Evaluation of Deep Learning and Machine Learning Algorithms for Building Occupancy Classification on Open Datasets

Georgiana Cretu, Iulia Stamatescu and Grigore Stamatescu

Abstract— **Accurately estimating and forecasting building occupancy represents an important tasks for higher level indoor energy management and control routines. Extended availability of public and open datasets reflecting indoor conditions through various sensor measurement and indirect proxies of human activity enable reliable benchmarking of new techniques for pre-processing and learning of occupancy patterns. In this work we present a comparative study between deep learning, such as convolutional neural networks, and conventional machine learning approaches, such as decision trees and random forests, on an a reference occupancy dataset. The various design decision and parametrisation options are discussed. The building occupancy classification task involves generating model outputs for various discrete occupancy categories. Standardised metrics such as accuracy, precision, recall and the F1-score are used for replicable benchmarking of the results. Main finding of the study is that, though generally the deep learning methods offer better overall results, the addition of relevant features (sensors) to the input dataset can yield better results for the conventional machine learning models with significantly lower training time and model size. This results in suitable, fast-inference, models for embedded deployment in physical proximity to the process.**

I. INTRODUCTION

As the built environment emerges as a key driver of energy usage and carbon emissions in the developed and developing world, new advanced algorithms play an role in energy management for building automation systems [1]. The share of energy consumed in building is steadily growing and thus intelligent monitoring and control solutions, deployed at scale, offer compelling energy savings and return on investment for both new and existing buildings, by retrofitting.

Reliable collection of large quantities of sensor measurements requires both intelligent sensors that can communicate their values in real time to distributed control units and suitable data infrastructures to store, visualize and stream available data. On-line learning algorithms can, in turn, leverage this data to produce short- and long-term occupancy predictions which allow to optimise the control logic that generates an optimal trade-off between occupant comfort and energy efficiency [2]. An important factor to consider is that most of the methods discussed in the literature use indirect and non-intrusive building occupancy detection rely on ambient measurements or on proxy metrics of human activity such as electricity usage, movement, door and window status, to infer occupancy levels of indoor spaces.

Machine learning models have the ability to extract patterns from representative samples of input data in both regression and classification applications. Conventional models for supervised learning such as linear and logistic regression, decision trees and random forests, support vector machines and others aim to associate the variations in features of the input data with output values or labels. Unsupervised learning methods can operate without output labels and try to group the input data into clusters, which can also then be used for two-stage learning in a supervised manner. Given recent advances in both algorithms and computing resources, highly complex, black-box models have been developed in the form of deep neural networks. As the data is processed by the network, from input to output, gradually higher level features are produces. Such examples include fully connected networks with large number of layers, recurrent neural networks, suitable for time series data, and convolutional neural networks. New developments in various sectors such as medical and industrial concern the study of explainable artificial intelligence methods (xAI) which aim to mitigate this drawback of deep networks in areas where the traceability of the output decision is key.

The application of these methods in the built environment has been driven by the dense spatial and temporal instrumentation of modern buildings which makes available large quantities of streaming data from sensors, actuators and controllers that monitor and operate various subsystems of the building. These include several functions such as heating, ventilation and air conditioning (HVAC), lighting, security and access control and others. Taking into account the current technical state-of-the-art and the building occupancy context, main contributions of the work are summarised next:

- A meta-study of recent open datasets for building sensor data with ground truth information, that can be used by occupancy estimation and forecasting algorithms;
- A comparative quantitative evaluation of parametrised deep learning and machine learning algorithms on a building occupancy classification task.

The rest of the paper is structured as follows. Section II discusses several available open datasets for building occupancy modelling and highlights the main points in selecting the types of measurements and appropriate context. Section III presents the chosen learning algorithms including the various design and parametrisation options. Experimental results are presented in Section IV, accounting for the varying choice of input size (number of sensors), algorithm type and performance metrics. Section V concludes the paper, with outlook of future work.

The authors are with the Department of Automation and Industrial Informatics, University Politehnica of Bucharest, 060042 Bucharest, Romania. {georgiana.cretu, iulia.stamatescu, grigore.stamatescu}@upb.ro

II. RELATED WORK

Smart buildings use Internet of Things (IoT) systems to monitor the environment and automatically control various functional subsystems, by collecting representative data at a high temporal and geographical scale. One of these types of systems is occupancy detection, which is very important because it can significantly reduce the energy emissions produced by a building. There are several ways to detect occupancy in buildings, including the use of environmental sensors (passive) with neural networks ([3], [4], [5]), monitoring the MAC and IP address of occupants in office buildings and using information about keyboard or mouse usage ([6]), using software defined radio ([7]), etc. In order to maximize the accuracy of detection while minimizing implementation costs, most systems use "passive" sensors in combination with machine learning models and neural networks. The use of such passive sensors has also been motivated by the fact that these systems do not raise issues of confidentiality and end-user privacy. The most common sensors used in occupancy detection applications in buildings are: temperature (measured in degrees Celsius °C), humidity (rH%), light (lux), sensors that measure CO2 concentration (parts-per-million), barometric pressure (Pa), indoor volatile organic compounds - VOC (parts-per-billion), etc. [8]

[3] presents the collection of a database with 34 sets of data collected from 15 countries and 39 institutions in 10 climate zones, representing both residential and commercial buildings. This database includes information about occupancy (presence and number of people) and occupant behavior (device and equipment usage). The data sets are collected at different times depending on the information obtained using various sensors:

- The state of the doors (OPEN/CLOSED) using magnetic cable sensors;
- The state of the fan (ON/OFF);
- HVAC measurements;
- Occupancy and number of occupants using cameras and passive infrared sensors for real detection;
- Plug load measured in power [w];
- The state of the windows (OPEN/CLOSED) using magnetic cable sensors;
- The state of the lights (ON/OFF);
- Indoor environment information (temperature [°C], humidity RH [%], CO2 concentration [ppm], pressure [Pa], etc.) and exterior (temperature [°C], humidity RH [%], wind speed [m/s], wind direction [deg], solar radiation [w/m2], precipitation [0/1]).

In [4], a study was conducted on the occupancy in six residential buildings in Boulder, Colorado. Measurements were taken both inside, at a frequency of 10 seconds, including information about the real presence of occupants. The measurements made in this study was collected with a mobile human presence detection system (HPDmobile) and contain information about: occupancy [0/1] and the occupant number, indoor environmental information (temperature ambient air [°C], room air relative humidity (rH) [%], CO2 concentration [ppm], indoor total volatile organic compounds (TVOC) [ppb], the light level in illuminance [lux]) and Audio Media information. The data collection was done using five sensor hubs (each containing passive sensors, a camera and a microphone), a computer with a Linux based operating system and a wireless router. Depending on the size of the living space, the number of sensor hubs deployed in a home ranged from four to six, and the location of these was influenced by the most frequent use patterns of the home. Also, the hubs were placed either near doors or in front of front doors and in kitchens, living rooms and dining rooms.

Another collection of data relevant to predicting occupancy in buildings is [5]. The data set contains 40,000 measurements taken at a frequency of one minute and combines information from the camera, representing real data from occupants, with information from environmental sensors. It is the first robust study of occupancy by considering multimodal inputs to a single output regression model. Using this data set with Random Forest resulted in an accuracy between 99.35% and 99.7%. The room where the measurements were taken is intended for socialization, and measures such as temperature, lighting, relative humidity, and CO2 levels are monitored. Real occupancy is measured using video cameras. Temperature and humidity are measured using a DHT22 sensor and an Arduino Uno board positioned 0.5 meters from the occupants. Lighting, relative humidity, and CO2 levels are measured using a Raspberry PI board and specific sensors located 2 meters from the occupants: a CO2 sensor, a thermal camera. In the case of applications for detecting occupancy in buildings, regardless of the algorithm used, the robustness, complexity, power consumption, and costs of collecting the data sets used are essential factors in the accuracy of the prediction.

In previous work, [9] and [10], a wireless non-intrusive and privacy preserving system has been designed and evaluated. It uses a thermal infrared array sensor to collect ground truth occupancy patterns using thermal footprints of persons entering or exiting a certain space. A method for using and bridging heterogenous datasets from multiple sources in order to increase the quantity of training data available for the machine learning pipeline has been described in [11]. The importance of data pre-processing for dimensionality reduction is discussed in [12], applied to indirect occupancy estimation from building ventilation system units.

III. METHODS

Our application aims to provide a discrete class label indicating the occupancy level of a building using indirect measurements from ambient sensors and contextual information. The resulting algorithm produces four probability scores, each corresponding to one of four output classes: empty (E), low (L), medium (M), and high (H). The development process involved several stages, including:

- Creating an algorithm that generates four probability scores, each associated with an output class.
- Using indirect measurements from ambient sensors and contextual information to train and test the algorithm.

- Identifying the most important features and variables that influence the occupancy level classification.

The indirect measurements are from Margarite Jacoby's work (HPDmobile), which presents the extraction of data from six residential houses, captured every ten seconds for a period of one year. The collection data contains information from the following sensors: room temperature (°C), room relative humidity (rH%), CO2 concentration (part-per-million), total volatile organic compounds (TVOC - parts-per-billion) and room illuminance (lux).

The residences have different occupancy levels and number of rooms: studios with a single occupant, apartments with one or two bedrooms with two occupants, and apartments with three bedrooms or houses with more than two occupants. The climate in Boulder, Colorado is temperate, with average annual precipitation of 54 cm and temperatures ranging from -6°C to 31°C. Additionally, the data was collected using 5 hubs (RS1 - RS5) that contain environmental sensors, a microphone, and a camera, placed in different parts of the rooms (kitchen, hallway, living room, etc.) to collect data from multiple areas of the houses.

The Figure 1 presents the variations in the values from each environmental sensor and the variations in the number of occupants from a representative data within the HPDmobile dataset. The measurement values from the lighting sensor and the sensor measuring carbon dioxide concentration have been scaled by a factor of 100 for better visualization.

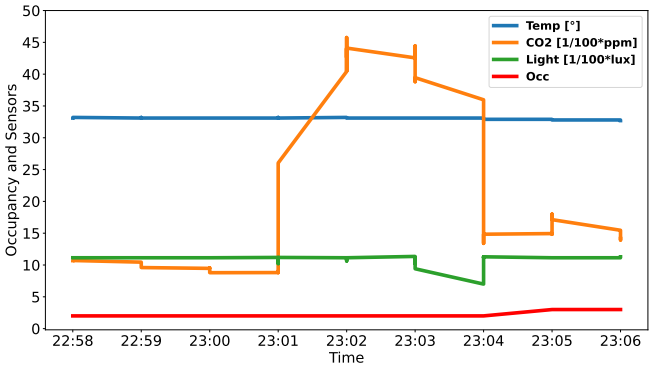

Fig. 1. The variations of environmental sensors and the number of occupants

In the study, two, three or four sensors measurements were used, more exactly: for two sensors - temperature and illuminance, for three sensors - temperature, illuminance and CO2 and for four sensors -temperature, illuminance, CO2 and humidity, as they has the most variations in values.

A. Neural Network Architectures

We use three deep learning, neural network, architecture models to estimate the level occupancy based on the dataset presented above. The three architectures include multiple layers, the main layers being:

- Convolutional Neural Network (CNN);
- Convolutional Neural Network (CNN) and Fully Connected (FC);
- Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM).

The convolutional models process the input data over various layers with specific operations. The last two models are created by paralleling two architectures, each of them including layers like maximum pooling (MaxPool2D), batch normalization or DropOut layer. Two of network architectures which are used in parallelization is illustrated in Figure 2 and Figure 3.

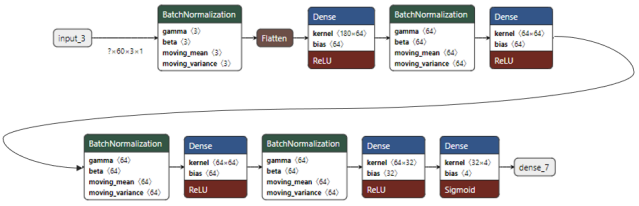

Fig. 2. FC Architecture for Occupancy Classification

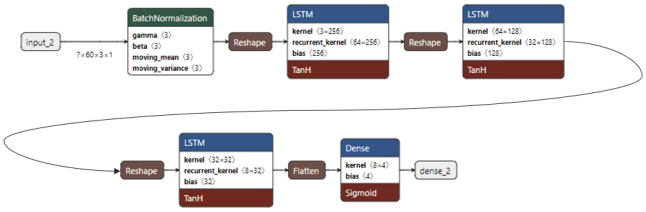

Fig. 3. LSTM Architecture for Occupancy Classification

B. Decision tree-based Models

Another learning method used is Random Forest. Random Forest is a classified machine learning algorithm based on Decision Trees, which is a traditional algorithm that supports various feature importance classes. In the Decision Tree algorithm, the data is repeatedly partitioned based on a particular attribute. All attributes build decision rules that are learned by the Decision Tree model and based on these, it predicts the desired variable. The model is created from decisions nodes and leaves, where both the outcome of the decision and the parameters that lead to the decision can be visualized. Entropy is the splitting criterion in this algorithm, which is computed for each decision node using the formula (1):

$$E(S) = \sum_{i=1}^{Nc} -p(i) \cdot \log_2 p(i) \tag{1}$$

where *Nc* is the number of classes and *p(i)* is the probability of selecting a data point belonging to class i and *S* is the current state. [13]

The entropy equation with four classification classes is computed using the formula (2):

$$E(S) = -p_e \cdot \log_2 p_e - p_l \cdot \log_2 p_l - p_m \cdot \log_2 p_m - p_h \cdot \log_2 p_h \tag{2}$$

For implementation the model were used 30 estimators and *gini* criterion to measure the quality of a split. This

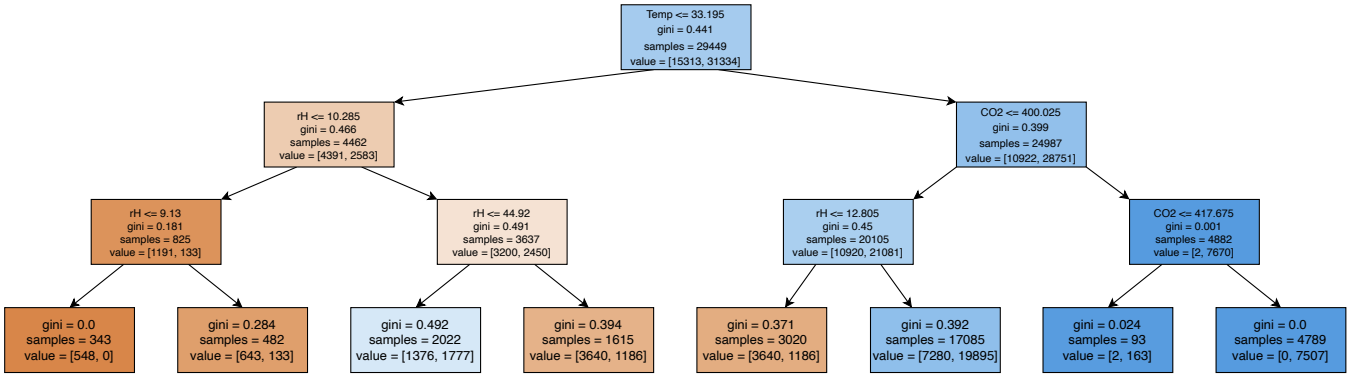

Fig. 4. Decision Tree Sample of Random Forest H5

criterion produces a visualization of featurs relevance called *Gini importance* and is a by-product of the Random Forest Classifier's training process. The main idea is: for the each node t in the decision tree T, an optimal split is determined by Gini entropy [13]. For compute the Gini Index is used the formula (3):

$$G = \sum_{i=1}^{Nc} p_i \cdot (1 - p_i) \tag{3}$$

IV. RESULTS

The input datasets are used with a step of reading from files of 2, i.e. 20 seconds. Pre-processing is carried out to convert the occupancy values into discrete occupancy classes suitable for the defined problem. Python programming language was used for implementation the four architectures: for the three neural network models the *Tensorflow* open-souce library has been used while for the Random Forest method the *sklearn* library was used.

For illustration the decision which Random Forest method achieves, Figure 4 presents one representative decision tree with a depth of tree. The graphic shows the splits achieved by the algorithm from the root node with the full dataset sample towards the leaf nodes and the classification result. Each box identifies the split variable, the associated gini coefficient (see Section III), number of samples and the values. The final random forest output is a weighted average of such individual decision trees with randomised parametrisation of the algorithm.

Both for the three deep learning neural network models and for the Random Forest learning method, the input dataset were randomly split into training (60%), validation (20%) and testing (20%) subsets with a fixed random seed parameter to control the train-validation-test. This assures control of the experiments runs and replicability of the results.

The ultimate goal of the results is to compare the performances of the three neural network models with the performance of the Random Forest method. The input datasets were generated from the HPDmobile collection presented in the section above and are generally based on 7 or 8 days, with 2 or 3 hubs and measurements from 2, 3, or 4 environmental sensors. The aggregated accuracy results are illustrated in Figure 5.

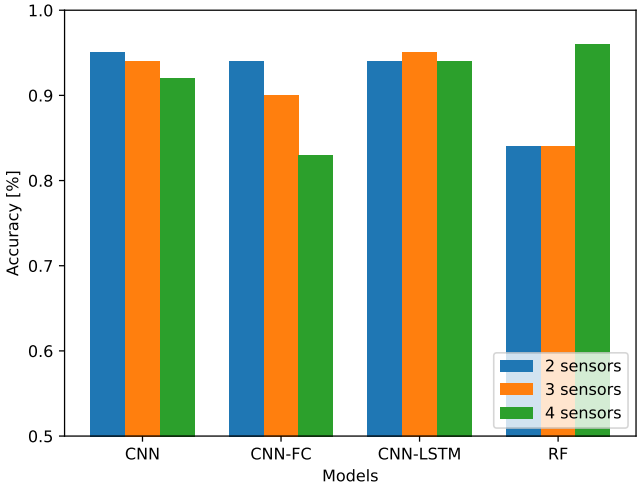

Fig. 5. Accuracy of learning models with 2, 3 and 4 sensor inputs

Table I and II presents the results regarding Accuracy (A) and Loss Function ($\mathcal{L}$) for Neural Networks models (CNN, CNN-FC and CNN-LSTM) and Random Forest model (RF) using datasets with two sensors (2sens), three sensors (3sens) and four sensors (4sens).

For a better visualization of the performance of each learning model, Table III presents the values of precision (P), recall (R) and F1-score (F1) in comparison with accuracy (A) and loss function ($\mathcal{L}$), in the case of H1_RS123 dataset and three sensors (3sens) as input. Even if these performance metrics were created for binary classification, they can be updated for multi-class level as well, computed for each class. The method that we used is one-versus-all which computes the metrics for each of the four classes E, L, M, H, against the sum of the remaining three. The reported values from Table III thus reflect the average for each metric per class.

Accuracy is computed by the formula (4) as the number of the total correct classifications divided to the total number of classifications (*True* + *False*). In opposition, the loss function

TABLE I

ACCURACY AND LOSS FUNCTION FOR NEURAL NETWORK MODELS WITH TWO SENSORS AS INPUT

Dataset	CNN						CNN-FC						CNN-LSTM					
	2sens		3sens		4sens		2sens		3sens		4sens		2sens		3sens		4sens	
	A	$\mathcal{L}$	**A**	$\mathcal{L}$	**A**	$\mathcal{L}$	**A**	$\mathcal{L}$	**A**	$\mathcal{L}$	**A**	$\mathcal{L}$	**A**	$\mathcal{L}$	**A**	$\mathcal{L}$	**A**	$\mathcal{L}$
H1_RS123	0.95	0.19	0.94	0.26	0.92	0.30	0.94	0.26	0.90	0.50	0.83	1.08	0.94	0.18	0.95	0.19	0.95	0.19
H1_RS24	0.87	0.32	0.87	0.31	0.78	0.63	0.89	0.26	0.86	0.3	0.80	0.89	0.88	0.30	0.88	0.30	0.56	0.86
H2_RS125	0.88	0.27	0.86	0.30	0.82	0.42	0.86	0.31	0.84	0.36	0.78	0.86	0.88	0.25	0.87	0.27	0.82	0.47
H2_RS14	0.92	0.14	0.87	0.27	0.82	0.42	0.91	0.19	0.85	0.29	0.76	0.91	0.92	0.18	0.90	0.22	0.80	0.54
H4_RS123	0.78	0.48	0.78	0.51	0.70	0.70	0.81	0.43	0.83	0.39	0.69	0.83	0.79	0.48	0.76	0.51	0.70	0.74
H5_RS345	0.98	0.05	0.98	0.05	0.98	0.03	0.95	0.06	0.96	0.10	0.98	0.03	0.98	0.05	0.98	0.05	0.99	0.02
H6_RS234	0.89	0.25	0.88	0.25	0.79	0.57	0.88	0.26	0.89	0.24	0.76	0.75	0.89	0.25	0.88	0.24	0.77	0.60

TABLE II

ACCURACY AND LOSS FUNCTION FOR RANDOM FOREST MODEL WITH TWO SENSORS AS INPUT

Dataset	RF		
	2sens	3sens	4sens
	A	**A**	**A**
H1_RS123	0.84	0.84	0.96
H1_RS24	0.67	0.68	0.86
H2_RS125	0.85	0.87	0.99
H2_RS14	0.86	0.88	0.99
H4_RS123	0.7	0.76	0.95
H5_RS345	0.93	0.94	0.98
H6_RS234	0.87	0.88	0.9

the categorical crossentropy is used to visualize how much the learning model fails, the formula being used (5).

Precision is defined by the formula (6) as a ratio between the number of correct classified values (*True*) and the total number of values in the data (*True* + *False*). The recall value is computed by the formula (7) as a division of the number of correct classified values (*True*) with the number of real values from the data. F1-score is a combined measure of precision and recall for each class, the formula being used (8). [14]

$$Accuracy(A) = \frac{Correct\ classifications}{Total\ number\ of\ values} * 100\% \quad (4)$$

$$Loss(\mathcal{L}) = -\sum_{i=1}^{Nc} y_i \cdot log(\hat{y}_i) \quad (5)$$

$$Precision(P) = \frac{TP_{class}}{TP_{class} + FP_{class}} \quad (6)$$

$$Recall(R) = \frac{TP_{class}}{TP_{class} + FN_{class}} \quad (7)$$

$$F1 - score(F1) = \frac{2 \cdot (Precision_{class}) \cdot (Recall_{class})}{Precision_{class} + Recall_{class}} \quad (8)$$

where y_i is the truth label and $\hat{y}_i$ is probability for i-th class and TP True Positive values, FN False Negative values, FP False Positive values and TN True Negative values.

TABLE III

PERFORMANCE METRICS FOR RANDOM FOREST MODEL AND NEURAL NETWORK MODELS WITH THREE SENSORS AS INPUT

Model	Perf	Classes			
		E	**L**	**M**	**H**
CNN	**P**	0.94	0.95	0.97	0.87
	R	0.98	0.99	0.88	0.95
	F1	0.96	0.97	0.92	0.91
	A	0.94			
	$\mathcal{L}$	0.26			
CNN-FC	**P**	0.89	0.92	0.93	0.86
	R	0.95	0.98	0.82	0.85
	F1	0.92	0.95	0.87	0.85
	A	0.90			
	$\mathcal{L}$	0.50			
CNN-LSTM	**P**	0.94	0.95	0.97	0.87
	R	0.98	0.99	0.87	0.96
	F1	0.96	0.97	0.92	0.91
	A	0.95			
	$\mathcal{L}$	0.19			
RF	**P**	0.75	0.63	0.53	0.89
	R	0.71	0.55	0.48	0.91
	F1	0.73	0.59	0.50	0.90
	A	0.84			

The CNN-LSTM model is the most representative model in terms of performances, the confusion matrix being presented graphically in the Figure 6. To make a comparison, confusion matrix of Random Forest is also represented in the Figure 7. As observed in the confusion matrix, although the overall accuracy values are good for the Random Forest model, it hides losses behind it. To this extent the F1-score, computed as the harmonic mean of precision and recall is a better indicator for the selectivity and the robustness of the model in a production operating environment. Lower variations are observed between the three CNN models that have been tested which indicates a better stability of such architectures for our particular problem.

The model size is very important if there are memory and power limitations, the neural network models having sizes up to 0.25 MB, while one of Random Forecast tree can reach up to 1.57 MB and the all model being computed from 30 trees. This aspect can be considered for embedded deployment on microcontroller-class devices with limited computing and communication resources for online operation. A larger model will result in slower inference which might limit its usefulness at higher sampling rates of the control system.

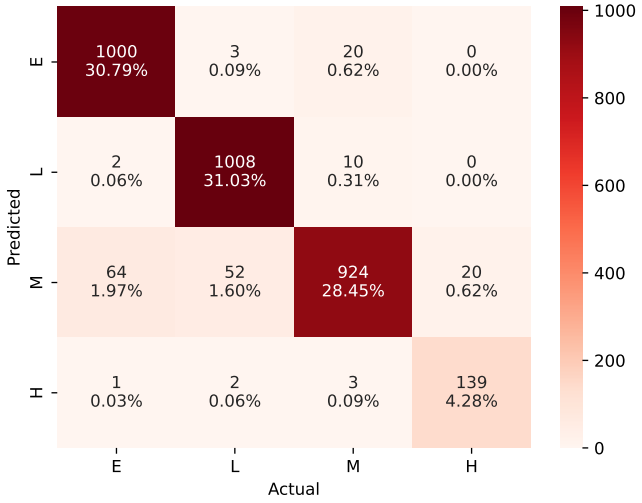

Fig. 6. Confusion Matrix for CNN-LSTM model with three sensors as input

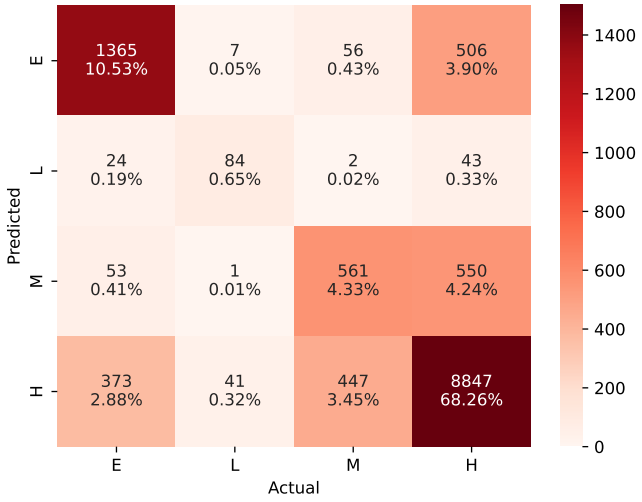

Fig. 7. Confusion Matrix for Random Forest model with three sensors as input

New automated machine learning approaches can also improve the model selection and parametrisation process, however domain expertise, in adapting a certain model type to the particular engineering problem remains. Explainability of the models [15] also has to be accounted for in driving acceptance of the proposed solution to the end users, the facilities managers in our case.

V. CONCLUSIONS

The paper presented an evaluation and comparison between deep learning and machine learning models on open building occupancy datasets. The results can be used in a predictive building energy control framework, for including accurate estimates of actual occupancy in the control decision. Various subsets of available data have been used to highlight the effect of considering various environmental parameters on the model performance. A complete evaluation of classification performance requires multiple metrics to assure robustness, while considering just the accuracy of the models, might limit the effectiveness of the models in practice.

Future work is focused on leveraging such trained models in practice by using the generated predictions in a receding horizon control application for building energy management.

REFERENCES

[1] V. L. Erickson, M. Á. Carreira-Perpiñán, and A. E. Cerpa, "Occupancy modeling and prediction for building energy management," *ACM Transactions on Sensor Networks (TOSN)*, vol. 10, no. 3, pp. 1–28, 2014.

[2] M. Quintana, Z. Nagy, F. Tartarini, S. Schiavon, and C. Miller, "Comfortlearn: Enabling agent-based occupant-centric building controls," in *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, ser. BuildSys '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 475–478. [Online]. Available: https://doi.org/10.1145/3563357.3566167

[3] B. Dong, Y. Liu, W. Mu, Z. Jiang, P. Pandey, T. Hong, B. Olesen, T. Lawrence, Z. O'Neil, C. Andrews, E. Azar, K. Bandurski, R. Bardhan, M. Bavaresco, C. Berger, J. Burry, S. Carlucci, K. Chvatal, M. De Simone, S. Erba, N. Gao, L. T. Graham, C. Grassi, R. Jain, S. Kumar, M. Kjærgaard, S. Korsavi, J. Langevin, Z. Li, A. Lipczynska, A. Mahdavi, J. Malik, M. Marschall, Z. Nagy, L. Neves, W. O'Brien, S. Pan, J. Y. Park, I. Pigliautile, C. Piselli, A. L. Pisello, H. N. Rafsanjani, R. F. Rupp, F. Salim, S. Schiavon, J. Schwee, A. Sonta, M. Touchie, A. Wagner, S. Walsh, Z. Wang, D. M. Webber, D. Yan, P. Zangheri, J. Zhang, X. Zhou, and X. Zhou, "A global building occupant behavior database," *Scientific Data*, vol. 9, no. 1, p. 369, 2022. [Online]. Available: https://doi.org/10.1038/s41597-022-01475-3

[4] M. Jacoby, S. Y. Tan, G. Henze, and S. Sarkar, "A high-fidelity residential building occupancy detection dataset," *Scientific Data*, vol. 8, no. 1, p. 280, 2021. [Online]. Available: https://doi.org/10.1038/s41597-021-01055-x

[5] M. S. Aliero, M. F. Pasha, D. T. Smith, I. Ghani, M. Asif, S. R. Jeong, and M. Samuel, "Non-intrusive room occupancy prediction performance analysis using different machine learning techniques," *Energies*, vol. 15, no. 23, p. 9231, Dec 2022. [Online]. Available: http://dx.doi.org/10.3390/en15239231

[6] Y. Zhao, W. Zeiler, G. Boxem, and T. Labeodan, "Virtual occupancy sensors for real-time occupancy information in buildings," *Building and Environment*, vol. 93, pp. 9–20, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360132315300330

[7] J. Liu, H. Mu, A. Vakil, R. Ewing, X. Shen, E. Blasch, and J. Li, "Human occupancy detection via passive cognitive radio," *Sensors*, vol. 20, no. 15, p. 4248, Jul 2020. [Online]. Available: http://dx.doi.org/10.3390/s20154248

[8] A. Mirugwe, "Accurate occupancy detection of an office room from light, temperature, humidity and co2 measurements using statistical learning models," *Temperature, Humidity and CO2 Measurements Using Statistical Learning Models (September 4, 2020)*, 2020.

[9] C. Chiţu, G. Stamatescu, I. Stamatescu, and V. Sgârciu, "Wireless system for occupancy modelling and prediction in smart buildings," in *2017 25th Mediterranean Conference on Control and Automation (MED)*, 2017, pp. 1094–1099.

[10] G. Stamatescu and C. Chitu, "Privacy-preserving sensing and two-stage building occupancy prediction using random forest learning," *Journal of Sensors*, vol. 2021, 2021.

[11] G. Cretu, I. Stamatescu, and G. Stamatescu, "Building occupancy classification from indirect sensing with heterogeneous datasets," in *2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, vol. 1, 2021, pp. 475–479.

[12] G. Stamatescu, I. Stamatescu, N. Arghira, and I. Fagarasan, "Data-driven modelling of smart building ventilation subsystem," *Journal of Sensors*, vol. 2019, 2019.

[13] R. Abedin and S. Waheed, "Performance analysis of machine learning models for intrusion detection system using gini impurity-based weighted random forest (giwrf) feature selection technique," *Cybersecurity*, vol. 5, 12 2022.

[14] V. Nguyen Thanh Le, B. Apopei, and K. Alameh, "Effective plant discrimination based on the combination of local binary pattern operators and multiclass support vector machine methods," *Information Processing in Agriculture*, vol. 6, no. 1, pp. 116–131, 2019.

[15] T. Tsoka, X. Ye, Y. Chen, D. Gong, and X. Xia, "Explainable artificial intelligence for building energy performance certificate labelling classification," *Journal of Cleaner Production*, vol. 355, p. 131626, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959652622012422