



Automated Classification of Datapoint Types in Building Automation Systems Using Time Series

Noah Mertens^(✉)  and Andreas Wilde 

Fraunhofer IIS EAS, Dresden, Germany
noah.mertens@eas.iis.fraunhofer.de
<https://www.eas.iis.fraunhofer.de/>

Abstract. To apply energy efficiency monitoring and optimizing systems for heating, ventilation and air conditioning (HVAC) systems at scale metadata of the system components must be extracted automatically and stored in a machine-readable way. The present work deals with the automated classification of datapoints using only historical time series data and methods from the field of artificial intelligence. The dataset used to conduct the research contains multiple time series from a total of 76 buildings and covers the months of January to November of 2018. Based on these data, relevant features of the time series are defined. Using these features, different classification models as well as the influence of seasonal effects are evaluated. Overall, our approach provides very good results and assigns on average about 92% of all datapoints to the correct class. The datapoints of the worst recognized class are still correctly classified to about 88% (validation data) and 90% (test data), respectively. Possible reasons for incorrect classified datapoints are discussed and a promising solution is proposed. The obtained results show that in practice these methods can reduce the effort of creating and validating digital twins and building information modeling (BIM) for retrofits.

Keywords: Datapoint · HVAC · Retrofit · Classification · Machine learning

1 Introduction

With the Paris Agreement's of a 1.5° target as a guideline efforts for reducing energy consumption must be increased. The building sector, with its 40% share of primary energy demand, is a particular focus. Since 90% of building floor space in non-residential buildings are not efficiently controlled [16], automation of heating, ventilation, and cooling is considered to have a realistic savings potential of 13% of building energy demand by 2035 [5].

Building automation systems (BAS) provide a lot of data about the condition of the building and the installed equipment. Often, errors in the design, programming or operation of the equipment lead to increased energy consumption in

buildings. Using a digital twin with all relevant information on the components of the HVAC-system including their spatial and functional relationships, those errors can be detected and processes can be improved automatically. As of now the manual implementation of a digital twin is not only time consuming, but also prone to human errors [13,17]. One is the lack of norms and standards for assigning datapoint names. The various manufacturers of technical equipment follow different systems for naming the same component, which so far made it impossible to link the plant structure and software in a fully automated way.

In order to expand the use of digital twins and automation in the building sector in a scalable manner, the demand of automated linking of components and software arises. The systems setup with the spatial and functional relationships of the components and their semantic correlations should be recognized fully automatically at best, but at least minimizing human interaction.

This work focuses on datapoint types from ventilation systems. An important intermediate step is the recognition of datapoint classes, i.e. a classification of datapoint into e.g. temperature sensor, pressure sensor or damper position. The problem of automated metadata detection has been addressed in recent years [17]. This classification can be based on two different sources of information. On one hand, the time series of the datapoint can be used, on the other hand, the datapoint names are available. In [1], the ZODIAC system is presented that automatically extracts metadata using datapoint identifiers and features of the associated time series. Fütterer et al. [7] present a method that is, however, only tested on a single building. The sampling rate of the datapoints used in this process is with 1 min much higher than what usually is available in real cases. Bode et al. [2] present an approach for classification using unsupervised learning methods. However, the results obtained are significantly worse compared to methods based on supervised learning. Gao et al. [8] and Shi et al. [15] evaluate several approaches to classify datapoints based on the time series and compare the respective results for multiple buildings. In a more recent study, Chen et al. [3] use only features of time series to detect classes. The approaches presented achieve classification accuracies of 90% and 85%, respectively, but little attention is given to the practical aspects in real applications such as availability of data to train a model or seasonal effects.

The following use case is assumed for our investigations: Historical data is available from an unknown building but the amount of historical data can vary between a few days and multiple years. A classifier assigns a datapoint class and a classification error probability to each datapoint. For this purpose we developed a concept by selecting and combining existing techniques for a suitable classifier and realized its implementation. The influences of seasonal effects are taken into consideration to evaluate the portability of the developed solution towards different climate zones. Finally, the classifier will be tested with the data of two buildings and the result will be evaluated with regard to the practical applicability, especially the necessary amount of operational data.

2 Dataset and Classification Pipeline

Our approach for classifying datapoints is based on the properties and characteristics of the time series only and consists of the following steps:

1. All data of each datapoint is divided into sections of 24 h (daily time series, DTS).
2. Characterizing features for classification are extracted from the DTS.
3. Based on the extracted features, a classification model is trained.
4. Steps 2 and 3 are repeated with different combinations of features and classification algorithms. The best one is chosen (measured by weighted F1-score).
5. For the final classification of a datapoint each DTS is classified individually, then the class of a datapoint is determined according to the majority principle.

The two-step procedure for classifying the datapoint distinguishes the present study from previous approaches and drastically improves results and robustness. The underlying idea is that there are typical daily courses for each datapoint class, but atypical progressions occur on individual days. Using a day-by-day classification we reduce the influence of atypical courses on the overall results.

In the following Sect. 2.1 the dataset is described in more detail. Subsequently, Sect. 2.2 further explains the procedure for the classification of datapoints.

2.1 Description of the Dataset

The dataset used in the paper is from the Mortardata database (Modular Open Reproducible Testbed for Analysis & Research) [6], which was published in 2019. This database collects and unifies a variety of measurements and metadata from over 100 buildings. At the time of the study, the database only includes buildings from the area around San Francisco, California (USA). Possible influences due to the geographic location on the results are considered in Sect. 3. The data is predominantly available with a temporal resolution of 15 min (G. Fierro (UC Berkeley), personal communication, March 11, 2020). For this study all time series were sampled at this rate.

To account for seasonal effects when classifying datapoints, data from almost the entire year 2018 is used. In the building sector, typical patterns are often observed on a daily basis, so the length of each time series was set to 24 h (DTS). Thus in our data each DTS is represented by a vector with 96 elements. Thus, one datapoint provides up to 365 individual time series of measured values in one year. Time series which provide less than 96 measured values are discarded.

Figure 1 shows the classes and the corresponding number of datapoints and DTS. They come from a total of 76 buildings. The dataset is divided into training, validation and test data. The training data is used to train models and the validation data is necessary for feature and hyperparameter selection. Test data is used for a final performance evaluation only. The dataset is split based on buildings (holdout set) to represent a real application. The validation data comes from 12 buildings that contain all classes. The test data consists of 2 buildings, each one containing all classes.

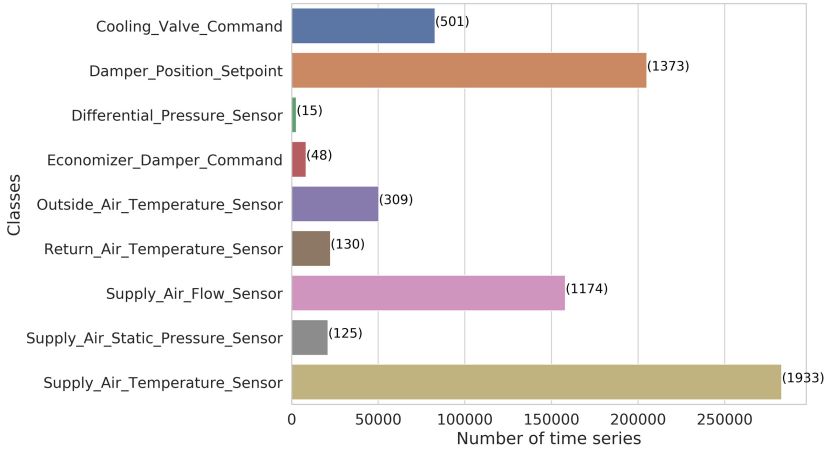


Fig. 1. Number of DTS per class (number of datapoints in brackets)

The dataset is highly imbalanced with respect to the number of DTS per class. To reduce negative effects on the models learning behavior, all classes in the training data are reduced to the size of the second smallest class (*Economizer_Damper_Command*) by undersampling (randomly removing samples) on a quarterly basis. The class sizes of both validation and test data remain unchanged.

2.2 Classification Procedure

To perform a classification features must be calculated from the DTS. In addition to standard features of descriptive statistics such as mean or extreme values, features based on patterns and shapes of the DTS are also taken into consideration. To find the best combination of time series features and classifications algorithms, we proceeded as shown in Fig. 2:

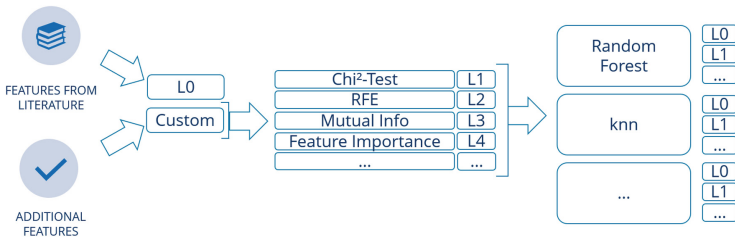


Fig. 2. Process of feature and model selection

An initial selection of features is made based on the literature (“L0”). Subsequently, the list “L0” is extended by multiple features from the library

TSFresh (0.16.0) [4] that we expected to bring high information gain, resulting in the list “Custom” with a total of 122 features. The additional features were chosen based on domain knowledge and are calculated using the Python library *TSFresh* with default settings.

We use different State of the Art approaches to determine feature relevance (Table 1). With each of these feature selection methods, all models from Table 2 are trained and evaluated respectively. For hyperparameter the default values from the Python libraries *Scikit-Learn*, *XGBoost* and *LightGBM* were used. The weighted F1-score is used as a measure of the performance.

Table 1. Feature selection approaches

Approach	#features	Encoding
Features from literature	60	L0
Extended list (L0 + additional features)	122	Custom
Chi2 Feature Importance	60	L1
RFE Default Random Forest	69	L2
Mutual Information	60	L3
Feature Importance Default Random Forest	58	L4
TSFresh Feature Selection	121	L5
Intersection of features of all approaches	23	L6
Features in at least 4 approaches selected	45	L7

Table 2. Used models

Model	Used in literature
Linear Discriminant Analysis (LDA)	[8, 9]
k-Nearest Neighbor (kNN)	[7–9]
Random Forest	[7–12, 14]
Gradient Boost	
ExtraTrees	
LightGBM	
XGBoost	

Comparing the results of all combinations of models and lists of features, the models *XGBoost* (F1 = 0.89) and *ExtraTrees* (F1 = 0.87) with the features of the list “L7” were selected based on the achieved results. *XGBoost* achieved the highest F1-score and is used for the final evaluation of our approach. *Extra Trees* has a significantly shorter training time and only slightly worse results and is therefore suitable for systematic tests concerning seasonal effects as described in the following section. Both models stem from the family of tree-based algorithms and therefore we expect a high transferability of the results.

3 Results

3.1 Influence of Seasonal Effects

Ideally, a classifier is trained with data from an entire year, thereby enabling classification of datapoints of unknown buildings independent of the recording date of the DTS. When considering the influence of seasonal effects, we investigate whether the performance of a model depends on the date of the time series. For this purpose, a model is trained with the training data of the whole year and evaluated for each month separately with validation data. In Fig. 3 the weighted F1 scores for each month are shown, where a red horizontal line represents the mean of all calculated F1 scores. Since neither training nor evaluation data are available for December, no result can be shown for this month. In the lower part of the figure, the number of DTS is plotted over the months.

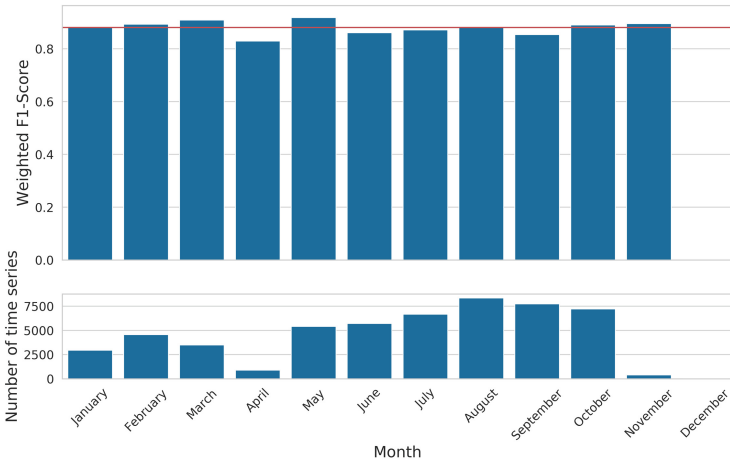


Fig. 3. Model trained with data of whole year, classification performed for each month individually (Extra Trees with “L7” Features)

It turns out that the performance of the classification is mostly independent of the date of the time series. Only in April there is a noticeable deviation from the mean value. One possible reason is the significantly small amount of training data for this month. On the other hand the training dataset contains very little data for November as well, but the performance in this month shows only a insignificant difference from the mean value. Therefore the number of training data does not seem to be the only influencing factor and seasonal effects could in fact play an important role.

The climate in San Francisco, California is subject to large fluctuations (as it can be seen in standard climate charts), especially in the months of spring and autumn. If only a few DTS are available during these months, information

about possible fluctuations of e.g. sensor data could only be partially recorded. As a result, the classifier will receive less information from this period of the year. With an extended dataset of multiple years a more comprehensive understanding of the influence of seasonality could possibly be achieved.

In terms of flexibility for using the classifier for unknown buildings, the annual climate seasonality plays an important role. For the present dataset of buildings from the San Francisco area, the seasonal temperature pattern seem to play minor role(cf. Fig. 3) though. To apply the classifier in central european countries such as Germany, the influence of the seasons on the performance must be verified with an appropriate dataset.

3.2 Classification: From Time Series to Datapoints

This section examines the results of the classifier with test data which were previously completely unknown to the model and thus represents a real use case scenario. The test is conducted with the *XGBoost* model, which achieved the best results during model and feature selection. For this purpose, the model is trained with a combination of training and validation data to maximize the amount of training data. As described in Sect. 2.2, in a first step the time series of all datapoints are classified. Subsequently the datapoint class is determined by majority principle. Figure 4 (left) shows the weighted F1-scores for the classification of DTS and datapoints. The classification of the DTS already achieves a F1-score over 0.925. After the second step of the datapoint classification, the result improve significantly and reach a weighted F1-score of about 0.975.

For practical applications with limited availability of data the question arises how many time series per datapoint are necessary in order to make a reliable statement about the datapoint class. This question can be answered using the binomial distribution. Thus, the probability $P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}$ that exactly k of n DTS are correctly classified is calculated, where p corresponds to the probability that a single time series is correctly classified. p varies for different datapoint classes. For a conservative estimation, the value of p is taken to be the worst datapoint class recognized on test and validation data ($p = 0.72$). For this approach, classification errors are assumed to occur randomly and the individual tests are assumed to be stochastically independent. The plot in Fig. 4 (right) shows that with 13 DTS per datapoint, the overall probability of determining the correct datapoint class is already greater than 95%.

When using the model with the data of unknown buildings, it is not only important to know the predicted class, but also the probability of the prediction being correct. This allows the class assignment to be fully automated for some classes and focusing expert knowledge on classes with low classification confidence. The prediction probability can be directly read from the main diagonal of the classification matrix in Fig. 5. They express the ratio of correctly assigned datapoints of a class to all datapoints of this class. More than half of the classes are recognized correctly in all cases. Only the three classes Damper_Position_Setpoint, Return_Air_Temperature and Supply_Air_Flow Sensor show classification errors at all.

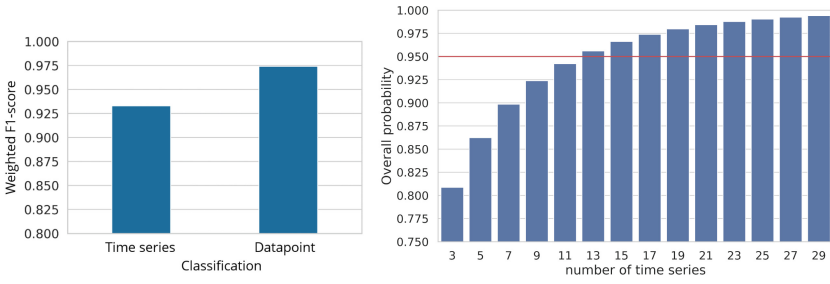


Fig. 4. Left: Comparison classification of individual time series and datapoints (XGBoost with “L7” features, test data), right: Overall probability over the number of time series per datapoint

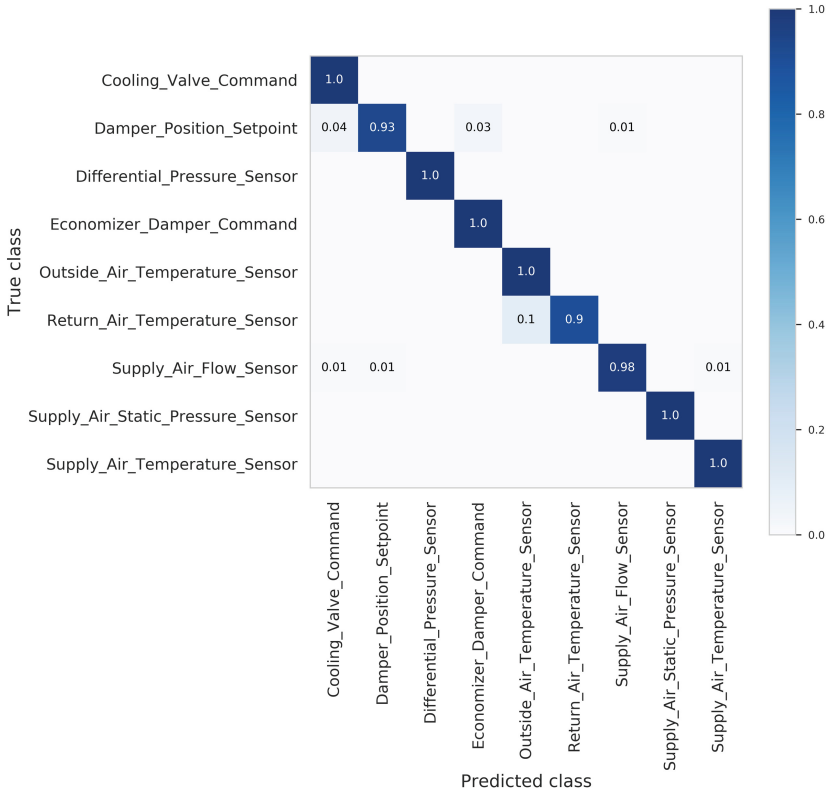


Fig. 5. Normalized classification matrix (XGBoost with “L7” features, Test data)

4 Discussion and Conclusions

For the data used in this research a very good recognition of datapoint types was achieved. The reliability increases with the number of available DTS, whereby on average a classification accuracy of 95% is exceeded if more than 12 DTS are available. In practice, this should lead to a drastic reduction of the expert effort in many cases. With respect to the season from which the DTS originate, the results are reliable for at least the investigated climate zone (San Francisco/USA). The transferability of this finding to other climate zones still has to be examined with an appropriate dataset. It is to be expected that problems can occur in climatic zones with more pronounced annual temperature variations: Heating systems in Germany, for example, do not operate for weeks in the summer, nor does cooling in the winter. In cases where the systems are not in operation, very little information can be extracted from corresponding time series. In these cases, data of multiple seasons must be provided in order to achieve reliable results. As a rough guideline, however, it remains to say that about 2 weeks of operating data make the datapoint of the respective active systems well automatically identifiable (cf. Fig. 4).

It turned out that the classification results for individual classes such as “Damper_Position_Setpoint” (flap position) was comparatively poor. The reason for this could be that this datapoint class was imprecisely defined in the dataset. Dampers can be used at very different locations in a ventilation system and can have very different operating characteristics depending on their corresponding function. Thus, the comparatively poor detection performance indicates a fuzzy taxonomy of datapoint rather than problems with data availability, feature selection or the overall classification approach.

Also some datapoint of the Return_Air_Temperature class are not recognized correctly. More detailed investigations showed that the DTS of the incorrectly classified datapoint have a significantly larger range of values than the training data. One possible reason for this is the different usage profile of a single room in the existing dataset. It is expected that additional buildings with similar usage profiles in the training data may improve the classification results.

To create a useful digital twin of a building, all relevant technical components need to be included. Knowing the type of datapoints can enable (semi)automatic datapoint mapping or support evaluation of manually mapped datapoints to avoid errors (e.g. check if components are mapped in a unusual order). The results obtained in this work thus show that the classification of datapoints based on features of the time series could be a practical way to reduce the expert effort in creating and validation digital twins and BIM for retrofit applications in the building sector.

In this work, however, a number of other aspects such as the automatic detection of functional relationships and the affiliation to different parts of the plant are not considered. For these tasks, other sources of information such as datapoint identifiers or graphical representations of the plant structures will play a major role. However, since these sources of information are often incomplete and

subject to errors, time series can offer valuable information for the plausibility check of information from other sources.

Acknowledgements. Supported by:



based on a resolution of the German Bundestag under the funding code *03ET1567A*.

References

1. Balaji, B., Verma, C., Narayanaswamy, B., Agarwal, Y.: Zodiac: organizing large deployment of sensors to create reusable applications for buildings. <https://doi.org/10.1145/2821650.2821674>
2. Bode, G., Schreiber, T., Baranski, M., Müller, D.: A time series clustering approach for building automation and control systems. <https://doi.org/10.1016/j.apenergy.2019.01.196>
3. Chen, L., Gunay, H.B., Shi, Z., Li, X.: A metadata inference method for building automation systems with limited semantic information. <https://doi.org/10.1109/TASE.2020.2990566>
4. Christ, M., Braun, N., Neuffer, J., Kempa-Liehr, A.W.: Time series Feature extraction on basis of scalable hypothesis tests (tsfresh - a python package). *Neurocomputing* **307**, 72–77 (2018)
5. Debusscher, D., Waide, P.: A timely opportunity to grasp the vast potential of energy savings of building automation and control technologies, p. 21
6. Fierro, G., et al.: Mortar: an open testbed for portable building analytics. <https://doi.org/10.1145/3366375>
7. Fütterer, J., Kochanski, M., Müller, D.: Application of selected supervised learning methods for time series classification in building automation and control systems. *Energy Procedia* **122**, 943–948 (2017)
8. Gao, J., Bergés, M.: A large-scale evaluation of automated metadata inference approaches on sensors from air handling units. <https://doi.org/10.1016/j.aei.2018.04.010>
9. Gao, J., Ploennigs, J., Berges, M.: A data-driven meta-data inference framework for building automation systems. <https://doi.org/10.1145/2821650.2821670>
10. Hong, D., Ortiz, J., Bhattacharya, A., Whitehouse, K.: Sensor-type classification in buildings. <https://doi.org/10.48550/arXiv.1509.00498>
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 - BuildSys 2015* (2015). <https://doi.org/10.1145/2821650.2821657>
12. Koh, J., Balaji, B., Akhlaghi, V., Agarwal, Y., Gupta, R.: Quiver: using control perturbations to increase the observability of sensor data in smart buildings. <https://doi.org/10.48550/arXiv.1601.07260>
13. Koh, J., Balaji, B., Sengupta, D., McAuley, J., Gupta, R., Agarwal, Y.: Scrabble: transferrable semi-automated semantic metadata normalization using intermediate representation. <https://doi.org/10.1145/3276774.3276795>
14. Park, J.Y., Lasternas, B., Aziz, A.: Data-driven framework to find the physical association between AHU and VAV terminal unit - pilot study, p. 9

15. Shi, Z., Newsham, G.R., Chen, L., Gunay, H.B.: Evaluation of clustering and time series features for point type inference in smart building retrofit. <https://doi.org/10.1145/3360322.3360839>
16. Waide, P., Ure, J., Karagianni, N., Smith, G., Bordass, B.: The scope for energy and CO2 savings in the EU through the use of building automation technology
17. Wang, W., Brambley, M.R., Kim, W., Somasundaram, S., Stevens, A.J.: Automated point mapping for building control systems: recent advances and future research needs. *Autom. Constr.* **85**, 107–123 (2018)