Automatic Classification of Sensors in Buildings: Learning from Time Series Data

Mashud Rana[1](✉), Ashfaqur Rahman[1], Mahathir Almashor[2], John McCulloch[1], and Subbu Sethuvenkatraman[2]

[1] Data61, CSIRO, Eveleigh, Australia
{mdmashud.rana,ashfaqur.rahman,john.mcculloch}@data61.csiro.au
[2] Energy, CSIRO, Eveleigh, Australia
{mahathir.almashor,subbu.sethuvenkatraman}@csiro.au

Abstract. Smart buildings are generally equipped with thousands of heterogeneous sensors and control devices that impact the operation of their electrical systems. Analytical tools that aim to optimise the energy efficiency within such complex systems requires prior mapping or (classification) of diverse set of sensors according to a standard. Prior research primarily focuses on exploiting the similarities in sensor names (text metadata) to categorise them into identical classes (or groups). However, the sensors within and across buildings often follow distinct naming conventions by different vendors. In addition the definition of the classes or groups also varies significantly amongst researchers. This limits the usability and portability of prior techniques when applied across buildings. There are standard ontologies (Brick, Haystack etc.) that provide a set of standardized classes for the sensors in the buildings. The work herein follows a new avenue to address this challenging classification problem by (i) utilizing only time-series data of sensors and not text metadata, (ii) developing a simple, effective and hitherto unexplored Machine Learning (ML) model to classify the sensors into a set of standard Brick classes, and (iii) evaluating the model on a large proprietary dataset comprising of 129 buildings. Experimental results demonstrate promising performance of the presented data driven model, with average classification accuracy in terms of weighted F-score at 0.78 (±0.14), and statistically significant improvements over prior methods.

Keywords: Sensor Metadata · Building Automation · Time Series

1 Introduction

Building Automation Systems (BAS) integrate various applications that monitor, analyse, and optimise the energy usage of modern buildings. Their increasing adoption reduces building operational costs, overall emissions, and enable us to achieve urgent sustainability goals [7]. Successful integration of diverse energy analytics with BAS requires access to both the semantic information of a large

T. Liu et al. (Eds.): AI 2023, LNAI 14471, pp. 367–378, 2024.
https://doi.org/10.1007/978-981-99-8388-9_30

number of entities (e.g., sensors, control points), and their associated operational data. The semantic information (aka *metadata*) is required to identify the equipment, locations, physical phenomenon being sensed, and the relationships between those entities. However, such information is often unavailable, unstructured or inconsistent [15]. Reasons for this include: changes in the physical configuration of the buildings over time, and heterogeneous entities that are installed, managed, and named by different vendors. As a result, the usability and portability of energy analytics across buildings are severely limited.

Consequently, several standard schemata like Brick [2] and Haystack [1] have been developed. These schemata describe the heterogeneous sensors and control devices in buildings and the complex relationships among them in a structured and consistent format using a predefined set of *classes* (or *tags*). This allows the use of machine readable representations of the different subsystems of buildings for diverse analytical applications. However, the mapping of entities in buildings by following a standard schema is a time and labour intensive, requiring significant effort from highly specialised domain experts and yet ultimately, is an exercise that is susceptible to errors. Our goal here is to investigate the automated and data-driven mapping of building sensors based on their time series data. In this paper, we use *sensors* and *entities* interchangeably.

Existing approaches predominantly address this classification problem from a purely text processing perspective, using information retrieved from entity names as inputs to the classifiers. The main reason for this is that important properties of the entities are embedded in their names in most cases. The embedded information can help to infer their types, locations, and relationships [10]. However, the success of these approaches are heavily dependent on intrinsic similarities of entity names in both source and target buildings. Any variations of entity naming conventions between source and target buildings negatively affect the portability and usability of the models across buildings. In such case, it is required to integrate domain expertise (e.g., [3]) or use knowledge from target buildings (e.g., [8]) which is not feasible. In contrast, the Time-Series (TS) data associated with similar types of sensors is expected to be consistent within and across buildings. Such data contains patterns that can be utilised as signatures for accurately classifying different types of sensors, regardless of their deployed buildings. Therefore, we've focused on the classification of sensor types based on TS data. Our contributions can be summarized as follows:

- We present a TS-data-driven approach for automatically classifying the sensors in buildings by utilizing XGBoost [4]. A set of statistical features representing the patterns in TS data is explored for classification. Since the approach only requires TS data, it provides better portability and usability across buildings compared to the approaches utilising entity names.
- In contrast to the existing approaches that use user-defined classes or Haystack tags, our approach aims to classify sensors according to the popular Brick schema and at a more granular level (as opposed to shallower levels in current literature) in the class hierarchy to facilitate more widespread applications.
- We evaluate the approach against 129 buildings contained within a proprietary dataset from Australian Data Cleaning House (DCH). The operational

patterns across its diverse building range varies greatly, leading to sensor data with different characteristics and statistical distributions. Using this dataset imposes a stricter challenge, as opposed to datasets that display more homogeneous properties.

- We provide a systematic comparison of XGboost against: (i) other ML classifiers such as Random Forest (RF), Neural Networks (NNs), and Support Vector Machines (SVM) used in prior data-driven approaches; and (ii) the Building Adapter model [11] which utilises both TS and text name space.

2 Related Work

Prior approaches for classification of sensors in buildings can be broadly categorised based on the type of data used [7]: (i) **name space**; and (ii) **time series**. Koh et al. [15] reviewed and implemented several state-of-the-art approaches from both groups. The first group of studies focus on utilising encoded information in text metadata (entity names, vendor specified descriptions, specification of target buildings) as inputs for classification. Balaji et al. [3] proposed "Zodiac", a semi-automatic model that grouped entities by processing their names thru a bag-of-words method and utilising the resultant count values. The entities in each cluster were classified by repeatedly training a Random Forest (RF) model and incorporating feedback from domain experts. He and Wang [8] utilized information extraction principles to merge differing text corpuses gathered from buildings. Text from source buildings were combined with additional knowledge (known as "specification files") that were synthesised from target buildings, which were then fed into a Bi-LSTM model. Scrabble [14] used the combination of Conditional Random Fields and Neural Networks (NNs) on text metadata to apply Brick classes to the sensors. Other prominent studies that adopted similar approaches include [12,18].

In contrast to the abundance of studies utilising text metadata, there are very few studies that rely on TS data to train ML classifiers. Gao et al. [7] utilised several statistical features (e.g., mean, mode, quantiles and deciles) from TS as input to train a set of ML models that including RF, k-Nearest Neighbour (kNN), and SVM. They trained these models for classifying both composite tags and individual tags from Haystack ontology. Hong et al. [9] studied clustering of TS data based on a similarity metric (*cross-predictability*) that was applied to group four types of sensors. The labels used in their study were defined manually instead of following any standard ontology. TS data was also used in Koc et al. [13] but only for inferring spatial relationships between sensors.

There exists studies that incorporate both TS and name space data for classification. The Building Adapter (BA) model in [11] stands as a prominent example. A group of classifiers (SVM, RF, and Logistic Regression (LR)) were trained using 44 TS-based features as inputs. It then transferred the knowledge from source to target building based on clustering of entities using text data from target buildings. Mishra et al. [17] also adopted a similar approach based on RF and SVM classifiers trained on TS data, and clustering on text data.

3 Datasets and Problem Statement

3.1 Datasets

We consider a dataset from DCH, which contains energy and operational data for more than 150 buildings from 30 sites across Australia. After excluding a small subset due to data quality issues, 129 buildings encompassing corporate offices, libraries and research labs remained. Some buildings recorded comprehensive sensor data, whilst others were limited to specific subsystems such as electrical systems. These buildings were modelled manually using the Brick schema by domain experts. The entities in different buildings are named using different naming conventions and by different vendors. Table 1 presents a sample of metadata in our dataset (with anonymised site and building names).

Table 1. Sample metadata for different entities.

Bld id	Entity name	Brick class
Bld_2(Site_5)	dsapi-tacit-silent-start-Site_5_Bld_2.Bld2LandP.VoltageBN	Voltage Sensor
Bld_1(Site_2)	dsapi-thick-gusty-rest-Building_1-Site_2.E5_07_OfficeL2Lighting.CurrentPhaseA	Current Sensor
Bld_1(Site_7)	dsapi-synonymous-tender-campaign-Site_7.B_1...EMeter_3_DB_LG_P2_kVAr	Electrical Power Sensor

Our classification approach relies on TS data attached to the entities in each building, which were typically recorded over several years. Nonetheless, we only considered data from 1^{st} Jan 2022 onwards due to quality issues and to avoid pandemic related anomalies. Moreover, the TS data for buildings in DCH were recorded at different resolutions varying from 5 to 45 min. When modeling the data, we extracted several features representing their statistical properties and patterns, and this is elaborated upon in Sect. 4.1.

3.2 Problem Statement

Given the following:

1. the TS data $TS_B^N = \{ts_1, ts_2, ..., ts_N\}$ of a set of N entities $E_B = \{e_1, e_2, ..., e_N\}$ in a building B where $ts_i = [o_i^1, o_i^2, ..., o_i^L]$ is vector of L time ordered numerical observations of a phenomenon sensed by an entity e_i.
2. the class label y_i of each entity $e_{i \in \{1\ to\ N\}}$. $Y_B = \{y_i, y_2, ..., y_N\}$ is the set of class labels for all entities in the same building.

The Goal is to develop a model, M, to classify the class labels for the entities in the set of E using information from TS_B^N. In other words, the model M intends to learn the mapping function $F(X) \rightarrow Y_B$ where X is the input feature set computed from TS_B^N.

In this study, the entity represents the set of sensors measuring different phenomena (e.g., current, voltage, energy usages, etc.) of electrical systems as well as the outside air temperatures of buildings. The class labels of the entities or sensors are assigned from Brick version 1.2, and we are specifically focused on sensors belonging to five main classes {*Electrical Power Sensor, Voltage Sensor, Current Sensor, Energy Sensor, Outside Air Temperature Sensor*} and their associated sub-classes in the Brick ontology.

4 Proposed Data-Driven Approach

Figure 1 presents a schematic diagram of the proposed approach for sensor types classification. It consists of two main steps: i) *model development*, and ii) *testing*.

4.1 ML Model Development

Time Series Feature Extraction. Feature selection is a crucial step for ML model development. It is the process of identifying a set of informative inputs that can represent the statistical distribution and the patterns in the data [16]. An appropriate feature set helps the ML model to learn both linear and non-linear relationships between inputs and target, and reduce the chances of over-fitting that consequently lead to better performance.

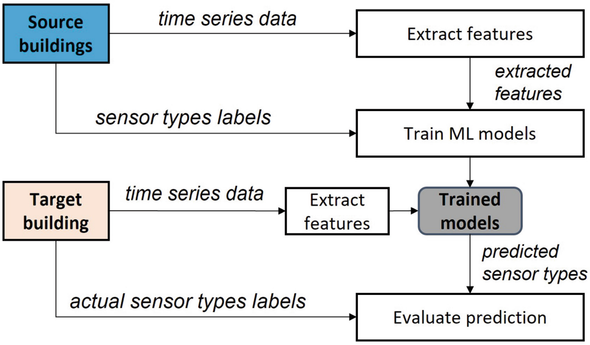

Fig. 1. Schematic diagram of the proposed ML approach for sensor type classification.

The TS data belonging to different groups of sensors shows different statistical distribution over times. Therefore, we aim to extract a set of features from each TS based on a windowing technique as described in [11]. We first segment each TS data into a set of fixed length window where each segment has 50% overlaps with previous one. The length of the window varies depending on the gaps between the consecutive samples in the TS since our dataset has TS with varying sampling rate. For TS with gaps between consecutive samples ≥15 min, we set the window length to be 1 h considering possible hourly pattern in TS. Otherwise, it is set dynamically such that the window contains a minimum number of samples (10 in our case). Following [11], for each window we then computed 11 statistical features representing 4 different statistical properties: i) extreme: *min and max*, ii) variability: *median, root mean square, 1st quartile, 3rd quartile, inter quartile range*, iii) moments: *variance, skewness, and kurtosis*, and iv) shape: *slope*. For each of the 11 statistical features, we then compute 4 summary statistics *min, max, variance, and standard deviation* from the series of values for all window. This will results in 44 final features (*11 features* × *4 summary statistics* for each feature) for each TS.

Model Training. As the classifier, we adapt the Extreme Gradient Boosting (XGBoost) model. XGBoost is a scalable and distributed gradient-boosted decision tree model that can be applied for supervised classification and regression. The main motivations of choosing XGBoost over other classical ML models like NNs include: i) it is more robust to the noisy data, ii) it is highly parallelizable and hence faster to train on large datasets, iii) it requires less computational resources, and iv) it has less parameters and is easier to tune.

XGBoost trains a set of base learners (shallow Decision Trees (DT)) iteratively such that each base learner focuses on the examples that were difficult to classify by the previous one. In other words, in each iteration XGBoost trains a new base learner which aims to minimise the error of the previous learner. The final prediction is computed by combining the predictions from all base learners based on their weights that are determined based on their performance on training data. In contrast to the RF algorithm which also applies a set of DTs each trained on separate subset of training data chosen based on bootstrap sampling and minimizes the variance and over-fitting, the XGBoost focuses on minimizing the bias and under-fitting during the training process. For more details on the theory of XGBoost we refer to [6].

The 44 statistical features extracted from all the TS streams and their respective Brick class labels, from all the source buildings are then fed into XGBoost model. The model learns the mapping between input features and targets (Brick classes) through a training process. We tune the parameters of the XGBoost model by applying a grid search strategy based on 10-folds cross validation of training data from all the source buildings combined. The searching space the of different parameters are presented in Table 2. After finding the best combination of parameters, the model is trained on the entire training data.

4.2 Testing

The evaluation of the trained model using the data from the target buildings begins with feature extraction. For each entity in a target building, we first compute the 44 features explained in Sect. 4.1. These features are the provided to the trained model as inputs and the model predicts the sensors types (Brick classes) for the entities in target buildings. The predicted sensor type labels are then compared with ground-truth Brick class labels to compute the performance of the model.

Table 2. Parameters of the XGBoost model used for grid searching.

Parameters	Description	Values used for grid search
n_estimators	number of base learners (gradient boosted trees) to train	[50, 100, 200]
max_depth	maximum depth each base learner can be expanded	[5, 10, 20, None (fully expanded)]
learning_rate	regularisation to control the contribution of base learners	[1, 0.5, 0.1]
max_leaves	maximum umber of leave nodes each base leaner can have	[0 (no limit), 10, 20, 50]

5 Results and Discussion

5.1 Evaluation Process

The performance of the proposed approach for sensor type classification are evaluated using two metrics: *accuracy* and *F-score*. *Accuracy* represents the percentage of total number of sensors that are correctly classified by the model. For an imbalanced classification problem like ours, *accuracy* alone may not provide sufficient insight of model's performance since it doesn't consider the ratio of observations in different classes. F-score is another assessment metric which evaluates the predictive skill of a model by considering its class-wise performance. For a binary classification task, it can be defined as the harmonic mean of *precision* and *recall* as in (1) where *precision* is the proportion of correctly predicted positive class relative to the all positive predictions and *recall* indicates the fraction of correctly predicted positive class with respect to the total number of observations belonging to actual positive class. For our multi-class classification task, we consider the weighted F-score which is computed as the average of the F-score of each class where the weights are determined by the number of observations in each class.

$$Fscore = 2 \times \frac{(precision \times recall)}{(precision + recall)} \tag{1}$$

Moreover, we evaluate the performance of the model on each building separately. Specifically, out of N sites in our DCH dataset, we consider one site $Site_{i \in \{1\ to\ N\}}$ as the target site and remaining $N-1$ sites $\{Site_{j\,=\,1\ to\ N\ \&\ j \neq i}\}$ as the source sites. The model was trained on the data from all buildings in source sites and tested on each building from the target sites. This process is repeated N times, each time we have a different set of source sites and a target site.

5.2 Model's Performance

Table 3 presents the performance of our model. Although our dataset consists of 129 buildings, for brevity we included the results for 11 buildings each having at least 100 sensors to be classified.

Classification results shows that performance of the proposed approach varies for different site/building pairs. The accuracy is the range of 0.53 to 0.94 and the F-score is between 0.55 to 0.95. The model shows the best performance on the buildings $Bld\_1(Site\_5)$, $Bld\_1(Site\_6)$, and $Bld\_1(Site\_8)$ with both accuracy and F-score over 0.90. On the other hand, the classification accuracy of the sensors belonging to the three buildings $Bld\_1(Site\_7)$, $Bld\_1(Site\_3)$, and $Bld\_1(Site\_9)$ are the lowest (0.53, 0.56, 0.61, respectively). The same trend is also observed if F-score is considered as the assessment metric. The main reason for the comparatively lower accuracy of the model for these three buildings is relatively poor quality of TS data. Although the TS data from most of the buildings was recorded for at least 1 year (Jan–Dec 2022), the sensors in buildings $Bld\_1(Site\_7)$, $Bld\_1(Site\_3)$, and $Bld\_1(Site\_9)$ has only few short bursts of data possibly due to outages. This consequently made the extracted features atypical due to the lack of sufficient samples and representative patterns associated to different sensor types. The accuracy or F-score of the model on the data from other buildings is $\geq$0.70.

Table 3. Performance of the proposed approach.

Bld id	Samples	Accuracy	F-score
$Bld\_1(Site\_1)$	1872	0.76	0.72
$Bld\_2(Site\_1)$	252	0.83	0.83
$Bld\_3(Site\_1)$	179	0.83	0.72
$Bld\_1(Site\_2)$	1003	0.70	0.71
$Bld\_1(Site\_3)$	312	0.56	0.55
$Bld\_1(Site\_4)$	276	0.88	0.87
$Bld\_1(Site\_5)$	218	0.94	0.95
$Bld\_1(Site\_6)$	216	0.92	0.91
$Bld\_2(Site\_6)$	165	0.87	0.88
$Bld\_1(Site\_7)$	131	0.53	0.61
$Bld\_1(Site\_8)$	124	0.91	0.91
$Bld\_1(Site\_9)$	118	0.61	0.59

Moreover, the distribution of the number of sensors in different buildings shows that buildings $Bld\_1(Site\_1)$ and $Bld\_1(Site\_2)$ have the highest number of sensors (1872 and 1003, respectively). The accuracy (or F-score) of the model computed using the sensors from these two buildings is 0.76 (0.72) and 0.70 (0.71), respectively. In addition, the overall accuracy/F-score (averaged over all the buildings) is 0.78 with a standard deviation of 0.14. All these results are obtained using the features extracted from TS data only. This highlights that the TS data contains signature information or patterns that can be utilised in conjunction with ML algorithms to classify the sensors in buildings.

5.3 Comparison

We assess and compare the performance of the proposed approach from different perspectives. Firstly, we assess the advantage and generalisation ability the XGBoost classifier for sensor type classification task by integrating several other ML algorithms as the classifiers in our approach. Secondly, we compare the performance of the proposed approach with a state-of-the-art model (BA [11]).

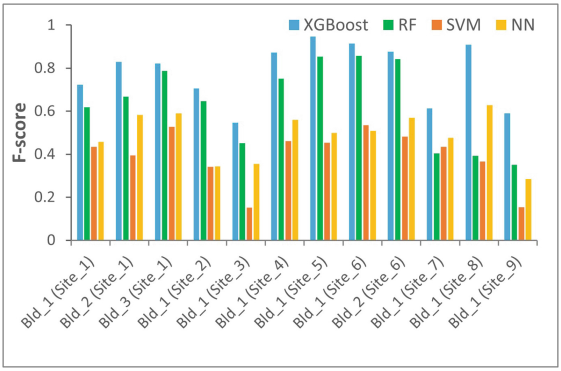

Fig. 2. Comparison of XGBoost with different ML models.

To study the effectiveness of using XGBoost classifier in our proposed approach, we evaluate the performance of proposed approach with three most widely used classical ML algorithms in the literature that include NNs, SVM, and RF. For a fair comparison, the evaluation is conducted using the same feature sets and following the same process applied with XGBoost.

Figure 2 presents the performance of the proposed approach with different ML models used as classifiers, evaluated using F-score. The graph using accuracy is similar and hence not included here. It shows that proposed approach achieves the best performance using the XGBoost as the classifier for all the buildings. The main reason for better performance of XGBoost is its robustness to noisy data as ours TS dataset. The pairwise differences of accuracy/F-score for XGBoost and any classifiers used for comparison is statistically significant (measured by Wilcoxon rank-sum test) at $p \leq 0.05$ for all the buildings except $Bld\_2(Site\_6)$. Among the three classifiers used for comparison, RF which uses ensemble of decision trees provides the highest classification accuracy. This highlights the better generalisation ability of ensemble based models in mapping input-output relationship for sensor types classification.

Moreover, the BA [11] model implemented for comparison utilises both TS and text names space of the entities as inputs. BA first trains a group of ML models (SVM, RF, and LR) using the 44 feature (discussed in Sect. 4.1) as inputs, extracted from the TS data of the sensors in the source buildings. The classification of sensors from a target building is done in two steps. Firstly, it groups the sensors in the target building into different clusters by using the a set of text features formed from the entity names by applying k-mers [5] method. Basically, these text features are the count values of each sub-string of consecutive characters of length k from the entity names. Secondly, the prediction from the ML models are weighted per instance basis based on the similarity between the neighbouring graphs produced from both clustering and prediction of base models for each entity in target buildings. Since our approach used the same TS feature set as in the BA, comparison with BA allows us to investigate if the different classifier (e.g., XGBoost) can provide better classification accuracy and to investigate whether the combination of text and TS feature is beneficial for our dataset.

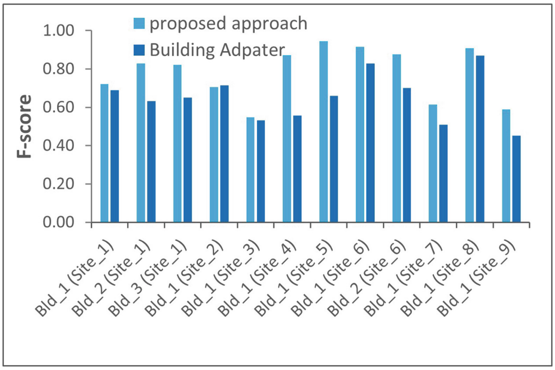

Fig. 3. Comparison with the Building Adapter model [11].

Figure 3 present the performance (F-score) of our proposed approach and the BA model implemented for comparison. The proposed approach using XGBoost as the classifier provides better classification accuracy compared to the BA which utilises SVM, LR and RF to model TS data. The proposed approach provides better classification accuracy for 9 (out of 11) buildings and for remaining 2 buildings ($Bld\_1(Site\_2)$, $Bld\_1(Site\_3)$) both show similar performance. Overall, the average F-score over all the buildings is 0.78 ± 0.14 for the proposed approach vs 0.65 ± 0.12 for BA. The improvement of classification over the BA model is also statistically significant at $p \leq 0.05$. This indicate that utilisation XGBoost instead of other classifiers used in BA to model TS data leads to better classification accuracy for our dataset. Moreover, the better performance of the proposed approach is obtained by using TS data only as opposed to both TS and text name space data in BA.

Although it is expected that name space data can provide useful additional information to the model for classification, in our case the BA model which utilises text data in addition to TS did not show better performance. The relatively lower performance of BA can be explained from two perspectives. Firstly, similar to our approach, BA trains the ML models on TS data from source buildings and utilises the text data from target buildings to provide weights to ML models. However, the among three ML models in BA, RF provided the most accurate prediction with high confidence. Hence the weighting based clustering utilising text data doesn't make big difference in the contributions of the base classifiers to compute final prediction. Secondly, to obtain features from text data, our implementation of k-mers used $k = 3$ following the actual BA model. However, sub-strings of lengths 3 obtained from entity names possibly cannot distinguish the sensors accurately and different values of k set based on empirical evaluation could be a better choice for our DCH dataset.

6 Conclusions

We presented a straight-forward and effective data-driven approach for classifying varying types of electrical and temperature sensors within buildings. The classification was performed in accordance with widely used Brick ontology, and at a more granular hierarchical level than in prior art. This approach was evaluated using a large Australian dataset comprising 129 buildings, with experimentation showing performance at up to 95% accuracy and an average F-score of 0.78 across all buildings. We also found that the classification accuracy for few buildings were not as high as others, with F-scores below 0.6. However, in contrast to the approaches based on text metadata, the main advantages of the proposed approach is its portability and usability. It can be applied across buildings without worrying about the variation of naming conventions of entities in source and target buildings, and it does not require any knowledge from target buildings at all. Building on these promising preliminary results, future work will focus on: (i) augmenting the feature set by using advanced signal processing methods; and (ii) adapting deep learning based classifiers to further improve classification accuracy.

References

1. Project haystack (2001). https://project-haystack.org/
2. Balaji, B., et al.: Brick: towards a unified metadata schema for buildings. In: Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments, pp. 41–50 (2016)
3. Balaji, B., Verma, C., Narayanaswamy, B., Agarwal, Y.: Zodiac: organizing large deployment of sensors to create reusable applications for buildings. In: Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments, pp. 13–22 (2015)
4. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016)
5. Eskin, E., Weston, J., Noble, W., Leslie, C.: Mismatch string kernels for SVM protein classification. In: Advances in Neural Information Processing Systems 15 (2002)

6. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. Stat. **29**, 1189–1232 (2001)
7. Gao, J., Ploennigs, J., Berges, M.: A data-driven meta-data inference framework for building automation systems. In: Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments, pp. 23–32 (2015)
8. He, F., Wang, D.: Cloze: a building metadata model generation system based on information extraction. In: Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation, pp. 109–118 (2022)
9. Hong, D., Gu, Q., Whitehouse, K.: High-dimensional time series clustering via cross-predictability. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, pp. 642–651 (2017). ISSN 2640-3498
10. Hong, D., Ortiz, J., Whitehouse, K., Culler, D.: Towards automatic spatial verification of sensor placement in buildings. In: Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings, pp. 1–8 (2013)
11. Hong, D., Wang, H., Ortiz, J., Whitehouse, K.: The building adapter: towards quickly applying building analytics at scale. In: Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments, pp. 123–132 (2015)
12. Jiao, Y., Li, J., Wu, J., Hong, D., Gupta, R., Shang, J.: SeNsER: learning cross-building sensor metadata tagger. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 950–960 (2020)
13. Koc, M., Akinci, B., Bergés, M.: Comparison of linear correlation and a statistical dependency measure for inferring spatial relation of temperature sensors in buildings. In: Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings, pp. 152–155 (2014)
14. Koh, J., Balaji, B., Sengupta, D., McAuley, J., Gupta, R., Agarwal, Y.: Scrabble: transferrable semi-automated semantic metadata normalization using intermediate representation. In: Proceedings of the 5th Conference on Systems for Built Environments, pp. 11–20 (2018)
15. Koh, J., Hong, D., Gupta, R., Whitehouse, K., Wang, H., Agarwal, Y.: Plaster: an integration, benchmark, and development framework for metadata normalization methods. In: Proceedings of the 5th Conference on Systems for Built Environments, pp. 1–10 (2018)
16. Koprinska, I., Rana, M., Agelidis, V.G.: Correlation and instance based feature selection for electricity load forecasting. Knowl.-Based Syst. **82**, 29–40 (2015)
17. Mishra, S., et al.: Unified architecture for data-driven metadata tagging of building automation systems. Autom. Constr. **120**, 103411 (2020)
18. Waterworth, D., Sethuvenkatraman, S., Sheng, Q.Z.: Advancing smart building readiness: automated metadata extraction using neural language processing methods. Adv. Appl. Energy **3**, 100041 (2022)