Energy-Efficient Smart Buildings by Occupancy Prediction

Mina Moghaddam
Department of Computer Engineering Middle East Technical University, Ankara, Turkey
Mina.moghaddam90@gmail.com

Tahsincan Köse
Department of Computer Engineering Middle East Technical University, Ankara, Turkey
tahsincan.kose@ceng.metu.edu.tr

M. Rasit Ozdemir
Department of Computer Engineering Middle East Technical University, Ankara, Turkey
e1942606@ceng.metu.edu.tr

Zakiah Utami
Department of Computer Engineering Middle East Technical University, Ankara, Turkey
utami.zakiah@metu.edu.tr

Seyda Ertekin
Department of Computer Engineering Middle East Technical University, Ankara, Turkey
seyda@ceng.metu.edu.tr

***Abstract*—Based on the increase in energy consumption, energy efficiency has become a paramount topic. Through the energy conservation studies, it can be achieved by predicting occupant presence in buildings. In this study, the accuracy of occupancy prediction has been tested on data from differences in levels of Light, CO2, and Humidity in intervals of 1, 3, and 5 minutes. Artificial Neural Networks (ANN), Support Vector Machines (SVM), and Decision Trees have been chosen as learning algorithms. Various setups and hyper-parameters are used and the accuracy is in range of 80 to 90 percent. The study shows that using data generated from only changes in values of parameters, despite being more general than absolute values, have still a high prediction accuracy.**

Index Terms — Building energy efficiency, machine learning, occupancy prediction, office room.

I. Introduction

In this era, occupation in human spaces impress different aspects of a building like light, temperature, humidity and so on. It is of little debate in todays world that modeling occupant behavior and its information is of great importance. It is quite expected to notice significant energy savings that can be achieved by the accurate prediction of occupancy in buildings. In this developed society, commercial office buildings require a large amount of area and energy to create a comfort zone for their occupants[1]. With this in mind, the office buildings are useful targets for modeling occupant behavior to manage its potential of energy sources accurately in order to increase energy savings. In a recent study, it is illustrated that accurate occupation prediction in buildings has been estimated at 15 to 25 % energy savings[2]. In another simulation-based study[3], the potential energy savings is suggested as approximately 30%.

Current status of industrial energy consumption possesses an important risk to society and environment in terms of sufficiency of resources. Although various governments around the globe have set quantified targets that anticipate 20%[4] energy usage reduction in overall, the total industrial consumption still follows an ascending trend. Also, commercial office buildings comprise the majority of in- floor areas in most of the developed countries hence consume the remarkable amount of energy in the logistics of building services. Having all these considered the problem with regards to the necessity of effective energy usage. Meanwhile, this implies the minimization of resource dissipation in such enormous in-floor areas. In fact, this is just a single part of the whole problem. Thinking on a global scope brings one to the problem of worlds inability to restore its natural resources fully anymore. Failing to capture this problem would result in higher energy wastage which becomes more and more profoundly threatening for the entire world. For that reason, resolving this sub-problem effectively would inspire all other sectors, thus affects the whole problem in long-term.

II. Method

The project's aim is to predict occupation in an office room during different hours based on the data from light, temperature, humidity and CO2 by using Machine Learning approach. There were used three datasets for training and testing, that two testing datasets for open and close door in the office [5].

The original article in which the dataset was gathered and analyzed used a variety of statistical approaches on the data. Our aim is to take ANN, SVM[6] and Decision Tree methods to increase the accuracy in a generalized version of the problem, which we believe that is a more extensive solution regarding to the overall problem. Additionally, we assume that augmenting the data with expert knowledge extracted from the already existing dataset can have a boosting effect on performance of our method.

The datasets consist of labeled training and test data with 7 attributes such as Time, Temperature, Relative Humidity, Light, CO2, Humidity Ratio, and Occupancy. Training data consists of 8143 entities. Test data is divided into two portions, one is data gathered while the door was closed, with 2665 rows of data, and the other one is data gathered while the door was open, with 9752 rows of data [7].

We have used the already existing information to augment the data with 9 new attributes. These 9 new attributes are: Differences of CO2 level in 1 minute interval, Differences of CO2 level in 3 minutes interval, Differences of CO2 level in 5 minute interval, Differences of Humidity level in 1 minute interval, Differences of Humidity level in 3 minute interval, Differences of Humidity level in 5 minute interval, Differences of Light level in 1 minute interval, Differences of Light level in 3 minute interval, and Differences of Light level in 5 minute interval.

This approach is to globalize the model. The previous study [8] have used absolute values which can be attributed to a certain location or time. By training a model using only the difference we are assuming the model will become more invariant to time and location.

III. Evaluation

Before anything, it is highly beneficial to make descriptive analysis on the data rather than throwing it into models. For this purpose, visualizing data is a good first practice. Recalling that the dataset is processed, it now constitutes of 9 attributes which require a hyperplane representation. In order to have a decent visualization, reducing the parameter space to 2 and 3 most important ones (only for visualization) using *Feature Selection*[9] is a common method. However, feature selection might not provide a clear separation between the distinct classes. Such a separation heavily depends on the complexity of data. ***sklearn*** library is used throughout the model setups in the course of feature manipulation tasks in which all these are readily implemented to facilitate one-line usages. For instance, **K** best features in terms of ***chi square***[10] independence (relatedness) score metrics are extracted through utilization of aforementioned library. All in all, best features are plotted in a scattered manner to gain a deeper understanding on the dataset. Nonetheless, we couldn't achieve a clear separation between the classes. Corresponding figures represent the best resulting layouts that have the highest visual separations.

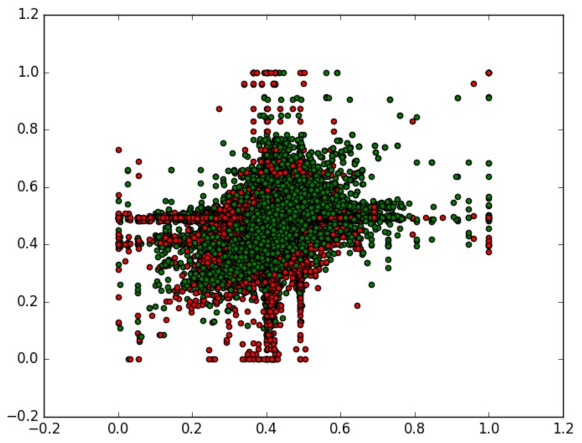

Fig. 1. Visualization with 2 Best Features

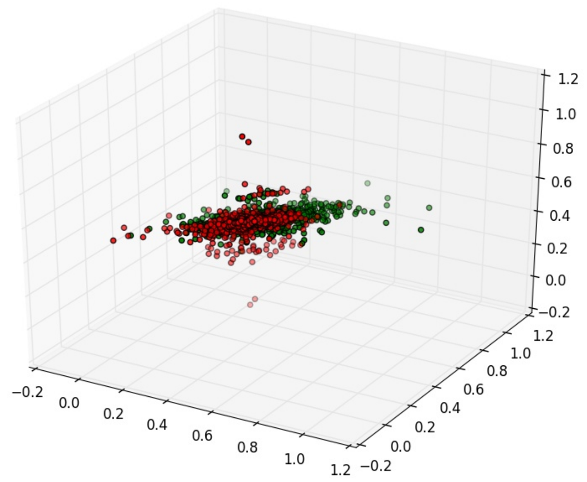

Fig. 2. Visualization with 3 Best Features

In order to automate the process, we implemented visualizer functions that consume all possibilities both in 2 and 3 dimensions. Through these methods, data can be visualized with any combination of attributes. In both Figure 1 and 2, data is roughly separated with horizontal and vertical planes respectively.

Lastly, the training data is composed of 1729 positive instances (occupied) whereas 6414 ones are negative instances (non-occupied). This is a slightly imbalanced dataset, thus considering accuracy would not be a bad choice. Also as a supporting performance metrics, *F1 score* or *gmeans* will possess a complementary role.

We have trained several ANNs, SVMs and Decision Trees with different sets of hyper-parameters to compare the estimation rates.

A. *Initial Test With Artificial Neural Network*

Our hypothesis states that it is possible to predict occupancy in a space with only using differences in Light level, CO2 level and Humidity level. In order to test the hypothesis, a simple neural network is planned for evaluation.

The proposed initial architecture is a Fully Connected Artificial Neural Network (ANN) which consist of 3 hidden layers. First two hidden layers have 10 neurons each and the last hidden layer have 8 neurons. There is a drop out layer in between second and third hidden layer to combat over fitting. Figure 3 is an illustration of this setup. The network uses ReLU as activation function for all neurons except last hidden layer which uses Softmax as activation function. It uses a simple Mean Squared Error (MSE) as Loss Function and also uses Adam optimizer [11] with learning rate of 0.01 to optimize it.

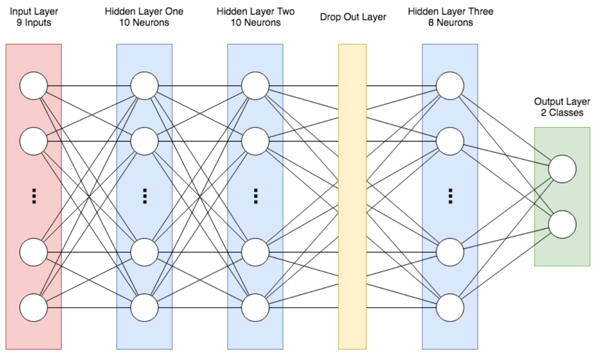

Fig. 3. Initial structure for the test ANN.

Tests have been performed on network in different epochs ranging from 500 to 8000. Results of the test are available in Table I. Maximum accuracy is **0.906** for closed door environment and **0.81** for open door environment.

TABLE I
EPOCH VS. ACCURACY OF TEST ARCHITECTURE

Epochs	*Accuracy - Closed Door*	*Accuracy - Open Door*
500	0.816	0.801
1000	0.831	0.801
1500	0.888	0.810
2000	0.901	0.809
2500	0.899	0.807
3000	0.901	0.804
3500	0.900	0.802
4000	0.904	0.805
4500	0.900	0.803
5000	0.900	0.809
5500	0.897	0.805
6000	0.900	0.803
6500	0.906	0.807
7000	0.903	0.802
7500	0.901	0.804
8000	0.899	0.805

The high accuracy in this test proves that it is possible to use this augmented data to predict occupancy of an environment. In the next section other methods, architectures and different hyper-parameter optimizations will be attempted.

B. *Additional Artificial Neural Network Architectures*

In the beginning, a neural network is constructed to initiate experimentation with 2 hidden layers having 10 and 8 hidden units respectively. Input layer contains 9 input units by the number of attributes, whereas output layer contains 2 output units. Learning rate is determined as $\alpha = 0.001$. Loss criterion is *cross entropy* and optimizer function is chosen as *Adam*[1] optimizer[11]. *Sigmoid* function is the activation function in all layers except the output layer. Output layer employs *softmax* activation function. Loss history is plotted below so as to determine the optimum number of epochs for this particular architecture.

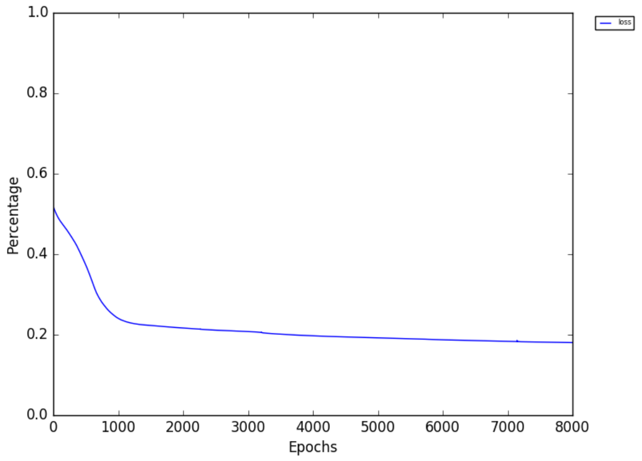

Fig. 4. Loss History

As it is seen in the figure 4 loss number reaches a local minima at roughly 1500 epoches. Since Adam is a variation of Stochastic Gradient Descent, loss does not stabilize and starts producing tiny fluctuations. Stochastic Gradient Descent methods have the ability to recover from local minimum and that is the underlying reason for this phenomenon. Its rate is determined by the nearly extinguished gradients. Therefore, learning process is in proximity of an equilibrium state. Continuing loss reduction only contributes to over-fitting, so learning with high number of epochs will not pay off. Below figure better illustrates the relationship between loss and accuracy/F1 score performance metrics.

[1]An extension to classical stochastic gradient descent. Here is a nice blog post that explains it in simple terms.

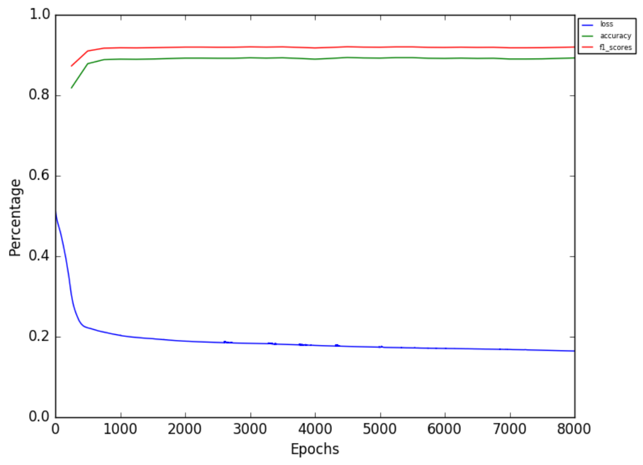

Fig. 5. Loss, Accuracy, F1 Score of $\alpha = 0.001$

Maximum accuracy is **0.8896**, F1 score is **0.9179** with 3000 epochs. Next experiments are done with the aim of increasing these rates above 90% instead of just trying all the hyper-parameters naïvely. In order to determine whether there exists a better minima in the loss function; learning rate should be configured.

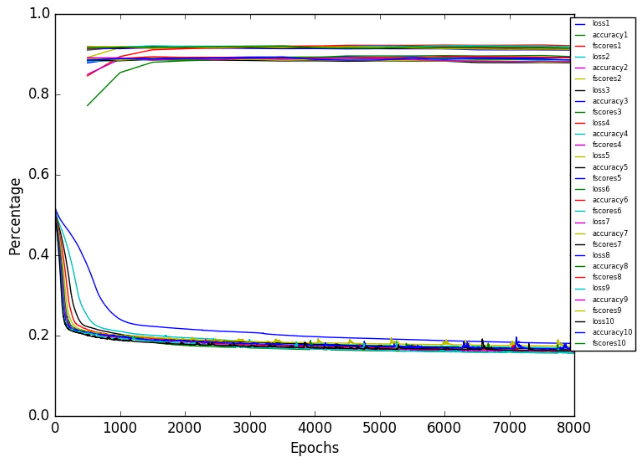

Fig. 6. Loss, Accuracy, F1 Score of $\alpha \in [0.001 - 0.01]$ with step 0.001

Figure 6 represents loss, accuracy, F1 score values with respect to learning rates and epochs. *loss1* denotes the loss curve with learning rate 0.001. All subsequent curves are plotted in this manner, i.e. index on right-hand side is multiplied by a factor of 0.001. This set of experiments output *accuracy* and *F1 score* with learning rate $\alpha = 0.009$ at epoch 2500 as at most **0.8961** and **0.9221** respectively. A very brief and concise comment can be brought about the figure above; that is, as learning rate increases the speed of loss convergence and fluctuations increase, which makes them *relatively bad* models especially with higher number of epochs.

Experiments thus far have been done with a simple model. Architecture adjustment is normally required when there are over-fitting or under-fitting issues in the learning process. Nonetheless, this model does not suffer from those two; thus, adjusting the number of layers and their sizes would not make any significant improvements on the estimations. In any case, this assumption should be tested via experimentation on these particular hyper-parameters. Note that learning rate is fixed to 0.009.

TABLE II
EPOCH VS. ACCURACY OF SEVERAL ARCHITECTURES - CLOSED DOOR

	9x10x10x8x4	*9x16x8x8x4*	*9x10x8x12x4 **	*9*	*9x8*
500	0.893	0.890	0.884	0.820	0.879
1000	0.898	0.890	0.893	0.820	0.891
1500	0.894	0.886	0.896	0.820	0.892
2000	0.892	0.883	0.894	0.820	0.889
2500	0.892	0.878	0.890	0.821	0.601
3000	0.894	0.879	0.887	0.821	0.878
3500	0.894	0.878	0.888	0.821	0.885
4000	0.890	0.875	0.885	0.821	0.880
4500	0.893	0.874	0.886	0.821	0.880
5000	0.893	0.872	0.885	0.821	0.883
5500	0.892	0.874	0.884	0.821	0.879
6000	0.890	0.875	0.884	0.821	0.882
6500	0.889	0.877	0.886	0.821	0.878
7000	0.891	0.875	0.886	0.821	0.883
7500	0.889	0.873	0.884	0.821	0.883
8000	0.890	0.873	0.882	0.821	0.880

* Hidden layer before the output layer utilizes *ReLU* activation function instead of sigmoid.

Except the linear model, all models roughly perform the same as can be seen in the figure. After several experiments, first architecture with 4 hidden layers with respectively 10, 10, 8 and 4 hidden units outputs the best accuracy of **0.898** at 1000 epochs among other configurations. Reaching such an accuracy rate in shorter training durations is also another advantage.

Above experiments serve the base for fine-tuning for overall model to improve and thus only consider first test set which is gathered when the door is closed. A similar table to Table VII is sketched below for open door test set.

Note that, initial architecture which employs different loss criteria and an additional dropout computed substantially less accuracy rates for open door dataset. This is simply because of the increasing noise in the sensor data when the door is opened.

TABLE III
EPOCH VS. ACCURACY OF SEVERAL ARCHITECTURES - OPENED DOOR

	9x10x10x8x4	*9x16x8x8x4*	*9x10x8x12x4 **	*9*	*9x8*
500	0.778	0.795	0.789	0.789	0.790
1000	0.804	0.805	0.799	0.789	0.797
1500	0.808	0.812	0.790	0.789	0.810
2000	0.807	0.810	0.794	0.789	0.815
2500	0.807	0.812	0.803	0.786	0.814
3000	0.804	0.810	0.801	0.789	0.814
3500	0.802	0.807	0.799	0.787	0.808
4000	0.801	0.804	0.798	0.786	0.803
4500	0.798	0.801	0.798	0.785	0.803
5000	0.793	0.798	0.797	0.785	0.800
5500	0.792	0.799	0.793	0.784	0.795
6000	0.790	0.801	0.792	0.783	0.794
6500	0.788	0.803	0.791	0.782	0.793
7000	0.787	0.804	0.786	0.782	0.792
7500	0.784	0.801	0.784	0.782	0.790
8000	0.784	0.799	0.784	0.781	0.787

In this set of experiments, last architecture reaches to the best accuracy of **0.815** rate among others as can be seen in Table III. Learning rate α is fixed to 0.009 as in the first set of experiments.

Consequently, initial architecture with MSE loss performs slightly better than the current set of architectures which utilizes cross entropy loss, regarding to the closed door test set. Contrarily, latter performs slightly better than the initial architecture in terms of open door test set.

C. Suport Vector Machines

For Support Vector Machine method, we configure the SVM parameters and report their results. We record the results for two performance metrics: accuracy and f-score. F-score is also important for us, because our data set is slightly imbalanced. F-score is a kind of weighted average of precision and recall.

We tried every configuration for hyperparameters like penalty parameter(C), tolerance for stopping criterion, kernel function and gamma(kernel coefficient for kernel functions) while creating Support Vector Machine model. In order to do that we trained a model for each configuration separately and measured accuracy and f-score for it. Our best accuracy results are %86.4520 for the first test dataset and %79.8351 for the second test dataset. Our best f-score is same for the best configuration for accuracy in first dataset. However, it is different in second dataset. The best f-score result is %83.5019 for the first dataset and %67.3889 for the second dataset.

TABLE IV
SVM CONFIGURATIONS AND ACCURACY AND F-SCORE RESULTS FOR THE FIRST TEST DATA

C	tol	gamma	kernel	accuracy	f-score
0.320	0.013	0.108	rbf	0.833395	0.819084
0.321	0.013	0.108	rbf	0.833395	0.819084
0.322	0.013	0.108	rbf	0.843771	0.819573
0.323	0.013	0.108	rbf	0.843399	0.819084
0.324	0.013	0.108	rbf	0.843771	0.819084
0.325	0.013	0.108	rbf	0.843771	0.819084
0.326	0.013	0.108	rbf	0.843771	0.819689
0.327	0.013	0.108	rbf	0.864146	0.820177
0.328	0.013	0.108	rbf	0.833771	0.819805
0.329	0.013	0.108	rbf	0.864146	0.820177
0.330	0.013	0.108	rbf	0.843771	0.819805
0.327	0.0133	0.108	rbf	0.864146	0.820177
0.327	0.0134	0.108	rbf	0.864146	0.820177
0.327	0.0135	0.108	rbf	0.864146	0.820177
0.327	0.0136	0.108	rbf	0.864146	0.820177
0.327	0.0137	0.108	rbf	0.864146	0.820177
0.327	0.0138	0.108	rbf	0.864146	0.820177
0.327	0.0139	0.108	rbf	0.864146	0.820177
0.327	0.0136	0.109	rbf	0.862273	0.818205
0.327	0.0136	0.110	rbf	0.861896	0.817717
0.327	0.0136	0.111	rbf	0.861896	0.817717
0.327	0.0136	0.112	rbf	0.863020	0.819063
0.327	0.0136	'auto'	rbf	0.864520	0.835019
0.327	0.0136	'auto'	linear	0.740938	0.583340
0.327	0.0136	'auto'	sigmoid	0.562589	0.435214

TABLE V
SVM CONFIGURATIONS AND ACCURACY AND F-SCORE RESULTS FOR THE SECOND TEST DATA

C	tol	gamma	kernel	accuracy	f-score
0.315	0.013	0.108	rbf	0.777945	0.673503
0.316	0.013	0.108	rbf	0.777945	0.673503
0.317	0.013	0.108	rbf	0.778047	0.673708
0.318	0.013	0.108	rbf	0.777842	0.673410
0.319	0.013	0.108	rbf	0.777739	0.673317
0.320	0.013	0.108	rbf	0.777739	0.673429
0.321	0.013	0.108	rbf	0.777739	0.673429
0.322	0.013	0.108	rbf	0.777842	0.673522
0.323	0.013	0.108	rbf	0.777842	0.673522
0.324	0.013	0.108	rbf	0.777739	0.673429
0.325	0.013	0.108	rbf	0.777739	0.673429
0.317	0.0133	0.108	rbf	0.778047	0.673708
0.317	0.0134	0.108	rbf	0.778047	0.673708
0.317	0.0135	0.108	rbf	0.778047	0.673708
0.317	0.0136	0.108	rbf	0.778047	0.673708
0.317	0.0137	0.108	rbf	0.778047	0.673708
0.317	0.0138	0.108	rbf	0.778047	0.673708
0.317	0.0139	0.108	rbf	0.778047	0.673708
0.317	0.0136	0.109	rbf	0.777947	0.673503
0.317	0.0136	0.110	rbf	0.777945	0.673503
0.317	0.0136	0.111	rbf	0.777945	0.673503
0.317	0.0136	'auto'	rbf	0.778150	0.673889
0.317	0.0136	'auto'	linear	0.798351	0.546185
0.317	0.0136	'auto'	sigmoid	0.674889	0.514299

D. Decision Trees

The main idea behind the decision tree algorithm is to build a tree-like model from root to leaf nodes. All nodes receive a list of inputs and the root node receives all the examples in the training set. Each node asks a true or false question for a feature and responses by partitioning into two subsets. The subsets then become the input the child nodes where the child node asks another question for one of the other features.

The challenge to building such a tree is which question to ask at a node and when. To do this, some experiments were done by tuning some parameter values, such as ”Criterion” to measure the quality of a split which can be ”gini” for the Gini impurity as a criterion to minimize the probability of misclassification or ”entropy” for the information gain i.e. the amount of information that’s gained by knowing the value of the entropy of the distribution before subtracted by the entropy of the distribution after the split, ”Max Depth” for maximum depth of the tree, ”Min Weight” to set the minimum weighted fraction of the sum total of weights required to be at a leaf node, ”Max Features” to set the number of features to consider when looking for the best split, and ”Random State” for the randomness of the state.

Below are the tables consists of some experimental data in two cases, closed and open door.

TABLE VI
ACCURACY AND F1-SCORE OF CLOSED DOOR CASE

Criterion	*Max Depth*	*Min Weight*	*Max Features*	*Random State*	*Accuracy*	*F1 Score*
entropy	None	0	auto	0	0.789	0.763
gini	None	0	None	None	0.783	0.763
gini	4	0	None	None	0.826	0.797
gini	6	0	None	None	0.834	0.80
gini	7	0	None	None	0.832	0.805
entropy	6	0	auto	0	0.822	0.790
entropy	6	0	sqrt	None	0.822	0.790
entropy	6	0.01	log2	0	0.822	0.785
gini	6	0.5	None	None	0.635	0.388
gini	6	0.1	None	None	0.809	0.763
gini	6	0.01	None	None	0.818	0.788
gini	6	0.01	auto	None	0.827	0.793
gini	6	0.01	sqrt	None	0.817	0.785
gini	6	0.01	log2	None	0.829	0.804
gini	6	0.01	2	None	0.823	0.792
gini	6	0	auto	None	0.828	0.800
gini	6	0	sqrt	None	0.818	0.787
gini	6	0	log2	None	0.826	0.790
gini	6	0	2	None	0.815	0.778
gini	6	0	auto	0	0.830	0.797
gini	6	0	sqrt	0	0.830	0.797
gini	6	0	log2	0	0.830	0.797
gini	6	0	2	0	0.834	0.803

As we can see from the table above, we started the experiment by comparing the Criterion function. We got the accuracy better when we use gini function instead of entropy. Focusing on gini function, we iterate to find the best number of the maximum depth of the tree, and find 6 is the best on this case. Then we tune the Min Weight value, and pick 0.0 and 0.1 to be taken for comparison in the continuing experiment. We then try some possible values for Max Features and Random State. As the result, the maximum *accuracy* after training for the case of closed door is **0.834** with *F1 score* is **0.803**, when gini criteria was used for the Gini impurity and the maximum depth of the decision tree is 6.

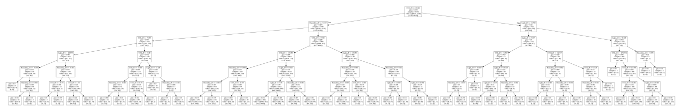
Fig. 7. Resulting Decision Tree - Closed Door

TABLE VII
ACCURACY AND F1-SCORE OF OPEN DOOR CASE

Criterion	*Max Depth*	*Min Weight*	*Max Features*	*Random State*	*Accuracy*	*F1 Score*
entropy	None	0	auto	0	0.730	0.639
gini	None	0	None	None	0.715	0.624
gini	4	0	None	None	0.796	0.667
gini	6	0	None	None	0.797	0.680
gini	7	0	None	None	0.801	0.684
gini	7	0	auto	None	0.795	0.672
gini	7	0	sqrt	None	0.791	0.665
gini	7	0	log2	None	0.795	0.676
gini	7	0	2	None	0.760	0.629
gini	7	0.5	None	None	0.798	0.673
gini	7	0.1	None	None	0.801	0.674
gini	7	0.01	None	None	0.796	0.667
gini	7	0.01	auto	None	0.781	0.659
gini	7	0.01	sqrt	None	0.784	0.660
gini	7	0.01	log2	None	0.802	0.660
gini	7	0.01	2	None	0.775	0.661
gini	7	0.01	auto	0	0.792	0.680
gini	7	0.01	sqrt	0	0.792	0.680
gini	7	0.01	log2	0	0.792	0.680
gini	7	0.01	2	0	0.784	0.646
entropy	7	0	auto	0	0.792	0.670
entropy	7	0	sqrt	None	0.797	0.670
entropy	7	0.01	log2	None	0.782	0.656

While in experiment of the open door case as shown in table above, by doing similar approach as done in previous case, the best *accuracy* obtained is **0.801** with *F1 score* is **0.660** when we use gini criterion with maximum depth is 7 and the number of maximum features is the log2 of the number of the features.

More, by comparing the ”Max Feature” values, we also see that the best accuracy is always obtained when the number of maximum features to consider when looking for the best split is set to be log2 for this dataset.

Fig. 8. Resulting Decision Tree - Open Door

IV. Conclusion

The main purpose of this research is to test different machine learning methods which can successfully predict occupancy status of an environment using the generalized version of data from [5]. Some of the crucial factors present in the original data which can give a very high predictive power, such as time of the day, is completely omitted from the data set in favor of generalization. The remaining factors have been processed for new attributes. Our hypothesis was, given a good model, Artificial Neural Networks and Support Vector Machines can reach a reliable accuracy. Our research result, which is present in the Evaluation section of the article shows that Both ANN and SVM have the capability of reaching up to 90 percent accuracy. We included prediction result using Decision Trees to compare with the original article.

Overall, despite that original authors have cast side Artificial Neural Networks and Support Vector Machines for not being accurate enough, the result of this research confirms that using generalized data to train Artificial Neural Networks and Support Vector Machines can reach a fair accuracy even without using time or other absolute attributes.

References

[1] T. Labeodan, et al, ***Occupancy measurement in commercial office buildings for demand-driven control applications - A survey and detection system evaluation.***
Journal of Energy and Buildings. 93(2015), pp. 303-314.

[2] V. L. Erickson, et al, ***OBSERVE: Occupancy-based system for efficient reduction of HVAC energy.***
Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks, 258-269 (2011).

[3] B. Dong, B. Andrews, ***Sensor-based Occupancy Behavioral Pattern Recognition For Energy And Comfort Management In Intelligent Buildings***
Eleventh International IBPSA Conference, 1444-1451 (2009).

[4] E.U, ***Directive 2010/31/EU of the European Parliament and of the Council of 19 May 2010 on the Energy Performance of Buildings (Recast)***, Official Journal of the European Union, vol. 18, no. 6, pp.1333, 2010.

[5] L. M. Candanedo, V. Feldheim, ***Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models***, Energy and Buildings. 112, (2016) 28-39.

[6] V. L. Erickson, M. . Carreira-Perpin, A. E. Cerpa, ***Occupancy Modeling and Prediction for Building Energy Management.*** ACM Transactions on Sensor Networks. 10, 1-28 (2014).

[7] S. H. Ryu, H. J. Moon, ***Development of an occupancy prediction model using indoor environmental data based on machine learning techniques.*** Building and Environment. 107, 1-9 (2016).

[8] T. H. Pedersen, K. U. Nielsen, S. Petersen, ***Method for room occupancy detection based on trajectory of indoor climate sensor data.*** Building and Environment. 115, 147-156 (2017).

[9] I. Guyon, A. Elisseeff ***An Introduction to Variable and Feature Selection*** Journal of Machine Learning Research 3 1157-1182 (2003).

[10] M. F. Zibran ***CHI-Squared Test of Independence***

[11] D. P. Kingma, J. L. Ba ***Adam: A Method for Stochastic Optimization*** International Conference on Learning Representations (2015).