# Predicting Human Occupancy with Recurrent Neural Networks and Ambient Sensors

Kristoffer Tangrand[1] and Bernt Bremdal[2]

[1,2]Department of Computer Science and Computational Engineering, UiT The Arctic University of Norway

## Abstract

This paper presents results from ongoing research with a goal to use a combination of time series from non-intrusive ambient sensors and deep recurrent neural networks to predict room usage at a university campus. Training data was created by collecting measurements from ambient sensors measuring room $CO_2$, humidity, temperature, light, motion and sound, while the ground-truth counts was created manually by human observers. Results include analyses of relationships between different sensor data sequences and recommendations for a prototype predictive model using deep recurrent neural networks.

***Index terms***— Indoor Air Quality, Occupancy Prediction, PCA, LSTM, GRU, Neural Architecture Search, Deep Learning, Internet of Things

## 1 Introduction

With the advent of Internet of Things (IoT) a multitude of monitoring and control opportunities arise. The development of smarter buildings, neighborhoods and cities have already embraced this. Energy use and indoor climate control are central aspects related to the performance of buildings. Today there is little knowledge on how a particular building is actually used. What part of the indoor area is populated at different hours? Selective energy use can lead to more efficient buildings [2]. Monitoring the number of people in the specific rooms of a building can be used to achieve a more focused and efficient use of energy in a building. That in turn requires the ability to compare an estimate of space occupation and energy use.

Room occupancy has historically been monitored by means of cameras and smart phones(Wi-Fi, Bluetooth). But this raises privacy issues. However, ambient sensors that can monitor levels of CO2, temperature, humidity, illumination and noise are present in many office and educational buildings today.

The problem addressed in this work is to explore ways to utilize existing indoor sensors and to determine the number of people in different rooms at different hours during the day with the data harvested from these. A prerequisite for this is that there exists a statistical and causal relationship between the observed parameters and the unobserved state(the number of people).

## 2 Related work

$CO_2$ has proven a reliable indicator for occupancy detection [10]. Further, $CO_2$, illumination and sound are known to be highly correlated with human occupancy [1]. Machine Learning algorithms like Support Vector Machines and Random Forest have shown promise on such sensor data [12]. Feed-forward Neural Networks are used in [6] to predict occupancy numbers from $CO_2$, sound, temperature and motion.

## 3 Method

To explore the possibilities described above a device that combines different sensors and enable synchronization of time series from each sensor was used (see Figure 1). This enabled individual and combined analyses of time series with a resolution of approximately 40 recordings per hour.



https://doi.org/10.7557/18.5159

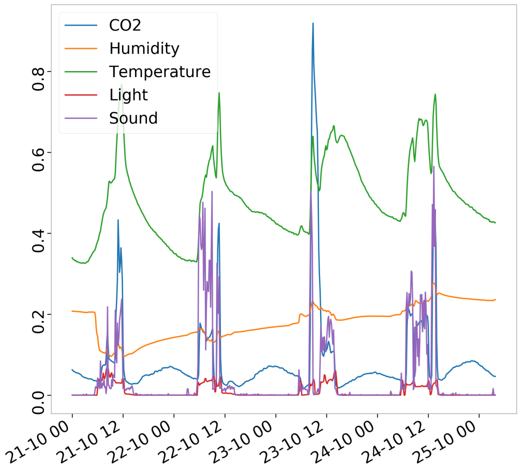


Figure 1: Four days of ambient sensor data. Note that the values are scaled and normalized for the plot.

The data set was collected from ambient sensors placed in 9 relatively small study rooms that students usually use between classes or for studying alone and in groups. The data the sensors provided:

- $CO_2$, as measured in parts per million(ppm)
- Humidity, as measured by the amount of water vapour in the air
- Temperature, measured in Celsius
- Illumination, measured in lux
- Motion, which is an PIP-sensor that returns a binary signal
- Sound level, measured in decibel

Human observers manually counted people to establish the ground-truth data for occupancy.

### 3.1 Preprocessing

First, the Pearson correlation between features was investigated. Pearson correlation returns a value between -1 and 1 where -1 is the maximum negative correlation and 1 is the maximum positive. The closer the value gets to 0 the lower the correlation in any direction, and at 0 there is no correlation.

A common approach to pre-process training data and (hopefully) reduce over-fit is to apply PCA to reduce the number of correlated features. PCA can be used to the determine the explained variance between the features. PCA helps to determine a variable's relative contribution to the overall. It is also used to determine statistical co-variance between variables.

### 3.2 Time-dependency Inspection

As can be seen in Figure 1: Time series of this type often display a high degree of seasonality on a daily basis. Inspection of the data suggested that there were profound differences between rooms depending on their use, location in the building, heating and ventilation. The CO2 variable would dominate in one part while temperature and humidity gradients would be more pronounced in other parts of the building. Latency issues became evident too. It takes time for CO2 levels, humidity and temperature to increase when people meet in a room. Ventilation may flush stale air more or less effectively. Seasonal and daily changes in weather may affect conditions too. Spring or autumn with low solar altitudes may produce rays that effectively heat up a room in one part, while rooms on the shady side may require heating. These occurrences suggested a high degree of dynamics. States could be affected by situations or actions that happened several hours or even days before. Dry periods with a lot of sun would heat up concrete constructions and create more dust than rainy and chilly days. A shift could impact the observations profoundly. The conditions in a room caused by six people having a morning meeting could be logged quite differently from a similar situation in the afternoon as former meetings that day could create an "atmospheric legacy".

### 3.3 Prediction Models

In recent years, RNN architecture Long Short-Term Memory(LSTM) has become immensely popular due to its proven effectiveness in a number of problem areas involving sequential data [8]. The main trick to achieving such great performance is an elaborate setup of gates which lets each neuron in the RNN control which information to forget or remember depending on the patterns in the training

| Percentage of rooms | 100% | 90% | 50% |
|---|---|---|---|
| Variance retained in only one component | 80% | 95% | 98% |

Table 1: Showing the variance single principal component is able to retain

sequence [9]. A simplified version called Gated Recurrent Units(GRU) was proposed more recently in [3]. Comparison between LSTM and GRU has been described in the literature, with performance being found to be roughly equal in [4].

Designing architectures for neural networks has traditionally been manually done by humans. With the advent of great generalization tools and more powerful hardware, this is beginning to change, and so-called Neural Architecture Search(NAS) is gaining traction [7]. With this in mind, 3 investigations were performed:

1. **Neural Architecture Search** First, an architecture search was performed using the Keras wrapper Talos [13]. This package makes it simple to set up extensive architecture searches to automate the process of optimizing hyperparameters.
2. **How different features influenced the learning process** A second analysis was performed to investigate the learning potential and influence each of the training features had on the model.
3. **Reproducibility and the influence of random initial parameters** Uncertainty is often experienced with regards to achieving reproducible results when training neural networks. The initial weights of a neural network are usually selected according to a random distribution.

# 4 Results

First, the Pearson correlation between the features was examined. The result is shown in Figure 2. We observe that most signals correlate to varying degrees, except humidity.

Next, PCA was performed, shown in Figure 3. The result confirms mostly the same as we know from the Pearson coefficients. Humidity is diverging from the other parameters, and there is not even correlation between humidity in different rooms. In Table 1 we observe that for most rooms, a very high percentage of information can be retained in only one component.

| | co2 | hum | temp | lsr | motr | sndr |
|---|---|---|---|---|---|---|
| **co2** | 1 | 0.07 | 0.411 | 0.185 | 0.326 | 0.208 |
| **hum** | 0.07 | 1 | -0.04 | -0.124 | -0.087 | -0.049 |
| **temp** | 0.411 | -0.04 | 1 | 0.378 | 0.467 | 0.291 |
| **lsr** | 0.185 | -0.124 | 0.378 | 1 | 0.531 | 0.259 |
| **motr** | 0.326 | -0.087 | 0.467 | 0.531 | 1 | 0.536 |
| **sndr** | 0.208 | -0.049 | 0.291 | 0.259 | 0.536 | 1 |

Figure 2: Pearson correlation between training features

The next step was finding a suitable neural network architecture for the data set. The NAS tested more than 2000 models varying hyperparameters and number of layers or nodes per layer. The architecture search showed that the dimensions of the neural network and training batches had little importance. Still, some models performed better than others. The result in Figure 4 shows the relation between prediction and truth predicted by the best-performing LSTM model(GRU performed similarly with equal parameters) on the test set(10% of the data). This architecture consist of a $16 \times 6$ input vector where the former represents time steps and the latter training features. 4 recordings per hour was used, such that each training sample contains 4 hours of previous data. The model has 3 hidden layers with shape $(64, 64, 32)$ in which the last layer is densely connected to a single ReLU node. This is thus a regression predicting a single floating number. The network was trained with Mean Squared Error(MSE) as the loss function and Adaptive Moment Estimation(ADAM, a variant of Stochastic Gradient Descent) as optimizer [11]. Dropout was turned off since this was found by searches to lead to worse performance. However, recurrent dropout led to better performance. A feature effectiveness search was performed to investigate how well each feature and all combinations of features trained the network. The shape of the timestep vector remained the same. Here, it is observed that a model trained only on $CO_2$ had great learning potential, and that the other features alone had little or even

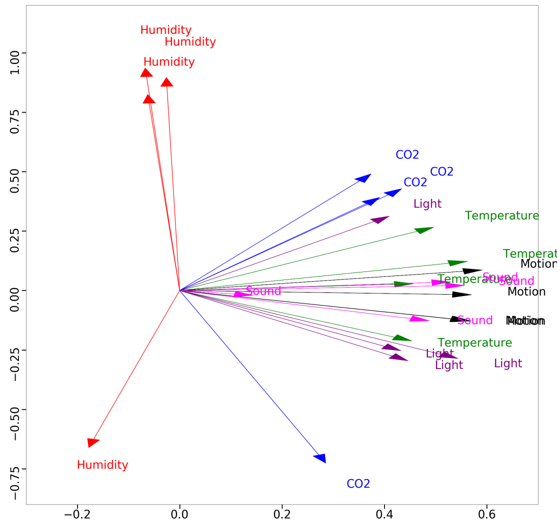


Figure 3: Alignment of raw components with 2D principal components. The same-colored arrows respond to the same components in different rooms

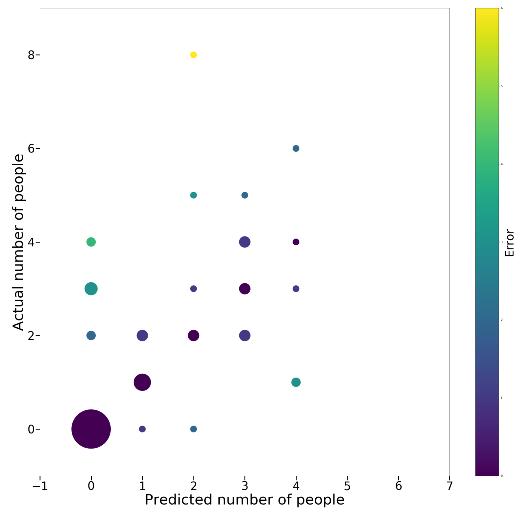


Figure 4: Prediction on the test set the best-performing LSTM model the architecture search converged to

no learning potential. A combination of all features except $CO2$ did however have almost as good potential as $CO2$ alone. At last, a number of tests with different initial parameters and random seeds were performed to test for reproducible results. In our case, for most seeds the network performed as expected, but in rare cases with some seeds the neural network would not train at all. If in such case one is using a callback function to stop a network from over-training by monitoring loss, the network will not train and the data may appear useless. It is observed that the model is able to generalize patterns in the training set, even if the low amount of available training data probably leads to under-performance on accuracy. The mean absolute error from the model in Figure 4 prediction on the test set was at 0.73, which meant 65% accuracy after rounding the prediction output.

# 5 Discussion

The data patterns found in the training samples are hard to analyse due to a large probability of noisy interventions because of the dynamic and unpredictable states the sensors are subject to. While some of the signals clearly contain more seasonality due the nature of the data it monitors, especially $CO_2$, others are a lot more noisy and may only occasionally contain trends, such as sound and light. $CO_2$, humidity and temperature signals are influenced by pre-existing building monitor systems such as ventilation and thermostats.

It is not surprising that the most of the information is contained in the $CO_2$ times series. This is consistent with previous research and our assumptions. Interestingly, during the architecture search a combination of the 5 other features is able to train a model that approaches the performance of a model trained only with $CO_2$ data on models trained from scratch. One could assume that the patterns contained in light and sound measurements would be noisy due to external influences from beyond the room itself. I.e. a solo person would probably not change the sound signal very much. Such is also the case for light, which is not exclusively influenced by indoor lighting, but also from sunlight. The motion sensor returns binary values and as such does not say anything about the actual amount of people, only if there are people or not. However, there was found little difference in training precision between networks that incorporate these features contrary to those that do. If anything, the models using all training features are seemingly more robust. It seems that recurrent neural network models like LSTM and GRU are able to filter the noisy parts of these signals and only use them in cases where they actually have predictive capability.

The same seems to also be true for the temperature feature. Temperature is controlled by thermostats and as such would balance out any human intervention in the heat signal. But this is also a signal that would contribute to the training of a recurrent neural network, since the temperature would first rise, and the thermostat would respond and adjust. These patterns could be present in the sliding time window training samples, depending on the time resolution.

Using human observers to gather ground-truth is costly and the yield is limited. However, attempts to introduce more automated means failed. The room booking system proved to be a very unreliable source for the same purpose. Any effort to let room users systematically share reliable measurements of room use and space occupancy proved very unreliable.

# 6 Conclusion

This research set out to investigate if a causal relationship between a set of sequential data describing various properties of rooms being in regular use by a variable number of people could be determined. Furthermore, if identified, this causal relationship would be used to create a deep learning model that could be used to predict a future number of people depending on the recorded past values of the sensor measurements. The analyses presented indicate strongly that such a causal relation exist, and that predictive deep learning models can be created for this purpose.

# 7 Further Work

Low-resolution thermal cameras with RNNs has proven promising as a non-intrusive method for monitoring the presence or actions of persons [5]. Such data could be used as training labels in the setting this paper describes, or act as a counting/monitoring device on its own.

# References


[1] I. B. A. Ang, F. Dilys Salim, and M. Hamilton. Human occupancy recognition with multivariate ambient sensors. In *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pages 1–6, March 2016.

[2] B. Bremdal, C. Skjerve-Nielsen, and G. Nereng. How the prosumer role in smart grids redefines the energy efficiency concept of buildings. World Sustainable Building Conference SB 2011, Helsinki, October 2011.

[3] K. Cho, B. van Merrienboer, Çaglar Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *ArXiv*, abs/1406.1078, 2014.

[4] J. Chung, Çaglar Gülçehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *ArXiv*, abs/1412.3555, 2014.

[5] A. Danielsen and B. Bremdal. Predicting bedside falls using current context. 12 2017.

[6] T. Ekwevugbe, N. Brown, V. Pakka, and D. Fan. Real-time building occupancy sensing using neural-network based sensor network. In *2013 7th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*, pages 114–119, July 2013.

[7] T. Elsken, J. H. Metzen, and F. Hutter. Neural architecture search: A survey. *J. Mach. Learn. Res.*, 20:55:1–55:21, 2018.

[8] K. Greff, R. K. Srivastava, J. Koutník, B. Steunebrink, and J. Schmidhuber. Lstm: A

search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28:2222–2232, 2017.

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[10] C. Jiang, M. K. Masood, Y. C. Soh, and H. Li. Indoor occupancy estimation from carbon dioxide concentration. *ArXiv*, abs/1607.05962, 2016.

[11] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[12] A. P. Singh, V. Jain, S. Chaudhari, F. A. Kraemer, S. Werner, and V. Garg. Machine learning-based occupancy estimation using multivariate sensor nodes. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6, Dec 2018.

[13] A. T. C. software]. Talos, 2019.