# Determining Vacancy in Buildings via Machine Learning Methods

Benjamen Neduva, Calvin Cramer, Lionel Quiambao, Lisa M. Slaughter, Maksim Nedorezov, Richard Li, Ryan Gosiaco, Victoria Salova

***Abstract*— It is known power dissipation can be reduced by introducing low-power mode and auto turn-off during vacant hours on devices in organization facilities. This paper develops a method for inferring vacancy from sensor outputs such as carbon dioxide level, number of WiFi connections, and electricity demand. Building data was processed and used to produce predictive models including logistic regression and artificial neural network. The best model was created through Artificial Neural Network (ANN) with an accuracy of 89%. Alternative models including Logistic Regression and ANN with different settings had accuracies ranging from 57% to 89%. The method demonstrated here is part of a larger body of work that will produce parameterized models to infer vacancy in buildings where ground truth is not available.**

## I. Introduction

Placeholder

## II. Methods

### A. Data Acquisition

The data was generated by the Student Community Center (SCC), a multi-use building that houses 7 student centers (such as the Cross Cultural Center, LGBTQIA Resource Center, and Undergraduate Research Center), 3 IET Computer Rooms, the CoHo South Cafe, a 200-occupancy multipurpose room, 5 meeting rooms, and outdoor and indoor seating/study spaces. Some relevant attributes of the SCC building are listed in the table below.

| Building Characteristics for Student Community Center | |
|---|---|
| Floor Area | $44,000 ft^2$ |
| Year Built | 2012 |
| Average Annual Electricity Consumption | 5338 MBTU |
| Energy Use Intensity (EUI) | $120 kBTU/ft^2$ |
| LEED Certification | Platinum |

Data for this building was obtained from an OSIsoft PI data historian using a custom Python client script. The historian stored time-series data sourced from the Siemens Building Management System (BMS) owned and maintained by the UC Davis Facilities Management Office. The dataset was interpolated in 15-minute increments between June 1st, 2018 at noon and November 8th, 2018 at noon. This time range included periods of missing or flatlined data. These windows of time were removed from the data set. The features of the data set are listed in the table below along with a description of their physical interpretation.

TABLE HERE

One of the study rooms in the SCC is dedicated to Extended Study (ES) hours, which provides extra time to study while the rest of the building is otherwise closed. Because the ES room is located inside the SCC, its special schedule was extended to the whole building, effectively increasing the overall business hours. These overall business hours are shown in Table II alongside a breakdown of the hours for Normal operation and Extended Study periods.

Vacancy data for the period was constructed according to the Overall Hours listed, with 30 minutes given after closing time for occupants to completely exit the building. The assumed periods of vacancy are listed in Table III.

When the Normal hours of operation come to a close, a security sweep of all floors is performed and occupants are ushered to a common area on the 1st floor until ES hours end. At this time everyone

TABLE I

FEATURE INTRODUCTION

TABLE II
OPERATIONAL HOURS FOR SCC

| | | Normal Hours | Extended Study Hours | Overall Hours |
|---|---|---|---|---|
| Summer Hours Jun. 16 - Sep. 23 | Mon - Thur | 7:00 am - 9:00 pm | 9:00 pm - 11:00 pm | 7:30 am - 11:00 pm |
| | Fri | 7:30 am - 6:00 pm | Closed | 7:30 am - 6:00 pm |
| | Sat | Closed | Closed | Closed |
| | Sun | Closed | Closed | Closed |
| Academic Year Hours Jun. 1 - Jun. 15 Sep. 24 - Nov. 8 | Mon - Thur | 7:30 am - Midnight | Midnight - 2:00 am | 7:30 am - 2:00 am |
| | Fri | 7:30 am - 9:00 am | 9:00 pm - Midnight | 7:30 am - Midnight |
| | Sat | 9:00 am - 7:00 pm | 7:00 pm - Midnight | 9:00 am - Midnigh |
| | Sun | Noon - 10:00 pm | 9:00 am - Noon<br>10:00 pm - Midnight | 9:00 am - Midnight |
| Holidays | Independence Day (Jul. 4) | Closed | Closed | Closed |
| | Labor Day (Sep. 3) | Closed | Closed | Closed |

TABLE III
VACANCY DATA

| | Vacancy Period Begin | | Vacancy Period End | |
|---|---|---|---|---|
| | Day of Week | Time | Day of Week | Time |
| Summer Jun. 16 - Sep. 23 | Mon | 11:30 pm | Tuesday | 7:30 am |
| | Tue | 11:30 pm | Wednesday | 7:30 am |
| | Wed | 11:30 pm | Thursday | 7:30 am |
| | Thur | 11:30 pm | Friday | 7:30 am |
| | Fri | 11:30 pm | Monday | 7:30 am |
| Academic Year Jun. 1 - Jun. 15, Sep. 24 - Nov. 8 | Mon | 12:30 am | Monday | 7:30 am |
| | Tue | 2:30 am | Tuesday | 7:30 am |
| | Wed | 2:30 am | Wednesday | 7:30 am |
| | Thur | 2:30 am | Thursday | 7:30 am |
| | Fri | 2:30 am | Friday | 7:30 am |
| | Sat | 12:30 am | Saturday | 9:00 am |
| | Sun | 12:30 am | Sunday | 9:00 am |
| Holidays | Jul. 4 | 7:30 am | Jul. 4 | 11:30 am |
| | Sep. 3 | 7:30 am | Sep. 3 | 11:30 am |

is forced to exit the building and a vacancy period begins.

### B. Normalization

Placeholder

### C. Feature Selection

Several methods were deployed for preprocessing the data to remove irrelevant features for the purpose of predicting the vacancy of a building in order to decrease the complexity of the problem. With many features it is possible to have redundant features which can lead to poor training of the artificial neural network model. Four feature selection methods, namly, Recursive Feature Elimination, Chi-Squared test, Extra Trees Classifier, and Random Forest Classifier, provided by the Scikit-learn library for Python, were deployed in an attempt to address this issue by establishing feature significance relative to other features.

The first method attempted was Recursive Feature Elimination (RFE) along with C-Support Vector Classification (SVC), an external estimator that is able to return the rank of each feature relative to other features. RFE uses this rank to eliminate features that are shown to be least important by considering smaller and smaller sets of features until the desired number of features are selected. SVC is a type of Support Vector Machine (SVM) algorithm that is used for classification problems, and its implementation in the Scikit-learn library is based on LIBSVM by Chih-Chung and Chih-Jen. In this case, the SVM model was trained with a linear kernel and a C parameter of 1. A linear kernel was used because the hypothesis was that there is a linear relationship in the data with the output. Intuitively, levels of CO2, Wifi AP connections, and electricity demand will increase as more people occupy a building. A C parameter of 1, which is the penalty of the error of misclassification, was used. All of the other parameters were set to default in the Scikit-learn library. Selecting to choose only one feature using RFE meant that all features were given a rank, where the lowest rank means highest significance. Out of roughly 19000 samples, only about 3000 were used for feature selection due to the quadratic time complexity of the algorithm, leading to extremely long training times. The samples were randomly selected from the original samples.

The second method deployed was the Chi-Squared test. It is a statistical method used for determining independence, and subsequently significance, between features in a dataset. Cramers V, a common strength test for Chi-Squared, was not performed on the features as it was unnecessary due to comparing this result with other feature selection algorithms. This test was run on the normalized data because negative values are invalid for the Chi-Squared test.

Next, Extra Trees Classifier (ETC) was performed to get the importance of each feature in the dataset (Geurts, 2006). Extra trees is a tree-based method for supervised datasets and it excels in being accurate and computationally efficient. It is different from random forests due to randomly choosing cut points instead of finding optimal cut points when building the decision trees. The Gini index was used to get the quality of each split and therefore, the importance of each feature at the end of the decision tree building process. The number of trees was set to 100 in the forest. This algorithm was performed on both the original dataset and on the normalized dataset and the result was similar.

Finally, Random Forest Classifier (RFC) was conducted to compare results with the Extra Trees Classifier. Both algorithms are very closely related but they differ in how they choose the cut point for each new tree. Whereas ETC randomly chooses cut points, RFC tests all possible splits and chooses the best one. This makes RFC more computationally expensive and intuitively more accurate, but it has been shown not to be true in all cases (Geurts, 2006). 100 decision trees were constructed, identical with ETC, and all other parameters were left as default values in the Scikit-learn library.

### *D. Outlier Detection*

Four outlier detection algorithms were compared to find the most optimal for this data set. The algorithms used were Isolation Forest, LOF, SVM, and Elliptic Envelope.

The first method is Isolation Forest. It works by randomly selecting a feature and randomly selecting split values of this features in order to isolate an observation. The less split values it takes to isolate a sample, the more likely it is to be an anomaly. For isolation forest, most of the parameters were set to default besides contamination and behaviour, which were set to auto and new, respectively. Contamination refers to the amount of contamination in the data set and since there is no way of know what percent of data are outliers, it is set to auto. Setting behaviour to new will match the decision function to other anomaly detection algorithm API and the decision function also depends on the contamination parameter. The second method is local outlier factor, which measures the density of each point with respect to its neighbors. The locality of each sample depends on k-nearest neighbors and the distance from these neighbors to the sample point determines the local density and the samples with significantly lower density are marked as outliers. For LOF, all the parameters stayed as default, besides contamination, which was set to auto for the same reasons as in Isolation Forest. The third method is one class SVM. This method works better for novelty detection than for outlier detection, however it can still predict outliers. This method estimates the support of a distribution by finding regions where most of the samples lie by nonlinearly projecting the data into feature space and then separating it from the origin by the largest margin. All the parameters for SVM are set to default. The last method is elliptic envelope and the parameters are set to default as well. This algorithm assumes that the data has Gaussian distribution and creates an ellipse that encapsulates the relevant data and leaves out the outliers.

The four algorithms were run on the original dataset and generated four new sets of outliers. To compare the accuracy of the selected outliers, the four updated datasets were split 50/50 for testing and training. A simple SVM model was fit on each of the four datasets. The results were compared using accuracy scores.

### *E. Model Construction*

This section explores the two classifying methods: Logistic Regression and Artificial Neural Network.

*1) Logistic Regression*

The Logistic Regression class from Scikit-learn was used for this purpose. The model used a Stochastic Average Gradient (SAG) solver. SAG performs very similarly to the traditional stochastic gradient method but differs in including a memory of the previous gradient values to increase convergence speed. The maximum iteration for SAG was set to 3000.

*Evaluation*

The Logistic Regression was used as a baseline for evaluating other models. Due to its simplicity, the solver for the model was not altered for investigation.

*2) Artificial Neural Network*

The Sequential model of the Keras library was deployed. The order of the data was randomized to reduce bias and the data was partitioned 70/30 for training and testing. The activation function for the hidden layers was explored in the grid search discussed below. Because the output is binary (vacant or not), the activation function *sigmoid*, which has an output between 0 and 1, would suffice. For the same reason, the loss function *binary cross entropy* was used. SGD, which implements stochastic gradient descent, from Keras was used as the optimizer for the model.

*Evaluation*

A grid search was performed for the ANN model over a range of activation functions for the hidden nodes, the number of nodes in each hidden layer, the number of hidden layers, and the learning rate for gradient descent to select the optimal parameters. A total of 300 different models were trained in the grid search.

The activation functions used include sigmoid, hyperbolic tangent, ELU (exponential linear unit), ReLU (Rectified linear unit), and softplus. These activation functions were chosen because of the relatively different structures each one exhibits. The number of nodes in each hidden layer ranged from 3 to 12, with a 3 step increment, and the number of hidden layers was either 1, 2 or 3. The batch size was set at a fixed 32, and the learning rate took the values of 0.001, 0.005, 0.01, 0.05, or 0.1.

For each permutation of the parameters of the ANN, the total training loss, training accuracy, testing loss, and testing accuracy was recorded. The loss was calculated using binary cross entropy. The most useful metric to measure the performance of each model is the testing accuracy, which determines how well the model predicts vacancy in new data.

## III. Result

With the given data, the Logistic Regression model and ANN model had the same performance. Logistic Regression achieved an accuracy of 0.893 with instance training time. The best ANN model had an ELU activation function, 9 hidden nodes per layer, 3 hidden layers, and a 0.1 learning rate. This model was fitted in 30 epochs with a batch size of 25, costing roughly 15 seconds. The training loss was 0.27690, training accuracy was 0.88446, testing loss was 0.26326, and testing accuracy was 0.894. However, considering the training time and oscilation of the result from data randomization, Logistic Regression is a superior choice.

The result of each stage in the pipeline is discussed below.

### A. Feature Selection

From the result of the 4 methods mentioned above, The number of features was reduced from 27 to 3, including *AP Connection Count*, *Electricity Demand kBTU/h*, and *Day of Week*.

### B. Outlier Detection

### C. Model Evaluation

Two steps were performed in this stage. First, grid search was performed for each ANN model. Then, ROC and PR were plotted for Logistic Regression and optimal ANN model.

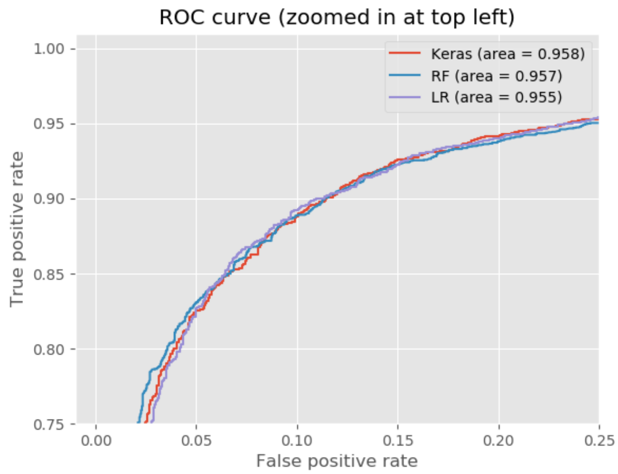
Table IV below includes a subset of result from the grid search.

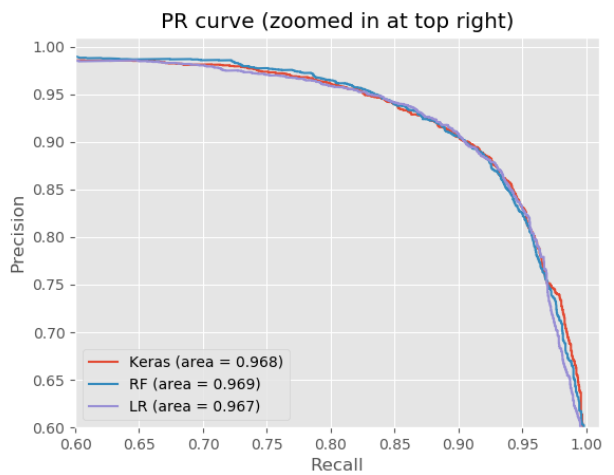
TABLE IV
BEST 5 ANN MODELS

| Act fnc | nodes hidden | hidden layers | learn rate | training accuracy | testing accuracy |
|---|---|---|---|---|---|
| elu | 9 | 3 | 0.1 | 0.89152 | 0.89084 |
| sigmoid | 6 | 2 | 0.05 | 0.89091 | 0.89030 |
| tanh | 3 | 3 | 0.1 | 0.89129 | 0.88976 |
| relu | 12 | 1 | 0.1 | 0.89114 | 0.88976 |
| elu | 6 | 2 | 0.1 | 0.89083 | 0.88958 |

As shown, ELU with parameter ELU, 9, 3, 0.1 for activation function, hidden nodes, hidden layers, and learning rate respectively performed the best.

Next, this model was compared with the Logistic Regression model and the ROC and PR curves are shown below.




ANN performed slightly better than the Logistic regression model; however, the difference is minimal.

## IV. Discussion

This disussion for each stage in the pipeline is discussed below.

### A. Feature Selection

By comparing all of the feature selection algorithms and the importance of each feature in each algorithm, all features that reference the CO2 level of a room were removed. This was expected because the AP connection count and electricity demand can be directly matched to a person being inside a building. With most people on campus having wireless devices and being connected to the campus wifi, as soon as someone walks into the the SCC, their wireless devices connect to the wifi indicating that the building is not vacant. Total electricity demand is a coarser measure of building vacancy because the same amount of lights can be on in a room regardless of head count. Similarly, the day of the week is unrepresentative because the building is mostly empty on weekends. While these features are immediately affected by a persons presence, the CO2 level exhibits a lag in the measurements. Therefore, it is difficult to use features with such an attribute to detect vacancy.

### B. ANN

It was determined the ANN with ELU activation function for hidden layers, 9 hidden nodes per layer, 3 hidden layers, and a learning rate of 0.1 was the most optimal. It is worthwhile to note that this model only exhibits a 0.05% increase in accuracy than the second best model. The small difference in performance between each model as well as the shuffling of data at runtime contribute to uncertainty that may be present in the results for each model.

A vast majority of the models trained in the grid search has a testing accuracy in 85% to 89% range, but a noteworthy number of models that had a sigmoid activation function did significantly worse in the 57% range. It is worthwhile to note that models with the sigmoid activation function did still have the ability to score within the 85% to 89% testing accuracy range.

An alternative ANN setup consists of non-uniform number of nodes for each hidden layer was not examined. It will serve as a good direction for future experiments.

## V. CONCLUSION

Placeholder

## References


[1] Chih-Chung Chang, Chih-Jen Lin, LIBSVM: A library for support vector machines. ACM Trans. Intell. Syst. Technol. 2, 3, Article 27, 27 pages, May 2011.

[2] Geurts, P., Ernst, D., Wehenkel, L., Extremely randomized trees. Mach Learn, 63:1, pp 3-42, Apr. 2006.

[3] M. Pal, Random forest classifier for remote sensing classification. International Journal of Remote Sensing, 26:1, pp 217-222, 2005.
[4] McHugh, M.L., The Chi-square test of independence. Biochemia Medica, 23:2, pp 143-149, 2013.
[5] Pedregosa et al., Scikit-learn: Machine Learning in Python. JMLR 12, pp. 2825-2830, 2011.

## VI. Contribution

*Benjamen Neduva*

- Initial feature analysis
- Report outline
- Presentation preparation

*Calvin Cramer*

- ANN
- Grid search
- Normalization

*Lionel Quiambao*

- ANN

*Lisa Slaughter*

- Data acquisition
- Logistic Regression
- Project management

*Max Nedorezov*

- Feature selection
- Project scoping

*Richard Li*

- Report finalization

*Ryan Gosiaco*

- Initial feature analysis
- Model evaluation

*Victoria Salova*

- Outlier detection

*Yasmin Sayeed*

- Presentation preparation