engineering proceedings

MDPI

Proceeding Paper

Machine Learning for Accurate Office Room Occupancy Detection Using Multi-Sensor Data †

Yusuf Ibrahim [1],*, Umar Yusuf Bagaye [2] and Abubakar Ibrahim Muhammad [2]

[1] Department of Computer Engineering, Ahmadu Bello University, Zaria 810211, Nigeria
[2] Department of Electrical and Electronics Engineering, Kaduna Polytechnic, Kaduna, Nigeria; uybagaye@gmail.com (U.Y.B.); ibngarko@yahoo.com (A.I.M.)
* Correspondence: yibrahim@abu.edu.ng or yusuf2007ee@gmail.com

† Presented at the 10th International Electronic Conference on Sensors and Applications, 15–30 November 2023; Available online: https://ecsa-10.sciforum.net/.

Abstract: In this paper, we present a comparative study of several machine learning (ML) approaches for accurate office room occupancy detection through the analysis of multi-sensor data. Our study utilizes the Occupancy Detection dataset, which incorporates data from Temperature, Humidity, Light, and CO_2 sensors, with ground-truth labels obtained from time-stamped images captured at minute intervals. Traditional ML techniques including Decision Trees (DT), Gaussian Naïve Bayes (NB), K-Nearest Neighbors (KNN), Logistic Regression (LR), Support Vector Machines (SVM), Multilayer Perceptron (MLP), and Quadratic Discriminant Analysis (QDA) are compared alongside advanced ensemble methods like RandomForest (RF), Bagging, AdaBoost, GradientBoosting, ExtraTrees as well as our custom voting and multiple stacking classifiers. Also, hyperparameter optimization was performed for selected models with a view to improving classification accuracy. The performances of the models were evaluated through rigorous cross-validation experiments. The results obtained highlight the efficacy and suitability of varying candidate and ensemble methods, demonstrating the potential of ML techniques in enhancing the detection accuracy. Notably, LR and SVM exhibited superior performance, achieving average accuracies of 98.88 ± 0.70% and 98.65 ± 0.96%, respectively. Additionally, our custom voting and stacking ensembles demonstrated improvements in classification outcomes compared to base ensemble schemes, as indicated by various evaluation metrics.

Keywords: machine learning; ensemble learning; room occupancy detection; multi-sensor data

Citation: Ibrahim, Y.; Bagaye, U.Y.; Muhammad, A.I. Machine Learning for Accurate Office Room Occupancy Detection Using Multi-Sensor Data. *Eng. Proc.* **2023**, *56*, x. https://doi.org/10.3390/xxxxx

Academic Editor(s): Name

Published: 15 November 2023

1. Introduction

Occupancy detection refers to the process of determining whether a space or area is currently occupied by people or objects. This can be accomplished through various means and technologies, and serves several purposes in different domains, including building management, safety, security, energy conservation, and automation. For instance, efficient energy management in office spaces is today a concern, where environmental sustainability and cost-effectiveness go hand in hand. Estimates indicate that precise office room occupancy detection can lead to energy savings ranging from 30% to 42% [1,2]. These savings can be further optimized, reaching up to 80%, when occupancy data is integrated into HVAC (Heating, Ventilation, and Air Conditioning) control algorithms [3]. Therefore, there is a growing need for accurate occupancy detection methods to harness the full potential of these energy-saving opportunities. This quest for precision in occupancy detection has led to substantial research efforts, especially in the application of ML models. Previous studies have shown that, with sufficient relevant data, the accuracy of occupancy detection can yield remarkable performance level [4–6]. In this paper, we utilize multi-sensor data which is becoming increasingly popular in ML applications as it can provide more accurate and reliable results compared to using a single sensor. The significant contributions of this paper include:

1. Systematic comparison of a wide range of ML models, from traditional to advanced ensemble methods.
2. Optimizing hyperparameters of selected models in order to enhance performance.
3. Evaluating a custom voting and multiple stacking classifiers and demonstrating their role in improving classification performance.

2. Related Work

Several ML-based data-driven techniques have been utilized for occupancy detection in buildings. Candanedo and Feldheim [7] assessed the accuracy of predicting office room occupancy based on data from light, temperature, humidity, and CO_2 sensors, using various statistical classification models in R programming language. They used three datasets for training and testing, considering whether the office door was open or closed during occupancy. The best accuracies (ranging from 95% to 99%) were achieved with Linear Discriminant Analysis (LDA), Classification and Regression Trees (CART), and RF models. The inclusion of time stamp information generally improved accuracy, and the LDA model achieved occupancy estimates of 85% and 83% using only the temperature predictor in two different testing sets. In their study, Yang et al. [8] employed KNN alongside various environmental and specialized sensors for identifying and counting the number of occupants. Their findings show the potential to attain accuracy levels ranging from 95.4% to 97.5% for binary occupancy detection. Additionally, when estimating the count of occupants, the root mean square error (RMSE) falls within the range of 0.121 to 0.79. Dong et al. [9] and Lam et al. [10] paired SVM with a sensor network to gauge the occupancy levels within an office building. Their investigations yielded a consistent accuracy rate of approximately 75% for detecting the number of people present. Zuraimi et al. [11] utilized a combination of CO_2 data and feed forward neural networks (FFNNs) to estimate the number of occupants in a theater resulting in an average accuracy of 70%. Similarly, Dong et al. [9] Lam et al. [10] introduced an environmental sensor network test-bed and demonstrated its utilization for detecting occupancy numbers within an office building. Their works employed a neural network to identify the number of occupants, achieving an accuracy rate of 75%. In their study, Kraipeerapun et al. [12] introduced two approaches for determining occupancy. The initial approach employed a combination of stacking and a multiclass neural network, while the second method fused stacking with a dual-output neural network specifically designed for occupancy detection. The validation outcomes demonstrated accuracy levels ranging from 68.87% to 91.18%. Kim et al. [2] introduced a label noise filtering method, which improves occupancy detection accuracy by eliminating noisy data collected from sensors. The results yielded an average accuracy increase of 1.5%, with the CART model showing a significant improvement from 94.3% to 97.6%. Dutta and Roy [13] developed the OccupancySense model which addresses occupancy detection and prediction by fusing Internet of Things (IoT) indoor air quality data with static and dynamic context data, achieving higher forecasting accuracy using CatBoost algorithm. The model outperforms other ML algorithms, and with a non-intrusive approach, accurately detecting occupancy, predicting headcount, and estimating room occupancy density at 99.85%, 93.2%, and 95.6% accuracy respectively. Elkhoukhi et al. [14] highlights the limitations of batch learning techniques and introduces three non-stationary ML algorithms for stream data processing. The experimental results demonstrate that these algorithms, integrated into an IoT-based platform, can accurately predict the number of occupants in smart buildings with an accuracy exceeding 83% while efficiently utilizing computational resources.

3. Materials and Methods

3.1. Data Collection and Preprocessing

We utilized the publicly available Occupancy Detection dataset, which includes sensor data from Temperature, Humidity, Light, and CO_2 sensors, as well as ground-truth labels obtained from time-stamped images captured at minute intervals [7].

3.2. Feature Engineering

We performed correlation analysis on the dataset to identify relevant features for the occupancy detection task. Most features have strong positive correlations with the target variable (occupancy) except for humidity and humidity ratio with relatively week correlation with the target variable. However, we retained all features without thresholding any sensor data in our analysis. Truncating below a specific threshold and trying other feature combinations is left for future research.

3.3. Model Selection

For our analysis, we selected a set of traditional ML as well as advanced (ensemble) models for the comparative study. The traditional ML models include Decision Trees, Gaussian Naïve Bayes, KNN, LR, SVM, MLP, QDA. The Ensemble methods include RF, Bagging, AdaBoost, GradientBoosting and ExtraTrees. Furthermore, we tried several Custom ensemble methods as follows:

1. Voting Classifier, consisting of LR, RF, and SVM
2. StackingClassifier1, consisting of LR, RF, and SVM as base estimators with LR as the final estimator
3. StackingClassifier2, consisting of Decision Tree, KNN, and MLP Classifiers as base estimators with LR as the final estimator
4. Stacking Classifier3, consisting of GaussianNB, SVM, and QDA as base estimators with LR as the final estimator
5. StackingClassifier4, consisting of RF, MLP Classifier, and SVM as base estimators with LR as the final estimator

3.4. Hyperparameter Optimization

In order to get better performance, we further performed parameter tuning via grid search for RF, SVM, and KNN classifiers. Each grid search was performed with 5-fold cross-validation. For RF, the search was conducted over: number of estimators (10, 20, 30), maximum depth (15, 20, 30, 50) and criterion (gini, entropy). Also, the SVM was tuned over: C (1, 10, 100) and kernel types (linear, poly, rbf, sigmoid). Finally, KNN was optimized by searching for the optimal number of neighbors (2, 3, 5, 10, 15, 20).

3.5. Model Training and Evaluation

Rigorous cross-validation experiments (using 5-fold cross-validation) were performed in order to assess the performance of the models. We then split the dataset into 70% training and 30% testing and retrained each model on the training set and evaluate the models' performance on the test set using accuracy, precision, recall, and F1-score as performance metrics.

4. Results and Discussions

Tables 1 and 2 respectively show the 5-fold cross validation as well as the testing results for the Traditional ML models while Tables 3 and 4 respectively show the 5-fold cross validation as well as the testing results for the ensemble models.

Table 1. Cross-validation results for the Traditional ML methods.

Model	Average Accuracy	Average Precision	Average Recall	Avergae F1-Score
SVM	0.9865 ± 0.0096	0.9500 ± 0.0368	0.9958 ± 0.0016	0.9720 ± 0.0193
LR	0.9888 ± 0.0070	0.9582 ± 0.0280	0.9958 ± 0.0029	0.9764 ± 0.0144
KNN	0.9639 ± 0.0102	0.9283 ± 0.0145	0.9151 ± 0.0582	0.9204 ± 0.0245
DT	0.8363 ± 0.1437	0.7511 ± 0.2595	0.8419 ± 0.1246	0.7477 ± 0.1506
NB	0.9368 ± 0.0249	0.7915 ± 0.0654	0.9983 ± 0.0011	0.8814 ± 0.0410
MLP	0.9699 ± 0.0162	0.9375 ± 0.0504	0.9377 ± 0.0806	0.9340 ± 0.0375
QDA	0.9482 ± 0.0393	0.8359 ± 0.1150	0.9954 ± 0.0024	0.9042 ± 0.0680

Table 2. Results of the Traditional ML methods on test set.

Model	Accuracy	Precision	Recall	F1-Score
SVM	0.9906	0.9644	0.9958	0.9798
LR	0.9904	0.9650	0.9943	0.9794
KNN	0.9908	0.9735	0.9866	0.9800
DT	0.9911	0.9809	0.9802	0.9805
NB	0.9668	0.8748	0.9979	0.9323
MLP	0.9531	0.8311	0.9986	0.9072
QDA	0.9825	0.9342	0.9936	0.9630

Table 3. Cross-validation results for the ensemble methods.

Model	Accuracy	Precision	Recall	F1-Score
RF	0.8589 ± 0.1249	0.7783 ± 0.2526	0.8703 ± 0.1073	0.7797 ± 0.1361
Bagging	0.9337 ± 0.0167	0.8925 ± 0.0908	0.8360 ± 0.1174	0.8516 ± 0.0424
AdaBoost	0.9352 ± 0.0283	0.8180 ± 0.1023	0.9530 ± 0.0291	0.8754 ± 0.0479
GBoosting	0.9552 ± 0.0200	0.8962 ± 0.0949	0.9322 ± 0.0435	0.9084 ± 0.0326
ExtraTrees	0.9140 ± 0.0266	0.8075 ± 0.0496	0.8311 ± 0.1465	0.8115 ± 0.0740
Voting	0.9861 ± 0.0096	0.9488 ± 0.0368	0.9954 ± 0.0018	0.9711 ± 0.0194
Stacking1	0.9889 ± 0.0072	0.9593 ± 0.0285	0.9952 ± 0.0026	0.9767 ± 0.0149
Stacking2	0.9765 ± 0.0119	0.9359 ± 0.0473	0.9682 ± 0.0366	0.9505 ± 0.0244
Stacking3	0.9880 ± 0.0082	0.9579 ± 0.0335	0.9933 ± 0.0051	0.9749 ± 0.0168
Stacking4	0.9874 ± 0.0098	0.9541 ± 0.0385	0.9952 ± 0.0028	0.9738 ± 0.0199

Table 4. Experimental results of the Ensemble methods on test set.

Model	Accuracy	Precision	Recall	F1-Score
RF	0.9935	0.9838	0.9880	0.9859
Bagging	0.9914	0.9823	0.9802	0.9812
AdaBoost	0.9903	0.9675	0.9908	0.9790
GBoosting	0.9908	0.9709	0.9894	0.9800
ExtraTrees	0.9932	0.9858	0.9844	0.9851
Voting	0.9906	0.9657	0.9943	0.9798
Stacking1	0.9932	0.9824	0.9880	0.9852
Stacking2	0.9921	0.9789	0.9866	0.9827
Stacking3	0.9885	0.9577	0.9936	0.9754
Stacking4	0.9930	0.9811	0.9887	0.9849

From Table 1, LR and SVM achieved the highest validation accuracies of 98.88% and 98.65%, respectively. These models also demonstrated strong precision, recall, and F1-Score values, indicating their suitability for accurate occupancy detection. Also, for the test data (Table 2), SVM, LR, KNN, and DT models exhibit high accuracy levels above 99%. The ensemble methods (Table 3), particularly our voting and stacking models, show high performance, with stackingclassifer1 achieving the highest validation accuracy of approximately 98.89 ± 0.72% outperforming others. Classification results on the test data (Table 4) indicate that most ensemble methods achieve high accuracy levels, with RF, ExtraTrees and StackingClassifier1 being particularly notable achieving above 99.30% accuracy. These models also exhibit strong precision, recall, and F1-Score values, reflecting their effectiveness in making accurate predictions. Also, the voting ensemble which recorded a slightly lower accuracy, still demonstrates a good balance between precision and recall. For the optimized models, we finally arrived at the following as the best hyperparameters for the respective algorithms: SVM (C = 10 and kernel = 'linear'), KNN (n_neighbors = 20), RF (n_estimators = 50, max_depth = 44, and criterion = 'entropy'). Utiliziing these parameters, the test results presented in Table 5 were obtained. The performance improvements recorded for KNN and RF show that hyperparameter optimization can improve the predictive accuracy of ML classifiers.

Table 5. Experimental results of the Optimized methods on test set.

Model	Accuracy	Precision	Recall	F1-Score
Grid-SVM	0.9887	0.9683	0.9957	0.9818
Grid-KNN	0.9920	0.9703	0.9946	0.9823
Grid-RF	0.9939	0.9807	0.9924	0.9865

5. Conclusions

In conclusion, this paper has presented a comparative study of ML approaches for office room occupancy detection using multi-sensor data. Our findings indicate that LR and SVM achieved impressive performance. Furthermore, our custom stacking ensembles demonstrated significant improvements over most base ensemble schemes. The study provides a comprehensive insight on the potential of several ML techniques in the domain of room occupancy detection.

Author Contributions: Conceptualization, Y.I. and A.I.M.; methodology, Y.I. and U.Y.B.; software, Y.I and U.Y.B.; validation, U.Y.B. and A.I.M.; Y.I.; investigation, Y.I.; resources, Y.I. and U.Y.B.; data curation, Y.I.; writing—original draft preparation, Y.I.; writing—review and editing, U.Y.B. and A.I.M.; visualization, A.I.M..; supervision, Y.I.; project administration, A.I.M..; funding acquisition, Y.I., U.Y.B. and A.I.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this study was obtained from the UCI Machine Learning Repository (https://doi.org/10.24432/C5X01N). The dataset is publicly available and can be accessed at https://archive.ics.uci.edu/dataset/357/occupancy+detection.

Acknowledgments: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Erickson, V.L.; Carreira-Perpiñán, M.Á.; Cerpa, A.E. Occupancy modeling and prediction for building energy management. *ACM Trans. Sens. Netw. (TOSN)* **2014**, *10*, 1–28. https://doi.org/10.1145/2594771.
2. Kim, Y.-M.; Lee, Y.-H.; Pyo, C.-S. Accurate Occupancy Detection via Label Noise Filtering Technique. In Proceedings of the 2020 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, South Korea, 21–23 October 2020; pp. 1381–1383.
3. Brooks, J.; Kumar, S.; Goyal, S.; Subramany, R.; Barooah, P. Energy-efficient control of under-actuated HVAC zones in commercial buildings. *Energy Build.* **2015**, *93*, 160–168. https://doi.org/10.1016/j.enbuild.2015.01.050.

4. Zemouri, S.; Gkoufas, Y.; Murphy, J. A machine learning approach to indoor occupancy detection using non-intrusive environmental sensor data. In Proceedings of the Proceedings of the 3rd International Conference on Big Data and Internet of Things, Melbourn, Australia, 22–24 August 2019; pp. 70–74.
5. Zhao, H.; Hua, Q.; Chen, H.-B.; Ye, Y.; Wang, H.; Tan, S.X.-D.; Tlelo-Cuautle, E. Thermal-sensor-based occupancy detection for smart buildings using machine-learning methods. *ACM Trans. Des. Autom. Electron. Syst. (TODAES)* **2018**, *23*, 1–21. https://doi.org/10.1145/3200904.
6. Toutiaee, M. Occupancy detection in room using sensor data. *arXiv* **2021**, arXiv:2101.03616 https://doi.org/10.48550/arXiv.2101.03616.
7. Candanedo, L.M.; Feldheim, V. Accurate occupancy detection of an office room from light, temperature, humidity and CO_2 measurements using statistical learning models. *Energy Build.* **2016**, *112*, 28–39. https://doi.org/10.1016/j.enbuild.2015.11.071.
8. Yang, Z.; Li, N.; Becerik-Gerber, B.; Orosz, M. A systematic approach to occupancy modeling in ambient sensor-rich buildings. *Simulation* **2014**, *90*, 960–977. https://doi.org/10.1177/0037549713489918.
9. Dong, B.; Andrews, B.; Lam, K.P.; Höynck, M.; Zhang, R.; Chiou, Y.-S.; Benitez, D. An information technology enabled sustainability test-bed (ITEST) for occupancy detection through an environmental sensing network. *Energy Build.* **2010**, *42*, 1038–1046. https://doi.org/10.1016/j.enbuild.2010.01.016.
10. Lam, K.P.; Höynck, M.; Dong, B.; Andrews, B.; Chiou, Y.-S.; Zhang, R.; Benitez, D.; Choi, J. Occupancy detection through an extensive environmental sensor network in an open-plan office building. **2009**.
11. Zuraimi, M.; Pantazaras, A.; Chaturvedi, K.; Yang, J.; Tham, K.; Lee, S. Predicting occupancy counts using physical and statistical Co_2-based modeling methodologies. *Build. Environ.* **2017**, *123*, 517–528. https://doi.org/10.1016/j.buildenv.2017.07.027.
12. Kraipeerapun, P.; Amornsamankul, S. Room occupancy detection using modified stacking. In Proceedings of the Proceedings of the 9th International Conference on Machine Learning and Computing, Singapore, 24–26 February 2017; pp. 162–166.
13. Dutta, J.; Roy, S. OccupancySense: Context-based indoor occupancy detection & prediction using CatBoost model. *Appl. Soft Comput.* **2022**, *119*, 108536. https://doi.org/10.1016/j.asoc.2022.108536.
14. Elkhoukhi, H.; Bakhouya, M.; El Ouadghiri, D.; Hanifi, M. Using stream data processing for real-time occupancy detection in smart buildings. *Sensors* **2022**, *22*, 2371. https://doi.org/10.3390/s22062371.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.