# Hybrid Edge-Cloud based Ensemble Learning for Forecasting Occupancy of Open-plan Offices

Fatemeh Jalali[1], Subhrajit Roy[2], Ramachandra Rao Kolluri[1], Maneesha Perera[1 4], Mahsa Salehi[3], John D. Vasquez[5], Julian de Hoog[1]
[1]IBM Research Australia, [2]DeepMind , [3]Monash University, [4]University of Melbourne, [5]IBM IoT Center

## ABSTRACT

Estimating and forecasting occupancy of open-plan and agile offices is of great value to many businesses, since it allows them to better allocate the necessary space, and save on energy and insurance costs. In this work we use a massive amount of data collected in an open-plan office having a large number of IoT sensors over an extended time period to explore how to best forecast occupancy in an open-plan office. We implement, evaluate, and compare statistical, machine learning, and deep learning forecasting methods, while using as inputs a variety of features including seasonal factors and weather data that affects occupancy patterns. It is shown that the best forecasts can be achieved using a deep learning model (LSTM), trained on a large amount of prior data. However, an ensemble model (composed of seasonal ARIMA and Decision Trees) is likely to be a more practical and reliable solution in hybrid Edge-Cloud scenarios, where less data may be available and privacy is important. In the ensemble approach, data can be split between Edge and Cloud platforms – keeping recent data and light models locally, and sending less recent data to the Cloud only when needed for model retraining.

## 1 INTRODUCTION

The physical environment of workspaces has evolved over the past decades. The shift from private and static space to shared and dynamic space among many enterprises and start-ups has been adapted very quickly due to multiple benefits such as increased collaboration, teamwork, creativity, cost-saving, etc [5]. The recent shift to working from home (due to the Covid-19 pandemic) has pushed many businesses to reconsider their need for static workplaces since it seems that work patterns will continue to evolve in the coming years, and the presence of people in office spaces will likely become ever more dynamic.

In an open-plan and agile workspace setting, employees do not have a dedicated workspace, but instead choose where they want to work each day and even change their workspace during the day. While this flexibility allows employees to choose the space that is best suited to their work, it allows greatly reduced real estate costs for enterprises by maximising space utilisation. The construction cost (for building walls or purchasing cubicles), the energy cost (for lighting, heating and cooling) and the insurance cost are the major cost sources that can be reduced in open-plan workspaces if the space is used efficiently. However, it is possible to under-utilise some parts of the office while still paying for real estate, energy and insurance costs [3, 8].

In order to understand how the space is used by occupants [2, 4, 5, 10] and identify the under-utilised and popular spaces for proper space planning, an extensive data-driven study for occupancy of open-plan offices is required.

In this work, we study the occupancy of an open-plan office equipped with a very high number of IoT-enabled devices and sensors installed. To understand the occupancy of individual desks, IoT devices (in this case a combination of PIR sensor and temperature sensor) are attached under desks to detect the presence of a person through a combination of body heat and motion. The data from these sensors have been collected over two years at a very high data resolution (every 2 minutes on average).

Currently, the data has been mainly used for providing information on the occupancy of the office in near real-time. However, the focus of this work is to use the collected data to forecast daily occupancy of the office. In addition, to have the best model for forecasting occupancy, there is another challenge to overcome in this work which is preserving the privacy of office data while taking advantage of Cloud computing platform for storing data. The purpose of this work is not only to find the model having highest forecast accuracy, but also to understand the relative needs of different forecasting approaches (such as prior data history) and impacts on privacy.

We analyse the motion and temperature data captured by the sensors and evaluate the occupancy of each desk per day and then calculate the daily occupancy of the office. In addition to the data coming from the sensors, external data such as calendar information (public holidays, weekdays, weekends, holiday seasons), weather are gathered over the last two years and added to our data sets.

In order to forecast the daily occupancy of the building, we first apply various individual forecasting models: classical time series models, machine learning models and deep learning models. We find that a SARIMA (seasonal autoregressive integrated moving average) forecast provides the best statistical approach, a DT (Decision Tree) forecast provides the best machine learning based approach, and an LSTM (Long Short-Term Memory) forecast provides the best deep learning approach. Among these three, LSTM provides the overall most accurate forecast and DT the least accurate.

Based on the observations and further analysis, the results show that a DT provides accurate forecasts on the occupancy of the days

The corresponding author's email is: fa.jalali@au1.ibm.com - Subhrajit Roy was with IBM Research Australia during working on this research project.



, ,



ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
10.1145/1122445.1122456

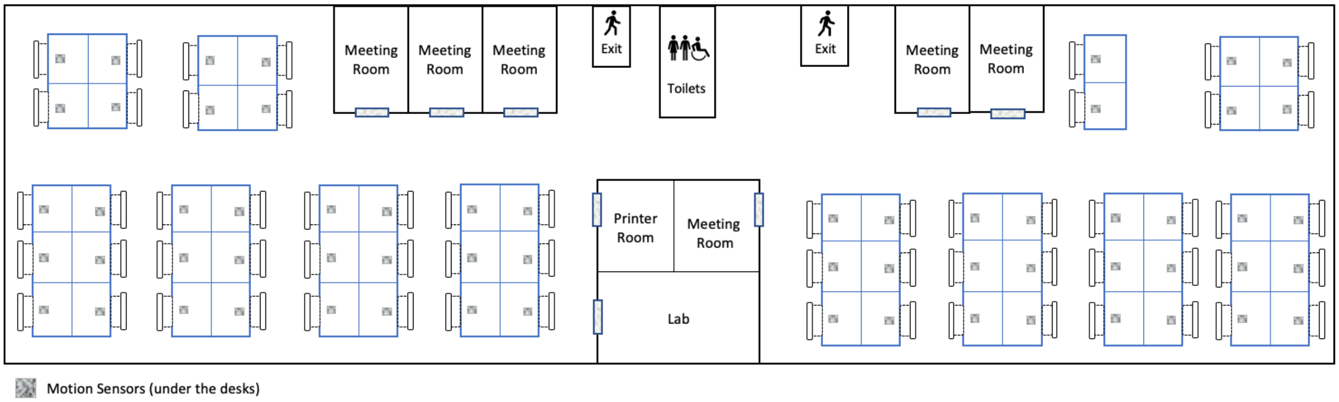


**Figure 1: A simplified diagram showing the type of smart open-plan office environment studied.**

that have special events and occasions although its overall forecast is not very accurate. On the other hand, our analyses show that SARIMA can forecast the occupancy of normal weekdays but not the occupancy of the days influenced by special events and calendar changes. Therefore, it is observed that these two models might be complementary to each other.

To validate whether SARIMA and DT can be combined to forecast the office occupancy, a stacking ensemble learning model is applied to combine the prediction results of the two models. The results indicate that the stacking model not only works better than individual SARIMA and DT models but also performs very close (slightly lower accuracy) to LSTM forecasts. To have a better comparison between LSTM and the ensemble model, both models are examined on various size of training data sets. It is observed that combining the results of two simpler models can yield to a better forecasting result when less data is available compared to the forecast result of a computationally expensive deep learning model such as LSTM that needs long sequences of observations (more data for training).

On the one hand, we want to leverage the benefits of a Cloud computing platform for data storage and retrieval, but on the other, we are conscious of the privacy implications of gathering and storing data of this nature. To address this, we come up with a solution that uses a combination of Cloud and Edge computing (local resources, near to the sensors) platforms for our forecasting model. Our results show that the ensemble forecasting model is the most suitable candidate for a hybrid Edge-Cloud platform as it can be easily divided into two processing parts for light and heavy computation models. The lighter model (ARIMA or SARIMA) can be run locally on a small resource such as a Raspberry Pi with recent 4-5 week (1 month) of data. While the heavy computation models (DT or LSTM) can run on the Cloud with a large amount of historical data (2 years). We propose a privacy-aware forecasting model for open-plan offices through this approach of holding recent data (4-5 weeks) locally and not sending the whole forecast model to the Cloud.

The proposed Hybrid Edge-Cloud platform for forecasting open-plan office occupancy allows businesses to better anticipate their office space and energy needs, and reduce their ongoing energy and insurance costs while preserving the privacy of building and occupants.

## 2 CASE STUDY

In this section, the details of the building map, sensors and datasets are described, and the architecture behind IoT applications and services is explained.

### 2.1 Office Description

Our case study is an open-plan office with no walls or cubicles for individuals (as shown in Figure 1). There are a few meeting rooms, but we do not study those rooms in this work as the daily occupancy of the office does not change based on the occupancy of the meeting rooms.

This office has 60 desks for employees and visitors. All desks have motion sensors and temperature sensors which are installed under the surface of the desk, close to the human body for detecting movements and temperature, to identify the presence of a person.

The employees of this office have the option to work remotely sometimes and their working hours are also flexible. Therefore, the occupancy of this office varies a lot and the prediction is not straight forward. We believe that the problem of forecasting occupancy in an office space of this nature may therefore, in fact, be more difficult than forecasting occupancy of offices having more regular occupancy patterns. As a result, we anticipate that the results of this study could be directly applicable to a wide range of open-plan office type environments.

### 2.2 Sensors

The whole office is equipped with several IoT devices and sensors in different areas: under each desk (Yanzi Presence Mini [1]), on the ceiling (Yanzi Motion+ [1]) and meeting room walls (Yanzi Comfort [1]). Under each individual desk, Yanzi Presence Mini sensor is mounted directly onto the surface to capture motion and temperature data. In each zone of the building (comprises four to six desks), a Yanzi Motion+ sensor is installed on the ceiling to detect motion, temperature, humidity, ambient light and sound pressure of the zone. Inside the meeting rooms, Yanzi Comfort

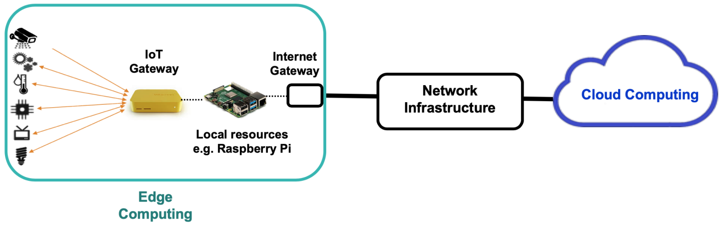


**Figure 2: A simplified architecture behind IoT applications with both Cloud and Edge resources**

sensors are mounted to the walls to monitor air quality, temperature, humidity, barometric pressure and ambient noise. Yanzi Comfort sensors measure levels of carbon dioxide (CO2) and volatile organic compounds (VOC). All these sensors are wirelessly connected to their closest IoT gateway (e.g. Yanzi Gateway 2 [1]) and send the data to the cloud.

### 2.3 Data Sets

Each sensor is individually timestamped. For example, for each Yanzi Presence Mini -installed under the desks- two data sets are saved, one data set for motion data and one for temperature measurements. The same approach is used for collecting various types of data from Yanzi Motion+ and Yanzi Comfort sensors.

For two years of data (from 7 June 2017 to 4 June 2019 - 727 days), each data set has between 600,000 to 800,000 rows of readings which means the data from each sensor is captured every 1.5 to 2 minutes on average. This level of data acquisition enables us to analyse the data across different data resolutions such as every few minutes, hours, days, etc. In this work, the raw data from motion sensors are transformed to daily occupancy data using a Hidden Markov Model (HMM). The train data set has 75% of the whole data set (545 days) and the test set has 25% (182 days). Currently, the data from sensors is used as live data for real-time actions and decision making and is also saved as historical data. This data set continues to grow over time.

In addition to gathering data from sensors inside the building, the data set is enriched with some exogenous and context variables to provide better interpretation and subsequently improve forecasting accuracy. The weather data of the location has been gathered in order to study the impact of weather on office occupancy. The weather data includes minimum temperature, maximum temperature, and weather status (rain, snow, etc).

### 2.4 System Architecture

Currently, a majority of Internet of Things (IoT) applications have used Cloud computing which alludes to large storage and computation resources located in geographically distant locations from the IoT local network. Cloud platform is popular for its mobility, scalability and on-demand computing amongst features. However, some IoT applications cannot be efficiently served by the Cloud due to restrictions such as latency, bandwidth, privacy and etc [9]. To serve these applications, Edge computing is introduced which alludes to local storage and computation resources close to the IoT devices, within the IoT local network [7, 9]. Figure 2 shows a simple architecture behind IoT applications with both Cloud and Edge resources.

## 3 MODELS & RESULTS

In this section, various classical time series, machine learning and deep learning models applied on the data are explained and the results are discussed. Next, we discuss the reasons for using ensemble learning models. In order to evaluate the performance of the models and to assess the accuracy of the predictions, Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) are chosen in this work as error metrics.

### 3.1 Time series models

ARIMA (Autoregressive Integrated Moving Average Model) and SARIMA (Seasonal ARIMA) are two popular time series forecasting approaches that have been used in many forecasting applications from economics to vehicle traffic systems [6].

*3.1.1 ARIMA:.* ARIMA is a generalised model of Autoregressive Moving Average that combines Autoregressive (AR) process and Moving Average (MA) processes and builds a combined model of the time series. The ARIMA forecasting model is written as:

$$y_t' = c + \overbrace{\phi_1 y_{t-1}' + \ldots + \phi_p y_{t-p}'}^{\text{AR}} + \overbrace{\theta_1 \epsilon_{t-1} + \ldots + \theta_q \epsilon_{t-q} + \epsilon_t}^{\text{MA}} \qquad (1)$$

where, $y_t'$ is the differenced series. The predictors on the right-hand side include both lagged values ($\phi$) of $y_t'$ and lagged errors ($\theta$). $\epsilon_t$ is white noise. This model is shown as **ARIMA**$(p, d, q)$, where $p$ is

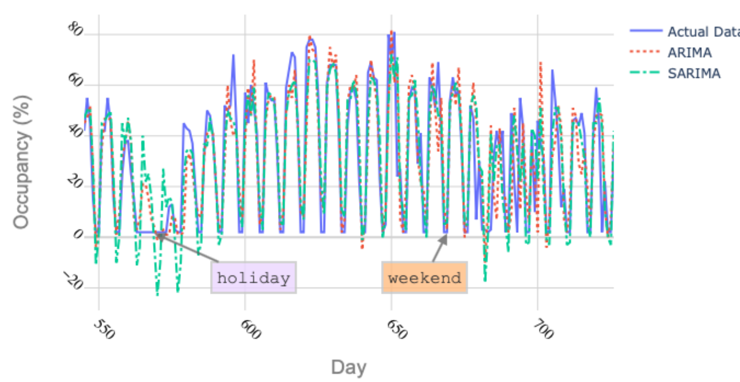


**Figure 3: Occupancy forecast using ARIMA (red dash line) and SARIMA (green dotted-dash line) vs. actual data (blue line).**

order of the autoregressive part, $d$ is degree of first differencing involved and $q$ is order of the moving average part.

Following a parameter sweep based analysis, $ARIMA(7, 1, 2)$ achieved the highest forecasting accuracy. It indicates the lag value is set to 7 for autoregression and uses a difference order of 1 to make the time series stationary, and finally an order of 2 is used for moving average window. The RMSE, MAE, MAPE of $ARIMA(7, 1, 2)$ model on our data set for one step ahead forecast is 14.8, 10.4 and 156 which are listed in the first row of Table 1. The result of the ARIMA forecast on the test data set (182 days - 546th day to 727th day) is shown in Figure 3 (red dotted line) and one weekend (2 days of low occupancy - which the pattern repeated in each weekly cycle) and one holiday (a few days of low occupancy) are marked.

*3.1.2 SARIMA:.* SARIMA (Seasonal ARIMA) is an extension of ARIMA that explicitly supports univariate time series data with a seasonal component. It adds three new hyperparameters to specify the autoregression (AR), differencing (I) and moving average (MA) for the seasonal component of the series, as well as an additional parameter for the period of the seasonality. It is written as follow:

$$\textbf{SARIMA}\ \overbrace{(p, d, q)}^{\text{Trend}}\ \overbrace{(P, D, Q)_m}^{\text{Seasonal}} \tag{2}$$

where, $p$ is the trend autoregression order, $d$ is trend difference order, and $q$ is trend moving average order. $P$ is seasonal autoregressive order, $D$ is seasonal difference order, $Q$ is seasonal moving average order and $m$ is the number of time steps for a single seasonal period.

Since there is a seasonal pattern in the data, SARIMA model is applied to the data. The best combination of hyperparameters to forecast occupancy is calculated as $SARIMA(0, 1, 2)(1, 0, 2)_7$. The RMSE of SARIMA is 13.1 which means a small improvement compared to ARIMA model. The results of SARIMA forecasts versus the actual data and ARIMA model are shown in Figure 3 in a green dash-dotted line.

Based on the results from MAPE metrics (a very high number for ARIMA and SARIMA since MAPE is more sensitive to wrong predictions around low occupancy) and observations from Figure 3, ARIMA and SARIMA models could not accurately forecast the occupancy of the days which have events such as holidays, special occasions, building shut down periods and even sometimes weekends. ARIMA models assume a standard relationship between current values and lagged values in the data. However, for unusual days (such as holidays, special occasions, etc.) these relationships do not hold. If exogenous variables are not accounted for, ARIMA based models are likely to perform poorly for these days.

## 3.2 Classical Machine Learning

Various machine learning such as Random Forest, Extremely randomized Trees (ExtraTrees), Support Vector Regressor (SVR) and Decision Tree (DT) are applied to study the forecast of the office occupancy. We only discuss the results of the DT model in this sub-section since its RMSE result is better than other models, and since it could accurately predict the occupancy of the days/periods with special events and occasions.

A Decision Tree (DT) is a tree-like graph where nodes represent condition statements, edges represent answers to those condition statements, and leaves represent actual output or class label.

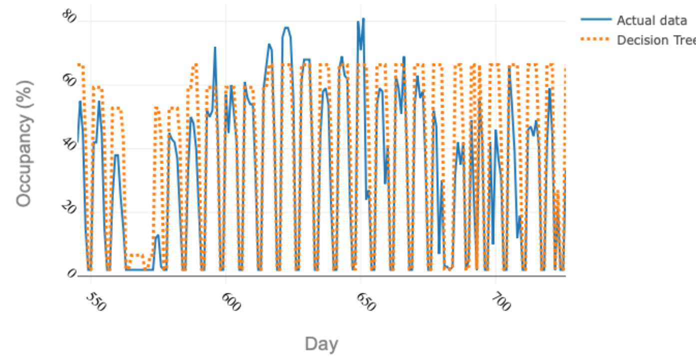


**Figure 4: Occupancy forecast using DT (orange dash line) vs. actual data (blue line).**

The goal is to create a model that forecasts the value of a target variable (i.e. occupancy in this work) by learning simple decision rules inferred from the data features (i.e. day of week, holiday season, public holidays, weather, etc.).

Regression DT is applied and hyperparameters are carefully tuned (e.g. max depth=4) using optimisation to avoid over-fitting and under-fitting the model. The result of the DT model as shown in Figure 4 indicates that the model forecasts office occupancy very well on special days such as weekends, public holidays and holiday seasons. The model shows features such as public holidays and weekends, holiday seasons, some days in a week (Friday) have a significant impact on the occupancy forecast while weather condition has less impact. The RMSE of this model is 14.8 with no improvement compared to SARIMA whereas the MAPE is improved a lot compared to SARIMA (as listed in the third row of Table 1). The huge improvement of MAPE is due to a better forecast for days with low occupancy (e.g. weekends, holidays).

## 3.3 Deep Learning

The Long Short-Term Memory (LSTM) recurrent neural network is a promising solution for forecasting time series data by learning over long sequences of observations.

Forecasting occupancy using neural networks is carried out using a sequential model that consists of 4 stacked layers followed by a few densely connected layers. Known features over the forecasting period of interest are provided in training with a masked label. The forecast result on the test set is shown in Figure 5 and the results of RMSE, MAE, MAPE metrics are listed in the fourth row of Table 1.

**Table 1: RMSE, MAE, MAPE of all the models**

| | RMSE | MAE | MAPE |
|---|---|---|---|
| **ARIMA** | 14.8 | 10.4 | 156 |
| **SARIMA** | 13.1 | 9.5 | 180 |
| **DT** | 14.8 | 9.4 | 53 |
| **LSTM** | 10.5 | 6.5 | 31.8 |
| **DT (SARIMA)** | 11.7 | 7.6 | 44 |
| **Ensemble (DT-SARIMA)** | 11.2 | 7.2 | 38 |

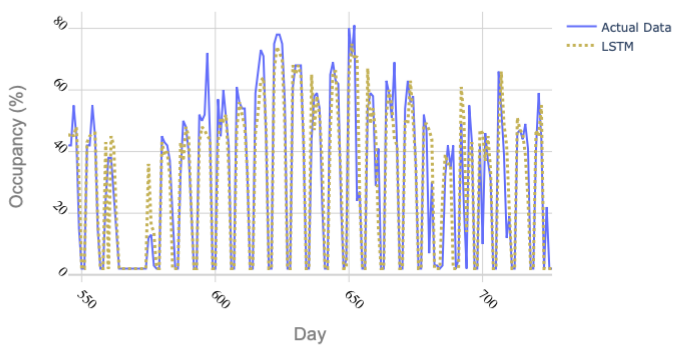


**Figure 5: Occupancy forecast using LSTM (gold dash line) vs. actual data (blue line).**

The initial results show that LSTM can forecast the occupancy more accurate than other individual models.

## 3.4 Ensemble Learning Models

Ensemble learning models combine the decisions from multiple learning models to improve the overall performance. This can be achieved in various ways. Since the results of SARIMA and DT are complementary and do not heavily overlap, the best ensemble method to be used is stacking.

Stacking works in two steps: (1) multiple base learners are used to predict the initial results; (2) a new learner is used to combine the predictions from the base learners. Three experiments are studied. In the first experiment, we only calculate the prediction of SARIMA model on the whole data set (train and test) and then add the result as a new feature to DT model. The results of the new DT with SARIMA feature is shown in Figure 6. The RMSE of this model is 11.7 which is better than both DT and SARIMA models.

In the second experiment, the occupancy is predicted using a new learner applied to the results of SARIMA and DT. For the second leaner, linear regression and DT are used as shown in Figure 7 and the RMSE results (12.72 for linear regression and 11.2 for DT) indicate that DT as the second learner for the staking ensemble can improve the occupancy forecast. The RMSE, MAE and MAPE results of these two ensemble models are listed in the fifth and sixth rows of Table 1. In the third experiment, a stacking ensemble model on LSTM, SARIMA and DT is used to forecast the occupancy but there was no improvement on the results.

# 4 DISCUSSION

The initial results from the previous section indicated that the LSTM's model performance on the test set when we use the whole training data set (545 day≈78 weeks) is the highest. To decide on the final model for our system, we investigate the performance of all the models on different size of training data sets and study the use of hybrid Edge-Cloud architecture for improving the performance as well as the privacy of the data and model.

## 4.1 Size of Training Set

In this subsection, the performance of SARIMA, DT, LSTM as well as the ensemble model (the stacking model on SARIMA and DT) on

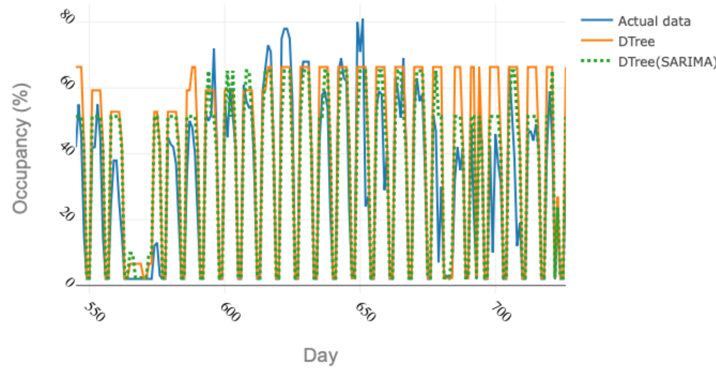


**Figure 6: Occupancy forecast using DT with SARIMA as a new feature (green dash line) vs. actual data (blue line) vs. DT (orange line).**

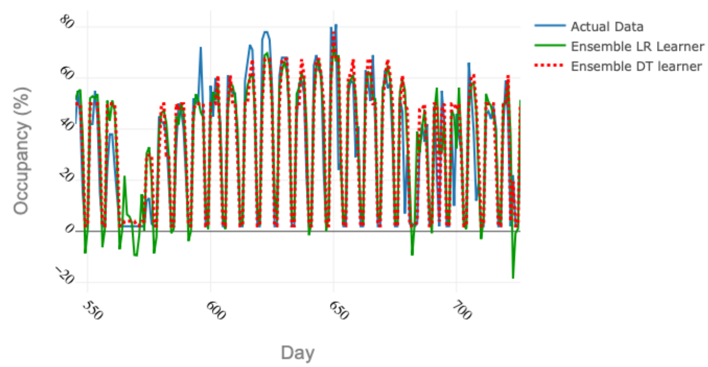


**Figure 7: Occupancy forecast using stacking ensemble models with DT learner (red dash line) vs. linear regression learner (green) vs. actual data (blue line).**

the same test set (182 days) are studied when the models are trained on 2 weeks of recent data to 78 weeks of training data sets (whole training set). The results in Figure 8 reveal that the performance of SARIMA model (blue dotted line) improves over time however after 4-5 weeks of data the improvement is very small.

It seems the performance of the DT model ( pink dotted line) decrease over time since we have used the recent data for smaller training sets and the recent data is a better representation of the occupancy of the test set in normal days. However, the use of DT is mainly for the special days (not all days) and as we train DT on more data, the model can learn how to forecast the special days more accurately. In order to explain how chronological order in this data set impacts the overall performance of the DT, the chronological order of the training set is removed when training the DT model and the results reveal that the performance of DT improves as the size of training set increases as shown by light grey line in the figure. The results from SARIMA and DT are complementary which means SARIMA is a suitable model to forecasting occupancy of normal days whereas DT is a better model for forecasting special days. When the results from these two models are combined (ensemble

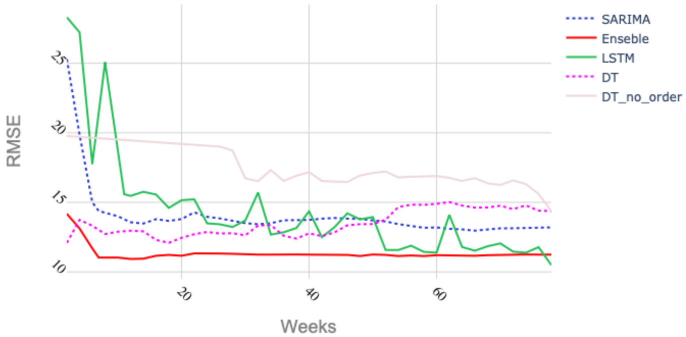


**Figure 8: Forecast error vs. the size of training data set.**

model), a better forecasting result is achieved, as indicated by the red line in the figure. For the LSTM model, the forecast improves over time when more data is used to train the model as shown by the green line. Although LSTM does a better forecast when the whole training data set is used, it does a poor performance when less data is available. Comparing the LSTM and ensemble models, it seems the forecasts using ensemble model are consistently low when the training data available is recent and its amount is low (4-5 weeks). Therefore, our conclusion is to use the ensemble model since it works well when less training data is available and there is no huge difference between its performance versus LSTM when the whole training data set is used.

## 4.2 Hybrid Edge-Cloud Architecture

Similar to the majority of IoT applications, the data generated from the sensors in this office are sent to the Cloud for processing. All the proposed models (discussed in the previous subsections) can be easily run in the Cloud and forecast the occupancy of the next day. However, in some cases there can be significant concern over the privacy implications of sending such types of data to the Cloud, in particular for very recent data. One solution that can address this issue is to store data only locally, and do the training plus scoring forecasts only at the Edge, with no Cloud interaction at all, but it is not always a practical solution for many customers since they want to use of the power of the Cloud.

In fact, there is a trade-off between taking advantage of the Cloud storage and resources and privacy of the data. However, if the customers would like to take advantage of the Cloud and store their data in a secure and low maintenance location but preserve the privacy of the data to a certain level, it is achievable by sending the historical data to the Cloud and keeping the recent data locally at the Edge in a low-cost computation resource such as a Raspberry Pi. It is also possible to breakdown the forecast model on hybrid Edge-Cloud platform instead of running the whole model from the Cloud to preserve security and privacy of the forecast model.

In order to breakdown the forecast model to the hybrid Edge-Cloud platform, it is important to keep the lighter models that need less computation and limited data to be trained locally and send the heavy computation models to the Cloud. The reason is that normally less computation and storage are available at the Edge as discussed in Section 2.4. Therefore, for such privacy-aware forecasting, the ensemble learning model is the best model since the SARIMA model can be easily run on a small device such as Raspberry Pi and the data of recent 4-5 weeks would be enough for forecasting while the historical data and the DT model, that requires more data to learn about special days, can run on the Cloud. This model not only provides the most accurate forecast but is also the best privacy-aware forecast model by running on the hybrid Edge-Cloud platform.

# 5 CONCLUSION

To understand the occupancy of an open-plan office and adopt an efficient space and energy management, it is required to study the occupancy of each desk extensively over time. This is possible by gathering the data continuously from the environment using sensors and IoT devices. In this work, new data-sets, which came from an open-plan office data collection over 2 years, are studied and analysed to forecast its daily occupancy.

Multiple forecasting methods such as time series, machine learning and deep learning models are applied to the data-set and the results highlight that forecasting occupancy is not trivial, and extra information and multiple features are required to improve the accuracy. However, by increasing the features, it may not be possible for one single model to capture the occupancy pattern. Therefore, ensemble models that combine the forecasts from various models shown to be effective. It is also important to find models whose combinations work well together, and whose predictions complement each other. Our results reveal that a combined model of seasonal ARIMA and Decision Trees not only generates most reliable occupancy forecast on small training set sizes but also promotes a privacy-aware forecast model when it runs on a hybrid Edge-Cloud platform.

# REFERENCES

[1] Yanzi sensors. https://www.yanzi.se/solution/. Accessed: 2019-09-13.

[2] Capozzoli, A., Piscitelli, M. S., Gorrino, A., Ballarini, I., and Corrado, V. Data analytics for occupancy pattern learning to reduce the energy consumption of hvac systems in office buildings. *Sustainable Cities and Society 35* (2017), 191 – 208.

[3] Chang, W.-K., and Hong, T. Statistical analysis and modeling of occupancy patterns in open-plan offices using measured lighting-switch data. *Building Simulation 6*, 1.

[4] Golestan, S., Kazemian, S., and Ardakanian, O. Data-driven models for building occupancy estimation. In *e-Energy* (2018).

[5] Howard, J., and Hoff, W. Forecasting building occupancy using sensor network data. In *Proceedings of the 2Nd International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications* (New York, NY, USA, 2013), BigMine '13, ACM, pp. 87–94.

[6] Hyndman, R., and Athanasopoulos, G. *Forecasting: Principles and Practice*, 2nd ed. OTexts, Australia, 2018.

[7] Jalali, F., Lynar, T., Smith, O. J., Kolluri, R. R., Hardgrove, C. V., Waywood, N., and Suits, F. Dynamic edge fabric environment: Seamless and automatic switching among resources at the edge of iot network and cloud. In *2019 IEEE International Conference on Edge Computing (EDGE)* (July 2019), pp. 77–86.

[8] Salimi, S., Liu, Z., and Hammad, A. Occupancy prediction model for open-plan offices using real-time location system and inhomogeneous markov chain. *Building and Environment 152* (2019), 1 – 16.

[9] Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., Kong, J., and Jue, J. P. All one needs to know about fog computing and related edge computing paradigms: A complete survey. *Journal of Systems Architecture 98* (2019), 289 – 330.

[10] Zhang, T., and Ardakanian, O. A domain adaptation technique for fine-grained occupancy estimation in commercial buildings. In *Proceedings of the International Conference on Internet of Things Design and Implementation* (New York, NY, USA, 2019), IoTDI '19, ACM, pp. 148–159.