

# RUS Boost Tree Ensemble Classifiers for Occupancy Detection

V. Murugananthan, Udaya Kumar Durairaj

**Abstract:** In this research paper, various ensemble classifiers are used to predict occupancy status using samples of light, temperature, humidity, CO<sub>2</sub>, humidity ratio sensor data. Occupancy detection will save energy making room for smart buildings in smart cities. It paves way to decide on heating, ventilation, cooling and lighting. To achieve 'white box' output and facilitate explanatory interpretation, decision tree was employed. Several weak learner decision trees were melded to form RUSBoosted Tree ensemble classifier. On investigation of the results, it is seen that RUSBoostedTree Ensemble gives the highest accuracy rate of 99%.

**Keywords:** Occupancy Detection, Classification, Ensemble, RUSBoosted Tree ensemble classifier, sensor data.

## I. INTRODUCTION

With the growing need for smart cities, control system is required for construction of smart buildings which saves energy and money to the tune of 20%-50% [1-4]. Collection of experimental data from the buildings reported energy savings upto 37% in [5] and between 29% and 80% [6] when the same was used as an input for HVAC (Heating, Ventilating and Air Conditioning) control algorithms [7-8]. This research has used dataset composed by samples obtained from light, temperature, humidity CO<sub>2</sub> sensors and a digital camera to establish ground occupancy for supervised classification model training.

Many supervised and unsupervised learning techniques such as Quick Propagation, Conjugate Gradient Descent, Quasi-Newton, Limited Memory Quasi-Newton, Levenberg-Marquardt, Online Back Propagation, Batch Back Propagation, SVM were dealt with. The paper is organized as follow. In [9], Decision trees were used for getting the Real time with an accuracy of 97.9% by using passive IR motion sensor.

In [10], SVM (Support Vector Machine) was used for occupancy detection with an accuracy rate of 88%. In [11], Hidden Markov Models with 73% accuracy and in [12], the RFID (Radio-frequency identification) with accuracy of 62%-88% is used in detecting occupancy. In [13], the authors reported occupancy detection accuracy between 92.2% and 98.2%.

In [14], a neural network model with CO<sub>2</sub>, sound, humidity, motion and temperature sensor data was used for occupancy detection with accuracy between 75% and 84.5%. In [15], Bayesian model used data of digital video cameras, passive infrared detection and CO<sub>2</sub> sensors with a model accuracy reduced from 70% to 11%. In [2], the researchers used data was gathered from a wireless sensor network for occupancy detection and they predicted that it is possible to save 42% of annual energy consumption. In [16], to detecting the number of occupants a model that used temperature, CO<sub>2</sub>, humidity, light, motion and sound sensors was introduced. They used neural network in MATLAB. The accuracy was 64.8%.

In section 2, occupancy detection dataset is discussed in detail. In section 3, dimension reduction is dealt with. Section 4 discusses in detail about supervised decision tree classifier for the occupancy detection. Ensemble RUSBoost tree classifier is elaborated highlighting the algorithm, prediction speed, accuracy and training rate. Confusion matrix to give the details of the true positive and false negative rates of the occupancy detection data are presented in Section 5. In section 6, short summary of the results are discussed.

## II. OCCUPANCY DETECTION DATASET

Occupancy dataset was collected from UCI Machine Learning repository: <http://archive.ics.uci.edu/ml/datasets/Occupancy+Detection>. Attributes for the occupancy includes: day information, Temperature (in °C), Light (in Lux), CO<sub>2</sub> (in ppm), Relative Humidity (in %), Humidity Ratio (in Kg water-vapor/Kg-air). The Output/target is Occupancy status (0 stands for not occupied and 1 stands for occupied). Data used in this study is 2665 with 5 attributes. There are six parameters, out of which date is index of the data and is not included in the model training data. The remaining five data are used for training the model.

Training data and testing data needs to be selected a priori. Here, in this work k-fold cross validation is used for deciding the training and testing data. k-fold cross validation is used to test machine learning model with a fixed data samples. Entire data is split into k groups randomly with 1 fold as the validation data and the remaining 4 folds combined to form the training data. In this work, k is selected as 5. For every split, model is trained with the training data and the validation data is used to get the prediction accuracy. Result is the average of the splits. Validation subsets may overlap.

**Revised Manuscript Received on June 22, 2019.**

V. Murugananthan, Lecturer, SEEMIT, Institute Technology Pertama, Mantin, Negeri Sembilan, Malaysia. mail\_muru@yahoo.com

Udaya kumar Durairaj, Lecture, FOET, Lipnton University College, Mantin, Negeri Sembilan, Malaysia. dr.udayakumar@ktg.edu.my

Algorithm for k-fold cross validation

- a. One-fifth data of 2665 (533) is selected as the validation data and 4/5 data of 2665 (2132) is chosen as the training data for a single fold.
- b. Train the model and validate the model with the test data.
- c. Compute the model error, difference between prediction responses and true responses.

**III. DIMENSION REDUCTION WITH PRINCIPAL COMPONENT ANALYSIS**

Data dimension needs to be reduced to minimize the vast data into lesser data retaining unique data and removing similar information thereby get the better feature for classification and help solve machine learning problems. Computation time is reduced by means of reducing the redundant features. Principal component analysis is used which transforms to a new variable sets. New variable set, the principle component is obtained by linear combination of the raw variables. First principle components are possible variation of raw data followed by the successive component with maximum variance. First component and the second component are orthogonal to each other. 5% reduction of redundant data is done.

**Table. 1 Accuracy, prediction speed, training time of the decision Tree models**

Classifier	Accuracy	Prediction Speed	Training Time	Maximum No. of splits
Fine Tree	98.5%	~5900 obs/sec	19.023 sec	Present : Medium Tree Maximum Number of splits : 100 Split Criterion : Gini's diversity index Surrogate decision splits : Off
Medium Tree	98.2%	~13000 obs/sec	1.41 sec	Present : Medium Tree Maximum Number of splits : 20 Split Criterion : Gini's diversity index Surrogate decision splits : Off
Coarse Tree	98.3%	~15000 obs/sec	1.23 sec	Present : Coarse Tree Maximum Number of splits : 4 Split Criterion : Gini's diversity index Surrogate decision splits : Off

On investigation of the decision trees, the maximum accuracy is 98.5% for the fine decision tree but the training time is 19 sec far higher than the medium and coarse tree which are 1.4 sec and 1.23 sec. The number of observations that can be predicted for the fine is only 5900 obs/sec (approx.) when compared to the medium and coarse tree which are ~13000 obs/sec & ~15000 obs/sec respectively. Table 2 shows the confusion matrix for the decision tree models. On analysis of the various decision trees, it is observed that TPR is 95% & >99% & PPV is 99% & 98% and minimum FNR is 5% & <1% & FDR is 1% & 2% among all the decision trees. To improve the performance

**IV. SUPERVISED DECISION TREE CLASSIFIER**

Decision tree classifier finds the relation between predictors and the target. Top down approach starting from the root down through the branches to the leaves is carried out. At root, the process starts by checking for a condition. Depending on the condition, it branches down until it reaches the leaf node. Depending on the condition of light, temperature, humidity ratio, relative humidity and CO<sub>2</sub>, the occupancy is classified.

Coarse Tree, Medium Tree & Fine Tree are used for classifying the Occupancy data. All the types of decision trees are fast with small memory usage and easy interpretability. The coarse decision tree have few leaves with maximum number of splits as four and the medium decision tree have finer distinction between classes with maximum split as twenty. Fine decision tree have the finest distinction between classes with maximum number of splits as 100. Table 1 shows the accuracy, prediction speed, training time and the maximum number of splits of the various decision tree models for the occupancy data.

even better, ensemble RUSBoost Tree is attempted. Receiver Operating characteristics (ROC) demonstrates the classifier performance by varying the threshold value. True positive rate (TPR) and false positive rate (FPR) are plotted against each other for varying threshold values.

False positive rate is the specificity. Area under the Akaike Information Criterion (AIC) curve ranks randomly chosen positive instance than the negative ones. It is an indicator of model performance. Table 3 shows the ROC and parallel co-ordinates plot for the decision tree models. Investigation of the ROC of the different decision tree models shown in table 3 shows that the performance accuracy is >98%.



Table. 2 Confusion matrix, True positive rates and Positive predictive values for the decision tree models

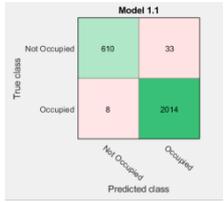
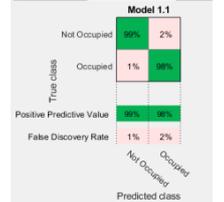
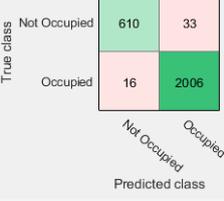
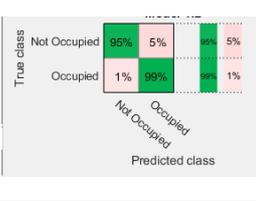
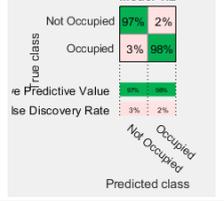
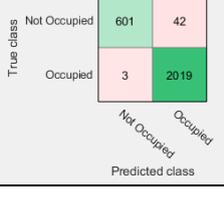
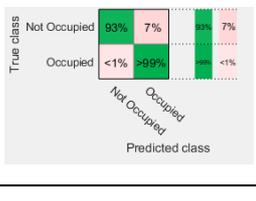
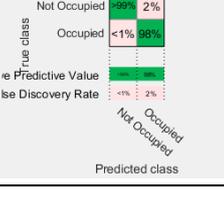
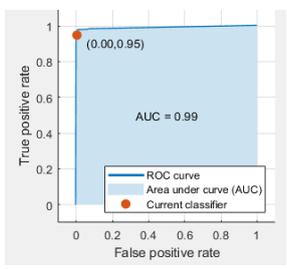
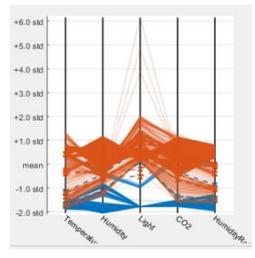
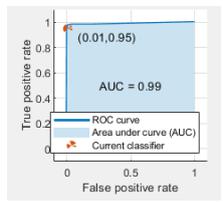
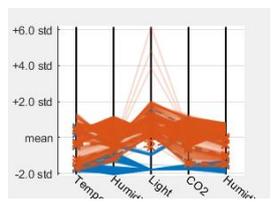
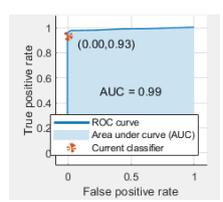
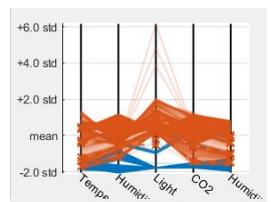
Classifier	Confusion Matrix	True Positive Rates (TPR), False Negative rates (FNR)	Positive Predictive values (PPV), False Discovery Rates (FDR)
Fine Tree			
Medium Tree			
Coarse Tree			

Table. 3 ROC and Parallel co-ordinates plot of the decision trees

Classifier	ROC Curve	Parallel Co-ordinates plot
Fine Tree		
Medium Tree		
Coarse Tree		

**V. ENSEMBLE RUSBOOST TREE CLASSIFIER**

Ensemble classifiers combines power of individual classifiers. Ensemble for Decision trees are most promising classifiers. Normally Decision trees are unstable and ensemble can eliminate this problem. Multiple classifiers are applied and weighted and combined to give superior performance compared to individual classifiers. There are different types of decision tree ensembles: boosting, bagging and random forest. Boosting is the popular decision tree ensemble. Distributed training data are run repeatedly on weak learners and combined into a strong classifier with high accuracy compared to individual tree. RUSBoost (Random Under Sampling) algorithm is applicableunequal group of data.

**RUSBoost Algorithm**

Given : Set S of examples  $(x_1, y_1), \dots, (x_m, y_m)$  with minority class  $y^r \in |Y| = 2$   
 Weak learner (decision tree), WeakLearn  
 Number of iterations, T  
 Desired percentage of total instances to be represented by the minority class, N

1. Initialise  $D_1(i) = 1/m$  for all i
2. Do For  $t = 1, 2, \dots, T$ 
  - a. Create temporary training dataset  $S_t'$  with distribution  $D_t'$  using random under-sampling
  - b. Call WeakLearn, providing it with examples  $S_t'$  and their weights  $D_t'$ .
  - c. Get back a hypothesis  $h_t: X \times Y \rightarrow [0, 1]$
  - d. Calculate a pseudo-loss (for S and  $D_t$ ):
 
$$\epsilon_t = \sum_{(i,y), y_i \neq y} D_t(i)(1 - h_t(x_i, y_i) + h_t(x_i, y))$$
  - e. Calculate the weight update parameter:
  - f. Create temporary training dataset  $S_t'$  with distribution  $D_t'$  using random under-sampling

- g. Call WeakLearn, providing it with examples  $S_t'$  and their weights  $D_t'$ .
- h. Get back a hypothesis  $h_t: X \times Y \rightarrow [0, 1]$
- i. Calculate a pseudo-loss (for S and  $D_t$ ):

$$\epsilon_t = \sum_{(i,y), y_i \neq y} D_t(i)(1 - h_t(x_i, y_i) + h_t(x_i, y))$$

- j. Calculate the weight update parameter:

$$\alpha_t = \frac{\epsilon_t}{1 - \epsilon_t}$$

- k. Update  $D_t'$ :

$$D_{t+1}(i) = D_t(i)\alpha_t^{2^{1+(h_t(x_i, y_i) - h_t(x_i, y; y \neq y_i))}}$$

- l. Normalise  $D_{t+1}$  :

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t}$$

3. Output the final hypothesis:

$$H(x) = \arg \max_{y \in Y} \sum_{t=1}^T h_t(x, y) \log \frac{1}{\alpha_t}$$

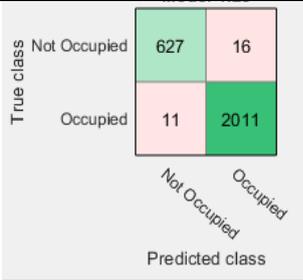
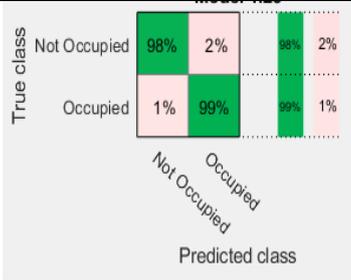
RUSBoost (adapted from Seiffert et al. 2010)

Number of Weak learners for the ensemble is 30. Table 4 shows the accuracy, prediction speed and training time for RUSBoosted Tree ensemble. Accuracy is 99% and is higher than the fine decision tree. Training time is found to be 10.701 and the number of samples that can be predicted per sec is approximately 2400 observations. Table 5 shows the confusion matrix, true positive & false negative rates, positive predictive values, false discovery rates of the ensemble RUSBoosted Tree. On investigation of the confusion matrix, it is seen that out of 2665 samples, 2638 samples are classified correctly with TPR and PPV 98% & 99%.

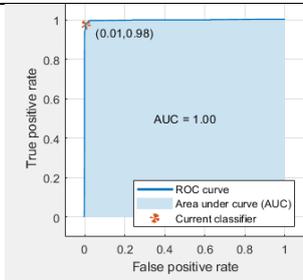
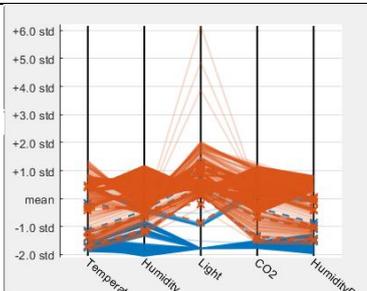
**Table. 4 Accuracy, training rate , prediction speed of Ensemble RUSBoosted Tree**

Classifier	Accuracy	Prediction Speed	Training Time	Maximum No. of splits
Ensemble: RUSBoosted Trees	99.0%	~2400 obs/sec	10.701 sec	No. of Learners : 30 Learning Rate : 0.1

**Table . 5 Confusion matrix, True positive rates, false negative rates, positive predictive values, false discovery rate of Ensemble RUSBoosted Tree**

Classifier	Confusion Matrix	True Positive Rates, False Negative rates	Positive Predictive values, False Discovery Rates
Ensemble: RUSBoosted Trees			

**Table. 6 ROC and parallel co-ordinate plots**

Classifier	Confusion Matrix	True Positive Rates, False Negative rates
Ensemble: RUSBoostedTrees		

**VI. CONCLUSION**

Occupancy detection is classified using decision tree methods such as fine tree, medium tree and coarse tree and validated. Among all the decision trees, fine decision tree model has an accuracy of 98.5%. Ensemble RUSBoosted Tree combines weak learners to produce a strong learner. The work shows the feasibility of RUSBoost Tree ensemble methods to improve performance of the classifier.

**REFERENCES**

1. Shen W., Newsham G, Implicit Occupancy Detection for Energy Conversation in Commercial Buildings: A Survey, Submitted to CSCWD 2016,2016
2. V.L. Erickson, M.Á. Carreira-Perpiñán, A.E. Cerpa, OBSERVE: Occupancy-based system for efficient reduction of HVAC energy, in: Proceedings of the 10th International Conference on, IEEE, Information Processing in Sensor Networks (IPSN), Chicago, IL, 2011, pp. 258– 269.
3. V.L. Erickson, M.Á. Carreira-Perpiñán, A.E. Cerpa, Occupancy modeling and prediction for building energy management, ACM Trans. Sensor Netw. (TOSN)10 (3) (2014) 42.
4. Dong B., Andrews B., (2009). Sensor-based occupancy behavioral pattern recognition for energy and comfort management in intelligent buildings. Proceedings of Building Simulation.
5. J. Brooks, S. Goyal, R. Subramany, Y. Lin, T.Middelkoop, L. Arpan, L. Carloni, P.Barooah, An experimental investigation of occupancy-based energy-efficient control of commercial building indoor climate, in: Proceeding of the IEEE 53rd Annual Conference on, IEEE, Decision and Control (CDC), Los Angeles, CA, 2014, pp.5680–5685.

6. J. Brooks, S. Kumar, S. Goyal, R. Subramany, P.Barooah, Energy-efficient control of underactuated HVAC ones in commercial buildings, Energy Build. 93 (2015) 160–168.
7. Shen W., Newsham G, Smart Phone Based Occupancy Detection in Office Buildings, Proceedings of the 2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design, pp. 632,636
8. Candanedo L.M., Feldheim V., 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
9. E. Hailemariam, R. Goldstein, R. Attar, A. Khan, Real-time occupancy detection using decision trees with multiple sensor types, in: Proceedings of the 2011 Symposium on Simulation for Architecture and Urban Design, Society for Computer Simulation International, San Diego, CA, 2011, pp. 141–148.
10. A. Ebadat, G. Bottegal, D. Varagnolo, B. Wahlberg, K.H. Johansson, Estimation of building occupancy levels through environmental signals deconvolution, in: Proceedings of the 5th ACM Workshop on Embedded Systems For Energy- Efficient Buildings, ACM, Rome, Italy, 2013, pp. 1–8.
11. B. Dong, B. Andrews, K.P. Lam, M. Höyneck, R. Zhang, Y.-S. Chiou, D. Benitez, An information technology enabled sustainability test-bed (ITEST) for occupancy detection through an environmental sensing network, Energy Build. 42 (7) (2010) 1038–1046.
12. N. Li, G. Calis, B. Becerik-Gerber, Measuring and monitoring occupancy with an RFID based systemfor demand-driven HVAC operations, Automat. Construct. 24 (2012) 89–99.



13. Z. Yang, N. Li, B. Becerik-Gerber, M. Orosz, A systematic approach to occupancy modeling in ambient sensor-rich buildings, *Simulation* 90 (8) (2014) 960–977.
14. T. Ekwevugbe, N. Brown, V. Pakka, D. Fan, Real-time building occupancy sensing using neural-network based sensor network, in: 7th IEEE International Conference on IEEE, Digital Ecosystems and Technologies (DEST), Menlo Park, California, 2013, pp. 114–119.
15. S. Meyn, A. Surana, Y. Lin, S.M. Oggianu, S. Narayanan, T.A. Frewen, A sensor-utility-networkmethod for estimation of occupancy in buildings, in: Decision and Control, 2009 held jointly withthe 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on, IEEE, Shanghai, P.R. China, 2009, pp. 1494–1500.
16. Z. Yang, N. Li, B. Becerik-Gerber, M. Orosz, A multi-sensor based occupancy estimation modelfor supporting demand driven HVAC operations, in: Proceedings of the 2012 Symposium on Simulation for Architecture and Urban Design, Society for Computer Simulation International, San Diego, CA, USA, 2012, pp. 49–56.