IET Wireless Sensor Systems

Special Issue: Smart Cities and Smart Sensory Platforms

IET Journals
The Institution of Engineering and Technology

Designing learned CO_2-based occupancy estimation in smart buildings

ISSN 2043-6386
Received on 31st January 2018
Revised 17th August 2018
Accepted on 29th August 2018
E-First on 13th November 2018
doi: 10.1049/iet-wss.2018.5027
www.ietdl.org

Colin Brennan[1,2] ✉, Graham W. Taylor[1,2], Petros Spachos[1]

[1]*School of Engineering, University of Guelph, Guelph, Canada*
[2]*Vector Institute for Artificial Intelligence, Toronto, Canada*
✉ *E-mail: cbrennan@mail.uoguelph.ca*

Abstract: Many applications, such as smart buildings, crowd flow, action recognition, and assisted living, rely on occupancy information. Although the use of smart cameras and computer vision can assist with these tasks and provide accurate occupancy information, it can be cost prohibitive, invasive, and difficult to scale or generalise to different environments. An alternative solution should bring similar accuracy while minimising the listed problems. This work demonstrates that a scalable wireless sensor network with CO_2-based estimation is a viable alternative. To support many applications, a solution must be transferable and must handle not knowing the physical system model; instead, it must learn to model CO_2 dynamics. This work presents a viable prototype and uses the captured data to train machine learning-based occupancy estimation systems. Models are trained under varying conditions to assess the consequences of design decisions on performance. Four different learning models were compared: gradient boosting, k-nearest neighbours (KNN), linear discriminant analysis, and random forests. With sufficient labelled data, the KNN model produced peak results with a root-mean-square error value of 1.021.

1 Introduction

As the trend in technology continues to make computational power and integration among devices more accessible, there is an increase in demand for intelligent, connected systems. Consumers seek systems that can anticipate their needs and act accordingly to make their lives easier. In order for any smart system to meet this challenge, it needs to have enough information about what specific action is required of it and about the state of the environment in which it must act [1]. This critical context must be observed and measured in some way to be integrated into the system for consideration.

Specific applications for these smart systems are already being addressed, including systems for smart homes and smart offices, crowd behaviour analysis, and human action recognition. Common to each of these applications is that these systems require context about where users are within the observable space. While some occupancy-capable systems do perform well for some of these applications, it is a challenge to produce a system that can generalise to work for a multitude of settings. The infrastructure available for a large office building seeking smart office control is quite different from that available in a smart home. Smart camera systems may be well-suited for such an office setting, but may be impractical for large crowds and may not be welcome within a user's home. To support a wider range of problems, the occupancy context should include an estimation of the number of persons present. In addition to generalising across applications, a suitable solution should therefore provide the capability for anonymous and non-invasive sensing.

Wireless sensor networks (WSNs) present a viable solution to this problem. By selecting sensors with environmental awareness, it is possible to create sensing nodes that can estimate occupancy while preserving the anonymity of occupants. By networking these sensing nodes, the occupancy estimation system can generalise different applications and domains of varying sizes. To address the challenge of domain-specific environmental variation, WSNs may be accompanied by machine learning models. While quite capable, the design space for these learned models is large and not easily bound by the traditional design constraints of a WSN. In order to design an effective overall system, the impact of design decisions on learning-based systems must be understood.

In this work, a WSN prototype is presented, which uses CO_2 measurements to facilitate occupancy estimation at its nodes. Deployed in a conference room setting, this prototype is used to train four learned estimation models. The consequences of some design decisions for these models are observed and discussed. The layout of this work is as follows: Section 2 contains an overview of sensing modalities and learning models used in related occupancy estimation works. The system implementation is provided in Section 3 which includes details on the proposed prototype as well as the conditions of its deployment. Section 4 pertains to the machine learning models, and expands upon the design parameters of interest and the scope of experiments. The results of these experiments are presented and discussed in Section 5 and are followed by concluding remarks in Section 6.

2 Related work

For applications that focus on the actions and behaviours of a few occupants, occupancy can be obtained through tracking each user's location. Wearables, such as a user's smart-phone or other communication sensor beacons, can be used to this end [2, 3]. Recently, Manley and Deogun [4] showed that it is possible to obtain location accuracies of five metres by using machine learning models based on received signal strength from statically placed beacons throughout a target space. WSN systems based around the inclusion of a wearable component can provide advantages in terms of specific user location accuracy. However, the per-person hardware requirements may limit the scale of installation, and may not abide by the desire to provide an anonymous, non-invasive solution for occupancy tracking.

Occupancy and location information have also previously been obtained through sequences of activations of unique sensors with known locations. In [5], a heterogeneous sensing network is used to identify the activities of smart home residents. To identify the activities of interest, some of the sensors considered include: motion, light, door, contact, and temperature. Many of the target activities, such as sleeping, cooking, and personal hygiene, can be uniquely associated with a single room of a smart home and thus provide room occupancy information. This type of network has been used for applications including activity recognition for assisted living [6], and in producing energy saving

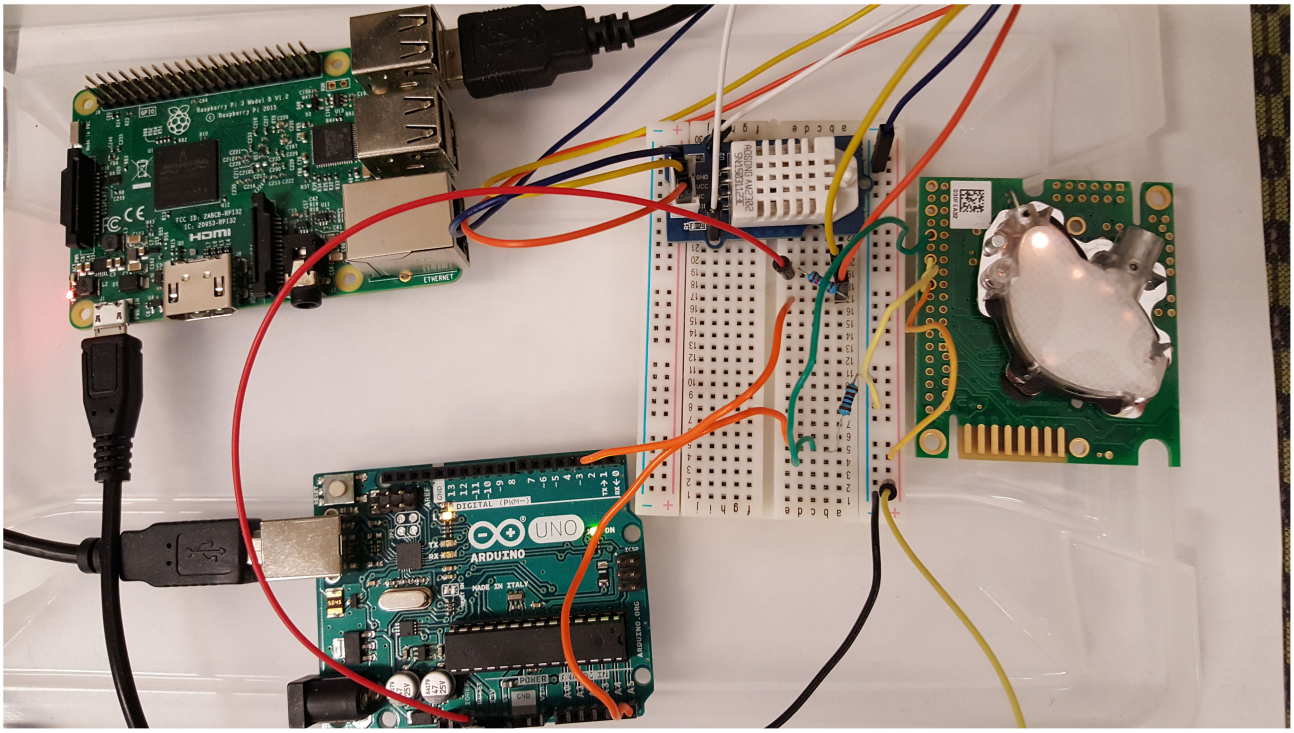

Fig. 1 *Prototype for the experiments. Left: Raspberry Pi and Arduino processing boards; centre: DHT22 sensor; right: K30 CO_2 sensor*

recommendations [7]. While capable of producing occupancy information as a byproduct, this information is only available if an occupant is participating in one of the target activities. Any space not associated with a task, or any task that cannot detect multiple occupants, restricts the generalisation of this kind of deployment.

In contrast to the user- and activity-centric occupancy methods, other systems provide occupancy information from the context associated with fixed landmarks, such as a room or entrance. Making use of sensors like door contact, passive infra-red (PIR), and motion, these systems are effective in detecting persons in a target space by observing the sensor activations over the network of deployed devices [8]. These deployments preserve privacy, are usually low cost, and are well-suited for WSN deployments of varying scales. However, the output of these sensors is limited to the binary occupancy state, as these modalities do not generalise well for measurement of the number of persons, unless paired with additional sensors. For the binary detection task, accuracies have been reported on the order of 99% [9], indicating that applications that can accommodate having only detection information can greatly benefit from these approaches.

Among the candidate sensors to be paired with occupancy detection sensors are environmental sensors, which obtain occupancy information through changes in environmental readings in a local proximity. Environmental assessment WSNs, such as those for indoor air quality applications [10, 11] and smart climate control systems [7, 8, 12, 13], make use of sensors that can observe volatile organic compounds, temperature, and humidity. By not focusing on specific users, the environmental methods satisfy anonymity and the indirect nature avoids requiring user attention and interaction.

A common metric exploited is the change in CO_2 production relative to the number of persons present. This relation is dependent on the deployment space, and as such, requires either knowledge of the target space to model the relation explicitly [12, 14, 15], or the ability to learn the relation from observation [9, 14, 16]. The required prior knowledge of the former model restricts the generalisation of a solution and would hinder any redeployment for a WSN; thus, a learned solution is preferable. Occupancy estimation systems using learned CO_2-to-occupancy models have been previously demonstrated [9, 16–19]. These solutions make use of techniques including artificial neural networks (ANN), classification and regression trees, gradient boosting (GB), linear discriminant analysis (LDA), random forests (RF), and their variants.

An important consideration for learned models, when compared to those designed by prior knowledge, is that the physical domain parameters are exchanged for design parameters associated with the machine learning system. Parameters, which impact the final performance of the system, include sensor and model selection, data representation and pre-processing, hyper-parameters associated with the model, and the parameters adjusted by learning. The resulting design space is large and the parameters are not as interpretable as the physical domain parameters. For the occupancy detection problem, effects on performance for some choices in this space have been previously investigated [9]. However, the models and decisions for estimating the number of persons differ from the occupancy detection domain, and must be investigated as well. When deploying a design, it is also important to decide how much sensor data to use in each estimation. This time window selection was studied in [17, 19] for their respective ANN-based models. Given the diverse behaviours of learning architectures, understanding how the significance of the windowing parameter is affected by the choice of a model could further inform design decisions. The consequences of such design decisions will be considered in this work.

3 System implementation

In this section, we provide a description of the hardware components that were used to build the prototype, followed by the node deployment and data collection process.

3.1 Hardware components

The proposed platform consists of a number of hardware components for each node. Components were selected to facilitate the sensing and distributed processing requirements of the system. The prototype, shown in Fig. 1, has the following two main units:

- *Sensing unit*: to achieve the desired environmental observation, two sensors are utilised: the K30 CO_2 sensor [20] and the DHT22 temperature and humidity sensor [21].

The K30 sensor, seen on the right of Fig. 1, measures the local concentration of CO_2. This sensor outputs a continuous voltage

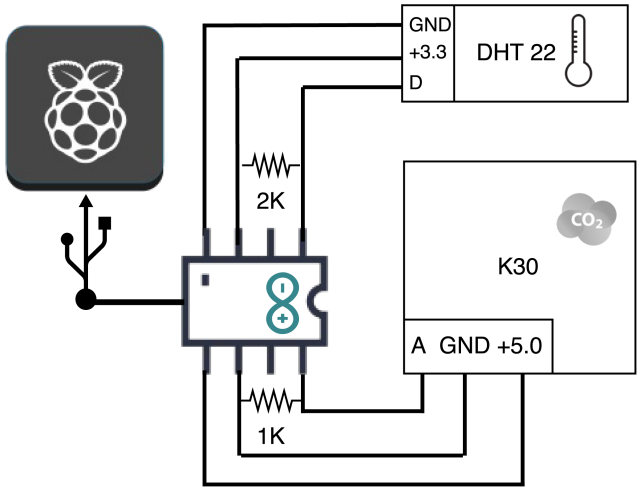

Fig. 2 *Prototype system wiring. The DHT22 is connected for a digital read at 3.3 V with a 2 kΩ pull-up resistor. The K30 is configured for an analogue measurement within 5 V with a 1 kΩ pull-down resistor*

proportional to the parts-per-million (ppm) concentration of CO_2. Received at a maximum rate of measurement of 2 s, these reports are in the range of 0–5000 ppm [20], which encompasses the expected range for an office setting, which is <1000 ppm [22].

The second sensor, the DHT22, as seen in the centre of Fig. 1, provides readings of temperature and humidity within the ranges of −40 to 80°C and 0–100% humidity [21]. Measurements are available at a sensing period of 2 s. These operational conditions are more than sufficient for the indoor setting where rapid environmental changes are rare due to HVAC regulation.

- *Processing unit*: for each sensing node, it is necessary to have interfacing capabilities with the appropriate sensors, the capability to both store and communicate readings to a centralised location, and the capacity to serve as a filtering point for data. For these reasons, we included an Arduino UNO [23] and a Raspberry Pi 3B [24] at each node.

Both of the selected sensors are well supported on Arduino, but the Arduino alone is insufficient for storage and communication requirements. As such, the Arduino code used in the prototype focuses on collecting and formatting the sensor data. This is achieved by polling each sensor in turn, converting from analogue voltage to ppm concentration, and communicating complete readings over serial to the Raspberry Pi. The Arduino is programmed to then await the next reading interval. The Raspberry Pi board satisfies the remaining processing requirements by enabling network connectivity and on-device data backups for the sensor nodes.

The selected sensors are wired to the Arduino for both power and communication. The K30 sensor runs from the available 5 V source, drawing 40 mA on average during operation [20], and is read using an analogue I/O pin. The DHT22 sensor, attached to the 3.3 V source, uses 2.5 mA during a data request [21]. This request is communicated through a digital I/O pin. The Arduino UNO board receives power from a USB connection to the Raspberry Pi, which also permits serial communication between the two devices. As the main processing unit, the Raspberry Pi is powered through a wall adapter, and for this experiment, is provided network connectivity through WiFi. Each node communicates its data to a shared data server. The prototype wiring is illustrated in Fig. 2.

3.2 Deployment and data collection

Two of the prototype sensing nodes were placed in a conference room of dimensions $6\,\text{m} \times 5\,\text{m} \times 3\,\text{m}$ $(W \times D \times H)$ to observe regular room usage. The test environment is shown in Fig. 3. The room has one external wall, and two doors that were closed during testing. The first sensor was positioned in the middle of the central conference table and the second sensor was placed on a window

Fig. 3 *Environment for the experiments. Prototypes positioned on the centre table and window sill*

sill at a similar height. Each prototype took environmental readings at a rate of 0.3 Hz in order to encompass the rate of measurement of the two sensors and the required time for communication.

The data received from the sensing nodes is reduced to an effective rate of one per minute by taking the median of any readings occurring in the same minute. All data was streamed to a database for inspection and labelling. An example of the daily variation in CO_2 concentration is shown in Fig. 4. This is a base-case scenario where the room was empty for a long time, and then an occupant entered the room. It can be seen that the room was resting at an overnight steady state of 430 ppm until the first occupant arrived at 9:30. After the occupant entered the room, the CO_2 concentration continued to increase until reaching 530 ppm. After the room was vacated at 15:00, the readings can be seen to decay back to the initial resting state.

It is clear that when an occupant enters the room after an overnight resting period, the sensing node detects it, as expected. However, the challenge is to detect variation in the number of occupants throughout the day. For training the estimation models, tests were conducted and labelled for occupancy levels up to four persons. All tests were manually labelled with start and end times, as well as the corresponding occupancy level. All tests started from zero occupants at the steady state, in order to observe model responses to transitions in the state. A summary of the data collected and the distribution of occupancy levels is shown in Table 1.

4 Evaluation models

With any learned system, it is important to consider the problem objectives in order to appropriately select a model for the desired output. Occupancy estimation may be considered as either a classification problem with integer occupancy classes or as a continuous value regression problem. Since a classification model may not preserve the significance of the distance between occupancy states, the regression formulation will be the focus in this work. Ordinal regression, which treats the outputs as discrete, but maintains their respective ordering, is also a suitable approach. However, it requires significant adaptation of the baseline methods, and is therefore reserved for future work.

For a smart sensor platform, it is desirable for candidate learning models to abide by the constraints of a distributed sensor network, such as limited computational resources. Linear models, variants of k-nearest neighbours (KNN), and decision trees are among the candidate learning models regularly considered due to their ease of deployment and low computational complexity [9, 25, 26].

Considering the constraints of our distributed platform and previous assessments in the occupancy detection problem, we selected the following models for evaluation:

i. GB,
ii. KNN,
iii. LDA, and
iv. RF.

Basic descriptions of these models can be found in Table 2. Each architecture will be evaluated by four performance metrics:

i. accuracy,
ii. root-mean-square error (RMSE),
iii. normalised root-mean-square error (NRMSE), and
iv. coefficient of variance (CV).

These metrics evaluate the success of model outputs y_i, in predicting true labels t_i, for a dataset of M examples.

As defined below, accuracy is the per cent of examples successfully predicted within rounding and RMSE is the mean distance between the prediction and true label, hence given

$$r(x) = \begin{cases} 1, & x < 0.5, \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$$\text{Accuracy} = \frac{1}{M}\sum_{i=1}^{M} r(|t_i - y_i|) \tag{2}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{M}(t_i - y_i)^2}{M}} \tag{3}$$

Accuracy and RMSE are standard metrics across problem domains for learning systems. However, in order to compare experiments of differing occupancy ranges and physical scales, performance indices should be normalised. For this reason, NRMSE and CV results will also be presented as defined below:

$$\text{NRMSE} = \frac{\text{RMSE}}{(\max(t) - \min(t))} \tag{4}$$

$$\text{CV} = \frac{\text{RMSE}}{\text{mean}(t)} \tag{5}$$

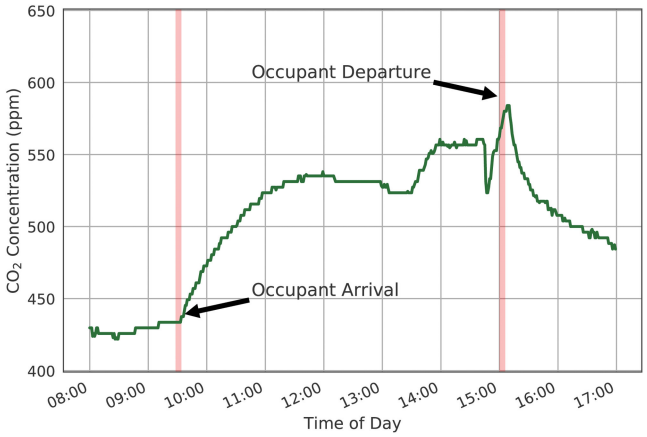

Fig. 4 *Example CO_2 readings from 12 January 2018 after median filtering. First arrival and last departures noticeably starting at 9:30 and 15:00*

4.1 Attributes of study

The performance of a learned model is dependent on both its own design as well as its training data. The attributes investigated in this work are from both categories of design.

The specific attributes under study are identified below, along with their considered values:

- *Series time window* (T_w)*:* Minutes of past sensor data to consider in each sample. T_w was tested in intervals of 2 min, up to 30 min, as well as at 1 min for smallest window comparison.
- *Patience* (p)*:* Minutes to delay output in order to wait for future context. Equivalently, it can be considered to be the time in minutes into the past to target prediction. Tests included values of p up to the value of T_w in intervals of 2 min.
- *Input feature combinations:* Input is chosen to be a concatenated combination of three features: the normalised sensor data series of length T_w, the mean of each sensor in the time window, and the time since the last occupancy state change. These features are labelled as Series, Mean, and Phase, respectively. Six combinations are considered, excluding the case of Phase alone, which would contain no environmental sensor data.
- *Neighbours* (k)*:* KNN specific, this determines the number of neighbouring training data examples to consider from dataset in making predictions. Consider $k \in \{1, 3, 5, 7, 9\}$.
- *Estimators* (n)*:* Capacity for GB and RF in terms of the number of weak estimators or trees. Consider $n \in \{10, 25, 50, 100, 250\}$.

The T_w parameter is of interest due to its effect on the size of the learned models. For real-time deployments, this parameter also dictates the necessary amount of sensor data that must be maintained by each node and thus is a concern for resource constraints. One of the anticipated complexities with estimating occupancy from CO_2 is that the concentration changes read by the sensing nodes are not immediate with respect to the occupant state changes. Being that the learned models depend on this sensor data, the resulting predictions will also be shifted in time from the true transitions. When permitted by an application, offsetting the data by this duration has been shown to reduce the final prediction error [17]. By searching over the patience term p for each T_w, it will be possible to observe the consequences optimising this time shift both independently of the window size as well as when the two parameters are optimised together.

The different input features have been selected to identify which attributes of the sensor data are most impactful for occupancy prediction. For each sample of the selected time window length, the Series feature captures the shape and trend of the sensor data. In contrast, the Mean feature ignores all trend information and presents only the magnitude of the sensor values. Since only one minute of data will be different between samples which neighbour in time, this feature will be the slowest changing. Since environmental sensing systems may be paired with detection sensors like PIR, the Phase feature is included to capture the state change information that this type of sensor may add.

In total, each model is subjected to 816 setting combinations. These tests are run for each of the model specific parameter settings, totalling 4080 combinations for the models other than

Table 1 Dataset distribution by number of occupants

Occupancy (persons)	0	1	2	3	4	Total
count, min	940	456	349	228	267	2240
percentage	42.0	20.3	15.6	10.2	11.9	100.0

Table 2 Candidate learning models

Model	Description
GB	additive model of decision trees. Each added tree aims to reduce the residual error of the existing model.
KNN	instance-based learning. Outputs an interpolation of the labels of the k most similar samples.
LDA	assumes Gaussian distributions in the data. Uses Bayes theorem to identify linear separation.
RF	ensemble of decision trees. Each tree is trained on subsets of the inputs and predictions are averaged.

Table 3 Preferred settings for each model, best RMSE performance in bold

Task	Model	Accuracy	RMSE	NRMSE	CV	T_w	p	Features	k	n
Estimation	GB	41.2	1.148	0.230	0.885	28	26	All	–	100
	KNN	47.8	**1.021**	0.204	0.787	10	4	All	9	–
	LDA	37.9	1.307	0.261	1.01	10	4	All	–	–
	RF	42.6	1.181	0.236	0.911	24	22	All	–	50

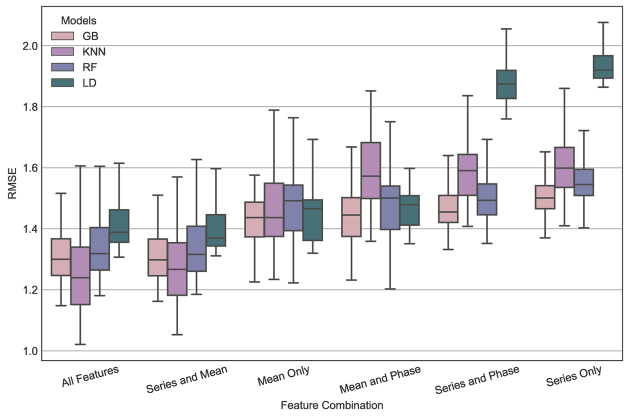

Fig. 5 *RMSE over all parameter tests for each input feature combination. Best candidates seen to include Series and Mean features*

LDA. The tests for each combination of settings are each run ten times and the average performance is taken. Models are trained with 70% of the available occupancy dataset and tested on the remaining 30%, randomly split for each of the ten iterations.

5 Results and discussion

For each model tested, parameter combinations exist that were able to estimate occupancy using data from the prototype. However, not every setting was able to learn the structure beyond bias in favour of the imbalanced labels. In order to assess which parameter combinations were successful, the results were grouped together by their attributes to observe the resulting variation. The settings found to provide the best performance for each model are shown in Table 3. Overall, within the parameter ranges tested, the highest observed accuracy was 47.8% and the lowest RMSE was 1.021 from the optimised KNN model. The model performing worst under its optimal settings was LDA, reporting an accuracy of 37.9% and RMSE of 1.307.

The model specific attributes, k and n, effectively shape the available capacity of the models. This kind of capacity tuning is a standard process for many learning systems and is not problem specific. Within the ranges considered in this work, it was found that the GB models would underfit when $n = 10$, but no other significant variation in performance was observed for the ranges tested.

Testing over the set of input feature combinations served to identify what form of data best supported occupancy estimation. The feature significance was observed to vary by model, indicating that the availability of preprocessing to generate these features must be considered when selecting a model. The range of RMSE results for each model and feature selection can be seen in Fig. 5.

For all four candidate models, it was observed that minimum RMSE was achieved when both the Series and Mean input features were used. Given this condition, the presence of the Phase feature yielded only slight improvement to the minimum RMSE reported by the KNN models. Based on performance when trained on either of these preferred features independently, having the mean sensor value provided better performance than having just the trend information. The LDA model was most significantly impacted by the absence of the Mean feature. This is seen by the near 15% worse performance in the two rightmost tests shown in Fig. 5, 'Series and Phase' and 'Series Only', which were the only two combinations lacking Mean. For the KNN model, performance decreased when testing with Mean and Phase together instead of

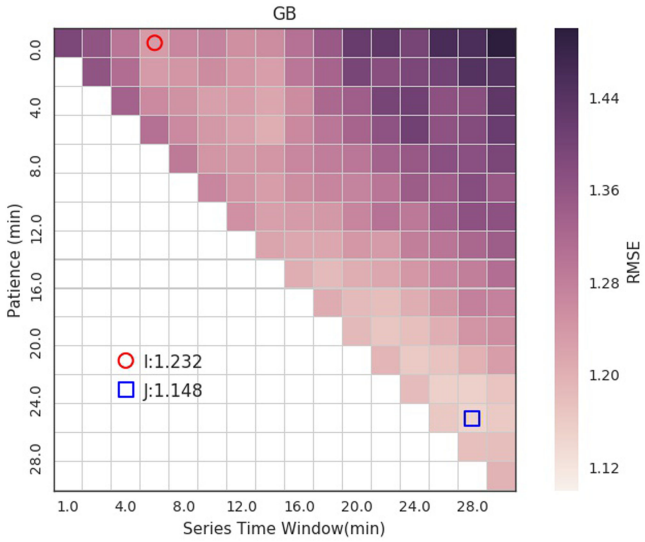

Fig. 6 *Best RMSE results from sweeping (T_w, p) pairs for the GB model. Final parameters from optimising parameters independently (I) and jointly (J) shown with circle and square markers*

with the Mean feature alone. While Mean is the most stationary feature between samples, Phase is constantly changing. This conflicting pace in the input data may have inhibited the algorithm in finding sufficiently similar candidate samples.

The GB and RF models present the closest behaviour across all feature combinations and remain fairly consistent when exposed to Mean or Series exclusively. Whereas the KNN and LDA models learn though similarity and linear separability in the training data, the tree-based models are capable of more complex segmentation. Given this flexibility in rule-based learning, it makes sense that these models would be more likely to find suitable solutions for the broader range of input structures. It is also likely that if a preferable tree was found by one of the models, a comparable one would be found by the other and thus both models would present similar behaviour.

The remaining data attributes, T_w and p, impact the scale of time considered in the data samples and labels. To study the impact of these parameters, the best RMSE for each model under each pair of settings is isolated. In general, with optimising T_w, it was observed to be mostly convex; however, the point at which the minimum occurred varied for each model when the p parameter is changed. This can be seen by the trough in the RMSE results running parallel to the $T_w = p$ line for the GB model in Fig. 6.

When optimising the time data parameters independently, it is easiest to first observe the relation between time window T_w and RMSE performance with $p = 0$. After optimising T_w from this relation, the model can be fixed at that size and tested at different patience shifts to identify the best RMSE for that model and when it can be expected. When optimising jointly, a more thorough search is required to test each pair of settings and find the resulting optimum. A condensed representation of these searches for each model can be seen in Fig. 7. For each model, the initial $p = 0$ surface is shown, as well as the performance curve for the value of p resulting in the global minimum for the model. These curves are joined by the surface of minimum RMSE solutions for each T_w at its optimal p. The solution from independent parameter selection for each model is marked with a circle, and the joint selection solution is marked with a square. The respective parameter pairs

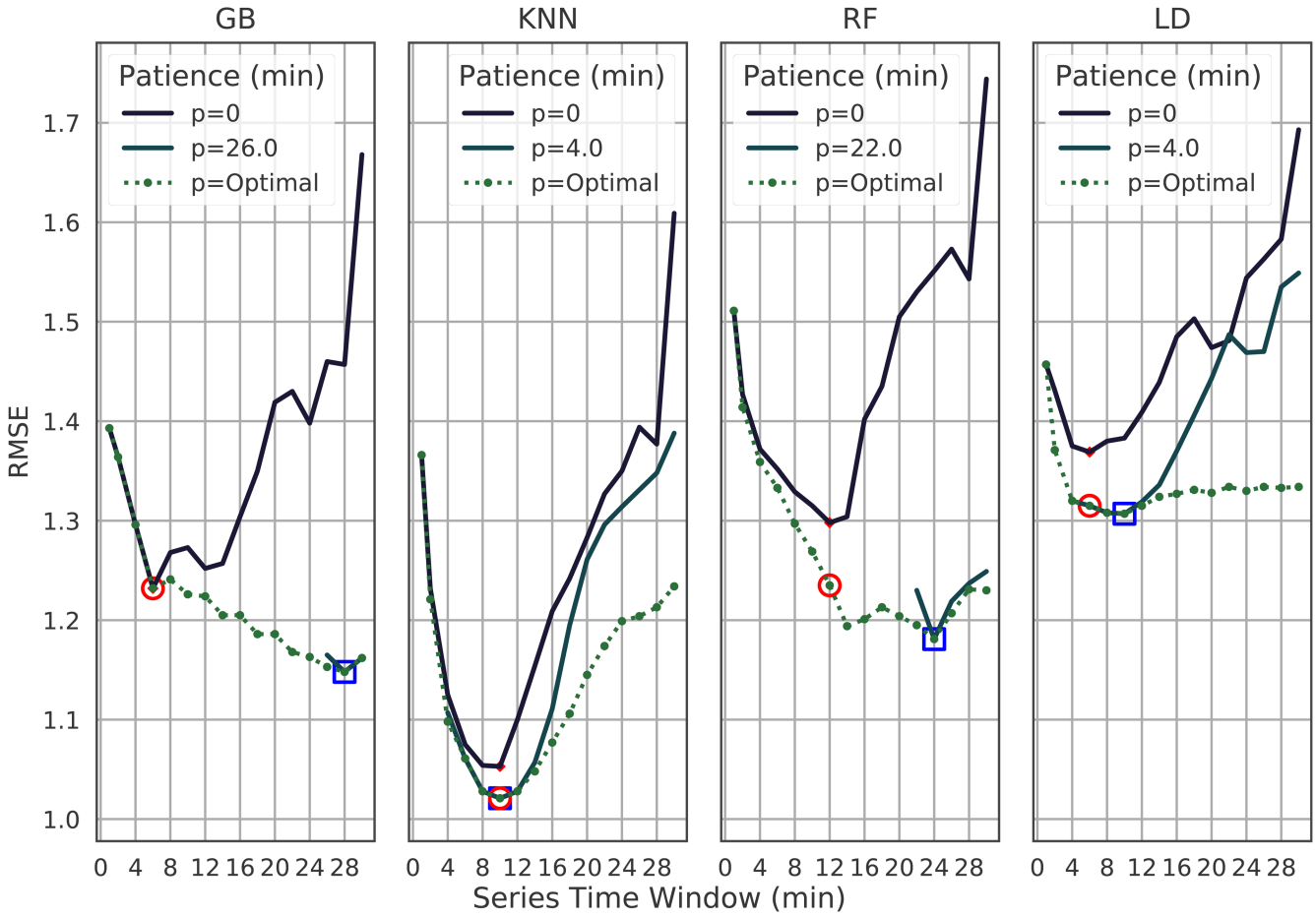

Fig. 7 *RMSE performance with respect to window size and patience parameters for each model. Final models from optimising parameters independently and jointly shown with circle and square markers*

Table 4 Model time parameter optimisations and differences

Model	Optimisation	T_w	p	RMSE
GB	independent	6	0	1.232
	joint	28	26	1.148
	difference	22	26	0.084
KNN	independent	10	4	1.021
	joint	10	4	1.021
	difference	0	0	0
LDA	independent	6	4	1.315
	joint	10	4	1.307
	difference	4	0	0.008
RF	independent	12	12	1.235
	joint	24	22	1.181
	difference	12	10	0.054

can also be read in Table 4 as well as the difference between two optima.

Of the candidate learning models, only the KNN model search converged to the same T_w and p settings for the joint and independent searches: $T_w = 10$, $p = 4$, RMSE = 1.021. The maximum displacement was observed in the GB solutions, as seen in Fig. 6, where the independent solution needed an additional 20 min for each parameter to reach the joint optima. The available RMSE improvement available between the two solutions was <0.09 for all models and often requires expanding both parameters. This improvement in performance remains an option only if the intended application can relax constraints for both model size and real-time performance to support these expansions. As the joint search process is more involved, only in select cases would it be worth pursuing.

When comparing the parameter solutions found across models, it is interesting that the RF and GB models require greater context and patience than the simpler models. Since the solution space for the tree-based models is larger, these models may require this additional context in order to eliminate a greater volume of ineffective solutions.

Within this investigation, none of the attribute permutation tests, including those with preferred settings in Table 3, exceeded 50% accuracy. The best RMSE value of 1.021 is also indicating that the average estimation error exceeds one person. While not surpassing any state-of-the-art results, the acquisition of these values still validates the occupancy context available through the prototype and how different learning architectures might be optimised for the estimation task. Fig. 8 shows an example test output from the optimised KNN model. During the state transition between 20–30 min into the test, it can be seen that while the prediction is converging to the expected state, this region would be

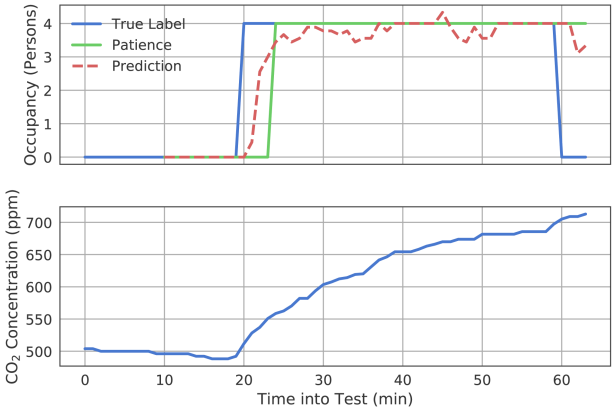

Fig. 8 *Example test output from KNN with preferred settings. Model prediction shows occupancy state transition occurring between the true label and patience delayed label at 23 min*

evaluated as inaccurate and contribute to a higher RMSE. RMSE was suitable in characterising the impact of the T_w and p parameters, but a metric for comparing transient behaviours in prediction models would be valuable to capture behaviour like this. It may be advantageous to consider attributes from control systems such as rise time or time to peak for this task.

In this work, four learning models were assessed, and KNN returned the overall best RMSE and accuracy. As an instance-based learning model, its performance relies on test cases being similar to training cases observed. This means that KNN models will likely not be able to generalise as well as GB or RF models. Applications and deployments where it is not possible to obtain a breadth of occupancy state examples may therefore prefer the RF model instead. Having a low n in its preferred settings, the RF model would also be attractive for computationally constrained applications, such as where occupancy information is needed at each sensor in a network and must be computed on the node itself. These application constraints must be considered alongside the attribute selection trade-offs.

6 Conclusion

We presented a prototype WSN for environmental monitoring in a smart building. Through localised environment measurement at each sensing node, the prototype provides the necessary information to enable occupancy context for a smart system as an anonymous, non-invasive, and simple solution. This prototype network facilitated the testing of four different architectures tasked with learning the relation between the CO_2 and occupant state. Of the candidate learning models which included GB, KNN, LD, and RF, the KNN model returned the best performance under preferred settings with an accuracy of 47.8% and an RMSE of 1.021. If sufficient labelled data can be obtained for a deployment, the KNN model would be an appropriate selection.

The data collected from a deployment of the prototype was used for characterising the consequences of different design decisions relating to the model and training data. It was found that the performance for all candidate models improved when the training data included information on both the magnitude and structure of the sensor data. The GB and RF models were seen to maintain more consistent performance when either type of information was absent. At peak performance, these two models also required more historical data and prediction patience than both the KNN and LD models. Jointly optimising the window size and delay parameters yields slight improvements to the RMSE over independent optimisation. The additional context required for the improved solution, however, may conflict with real-time performance constraints.

In future work, this deployment will be scaled to a larger environment and occupancy range to assess the effect of task complexity on these observations. Consideration of performance metrics which incorporate transient behaviour may further improve comparisons between models and facilitate more appropriate attribute optimisations.

7 References

[1] Zanella, A., Bui, N., Castellani, A., *et al.*: 'Internet of things for smart cities', *IEEE Internet Things J.*, 2014, **1**, pp. 22–32

[2] Zhu, Q., Chen, Z., Soh, Y.: 'Smartphone-based human activity recognition in buildings using locality-constrained linear coding'. IEEE 10th Conf. on Industrial Electronics and Applications (ICIEA), Auckland, New Zealand, June 2015, pp. 214–219

[3] Biton, R., Katz, G., Shabtai, A.: 'Sensor-based approach for predicting departure time of smartphone users'. 2nd ACM Int. Conf. on Mobile Software Engineering and Systems, Florence, Italy, May 2015, pp. 146–147

[4] Manley, E., Deogun, J.: 'Location learning for smart homes'. 21st Int. Conf. on Advanced Information Networking and Applications Workshops AINAW '07, Niagara Falls, Canada, May 2007, pp. 787–792

[5] Chen, L., Nugent, C., Wang, H.: 'A knowledge-driven approach to activity recognition in smart homes', *IEEE Trans. Knowl. Data Eng.*, 2012, **24**, pp. 961–974

[6] Fahad, L., Rajarajan, M.: 'Anomalies detection in smart-home activities'. IEEE 14th Int. Conf. on Machine Learning and Applications (ICMLA), Miami, Florida, December 2015, pp. 419–422

[7] Schweizer, D., Zehnder, M., Wache, H., *et al.*: 'Using consumer behaviour data to reduce energy consumption in smart homes: applying machine learning to save energy without lowering comfort of inhabitants'. IEEE 14th Int. Conf. on Machine Learning and Applications (ICMLA), Miami, Florida, December 2015, pp. 1123–1129

[8] Lu, J., Sookoor, T., Srinivasan, V., *et al.*: 'The smart thermostat: using occupancy sensors to save energy in homes'. Proc. of the 8th ACM Conf. on Embedded Networked Sensor Systems (SenSys '10), Zurich, Switzerland, November 2010, pp. 211–224

[9] Candanedo, L., Feldheim, V.: 'Accurate occupancy detection of an office room from light, temperature, humidity and CO_2 measurements using statistical learning models', *Energy Build.*, 2016, **112**, pp. 28–39

[10] Spachos, P., Hatzinakos, D.: 'Real-Time indoor carbon dioxide monitoring through cognitive wireless sensor networks', *IEEE Sens. J.*, 2016, **16**, pp. 506–514

[11] Gadyszewska-Fiedoruk, K.: 'Correlations of air humidity and carbon dioxide concentration in the kindergarten', *Energy Build.*, 2013, **62**, pp. 45–50

[12] Imanishi, T., Tennekoon, R., Palensky, P., *et al.*: 'Enhanced building thermal model by using CO_2 based occupancy data'. IECON 2015–41st Annual Conf. of the IEEE Industrial Electronics Society, Yokohama, Japan, November 2015, pp. 003116–003121

[13] Oldewurtel, F., Sturzenegger, D., Morari, M.: 'Importance of occupancy information for building climate control', *Appl. Energy*, 2013, **101**, pp. 521–532

[14] Weekly, K., Bekiaris-Liberis, N., Jin, M., *et al.*: 'Modelling and estimation of the humans' effect on the CO_2 dynamics inside a conference room', *IEEE Trans. Control Syst. Technol.*, 2014, **23**, pp. 1770–1781

[15] Ebadat, A., Bottegal, G., Varagnolo, D., *et al.*: 'Regularized deconvolution-based approaches for estimating room occupancies', *IEEE Trans. Autom. Sci. Eng.*, 2015, **12**, pp. 1157–1168

[16] Jiang, C., Masood, M., Soh, Y., *et al.*: 'Indoor occupancy estimation from carbon dioxide concentration', *Energy Build.*, 2016, **131**, pp. 132–141

[17] Han, H., Jang, K., Han, C., *et al.*: 'Occupancy estimation based on CO_2 concentration using dynamic neural network model'. Proc. of the 34th AIVC–3rd TightVent–2nd Cool Roofs'–1st venticool Conf., Athens, Greece, September 2013

[18] Alam, A.G., Rahman, H., Kim, J.K., *et al.*: 'Uncertainties in neural network model based on carbon dioxide concentration for occupancy estimation', *J. Mech. Sci. Technol.*, 2017, **31**, pp. 2573–2580

[19] Zuraimi, M., Pantazaras, A., Chaturvedi, K., *et al.*: 'Predicting occupancy counts using physical and statistical CO_2-based modelling methodologies', *Build. Environ.*, 2017, **123**, pp. 517–528

[20] 'K30 CO2 Sensor Module', Available at https://www.co2meter.com/products/k-30-co2-sensor-module, accessed 26 September 2017

[21] 'DHT22 Temperature-Humidity Sensor+Extras', Available at https://www.adafruit.com/product/385, accessed 18 April 2017

[22] Nathanson, T.: 'Indoor Air quality for office buildings'. Health Canada, Cat:H46-2/93-166E, 1993

[23] 'Arduino Uno Rev3', Available at https://store.arduino.cc/usa/arduino-uno-rev3, accessed 25 January 2018

[24] 'Raspberry PI 3 Model B', Available at https://www.raspberrypi.org/products/raspberry-pi-3-model-b/, accessed 25 January 2018

[25] Beltran, A., Erickson, V., Cerpa, A.: 'Thermosense: occupancy thermal based sensing for HVAC control'. Proc. of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings, Roma, Italy, November 2013, pp. 1–8

[26] Sangogboye, F., Kjaergaard, M.: 'PROMT: predicting occupancy presence in multiple resolution with time-shift agnostic classification', *Comput. Sci. Res. Dev.*, 2017, **33**, pp. 105–115