Situation awareness of smart homes based on smart appliances energy consumption

Nikola Sekulovski[1], **Andrej Košir**[1]

[1]*Faculty of Electrical Engineering, University of Ljubljana*
Email: andrej.kosir@fe.uni-lj.si

Abstract

In this paper, we propose a scheme to predict the presence of individuals in the household based on energy consumption data provided by smart appliances. This situation-awareness task is relevant for several smart home services. We evaluated the Support vector machine (SVM) model and Hidden Markov Model (HMM) for time series where the latter achieved better classification performance.

1 Introduction

Smart homes have emerged as a transformative technological paradigm, revolutionizing the way we interact with our living spaces and redefining the concept of modern living. The number of smart homes in Europe, as well as globally is relatively slowly but steadily growing. According to Berg Insight [1], the number of smart homes in Europe in 2022 was around 63 million and it is projected to reach 112 million five years later. Meanwhile, the global number of smart homes is projected to reach around 670 million by 2027, according to Statista [6]. Thus, no doubt exists about their future presence and potential.

One of the first definitions of situation-awareness was given by [2] as *the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future.* Note that situation awareness does not involve only recognition of selected situations but also its relation to a given service, i.e. the relevance of a given event must also be established.

Situation awareness in smart homes further enhances this transformative experience by enabling the system to dynamically respond to changing conditions, ensuring optimal functionality and safety for occupants at all times. It is applicable to advanced smart home dedicated services such as elderly support, children at home alone etc.

This paper utilizes energy consumption data taken from smart home smart appliances with the goal of modeling the presence/absence of people in the household with respect to time by utilizing machine learning algorithms. At this stage of research, we limited to Microwave only to estimate the potential of the proposed approach.

2 Smart home and situation awareness

2.1 Smart homes

The term ”smart home”, also known as home automation or smart house, refers to a residential property equipped with advanced technology to control, automate, and monitor a number of equipment and systems. These technologies are designed to enhance the comfort, convenience, energy efficiency, security, and overall well-being of the occupants. Smart homes utilize Internet of Things (IoT) devices, sensors, and artificial intelligence to create a networked environment where devices can interact and respond intelligently to users' needs and preferences.

2.2 Situation awareness of smart systems

Situation awareness is the perception and understanding of the current environment and the events unfolding within it. It involves being aware of relevant information, recognizing patterns, and comprehending the implications of various factors in a given situation. In the context of smart homes, situation awareness refers to the ability of the home automation system to monitor and comprehend the current environment and activities within the house. It involves collecting data from various sensors and devices deployed throughout the home, analyzing that data, and understanding the context to make informed decisions and provide appropriate responses.

3 Classification using Hidden Markov chains and Support vector machine

In this section, we briefly introduce a classification of time series using Hidden Markov models and Support vector machines.

3.1 Hidden Markov models

Hidden Markov Models (HMMs) are probabilistic models used to describe a system that evolves over time and produces a sequence of observable outcomes. The key idea behind HMMs is that the underlying system's state is not directly observable, hence the term ”hidden”. Instead, we can only observe the outcomes or emissions associated with each state [8].

In the case of a smart home we are presenting here, observable states are derived from the energy consumption of smart home appliances, and hidden states are situation awareness-related states like being present or absent. We select HMM approach since smart home data is noisy by its nature and probabilistic models may handle this randomness more effectively than classic classification approaches.

Given a time series dataset, the HMM can solve one of the following three types of problems [4]:

- Given a set of observations and the 3 model parameters (initial hidden state, emission, and transmission probabilities) calculate the occurrence probability of the observations. This type of problem is solved via the forward-backward algorithm.
- Given a set of observations and the 3 model parameters, determine the optimal set of hidden states that result in observable states. This type of problem is solved via the Viterbi algorithm.
- Given only a set of observations, determine the optimal set of model parameters. This type of problem commonly uses the Baum–Welch algorithm and it is also the type of problem we aim to solve in this paper.

In this research, we applied the first alinea.

3.2 Classification of contextual state

We pose a problem of contextual state recognition as a classification problem in machine learning.

Support Vector Machine (SVM) is a powerful and widely used supervised machine learning algorithm for classification and regression tasks. In the context of classification, SVM is primarily used for binary classification, where the goal is to separate data points into two classes based on their features.

In this paper, we use SVM as a baseline algorithm and we compare it to the classification performance of the Hidden Markov model.

4 Experimental results

In this section, we present experimental results of classification as a solution to contextual state recognition. We compare the performance of the SVM classifier to the Hidden Markov Model approach.

4.1 Test data

The data set used for this paper was the smart home data set from Kaggle [7]. The data set consists of time series of energy consumption of multiple household appliances as well as entire rooms, for instance, the electricity use of the microwave, fridge, kitchen, living room, etc.

Before we construct a training set for classification, we need to perform a few preprocessing steps:

- Firstly, we replace the indexes of our data frame with dates starting from the first of January at 5 o'clock in the morning. This date has no particular relevance as it was chosen randomly. The data frame has a sampling rate of one minute.

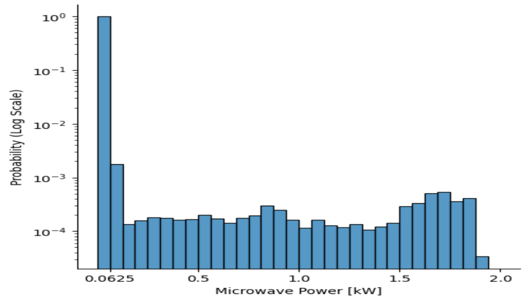

Figure 1: Log scale distribution plot of microwave operation

- The second step is to choose which features we will use for our models. We decided to use the functioning of the microwave and a default feature that will be discussed in a moment. The reasoning behind choosing the microwave was its frequent use in the typical household as well as its operational nature.
- As mentioned in the previous step, we constructed a new feature, named the default contextual status indicating the most likely state regarding being present at home. The default feature has a periodic nature, whose values are determined based on sound reasoning and rationality. In our case, we assume that between the hours of 00:00 and 06:00 and between 18:00 and 00:00 someone is present at home, while for the rest, the home is empty.
- If we graph the presence/absence in the home where the presence is denoted by 1 and absence by 0 with respect to time, we can observe spikes at instances where the value of the functioning microwave jumps above the threshold and immediately falls below the threshold. Since it is not rational for someone to be one minute at home and then immediately out of it, we removed these spikes.
- Lastly, we built a function that would set the ground truth values. For contextual purposes let us describe the routine of the person in question. This person sleeps between the hours of 00:00 and 06:00, then goes for a run or to the gym between the hours of 06:00 and 08:00. The hours between 08:00 and 10:00 are used for breakfast and getting ready. Afterwards, this person is at work from 10:00 to 18:00. The hours between 18:00 and 23:00 he/she is at home, and finally between 23:00 and 00:00 perhaps this person takes the dog for a brief walk before going to bed. These values were determined purely on a reasonable assumption.

We can observe the input values over a period of one week in Fig. 2. The input into our models (observable states) can have one of four values (00, 01, 10, 11), where 00 means that both, the microwave and default features imply that the home is empty, 01 means that the microwave implies the absence while the default implies presence. The same applies to the remaining values.

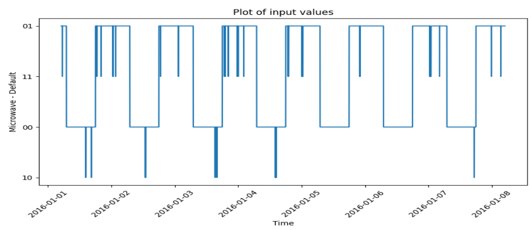

Figure 2: Observable values of the observed smart home

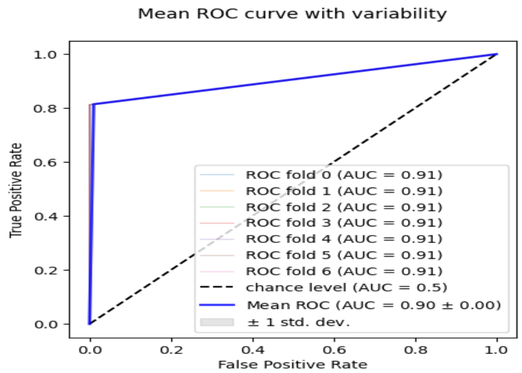

Figure 3: Same day context recognition - SVM

4.2 Same day context recognition

In this subsection, we evaluate the performance of the context classification on the same day of the week.

The cross-validation methodology in this subsection is the following: both the training and testing set contain data from a 24-hour interval, however, the testing set is shifted by a time period of 7 days. Since each fold represents a 24-hour interval, as the algorithm progresses into the next fold, that 24-hour interval for the training and testing set is shifted by 24 hours. By the end of the seventh fold, the algorithm has traversed over an entire week, training one day and testing on the same day, the following week.

4.2.1 Support vector machine model

As a baseline model, we applied a SVM classifier to classify contextual states. It is of great importance to test multiple models and compare the results. For this reason, we chose to implement a support vector machine model with a linear kernel as it is an easy model in terms of implementation and it provides us with the perspective of how good or bad the evaluation results could be. Figure 3 represents the receiver operating characteristic for each fold.

4.2.2 Hidden Markov model

In order to acquire a better understanding of the problem, we first constructed the model, shown in Fig. 4. In this Figure, the hidden states are represented as circles along with their denominations, while the observable states are given as squares along with their values (see subsection 4.1 for further details about the naming scheme). Transmission probabilities are denoted with

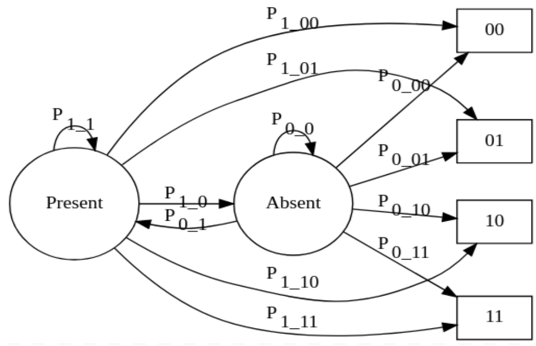

Figure 4: Hidden Markov model assumed in the presented smart home modeling

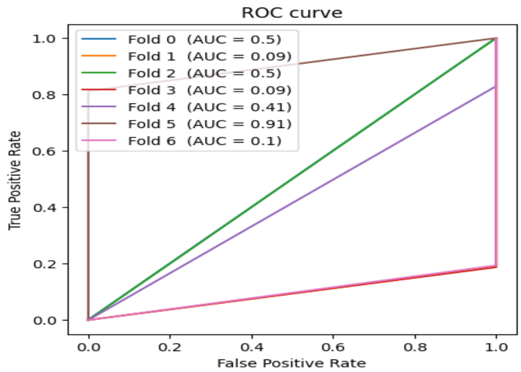

Figure 5: Same day context recognition - HMM

the $P_{HiddenState\_NextHiddenState}$, emission probabilities are denoted with the $P_{HiddenState\_ObservableState}$.

The model was constructed using the HMM Python library [3]. The parameters were set as follows: the number of components is the number of hidden states i.e. 2 (absent and present), the number of trials was set to 1, the number of features is the number of observable states i.e. 4 and we set the optional parameters ”init_params” and ”params” to the value ”ste”. This initializes the starting, transmission, and emission probabilities based on the initial distribution determined by the model and updates their values during training (hence the value ”ste”).

4.3 Beetween days context recognition

The cross-validation methodology in this subsection is almost identical to that in subsection 4.2. The difference lies in that, the testing set is not shifted by a time period of 7 days, but by a time period of anywhere between 1 and 6 days or between 8 and 13 days. What this allows us is to test the models on a different day either the same or the following week. The importance of this kind of testing is to determine whether there is some similarity or periodicity between the data of different 24 hour periods i.e. days which would yield a different result than in the previous subsection.

4.3.1 Support vector machine model

We receive the same result even if we test different days the following week which implies that there is some sea-

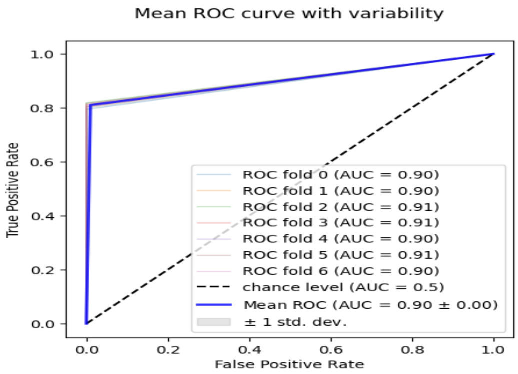

Figure 6: Random cross validation - SVM

sonality present between the days that causes our algorithm to be unable to differentiate. From these, we conclude the working day specific (from Monday to Friday) is not very important in this context recognition task.

4.3.2 Hidden Markov model

Similar to the within-day evaluation, we receive similar results as with the Hidden Markov model in the previous subsection (see figure 5). As such we draw the same conclusion, that is the specifics of a single day do not play a significant role in this context recognition task.

4.4 All-days context recognition

As an alternative to grouping the data in intervals of 24 hours, we decided to also try an all-day approach where we use train and test data regardless of the day. First of all, the training set represents 70% of data within a one-week time period and the testing set represents the other 30%. Secondly, the cross-validation methodology in this subsection is either on a random basis, implementing the shuffle split validator [5] or by preserving the percentage of samples for each class, implementing the Stratified K-Folds cross-validator [5].

4.4.1 Support vector machine model

The support vector machine model produces excellent results (see figure 6 by both methods of cross-validation. This is quite a positive result after the inconsistency of the hidden Markov model, described in the previous subsection (see subsection 4.4.2).

4.4.2 Hidden Markov model

The models' parameters were set as follows: the number of components is the number of hidden states i.e. 2, the number of trials was set to 1, the number of features is the number of observable states i.e. 4 and we set the optional parameters "init_params" and "params" to empty strings. We will discuss the values of the starting, transmission, and emission probabilities in a moment. The approach discussed in subsection 4.2.2 with the parameters "init_params" and "params" set to "ste" gave us the same results as in figure 5. Manually setting the values of the starting, transmission, and emission probabilities gave us stable and consistent results between folds, however satisfactory results were achieved (AUC values are practically the same as at 3).

5 Discussion

As shown by experimental results, the baseline model (SVM) and the HMM model produce similar results. The working day specifics do not play a significant role. Observe also that the classification of the first context situation (absent) yields perfect classification while all errors are encounters classifying the second context situation (present). As a baseline, a majority classifier was also considered in all settings, and its performance was always $AUC = 0.5$.

It is quite clear from our findings that there is no difference whether we are testing on the same day as we had trained or on a different one (see subsections 4.2 and 4.3) irrespective of the model. Those two cross-validation methodologies yielded the same results, the HMM was chaotic whose auc values changed every time the code was run, while the SVM was overfitting our data.

The cross-validation methodology discussed in subsection 4.4 yielded different results. The HMM was not as chaotic since AUC values were quite consistent between folds but were dependent on the initial values of the model's parameters. On the other hand, the SVM displayed satisfactory and consistent results.

References

[1] Berg Insight. Smart home - europe, 2023. accessed: 20.07.2023.

[2] Mica R. Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors*, 37(1):32–64, 1995.

[3] Lukas Lopatovsky. Python hidden markov model library. `https://pypi.org/project/hmms/`. accessed: 20.07.2023.

[4] Y. Natsume. Hidden markov models with python. `\smallhttps://medium.com/@natsunoyuki/hidden-markov-models-with-python-\ \c026f778dfa7`. accessed: 20.07.2023.

[5] Opensource community. Python library scikit-learn. `\smallhttps://scikit-learn.org/stable/modules/generated/sklearn.model_selection.ShuffleSplit.html`, accessed: 20.07.2023.

[6] Statista. Number of users of smart homes worldwide from 2018 to 2027. `\smallhttps://www.statista.com/forecasts/887613/number-of-smart-homes-in-the-smart\ -home-market-in-the-world`, 2023. accessed: 20.07.2023.

[7] The Kaggle team. Kaggle datasets. `https://www.kaggle.com/datasets/taranvee/smart-home-dataset-with-\ weather-information?resource=download`. accessed: 20.07.2023.

[8] Ingmar Visser. Seven things to remember about hidden markov models: A tutorial on markovian models for time series. *Journal of Mathematical Psychology - J MATH PSYCHOL*, 55:403–415, 12 2011.