

Forecasting building occupancy: A comparative study of approaches

James Howard
Colorado School of Mines
1500 Illinois St.
Golden, CO 80401
jahoward@mines.edu

William Hoff
Colorado School of Mines
1500 Illinois St.
Golden, CO 80401
whoff@mines.edu

ABSTRACT

Occupancy knowledge of a building can lead to significant improvements in smart heating and cooling systems. In this work we gather occupancy information using passive infrared motion sensors densely placed throughout an office building and a classroom building.

We then compare a number of forecasting algorithms including a Bayesian combined predictor made up of other simpler algorithms on these datasets. Because the buildings have such different occupancy profiles we compare our algorithms on each dataset to get a better idea of what forecasting algorithm is most ideal for a given set of conditions.

1. INTRODUCTION

According to the U.S. Department of Energy energy for heating and cooling accounts for approximately 35 - 45% [4] of the total expenditure within a building. With such a large investment of energy being used to regulate the temperature of a building, any possible areas of improvement in this area are heavily sought after. Knowledge of occupancy of people within a building are an important component to intelligent heating and air condition systems. True levels of occupancy are rarely known and as a results most buildings instead rely on existing systems such as carbon dioxide sensors or motion activated lights.

In this paper we explore the forecasting accuracy of multiple common time series forecasting models as well as a Bayesian combined predictor trained using those models on real world occupancy data using a number of forecasting measurement metrics. Occupancy data is difficult to acquire and accurate ground truth values are rare as most buildings do not have sufficient infrastructure to properly sense people accurately throughout the building. Thus there has been considerable work in generating simulated models of occupancy [11, 5]. These simulations often involve agent based models where the concern is more with the accuracy of the models occupancy at any given time instead of the model's forecasting

abilities. Agent based models also tend not to scale well to large buildings where the where large numbers of agents, rooms and interactions lead to non-trivial solutions. A number of other researchers have created local wireless network occupancy sensing devices using a combination of sensors.

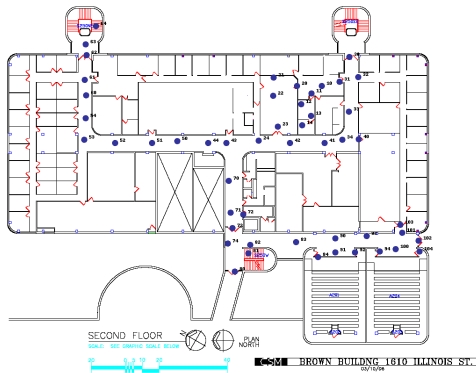
To acquire real occupancy estimates without simulation there have been multiple systems using a combination of simple sensors and wireless motes. Agarwal, et. al [1] has created motes using a combination of an IR sensors reed switch place on a door to determine the likelihood that a room is occupied. Mamidi [9] and the University of Southern California have developed a building-level energy management system using a combination of motion detectors and environmental sensors. Ground truth was collected and used as the basis for target values which were then run through a machine learning algorithm to accurately estimate occupancy. In both of these instances the occupancy estimation was limited to a small set of rooms only.

Our occupancy estimates are instead derived from a set of infrared sensors densely placed around a building. The data is collected from two different types of buildings, one is an office building, the other a classroom and research building with different schedules of levels occupancy. We also are not concerned only with short term forecasting as pre-heating and pre-cooling building control systems accept forecasts up to one day into the future [?]. Thus our analysis will be on forecasts from 10 minutes to hours into the future. We believe that this work will lead researchers to have a better idea of what forecasting algorithms will perform best under the specific conditions of our buildings.

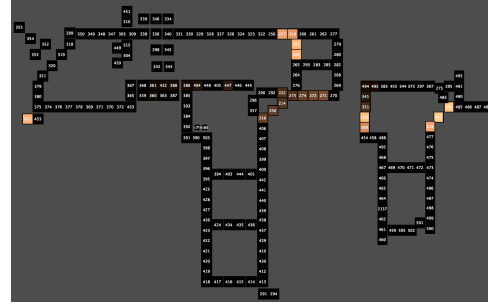
The remainder of the paper is laid out as follows: Section 2 introduces the datasets and gives a brief description of the data collection method and the types of buildings from which the data was generated. Section 3 discusses the forecasting algorithms along and any specific considerations we had to take to apply that algorithm to our problem. In section 4 we discuss the metrics we will use to analyze the data and show the results of our forecasts. Finally the paper concludes with a discussion of our conclusions in section 5.

2. OCCUPANCY DATA

Our datasets come from two sources. The first is a combined research and office building from Mitsubishi's MERL dataset [14]. The second is a classroom and office building from the Colorado School of Mines Brown Building (CITE



(a) Colorado School of Mines Brown building second floor



(b) Merl research lab 7th and 8th floor

Figure 1: Sensor locations for buildings

OUR PRIOR WORK). Both datasets use passive infrared sensors to estimate motion in an area. We use this data as our occupancy estimation. To handle instances where an individual may stand under a sensor for an extended period of time thus creating multiple readings when only one is necessary, we first remove the largest outliers from our datasets as defined by the standard deviation of the counts of people at that time.

2.1 Mitsubishi’s Dataset

The Mitsubishi’s electronic research labs dataset (MERL) a collection of over 200 passive infrared sensors place densely throughout the 7th and 8th floor of a research office building. The sensors are placed roughly two meters apart on the ceilings creating a dense sensing area with little non-sensed space. Readings are taken at the millisecond level, but due to the sensors settling times the interdetection time of motion is approximately 1.5 seconds.

The data was collected from March 2006 through March 2008 and there are roughly 53 million sensor readings. This building similar to most office buildings with a number of personal offices along with labs and conference rooms. Employees have roughly set schedules and holidays are observed as normal.

The counts of sensor activations have been aggregated every 10 minutes. Because of the lack of significant motion in the night, we look only at activations that occur between 6:00am and 7:00pm daily. A plot of the average activations of all Wednesdays for a single sensor along with a range of one standard deviation is given in figure 3b. Peak motion unsurprisingly occurs during the middle of the day corresponding to lunch time. There is another small peak of motion near the start of the day corresponding to people entering. Near the end of the day, instead of a peak there is a region corresponding to high variance. This seems to imply that while people enter at roughly the same time, there is a significant variance on when people leave the building.

2.2 Colorado School of Mines Dataset

The Colorado School of Mines dataset (CSMBB) is a collection of 50 passive infrared sensors mounted on the ceiling of the second floor a class and office room building. The den-

sity of the sensor placement depends on the location within the building. Outside auditorium is a dense collection of sensors place every meter. Throughout the rest of the building the sensors are placed roughly every 5 meters. Data was collected for one academic school year 2008 to 2009. To acquire readings, the sensors were polled every second and recorded data if motion was detected.



Figure 2: Passive infrared motion detector

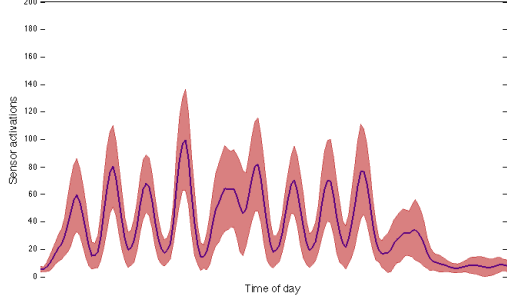
This building operates much differently than the MERL dataset as classes provide motion on a rigid schedule during the day. Also as students have exams and projects, late night motion is sporadic based on the time of year. The counts of sensor activations have been aggregated every 10 minutes. Despite occasional late night motion during exam time, most nights have no significant motion. For this reason we only collect data between 7:00am and 7:00pm daily. A plot of the average activations of all Wednesdays for a single sensor along with a range of one standard deviation is given in figure 3a. The defined peaks in the dataset correlate to class start and end times when most students will be in the hallways of the building.

3. FORECASTING MODELS

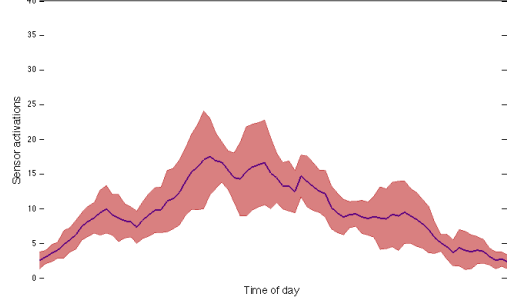
For comparison we select from some of the most common models used for time series forecasting. This section gives a brief introduction to each of these forecasting models.

As an introduction to some notation used for these models we define the time series dataset as $\{T_t^{(m)}\}$ data used for these models comes from a set of M binary infrared sensors. Each T_t^m is a 10 minute aggregate of sensor m reading at time block t .

3.1 Seasonal ARIMA Model



(a) CSM Brown Building typical day



(b) MERL typical day

Figure 3: Average sensor activations for a specific sensor on Wednesdays with one standard deviation range

The Auto Regressive Moving Average Model (ARMA) or derivations on its form (Auto regressive integrated moving average, Seasonal auto regressive moving average, etc) have been used in numerous forecasting applications from economics to vehicle traffic systems. While we have been unable to find ARMA based models used on building occupancy data directly, we have found it used to forecast building energy usage and vehicle occupancy [13, 7, 10]. Its forecasting accuracy is quite strong it can serve as a strong baseline of comparison for a forecasting problem.

Due to our traffic data having periodic trends and a non stationary mean, a seasonal ARIMA model is best suited to fit our data from the class of ARMA models. The seasonal ARIMA model is defined as:

$$\phi_p(B)\Phi_p(B^s)\nabla^d\nabla_s^DT_t = \theta_q(B)\Theta_Q(B^s)e_t \quad (1)$$

where $\{T_t\}$ is our observed time series and $\{e_t\}$ represents an unobserved white noise series ($e_t \sim N(0, \sigma^2)$) the values of which are computed through model training and are not known a priori. B is the backshift operator which is a function that allows access to older time readings. For example $BT_t = T_{t-1}$ and $B^5T_t = T_{t-5}$. ∇_s^D is the seasonal difference operator ($\nabla_s^DT_t = (1-B^s)^DT_t$) and $\phi, \Phi, \theta, \Theta$ are trainable parameters.

Seasonal ARIMA models are defined as $\text{ARIMA}(p,d,q)(P,D,Q)_s$ where p is the number of autoregressive terms, d is the number of differences and q is the number of moving average terms. P, D , and Q all correspond to the seasonal equivalents of p, d , and q . The parameter s is the seasonality of the model. For a full discussion of seasonal ARIMA models see Box and Jenkins [2].

Finding the correct values of p, d, q, P, D, Q, s is traditionally a hard problem. For a full discussion of finding the correct parameter values applied to a forecasting problem see Williams [13]. As a verification of our model, we applied the Ljung-Box test [8] on our set of residual data for each model. Both residual sets passed: $p = 0.9964$ MERL and $p = 0.1072$ CSMBB. Our final model parameters can be seen in the table below. Notice that the season is different for each model due to a difference in window of time for each day that we extracted data.

Dataset	p	d	q	P	D	Q	s
MERL	0	0	1	0	1	5	78
CSMBB	0	1	1	0	1	3	72

Forecasting from this model is performed by iteratively forward feeding values of the model into itself. Since the set of residuals e from a properly trained seasonal ARIMA model is described by a white noise gaussian distribution $N(0, \sigma^2)$, we can take the expected value of the residual at time e_{t+1} to be 0. This leaves us with the following forecasting equation:

$$\phi_p(B)\Phi_p(B^s)\nabla^d\nabla_s^DT_{t+1} = \theta_{q-1}(B)\Theta_{Q-1}(B^s)e_t \quad (2)$$

3.2 Historic Average

This model is simply the per day average of readings at each time step. For certain types of data this model is has been shown to be more accurate than seasonal ARIMA forecasting [10], specifically when the data has a strong historic correlation. Average forecasts have the advantage of being extremely computationally fast and having a forecast accuracy that does not depend on the forecasting horizon. This result will be shown later.

3.3 Time Delayed Neural Networks

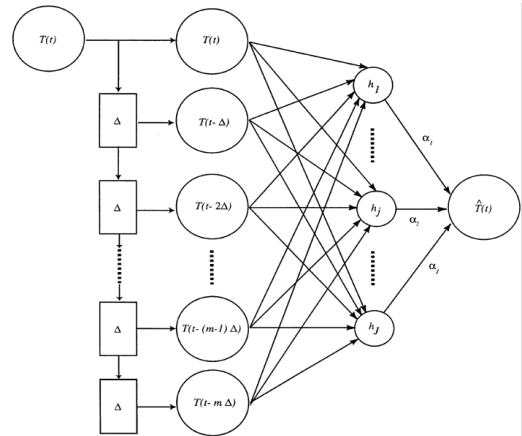
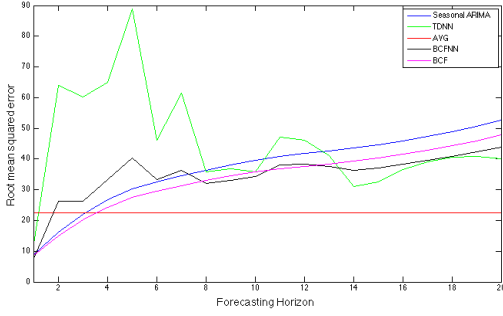
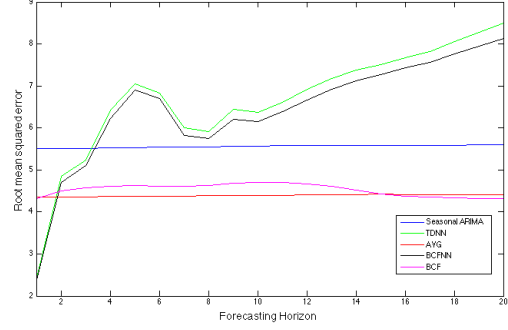


Figure 5: Architecture of a time delayed neural network with $m + 1$ inputs and J outputs [6].



(a) CSMBB forecasting model errors



(b) MERL forecasting model errors

Figure 4: Root mean square error of forecasting for each model vs forecasting horizon

Time delayed neural networks are a special subset of regression neural networks where the input data is a local history of data from the time series. Commonly the output is a single point forecast from that same time series at some point $t + \delta$ in the future. The form of our 1 hidden layer time delayed neural network is:

$$T_{t+1} = \phi\left\{\sum_{j=1}^J w_j \psi_j \left[\sum_{l=0}^m w_{jl} T_{t-l\delta} + w_{j0} \right] + w_0\right\} \quad (3)$$

where $\phi()$ is a linear activation function on the output layer and $\psi()$ is the standard sigmoid function.

Forecasting is performed by computing the output for a $m+1$ length window of time and then iteratively forecasting a set of time steps in the future by using forecast data as inputs to the next forecast.

3.4 Bayesian Combined Forecasting

The Bayesian combined forecasting (BCF) approach is one of several types of approaches which attempt to combine several forecasting methods. The approach works by computing a normalized likelihood for each forecasting model's accuracy of forecast at that point in time in the dataset. For example if we are performing a combined forecast on a seasonal ARIMA model and a time delayed neural network model then at each forecast point t we compute the likelihood of each model's prediction at that point in time. From this set of likelihood we can then perform a final prediction for $t + \delta$.

A full description of the model is done by Petridis [12]. The final forecasting equation is given by:

$$p_t^k = p(k|T_{1:t}) = \frac{p_{t-1}^K \cdot e(T_t - \bar{T}_{1:t-1}^k)}{\sum_{j=1}^K p_{t-1}^j \cdot e(T_t^j - \bar{T}_{1:t-1}^j)} \quad (4)$$

where $p(k|T_{1:t})$ is the probability of model k given data T_t , K is the total number of models, \bar{T}_t^k is the forecast from model k , and $e()$ is the probability mass function (PMF) of the forecasting error for model k .

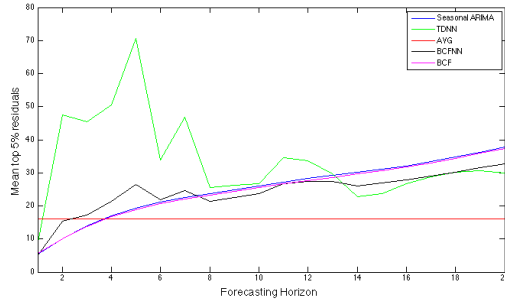
For this work we assume our PMF to be a constant variance white noise gaussian trained for each model. For future work, we plan to incorporate a more realistic error function based on the changing variance in our forecasting functions and the data. One such approach would be to use a generalize autoregressive conditional heteroskedastic to forecast what future forecasting variance will be for each model based on local conditions. We have not yet implemented this and instead simply have a constant gaussian PMF for each model similar to the work done by Zheng [15] on traffic forecasting using Bayesian Combined prediction.

Forecasting using a BCF approach is done by either computing a weighted forecast δ time steps into the future for each forecasting model or by simply selecting the model with the highest likelihood. For this paper we forecast using a weighted forecast of all models.

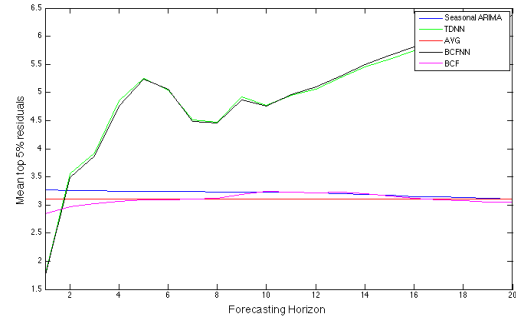
$$T_{t+\delta} = \sum_{k=1}^K p_t^k \cdot \bar{T}_{t+\delta}^k \quad (5)$$

Because the Bayesian combined forecasting approach is iterative It is possible that a long section of forecasts that indicate one model correct or incorrect can lead to likelihood underflow. Due to this problem we adjust our normalized likelihoods so that no model may reach a value below 0.001. This empirically chosen value has seemed low enough to not have a great impact on forecasts while still being high enough to allow model likelihoods to change quickly.

One other note on BCF comes from Diebold [3] who cautions use of forecasting using a Bayesian combination of models in all cases. Diebold points out that under certain situations a convex combination of forecasts for models may not be optimal and cases exist where taking negative likelihood weighting may be optimal. He mentions that these conditions are likely to arise during instances where the data may not be accurately described by any of the forecasting models. Searching for such situations is beyond the scope of our work and will be saved for later work. However because our forecasting models were all trained on the same datasets, if such a situation arose, each individual dataset would suffer from the same forecasting problem.



(a) CSMBB top 5% miss-forecasts



(b) MERL top 5% miss-forecasts

Figure 6: Average root mean square of the top 5% of miss-forecasts

4. RESULTS

Here we give the results of implementations for each of our models on each dataset. All of these results are run on the same set of models as described above. With the only exception being two different versions of a BCF. BCFNN is a Bayesian combined forecaster with an average model, a seasonal ARIMA model and a TDNN model, while BCF is simply a Bayesian combined forecaster without a TDNN model. This comparison was performed due to the TDNN model having a significant affect on the performance of the BCF.

All of the models were trained on 60% of the total datasets. Another 20% was used for model validation the final 20% used for testing. All results that follow are on the test set only.

4.1 Data metrics

4.2 Discussion of results

Figure 7 shows the results of the root mean squared error of forecasts across a forecast horizon up to 20 time steps (200 minutes) into the future for each model. The average model shows itself to be a strong indicator of future activity for forecasts beyond 30 minutes into the future. In the CSMBB dataset the Seasonal ARIMA model was a good forecaster of future activity while in the MERL set is performed significantly worse than even the average model on all forecasting horizons. This is likely due to a stronger seasonal component to the CSMBB dataset due class schedules. Instead on the MERL dataset there is little seasonal correlation and thus natural variance from a prior season may incorrectly affect current forecasts.

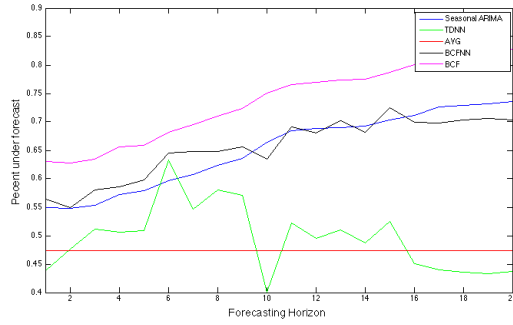
The BCFNN and BCF forecasting models always performed better the worst component model and in general seems to offer an improvement in forecasting accuracy. In general the BCFNN models follow the TDNN model. This is likely due to the model's strong short term forecasting accuracy which effects the likelihood of that model forecasting δ steps in the future.

5. CONCLUSION

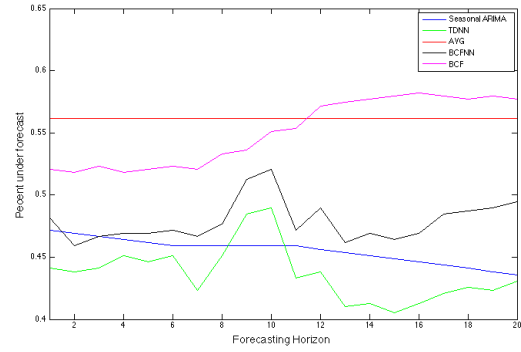
6. REFERENCES

- [1] Y. Agarwal, B. Balaji, R. Gupta, J. Lyles, M. Wei, and T. Weng. Occupancy-driven energy management

- for smart building automation. In *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building - BuildSys '10*, page 1, New York, New York, USA, 2010. ACM Press.
- [2] G. Box, G. Jenkins, and G. Reinsel. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, Inc., 4th edition, 2008.
- [3] F. Diebold. A Note on Bayesian Forecast Combination Procedures. In A. H. Westlund and P. Hackl, editors, *Economic Structural Change: Analysis and Forecasting*, pages 225–232. Springer-Verlag, 1991.
- [4] U. DOE. Building Energy Databook, 2010.
- [5] R. Goldstein, A. Tessier, and A. Khan. Schedule-calibrated occupant behavior simulation. In *Proceedings of the 2010 Spring Simulation Multiconference on - SpringSim '10*, page 1, New York, New York, USA, 2010. ACM Press.
- [6] J. Hansen and R. Nelson. Forecasting and recombining time-series compents by using neural networks. *Journal of the Operational Research Society*, 54:307–317, 2003.
- [7] W.-C. Hong, Y. Dong, F. Zheng, and S. Y. Wei. Hybrid evolutionary algorithms in a SVR traffic flow forecasting model. *Applied Mathematics and Computation*, 217(15):6733–6747, Apr. 2011.
- [8] G. M. Ljung and G. E. P. Box. On a Measure of a Lack of Fit in Time Series Models. *Biometrika*, 65(2):297–303, 1978.
- [9] S. Mamidi, Y. Chang, and R. Maheswaran. Improving building energy efficiency with a network of sensing, learning and prediction agents. In *International Conference on Autonomous Agents and Multi Agent Systems. AAMAS 2012*, 2012.
- [10] G. Newsham and B. Birt. Building-level occupancy data to improve ARIMA-based electricity use forecasts. In *2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings (BuildSys 2010)*, 2010.
- [11] J. Page, D. Robinson, N. Morel, and J.-L. Scartezzini. A generalised stochastic model for the simulation of occupant presence. *Energy and Buildings*, 40:83–98, 2008.
- [12] V. Petridis, A. Kehagias, L. Petrou, A. Bakirtzis, S. Kiartzis, H. Panagiotou, and N. Maslari. A Bayesian multiple models combination method for time series prediction. *Journal of intelligent and*



(a) CSMBB forecasts under actual data



(b) MERL forecasts under actual data

Figure 7: Percent of forecasts under the actual data per model vs forecast horizon

robotic systems, 31(1):69–89, 2001.

- [13] B. M. Williams and L. a. Hoel. Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical Basis and Empirical Results. *Journal of Transportation Engineering*, 129(6):664, 2003.
- [14] C. Wren, Y. Ivanov, D. Leigh, and J. Westbues. The MERL motion detector dataset: 2007 workshop on massive datasets. In *ICMI Workshop on Massive Datasets*, 2007.
- [15] W. Zheng, D.-H. Lee, and Q. Shi. Short-Term Freeway Traffic Flow Prediction: Bayesian Combined Neural Network Approach. *Journal of Transportation Engineering*, 132(2):114, 2006.