3.753*	*5.232*	*4.681*	*5.004*

Table 7: Characteristics of the clusters in the Greek pilot

Cluster	No. of users	Share of the pilot	Metric	Ascription of responsibility	Personal norm	Self-efficacy	Intention to save energy
Pre-contemplation-lower	22	20.37 %	Median:	3.000	3.400	3.333	4.000
			Mean:	2.773	3.445	3.394	3.561
Pre-contemplation-higher	42	38.89 %	Median:	3.750	4.500	5.000	5.000
			Mean:	3.661	4.519	5.040	4.810
Contemplation	18	16.67 %	Median:	4.250	5.500	4.000	5.333
			Mean:	4.194	5.556	3.759	5.370

Action	26	24.07 %	Median: Mean:	4.500 4.462	6.000 6.269	5.667 5.474	6.000 6.128
Greek pilot overall	*108*	*100.00 %*	Median: Mean:	*3.875* *3.762*	*4.900* *4.894*	*4.667* *4.596*	*5.000* *4.966*

Table 8: Characteristics of the clusters in the Swiss pilot

Cluster	***No. of users***	***Share of the pilot***	***Metric***	***Ascription of responsibility***	***Personal norm***	***Self-efficacy***	***Intention to save energy***
Pre-contemplation-lower	*2*	*3.33 %*	*Median:* *Mean:*	*1.000* *1.000*	*2.400* *2.400*	*4.667* *4.667*	*7.000* *7.000*
Pre-contemplation-higher	*20*	*33.33 %*	*Median:* *Mean:*	*3.500* *3.123*	*5.200* *5.260*	*3.667* *3.667*	*5.000* *4.817*
Contemplation	*29*	*48.33 %*	*Median:* *Mean:*	*4.000* *4.000*	*5.000* *4.931*	*4.667* *4.897*	*5.000* *4.920*
Action	*9*	*15.00 %*	*Median:* *Mean:*	*4.500* *4.611*	*6.600* *6.622*	*5.000* *5.370*	*6.333* *6.333*
Swiss pilot overall	*60*	*100.00 %*	*Median:* *Mean:*	*4.000* *3.729*	*5.200* *5.210*	*4.667* *4.550*	*5.000* *5.167*