2_C13

Machine Learning for Occupancy Detection through Smart Home Sensor Data

Sundaravelpandian Singaravel **Steven Delrue** **Ivan Pollet** **Steven Vandekerckhove**

ABSTRACT

Data from mechanical extract ventilation units of Renson Ventilation nv installed in Belgium is utilized to detect space occupancy through machine learning. Challenges with the detection of occupancy using data captured by these smart devices are (1) absence of labelled data for training a machine learning model, and (2) occupant's CO_2 generation rate and building layouts influence the measured CO_2 concentrations, which prevents simple rule-based models to be used for data labelling. Therefore, the methodology proposed here to detect occupancy consists of a two-step process. The first step utilizes a gradient-based method to generate occupancy labels for a given time series. In the second step, a neural network algorithm is trained on the labelled time series. Training the neural network on the generated labels is done to remove statistically insignificant mislabelling of the gradient-based method. The method is tested on two different stages. The first stage utilized the data from a single device for which actual occupancy is known. The developed neural network model has a test accuracy of 95% (on actual occupancy labels) or 85% (on generated labels). The second stage utilized data from 35 devices, from which the data from 25 devices are used for training and cross-validating the neural network models. The remaining device data is used for testing the model. The developed neural network model has a test accuracy of 60% (on generated labels). Since the accuracy is estimated on generated labels, which contains few mislabels, it is expected that actual accuracy is higher than 60%. The relatively high test accuracy indicates the potential for transferring a model developed on selected device data to other similar devices.

INTRODUCTION

The connected mechanical extract ventilation (MEV) units of Renson Ventilation nv (hereafter called Renson) have sensors located at extraction points to measure parameters like CO_2 concentration, humidity, volatile organic compounds, and temperature. These measurements allow the MEV to operate based on indoor air quality (IAQ) requirements defined in standards and regulations. Identifying the occurrence of events like space occupancy allows for smarter control that, apart from IAQ requirements, also accounts for non-IAQ objectives like energy efficiency. The paper evaluates the possibility of detecting occupancy through the measurements obtained from Renson MEV units.

Sundaravelpandian Singaravel is a data scientist in the R&D Digital Innovation department of Renson Ventilation nv, Maalbeekstraat 10, Waregem, Belgium. **Steven Delrue** is R&D Manager Data Analytics in the R&D Digital Innovation department of Renson Ventilation nv, Maalbeekstraat 10, Waregem, Belgium; and affiliated with Science, Engineering and Technology Group of KU Leuven Campus Kulak Kortrijk, E. Sabbelaan 53, Kortrijk, Belgium. **Ivan Pollet** is Research Manager in the R&D Digital Innovation department of Renson Ventilation nv, Maalbeekstraat 10, Waregem, Belgium. **Steven Vandekerckhove** is R&D Manager of the R&D Digital Innovation department of Renson Ventilation nv, Maalbeekstraat 10, Waregem, Belgium; and affiliated with Science, Engineering and Technology Group of KU Leuven Campus Kulak Kortrijk, E. Sabbelaan 53, Kortrijk, Belgium.

A simple method to detect occupancy is by estimating gradients and setting thresholds for occupancy (Ansanay-Alex 2013). Calì et al. (2015) pointed out the simplicity of the gradient-based method for occupancy detection, which makes it ideal for large scale occupancy detection. At the same time, however, the gradient method is prone to sudden changes in CO_2 concentration due to window opening and closing or operation of the HVAC system (Calì et al. 2015). Other occupancy detection methods proposed in the literature are using rule-based models, probabilistic models, neural networks, and grey-box models (Chen, Jiang, and Xie 2018). However, the limitations mentioned by Calì et al. (2015) will be observed for any occupancy detection methodology for Renson MEV data, as the data is inclusive of all the effects caused by window or door openings. Removing those effects is impossible as the occurrence of these events is unknown to Renson. Therefore, this paper presents a methodology to extend the gradient-based occupancy detection to meet the needs of smart home data.

Varying factors like floor plan, occupant's age, presence of window grills, etc. influence the amplitude of measured CO_2 concentration. Therefore, for each device or for each group of devices, the thresholds need to be determined. The sheer amount of connected devices makes it impossible to determine thresholds manually. Therefore, a method for occupancy detection that automatically adapts to the context of the device is important. At the same time, the absence of labelled data prevents the direct utilization of machine learning approaches. Therefore, in this paper, the gradient method is extended to an automatic labelling process. The labels are converted into probabilistic estimation through a machine learning algorithm, which is used for further predictions. The probabilistic estimation enables the implementation of an uncertainty based decision-making process over the final predictions on occupancy. Other challenges include the distinction of CO_2 variations due to occupancy on the one hand and internal airflow, on the other hand. Therefore, the underlying assumption for the gradient-based labeller is that any change in CO_2 is due to the presence or absence of an occupant.

In this paper, a gradient-based method for data labelling and a neural network model to detect occupancy are presented. Section 2 describes the methodology for the gradient-based method and the machine learning model for occupancy detection. In sections 3 and 4, the method and results to validate the proposed methodology are presented. Finally, section 5 and 6 discuss and conclude the findings in this study.

MACHINE LEARNING FOR OCCUPANCY DETECTION

Gradient-based segregation for data labelling

For occupancy detection, the time series data can be segregated into two types of states, which are steady state and transition state. Steady state is a condition in which the measured signal is balanced by appropriate ventilation rates. While transition states are conditions in which the measured signal is not in balance with ventilation rates. For instance, a steady state occurs during stable occupancy when the CO_2 generation rate is balanced by equivalent fresh air. Similarly, a transition state occurs when an occupant leaves or enters a space, as the CO_2 generation rate is not balanced by equivalent fresh air. The objective of gradient-based segregation is to separate time series into steady and transition states based on gradients. The segregated regions are associated with appropriate labels for occupancy based on generic rules.

The gradient-based segregation is a two-step process. The first step is model identification for gradient estimation, followed by segregation of gradients into two states (steady and transient). These states are then further clustered to associate different states of occupancy: entering, leaving, occupied, and not occupied. In this section, the different stages of the methodology are further elaborated.

Model identification for gradient estimation. Equation (1) shows the formula to estimate gradients for a function $f(\boldsymbol{x})$:

$$f'(\boldsymbol{x}) = \frac{f(\boldsymbol{x}_t) - f(\boldsymbol{x}_{t-1})}{\boldsymbol{x}_t - \boldsymbol{x}_{t-1}} \tag{1}$$

Gradients show the changes in $f(\boldsymbol{x})$ for small changes in $\boldsymbol{x}$. In this paper, $f(\boldsymbol{x})$ predicts CO_2 concentration at current timestep t based on $\boldsymbol{x}$ = (CO_2 concentration, airflow rate) at previous timestep t-1. Ansanay-Alex (2013) used actual data to estimate gradients; this approach results in undefined values when data points in subsequent timesteps do not change. A simple data model of the form shown in equation (2) is utilized to prevent the occurrence of undefined values:

$$f(\boldsymbol{x}) = \beta + \sum_{i=1}^{N} \alpha_i \cdot x_i^n \tag{2}$$

The simple model is developed through the ordinary least squares regression methodology. Furthermore, estimating gradients through a model opens the possibility to include other relevant environmental signals like noise into the occupancy estimation process. Identifying relevant features is out of the scope of this paper. Equation (3) calculates the gradient for the function shown in equation (2):

$$f'(x_i) = n \cdot \alpha_i \cdot x_i^{n-1}, \qquad n > 1 \tag{3}$$

It has to be noted that power n for $\boldsymbol{x}$ needs to be greater than 1; to ensure gradients that are significantly different can be easily distinguished by a clustering algorithm.

Gradient segregation. Gradient segregation is a hierarchical process. First, gradient segregation ratios are measured using Equation (4), in which gradients are estimated using Equation (3):

$$Gradient\ segregation\ ratio = \frac{f'(CO_2)^2}{f'(Airflow)} \tag{4}$$

Then, the ratios are clustered using the k-means algorithm to automatically determine transient (entering/leaving) and steady states (occupied/not occupied). Finally, generic rules are added to associate a state with occupancy, as shown in Table 1. Information on the transition states and the conditions based on CO_2 reduces the oscillation between all four occupancy states.

Table 1. Generic rules for Occupancy Detection

Occupancy states	Rule
Entering	(Transient state of gradients) AND ($CO_{2,\,t-1} > CO_{2,\,t}$)
Leaving	(Transient state of gradients) AND ($CO_{2,\,t-1} < CO_{2,\,t}$)
Occupied	(Steady state of gradients) AND ($CO_{2,\,t} >$ mean(CO_2))
Not occupied	Steady state of gradients

Neural networks for occupancy detection

Equations (3) and (4) are used to identify if the data is in a steady or transient state of a dataset. The states, along with the generic rules defined in Table 1, make it possible to label a data point in a time series. The result of the data labelling process is a table that contains CO2 at t and t-1, and the airflow rate at timestep t, and corresponding occupancy states. This information is captured by a neural network (NN) model. The NN input layer obtains CO2 at t and t-1, and the airflow rate at timestep t. The output layer of the NN model is a softmax activation, which transforms the binary 0 or 1 per occupancy state into probabilities of the occupancy states. The predicted occupancy state with the highest probability is considered as the final occupancy.

End-to-end representation of occupancy detection method

The scalability of the occupancy detection method is important due to the growing number of connected Healthbox systems. Figure 1 shows the schematic representation of the cloud architecture utilized for occupancy detection. In this architecture, a containerized program, which implements the process mentioned in the above sections extracts historic CO_2 time-series data from every device to develop a user-specific occupancy detection model. The developed models are stored in a server that a user's HealthBox can utilize to infer on occupancy states of each room.

It has to be noted that Renson does not collect information on building type, building layout, room types, number of occupants, and the exact number of rooms in a building. The number of rooms known to Renson is based on the

configuration of the ventilation systems. Therefore, the data labelling method has to be general enough that the resulting occupancy models generalise for users on a large scale.

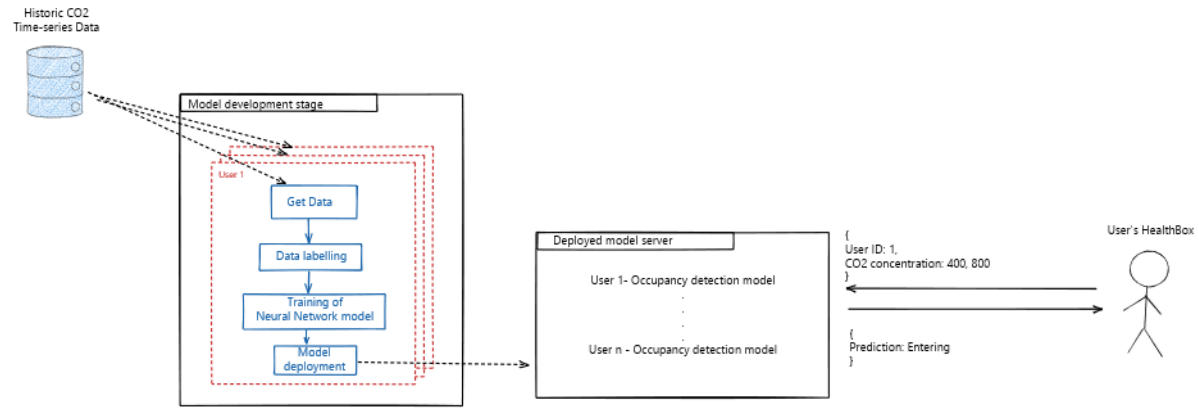

Figure 1 Schematic representation of the occupancy detection method(a).

VALIDATION OF THE OCCUPANCY DETECTION METHOD

The method is evaluated in two stages. The validation starts by evaluating the method to detect occupancy for a single device. Subsequently, the method is validated on a small group of randomly chosen Renson MEV units.

Validation of the method in a single device

The Data collected for training and testing is for a single device. The objective of performing the analysis on a single device is to verify if the method is behaving as intended. CO_2 and airflow rate from a bedroom connected to a Renson MEV unit located in Belgium is collected in two stages. In the first stage, data is collected between January 7 and January 21, 2020. This data is used for data labelling and training of the NN model (*Note: at this point, no information on occupancy is available).* In the second stage, data is collected between January 21 and January 24, 2020, while the user provides feedback on his occupancy. This information is used to test the developed model.

The training data is used to train a NN model using the methodology presented in the previous section. The NN model is developed in Flux.jl package (Innes 2018). The NN model is trained using a Poisson loss function, which measures the difference in the predicted distribution and the expected distribution. The NN model is optimized using the ADAM algorithm.

The developed NN model is utilized to estimate occupancy for the test data. The predicted occupancy is compared with the occupancy provided by the user. Equation (5) shows the formula to determine accuracy:

$$Accuracy = \frac{Total\ number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \times 100 \quad (5)$$

Validation of the method on multiple devices

The objective of this step is to verify (1) the validity of the method on a larger scale, and (2) if models developed with data for groups of devices can be transferred to other devices. Therefore, a small group of devices is randomly chosen for analysis.

CO_2 and airflow rate from bedrooms of 35 randomly chosen Renson MEV units located in Belgium are collected from January 2019. The choice for the number of devices is limited to 35 to reduce the size of the NN model required to train it.

However, the choice data from 2019 is arbitrary.

Out of the 35 devices, data from 15 devices are used to train the NN model, data from 10 devices are used to cross-validate (CV) the model, and the remaining device data is used to test the model. The above segregation in training, CV, and testing device results in 121,776 data points for training the model, 88,944 data points to CV the model, and 98,153 data points to test the model. In this paper, CV data is utilized to manually tune the hidden units with the NN model and regularization parameters. Testing data is an independent dataset that is not utilized in the training process. Since Renson has no occupancy information on a large scale, the accuracies (based on equation 4) are based on the generated labels. Furthermore, visual inspection is used to determine the quality of a prediction.

RESULTS

Occupancy detection – Model development and testing for data from a single device

For occupancy detection, the coefficient n in equation 2.1 is set equal to 2 as it allows for effective segregation of gradients into the two states. Other values of n did not result in effective segregation of the steady and transient states. Equation (6) is the model obtained through an ordinary least squares regression method:

$$CO_{2\,t+1} = 296.6 + 0.00081 \times CO_{2\,t}^{2} - 0.0092 \times airflow\ rate_{t}^{2} \tag{6}$$

This model is used to generate the gradients for segregation. Figure 2 shows the data labels generated for the training data. It can be noted from Figure 2 (middle and bottom) that the segregation ratio during occupied and non-occupied periods are similar. Likewise, transition periods have similar segregation ratios. The appropriate labels associated with the data shown in Figure 2 (bottom) are determined using the generic rules in Table 1. However, it has to be noted from Figure 2 (bottom) that the labels for entering and leaving fluctuate a lot, mainly caused by the quick fluctuations in the data at subsequent timesteps. Data smoothing could reduce fluctuation between entering and leaving, which has not been evaluated in this study.

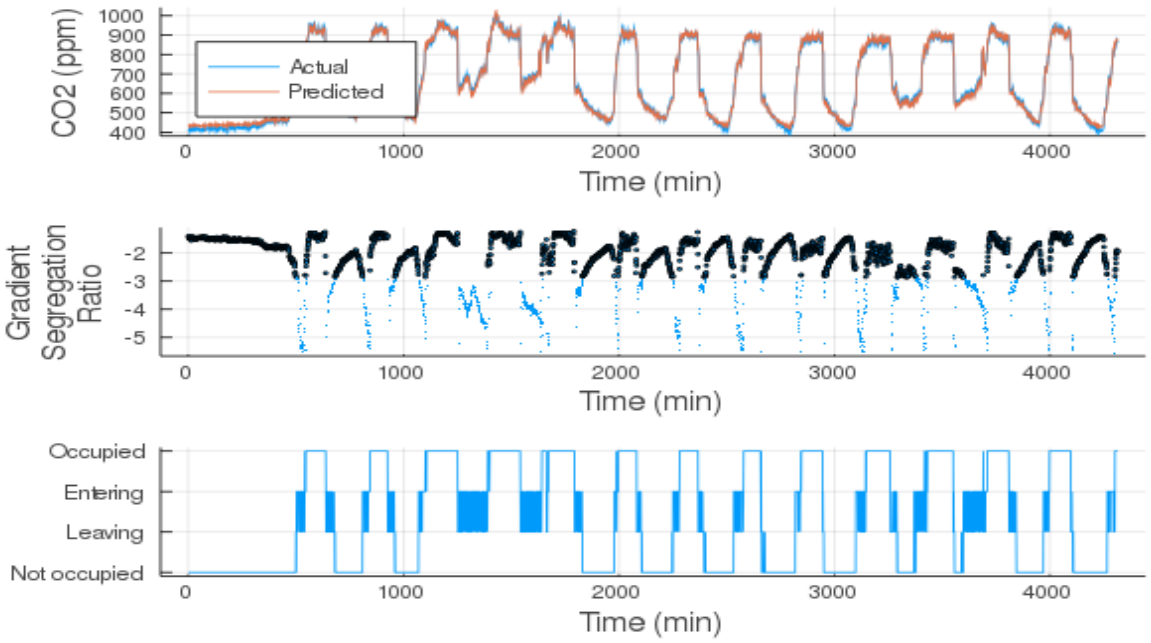

Figure 2 (Top) CO_2 concentration for the training period. (Middle) Gradient segregation ratio for the training period. (Bottom) Derived data labels.

Figure 3 shows the predictions on the test data. Figure 3(middle) shows the softmax output probabilities from the NN model. It can be noted that depending on the CO_2 concentration and airflow rate, the probabilities of different occupancy states vary. The final occupancy (shown in Figure 3(bottom)) is the occupancy state with the highest probability. Periods of room entering are more extended than a pure spike in occupancy state; this is due to the slowly evolving nature of the CO_2 concentration signal. However, room leaving is only for a short period; this is due to a steep reduction in CO_2 concentration. Since the user feedback mentions only a duration[1] of occupancy, entering and leaving states are rounded to occupied and non-occupied while estimating accuracy.

The accuracy based on true labels (i.e., through user feedback) is 95%, and accuracy based on generated labels is 85%. It can be noted that the accuracy estimated through generated labels is lower than accuracy estimated with true labels. The reason for the reduction is due to inaccuracies in generated labels. For example, Figure 2 (bottom) shows that regions of labels for entering and leaving are fluctuating a lot. These fluctuations are removed during the training process, as NN learns statistically significant labels.

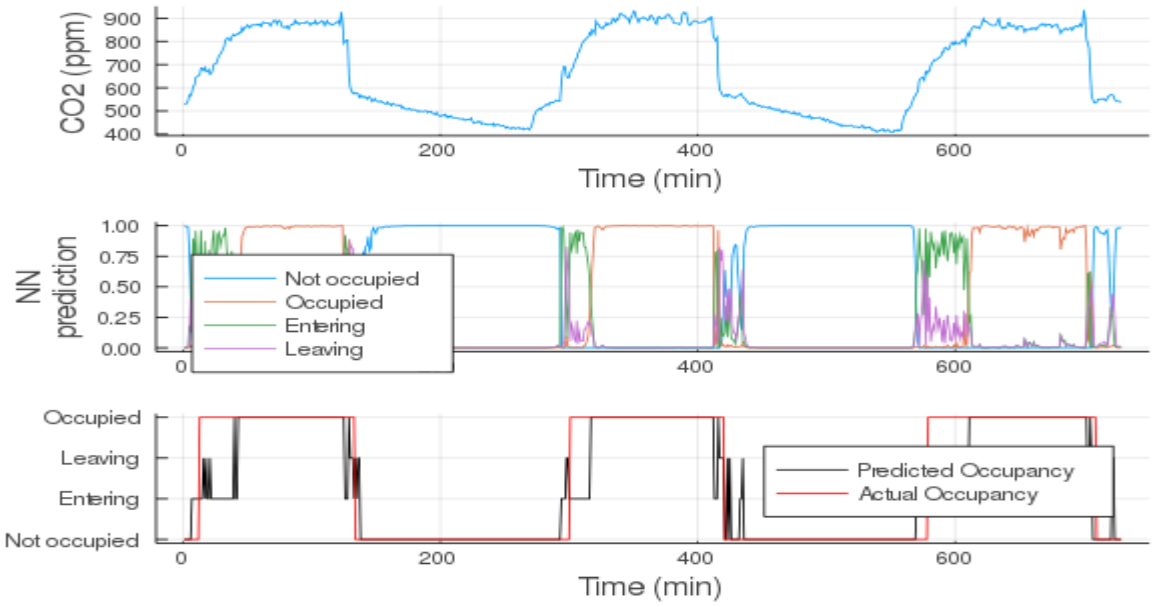

Figure 3 (Top) CO_2 concentration for the test period. (Middle) NN predictions of occupancies. (Bottom) Predicted occupancy compared to actual occupancy data.

Occupancy detection – Model development and testing for data from multiple devices

The training, CV, and test accuracies based on generated labels are 65%, 70%, and 60%, respectively. Figure 4 and Figure 5 show the predictions for seven days from two test devices. The CO_2 signal in Figure 4 has a lot of fluctuations and does not have a clear cyclic behaviour, as in Figure 3. Even with a difficult dataset, the model predicts occupancy that appears logical (based on visual inspection). Similarly, Figure 5 shows the prediction for a test device that has a cyclic behaviour. The prediction for this device also appears logical. Therefore, the actual accuracy in predicting occupancy can be higher than estimations based on generated labels. Furthermore, device data shown in Figure 4 is noisier than the device data shown in Figure 5. Hence, accuracies on a device level can be higher. Finally, the absence of true labels makes it hard to verify predictions during short occupancy periods, which are the result of CO_2 spikes. These spikes could also be due to air flowing from neighbouring rooms. Hence, a form of user feedback is important to refine model predictions.

[1] i.e., Occupied between ___P.M to ___A.M

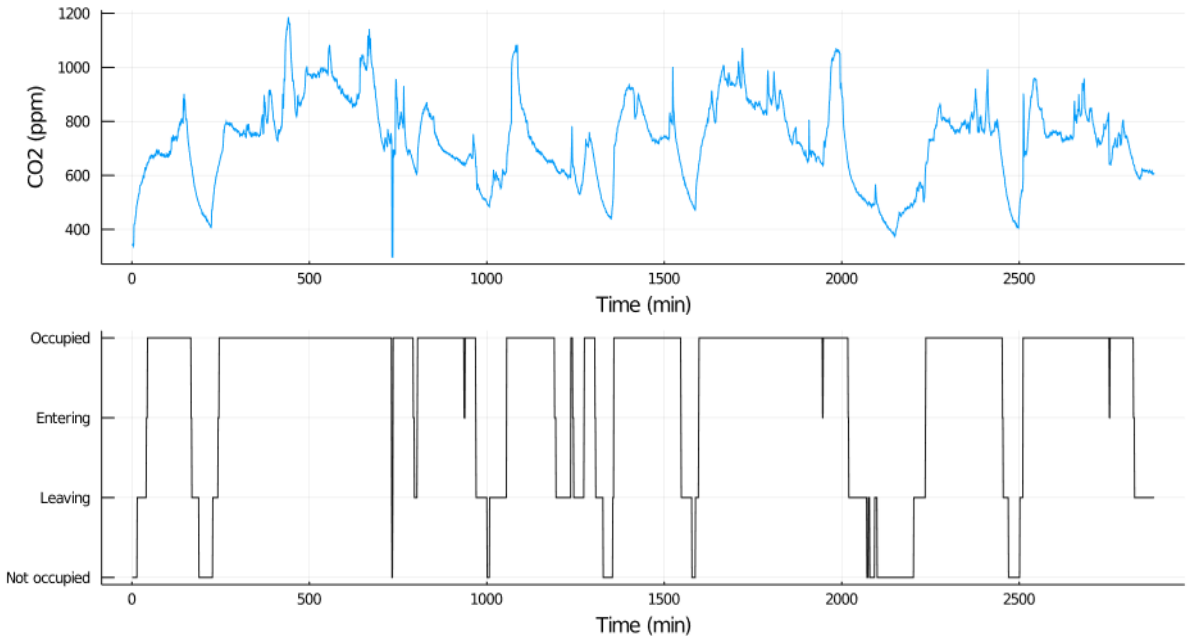

Figure 4 Occupancy detection (for seven days) in a test device with difficult CO_2 signal.

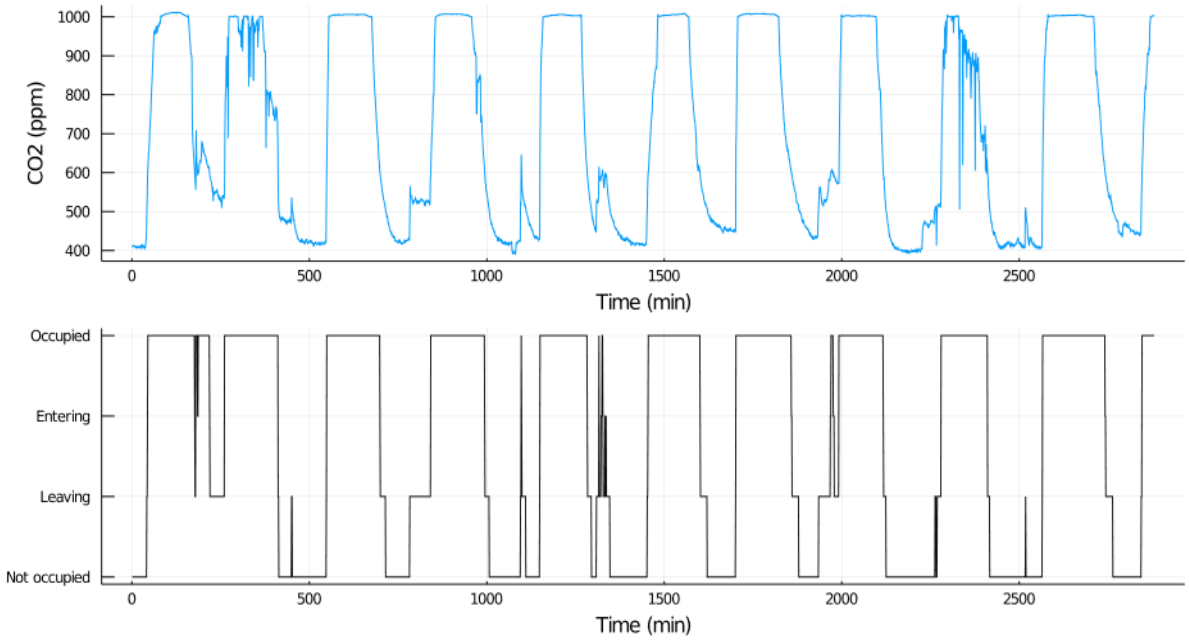

Figure 5 Occupancy detection (for seven days) in a test device with easier CO_2 signal.

CONCLUSION

The proposed methodology enables Renson to overcome the challenge of labelled data, adaptability, and scalability. The gradient-based labelling methodology enables the algorithm to adapt to a new context, and reduces the need for manual intervention. The NN model converts the labelled data into probabilities, which facilitates effective decision making. The two approaches combined enable a scalable method for occupancy detection.

REFERENCES

Ansanay-Alex, Guillaume. 2013. "Estimating Occupancy Using Indoor Carbon Dioxide Concentrations Only in an Office Building: A Method and Qualitative Assessment." REHVA World Congress on Energy Efficient, Smart and Healthy Buildings (CLIMA).

Calì, Davide, Peter Matthes, Kristian Huchtemann, Rita Streblow, and Dirk Müller. 2015. "CO2 Based Occupancy Detection Algorithm: Experimental Analysis and Validation for Office and Residential Buildings." Building and Environment. https://doi.org/10.1016/j.buildenv.2014.12.011.

Chen, Zhenghua, Chaoyang Jiang, and Lihua Xie. 2018. "Building Occupancy Estimation and Detection: A Review." Energy and Buildings. https://doi.org/10.1016/j.enbuild.2018.03.084.

Innes, Mike. 2018. "Flux: Elegant Machine Learning with Julia." Journal of Open Source Software 3 (25): 602. https://doi.org/10.21105/joss.00602.