Generating Generic Data Sets for Machine Learning Applications in Building Services Using Standardized Time Series Data

F. Stinner[a], Y. Yang[a], T. Schreiber[a], G. Bode[a], M. Baranski[a], and D. Müller[a]

[a] RWTH Aachen University, E.ON Energy Research Center, Institute for Energy Efficient Buildings and Indoor Climate, Germany
E-mail: fstinner@eonerc.rwth-aachen.de

Abstract –

Machine Learning Algorithms (ML) offer a high potential with low manual effort to discover appropriate energy efficiency measures for buildings. Although many building automation systems (BAS) record a high amount of data, technical systems such as boilers provide only a few data points per building. However, machine-learning algorithms require training based on a sufficient number of instances of a technical system in order to enable cross-building use.

In contrast to electrical systems, few data sets of actual operation of thermal systems are publicly available. Since 2012, the monitoring system in our test object has continuously provided threshold-based data with a maximum resolution of 1 minute. We monitor the plants, energy consumption and comfort parameters with 9239 data points in total. In this paper, we show how our published data set from this building is structured. In order to facilitate the use of ML, each data point receives a uniform label according to a previously developed approach.

Since the documentation of ML data sets varies in the building sector, we show an approach to standardize data sets with special datasheets for thermal systems to provide sufficient information for application of ML.

We use the Brick Schema, a unified ontology standard for the description of topology in buildings, which is part of the future ASHRAE Standard 223P. We couple this with an approach we developed for the structured labeling of data points in buildings.

We show how to semi-automatically generate physical models based on an open-source Modelica library from this ontology-based model. We show that the models, enriched with real time series data and data sheets, are in good agreement with the measured data.

Finally, we show with an ML example that our approach based on Brick Schema and Modelica is able to deliver ML compliant data sets.

Keywords –
Standardized data sets; Machine Learning; Simulation; Modelica; Building Energy Systems

1 Introduction

Machine Learning Algorithms (ML) are increasingly finding their way into building energy systems (BES) in order to increase the systems' energy efficiency. One application of ML algorithms in BES is the optimization of the control system. Further applications are the modelling and prediction of behavior of energy systems or energy consumption [1]. They can support the normalization of metadata of data points [2] or fault detection [3]. Since different disciplines use the term "data point" differently, we use the following definition, which we derived from the use in building automation (BA): a data point is an information carrier that continuously provides information about a state. The use cases and methods for the application of ML in energy technology are manifold. [4] and [5] give an overview for the application of supervised learning in energy systems.

Each data point receives a label in the building, the so-called data point identifier. Therefore, there is no common standard or exchange format for data point identifiers for all buildings. This leads to the fact that applications in the building are not or only with difficulty transferable to other buildings. Thus for ML, most time series from BES are unstructured.

However, a large amount of structured data is required for the development and application of most ML use cases in BES. These data should be retrieved from as many different systems as possible, so that a comparison with different building types is possible [6]. Some technical system types exist only in small numbers within a building, e.g. often there is only one heat pump in a building. In order to develop special classification applications for these technical systems, a clearly sufficient number of structured time series is necessary.

[7] shows an approach to structure or classify data points of BA in 22 classes. In practice, the data points in BA have a higher differentiation.

Published data sets on buildings or building operation originate mainly from the electrical monitoring [8][9] or the comfort [10]. If the operation in thermal systems is covered, then mainly in 15-minute samples [11][12] or for special applications, such as air handling units [6] or subways [13]. The 15-minute rate is not sufficient to reflect the technical monitoring in control as described in VDI 6041 [14].

In existing simulative approaches for virtual data sets, models of buildings were used to generate a virtual data set for the comparison of building consumption [15] or synthetic data for Non-intrusive Load Monitoring (NILM) were produced [16].

For the testing and benchmarking of ML algorithms, it has become standard practice to use test data sets. According to [17], developers and users of ML approaches need a precise description of the data sets in order to interpret them correctly. The existing approaches for the description of building datasets provide information for a limited number of use cases. In particular, the data sets do not describe the difficulties of data collection or data application. [17] developed a comprehensive questionnaire for describing ML data sets. However, the use of BES data in ML algorithms requires more information that is special. Therefore, in our questionnaire, we added suitable questions and removed unsuitable questions for BES in [17].

Physical simulation models can supplement existing data or multiply existing time series with variants of technical systems for the application of ML. For example, simulations support different operating conditions of a wide range of technical systems. However, the creation of such simulation models is a very time-consuming process, and the reuse of existing data point identifiers can significantly reduce this effort.

Here, the use of a unique machine-readable data point identifier is necessary. There are currently countless standards and norms that promise universal descriptiveness of building data. However, no standard can support all requirements for the simple creation of simulation models of BES. Building Information Modeling (BIM) with IFC4 could help here. However, IFC4 describes only few tags in the description of metadata for data points [18] and, in a survey conducted by the German Architects' Association [19], only 12% of the respondents used BIM in projects at all. The approach for naming and which we described in [20], together with the Brick Schema [21] which will be included in the future ASHRAE Standard 223P [22], fulfils the requirements for simulation and ML.

However, to the best of our knowledge, there is no combined approach for real-life and simulated time series data for the training of supervised learning algorithms for classification of rare time series in BES.

In this paper, we show how to standardize real time series data sets in building energy system (BES) with a unique machine-readable data point identifier for usage in Machine Learning (ML). We propose a questionnaire for BES that a data set provider should fill out to support the use for ML practitioners. We show that our developed toolchain can extract information from data point identifiers and use it for the semi-automated creation of a simulation model. We also present how our toolchain can create a simulated data set of rare time series in the BES. In a use case, we demonstrate how ML applications can use our simulated data set for the classification of rare time series in BES. We point out whether the developed approach is suitable for application in ML.

2 Standardized data sets for timeseries data

2.1 Questionnaire for Time Series Data Sets in Building Energy Systems

In order to standardize the description of data sets, we defined a set of questions that an owner of data sets, who intends to publish the sets, should answer. We used an existing approach [17] and added questions suitable for BES. We want to make a user aware of the obstacles in the use of the data set.

The main aspects of these questions are as follows:

- Motivation for data set creation.
- Data set composition
- Data collection process
- Data preprocessing
- Data set distribution
- Data set maintenance
- Legal & ethical considerations

The static or dynamic boundary conditions under which the data was recorded helps the user of ML in BES. This is one question of the questionnaire: “What information (static and dynamic) can be given about the system(s) in which the data was recorded?”

The structure and design of the BES has a high influence on the energy efficiency of the individual components. It is not only decisive which components in what dimension the BES contains, but also their actual connection plays an important role. A graph-based system offers the possibility to map the topology of data points and facilities. The topology offers the possibility that e.g. all data points of a special plant are searchable.

An insight into the influences of connected systems is thus conceivable. Related questions are therefore as follows: “Are relationships between instances made explicit in the data? Are schemas of the technical equipment included? Is there a representation of the schemas as a graph model?”

Time series data often have gaps or are not complete. Therefore, we have added questions to this questionnaire in the section “Data Collection Process”: “How many gaps are in the data set? How can a user of the dataset recognize them?”

As the full questionnaire is very extensive, readers may refer to the corresponding Github repository for the complete questionnaire (https://github.com/RWTH-EBC/JUDO).

2.2 Just Unified Data set for building Operation (JUDO)

The building selected for this paper is the E.ON ERC main building in Aachen, Germany. It has a variety of different energy systems, providing a representative example of the building landscape. The energy systems include, among others, a combined-heat and power unit, a boiler, a geothermal field with 41 boreholes and comprehensive monitoring, a chiller, a sorption system, facade ventilation units, concrete core activation, underfloor heating, radiators. The zones of the building include offices, seminar rooms, computer rooms, workshops and laboratories. This data set contains different usage behavior of different systems at different times. More information about the building and the operation strategy of the building can be found in [23].

We added information about the dimensions of the BES and the connection of the systems with the Brick Schema to this dataset. We also provided instructions on how to interpret the data in the data set. The dataset includes all existing data points of the selected plants. This includes, for example, all status messages, valve positions and set points. Therefore, the data set shows the overall picture of the operation of the plants.

For the first standardized data set, we selected the time series of 765 data points of 31 technical devices in four representative months. Each of these months represents one of the four seasons. The name of this data set is JUDO (Just Unified Data set for building Operation) and is available at https://github.com/RWTH-EBC/JUDO.

This set includes the data points sorted by technical system (Table 1) and types (Table 2).

Table 1. Data points in JUDO sorted by technical system (main examples)

Technical System	Number of Data Points
Geothermal field	221
Heat Pump	181
Glycol cooler	68
Heat Exchanger	57
Concrete Core Activation	56
Boiler	29
Combined Heat and Power	26
Chiller	8

Table 2. Data points in JUDO sorted by data point type (main examples)

Data point type	Number of Data Points
Command	240
Temperature sensors	189
Status	81
Alarm	65
Volume Flow	58
Position	26

3 Process overview of generating generic simulated data sets

The whole process of generating Modelica simulation models from an ontology-based model (BrickModel) is a tool-chain of various special-purpose tools. Figure 1 shows the various steps of the process. First, a user fills out an Input Data File. Based on this file, a tool called BuToOn creates a brick model and a data property file, in which the user has the possibility to enter further information. At the same time, the tool BouGen determines the time series data for the boundaries of the model. OnWithData creates a common Brick model for all inputs. OnToMo converts the Brick model into a Modelica model.

The whole toolchain is written in Python. The subsystems of the simulation models are taken from the open source Modelica library Aixlib [24].

In the following, we describe the underlying data schema approaches of the toolchain and describe the toolchain itself.

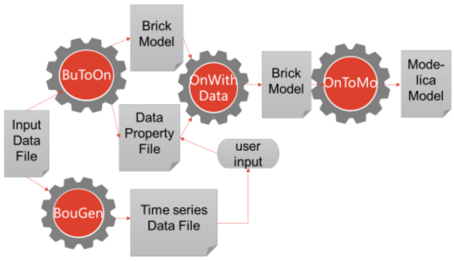

Figure 1. Process of generating Modelica simulation models from an ontology-based model

3.1 Used approaches

3.1.1 BUDO Schema

In this work, data points named by "buildings unified data point naming schema for operation management" (BUDO) [20] are used for the modelling. A BUDO key consists of several parts as shown in Figure 2. Each part provides a hierarchy level for the topology model and the corresponding simulation model. Information in system, subsystem, subsubsystem, medium/position and type of the BUDO key can be extracted to determine entity types in BrickModel.

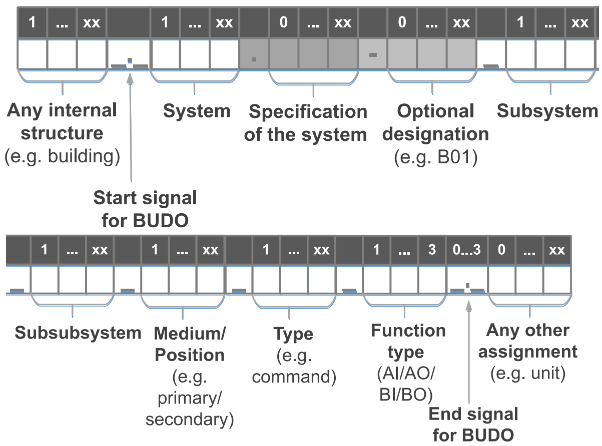

Figure 2. Structure of BUDO Key

3.1.2 BrickModel

The BrickModel is based on the Brick Schema developed by Balaji et al (2018) [21]. Figure 4 displays a sample of Brick's class hierarchy. Two classes of Brick are important for BrickModel: Equipment and Point. However, Brick can only describe which components are connected, but falls short in describing more detailed positions of the individual components, for example whether the primary or secondary side of a heat pump is connected to another equipment. In addition, the Modelica language uses connectors between two components. In order to describe the physical connection between equipments more precisely and meet the requirements of Modelica, we developed another class named Port to complement this ontology. One definition of these three important classes is as follows.

- Equipment: "Physical devices designed for specific tasks controlled by points belonging to it. E.g., light, fan, AHU" [21].
- Point: "Points are physical or virtual entities that generate time series data. Physical points include actual sensors and setpoints in a building, whereas virtual points encompass synthetic data streams that are the result of some process which may operate on other time series data, e.g. average floor temperature sensor" [21].
- Port: Ports represent the boundary and the physical connector of physical devices and store related static design data such as nominal mass flow.

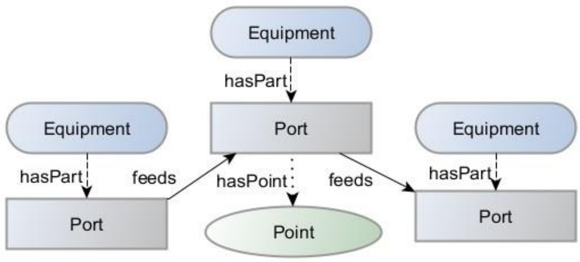

Figure 3. Information Concepts in BrickModel

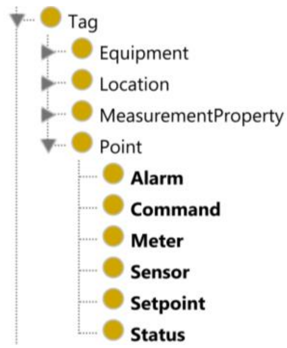

Figure 4. A subset of the Brick hierarchy [21]

As shown on the left side of Figure 5, we defined ports for heat flows, flows, electric currents, weather, equipment, comfort, transfer to rooms, control and fuels. Ports for heat flow are the most used in this work. Primary and secondary ports define the physical position of ports. In addition, we defined several data properties, which represent the design data of equipments and devices. On the right side of Figure 5, there is a subset of the data property hierarchy.

Data properties such as nominal mass flow, nominal temperature and nominal pressure difference are defined, which can be used to set corresponding parameters required in Modelica simulation model. Furthermore, relationships in Brick connect the different entities in the building and describe the physical topology. Table 3 summarizes the definitions of relationships used in BrickModel. Figure 3 demonstrates how relationships connect entities of different classes with each other in BrickModel.

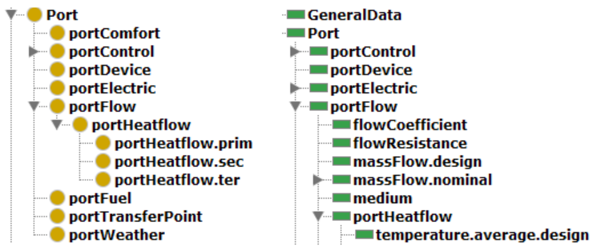

Figure 5. A subset of the port class and data property hierarchy

Table 3 List of the used Brick relationships and their definition

Relationship (Inverse)	Definition	Endpoints
hasPart (isPartOf)	A has some component or part B (typically mechanical)	Equip./Port Equip./Equip.
hasPoint (isPointOf)	A is measured by or is otherwise represented by point B	Equip./Point Port/Point
feeds (isFedBy)	A "flows" or is connected to B	Port/Port

3.2 Generating the BrickModel

The starting point for the workflow is manually gathering information and storing them in an Excel file. In addition, users can choose to specify the period of the time series and the data points they want to integrate into the Modelica simulation model.

If the user has filled out the Input Data File, BuToOn generates a BrickModel. The first generated BrickModel still lacks some information required by the Modelica simulation model. Therefore, BuToOn generates a data property file at the same time. The user has to input corresponding parameter values for the Modelica Model in this file. A tool named BouGen extracts the corresponding data from the time series data set and writes them into a file readable by Modelica. For the requested period, the time series database provides the corresponding time series for each data point and BouGen stores them in a Modelica-compatible format.

The tool OnWithData combines the Brick Model from BuToOn with the additional information from the data property file and the generated files of the time series data from BouGen and creates a BrickModel from all previous information.

3.3 Transformation To Modelica model

In the final step, the tool OnToMo extracts the information of the BrickModel and generates the corresponding Modelica model. We developed this tool using a code templating tool (CoTeTo) [25] and the SPARQL Protocol And RDF Query Language (SPARQL) [26].

The approach to generate Modelica code is template-based. CoTeTo supports the template engine Mako [27] that allows users to predefine templates for the corresponding Modelica code and conveniently store templates for each Modelica module in a separate template document.

SPARQL queries specify constraints and patterns of triples, and traverse the BrickModel to return those that match. Table 4 shows two examples of extracting information from BrickModel.

The first example demonstrates a SPARQL query for searching an instance of class Pump. The query searches in BrickModel for instances that match the triple pattern and return them.

The second example shows the extraction of the physical topology of the model, which is used to connect the modules. The existence of the Port class provides sufficient connection information for Modelica. This query returns the port, which is fed by the specified pump port as result.

After extracting information from BrickModel, the toolchain generates the Modelica simulation model.

Table 4. SPARQL Query Examples

SELECT ?instance WHERE {?instance rdf:type brick:Pump.}
SELECT ?port WHERE { ex:BL-4120_._HX-H03_PU-M02_WS.H.SEC_PH bf:feeds ?port.}

4 Modelling Use Case

To prove the functionality of the tool-chain, several use cases are developed based on the main building of E.ON Energy Research Center. In total, 13 simulation models, 244 modules and 211 connections between modules were established and 2153 lines of Modelica code were automatically generated. In this section, only one selected use case is illustrated: a heat pump system with real time series data. As shown in Figure 6, the heat pump system consists of one heat pump and four temperature sensors. Figure 7 shows the corresponding Modelica model. During the modelling process, our developed approach transforms and connects all the modules automatically and correctly. In addition, this model successfully integrates time series data of temperature and mass flow from a database as boundary conditions. Moreover, for the control of the heat pump, the time series data of the electrical power of the compressor are integrated into the model of the heat pump. The heat pump does not provide a control signal. This creates uncertainty if the signal integrated into the model will affect the real behaviour.

Figure 8 compares the simulated and actual measured return water temperatures of the condenser side of the heat pump. The blue curves are the simulation results and the green curves represent the measured data. It is obvious that the simulated data of the heat pump model are in good agreement with the measured data, although there are still occasional deviations. The Root Mean Square Error (RMSE) of the return temperature on the condenser side is 1.414 K and the Normalized Root Mean Square Error (NRMSE) amounts to 0.086. For the evaporator side, the return temperature reaches a RMSE of 2.040 K and a NRMSE of 0.156.

We use this model to generate more rare time series for a classification approach. For this purpose, we scale the power and mass flows of the heat pump shown here.

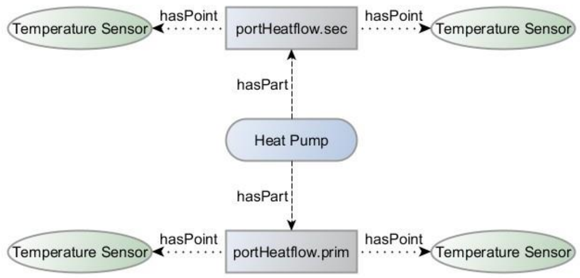

Figure 6. BrickModel structure of a heat pump system

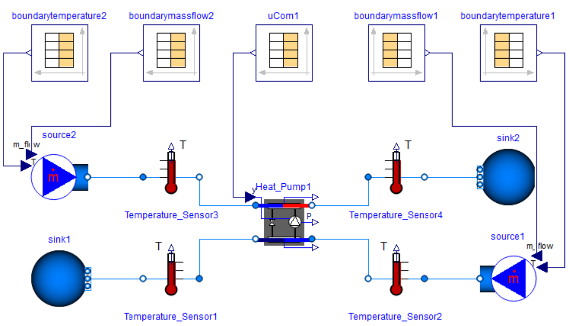

Figure 7. Modelica simulation model of a heat pump system

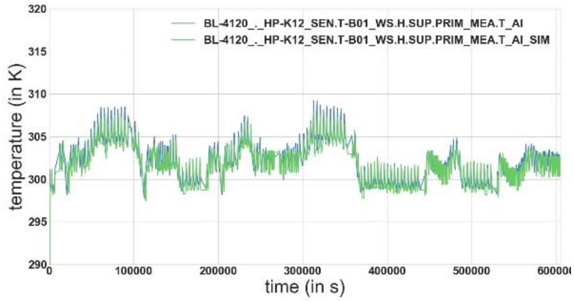

Figure 8. Comparison of simulation result with measured data of heat pump return water temperature on condenser side. Corresponding to Temperature_Sensor4 in Figure 7

5 Classification Use Case

In this section, we demonstrate how generated data sets can be used to train algorithms for classification of real time series classification. We train ML classifiers with both real and generated data points. For this purpose, two heat pump temperature time series (T_HP.Cond, T_HP.Evap) are generated with the model illustrated in Figure 7. We used the following further classes from real measurements:

- room air temperatures (T_AIR)
- temperature of concrete core activation in room automation (T_RA_CCA)
- temperature of concrete core activation in distribution (T_DIST_CCA)
- façade ventilation units in room automation (T_RA_FVU)

Due to the small temperature difference, the flow and return of the CCA and FVU data points are combined in one class in each case. The final data set contains the following data points:

- 180 samples from each T_HP.Cond and T_HP.Evap
- 340 samples from T_AIR
- 76 samples from T_RA_CCA
- 120 samples from T_ DIST_CCA
- 100 samples from T_RA_FVU

Each sample consists of the data point's label and time series data of one month. The used time series have a resolution of one minute. We detect and eliminate outliers with hampel filter [31].

For the training of the classifiers except of heat pump, we split the data set into 70% training data and 30% testing data. We extract six statistical features, following the recommendations published in [28]. We have also sorted out time series that did not have any usable statistical characteristics. We apply nine of the most suitable classifiers for real world classification problems according to, using the implementations provided by scikit-learn [29]. In Figure 9, we present the classification scores of the algorithms when applied to the test data set.

Four out of the nine algorithms reach more than 70 % classification accuracy [30]. The best performing classifier is "Gradient Boosting" with 80.7% top score.

For the second test case, we use 10 real temperature measurements from both T_HP.Cond and T_HP.Evap to our test data set. Figure 10 shows the classification done by classifiers. "AdaBoost" classifier correctly assigned 9 of 10 of T_HP.Cond and 8 of 10 T_HP.Evap.

We conclude that the training of classification algorithms with simulated data also has a high potential for data point mapping in modern building automation systems. However, for a robust application in building automation further investigations are necessary.

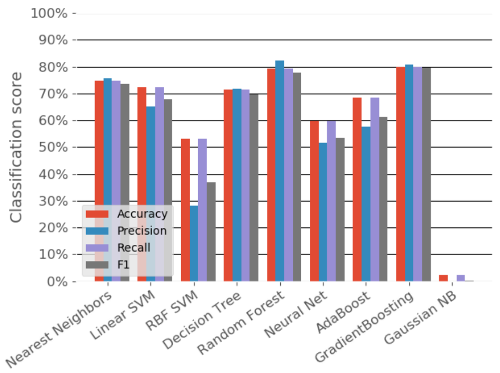

Figure 9. Scores of nine selected classifiers trained with 70 % of the real data set and tested with 30 % of it

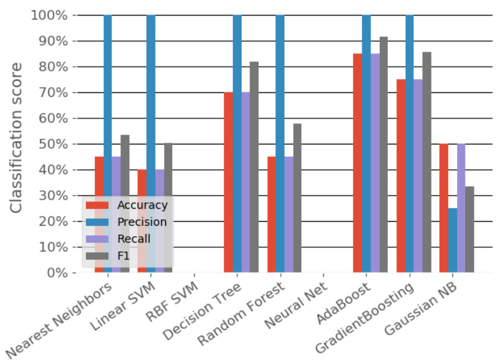

Figure 10. Scores of nine selected classifiers trained with simulated data sets and tested with real data sets of Temperatures of Heat Pump

6 Conclusion

In this paper, we presented a data set extracted from the real operation of a multifunctional building with various energy plants, the Just Unified Data set for building Operation (JUDO). We extended a special questionnaire for Machine Learning (ML) users with specific questions for Building Energy Systems (BES) and filled it in for JUDO. This questionnaire can be used for further data sets from BES. Consequently, we named the data set with a labelling schema for building data (BUDO). In future, we provide a Brick model of this.

We were able to show that our developed toolchain can semi-automatically generate simulation models using the standardized data point identifiers of time series data sets from the operation of BES. Finally, the toolchain creates an ML-compatible simulation data set. An extension to arbitrary periods and facilities of this and other buildings is possible.

We were able to apply this approach to the ML application of data classification. The current approach showed first good results but there is still a great need for research. We expect that a combination with calibration methods could increase the automation and accuracy of the approach. We will expand the considered technical facilities so that we can simulate a wider field of BES. We will extend the existing data set of currently 765 data points by further data points. We also consider an automatic generation of the currently required static information of the technical equipment from data sheets.

7 Acknowledgements

The authors would like to acknowledge the financial support of the German Federal Ministry of Economic Affairs and Energy promotional reference 03SBE0006A.

8 References

[1] Deb C, Zhang F, Yang J, Lee SE, Shah KW. A review on time series forecasting techniques for building energy consumption. Renewable and Sustainable Energy Reviews 2017;74:902–24.

[2] Koh J, Balaji B, Akhlaghi V, Agarwal Y, Gupta R. Quiver: Using Control Perturbations to Increase the Observability of Sensor Data in Smart Buildings; 2016.

[3] Benndorf GA, Wystrcil D, Réhault N. Energy performance optimization in buildings: A review on semantic interoperability, fault detection, and predictive control. Applied Physics Reviews 2018;5(4):41501.

[4] Wei L, Tian W, Silva EA, Choudhary R, Meng Q, Yang S. Comparative Study on Machine Learning for Urban Building Energy Analysis. Procedia Engineering 2015;121:285–92.

[5] Molina-Solana M, Ros M, Ruiz MD, Gómez-Romero J, Martin-Bautista MJ. Data science for

building energy management: A review. Renewable and Sustainable Energy Reviews 2017;70:598–609.

[6] Fierro G, Pritoni M, AbdelBaky M, Raftery P, Peffer T, Thomson G et al. Mortar: An Open Testbed for Portable Building Analytics. In: The 5th ACM International Conference on Systems for Built Environments (BuildSys '18); 2018.

[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 2017;122.

[8] Kolter JZ, Johnson MJ. REDD: A Public Data Set for Energy Disaggregation Research. In: SustKDD 2011; 2011.

[9] Barker S, Mishra A, Irwin D, Cecchet E, Shenoy P, Albrecht J. Smart*: An Open Data Set and Tools for Enabling Research in Sustainable Homes. In: SustKDD12; 2012.

[10] Abrol S, Mehmani A, Kerman M, Meinrenken CJ, Culligan PJ. Data-Enabled Building Energy Savings (D-E BES). Proceedings of the IEEE 2018;106(4):661–79.

[11] Babaei T, Abdi H, Lim CP, Nahavandi S. A study and a directory of energy consumption data sets of buildings. Energy and Buildings 2015;94:91–9.

[12] Miller C, Meggers F. The Building Data Genome Project: An open, public data set from non-residential building electrical meters. Energy Procedia 2017;122:439–44.

[13] Wang Y, Feng H, Qi X. SEED: Public Energy and Environment Dataset for Optimizing HVAC Operation in Subway Stations; 2013.

[14] Verein Deutscher Ingenieure. VDI 6041 - Facility-Management - Technisches Monitoring von Gebäuden und gebäudetechnischen Anlagen; 2017.

[15] Nikolaou T, Skias I, Kolokotsa D, Stavrakakis G. Virtual Building Dataset for energy and indoor thermal comfort benchmarking of office buildings in Greece. Energy Build. 2009;41(12):1409–16.

[16] Henriet S, Simsekli U, Richard G, Fuentes B. Synthetic Dataset Generation for Non-intrusive Load Monitoring in Commercial Buildings. In: Proceedings of the 4th ACM International Conference on Systems for Energy-Efficient Built Environments. New York, NY, USA: ACM; 2017, 39:1 - 39:2.

[17] Gebru T, Morgenstern J, Vecchione B, Vaughan JW, Wallach H, Daumeé H III et al. Datasheets for Datasets; 2018.

[18] Bhattacharya A, Ploennigs J, Culler D. Short Paper: Analyzing Metadata Schemas for Buildings: The Good, the Bad, and the Ugly. In: Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments. 2821674: ACM; 2015, p. 33–34.

[19] Reiß & Rommerich. Bericht zum Thema Building Information Modeling (BIM): Bundesweite Befragung der Mitglieder der Architektenkammern der Länder; 2017.

[20] Stinner F, Kornas A, Baranski M, Müller D. Structuring building monitoring and automation system data. In: REHVA, editor. The REHVA European HVAC Journal - August 2018; 2018, p. 10–15.

[21] Balaji B, Bhattacharya A, Fierro G, Gao J, Gluck J, Hong D et al. Brick Metadata schema for portable smart building applications. Appl. Energy 2018;226:1273–92.

[22] Haynes A. ASHRAE's BACnet Committee, Project Haystack and Brick Schema Collaborating to Provide Unified Data Semantic Modeling Solution. ATLANTA, BERKLEY, CA and RICHMOND, VA; 2018.

[23] Bode G, Fütterer J, Müller D. Mode and storage load based control of a complex building system with a geothermal field. Energy and Buildings 2018;158:1337–45.

[24] Müller D, Lauster M, Constantin A, Fuchs M, Remmen P. AixLib - An Open-Source Modelica Library within the IEA-EBC Annex 60 Framework. In: BauSim 2016; 2016, p. 3–9.

[25] Nytsch-Geusen C, Inderfurth A, Kaul W, Mucha K, Rädler J, Thorade M et al. Template based code generation of Modelica building energy simulation models. In: Proceedings of the 12th International Modelica Conference, Prague, Czech Republic, May 15-17, 2017. Linköping University Electronic Press; 2017, p. 199–207.

[26] Prud'hommeaux E, Seaborne A. SPARQL query language for RDF 2007.

[27] Bayer M. Mako Templates for Python. [January 02, 2019]; Available from: https://www.makotemplates.org/.

[28] Fernández-Delgado M, Cernadas E, Barro S, Amorim D. Do We Need Hundreds of Classifiers to Solve Real World Classification Problems? J. Mach. Learn. Res. 2014;15(1):3133–81.

[29] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O et al. Scikit-learn: Machine Learning in Python 2011.

[30] Nanopoulos A, Alcock R, Manolopoulos Y. Feature-based Classification of Time-series Data: Information Processing and Technology. In: Mastorakis N, Nikolopoulos SD, editors. Commack, NY, USA: Nova Science Publishers, Inc; 2001, p. 49–61.

[31] Pearson R, Neuvo Y, Astola J, and Gabbouj M. Generalized hampel filters. EURASIP Journal on Advances in Signal Processing, 2016(1):115, 2016.