Comparative study of supervised algorithms for topology detection of sensor networks in building energy systems

Florian Stinner[a,**] (Co-ordinator), Belén Llopis-Mengual[a,b,*] (Researcher), Thomas Storek[a,*], Alexander Kümpel[a,*] and Dirk Müller[a]

[a]RWTH Aachen University, E.ON Energy Research Center, Institute for Energy Efficient Buildings and Indoor Climate, Mathieustrasse 10, 52074 Aachen, Germany

[b]Instituto Universitario de Investigación de Ingeniería Energética (IUIIE), Universitat Politècnica de València (UPV), Camino de Vera s/n, 46022 Valencia, Spain

ARTICLE INFO

Keywords:
Model-based time series generation
topology detection
building energy systems
relation inference
building automation
supervised learning

ABSTRACT

Optimizing the operation of building energy systems holds great potential to reduce energy consumption in buildings. However, this requires detailed system information, such as the relationship of sensor data. Automatic detection of this information requires monitoring data from buildings, which is rarely available in the needed quality for automatic assignment. This study bases on 200 weeks of data collected from eight temperature sensors of a heat pump and a heat exchanger in 5-minute samples. We use this data to auto-generate grey-box models to extend the data set with 500 weeks of simulated data. We train six supervised deep learning algorithms with all the data to test whether detecting connections is possible. The maximum F1 score of 94.9 % compared to real-based results with a maximum of 34.2 %, which is over 60 % better. The advantage of the proposed approach is its independence from the low availability of real data.

1. Introduction

Climate change is the greatest economic challenge of the present and future [1]. Including indirect emissions, buildings represent 36 % of European CO_2 emissions [2]. In existing buildings, there is an increased need for CO_2 reduction, which cannot be met by a current renovation rate of 1 % [3], as 3 % would be required [2]. Therefore, it is necessary to implement automated measures to reduce emissions of the building stock.

Especially, non-residential buildings are equipped with complex building automation systems (BAS). Improved control can reduce the total energy consumption in non-residential buildings equipped with BAS by approximately 20-30 % [4]. Advanced control systems can include reinforcement learning [5], model predictive control [6, 7] or occupant-centric control systems [8, 9]. Machine learning can be applied in every stage of building energy system's life cycle [10, 11].

Although most building is unique, modern control and analysis methods can be developed in a scalable manner. For example, the components of a building energy system (BES) (e.g., air handling units or chillers) reoccur in buildings, and their operation is similar despite different manufacturers. The components are connected to each other in a similar way by means of ducts or pipes. However, novel control approaches require detailed information about the BES to be controlled.

*Corresponding author
**Principal corresponding author
fstinner@eonerc.rwth-aachen.de (F. Stinner); belen.llopis@iie.upv.es (B. Llopis-Mengual); tstorek@eonerc.rwth-aachen.de (T. Storek); akuempel@eonerc.rwth-aachen.de (A. Kümpel)
https://www.researchgate.net/profile/Florian_Stinner (F. Stinner)
ORCID(s): 0000-0002-6638-4276 (F. Stinner); 0000-0003-0238-4231 (B. Llopis-Mengual)

Graph-based models based on ontologies can represent this information. However, since no current building ontology represents building operation well [12], the Brick Schema [13] has been developed and is suitable to address this issue. Information includes the types of data streams in the building, the contained technical building equipment (TBE) and the interconnections of the TBE. In the following, we refer to the interconnection of TBEs as *topology*.

Topology mapping can be used to study the proper use of energy flows in buildings, to find the source of error for faulty operation or as input of BES modelling. Especially, when an accurate model of the building is required, as with model predictive control [6, 14], the exact mapping of data streams and the topology is essential. This mapping is often not available in a directly analyzable form.

Data streams in BAS often have labeling guidelines that differ depending on the building and operator [12, 15]. The information from these labels is usually very labor-intensive to extract [15]. Labels often only contain information about the type of data stream (e.g. temperature measurement) and possibly the TBE (e.g. air handling unit). Information about the interconnection of TBE and therefore topology of BES is mostly missing. A correlation of labeled sensors with piping and instrumentation diagrams is difficult due to the lack of standardized sensor and actuator labeling in BES diagrams[16]. Building information models (BIM) could also provide this information [17]. However, their application in existing buildings is not yet widespread, and information on BES operation is scarce [18]. However, deriving the topology from available time series data is a promising scalable approach.

Time series data of BAS provide a valid source if TBE are connected. When a component of TBE starts up or is switched on, a neighboring TBE component in downstream direction reacts to this switching operation. For example, the value of a temperature sensor located at a boiler's output will increase if the boiler is switched on. This phenomenon can be used for the topology detection (TD) of BES.

In previous approaches, mainly unsupervised learning was used to identify the topology. Since the number of types of TBE in energy systems is limited, supervised learning can also be used for TD.

A major problem for the implementation of supervised learning for TD is the lack of good publicly available data sets [19, 20]. In particular, rare connection types are either not included in public data sets, or are included too infrequently. This is also related to the number of types of systems in buildings. For example, an air handling unit or variable air volume box is more often present in a specific building than a boiler, a heat pump or a combined heat and power device. However, the availability of building simulation models has increased [17]. In [19], a toolchain has been introduced to generate building simulation models based on labels of data streams in buildings. These models can generate data streams of sensors and actors containing combinations of TBE and the signature that occurs when a change of state occurs.

The approach presented in this work aims to learn the relation inference (topology) of a multi-functional office building. The available monitored data comprises the historic data for four water temperature data streams from a heat pump and four from a heat exchanger, collected over 200 weeks in 5 minutes samples. For this purpose, this study mainly investigates two aspects:

1. how to use this information from the data streams to generate generic simulation models and thus extend the data set,
2. how to apply current methods of supervised learning to detect the connection between the data streams

The structure of the paper is as follows: The following section introduces the background, related work, and are discusses potential methods . In the method, we first describe the toolchain to create of building energy system models. Here, the generation of generalized time series using the models is explained. Second, the developed use cases for the supervised detection of building energy systems are presented. Afterward, we introduce the toolchain to apply six used deep learning multivariate time series classification algorithms. We investigate the toolchain's performance in three different use cases. The results are classified and discussed and it is explained how to develop and use the presented toolchain in future work.

2. Related work

This term is adopted here since topology detection in electrical grids is the most researched. Moreover, events are prevalent in electrical networks compared to thermal networks. Thus, Huchtkoetter and Reinhardt [21] recommend a resolution of about 1 kHz for event detection in power grids. In buildings, a temporal resolution of at least 0.03 Hz is joint. More approaches exist in the literature for topology detection in power grids due to the higher number of measurements in electrical systems. Topology detection also has the name relation inference in building energy systems. In electrical grids, the term "topology detection" is more common.

We use the following definition for the term data stream: a data stream is an information carrier that continuously provides information about a state [19]. Supervised learning supports the automatic determination of data stream types in building automation systems. Three different types of inputs must be distinguished: time series of data streams, their features like physical unit, and their labels. Furthermore, hybrid versions of the inputs exist [15].

Wang et al. [15] gives an overview of different methods for automatic data stream mapping in building automation systems. However, one problem in comparing different approaches to topology detection and metadata extraction is the comparability between the approaches. Different connection types (topology detection) or different data stream types (metadata extraction) are used. The use of standardized data sets or the publication of the test and training data supports the development of an algorithm and its comparison. We analyzed various publicly available data sets of building automation systems to see if they had suitable time series for our approach. However, none of the 120 found papers and data sets contained appropriate time series data for our use cases. Most data sets were only at the aggregation level or contained only electrical data. However, this is not the focus of our approach.

Kazmi et al. [22] has analyzed different energy data sets for their frequency and containing data. None of the analyzed data sets contained data on the topology of the BES. If thermal usage data was present, it was only at the aggregation level and included mainly heat flows.

The Mortar data set [23] is the most promising data set with time series data from 107 buildings. Nevertheless, it also hardly contains any data on the waterside of energy production and its direct distribution. For example, the tag "boiler" appears in only one building. This amount of data is usually not sufficient for deep learning processes. However, it is a good database for topology detection of air handling unit based systems.

We also analyzed whether the papers cited in the following have suitable time series for our use case or data sets publicly available. Unfortunately, none of the papers on topology detection provided the data sets to be directly usable for the use cases used here. Either parts of the data set are missing, as can be assumed from the source codes, the time series data itself was not included, or the used connection types did not correspond to the connection types used here.

Current and past research focuses mainly on detecting data stream types based on the associated time series and labels. Detecting connections between two or more components in building energy systems are rarely the subject of research. In the following, we show the related research approaches known by the authors.

Zhou [24] uses an expert-based approach to identify typical patterns of temperature responses to a switching signal. These include linear, exponential, step-based, and peak patterns. Patterns that do not correspond to these cases are challenging to detect.

Active control offers an alternative. Pritoni et al. [25] and Koh et al. [26] use active control of the TBE to detect further connected TBE. Active control is not applicable in buildings during normal operating hours. Otherwise, the comfort in rooms is compromised. Fürst et al. [27] identify relationships based on a human-in-the-loop approach, where users either perform actions
(switch on/off) or read information (temperature display in the room). If every room with its relationships has to be identified manually by a user, this can cause a lot of manual work and related costs.

Hong [28] has closed the research gap from the detection of data stream types to the detection of the topology of an energy system. The topology detection uses an unsupervised procedure that first generates a Markov event model. This model identifies transitions and assigns them to the associated events based on this model. These events are filtered so that only those events remain that are unique between the systems. However, only air handling units (AHU) connected to variable air volume systems (VAV) were detected. This approach corresponds only to the airside of the connections within the building and disregards the waterside supply.

The same connection types have been considered by an approach of Li et al. [29] using supervised learning based on Short-Time Fourier Transformation with Triplet Network (STN). Its advantage is that it can extract highly nonlinear features. However, the approach only deals with the supply of air and thus does not address the waterside of the building.

To the best of the author's knowledge, no supervised learning approach exists for the water-based heating and cooling system of building energy systems other than an approach of Stinner et al. [30], which only achieves a maximum accuracy of 52.1 %.

According to Wang et al. [15], supervised learning is an established approach for the identification of types of data streams in buildings. The labels, metadata, and data streams themselves provide input here. Therefore, supervised learning is a suitable method for the classification of data streams types, achieving over 90 % accuracy [15]. In a typical building, there are only a limited number of connection types between different technical systems. For example, in thermal systems, the temperature is a signal that reacts strongly to changes in the previous system and, therefore, its temperature signal. This reaction corresponds to multivariate time series.

This work shows that supervised learning can detect individual connections in BES can be detected by supervised learning based on multivariate temperature signals. For supervised learning of multivariate sensor data, we use six classifiers for multivariate time series classification [31, 32, 33, 34]. We use convolutional neural networks (CNN), which performed in the top group in time series classification on the UCR time series archive [35]. For correct classification, CNN requires more time series than classical methods (e.g., random forest). However, they offer the potential that they can classify in a generalized manner [35].

A problem with the application of CNN-based supervised learning in BES is the lack of historical time series data of data streams from BES and the lack of documentation of the BES. However, this is crucial for the usage of data in topology detection. Physical simulation models can be used to generate time series from BES data streams. The evaluation of BES in connection with their usage is the primary usage of synthetic data based on grey-box models [36, 37]. Nevertheless, the use of simulation data to feed machine learning algorithms is an option used especially in fault detection [38].

Stinner et al. [19] developed a toolchain for generating generic data streams based on Modelica models for detecting data stream types. However, the approach is limited to only a single heat pump. Here, we further developed this toolchain and extended it for scalable use. We take labels named using the BUDO Schema and export them to an ontology model using the Brick Schema. The Brick schema is able to represent the connections of technical systems (e.g. using pipes). We use a Design of Experiment approach to generate generalized data, which identifies and validates parameters in simulation models. Generalized data have the advantage that not only the connection of a specific technical system can be detected, but a more comprehensive range of systems as well.

3. Methodology

The implemented toolchain consists of two parts: in subsection 3.1, we describe the process of generating the generic time series data using grey-box simulation models. The second part consists of the preprocessing of the time series and the used supervised machine learning algorithms, which is introduced in subsection 3.4. We illustrate the entire process chain in Figure 1 with the required inputs. The program code (classifier and toolchain) and the used data sets are stored separately in the repository [39] and are published under the MIT license (link: `https://github.com/RWTH-EBC/Deep-learning-supervised-topology-detection`).

3.1. Toolchain for generating generic data sets for Machine Learning applications

While generating data sets for machine learning tasks, we have developed a tool that permits us to take the information contained in BUDO schema (a standard to label data streams in buildings [40]) and transforms it to Brick schema. Therefore, we obtain a model in Modelica of the real system [41] that can be simulated for obtaining time series, for instance, for machine learning applications.

Following the schema in Figure 2, the tool performs the following components:

1. **BouGen**: Downloads time series data from real sensors in **.mat* file format.

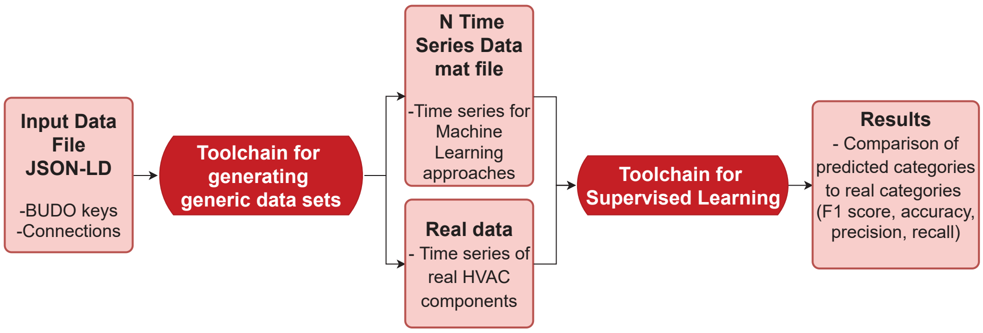


Figure 1: Process overview of the toolchain for supervised learning algorithms of topology detection.

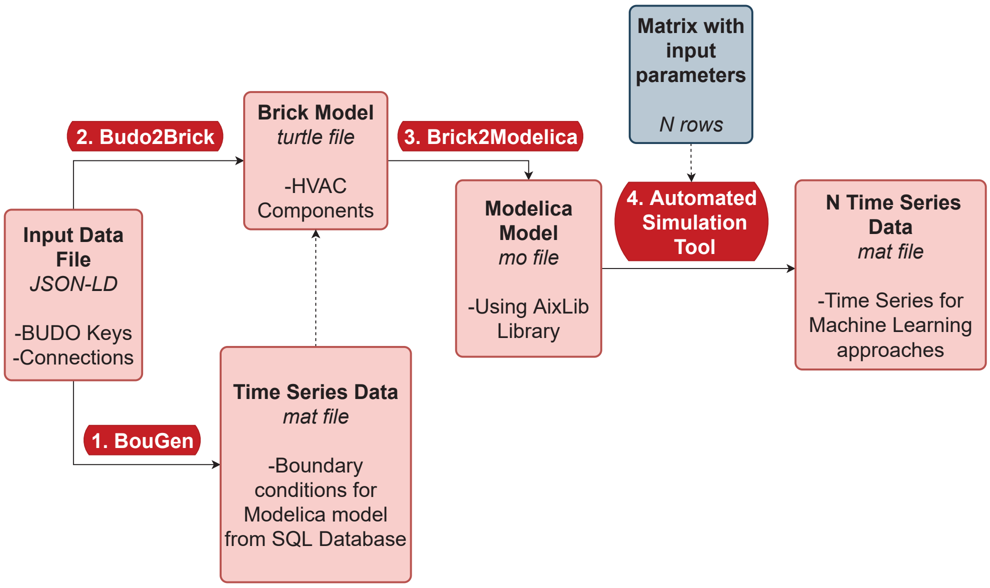


Figure 2: Process overview of the toolchain for generating generic data sets.

2. **Budo2Brick**: In this step, we use JSON-LD data, a mechanism of encoding linked data using JSON. It is advantageous to utilize if we use terminology from different ontologies and schemes. Thus, Budo2Brick, takes the information from the BUDO keys in JSON-LD file and transforms it into Brick, using the *turtle* format for the model. Besides, it adds the data properties and boundary conditions of the system.
3. **Brick2Modelica**: In this step, we extract data from the Brick model, and generate the Modelica model. This process is done using the SPARQL Protocol, and RDF Query Language [42].
4. **Automated simulation tool**: After the generation of the model in Modelica, our tool automatically simulates the model using the *Dymola-Python interface* [43]. The tool takes parameters from a matrix to modify each simulation and obtains the desired time series. A Design of Experiments approach provides the appropriate matrix of different parameters that vary in the simulations. Thus, with this tool, all the simulations can be generated automatically with the model in Modelica, and the data sets can be computed for machine learning purposes.

3.1.1. JSON-LD and BudoOnt Ontology

It is important to emphasize that we describe the initial information of the system in JSON-LD format. As JSON-LD uses terms linked by ontologies, the information in BUDO schema must be represented by ontologies. In addition, we need further terms relating to HVAC systems that Brick schema does not contain but that we require to create the Modelica model of an energy system. For these reasons, we developed the BudoOnt ontology, previously initiated by Stinner et al. [19] and extended it in this work.

Figure 3 shows some of the classes added to the BudoOnt ontology, where the hierarchical structure of the BUDO schema (e.g., system, subsystem) and other terms for detecting the connections of the data streams are defined. In figure 3 there

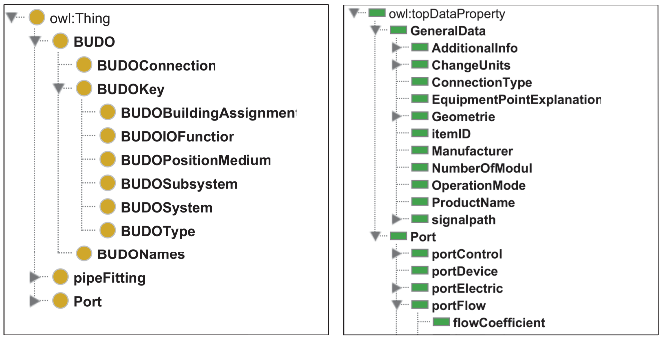


Figure 3: Subset to illustrate BudoOnt ontology in *Protégé*

is also a subset of the data properties of the BudoOnt ontology. In this case, we add some properties to facilitate the transformation from the initial input to Modelica. For instance, terms for the conversion of units of measurements, concepts to know the nature of the connection, or specific properties of a port.

The implementation of the JSON-LD input file in Python is done through the package *pyld* [44]. The syntax of JSON-LD requires a context and the document. A context is used to map terms to IRIs (Internationalized Resource Identifier). Following the example of the context depicted in figure 4, first, the ontologies to be used and their IRIs are defined, and then each of the terms that are going to be used in the document and to which ontology previously defined they belong. For instance, the term *TimeEnd* belongs to the ontology *Schema*, so when it is used in the document, it means that it has to follow the definition given by this ontology. In the case of HVAC terms, this file has terms from Brick like *Zone* and terms from BudoOnt like *BUDOBuildingAssignment*. Thus, the framework of this file is fully characterized.

```
context = {
        "schema": "http://www.w3.org/2001/XMLSchema#",
        "brick": "http://brickschema.org/schema/1.0.3/Brick#",
        "BUDO-M": "https://git.r[...]",
        "Zone": {"@id": "brick: HVAC Zone"},
        "TimeStart": {"@id": "schema: StartDate"},
        "TimeEnd": {"@id": "schema: EndDate"},
        "BUDOBuildingAssignment": {"@id": "BUDO-M: BUDOBuildingAssignment"}}
doc = {
        "@context": context,
        "Zone":"K12",
        "TimeStart": "2016-07-09 00:00:00",
        "TimeEnd": "2016-07-16 00:00:00",
        "BUDOBuildingAssignment": "BL-4120" }
```

Figure 4: *Context* of a JSON-LD input file

The rest of the document is intuitive since it follows a syntax practically identical to the JSON data structure. JSON is organized in key-value pairs, being the keys the names of the terms previously defined in the context.

3.2. Study Area

We apply the developed methodology to the main building of the E.ON Energy Research Center located in Aachen, Germany [45]. Its energy system consists of a ground source heat pump, two condensing boilers, and a gas-fired CHP. A chiller completes the energy conversion as a cold producer. The energy system supplies different offices and laboratories with distribution systems such as concrete core activation or facade ventilation. The distribution systems have three different temperature levels: a low temperature (35 °C), a high temperature (87 °C) and a cold temperature (10 °C).

Figure 5 shows the investigated part of the energy system, consisting essentially of the heat pump (HP) and a heat exchanger (HX) between the high and low-temperature loops. The heat pump transfers energy between two sources: the cold side, which is connected to a cold storage tank at about 10 °C (T_2) and returning to the same tank (T_4). The hot side comes from a heat storage tank at 35 °C (T_1), and the outlet of this loop returns to it (T_3). At the same time, if required, heat can be produced utilizing a Combined Heat and Power (CHP) system and two condensing boilers at a temperature of 87 °C. The boilers and CHP are connected with the distribution network by a hydraulic separator.

As these systems cause the water to be heated up to about 87 °C, this is used to heat the water coming from the hot tank if required. This is regulated by the cold side of the heat exchanger through a three-way valve, which depending on the temperature coming from the hot tank, the temperature required in the distribution systems, and the temperature generated by the high-temperature systems, opens and passes through the heat exchanger or goes directly to the distribution systems. This is shown in figure 5, where T_1 and T_2 in the heat exchanger represent the high-temperature side coming from these heating systems, and T_3 and T_4 the low-temperature side coming from the heat pump, with the entrance to the heat exchanger regulated.

More systems related to these take part in them, but they will be isolated from the rest, and the cases in this work will focus on the heat pump and the heat exchanger.

3.3. Obtaining the simulated time series from the model

The toolchain described above is used to obtain the models in Modelica of the heat pump and the heat exchanger. The models originate from the *AixLib* library [46]. In the case of the heat pump, the model uses a temperature-dependent coefficient of performance (COP); the heat exchanger model adopted from this library uses constant efficiency.

These models offer more data streams, such as temperatures and volume flow rates, than measured in the real system. Thus, augmenting the data sets with a Design of Experiments (DoE) methodology is possible. We choose Taguchi orthogonal arrays for DoE [47]. This approach is preferred over other traditional DoE, such as full factorial design or central composite design. When having several levels for each factor, a full factorial design has a too high cost (computational time) since with 5 factors and 3 levels, 243 simulations would be needed. It would be possible to decrease the number of levels, but then the variability and similarity to the real time series would decrease. For these reasons, Taguchi proves to be more efficient than other DoE method-

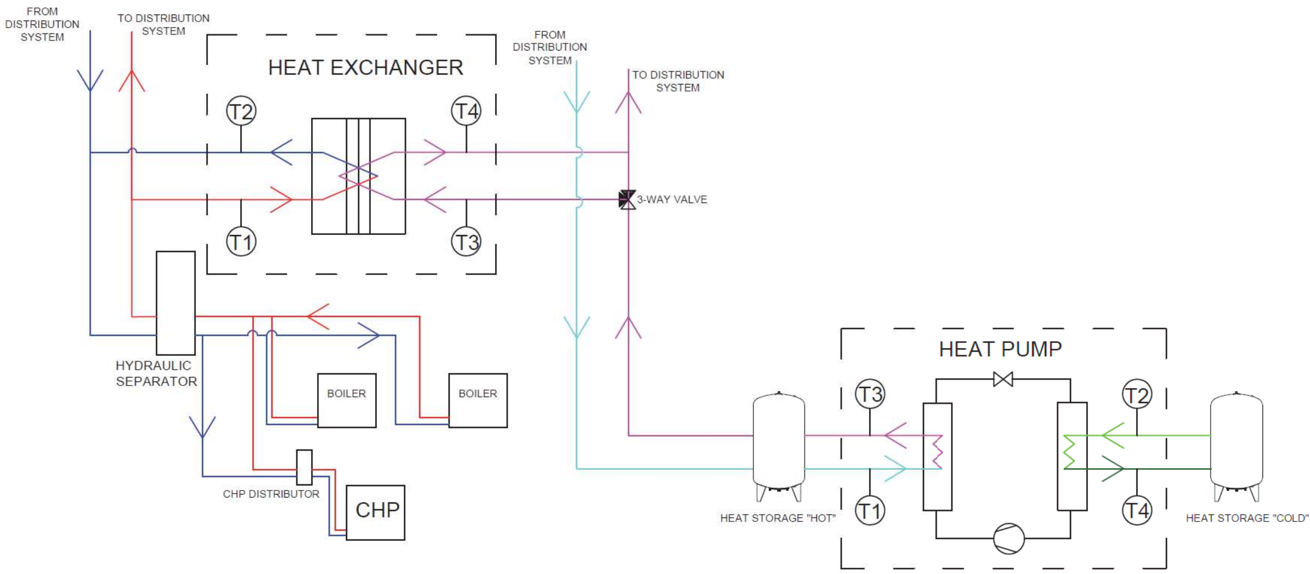


Figure 5: Energy systems of the use case.

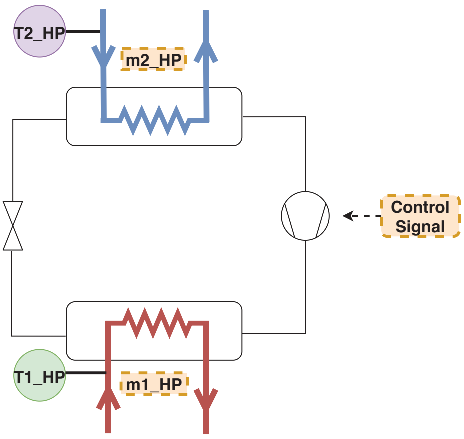


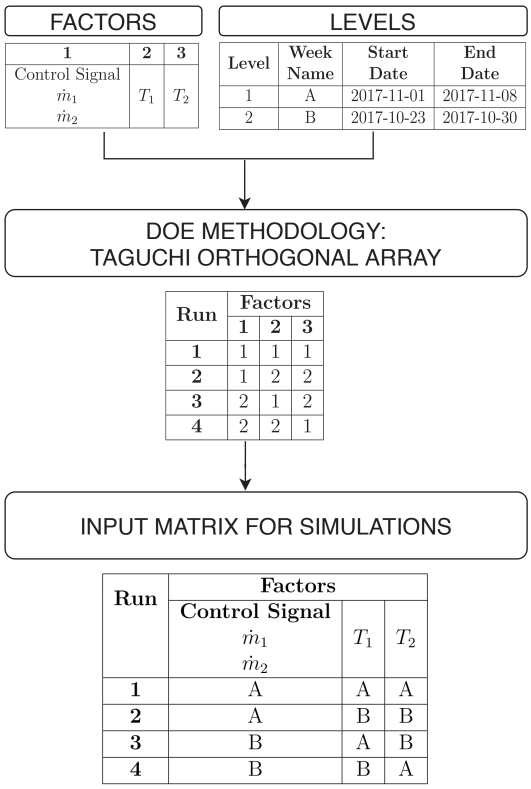
Figure 6: Factors that vary in the heat pump model simulations.

ologies. It minimizes the number of simulations to perform without decreasing the accuracy substantially.

3.3.1. Heat Pump

To design the orthogonal array, we first settle which factors to modify in each of the simulations and in which levels they vary. In the case of the heat pump model, we decide these factors (see figure 6): first, the simultaneous measurements of the control signal of the compressor, the mass flow rate of the high-temperature side (m1_HP), and the mass flow rate of the low-temperature side (m2_HP). The second is the hot side's inlet temperature (T1_HP), and the third is the cold side's inlet temperature (T2_HP).

Figure 7 shows an example of the procedure that we follow. With the factors mentioned above, we consider two different levels consisting of the actual measurements of the sensor of the monitoring system that corresponds to that factor. In this case, we take weeks A and B as different levels of each factor. Following Taguchi's orthogonal array design, the matrix is designed with four simulations to be obtained from the model.



1	2	3
Control Signal $\dot{m}_1$ $\dot{m}_2$	T_1	T_2

Level	Week Name	Start Date	End Date
1	A	2017-11-01	2017-11-08
2	B	2017-10-23	2017-10-30

Run	Factors		
	1	2	3
1	1	1	1
2	1	2	2
3	2	1	2
4	2	2	1

Run	Factors		
	Control Signal $\dot{m}_1$ $\dot{m}_2$	T_1	T_2
1	A	A	A
2	A	B	B
3	B	A	B
4	B	B	A

Figure 7: Example with the procedure followed to obtain the time series with the heat pump model.

Within this work, we have made a set of simulations with these same 3 factors and with 10 different levels (10 weeks of measurements), resulting in 100 simulations, corresponding to 100 time series of duration one week each.

3.3.2. *Heat Exchanger*

Following the procedure described with the heat pump model, the methodology to obtain the time series from the heat exchanger model is similar to this one. In this case, it consists in maintaining as one factor in each simulation real values of simultaneous measurements of the 4 boundary conditions (inlet temperature and mass flow rate of both sides: m1_HX, m3_HX, T1_HX, T3_HX). The other factor in changing in each of the simulations is the efficiency (η) of the heat exchanger (see figure 8).

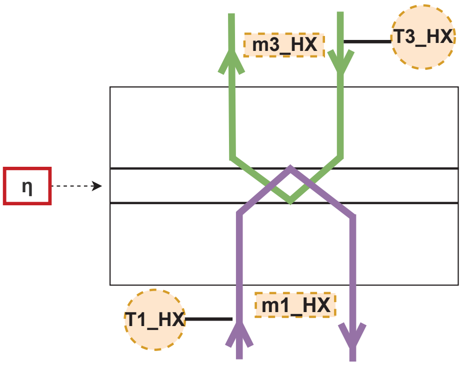


Figure 8: Factors that vary in the Heat Exchanger model simulations.

A Taguchi design is also used with 2 factors and 10 levels for these simulations, resulting in 100 simulations. The heat exchanger efficiency is assumed to have values between 0.5 and 0.95 with intervals of 0.05.

3.3.3. *Heat pump connected to the heat exchanger*

Apart from the cases of the isolated systems of the heat pump and the heat exchanger, a simulated case connecting the two isolated systems mentioned above is studied. For this purpose, we make several simplifications concerning the actual case.

The primary assumption is that the circuit outlet that exchanges heat with the heat pump's condenser is directly connected with the heat exchanger. Therefore, T3_HP and T3_HX are equal (and thus in figure 9 called directly T3_HP), as well as m1_HP and m3_HX (in scheme, m1_HP). In this simplification, the heat storage is omitted (as seen in figure 5). Furthermore, the water leaving the tank does not always enter the heat exchanger before going to the distribution system but depends on the regulation of the three-way valve. Therefore, in these simulations, the heat storage tank and the three-way valve are ignored, which play an important role in how these two systems are connected.

In this model, we have taken the data set of the simulated time series with the heat pump, and we have used them as input of the heat exchanger. Specifically, the simulated results of T3_HP and m1_HP have been used as substitutes for T3_HX and m3_HX in the heat exchanger. Thus, the

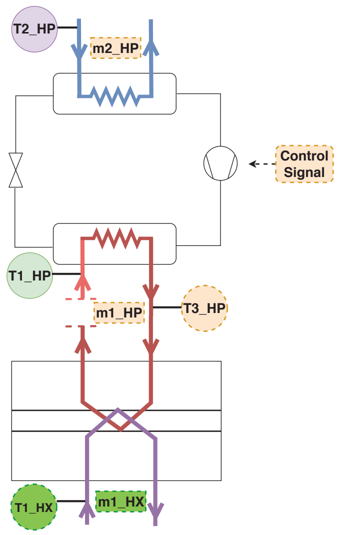


Figure 9: Factors changed in the case with the heat pump connected to the heat exchanger.

heat exchanger model has been simulated separately with these variables as input, taking for T1_HX and m1_HX the real simultaneous measurements of the previously considered weeks.

3.4. Toolchain for Machine learning

After generating generic data sets, we develop a toolchain for topology detection with supervised learning algorithms. Thus, following figure 10, we use the simulated and real time series and preprocess them. After that, we have three use cases in which we assign the corresponding classes for applying the algorithms. We can then use the supervised learning algorithms and compare the results in the considered use cases.

3.4.1. *Class assignation to the data sets for classification with Supervised Learning*

We establish three different cases with a focus on detecting the topology and potential connections between energy systems. The classification tasks have been carried out with real data streams and simulated data streams in all of them. This work also explores the case of training the algorithm with simulated time series, which we then validate with real time series.

It is possible to see the data streams that are considered connected and not connected in each case. In the case of the connected ones, we distinguish into *directly connected* if both data streams belong to the same hydraulic circuit or *indirectly connected* if they are from different loops but in the same system).

In particular, the studied cases are the following:

- **Case 1 - Connections in the HP and no connection between the isolated HP and HX**: *Direct connection* of two temperature sensors on the same side of the heat pump (T1_HP and T3_HP), *indirect connec-*

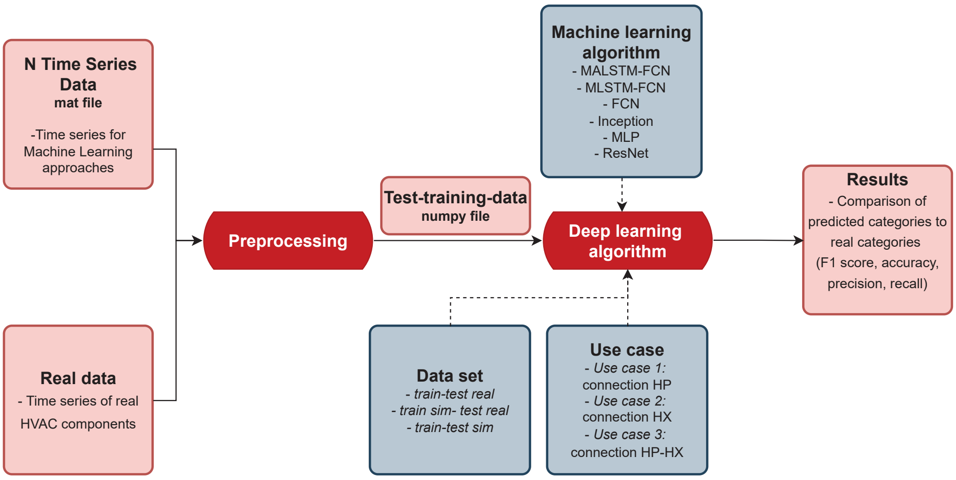


Figure 10: Process overview of the toolchain for supervised learning algorithms of topology detection.

tion of temperature sensors on different sides (high and low temperature) of the heat pump (T3_HP and T4_HP) and *no connection* between a sensor of the heat pump and a sensor of the heat exchanger (T3_HP and T2_HX). This is illustrated in figure 11.

- **Case 2 - Connection in the HX and no connection between the isolated HP and HX**: *Connection* of two temperature sensors of the heat exchanger (T2_HX and T4_HX) and *no connection* between a sensor of the heat pump and a sensor of the heat exchanger (T4_HX and T4_HP), as seen in figure 12.

- **Case 3 - Connection in the HP connected to the HX and no connection between the isolated HP and HX**: *Connection* of a temperature sensor of the heat pump and another of the heat exchanger when we have simulated them following the subsection 3.3.3 (T4_HP and T2_HX_*CON*) and *no connection* between a sensor of the heat pump and a sensor of the heat exchanger isolated one from each other (T4_HP and T2_HX). This is illustrated in figure 13.

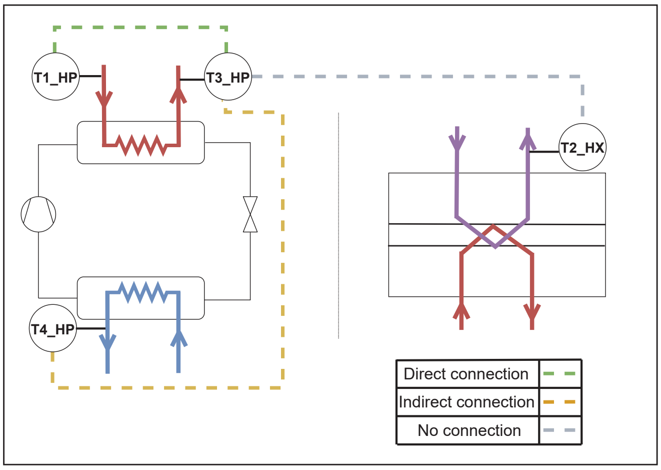


Figure 11: Case 1: Classification of data streams connected (direct and indirect) and not connected with a heat pump and a heat exchanger (isolated one from the other).

3.4.2. *Data preprocessing*

The described models and the approaches used for getting the data sets are all simulated with the automated simulation tool. As explained above, this tool uses as inputs the model in Modelica and an input matrix with the values to be changed in each simulation.

This procedure allows parameters and start values to be set before the simulation and the final values obtained at the end of the simulation. We have used the following settings in all simulations:

- Start time: 0 s, Stop time: 604800 s.
- Interval length: 300 s.
- Solver: Dassl. It is an implicit, higher order, multi-step solver with a step-size control. In particular, it is the default integration algorithm of Dymola [43].

In order to use the available data sets and implement the classification cases described above in the algorithm, preprocessing of the time series is required.

The real time series of data streams are downloaded from the database and divided into time series of one-week durations to have them in the same format as the simulated time series. Subsequently, those weeks that do not provide sufficient information in the classification tasks are eliminated, either because of errors or constant values in the measurements. The criterion adopted for deleting weekly time series is based on the dynamic standard deviation. Thus, in the case of the heat exchanger, we delete the weeks that have in any of the four temperature sensors a standard deviation less than 0.3 °C to ensure dynamics. With the heat pump, we adopt

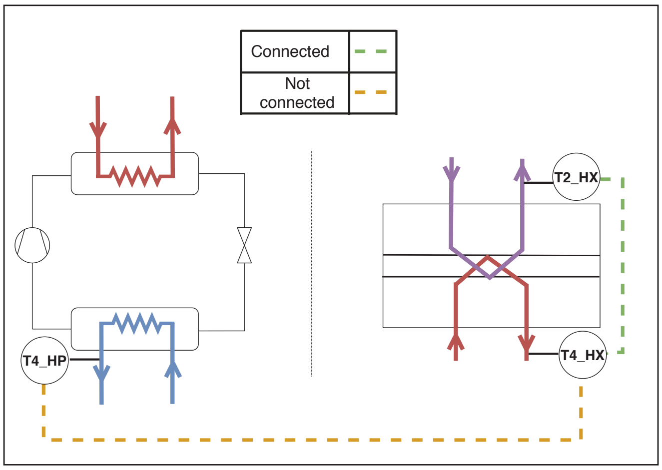


Figure 12: Case 2: Classification of data streams connected and not connected with a heat pump and a heat exchanger isolated one from each other.

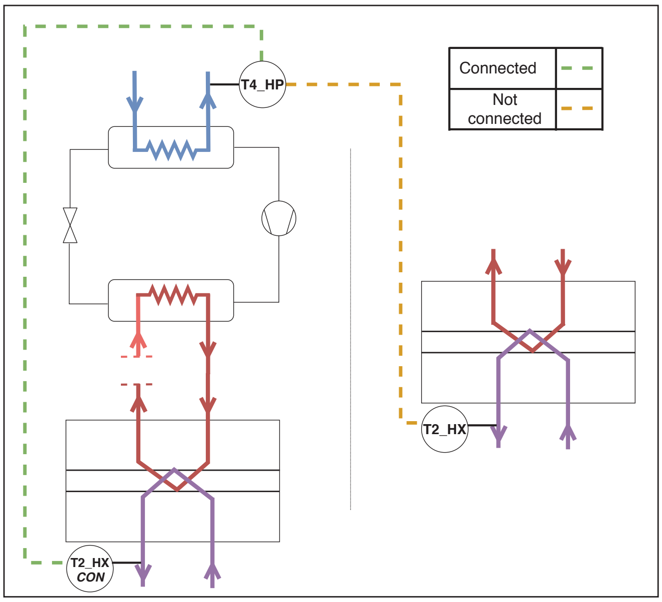


Figure 13: Case 3: Classification of data streams connected and not connected with a heat pump connected to a heat exchanger and an isolated heat exchanger.

a less restrictive criterion, where the standard deviation is limited to 0.5 °C.

Afterward, we process both real and simulated time series of data streams in the same way. First, subsets of the weeks are broken down to days, and then the measurements are resampled in steps of 5 min, resulting in time series of length 288. We do the resampling by applying the mean or backward or forward interpolation since the appropriate method is different depending on each time series.

After resampling the time series, we apply a Hampel filter to remove outliers [48]. It uses a sliding window of configurable width to go over the data. In this case, it is applied with a window size of 7 and a threshold of 3.

After these steps, the time series are packed in NumPy arrays [49]. Depending on the case to study, they are divided differently for training and testing. In the case in which the classification is made with simulated data, these belong to different simulation tests, so after assigning a label to each class, they are divided into 70 % for training and 30 % for testing and then we shuffle them. We do this with the help of the Scikit Learn library [50]. We proceed in the same way for the cases in which we use real measurements. Finally, in the cases in which time series from simulation and real measurements are mixed, only the simulated ones are used to train and the real ones to validate, being able to check in these cases if training the algorithm with simulated time series improves classification of the real measurements.

3.4.3. Deep learning algorithms

The toolchain implements six different algorithms based on Convolutional Neural Networks (CNN) for classification.

We use the implementation of Multivariate Long-Short-Term-Memory with Fully Convolutional Network Layer (MLSTM-FCN) provided by Karim et al. [31]. In a comparison of different deep learning approaches on the UCR data set (on univariate [32], and multivariate [31] time series classification), this implementation outperforms the approaches developed until 2018 the most. In addition, we use the algorithm with attention mechanism (MALSTM-FCN), which is supposed to enhance the performance, since in theory, it focuses on the essential parts of the time series [31]. The available adjustments of the algorithm are the number of epochs and the batch size. An epoch refers to all the training samples passing through the entire network each time. It is adjusted in all cases to ten since the time series sizes are not large, and a more significant number of time series instances is not needed to improve the results. The batch size refers to the number of samples needed to run before adjusting the neural network weights. The batch size is set to 128 for all cases, as it is the number recommended by the authors.

Furthermore, we selected four implementations of algorithms from a comparison presented by Ismail Fawaz et al. [34]. A total of nine deep learning algorithms are implemented in the approach Ismail Fawaz et al. [34]. Unfortunately, the other algorithms are not usable because, among other things, they did not deliver results for such short time series that we used in each case.

Wang et al. [51] propose deep multilayer perceptrons (MLP) for the classification of time series. The advantage is its simplicity. Its disadvantage is the required determination of the length of the time series. It contains a Fully Connected Network (FC), which does not consider the temporal dependencies because each timestamp is considered independently from the others [34]. They compared the MLP with a Residual Network (ResNet) implementation [51]. ResNet is the most complex layered approach in our comparison (11 layers). In this case, many layers mean a high training capacity and abstraction of the trained classes, which needs many training data.

The third approach used by Wang et al. [51] is a Fully Convolutional Network (FCN). Here the Fully Convolutional

Layer is used as a feature extractor. This layer offers the advantage of extracting individual sections of the time series as individual features.

Ismail Fawaz et al. [33] developed InceptionTime , which is inspired by the Inception-v4 architecture [52] (a ResNet variant) and should serve as an equivalent to AlexNet [53], which is a classic deep learning model for image classification. Its advantages are low dependence on training data and fast execution with consistent or better results. We call this algorithm in the next Inception.

4. Results

4.1. Comparison of the generated data-sets from the models developed with the toolchain

We compare the results of the simulated models with the actual measurements of these systems on the same dates and under the same conditions. To analyze the model's performance relative to reality, we use the Root Mean Square Error (RMSE) of one simulation week. Figure 14 presents an example of the results of the heat pump model with the outlet temperatures from both external loops of the heat pump (T3_HP and T4_HP). In this instance, the simulation results and the actual measurements are very similar (RMSE = 2.82 K for T4_HP and RMSE = 1.65 K for T3_HP), and we can see that the time series present the same tendency and behavior.

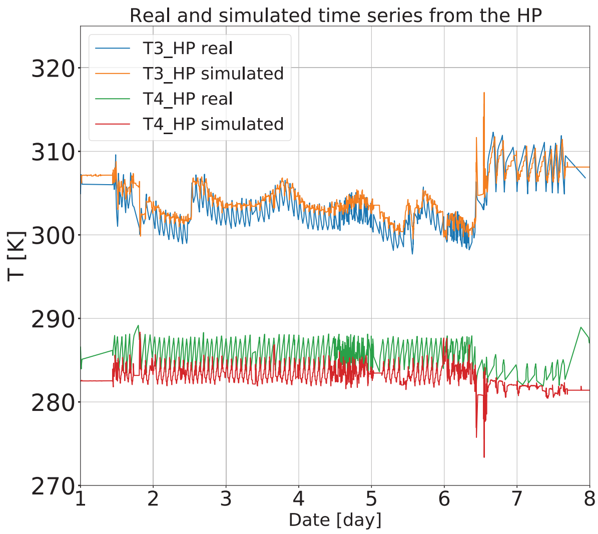


Figure 14: Comparison of the simulation results of the heat pump model and the real measurements of one week (T3_HP and T4_HP).

Regarding the heat exchanger model, figure 15 illustrates the results of the output temperatures of the high and low-temperature sides, comparing real and simulated time series. We execute these simulations according to the methodology explained in 3.3.2 with a heat exchanger efficiency of 0.8. It is shown that for T2_HX, the simulated model and the real measurements are in agreement, as the RMSE for a one-week simulation is 3.08 K. However, T4_HX differs significantly from the real case to the simulated one (RMSE = 13.48 K), seeming to indicate that the heat exchanger model with constant efficiency is not adequate in this case. Despite this, the behavior and trends of both time series are comparable (real and simulated) as the changes in dynamics are corrected, resulting in a convenient model to get the simulated time series for subsequently training the algorithms.

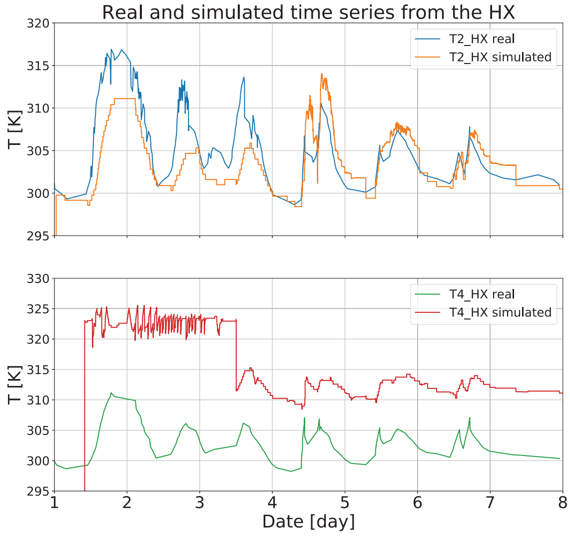


Figure 15: Comparison of the simulation results of the heat exchanger model and the real measurements of one week (T2_HX and T4_HX).

Figure 16 shows two further examples of simulations with the heat exchanger model, with results of T2_HX and T4_HX. With the time series in this figure, we evidence how different samples are obtained with the same model and how they are consistent with reality. Hence, this tool allows scalability when getting new data for succeeding applications.

Regarding the model with the heat pump connected to the heat exchanger, we show one week of the time series as an example in figure 17. It shows T4_HP and T2_HX, comparing the actual measurements with the simulated ones using the connected case of the same weeks as boundary conditions. We observe that the simulated temperatures are very similar to the real ones in both cases (T4_HP has RMSE = 2.78 K, T2_HX has RMSE = 4.54 K) and that the approach of connecting these systems ignoring certain real constraints that occur is valid. The difference between the modeled system and the real existing system is that the T3_HP (the output of the high-temperature side of the heat pump) goes directly to the heat exchanger (T3_HX). In the real system, there is a storage tank and valves that regulate its input. Nevertheless, figure 18 shows the real measurements of T3 in both systems. This example indicates how this temperature is practically the same in both systems. Thus, the assumption of directly using the heat pump's outlet to pass through the input of the

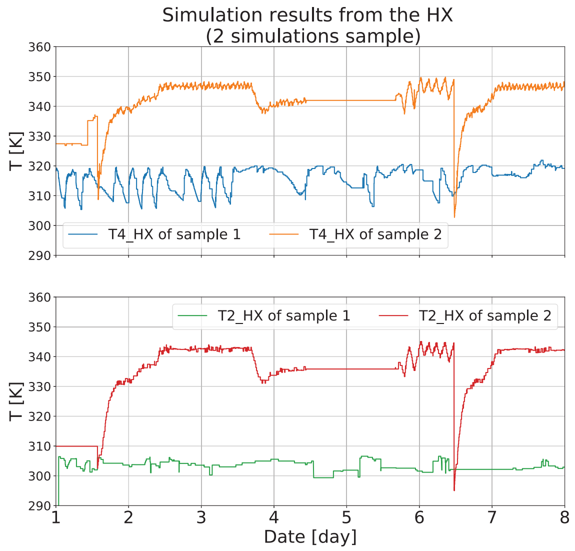


Figure 16: Simulated data streams of two samples of the heat exchanger (T2_HX and T4_HX).

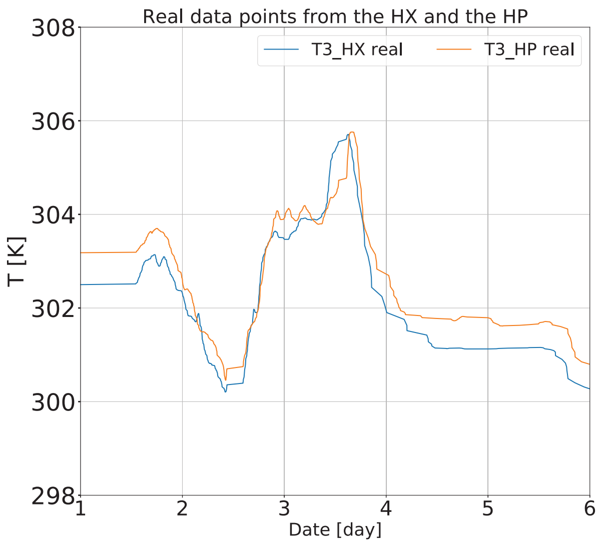


Figure 18: Real data streams of T3 in the heat pump and T3 in the heat exchanger.

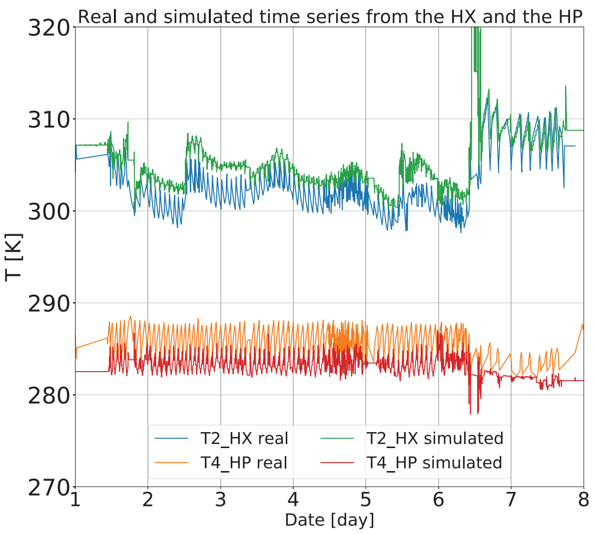


Figure 17: Comparison of the simulation results of the heat pump connected to the heat exchanger model and the real measurements of one week (T2_HX and T4_HP).

heat exchanger is justified.

4.2. Classification results

As described, the classification results are divided into three use cases. Each of the use cases represents a different building energy system. In each use case, a distinction was made between training and testing with real data (classic method), training with simulation data and testing with real data (our new method), and training and testing with simulation data (maximum achievable results).

4.2.1. Case 1: Connections in the HP and no connection between the isolated HP and HX

In case 1, the heat pump and the heat exchanger are isolated. It is apparent from the results (see table 1) that the F1 score increases to a value between 97.1 % and 97.9 % (Inception) in the cases with simulated data (maximum achievable results). However, testing and training with real data (classical method) show that the algorithm cannot detect the topology. F1 score is between 16.9 % and 18.8 % (FCN) and accuracy is between 33.9 % and 35.4 % (FCN) and only one class is identified.

Table 1
F1 score of all used algorithms in case 1 ("real" means that training and testing were done with real data, "sim" means that the input for training and testing were simulation data, "sim real" means that the input for training were simulation data and the input for testing were real data)

case	data set	MALSTM-FCN	MLSTM-FCN	FCN	Inception	MLP	ResNet
1	real	17.1	17.1	**18.8**	16.9	16.9	16.9
1	sim real	69.6	66	61.9	**70.5**	24.4	67.9
1	sim	97.6	97.1	97.4	**97.9**	62.6	97.7

By training and testing the algorithm with the simulation data set, the algorithms reached an F1 score above 97 % except for MLP. Nevertheless, training with simulation data set and testing with real data makes it possible to categorize the real data streams. As a result, the F1 score reaches a maximum value of 70.5 % (Inception) by testing with real data streams. In this case, the accuracy is 72.3 %. These results demonstrates the improved results from our chosen approach.

If we take a closer look at the results in the categories within the algorithms, we find that the results shown on the left side of figure 19 are typical for all used algorithms except MLP. For example, the algorithm identify both direct and indirect heat pump connections. However, by analyzing the confusion matrices, the non-connections (i.e. the heat pump data streams and the ones of the heat exchanger, both isolated from each other) are, in the best of cases, identified only 33.8 % of the time (Inception).

MLP does not achieve satisfactory results in any of the data sets. On the contrary, except for testing and training with real data, where it scores as poorly as the other algorithms, it shows strongly deviating results (differences in the F1 score of up to 46 %).

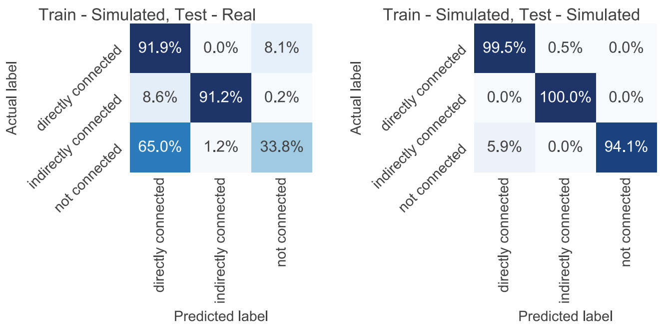


Figure 19: Accuracy of predicted classes of test data with the Inception algorithm in case 1 (left: trained with simulated and tested with real data, right: trained and tested with simulated data).

4.2.2. Case 2: Connection in the HX and no connection between the isolated HP and HX

The example of this case proposes recognizing the different topology of the connection inside a heat exchanger unlike two isolated systems (heat exchanger and heat pump). Therefore, there are two labels in this instance, and the same simulation tests as in case 1 are used.

Table 2 shows that the detection of each of these two classes occurs with an F1 score of 100 % with simulated data (FCN, Inception, ResNet). With the real time series of data streams, as with the rest of the cases, the classification does not work correctly because the algorithm correctly classified only one of the two labels. However, when training the algorithm with the simulated data, the real data is validated with a 94.9 % F1 score (with FCN). Thus, this algorithm produces a remarkable improvement with our method as almost the same results as training and testing only with simulated data are accomplished (maximum achievable results). Remarkably, the F1 score for the MALSTM-FCN when training with simulated data and testing with real data is 13.1 % lower than the comparable MLSTM-FCN (80.9 % vs. 94.0 %), which differs only in the attention mechanism. Figure 20 shows

Table 2
F1 score of all used algorithms in case 2 ("real" means that the input for training and testing were real data, "sim" means that the input for training and testing were simulation data, "sim real" means that the input for training was simulation data and the input for testing was with real data)

case	data set	MALSTM-FCN	MLSTM-FCN	FCN	Inception	MLP	ResNet
2	real	**34.2**	**34.2**	33.7	33.7	33.7	33.7
2	sim real	80.9	94	**94.9**	92.2	67.2	92.6
2	sim	99.8	99.8	**100**	**100**	87.4	**100**

the resulting confusion matrices with the best tests of this case. The actual connections were identified in the best result to a 100 % true positive rate. The algorithm identified non-existing compounds as connected to a 10.2 % false negative rate. The different results of the other algorithms (except MLP and MALSTM-FCN) are only due to the different results for the non-connections. Each of them has detected the connections to a 100 % true positive rate.

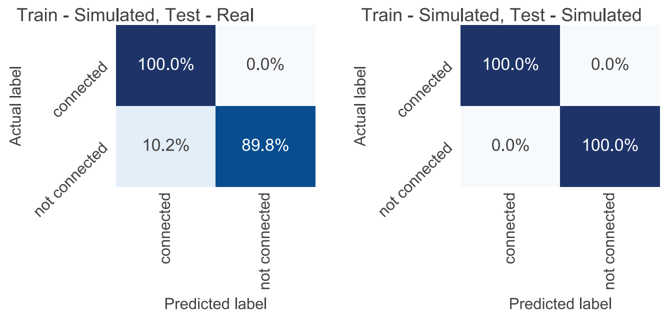


Figure 20: Accuracy of predicted classes of test data with the FCN algorithm in case 2 (left: trained with simulated and tested with real data, right: trained and tested with simulated data).

In this case (a heat exchanger), it is essential to consider which temperatures are connected and which are not because not all temperature sensors appear connected inside the heat exchanger equipment. Thus, observing the scheme of the systems (figure 5), it could be said that T1_HX and T2_HX are connected. However, this cannot be considered a connection in the classification tasks made in this work since T1_HX comes from the high-temperature systems and T2_HX reaches the temperature established in the heat balance, always being a few degrees higher than T4_HX. Therefore, we have considered that T2_HX, T3_HX, and T4_HX are connected, but T1_HX is not. Thus, the results of this case, with T2_HX and T4_HX as a class of connected time series, are 94.9 % accurate when training with the simulated data sets and validating with the real measurements. The results are the best obtained.

Data streams could be connected but are not connected in the classification based on piping and instrumentation diagrams. If experienced technicians manually check these, no connection can be detected either. Because of this fact, they are not connected in the classification tasks.

4.2.3. *Case 3: Connection in the HP connected to the HX and no connection between the isolated HP and HX*

The last case to analyze is where the heat pump connected to the heat exchanger is used to find the connection between these two different systems, comparing it with the detection of the non-connection of the heat pump and the heat exchanger separated. Accordingly, there are two labels in this case, namely for the connection and no connection classes.

Table 3
F1 score of all used algorithms in case 3 ("real" means that the input for training and testing were real data, "sim" means that the input for training and testing were simulation data, "sim real" means that the input for training was simulation data and the input for testing was done with real data)

case	data set	MALSTM-FCN	MLSTM-FCN	FCN	Inception	MLP	ResNet
3	real	**34.2**	**34.2**	33.7	33.7	33.7	33.7
3	sim real	55.7	52.0	36.3	**56.1**	34.4	50.5
3	sim	63.3	76.7	76.2	71.1	53.6	**89.7**

As indicated with the results in table 3, the F1 score achieved with simulated data streams is 89.7 % (with ResNet). Concerning the real data streams, the results agree with the previous cases in which the classification was not successful, and the algorithms classified only one of the classes correctly. Unlike in the previous cases, the improvement is not very big regarding the real data streams trained with the simulation data sets. In the best case (with Inception), it goes from an F1 score of 50.8 % training with real time series to 57.5 % training with the simulated data. FCN and MLP have similar values as when training with real data (˜ 33-36 % F1 score) and therefore do not generate usable information. Confusion matrices from these tests are shown in figure 21. In contrast to the other cases, the other algorithms differ significantly from the best algorithm in classifying the simulated data. These were recognized with an F1 score of 63.3 % to 76.7 %.

These results suggest that simulated time series may not be as similar to the real ones in this model compared to the other cases. The reason for this is that the data from the simulated heat exchanger takes two boundary conditions from the simulation results of a heat pump test and the other two boundary conditions as real measures of this system. However, the real measurements used as boundary conditions of the heat pump were not of the exact dates as those used in the heat exchanger simulations. Although the validation of the model is successful, there are certain discrepancies. Developing a model that connects both systems and considers a test with the conditions of both taken simultaneously could solve this.

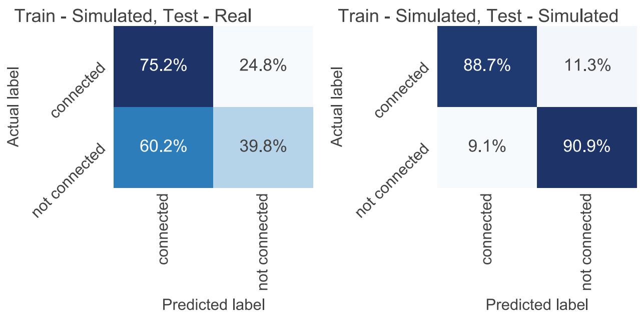


Figure 21: Accuracy of predicted classes of case 3 with the Inception algorithm (training - simulated data, testing - real data) and ResNet (training, testing - simulated).

4.2.4. *Overall results*

Table 4 sums up the results (average F1 score and rank) of our algorithm comparison. The results are overall mixed. Due to the small number of data sets, the explanatory power of mean rank is difficult to determine. If we consider only the mean rank, the results of the M(A)LSTM-FCN algorithms are generally better. However, they achieved good ranks mainly in training and testing with real data. They achieved higher F1 scores and accuracy than the other algorithms but marginally better and did not produce valuable results. ResNet achieved the highest average F1 score over all cases and data sets.

None of the algorithms show usable results when tested and trained with real data. Sometimes they are worse than a randomized selection of categories (<50 % F1 score for two categories). The simulated data provided the best results across all algorithms and data sets. Here, values of up to 100 % are achieved depending on the use case. Only use case 3 achieved a maximum F1 score of 89.7 %.

This approach aims to train with simulation data and testing with real data. Here the results are to be judged differently. In use case 2, the F1 score reached a value of 94.9 %. In contrast, use case 3 is only marginally above the random selection (F1 score 56.1 %). Use case 1 is not recognized with an F1 score that is useful for direct use (maximum 70.5 %), which, however, is significantly better than in use case 3.

MLP does not achieve usable results as an algorithm. Here, the missing convolutional layer and the lack of consideration of continuous time series becomes visible.

5. Discussion

5.1. Generating generic data sets

The difference between an algorithm trained with real data and an algorithm trained with data from simulation mod-

Table 4
Mean F1 score and rank of all used algorithms and all used data sets (real and simulated)

	MALSTM-FCN	MLSTM-FCN	FCN	Inception	MLP	ResNet
all data sets from all cases:						
mean rank	3.2	**3.1**	3.8	3.6	6.0	3.8
mean F1	61.4	63.5	61.4	63.6	46.0	**64.7**
only trained with sim and tested with real (sim real):						
mean rank	3.0	3.0	3.7	**2.0**	6.0	3.3
mean F1	68.7	70.7	64.3	**73.0**	42.0	70.3

els is significant (22 to 60 %), depending on the use case. Nevertheless, the results show that the idea of the generation of time series data using simulation models works. Furthermore, algorithms trained with simulated data achieved better results than those trained with real data in all considered cases and algorithms.

Especially use case 2, with an F1 score of 94.9 % and an accuracy of 94 %, shows the potential of our time series generation method. Not every use case reached these promising results, but training with simulated time series achieved better results in all of them. Since we considered only three use cases, a general assertion is difficult to derive.

Remarkably, the real data has successfully trained and validated the algorithm in none of the cases. Some of the reasons that may explain this outcome are errors in the real measurements and constant measurements on many occasions, which do not provide information to the algorithm. Another possible factor is that in most cases, the number of samples used in tests with real data has been lower than in cases with simulated data (about 20 % more samples with simulated than with real data).

The generation of results strongly depends on the quality of the simulation models as the models must represent the correct dynamics. This circumstance limits the general applicability of the approach. However, we can see from the exemplary simulation results that even unvalidated simulation models generate time series similar to those found in existing systems. For the training of algorithms, the simulated time series have the advantage that they can represent several energy systems with different scalings (e.g., the different heating power of a boiler or a heat pump). In theory, this enables the algorithm to learn and abstract the typical physical behavior of different energy systems. The difference between training with simulated data and real data may indicate that this theoretical goal is partially achievable.

In this approach, we have considered three different connections of only two different systems of technical building equipment. This circumstance limits the statement about the general application of our approach.

5.2. Algorithms

The results of the multivariate time series classification algorithms provide no general statement. However, the Inception algorithm was the best in two of the three use cases for training with simulated data and testing with real data. It achieved only slightly different results in the third case. Therefore it can be recommended here. However, it has to be checked with other topology connections in the building if good results are achievable here.

When training and testing with real data, all algorithms fail. Thus, using real data cannot be recommended. Here, it is also questionable whether changes in pre-processing, other technical connections, or other algorithms can achieve an improvement. The characteristics of the simulated time series, such as a permanent deviation of the temperature or no disturbances, suggest that classical classification methods like random forest do not obtain the necessary information for classification.

The poor results of MLP show that for topology detection in building energy systems, due to the high dead times (flow through the building and heat transfer), the time dependencies must be considered. However, the other algorithms generally achieve this with significantly better overall results.

5.3. Overall process

Use case 2 with an F1 score of 94.9 % when training with simulated data and testing with simulated data shows that our approach of supervised topology detection with generalized generated data works. However, the other use cases also show the limitations of the current methodology. For example, comparing different algorithms is challenging due to the lack of supervised algorithms and data sets for topology detection in building energy systems.

With unsupervised methods of topology detection, accuracies of >90 % have already been achieved [54]. Since the systems are very different (hydraulic system versus air handling unit), a comparison of the results is questionable. The results in [30] with a maximal accuracy of 52.1 %, which used the same energy system in the same building, but without the same focus on technical equipment, are the most comparable results. Especially the lack of reaching steady states was a problem in detecting connections. In our supervised algorithm, this is not a requirement. The comparable results show that our approach delivers equivalent or better results.

6. Conclusion

The results show that it is possible to detect connections in building energy systems (BES) based on supervised learning trained using generalized time series from grey-box simulation models. Nevertheless, some research still needs to be done to apply this approach in a scalable way. Furthermore, a comparison of this approach against other connection cases between technical building equipment types is needed.

The used algorithms based on CNN showed especially in use case 2 results that were above 90 % F1 score. The inception algorithm was the best algorithm on average with our method. It is questionable whether classical machine learning algorithms also benefit from the approach developed here.

The comparison of training using simulated and real data shows that simulated data can be an alternative to real data in identifying connections in energy systems when not enough data is available. Nevertheless, publicly available building energy system data containing the topology data is rare. So, the generation of generalized data covering a more comprehensive range of technologies than available real data sets is required. For this case, the presented method has high potential.

The physical simulation models reflect the energy systems without disturbances which usually occur in existing systems. However, this can be a disadvantage, especially for the application in machine learning algorithms. These algorithms are then not necessarily robust. Here, integrating disturbances into the physical model could help represent the actual operation more robustly.

We could transfer our results to thermal systems with higher thermal inertia, such as underfloor heating or concrete core activation. Here, other algorithms may be required, especially to cope with the high dead time.

Whether the supervised topology detection would also work with different systems still needs to be researched. The results indicate that a generalized application of connection detection is complex. If the connections can be clearly defined and no neighboring systems produce disturbances, the approach shown here can provide suitable results.

CRediT authorship contribution statement

Florian Stinner: Conceptualization of this study, Methodology, Data curation, Production of results, Writing. **Belén Llopis-Mengual:** Methodology, Investigation, Data curation, Writing - Original Draft. **Thomas Storek:** Writing - Review & Editing. **Alexander Kümpel:** Writing - Review & Editing, Support. **Dirk Müller:** Writing - Review & Editing, Support.

Acknowledgments

We gratefully acknowledge the financial support provided by the BMWi (Federal Ministry for Economic Affairs and Energy) with promotional reference 03SBE006A. B. Llopis-Mengual acknowledges the Ministry of Universities of Spain through the "Formación de Profesorado Universitario" programme ref. FPU 19/04012.

References

[1] Nordhaus, W.. Climate change: The ultimate challenge for economics. American Economic Review 2019;109(6):1991–2014. doi:10.1257/aer.109.6.1991.

[2] European Commission, . Directive (EU) 2018/844 of the European Parliament and of the Council of 30 may 2018 amending directive 2010/31/EU on the energy performance of buildings and directive 2012/27/EU on energy efficiency. 2018. URL: https://eur-lex.europa.eu/eli/dir/2018/844/oj; , Accessed date: August 2021.

[3] DAgostino, D., Zangheri, P., Castellazzi, L.. Towards nearly zero energy buildings in Europe: A focus on retrofit in non-residential buildings. Energies 2017;10(1):117. doi:10.3390/en10010117.

[4] International Energy Agency, . Transition to Sustainable Buildings: Strategies and Opportunities to 2050. OECD; 2013. ISBN 9789264202412. doi:10.1787/9789264202955-en.

[5] Wang, Z., Hong, T.. Reinforcement learning for building controls: The opportunities and challenges. Applied Energy 2020;269:115036. doi:10.1016/j.apenergy.2020.115036.

[6] Hazyuk, I., Ghiaus, C., Penhouet, D.. Model predictive control of thermal comfort as a benchmark for controller performance. Automation in Construction 2014;43:98–109. doi:10.1016/j.autcon.2014.03.016.

[7] Yang, S., Wan, M.P., Chen, W., Ng, B.F., Dubey, S.. Model predictive control with adaptive machine-learning-based model for building energy efficiency and comfort optimization. Applied Energy 2020;271:115147. doi:10.1016/j.apenergy.2020.115147.

[8] Klein, L., Kwak, J.y., Kavulya, G., Jazizadeh, F., Becerik-Gerber, B., Varakantham, P., et al. Coordinating occupant behavior for building energy and comfort management using multi-agent systems. Automation in Construction 2012;22:525–536. doi:10.1016/j.autcon.2011.11.012; Planning Future Cities - Selected papers from the 2010 eCAADe Conference.

[9] Pang, Z., Chen, Y., Zhang, J., O'Neill, Z., Cheng, H., Dong, B.. Nationwide HVAC energy-saving potential quantification for office buildings with occupant-centric controls in various climates. Applied Energy 2020;279:115727. doi:10.1016/j.apenergy.2020.115727.

[10] Pan, Y., Zhang, L.. Roles of artificial intelligence in construction engineering and management: A critical review and future trends. Automation in Construction 2021;122:103517. doi:10.1016/j.autcon.2020.103517.

[11] Hong, T., Wang, Z., Luo, X., Zhang, W.. State-of-the-art on research and applications of machine learning in the building life cycle. Energy and Buildings 2020;212:109831. doi:10.1016/j.enbuild.2020.109831.

[12] 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. ACM; 2015, p. 33–34. doi:10.1145/2821650.2821669.

[13] Balaji, B., Bhattacharya, A., Fierro, G., Gao, J., Gluck, J., Hong, D., et al. Brick : Metadata schema for portable smart building applications. Applied Energy 2018;226:1273–1292. doi:10.1016/j.apenergy.2018.02.091.

[14] Prívara, S., Cigler, J., Váňa, Z., Oldewurtel, F., Sagerschnig, C., Žáčeková, E.. Building modeling as a crucial part for building predictive control. Energy and Buildings 2013;56:8–22. doi:10.1016/j.enbuild.2012.10.024.

[15] 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. Automation in Construction 2018;85:107–123. doi:10.1016/j.autcon.2017.09.013.

[16] Stinner, F., Wiecek, M., Kümpel, A., Baranski, M., Müller, D.. Automatic digital twin data model generation of building energy systems from piping and instrumentation diagrams. In: Proceedings of ECOS 2021 - The 34th International Conference On Efficiency, Cost, Optimization, Simulation and Environmental Impact of Energy Systems : June 27 - July 2, 2021, Taormina, Italy. 2021,doi:10.48550/arXiv.2108.13912; , Accessed date: August 2021.

[17] Gao, H., Koch, C., Wu, Y.. Building information modelling based building energy modelling: A review. Applied Energy 2019;238:320–343. doi:10.1016/j.apenergy.2019.01.032.

[18] Lange, H., Johansen, A., Kjærgaard, M.B.. Evaluation of the opportunities and limitations of using IFC models as source of building metadata. In: Ramachandran, G.S., Batra, N., editors. 5th ACM International Conference on Systems for Energy-Efficient Built Environments. The Association for Computing Machinery. ISBN

9781450359511; 2018, p. 21–24. doi:10.1145/3276774.3276790.

[19] Stinner, F., Yang, Y., Schreiber, T., Bode, G., Baranski, M., Müller, D.. Generating generic data sets for machine learning applications in building services using standardized time series data. In: Al-Hussein, M., editor. Proceedings of the 36th International Symposium on Automation and Robotics in Construction (ISARC). Proceedings of the International Symposium on Automation and Robotics in Construction (IAARC); International Association for Automation and Robotics in Construction (IAARC); 2019,doi:10.22260/ISARC2019/0031.

[20] Miller, C.. More buildings make more generalizable modelsbenchmarking prediction methods on open electrical meter data. Machine Learning and Knowledge Extraction 2019;1(3):974–993. doi:10.3390/make1030056.

[21] Huchtkoetter, J., Reinhardt, A.. A study on the impact of data sampling rates on load signature event detection. Energy Informatics 2019;2(1):24. doi:10.1186/s42162-019-0096-9.

[22] Kazmi, H., Munné-Collado, Í., Mehmood, F., Syed, T.A., Driesen, J.. Towards data-driven energy communities: A review of open-source datasets, models and tools. Renewable and Sustainable Energy Reviews 2021;148:111290. doi:10.1016/j.rser.2021.111290.

[23] Fierro, G., Pritoni, M., Abdelbaky, M., Lengyel, D., Leyden, J., Prakash, A., et al. Mortar: An open testbed for portable building analytics. ACM Transactions on Sensor Networks 2019;16(1):1–31. doi:10.1145/3366375.

[24] Zhou, X.. A plug and play framework for an HVAC air handling unit and temperature sensor auto recognition technique. Ph.D. thesis; Iowa State University, Mechanical Engineering, PHD (Doctor of Philosophy), 2010; Ames, Iowa; 2010. doi:10.31274/etd-180810-1952.

[25] Pritoni, M., Bhattacharya, A.A., Culler, D., Modera, M.. Short paper: A method for discovering functional relationships between air handling units and variable-air-volume boxes from sensor data. In: Culler, D., Agarwal, Y., Mangharam, R., editors. Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments. ACM. ISBN 9781450339810; 2015, p. 133–136. doi:10.1145/2821650.2821677.

[26] 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. URL: http://arxiv.org/abs/1601.07260. arXiv:1601.07260.

[27] Fürst, J., Chen, K., Katz, R.H., Bonnet, P.. Crowd-sourced bms point matching and metadata maintenance with Babel. In: 2016 IEEE International Conference on Pervasive Computing and Communication workshops. IEEE. ISBN 978-1-5090-1941-0; 2016, p. 1–6. doi:10.1109/PERCOMW.2016.7457102.

[28] Hong, D.. Towards automatic context inference for sensors in commercial buildings. Ph.D. thesis; University of Virginia, Computer Science - School of Engineering and Applied Science, PHD (Doctor of Philosophy), 2018; 2018. doi:10.18130/V3-9W7H-4186.

[29] Li, S., Hong, D., Wang, H.. Relation inference among sensor time series in smart buildings with metric learning. Proceedings of the AAAI Conference on Artificial Intelligence 2020;34(04):4683–4690. doi:10.1609/aaai.v34i04.5900.

[30] Stinner, F., Raßpe-Lange, L., Baranski, M., Müller, D.. Takeshi: Application of unsupervised machine learning techniques for topology detection in building energy systems. Journal of Physics: Conference Series 2019;1343:012041. doi:10.1088/1742-6596/1343/1/012041.

[31] Karim, F., Majumdar, S., Darabi, H., Harford, S.. Multivariate LSTM-FCNs for time series classification. Neural Networks 2019;116:237–245. doi:10.1016/j.neunet.2019.04.014.

[32] Karim, F., Majumdar, S., Darabi, H.. Insights into LSTM fully convolutional networks for time series classification. IEEE Access 2019;7:67718–67725. doi:10.1109/ACCESS.2019.2916828.

[33] Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D.F., Weber, J., et al. InceptionTime: Finding AlexNet for time series classification. Data Mining and Knowledge Discovery 2020;34(6):19361962. doi:10.1007/s10618-020-00710-y.

[34] Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.. Deep learning for time series classification: a review. Data Mining and Knowledge Discovery 2019;33(4):917–963. doi:10.1007/s10618-019-00619-1.

[35] Ruiz, A.P., Flynn, M., Large, J., Middlehurst, M., Bagnall, A.. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Mining and Knowledge Discovery 2021;35(2):401–449. doi:10.1007/s10618-020-00727-3.

[36] Remmen, P., Lauster, M., Mans, M., Fuchs, M., Osterhage, T., Müller, D.. TEASER: an open tool for urban energy modelling of building stocks. Journal of Building Performance Simulation 2018;11(1):84–98. doi:10.1080/19401493.2017.1283539.

[37] Hong, T., Macumber, D., Li, H., Fleming, K., Wang, Z.. Generation and representation of synthetic smart meter data. Building Simulation 2020;13(6):1205–1220. doi:10.1007/s12273-020-0661-y.

[38] Granderson, J., Lin, G., Harding, A., Im, P., Chen, Y.. Building fault detection data to aid diagnostic algorithm creation and performance testing. Scientific Data 2020;7(1):65. doi:10.1038/s41597-020-0398-6.

[39] [dataset], Stinner, F. , Llopis, B., Kümpel, A., Müller, D., . Deep learning supervised topology detection. 2021. URL: https://github.com/RWTH-EBC/Deep-learning-supervised-topology-detection; [Github repository], Accessed date: August 2021.

[40] 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. REHVA Journal; 2018, p. 10–15. URL: https://www.rehva.eu/fileadmin/user_upload/10-15_RJ1804_WEB.pdf; , Accessed date: August 2021.

[41] Modelica Association, . Modelica - a unified object-oriented language for systems modeling: Language specification - version 3.4. 2017. URL: https://modelica.org/documents/ModelicaSpec34.pdf; , Accessed date: September 2020.

[42] Prud'hommeaux, E., Seaborne, A.. SPARQL query language for RDF. 2007. URL: http://www.w3.org/TR/rdf-sparql-query/; , Accessed date: August 2021.

[43] Dassault Systemes, . Dymola systems engineering. 2020. URL: https://www.3ds.com/products-services/catia/products/dymola/; , Accessed date: September 2020.

[44] Digital Bazaar, Inc., . Pyld. 2020. URL: https://github.com/digitalbazaar/pyld; , Accessed date: September 2020.

[45] Fütterer, J., Constantin, A., Schmidt, M., Streblow, R., Müller, D., Kosmatopoulos, E.. A multifunctional demonstration bench for advanced control research in buildings: Monitoring, control, and interface system. In: IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society. 2013, p. 5696–5701. doi:10.1109/IECON.2013.6700068.

[46] Müller, D., Lauster, M., Constantin, A., Fuchs, M., Remmen, P.. AixLib - an open-source Modelica library within the IEA-EBC Annex 60 framework. In: Proceedings of the CESBP Central European Symposium on Building Physics and BauSIM 2016, Dresden , Germany. Stuttgart: Fraunhofer IRB Verlag. ISBN 978-3-8167-9798-2; September 2016, p. 3–9. URL: https://publications.rwth-aachen.de/record/681852; , Accessed date: September 2020.

[47] Taguchi, G., Chowdhury, S., Wu, Y.. Quality Engineering: The Taguchi Method; chap. 4. John Wiley & Sons, Ltd. ISBN 9780470258354; 2004, p. 56–123. doi:10.1002/9780470258354.ch4.

[48] Pearson, R.K., Neuvo, Y., Astola, J., Gabbouj, M.. Generalized hampel filters. EURASIP Journal on Advances in Signal Processing 2016;2016(1):87. doi:10.1186/s13634-016-0383-6.

[49] Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., et al. Array programming with NumPy. Nature 2020;585(7825):357–362. doi:10.1038/s41586-020-2649-2.

[50] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 2011;12:2825–2830. URL: http://arxiv.org/abs/1201.0490. arXiv:1201.0490; , Accessed date: September 2020.

[51] Wang, Z., Yan, W., Oates, T.. Time series classification from scratch with deep neural networks: A strong baseline. In: 2017 Inter-

national Joint Conference on Neural Networks. IEEE. ISBN 978-1-5090-6182-2; 2017, p. 1578–1585. doi:10.1109/IJCNN.2017.7966039.

[52] Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.. Inception-v4, Inception-ResNet and the impact of residual connections on learning. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence; vol. 31 of *AAAI'17*. AAAI Press; 2017, p. 4278–4284. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14806/14311; , Accessed date: August 2021.

[53] Krizhevsky, A., Sutskever, I., Hinton, G.E.. Imagenet classification with deep convolutional neural networks. Communications of the ACM 2017;60(6):8490. doi:10.1145/3065386.

[54] Hong, D., Cai, R., Wang, H., Whitehouse, K.. Learning from correlated events for equipment relation inference in buildings. In: Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation. ACM. ISBN 9781450370059; 2019, p. 203–212. doi:10.1145/3360322.3360852.