Building occupation modelling using motion sensor data

Nils-Olav Skeie[1] Jørund Martinsen[2]

[1,2]Department of Electrical Engineering, Information Technology and Cybernetics
University of South-Eastern Norway, Porsgrunn, Norway, Nils-Olav.Skeie@usn.no

Abstract

In smart building environments, both office and residential buildings, it is important to have some information about the use and occupation. Today this is normally solved by a fixed time schedule meaning the occupants must adapt to the system, not the other way around. This paper discuss the usage of a top hat probability models, based on a four weeks history from inexpensive sensor devices, for prediction of the occupation in the next week. The model was divided into seven groups, one group for each of day of the week. A software system, based on several modules, was developed. One module was used to record the information from the motion sensors and stored the data as historical data. One module was used to create the model, and another module was used to prediction of occupation for the next days, up to a week. The models are working satisfactory as long as the behavior patterns are similar for the training and prediction period. However, the models are sensitive to changes in the daily behavior pattern of the occupants, like holidays or taking a day off.

Keywords: probability model, building occupation, PIR sensor devices, motion sensor devices, building occupation prediction.

1 Introduction

1.1 Background

Important aspects in SMART building environments are energy efficiency, energy savings, and welfare assistance. More than 50% of the energy used in buildings in norther countries is for space heating (Perera *et al.*, 2014). The temperature should be at a comfort temperature only when the building is in use, any energy saving strategies should be in focus when the building is not in use or when the occupiers are sleeping. In most building energy management systems (BEMS) today this schedule is configured with fixed time intervals when to keep the comfort temperature, and when to save energy. Regarding many welfare assistance systems, these systems also need to know when the occupiers are using the building.

Such smart building systems should adapt to the occupiers use of the building, and should not be based only on a fixed time schedule. A good solution can be to start with a fixed time schedule but allow for deviations based on the real usage of the building. Any system should adapt to how the buildings are used by the occupiers, the occupiers should not need to adapt to a fixed configuration.

Today many buildings already have an alarm system based on motion sensor devices in many rooms. These alarm systems are activated only when turned on by the user, however the system, including the sensor devices, are working also when the alarm system is deactivated. Based on the information from these sensor devices it should be possible to predict when the building is in use, and also when the occupiers are sleeping. The prediction of the building occupation can be used to optimize the time when having the comfort temperature, when to save energy, and when to activate any welfare assistance systems. This prediction can even be used to turn on the alarm system.

The aim of this paper is to show the development of a set of models to predict a building occupation based on information from low-cost motion sensor devices. The models should be easy to implement in most programming languages.

1.2 Previous work

Occupancy behavior in buildings are becoming an important topic as building systems are becoming more sophisticated and people are spending a lot of time in the buildings. The occupancy behavior is one of the leading influences of energy consumption in buildings but not used that much as input in existing models (Yan *et al.*, 2015; Clevenger and Haymaker, 2006; Adamopoulou *et al.*, 2016). Previous work is based mostly upon occupant surveys and interviews affecting the energy efficiency of buildings (Yan *et al.*, 2015) not that much work based on the occupier use of the buildings. An extended work based on occupancy behavior is described in (Adamopoulou *et al.*, 2016) which is dividing the building into zones and using a Monte Carlo approach to model the usage of each zone, including the number of occupants in each zone. Zones are collected in zone groups based mainly on periods of use. The models take into account also the seasons and day of week, making separate models for each instance of these zone groups. The historical data is based mainly on image processing from several depth-image cameras, but also on acoustic and infra-red (PIR) sensors. The work described in (Ryu and Hyeun, 2016) is based on

DOI: 10.3384/ecp2017043

decision tree as machine learning technique and hidden Markov model as a probability technique. The historical data is based on carbon dioxide (CO_2) concentration and electricity consumptions. The work described in (Wang *et al.*, 2018) is based on the k-nearest neighbors (knn), support vector machine (svm) and artificial neural network (ann) machine learning algorithms. The historical data is based on the fusion of environmental data and information from the WiFi network. In this work the ann based model gave the best result. The work described in (Habib and Zucker, 2017) is based on the indoor air quality information from the ventilation system and a k-means clustering algorithm for occupancy prediction in a building. The k-means clustering is a machine learning technique. The work described in (Shi and Yao, 2016) is using a novel statistical model, based on a logistic regression model, for occupancy prediction at a specific time. The model is only simulating based on time of the day. The focus in this work is a model predictive model (MPC) for an efficient operation of a heating, ventilation and air condition (HVAC) system. The work described in the master thesis (Martinsen, 2019) is based on information from infra-red (PIR) sensors and a simple probabilistic model for each day of the week, and is discussed in this paper. The model can easily be implemented in most software languages.

1.3 Outline of the paper

Section 2 provides a discussion of the movement sensor device, data collection and the building used. Section 3 gives an overview of the model used. Section 4 gives an overview of the model fitting and validation. The results are discussed in section 5, and some conclusions are drawn in Section 6.

2 System description

A building located in South Eastern of Norway has installed a set of movement sensors and the information from these movement sensors have been logged for at least one year. The data is logged on comma separated values (csv) based text files, one file for each month.

The building has three floors, ground floor, first floor and second floor. The sensors are located in the living room and kitchen at the first floor, and top of the staircase and in an extra living room at second floor. Two movement sensors are installed in the living room at first floor.

An overview of the first floor and second floor, with the sensor locations marked with green rectangles, shown in Figure 1.

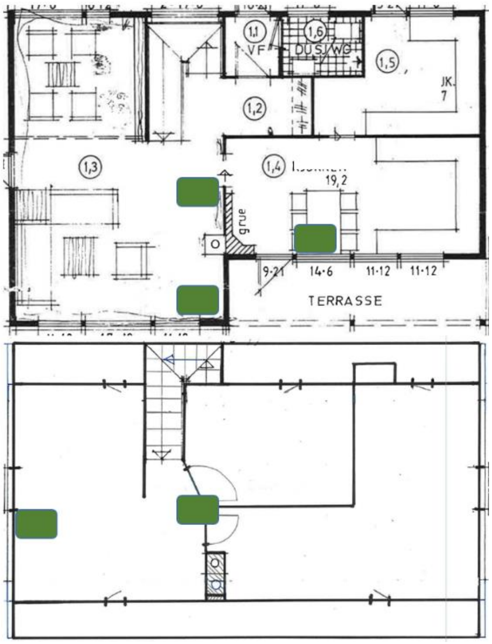


Figure 1. The first and second floor of the building with the motion sensor devices as green rectangles. First floor, at the top, with three sensor devices, one in the kitchen and two in the living room. Second floor, at the bottom, with two sensor devices, one at the staircase and one in the living room.

These motion sensor devices are based on the passive infrared (PIR) type of sensors, detecting if an object with a higher temperature than the environment is moving into the measurement area of the sensor device. The measurement principle is based on heat radiation from an object with a higher temperature than the environment. The sensor converts any resulting change in the incoming infrared radiation into triggering an event giving a digital output voltage pulse. A Fresnel lens is used to divide the measurement area of the sensor device into sectors, so any change in the infrared radiation within any of the sectors will trigger an event. The output signal from these sensor devices will be a digital pulse signal, an event, indicating that a moving object has been detected. The sensor devices are connected to the digital input ports on a data acquisition (DAQ) device.

Figure 2 shows the number events (triggers) from two of the sensor devices on the first floor, for a typical weekend day. The red indication for the sensor on the kitchen on the first floor, and the blue indication for the sensor in the living room, close to the window. The number of triggers is in the range of 200 to 800 for each hour, for the sensor device in the kitchen.

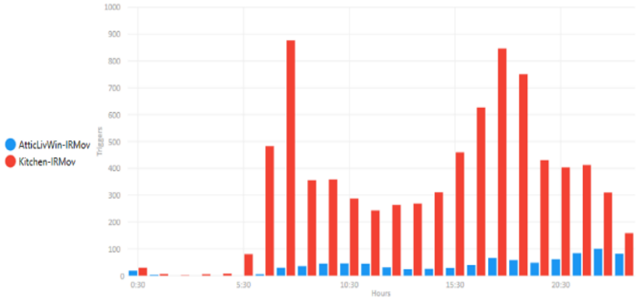


Figure 2. The number of events (triggers) from two of the sensor devices on the first floor shown for a weekend day for 24 hours. Starting at 00:00 and ending at 23:59, one red and blue column for each hour.

The software consists of four main functions, shown in Figure 3. The figure is a use case diagram (Fowler and Scott, 1997) drawn using the Unified Modeling Language (UML) diagram. A use case diagram shows the main functionality of the software together with the input/output device or external modules known as actors. The functionality of software are 1) collect the data from the motion sensor devices and save the time stamped data on a comma separated values (csv) text file. The motion sensor devices are connected to the DAQ system. 2) create the models based on the data from the csv file and configuration data. The operator or a specific time event can start the create operation for the model. 3) perform prediction based on a specific time. 4) handle configuration of the system, by the operator. The configuration data are saved on an xml type text file loaded every time the software is starting.

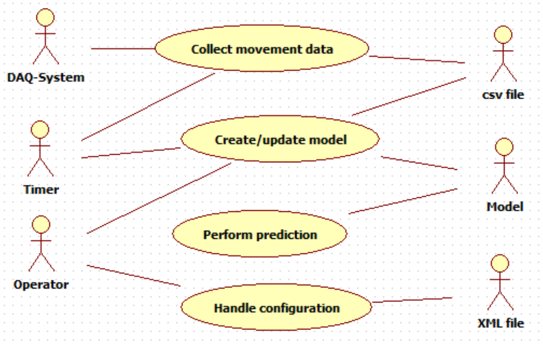


Figure 3. A use case diagram of the main functionality and the actors of the software.

The “Perform prediction” function will use the model to predict the occupation sometime in the future. This is a manual operation in the current software version but should have a type of application programming interface (API) to let other software applications like the BEMS communicating with this software. Based on the model the prediction can be one week ahead.

3 Model development

3.1 Model selection

The basis is normally a fixed time configuration when installing systems like BEMS in buildings. This paper will propose an extension to such system by adding a model for predicting when the building is actually in use. There is a need to differentiate between an office building and a residential building where the main difference is the use during the night. An office building will typically be used during the working days and may be on Saturdays, while a residential building will be the opposite. The workers will normally be either at work or at home. The challenge with a residential building is the use at night without any measurements of use.

The focus for this paper is residential buildings assuming that the buildings are in use in the morning, in the afternoon, and during night. As a starting point, it is assumed that the buildings will be empty during the daily working hours, and used all day during the week-ends.

A data driven approach is the basis for this model development, based on historical data from the building. Based on the historical data a mathematical model predicting how the building is going to be used for the next days is wanted. In this case more focus on the behavior of the occupations is wanted, and allow for variation of this behavior. Based on this assumption a probability function model is chosen over a model based on a machine learning approach (Bzdok *et al.*, 2018). A machine learning approach, like the artificial neural network (ann) methods, requires a large amount of data. As the occupancy prediction will depend on the seasons, as shown in (Adamopoulou *et al.*, 2016), giving less data, a probability approach is chosen. An autonomous approach for model update is also wanted for updating the occupancy model based on the seasons.

The historical data is based on the information from the PIR sensor devices normally used in any alarm system. The nature of the PIR sensor device is an event, giving an electrical signal with duration of about one second when detecting a movement of a warm object in the area. The probability is one when detecting the movement and declining to zero after a period when not detecting any more movement of warm objects. A simple probability function is wanted and the triangle function, see Figure 4, is the first probability function to evaluate.

DOI: 10.3384/ecp2017043

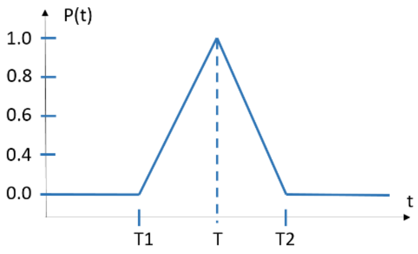


Figure 4. The triangle function.

The triangle function must be linked to every sensor event giving a more complex model because the parameters T1 and T2 must be adapted to each event. Figure 2 shows that there can be a lot of events for every hour when the building is occupied. Sensor devices events for a typical working day is shown in Figure 5, showing the events from the sensor devices at the first floor, with reference to Figure 1.

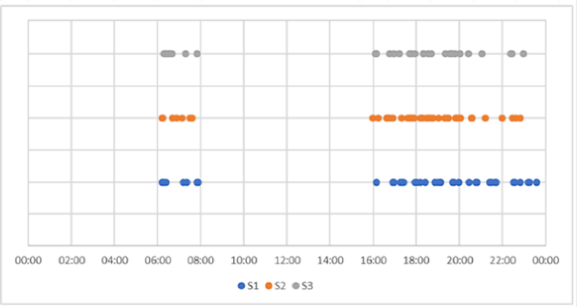


Figure 5: The sensor device events from the three sensors devices at the main floor.

Figure 5 shows that there is high probability that there will be a sequence of events when first detecting an event. The model should be able to predict these sequences of events. Based on a maximum time between these events, the events are grouped into one or several sequences. This maximum time is a parameter that can be configured in the software. The current value is 60 minutes. Figure 6 shows the start and stop events (red dots) to indicate the start and stop of each sequence according to this maximum time, and how these events are converted to the sequences for the use of the building for each day of the week. A morning and evening section for the first five days (working days), and only one section for the last two days, the weekend days.

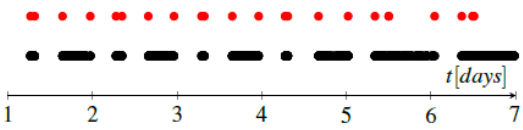


Figure 6. A week with the events (red dots) and the sequence of use of the residential building (black dots/lines).

A probability function that can handle a sequence of events within a period can be a better solution, where the duration of P(t)=1 and the different slopes can be estimated based on the sensor information. A top hat probability function may be a better approach (Boyd, 2006) and is evaluated. Figure 7 shows a simplified top hat probability function.

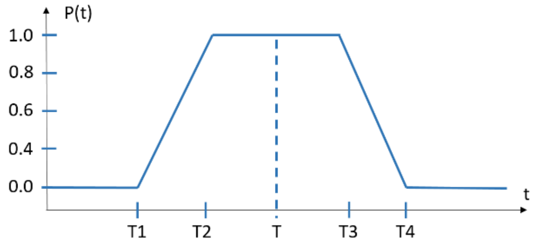


Figure 7. A simplified top hat function, based on (Boyd, 2006).

The simplified top hat function (Boyd, 2006), shown in Figure 7, shows the parameter T for the center of the P(t)=1, the length (T2 to T3) of the P(t) = 1 area, and the length (T1 to T4) of the P(t) > 0 area. These parameters, estimated during the training sequence of the model, are listed in Table 1.

Table 1. Estimated parameters for the simplified top hat function.

Parameter	*Function*
T	Center of top hat
[T2,T3]	The area for the function P(t)=1
[T1,T4]	The area where 0 < P(t) <= 1

The equation for the top hat function is:

$$P(t)=\begin{cases} 0,\ t < T1 \\ (T2\text{-}T1)(t\text{-}T1),\ t \in [T1,T2] \\ 1,\ t \in [T2,T3] \\ (T4\text{-}T3)(t+T3),\ t \in [T3,T4] \\ 0,\ t > T4 \end{cases} \qquad (1)$$

A top hat function can only handle one sequence of sensor device events and as shown in Figure 5 there can be several sequences during a day. Several top hat functions can be combined to handle these numbers of sequences, and make a more complex modelling of the building use. An example of a working day is shown in Figure 8, combining two functions, one in the morning and one in the evening. The regions for each top-hat function must be estimated, and each top-hat function will have its own sets of parameters, with reference to table 1. These regions will be estimated based on the sequence of events from the sensor devices, from the first detection to the last detection in a specific period.

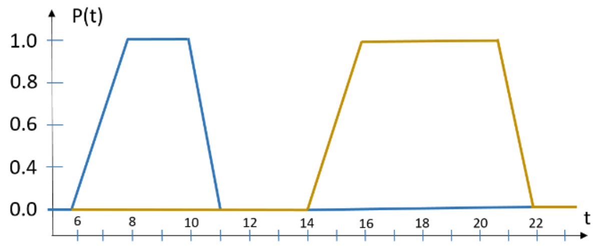


Figure 8. The combined top hat functions for a 24 hours period.

As the residents may have different schedules for every day in a week, but with a high probability that these schedules are repeated every week, giving specific occupancy for each day of the week. Based on this assumption it was decided to split the sensor data into groups, one group for each day of the week.

There can also be deviations for each day of the week so it was decided that a number of events outside a time limit, based on the average, could be removed from the data set. These numbers are defined in the configuration part of the software, the time deviation and number to remove. These events can typically be an event like going to the kitchen for a glass of water or milk during the night, or taking a day off during the working week.

The principles for selecting the candidates for the start and stop events for the model is shown in Figure 9.

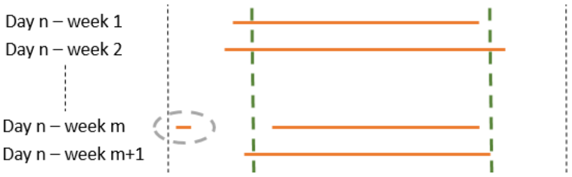


Figure 9. The principles for selecting the start stop events for a sequence.

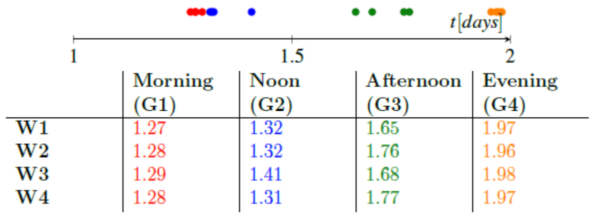
The events for a specific day are grouped together within a period, configured as a maximum time in the software system. Each day will have a group for every week, as shown in orange lines in Figure 9. The green dotted lines are the group average start and stop positions and the start of week m is outside the maximum time limit, configured in the system, and removed from the data set.

For the building shown in Figure 1 a separate set of models are created for each floor.

3.2 Model training

Sensor data was collected for the last year and first all the data was used for training the daily sequences. Due to holidays and variations trough out the year, this was not a good approach. This is also mentioned in (Adamopoulou et al., 2016) as the occupancy behavior will depend on the seasons, and the data sets for model development should be smaller. The new approach is to use the data from the last eight and four weeks for training the model. It was also configured that only one event can be removed from the data set if more than 60 minutes outside the average for the start or end of an event sequence (as shown in Figure 9).

Figure 5 shows two set of event sequences for each of the sensor devices, S1, S2 and S3. The Create/update model function of the software application, see Figure 3, was used to create the model by finding the candidates for the desired periods. Figure 10 shows the top-hat parameter estimation for a specific day, using a data set of four weeks.



	Morning (G1)	Noon (G2)	Afternoon (G3)	Evening (G4)
W1	1.27	1.32	1.65	1.97
W2	1.28	1.32	1.76	1.96
W3	1.29	1.41	1.68	1.98
W4	1.28	1.31	1.77	1.97

Figure 10. Estimation of the model parameters for a data set of four weeks. These group values, for day number 2, are used to define the T parameters according to Table 1.

The T parameters, according to Table 1, for the first top hat functions are selected as shown in Table 2.

Table 2. The group values for the top hat function for day number 2.

Parameter	*Value*	*Description*
T1	1.27	G1 minimum
T2	1.29	G1 maximum
T3	1.31	G2 minimum
T4	1.41	G2 maximum

Figure 11 shows the data flow for training the models, estimating the T parameter for each of the top hat functions. The training starts with selecting a data set from the historical data sets, estimating the average start and stop times for each of the periods for the day groups and top hat functions for the day. Any events outside a configured maximum time will be removed and a new estimation will be performed. In Figure 11 a maximum time of 60 minutes is used, and this maximum time is shown in Figure 9. The number of events that can be removed can also be configured in the software, only one is used for this training. When any events outside the limits are removed, will the minimum and maximum times for each group be assigned the T1, T2, T3 and T4 for each of the top hat functions.

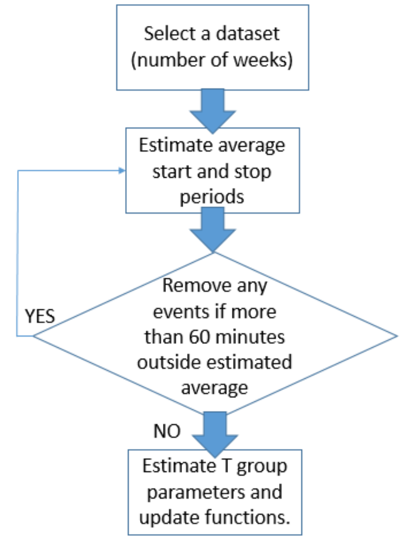


Figure 11. The flow chart for training the models.

A model was created based on a data set from the last eight weeks, and the prediction output from the model is shown in Figure 12. The model is created only for first floor and only the sensor devices on first floor

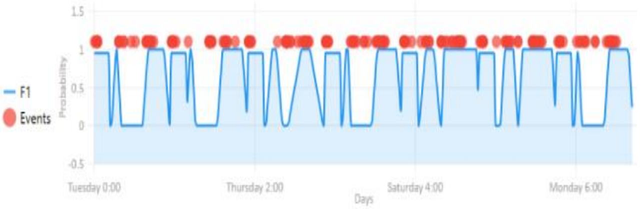


Figure 12. Training the model with eight weeks of data with prediction output for all days. The model starts on Tuesday and ends on Monday.

Another model was created based on the data set from the last four weeks, and the prediction output from that model is shown in Figure 13. The model was created on the same data set as the model shown in Figure 12.

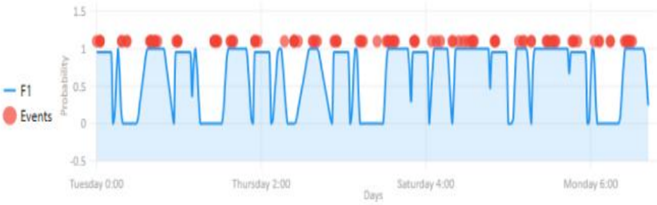


Figure 13. Training of the model with four weeks of data and showing the prediction output for all days of the week. The model starts on Tuesday and ends on Monday.

Comparing the model output from Figure 12 and Figure 13 shows almost the same prediction. The prediction starts on Tuesday, first the morning, no occupation during the day, and occupation during the evening, night and Wednesday morning. The number of events are higher for the eight weeks data set, as shown in Figure 12. However, the same model also shows a higher number of events that must be removed during the training of the model. Figure 13 shows almost the same model prediction, but with fewer events removed. The model in Figure 13 was used for validation. Both figures show that the T1 to T4 parameters are different for each of the top hat functions, indicated by the slopes.

4 Results

The models are trained and validated for one floor only, as an option in software to make a separate model for each floor. The models were trained based on data from both four and eight weeks of data. The models based on eight weeks of data gives a more detailed model but any holidays, days off or deviations during this period introduces more model errors. The best approach seems to use four weeks of data for the training period, and have the option to remove one deviation in any of the weeks.

The model validation is based on three test cases. One normal test case with high quality training data, data without any events that must be removed. The training data set is from 5-JAN to 4-FEB, and the prediction period is 5-FEB to 13-FEB. The next test case is the holiday test case with a training set with many events that must be removed. The training set is from 5-JUN to 4-JUL, and the prediction period is 5-JUL to 13-JUL. The last test case has focus on a single day, the Friday test case with missing data. The training set is from 30-OCT to 29-NOV, and the prediction period is 30-NOV to 7-DEC.

The results for these three test cases are, using four weeks of training data:

- Normal test case; models accuracy are between 94% and 98%,
- Holiday test case; models accuracy are between 32% and 38%,
- Friday test case; models accuracy are between 11% and 17%.

Figure 14 shows the model prediction for the normal test case, the measured data as red indications and the candidates for creating the model as orange indications. The model shows a good match between the measured data and the prediction.

DOI: 10.3384/ecp2017043

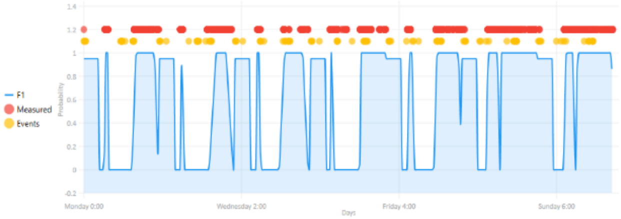


Figure 14. Validation of the normal test case, created the model with a high quality data set. The data starts from Monday and ends on Sunday.

Figure 15 shows the model prediction for the holiday test case where the training set is from the last working weeks before the holiday while the prediction should be for a holiday week. The measurements indicate that the building was not used the first part of the week, and used all the day the last part of the week. This shows that the pattern in the training data must correspond with the pattern for the prediction week.

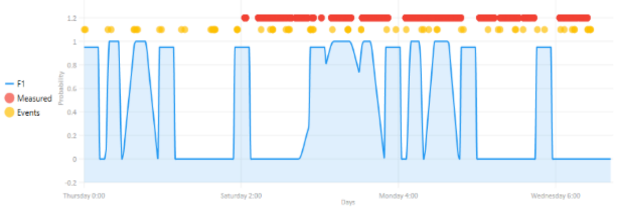


Figure 15. Validation of the holiday test case, created a model with a different behavior pattern then the expected pattern for the prediction week. The data starts from Thursday and end on Wednesday.

The last test case, the day test case (Friday test) shows the same deviation as the holiday test case. The test case is shown in Figure 16. Only the Friday contains measured data while the candidate data has created a prediction model for the whole week, and the measured data is inconsistent with the model prediction. The behavoir seems that the occupants has left the building at Friday and been away the whole next week.

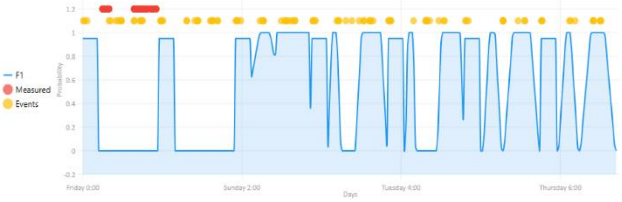


Figure 16. Validation of a day test case where the created model was created based on different behavior pattern then the actual behavior pattern for the prediction week. The data starts from Friday and ends on Thursday.

5 Discussion

The pattern consistency is important to be able to create a good prediction model. This means that the behavior pattern for occupants must be the same for training set as for the prediction period. The first test case, the normal test case, shows that if there is good connection between the behavior pattern for the training set and the behavior pattern for the prediction week, the model will predict with a good accuracy.

The deviation of the occupancy behavior for the training and validation period is difficult to fulfill as human will make random decisions to take a day off, or going away for a holiday.

This indicates that to have a good building occupation model the behavior pattern for the training set must be of the same type as for the prediction week. Some sort of building calendar can be a good extension to the system for defining any deviation from the normal behavior pattern. This way it is possible to extend the model with day models for both working days, holiday staying at building (same as weekends) and holiday staying away from the building. This approach should also be applicable for office buildings during special holidays like Christmas.

In the normal test case the deviation is detection of movement during the night which should be a deviation as the energy system should not adjust the comfort temperature according to these short movements. An option can be to have a minimum period during the night for detection of movement events.

The holiday test case is trained for a normal week. However, in the prediction week the occupants were away half of the week, and stayed at home, all day, the rest of the week. A calendar option could have made this prediction better by defining holiday away and at home states.

The Friday test case is almost the same as the holiday test case. However, in the prediction week the occupants stayed home only on Friday. A calendar option could have made this prediction much better.

The software is designed to create a new model at fixed times so this software will be an autonomous system updating the models every week. This is also a reason for using a probabilistic model that can be easily implemented in software.

The software, as indicated in Figure 3, was implemented using the C# programming language. Most of the figures in this paper is based on screen dumps from the user interface from this C# application.

The system will also work independent of the number of occupants in the building as the number of events does not matter, only the start and stop events of a sequences of events.

A PIR sensor device at the main door can also make the system better in estimating the occupation pattern as events will be created when entering or leaving the building. Such a configuration may also be used for turning the alarm system automatically on.

The software was configured to allow a period of up 60 minutes for a continuous occupancy behavior while the work in (Adamopoulou *et al.*, 2016) used 30 minutes. A shorter period can be a better approach.

6 Conclusion

The paper has shown that the top hat functions can be used for predict the occupancy behavior of a residential building. A training period of four weeks, with the option to remove one extreme event sequence from the motion sensor devices can give a good prediction horizon of one week. The solution depends on the behavior patterns for the occupants, and this pattern has to be the same type during both training period and the prediction period.

References

Anna A. Adamopoulou, Athanasios M. Tryferidis, and Dimitrios K. Tzovaras. A context-aware method for building occupancy prediction. *Energy and Buildings*, 110: 229-244, 2016.

J. P. Boyd. Asymptotic Fourier coefficients for a C bell (smoothed “top-hat”) and the Fourier extension problem. *Journal of Scientific Computing*, 2006.

Danilo Bzdok, Naomi Altman, and Martin Krzywinski. Statistics versus machine learning. *Nature methods*, 15:233-234, 2018.

C. M. Clevenger and J. Haymaker. *The impact of the building occupant on energy modeling simulations*. IEA-EBC Annex 66, 2006.

M. Fowler and K. Scott, K. UML *Distilled: Applying the standard object modeling language*. AddisonWesley, USA, 1997.

U. Habib and G. Zucker. Automatic occupancy prediction using unsupervised learning in buildings data. In *IEEE International Symposium on Industrial Electronics (ISIE)*, 2017. DOI: 10.1109/ISIE.2017.8001463.

Jørund Martinsen. *Modeling of building occupation using motion sensor data*. Master’s Thesis, University of SouthEastern Norway (USN). 2019,

D. W. U. Perera, C. Pfeiffer, and N.-O. Skeie. Modelling the heat dynamics of a residential building unit: Application to Norwegian buildings. *Modeling, Identification and Con trol (MIC)*, 35:43-57, 2014

S. Ryu and J. M. Hyeun. Development of an occupancy prediction model using indoor environmental data based on machine learning techniques. *Building and Environ ment* 107:1-9, 2016. DOI: 10.1016/ j.buildenv.2016.06.039.

J. Shi, N. Yu, and W. Yao. Energy efficient building HVAC control algorithm with real-time occupancy prediction. *Energy Procedia*, 111:267-276, 2016

W. Wang, J. Chen, and T. Hong. Occupancy prediction through machine learning and data fusion of the environmental sensing and Wi-Fi sensing in buildings. *Automation in Construction*, 94:233-243, 2018 DOI: 10.1016/.j.autcon.2018.07.007.

D. Yan, W. O’Brien, T. Hong, X. Feng, H. B. Gunay, F. Tahmasebi, and A. Mahdavi. Occupant behavior modeling for building performance simulation: Current state and future challenges. *Energy and Buildings*, 107:264-278, 2015.