AT-23-C079 Published in ASHRAE Transactions, Volume 129, Part 1

Short-Term Occupant numbering Prediction Via Machine Learning Approaches

Zixin Jiang
Student Member ASHRAE

Bing Dong*, PhD
Member ASHRAE

ABSTRACT

Occupancy behavior plays an essential part in smart building operation. Developing an appropriate algorithm to predict occupancy information will bring a better control for Heating Ventilation & Air Conditioning system, and indoor health. However, due to the strong stochasticity of occupancy behavior, it is much harder to predict occupant count than occupant state. There is a lot of studies working on occupancy presence or arrive-departure time prediction, only a few researchers focus on the occupant count prediction. The lack of occupant count prediction limits the development of demand-controlled ventilation. In this study, 1) A set of ground truth data was collected via state-of-the-art people counting sensor. 2) A flatten preprocessing method was used to smooth the collected data of occupant number. 3) Seven different models (ARMA_ANN model, RNN model, LSTM model, Nonhomogeneous Markov with change point detection model, XGBoost model, Random Forest model and ANN_Range model) were used to predict the room occupant count from 15 minutes to 24 hours ahead. We found that XGBoost model, Random Forest model and ARMA_ANN model have similar performance and they all outperforms than the other models by a 3% to 13% mismatch rate reduction and reduce the computation time. Each model could predict the number of occupants with 85% accuracy with one-person offset and the accuracy for 15 minutes ahead prediction could reach 95% with one-person offset. LSTM model works slightly better than RNN model and both of them had a smoother prediction. None of these seven models could track the abrupt changes.

INTRODUCTION

Buildings account for more than 30% of total energy consumption in the world [Pérez-Lombard. 2011]. Among them, the lighting operations, consumes around one-third of the total energy in commercial buildings [Cooper, S. 2004] and the Heating, Ventilation & Air-Conditioning (HVAC) system is another major consumer, which accounts for approximately 40% of total energy usage [Yang, L. 2014]. These numbers are still increasing rapidly because people spend more time in buildings due to the pandemic and the HVAC system is operating longer with higher outdoor flow rate required by governments during COVID-19 [Zheng, W. 2021] [Guo, M. 2021]. Occupancy has becoming one of the most important factors influencing building energy consumption [Mirakhorli, A. 2016], which is playing a vital role in building energy management [Hong, T. 2017].

People spend more than 80% of their time in buildings [Klepeis, N. 2001], their presence, movement, and behaviors have significant influence for building energy consumption [D'Oca, S. 2018]. Occupancy can be used for demand-controlled ventilations. Wang et al. [Wang, J. 2021] proposed an intelligent ventilation control strategy, which could calculate the demand ventilation rate by occupancy level in real-time. The case studies showed that 11.7% of energy could be saved while controlling the probability of infection below 2%. Mokhtari et al. [Mokhtari, R. 2021] investigated the impact of occupancy pattern, air exchange rate, and class duration time of a university building by multi-objective optimization. The results concluded that a uniform population distribution could reduce the infection cases

Zixin Jiang is a PhD student in the Department of Mechanical and Aerospace Engineering, Syracuse University, Syracuse, New York. **Bing Dong** is an associate professor in the Department of Mechanical and Aerospace Engineering, Syracuse University, Syracuse, New York. Built Environment Science and Technology (BEST) Laboratory bestlab.syr.edu.

by 56% and the energy consumption by 32%. Schibuola et al. [Schibuola, L. 2021] designed a CO2-based air change hour (ACH) calculation method to trigger the ventilation control for reducing the reproduction number and the infection risk. The energy-saving could reach 60% and 72% by using a high-efficiency air handling unit without and with a thermal recovery system, respectively. Occupancy can be used for model predictive control. Dong et al. [Dong, B. 2014] designed a nonlinear MPC model based on occupancy prediction. The experiment showed a 30.1% energy reduction to measurement in the heating season and 17.8% in the cooling season. Mirakhorliet et al. [Mirakhorli, A. 2016] reviewed occupancy behavior-based model predictive control for building systems. The results showed that MPC could save 10% energy compared with the rule-based control; and occupancy-based MPC could save more than 20%. Occupancy can be used for occupancy central control. Kong et al. [Kong, M. 2022] implemented a side-by-side occupancy-based control experiment. The results demonstrated occupancy-based control can maintain good thermal comfort and save the energy up to 17 and 24% weekly. Occupancy can be used for smart lighting control. Zou et al. [Zou, H. 2018] designed a smart lighting control system driven by occupancy state estimated by WIFI signal. The results achieved 93.09% and 80.27% energy savings.

However, due to the strong stochastic characteristic of occupancy behaviors, only about 10% of researchers have considered the occupancy information in building load predictions [Sun, Y. 2020]. The most commonly used control strategy in HVAC system is still rule based control. And the occupancy schedule is predefined in most of standard documents or policy guideline. This simplification may sometime drive the HVAC system away from the actual situation. A predefined schedule always overestimated the energy consumption and oversize the HVAC plant, which may operate the system with low efficiency and waste energy [Yang, Y. 2022]. Therefore, occupancy prediction is the major obstacle to develop the optimized control policy [Mirakhorli, A. 2016]. In previous study, occupancy prediction can be further divided into occupancy state prediction (occupied or unoccupied); occupant count prediction and occupancy distribution prediction. For occupant count prediction, the widely used prediction algorithms are Random Forest, Recurrent neural network / Recurrent neural network with long short-term memory units, Hidden Markov model [Jin, Y. 2021]. The commonly used input features are 1) historic data, 2) contextual information such as time of a day, day of a week, outlook calendar and 3) environment parameters such as temperature, CO_2 level, humidity, WIFI signal, RFID signal, plug load and so forth. Wang, W et al. [Wang, W, 2018] proposed a WIFI based Markov feedback recurrent neural network (M-FRNN) algorithm to model and predict the occupancy. The results have an 80.9%, 89.6%, and 93.9% accuracy with a tolerance of 2, 3, and 4 occupants respectively. Liang, X et al. [Liang, X, 2016] proposed a machine learning based occupancy prediction method used in an office building, which has mean average error lower than 10. Li, Z et al. [Li, Z, 2017] designed an inhomogeneous Markov model based on change point analysis, which could predict occupant count with 0.34 RMSE and 0.23 MAE. Salimi, S et al. [Salimi, S, 2019] built an occupancy prediction model based on Markov chain and has an 86% to 68% accuracy for the purpose of the lighting and HVAC systems control, JoyDutta et al. [JoyDutta, 2022] implemented CatBoost model in occupancy detection and prediction, which achieved 99.85% accuracy for occupancy detection and 92.9% for count prediction.

However, most occupancy forecasting studies focused on presence prediction, like first arrive time and last departure time. There is still a research gap in occupant count prediction. This paper firstly collected occupancy data from a depth radar-based sensor, which doesn't have invasion of privacy problem. Then we used a flattened data preprocessing to clean the noise data. And finally, seven prediction models were evaluated by mismatch rate.

METHODOLOGY

The overall methodology framework of this paper is shown in **Figure 1**. One occupancy sensor was installed on the door framework and the occupancy data was collected every 15 minutes. A python API is called to request the data which then is stored in pgAdmin database. Then seven different occupant count prediction models were trained based on a data flatten preprocessing method. Finally, the results were evaluated by mismatch rate index.

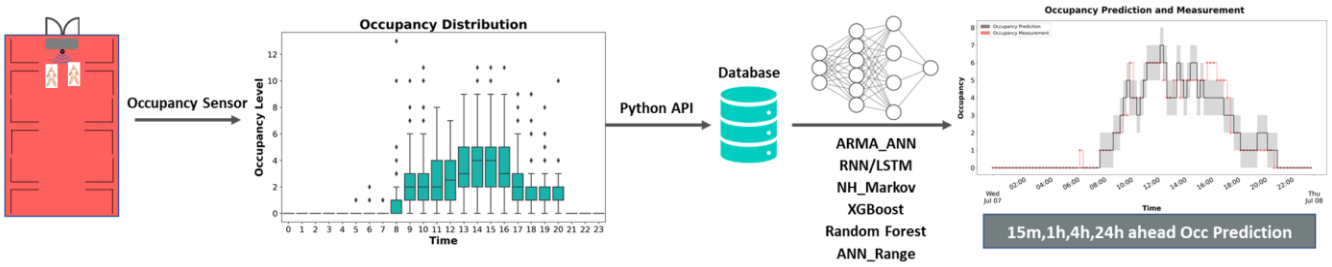


Figure 1 Framework for occupant count prediction.

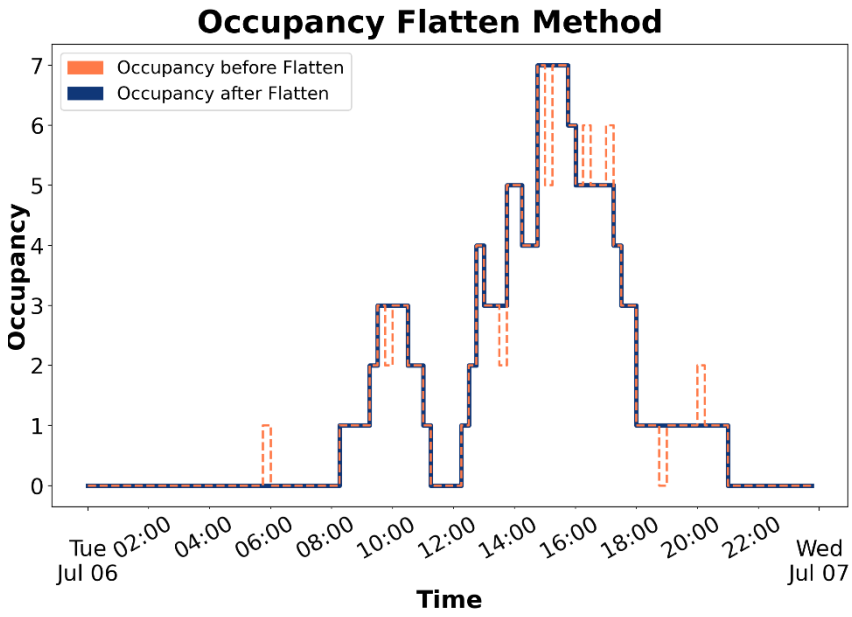
Data Collection

In this study, occupancy data were collected in Link Hall 381A, a student office at Syracuse University (SU), New York, USA. There were up to 12 PhD student working here and the office is normally occupied in the weekday and unoccupied in the weekend. The first arrival time always starts from 8:00AM in the morning and the last departure time was approximately 18:00PM in the afternoon. The occupancy data were collected from one depth radar-based sensor. The sensor was installed on the door framework and can detect how many people entered the space and how many people exited. The sensor would auto reset to zero at 4:00AM everyday to preventing miscounting from happening in the daytime. The occupancy data were collected every 15 minutes from March to July 2020 and then stored to a local database via Python API.

Data Preprocessing

After collecting the occupancy data, we did data cleaning to drop the outliers caused by the internet issue or system error. If the missing data was within one hour (short time, only several points), we would calculate the average of two closest data points then rounded to the closet integer; if the missing data exceeded one hour (long time, a lot of points), we would use the average of the same day in other weeks and then rounded to the closet integer. Furthermore, due to the stochastic and complex characteristics of the occupancy behaviors [Dong, B. 2018], it was unpredictable for a temporal departure like an emergency meeting, a phone call, or going to the restroom. For the HVAC system, it was also unnecessary to adjust the operation for such a small and temporary occupancy variation. Thus, a flatten method was used to process the temporal occupancy variation within 15 minutes as shown in **Figure 2**(a).

Most occupants came in the morning and left in the afternoon. We supposed that the occupancy level was more likely to increase in the morning, fluctuate in the noon, and drop in the afternoon. For example, it was considered that the occupancy level would increase by one if the occupancy sensor detected the arrival of an occupant, but it would drop by one only when the occupancy sensor detected two continuous leaving in the morning. To do that, firstly, the collected data were divided into morning (7:00-13:00), afternoon (13:00-16:30), evening (16:30-23:30), and midnight (23:30-7:00); secondly, for each period, temporal changepoint was modified by pseudocode shown in **Figure 2** (b).



(a)

Data Flatten Algorithm

If Morning **do**
  If Occ(T)> Occ(T-1) **do**
    State(T) +=1
  If Occ(T)< Occ(T-1) and Occ(T+1) <Occ(T-1) **do**
    State(T) -=1
If Afternoon **do**
  If Occ(T)>Occ(T-1) and Occ(T+1)>Occ(T-1) **do**
    State(T) +=1
  If Occ(T)<Occ(T-1) and Occ(T+1) <Occ(T-1) **do**
    State(T) -=1
If Evening **do**
  If Occ(T)>Occ(T-1) and Occ(T+1)>Occ(T-1) **do**
    State(T) +=1
  If Occ(T)<Occ(T-1) **do**
    State(T) -=1
End

(b)

Figure 2 (a) Flatten method for occupancy preprocessing. (b) Flatten method Pseudocode.

Model development

After the data preprocessing, seven occupancy prediction models were established - ARMA_ANN Model (Autoregressive moving average Artificial neuron network model), RNN Model (Recurrent neural network), LSTM Model (Long short-term memory neuron network model), NH Markov Model (Non-homogeneous Markov with change point detection), XGBoost model, Random Forest model and ANN Range Model. Among them, ARMA_ANN model and NH Markov Model were state-of-the-art methods in occupancy prediction, RNN/LSTM was widely used in time series data prediction, XGBoost model and Random Forest model have shown good performance with tabular data. All the programs were completed in Python 3.9, and all the deep learning-based models were finished by Pytorch library, XGBoost model and Random Forest model were completed by "xgboost" and "sklearn" package.

ARMA_ANN Model. Artificial neural network (ANN) was widely used for load forecasting and occupancy prediction. Feature selection played a vital role in model performance, and in this study an ARMA model was used for ANN feature selection. We did autocorrelation factor (ACF) and partial autocorrelation factor (PACF) analysis to find the term order for AR model and a MA model. And in this study, the last two terms were most relevant with the next step's occupancy level. So, in this case, last two steps occupancy were the input for the ANN model. The hour of a day and day of a week were another two input features. Historical occupancy data were normalized into (-1,1), time data were normalized by sine and cosine functions which could eliminate the sudden jump which happened on 24:00 and Sunday. Overall, the input features are hour of a day, day of a week, Occ(T), Occ(T-1), Occ(T-2) and the output target is Occ(T+n), n is the prediction horizon.

RNN and LSTM Model. Compared with the ARMA_ANN model, a recurrent neural network (RNN) could memorize the inputs from previous steps and share the parameters for different time steps. This characteristic made the RNN perform better in the prediction of time-series data. However, the recurrent connection of the RNN model increased the difficulty for model tuning and another disadvantage of the RNN model was gradient vanishing. To overcome these drawbacks, a clipping method was used in this study to stabilize the training process. Moreover, the

long short-term memory (LSTM) model could control the input and output of the RNN by an input gate, an output gate, and a forget gate. These gates could determine how long and how much information might be kept in the cell, which could handle the gradient vanishing problem in RNN.

Non-homogeneous Markov Model. The Markov possibility transform matrix is changing overtime, so we used a non-homogeneous Markov model with a change point detection method. Firstly, an occupancy distribution would be estimated by the weighted average of the data in three previous days, as shown in Eq. ($\boldsymbol{O}(T+1) = \frac{\sum_{n=0}^{z} \lambda^n \boldsymbol{O}(T-n)}{z}$ (1).

$$\boldsymbol{O}(T+1) = \frac{\sum_{n=0}^{z} \lambda^n \boldsymbol{O}(T-n)}{z} \quad (1)$$

Where O was the occupancy level for a whole day, and λ was a forgetting factor less than 1. This factor could adjust the weights of the occupancy distribution estimation. The Pearson divergence was calculated to find the possible change points based on this estimated occupancy pattern. Secondly, the transition matrix within each change point was calculated by Eq. ($p_{ij} = \frac{n_{ij}+\alpha}{\sum_{k=0}(n_{ik}+\alpha)}$ (2).

$$p_{ij} = \frac{n_{ij}+\alpha}{\sum_{k=0}(n_{ik}+\alpha)} \quad (2)$$

Where i was occupancy level for the current step, j the occupancy level for the next step. α was a smooth factor used to avoid vanishing the transfer matrix.

XGBoost and Random Forest model. XGBoost and Random Forest model are both ensemble learning algorithm, which can combine a set of small and weak tree models together to generate a collectively strong model. The tree models can study parallelly and improve its performance iteratively. The input features for XGBoost and Random Forest model are same as the input features for ANN model.

ARMA_ANN Range Model. Above methods are occupancy point prediction, the output was a single value. However, considering about the strong stochastic characteristics of occupancy behavior, an interval prediction could provide more flexibility for control algorithm design. For this case study, the maximum occupancy level is 12 and in most time the occupancy level is lower than six, so in order to further narrow the prediction boundary, an ANN range prediction was designed. For this range prediction, the output of neural network was an interval with a customized boundary (the difference between upper bound and lower bound is one for this case).

As a regression problem, the standard procedure minimized the squared error between the target and the output. But for the range prediction, the loss function was a piecewise function: when the output lied within the range, the loss would be zero; when the output was larger than the upper bound or smaller than the lower bound, the loss would be the squared error between output and the bound, as shown in Eq. ($backward\ loss = \begin{cases} (output - y_{upper})^2, when\ output > y_{upper} \\ 0,\ when\ y_{lower} < output < y_{upper} \\ (output - y_{lower})^2, when\ output < y_{lower} \end{cases}$ (3). People can customize the prediction boundary by adjusting the difference between y_upper and y_lower.

$$backward\ loss = \begin{cases} (output - y_{upper})^2, when\ output > y_{upper} \\ 0,\ when\ y_{lower} < output < y_{upper} \\ (output - y_{lower})^2, when\ output < y_{lower} \end{cases} \quad (3)$$

Model Tuning and Evaluation

The grid search method was used for hyperparameter tuning. The training data for these seven models were 10 days with 960 data points, and the testing data is one day with 96 points. We were using rolling window method to generate training data from our dataset and then test model performance day by day on testing data. After grid search, the parameters used in this case was 2 hidden layers with 24 modes and 36 nodes. The learning rate was 0.001 with 1200 training epochs. For the RNN/LSTM model, the hidden layer was one with six nodes, the learning rate was 0.001 with 100 training epochs. The clipping value was 0.5 and the optimizer is RMSprop with a momentum equal to 0.8. The training sequence for RNN/LSTM was 4 hours ahead of prediction. For different prediction horizons, we use same inputs and same structures, but the output targets are different (15m, 1h, 4h, and 24h ahead occupant count). So the model can learn the hidden relation and shows a good transportability.

As a regression problem, the output was always a real number in float type like 1.6 person. We assume 1.6 = 60% × 2 + 40% × 1, the output has 40% and 60% probability to be 1 or 2. And we use the upper bound as our occupancy prediction results. After prediction, the mismatch rate was used to evaluate the model performance. We firstly count how many mismatch points (the prediction is not equal to the ground truth data), and then divided by the total testing points which was calculated by Eq. ($Mismatch = \frac{\sum_{n=1}^{N} I_n}{N}; I_n = \begin{cases} 0, & if\ X_n = Y_n \\ 1, & if\ X_n \neq Y_n \end{cases}$ (4):

$$Mismatch = \frac{\sum_{n=1}^{N} I_n}{N}; I_n = \begin{cases} 0, & if\ X_n = Y_n \\ 1, & if\ X_n \neq Y_n \end{cases} \quad (4)$$

Mismatch rate with one person tolerance was calculated by Eq. ($Mismatch\_1 = \frac{\sum_{n=1}^{N} I_{n1}}{N}; I_{n1} = \begin{cases} 0, & if\ Y_n - X_n \leq 1 \\ 1, & if\ Y_n - X_n > 1 \end{cases}$ (5):

$$Mismatch\_1 = \frac{\sum_{n=1}^{N} I_{n1}}{N}; I_{n1} = \begin{cases} 0, & if\ Y_n - X_n \leq 1 \\ 1, & if\ Y_n - X_n > 1 \end{cases} \quad (5)$$

RESULTS AND DISCUSSION

ARMA_ANN Model. Figure 3 shows the prediction results of the ARMA_ANN model with different prediction horizons over three days. The overall mismatch rate was 21.88% for the 15-minute ahead prediction, the mismatch rate with more than one person offset is 4.16%. The prediction error for first arrival time and last departure time was always less than 15 minutes. For 1-hour, 4-hour, and 24-hour ahead predictions, the overall mismatch rate was 36.11%, 39.23%, and 38.89%, respectively. The mismatch rate with one-person tolerance was 12.5%, 14.93%, and 14.58%, respectively. They all showed good performance for regular occupancy patterns as shown in the last two days of Figure 3. However, the model failed to predict the temporal and sudden changes that happened at the noon of the first day. One of the possible reasons could be that this kind of random change was not regular, and the machine learning model did not learn a similar trend before. For 15-minute to 24-hour ahead predictions, they all failed to predict the random presence that happened in the early morning of the last two days. The possible reason was similar because this random early occupied pattern was not a common situation and would be considered as a noise. Unless it happened many times and had a typical pattern, otherwise this kind of change was unpredictable.

RNN/LSTM Model. For RNN/LSTM model 15 minutes ahead prediction, LSTM performed slightly better with a 23.95% mismatch rate compared with 32.29% for RNN. If one person tolerance was allowed, LSTM has an offset-one mismatch rate of 3.82% compared with 6.25% for RNN. LSTM and RNN performed similarly for 1-hour, 4-hour, and 24-hour ahead prediction. The overall mismatch rates for RNN in these prediction horizons were 38.54%,

39.58%, and 41.66%, respectively. The overall mismatch rate for LSTM was 34.02%, 37.84%, and 40.63%, respectively. For mismatch rate with one person offset, RNN can correctly predict about 84% datapoints of a day, slightly less than LSTM with 90%. The output results of RNN and LSTM were flatter for 1-hour to 24-hour ahead prediction, they tended to predict averaged patterns as outputs, so that these models could not reflect the actual fluctuated pattern of occupancy behaviors compared with the ARMA_ANN model. That was because the training sequences for the RNN and LSTM models were much longer than the ARMA_ANN model. The longer sequences to some extent confused the training of the neural network to give averaged outputs. In short, these three models worked well for regular days with a typical pattern, but they all failed to capture the trend for a sudden large-scale change that happened on the first day.

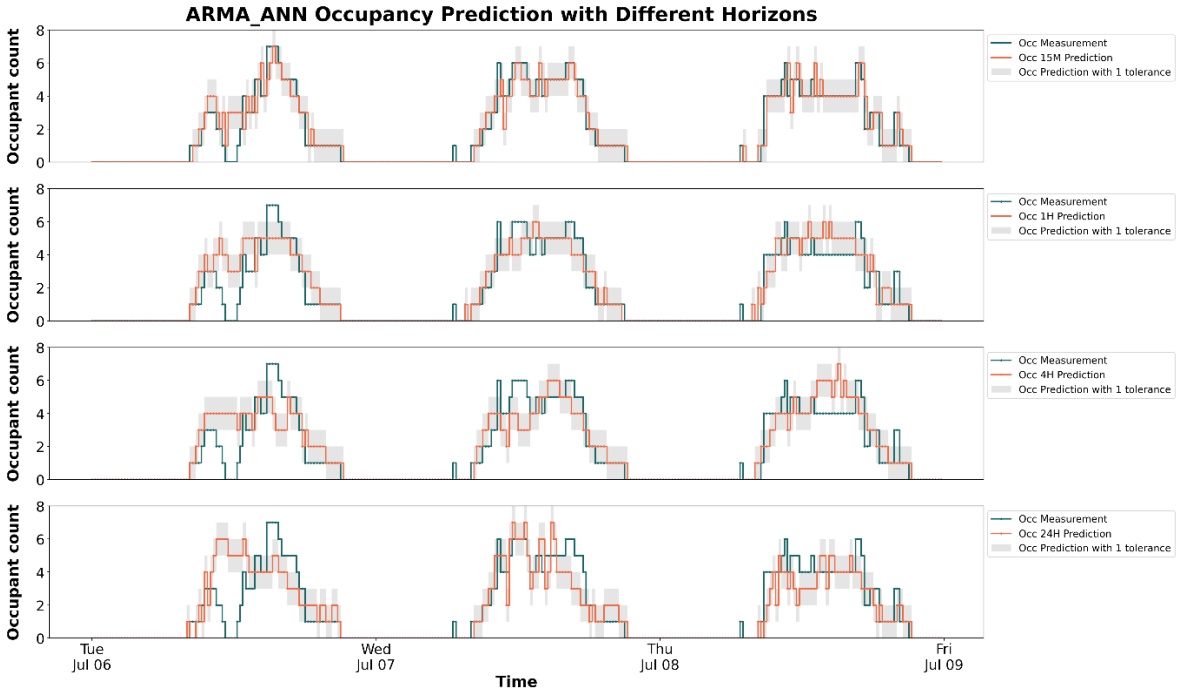


Figure 3 ARMA_ANN model prediction results with 15-minute1-hour, 4-hour, and 24-hour ahead predictions in three days.

Non-homogeneous Markov Model. Non-homogeneous Markov Model with change-point detection could only be used for short-term occupancy prediction (15 minutes ahead and one hour ahead). Otherwise, the prediction horizon may be outside the change point range. For 15-minute ahead occupancy prediction, mismatch rates without and with one person offset were 28.82% and 5.9%. For one-hour ahead prediction, the mismatch rate rose to 41.32% and 16.32%, respectively. There was a lag phenomenon for the Markov models when the prediction horizon exceeded one time step. This was because the current occupancy was only influenced by the last step (Markov property), and four steps prediction (one hour ahead) may lead to error accumulation.

XGBoost/Random Forest Model. XGBoost and Random Forest have similar performance with ANN model. The 15 minutes ahead occupancy prediction has a mismatch rate 6% with one person tolerance. The overall mismatch rate for 1 hour to 24 hours ahead prediction was 0.32 to 0.44, and the accuracy could reach 79%-88% if one person offset was allowed.

ANN Range Model. Finally, as for the ANN range model, it could customize the output range by setting the boundary limitations. For this low-density case study, we used range one prediction. The difference between the upper

bound and lower bound was only one person, tighter than the range used above (plus one and minus one was two). The overall accuracy of the ANN range model could reach 84.02% for 15-minute ahead of prediction. For 1-hour to 24-hour, the accuracy was about 76% to 79%, respectively.

Rolling window method was used to test the overall performance for different prediction algorithms from June 15th to July 15th, 2020. The summary of prediction results with different prediction horizons is shown in Table 1. The ARMA_ANN, XGBoost and Random Forest performed better than other models. For one month evaluation, ARMA_ANN, XGBoost and Random Forest could predict 15 minutes ahead of occupancy with 94% accuracy with one person offset. The overall mismatch rate for 1 hour to 24 hours ahead prediction was 0.32 to 0.44, and the accuracy could reach 79%-88% if one person offset was allowed. ANN_Range could give a range one interval prediction with 85% accuracy for 15 minutes ahead prediction and 72% accuracy for 1 hour to 24 hours ahead prediction. LSTM worked better than RNN model because the "gates" could determine which information was useful for occupancy prediction, but they all did not perform as well as ARMA_ANN model. The longer sequence not only added noise for learning but also slowed down the calculation speed. The HN_Markov had similar performance with LSTM model, but it could only be used for short term prediction. None of the above models could track the abrupt changes even we use flattening data preprocess, machine learning models need to study the internal relation from the historical data, and they do not have a good performance on unseen dataset.

Table 1. Model performance with different prediction algorithms and horizons

Model	Evaluate index	15 min	1 hour	4 hours	24 hours
ARMA_ANN	Mismatch	0.239	0.373	0.413	0.420
	Mismatch_1	0.039	0.118	0.152	0.163
RNN	Mismatch	0.367	0.481	0.476	0.482
	Mismatch_1	0.084	0.197	0.215	0.209
LSTM	Mismatch	0.305	0.437	0.455	0.466
	Mismatch_1	0.046	0.14	0.184	0.195
HN_Markov	Mismatch	0.271	0.429	-	-
	Mismatch_1	0.071	0.185	-	-
XGBoost	Mismatch	0.253	0.326	0.368	0.441
	Mismatch_1	0.059	0.146	0.187	0.215
Random Forest	Mismatch	0.229	0.329	0.368	0.406
	Mismatch_1	0.059	0.107	0.167	0.191
ANN_Range	Mismatch_range	0.158	0.251	0.278	0.284

CONCLUSION

In this study, seven machine learning models were used for 15-minute to 24-hour ahead of occupancy prediction. The ARMA_ANN, XGBoost and Random Forest performed better than other models with an accuracy of 95% with one person offset for 15 minutes ahead prediction and 84%-88% accuracy for 1 hour to 24 hours ahead prediction. The HN_Markov has similar performance with LSTM model, they all perform better than RNN model. The overall mismatch rate was about 36%-45%, with no significant difference for 1-hour ahead to 24-hour ahead prediction. Considering about the computation time and model complexity, ARMA_ANN, XGBoost and Random Forest are recommended for occupancy count prediction.

ACKNOWLEDGMENTS

The research work presented in this paper is supported by the U.S. National Science Foundation (Award No. 1949372)

REFERENCES

Pérez-Lombard, L., Ortiz, J., Coronel, J. F., & Maestre, I. R. (2011). A review of HVAC systems requirements in building energy regulations. Energy and buildings, 43(2-3), 255-268.

Cooper, S. (2004). Managing Energy Costs in Office Buildings. South Carolina State Documents Depository.

Yang, L., Yan, H., & Lam, J. C. (2014). Thermal comfort and building energy consumption implications–a review. Applied energy, 115, 164-173.

Zheng, W., Hu, J., Wang, Z., Li, J., Fu, Z., Li, H., ... & Yan, J. (2021). COVID-19 impact on operation and energy consumption of heating, ventilation and air-conditioning (HVAC) systems. Advances in Applied Energy, 3, 100040.

Guo, M., Xu, P., Xiao, T., He, R., Dai, M., & Miller, S. L. (2021). Review and comparison of HVAC operation guidelines in different countries during the COVID-19 pandemic. Building and Environment, 187, 107368.

Mirakhorli, A., & Dong, B. (2016). Occupancy behavior-based model predictive control for building indoor climate—A critical review. Energy and Buildings, 129, 499-513.

Hong, T., Yan, D., D'Oca, S., & Chen, C. F. (2017). Ten questions concerning occupant behavior in buildings: The big picture. Building and Environment, 114, 518-530.

Klepeis, N. E., Nelson, W. C., Ott, W. R., Robinson, J. P., Tsang, A. M., Switzer, P., ... & Engelmann, W. H. (2001). The National Human Activity Pattern Survey (NHAPS): a resource for assessing exposure to environmental pollutants. Journal of Exposure Science & Environmental Epidemiology, 11(3), 231-252.

D'Oca, S., Hong, T., & Langevin, J. (2018). The human dimensions of energy use in buildings: A review. Renewable and Sustainable Energy Reviews, 81, 731-742.

Wang, J., Huang, J., Feng, Z., Cao, S. J., & Haghighat, F. (2021). Occupant-density-detection based energy efficient ventilation system: Prevention of infection transmission. Energy and Buildings, 240, 110883.

Mokhtari, R., & Jahangir, M. H. (2021). The effect of occupant distribution on energy consumption and COVID-19 infection in buildings: A case study of university building. Building and Environment, 190, 107561.

Schibuola, L., & Tambani, C. (2021). High energy efficiency ventilation to limit COVID-19 contagion in school environments. Energy and Buildings, 240, 110882.

Dong, B., & Lam, K. P. (2014, February). A real-time model predictive control for building heating and cooling systems based on the occupancy behavior pattern detection and local weather forecasting. In Building Simulation (Vol. 7, No. 1, pp. 89-106). Springer Berlin Heidelberg.

Mirakhorli, A., & Dong, B. (2016). Occupancy behavior-based model predictive control for building indoor climate—A critical review. Energy and Buildings, 129, 499-513.

Kong, M., Dong, B., Zhang, R., & O'Neill, Z. (2022). HVAC energy savings, thermal comfort and air quality for occupant-centric control through a side-by-side experimental study. Applied Energy, 306, 117987.

Zou, H., Zhou, Y., Jiang, H., Chien, S. C., Xie, L., & Spanos, C. J. (2018). WinLight: A WiFi-based occupancy-driven lighting control system for smart building. Energy and Buildings, 158, 924-938.

Sun, Y., Haghighat, F., & Fung, B. C. (2020). A review of the-state-of-the-art in data-driven approaches for building energy prediction. Energy and Buildings, 221, 110022.

Yang, Y., Yuan, Y., Pan, T., Zang, X., & Liu, G. (2022). A framework for occupancy prediction based on image information fusion and machine learning. Building and Environment, 207, 108524.

Jin, Y., Yan, D., Chong, A., Dong, B., & An, J. (2021). Building occupancy forecasting: A systematical and critical review. Energy and Buildings, 251, 111345.

Wang, W., Chen, J., Hong, T., & Zhu, N. (2018). Occupancy prediction through Markov based feedback recurrent neural network (M-FRNN) algorithm with WiFi probe technology. Building and Environment, 138, 160-170.

Liang, X., Hong, T., & Shen, G. Q. (2016). Occupancy data analytics and prediction: A case study. Building and Environment, 102, 179-192.

Li, Z., & Dong, B. (2018). Short term predictions of occupancy in commercial buildings—Performance analysis for stochastic models and machine learning approaches. Energy and Buildings, 158, 268-281.

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

Dutta, J., & Roy, S. (2022). OccupancySense: Context-based indoor occupancy detection & prediction using CatBoost model. Applied Soft Computing, 119, 108536.

Dong, B., Yan, D., Li, Z., Jin, Y., Feng, X., & Fontenot, H. (2018, October). Modeling occupancy and behavior for better building design and operation—A critical review. In Building Simulation (Vol. 11, No. 5, pp. 899-921). Springer Berlin Heidelberg.