

Highlights

Deep learning supervised topology detection of building energy systems by generated time series of generic grey-box models

Florian Stinner, Belén Llopis, Alexander Kümpel, Dirk Müller

- Automatic generation of grey-box models based on historical data of building energy systems.
- Production of simulated data sets for implementation in smart building applications.
- Use of supervised learning approaches for topology detection (relation inference) in building energy systems.

Deep learning supervised topology detection of building energy systems by generated time series of generic grey-box models

Florian Stinner^{a,**} (Co-ordinator), Belén Llopis^{a,b,*} (Researcher), Alexander Kümpel^{a,*} and Dirk Müller^a

^aRWTH Aachen University, E.ON Energy Research Center, Institute for Energy Efficient Buildings and Indoor Climate, Mathieustrasse 10, 52074 Aachen, Germany

^bInstituto 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

Building Automation Systems are essential to reduce energy consumption in buildings. Thus, automatic control systems must have detailed information, such as the relation inference among sensors. This requires real data, which is rarely available in the needed quality for automatic assignment. This study is based on real data from eight temperature sensors of a heat pump and a heat exchanger, collected at a frequency of 5 minutes over 200 weeks. We use this data to develop a methodology that generates grey-box models and extend the data set with 500 simulated weeks. We train six supervised deep learning algorithms with all the data to test whether detecting connections is possible. Our method results in a 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 [32]. Including indirect emissions, buildings represent 36 % of European CO₂ emissions [8]. In existing buildings, there is an increased need for CO₂ reduction, which cannot be met by a current renovation rate of 1 % [5], as 3 % would be required [8]. Therefore, it is necessary to implement automated measures to reduce emissions.

Especially, non-residential buildings are provided with complex building automation systems (BAS). Approximately of the total energy consumption in non-residential buildings equipped with BAS can be avoided only by improved control [20]. Modern control systems can include reinforcement learning [51], model predictive control [14, 53] or occupant-centric control systems [26, 35]. Machine learning can be applied in every life cycle area of a building energy system [34, 18].

Although each 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) can be found several times 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.

Graph-based models based on ontologies can represent this information. Since no current building ontology represents building operation well [3], the Brick Schema [2] was 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*.


Thus, the data streams in BAS have labels that differ depending on the building and operator [3] [50]. The information from these labels is usually very labor-intensive to extract [50]. 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.

The 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 [39, 14], the exact mapping of data streams and the topology is essential. However, this mapping is often not available in a directly analyzable form. For this purpose, the use of time series is a promising scalable approach.

BAS provide knowledge on the operating states of BES over time and therefore information about the BES itself. In addition, BAS contain raw data about the topology of technical building equipment (TBE) and thus the (energy) flows in the BES. This knowledge can also be obtained from piping and instrumentation diagrams. However, a correlation of time series data with the diagram data is difficult due to the lack of standardized sensor and actuator descriptions on the

*Corresponding author

**Principal corresponding author

 fstinner@eonerc.rwth-aachen.de (F. Stinner);

belen.llopis@iie.upv.es (B. Llopis); akuempel@eonerc.rwth-aachen.de (A. Kümpel)

 https://www.researchgate.net/profile/Florian_Stinner (F. Stinner)

Stinner)

ORCID(s): 0000-0002-6638-4276 (F. Stinner); 0000-0003-0238-4231 (B. Llopis)

diagrams in BES [45]. Building information models (BIM) could also provide this information [12]. However, their application in existing buildings is not yet very high and information about the operation of BES are rarely encountered [28]. Therefore, alternative methods must be developed.

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 the flow direction reacts to this switching operation. For example, the value of a temperature sensor in the output of a boiler increases when the boiler is switched on. This phenomenon can be used for the topology detection (TD) of BES (interconnection of all TBE in a 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 [46] [4]. In particular, rarer connection types are 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 [12]. In [46], 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 goal of the approach presented in this work is to learn the relation inference (topology) of a multi-functional office building, for which real monitored data is available for 4 water temperature data streams from a heat pump and 4 from a heat exchanger, collected at a frequency of 5 minutes for 200 weeks. 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 increase the data set, and 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: first, the background and related work is presented and then, the methods used are discussed. In the method, we first describe the toolchain for the creation of building energy system models. Here, the generation of generalized time series using the models is explained. Based on this, the developed real use cases for the supervised detection of building energy systems are presented. Afterward, we introduce the toolchain for the application of the six used deep learning multivariate time series classification algorithms. The results are presented with three different use cases. The results are classified and discussed and it is explained how they can be further developed and used.

2. Background and related work

Events are very common in electrical networks compared to thermal networks. Thus, Huchtkoetter and Reinhardt [19] 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 common. Topology detection is better researched in the power grid topic due to the higher number of measured values of an electrical measurement system. Topology detection is also described in the literature as relation inference for building energy systems. In electrical grids, the term "topology detection" is more common. Since topology detection in electrical grids is the most researched one, this term is adopted here.

The following definition is used for the term data stream: a data stream is an information carrier that continuously provides information about a state [46]. Supervised learning is used to determine data stream types in building automation systems. Three different types of inputs must be distinguished: time series of data streams, their features and their labels. Furthermore, mixed versions of the inputs exist [50].

Wang et al. [50] give an overview of different methods for automatic data stream mapping in building automation systems. One problem in comparing different approaches to topology detection as well as metadata extraction is the lack of comparability between the approaches. Different connection types (topology detection) or different data stream types (meta data extraction) are used. The use of standardized data sets or the publication of the test and training data is very helpful here. We analyzed various publicly available data sets of building automation systems to see if they had suitable time series for our approach, but 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 in our approach.

Kazmi et al. [25] has analyzed the different 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 [9] is the most promising data set with time series data from 107 buildings. But it also hardly contains any data on water side 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 use in 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 matching time series or data sets publicly available. None of the papers on topology detection provided the data sets used in such a way that they were directly usable for the use cases used here. Either parts of the data set were missing, as can be assumed from the source codes, the time series data itself were not included, or the used connection types did not correspond to the connection types used here.

Current and past research focuses mainly on the detec-

tion of data stream types based on the associated time series and their labels. The detection of connections between two or more components in building energy systems is rarely researched. In the following, related research results known to the authors are listed.

Zhou [54] 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 difficult to detect.

Pritoni et al. [38] and Koh et al. [27] use active control of the TBE to detect further connected TBE. Active control can rarely be used in buildings during normal operating hours. Otherwise, the comfort in rooms is compromised.

Fürst et al. [10] 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 corresponding costs.

Hong [15] has closed the toolchain from the detection of data stream types to the detection of the topology of an energy system. For the detection of the connection, it uses an unsupervised procedure that first generates a Markov event model. Based on this model, transitions are recognized that correspond to events. These events are filtered so that only those events remain that are unique between the systems. However, only the connection from air handling units (AHU) to variable air volume systems (VAV) was detected here. This corresponds only to the air side of the connections within the building and disregards the water-led supply.

This approach has been implemented by 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, they only deal with the supply of air in the building. The water side of the building is not addressed.

As far as the authors know, a supervised learning approach has not yet been implemented for the water-based heating and cooling system of building energy systems except for an approach of Stinner et al. [44], where only a maximum accuracy of 52.1 % was reached.

Supervised learning is established in the identification of types of data streams in buildings [50]. The labels and data streams themselves were used as input. Supervised learning is a good method for the classification of data streams types, achieving over 90 % accuracy [50]. In a typical building, only a limited number of technical systems are connected to each other. In thermal systems, the temperature is a signal that reacts strongly to changes in the previous system and therefore its temperature signal. This complies with multivariate time series.

This work aims to show that special connections in BES can be detected by supervised learning of the multivariate temperature signals. For supervised learning of multivariate sensor data, we use six classifiers that were developed for multivariate classification of time series [24, 23, 22, 21]. We

use convolutional neural networks (CNN), which performed in the top group in time series classification on the UCR time series archive[42]. CNN require more time series than classical methods (e.g. random forest) for correct classification. However, they offer the potential that they can classify in a generalized manner[42].

A problem with the application of CNN-based supervised learning in BES is the lack of availability 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. Synthetic data based on grey-box models are mainly used to evaluate BES in connection with their usage [41, 17]. Nevertheless, the use of simulation data to feed machine learning algorithms is an option used especially in fault detection [13].

Stinner et al. [46] developed a toolchain for generating generic data streams for detecting real data stream types. However, the application is limited to one heat pump. Here, this toolchain was further developed and extended for scalable use. For the generation of generalized data, a Design of Experiment approach was used, which is currently used for the identification and validation of parameters in simulation models. Generalized data have the advantage that not only the connection of a certain technical system can be detected, but a wider range of systems as well.

To the authors' knowledge, there is currently no method for detecting topology relationships in water-side BES neither using data generated from grey-box models nor comparing different supervised learning approaches.

3. Methodology

3.1. Total toolchain

Our toolchain consists of two parts: in subsection 3.2, the process of generating the generic time series data using grey-box simulation models is described. The second part consists of the preprocessing of the time series and the used supervised machine learning algorithms, which is introduced in subsection 3.5. The entire process chain is illustrated in figure 1 with the required inputs. The program code (classifier and toolchain) and the used data sets are stored separately in the repository [47] and can be used with the MIT license.

3.2. 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) [43] and transforms it to Brick schema (a metadata scheme to describe buildings [2]), in order to obtain a model in Modelica of the real system [30]. Therefore, this can be simulated for obtaining time series for instance machine learning applications.

Following the scheme in figure 2, the tool performs the following steps:

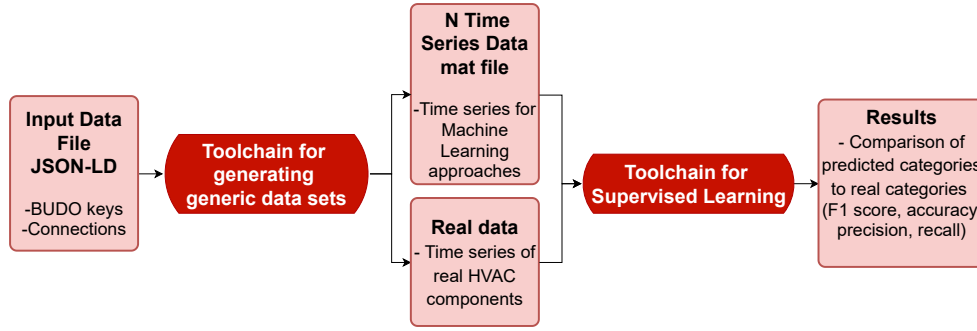


Figure 1: Process overview of the toolchain for supervised learning algorithms of topology detection.

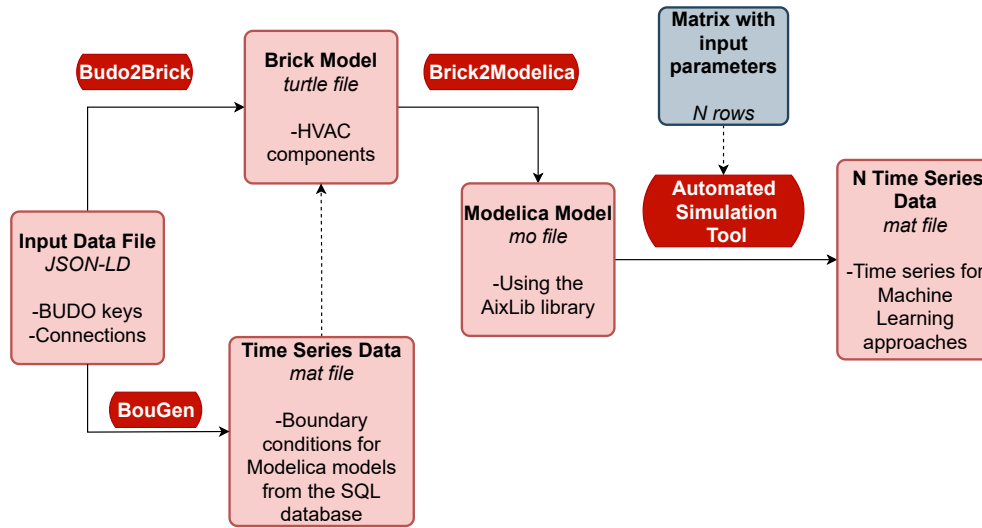


Figure 2: Process overview of the toolchain for generating generic data sets.

- **BouGen:** Downloads time series data from real sensors in *mat* format.
- **Budo2Brick:** Takes the information from the BUDO keys in JSON-LD file that uses terms with Linked Data from different ontologies. This is transformed into Brick, using the *turtle* format for the model. It also adds the data properties and boundary conditions of the system.
- **Brick2Modelica:** In this step, data is extracted from the Brick model and the Modelica model is generated. This is done using the SPARQL Protocol and RDF Query Language [40].
- **Automated simulation tool:** After the generation of the model in Modelica, our tool automatically simulates the model using the *Dymola-Python interface* [6]. The tool takes parameters from a matrix to be modified in each simulation and obtains the desired time series. A Design of Experiments approach provides here the appropriate matrix of different parameters that are varied 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 to use them for machine learning purposes.

3.2.1. JSON-LD and BudoOnt Ontology

It is important to emphasize that the initial information of the system is described in JSON-LD format. As JSON-LD uses terms linked by ontologies, the information contained in BUDO schema must be represented by ontologies. In addition, other terms are needed related to HVAC systems that are not contained in Brick schema, but that are required to create the Modelica model of an energy system. For these reasons, the BudoOnt ontology is developed, previously initiated by Stinner et al. [46] and expanded 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 is also a subset of the data properties of the BudoOnt ontology. In this case, some properties are added in order to facilitate the transformation from the initial input to Modelica,

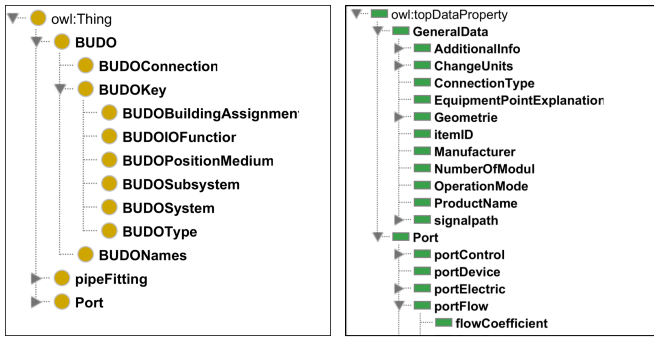


Figure 3: Part of BudoOnt ontology in Protégé

such as 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* [7]. 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 context shown 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 *BUDOBBuildingAssignment*. 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"},
    "BUDOBBuildingAssignment": {"@id": "BUDO-M: BUDOBBuildingAssignment"}}
doc = {
    "@context": context,
    "Zone": "K12",
    "TimeStart": "2016-07-09 00:00:00",
    "TimeEnd": "2016-07-16 00:00:00",
    "BUDOBBuildingAssignment": "BL-4120" }
```

Figure 4: Context of a JSON-LD input file

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

3.3. Study Area

We apply the developed methodology to the E.ON Energy Research Center Main Building [11]. Its energy system consists mainly of a heat pump, two condensing boilers and a gas fired CHP. A chiller completes the energy conversion as a cold producer. The energy system supplies differ-

ent 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. As can be seen, 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 can be seen in the diagram, 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.

There are more systems related to these that take part in them but they will be isolated from the rest and the case will focus on the heat pump and the heat exchanger.

3.4. 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 stem from the *AixLib* library [31]. In the case of the heat pump, the model uses a coefficient of performance (COP) that varies with temperatures in the same way that the Carnot efficiency changes; the heat exchanger model adopted from this library uses constant efficiency.

These models offer a higher amount of data streams, such as temperatures and volume flow rates than measured in the real system. Thus, it is possible to generate a greater amount of time series data than available with the monitoring system in the building to train the algorithms. Following this, a Design of Experiments methodology is applied in the interest of getting as many time series as feasible and the most similar to reality.

3.4.1. Heat Pump

With a Design of Experiments technique, it has to be settled first which factors to modify in each of the simulations and in which levels they vary. In the case of the heat pump model, the factors that are decided are (see figure 6):

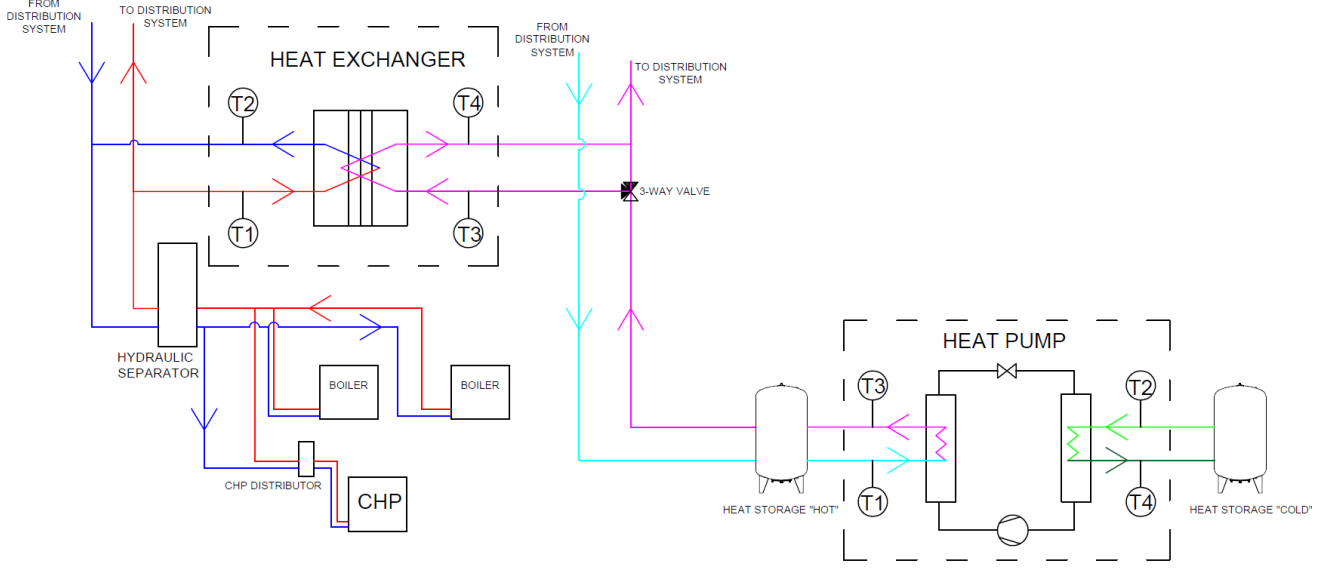


Figure 5: Energy systems of the use case.

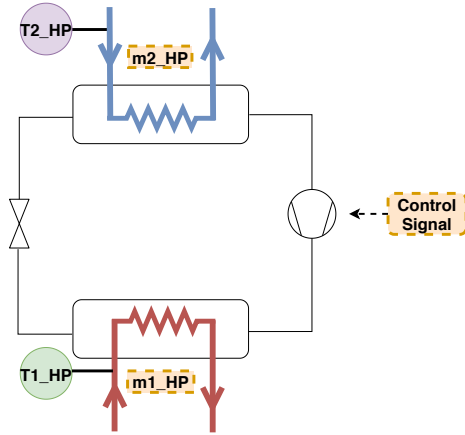


Figure 6: Factors that vary in the heat pump model simulations.

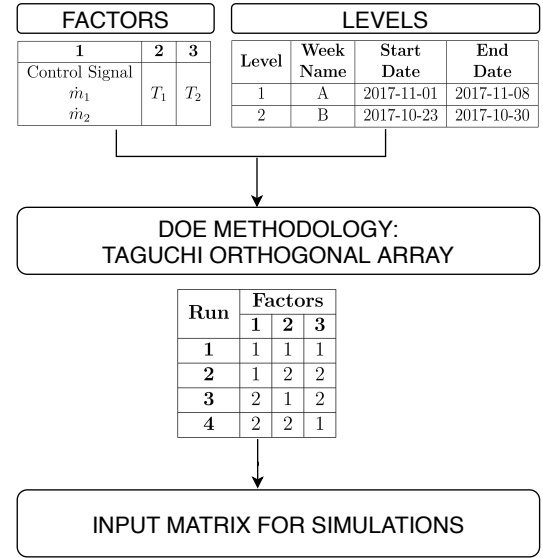


Figure 7: Example with the procedure followed to obtain the time series with the heat pump model.

first, the simultaneous measurements of the control signal of the compressor, the mass flow rate of the high temperature side (\dot{m}_1) and the mass flow rate of the low temperature side (\dot{m}_2). Second, the inlet temperature of the hot side (T_1) and third, the inlet temperature of the cold side (T_2).

Taguchi orthogonal arrays are chosen for the design of the simulations [49]. This approach is preferred over others 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 with respect to other Design of Experiments method-

ologies, because it minimizes the number of simulations to perform but without decreasing the accuracy substantially.

Figure 7 shows an example of the procedure that has been followed: with the factors mentioned above, two different levels are considered consisting of the real measurements of the sensor of the monitoring system that corresponds to that

factor. In this case, weeks A and B are taken as different levels of each factor. With this, following Taguchi's orthogonal array design, the matrix is designed with four simulations to be obtained from the model.

Within this work, a set of simulations has been made 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.

3.4.2. Heat Exchanger

Following the procedure described with the heat pump model, the methodology to obtain the time series from the model of the heat exchanger is similar to this one. In this case, it consists in maintaining as a 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) and to change as another factor of each of the simulations the efficiency (η) of the heat exchanger (see figure 8).

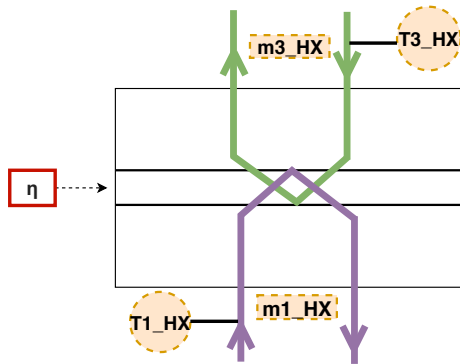


Figure 8: Factors that vary in the Heat Exchanger model simulations.

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

3.4.3. 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 that consists of connecting the two isolated systems mentioned above is studied. To do this, several simplifications are made with respect to the real case.

The main assumption is that the output of the circuit that exchanges heat with the condenser of the heat pump 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

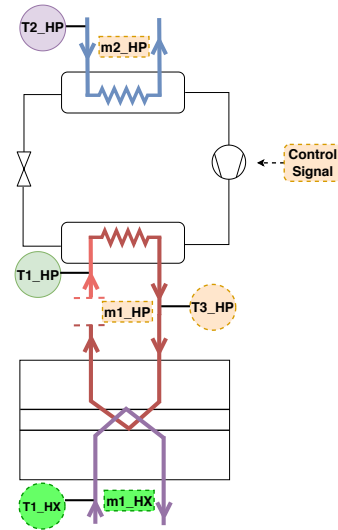


Figure 9: Factors changed in the case with the heat pump connected to the heat exchanger.

play an important role in the way these two systems are connected.

In order to simulate this system, the data set of the simulated time series with the heat pump has been taken and its output has been used as input by 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 heat exchanger model has been simulated separately and with these variables as input, taking for T1_HX and m1_HX the real simultaneous measurements of the weeks previously considered.

3.5. Toolchain for Machine learning

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

With a focus to detect the topology and potential connections between energy systems, three different cases are established. In all of them, the classification tasks have been carried out with real data streams and with simulated data streams. This work also explores the case of training the algorithm with simulated time series, which is then validated with real time series.

It is possible to see in each case the data streams that are considered connected (in some cases distinguishing directly if both data streams belong to the same hydraulic circuit or indirectly if they are from different loops but in the same system) and those that are considered not connected.

In particular, the cases studied and their different assumptions are:

- **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 connection* of temperature sensors on different sides (high and low temperature) of the heat pump (T3_HP and

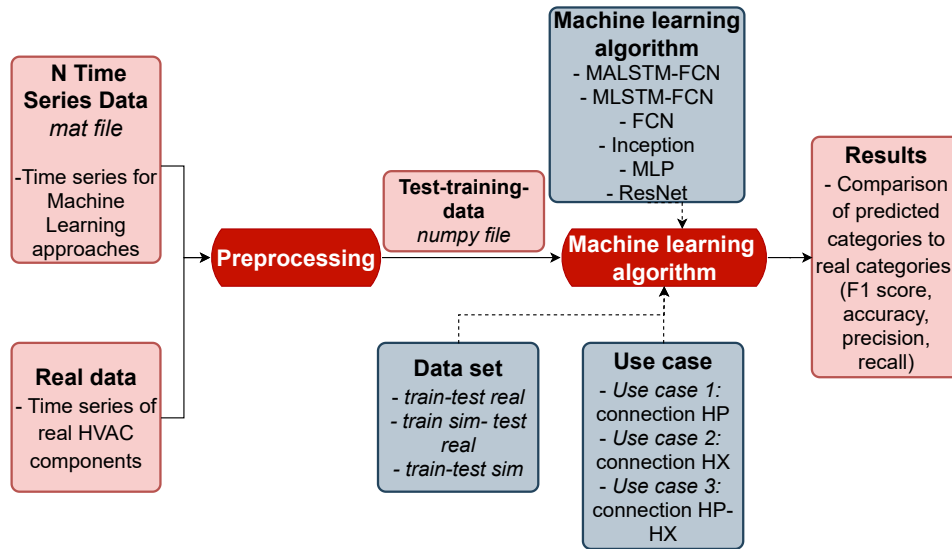


Figure 10: Process overview of the toolchain for supervised learning algorithms of topology detection.

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 they have been simulated following the subsection 3.4.3 (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.

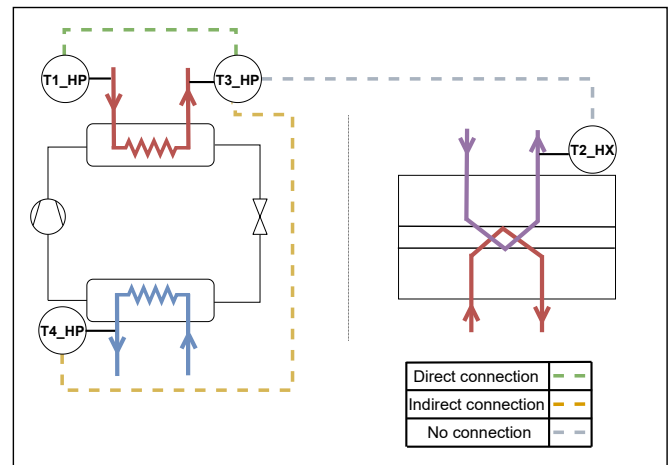


Figure 11: Case 1: Classification of data streams connected (direct and indirect) and not connected with a heat pump and a heat exchanger (isolated one from the other).

3.5.2. Preprocessing

The described models together with 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 to be obtained at the end of simulation. The common settings that have been used in all simulations are:

- Start Time: 0 s, Stop Time: 604800 s. All simulations have been performed with a duration of one week.
- Interval length: 300 s. Constant steps of 5 minutes are adopted.

- 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.

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.

As for the real time series of data streams, these 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 there are errors in the measurements or because there are constant values during a period of time. The criterion adopted for the elimination of weekly time series is based on the standard deviation. Thus, in the

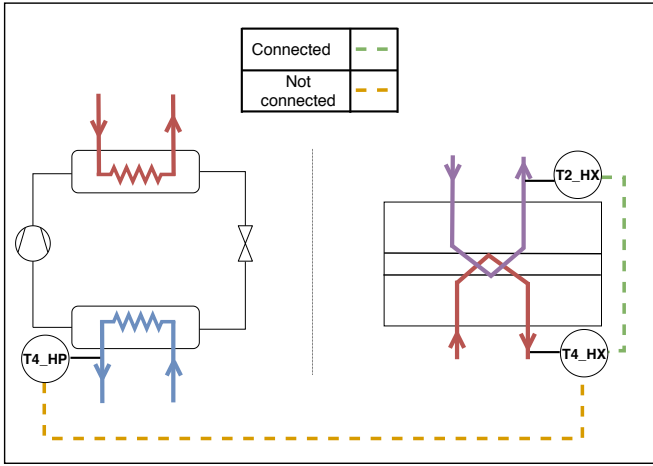


Figure 12: Case 2: Classification of data streams connected and not connected with a heat pump and a heat exchanger isolated one from each other.

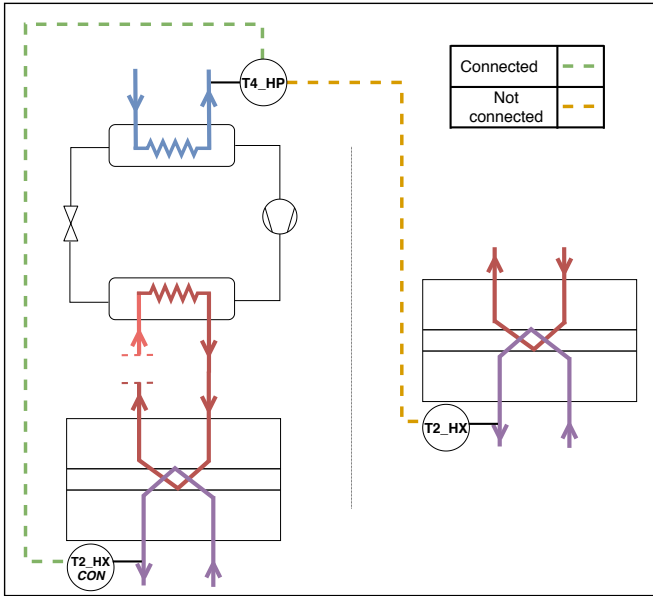


Figure 13: Case 3: Classification of data streams connected and not connected with a heat pump connected to a heat exchanger and an isolated heat exchanger.

case of the heat exchanger, if any of the weeks has in any of the four temperature sensors a standard deviation less than $0.3\text{ }^{\circ}\text{C}$, it is eliminated. With the heat pump a less restrictive criterion is adopted, where the standard deviation is limited to $0.5\text{ }^{\circ}\text{C}$.

Afterwards, both real and simulated time series of data streams are processed in the same way. First, subsets of the weeks are broken down to days (to have more time series to train and test) and then the measurements are resampled in steps of 5 min, resulting in time series of length 288. Resampling is done by applying the mean or backwards or forward interpolation, since depending on each time series, the appropriate method is different.

After resampling the time series, a Hampel filter is ap-

plied to remove outliers [36]. 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 [33], and 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 are shuffled. This is done with the help of the Scikit Learn library [37]. For the cases in which real measurements are used, we proceed in the same way. 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.

3.5.3. Algorithms

We use six different algorithms based on Convolutional Neural Networks (CNN) for classification in our use cases.

We use the implementation of Multivariate Long-Short-Term-Memory with Fully Convolutional Network Layer (MLSTM-FCN) provided by Karim et al. [24]. In a comparison of different deep learning approaches on the UCR data set (on univariate [23] and multivariate [24] time series classification), this implementation outperforms the other approaches 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 important parts of the time series [24]. The adjustments that can be made in the algorithm are the number of epochs and the batch size. An epoch refers to each time all the training samples pass through the entire network. It is adjusted in all cases to 10, since the time series sizes are not high and a larger number is not needed to improve the results. As for the batch size, this refers to the number of samples needed to run before the neural network weights are adjusted. This is set to 128 for all cases, as it is the number recommended by the authors.

Furthermore, four implementations of algorithms from Ismail Fawaz et al. [21] are used. A total of nine deep learning algorithms are implemented in the approach Ismail Fawaz et al. [21]. Unfortunately, the other algorithms could not be used, because among other things, they did not deliver results for such short time series that we used in each case.

Wang et al. [52] 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 [21].

They compared the MLP with a Residual Network (ResNet) implementation [52]. This 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 however, must also be fed with substantial training data.

The third approach used by Wang et al. [52] is a Fully Convolutional Network (FCN). Here the Fully Convolutional Layer is used as a feature extractor. This offers the advantage that it extracts individual sections of the time series as individual features.

Ismail Fawaz et al. [22] developed InceptionTime, which is inspired by the Inception-v4 architecture [48] (a ResNet variant) and should serve as an equivalent to AlexNet [1], which is a classic deep learning model for image classification. Its advantages are low dependence on training data and fast execution with constant or better results. We call this algorithm in the following Inception.

4. Results

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

The results of the simulated models are compared with the actual measurements of these systems on the same dates and under the same conditions. In order to analyze the performance of the model 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 results of the simulation and the real measurements are very similar (RMSE = 2.82 K for T4_HP and RMSE = 1.65 K for T3_HP), and we can see in the figure that the time series present the same tendency and behaviour.

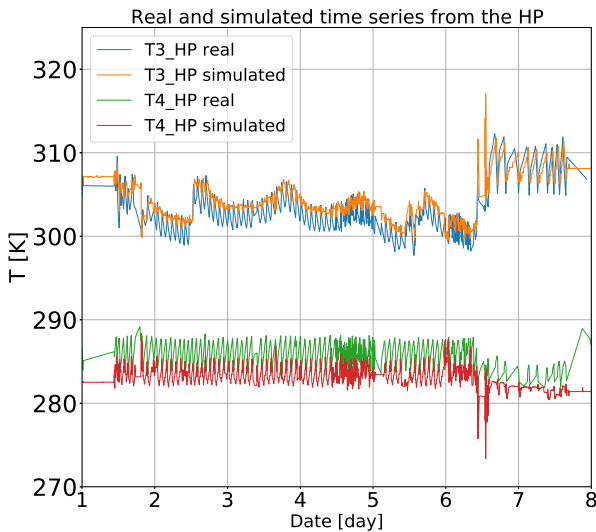


Figure 14: Comparison of the simulation results of the heat pump model and the real measurements of one week (T3_HP and T4_HP).

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. These simulations are executed according to the methodology explained in 3.4.2 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. In spite of this, the behavior and trends of both time series are comparable (real and simulated), resulting in a convenient model to get the simulated time series for subsequently training the algorithms.

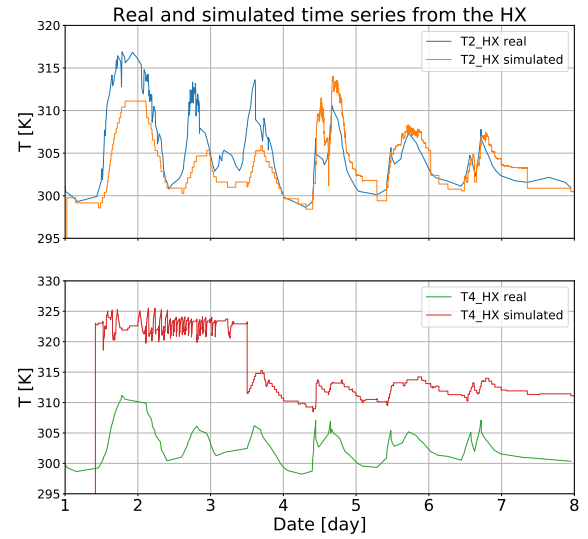


Figure 15: Comparison of the simulation results of the heat exchanger model and the real measurements of one week (T2_HX and T4_HX).

Two further examples of simulations with the heat exchanger model are shown in figure 16, with results of T2_HX and T4_HX. With the time series in this figure, it can be observed how different samples are obtained with the same model and how all of them are consistent with reality. Hence, this has the result of being a tool that allows scalability when obtaining new data for subsequent applications.

With reference to 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 real measurements with the simulated ones using the connected case of the same weeks as boundary conditions. It is observed 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 to heat up (T3_HX).

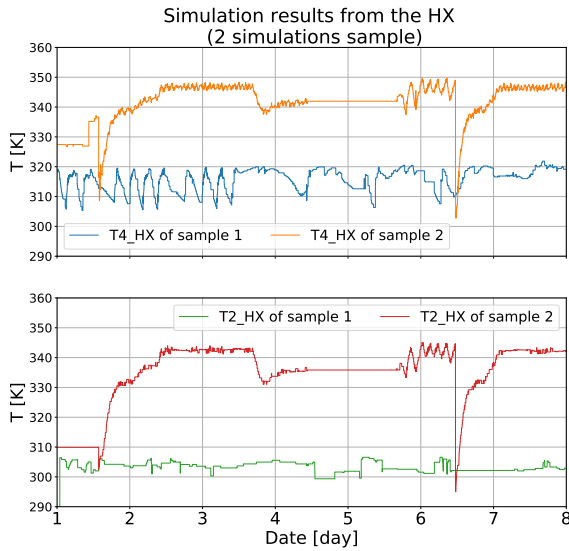


Figure 16: Simulated data streams of two samples of the heat exchanger (T2_HX and T4_HX).

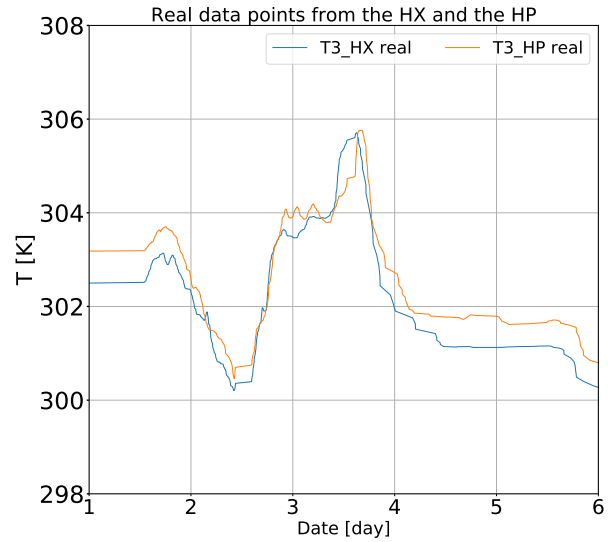


Figure 18: Real data streams of T3 in the heat pump and T3 in the heat exchanger.

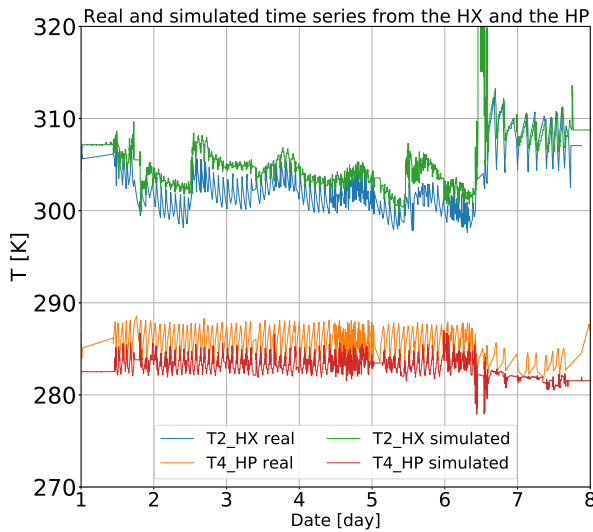


Figure 17: Comparison of the simulation results of the heat pump connected to the heat exchanger model and the real measurements of one week (T2_HX and T4_HP).

In the real system, there is a storage tank and valves between 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 output of the heat pump to pass through the input of the heat exchanger is justified.

4.2. Classification results

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 used in isolation. It is apparent from the results (table 1) that the F1 score increases to a value between 97.1 % and 97.9 % (Inception) in the cases with simulation data (maximum achievable results). However, testing and training of real data (classical method) shows that the algorithm is not able to 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).

Nevertheless, by training the algorithm with the results of the test with the simulation data set (above 97 % F1 score except for MLP), it is possible to categorize the real data streams. The F1 score reaches a maximum value of 70.5 % (Inception) by testing with real data streams. The accuracy in this case is 72.3 %. This 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. With this algorithm, both direct and indirect heat pump connections are identified, but by analyzing the confusion matrices, the non-connections (i.e. the heat pump data streams and the ones of the heat exchanger, both isolated

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 training and testing were done with simulation data, "sim real" means that training was done with simulation data and testing was done with 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

Table 2

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

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. 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 %).

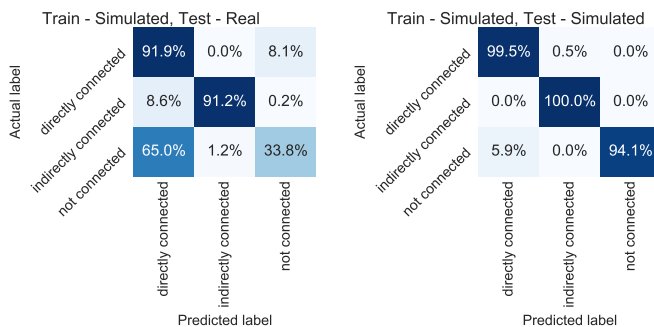


Figure 19: Accuracy of predicted classes of test data with the Inception algorithm in case 1 (left: trained with simulated and tested with real data, right: trained and tested with simulated data).

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

The example of this case proposes to recognize the different topology of the connection inside a heat exchanger unlike two isolated systems (heat exchanger and heat pump). Therefore, in this instance there are two labels, 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 only one of the two labels is identified. 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 ac-

complished (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. The resulting confusion matrices with the best tests of this case are shown in figure 20. The actual connections were identified in the best result to a 100 % true positive rate. Non-existing compounds were identified 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.

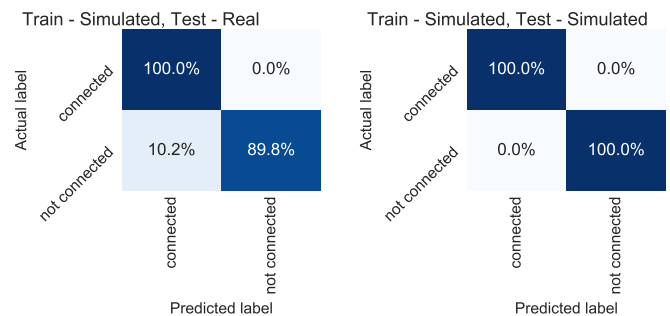


Figure 20: Accuracy of predicted classes of test data with the FCN algorithm in case 2 (left: trained with simulated and tested with real data, right: trained and tested with simulated data).

In this case (a heat exchanger), it is important to consider which temperatures are connected and which are not, because not all temperature sensors can be considered 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 as a connection in the classification tasks that have been made in this work, since T1_HX comes from the high temperature systems and T2_HX reaches the temperature that is established in the heat balance, always being a few degrees higher than T4_HX. With this, it has been considered that T2_HX, T3_HX and T4_HX are connected

but that 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. This is the best result obtained.

Based on piping and instrumentation diagrams, data streams could be connected but cannot be considered connected in the classification. If these are manually checked by experienced technicians, no connection can be detected either. Because of this fact, they cannot be considered as connected in the tasks of classification.

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, in this case there are two labels, namely for the connection and no connection classes.

Table 3

F1 score of all used algorithms in case 3 ("real" means that training and testing were done with real data, "sim" means that training and testing were done with simulation data, "sim real" means that training was done with simulation data and 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). In relation to the real data streams, the results agree with the previous cases in which the classification is not obtained and only one of the classes is identified. Regarding the real data streams trained with the simulation data sets, unlike in the previous cases, the improvement that occurs is not very high. 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 (\approx 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 also differ significantly from the best algorithm in the classification of 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 that similar to the real ones in this model, compared to

the ones from other cases. The reason for this is that the heat exchanger has been simulated taking 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 same dates as those used in the simulations of the heat exchanger, so although the validation of the model is successful, there are certain discrepancies. This could be solved by developing a model that connects both systems and considers a test with the conditions of both taken at the same time.

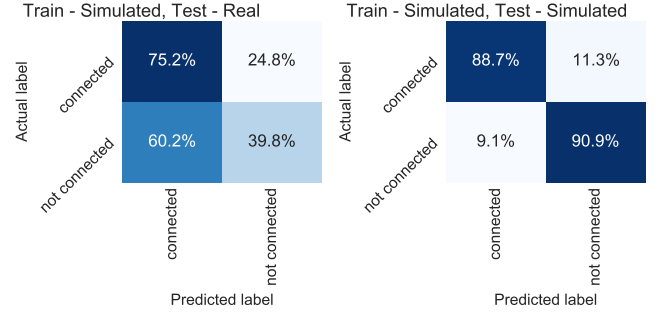


Figure 21: Accuracy of predicted classes of case 3 with the Inception algorithm (training - simulated data, testing - real data) and ResNet (training, testing - simulated).

4.2.4. Overall results

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

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

The goal of this approach is training with simulation data and testing with real data. Here the results are to be judged differently. In use case 2 an F1 score of 94.9 % is achieved, but use case 3 is only marginally above the random selection (F1 score 56.1 %) and 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.

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

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 models is large (22 to 60 %), depending on the use case. This shows that the idea of the generation of time series data using simulation models works. In all considered cases and used algorithms, algorithms trained with simulated data achieved better results than those trained with real data.

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. These good results were not achieved in every use case, but in all of them a better result was achieved with simulated time series. Since only three use cases were considered, a general assertion cannot be made.

The fact that none of the cases has obtained successful results by training and validating the algorithm with real data is highlighted. Some of the reasons that may explain this outcome are errors in the real measurements and constant measurements on many occasions, which does 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. This 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 real 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. 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 with real data may indicate that this theoretical goal was partially achieved.

In this approach, three different connections of only two

different systems of technical building equipment were presented. This limits the statement about the general application of our approach.

5.2. Algorithms

No general statement can be made about the multivariate time series classification algorithms. For training with simulated data and testing with real data, the Inception algorithm was the best algorithm in two of the three use cases and 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 can be achieved here as well.

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 with 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. The other algorithms generally achieve this with overall significantly better 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. A comparison of 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 [16]. Since the systems are very different (hydraulic system versus air handling unit), a comparison of the results is also questionable. The results in [44] with a maximal accuracy of 52.1 %, which were created with 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 the detection of 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, which is trained using generalized time series from grey-box simulation models. Nevertheless, some research still needs to be done, so that this approach can be applied in a scalable way. This approach must be checked against further

cases of connection between technical building equipment types.

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 physically simulated data and real data shows that physically simulated data can be an alternative to real data in identifying connections in energy systems. Since publicly available building energy system data is rare, in which the topology data is also stored, generation of generalized data covering a wider 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 real systems. However, this can be a disadvantage, especially for the application in machine learning algorithms. These algorithms are then not necessarily robust. Here, integration of disturbances into the physical model could help to represent the real operation more robustly.

Transferring our results to thermal systems that have higher thermal inertia, such as underfloor heating or concrete core activation, is also interesting. Here, it may be that other algorithms are 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 difficult. If the connections can be clearly defined and no neighbouring 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:** Methodology, Investigation, Data curation, Writing - Original Draft. **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 and the Ministry of Universities of Spain through the "Formación de Profesorado Universitario" programme ref. FPU 19/04012.

References

- [1] Alex Krizhevsky, Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc, pp. 1097–1105.
- [2] Balaji, B., Bhattacharya, A., Fierro, G., Gao, J., Gluck, J., Hong, D., Johansen, A., Koh, J., Ploennigs, J., Agarwal, Y., Bergés, M., Culler, D., Gupta, R.K., Kjærgaard, M.B., Srivastava, M., Whitehouse, K., 2018. Brick : Metadata schema for portable smart building applications. *Applied Energy* 226, 1273–1292.
- [3] Bhattacharya, A., Ploennigs, J., Culler, D., 2015. 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, pp. 33–34.
- [4] Clayton Miller, 2019. More buildings make more generalizable models: Benchmarking prediction methods on open electrical meter data.
- [5] D'Agostino, D., Zangheri, P., Castellazzi, L., 2017. Towards nearly zero energy buildings in europe: A focus on retrofit in non-residential buildings. *Energies* 10, 117.
- [6] Dassault Systemes, 2020. Dymola systems engineering. URL: <https://www.3ds.com/products-services/catia/products/dymola/>.
- [7] Digital Bazaar, Inc., 2020. Pyld. URL: <https://github.com/digitalbazaar/pyld>.
- [8] European Commission, 2018. 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.
- [9] Fierro, G., Pritoni, M., AbdelBaky, M., Raftery, P., Pfeffer, T., Thomson, G., Culler, D.E., 2018. Mortar: An open testbed for portable building analytics, in: *The 5th ACM International Conference on Systems for Built Environments (BuildSys '18)*. URL: http://bets.cs.berkeley.edu/publications/2018buildsys_mortar.pdf.
- [10] Fürst, J., Chen, K., Katz, R.H., Bonnet, P., 2016. Crowd-sourced bms point matching and metadata maintenance with babel, in: *2016 IEEE International Conference on Pervasive Computing and Communication workshops (PerCom workshops)*, Ieee, pp. 1–6.
- [11] Fütterer, J., Constantin, A., Schmidt, M., Streblow, R., Müller, D., Kosmatopoulos, E., 2013. 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*, pp. 5696–5701.
- [12] Gao, H., Koch, C., Wu, Y., 2019. Building information modelling based building energy modelling: A review. *Applied Energy* 238, 320–343.
- [13] Granderson, J., Lin, G., Harding, A., Im, P., Chen, Y., 2020. Building fault detection data to aid diagnostic algorithm creation and performance testing. *Scientific Data* 7, 65.
- [14] Hazyuk, I., Ghiaus, C., Penhouet, D., 2014. Model predictive control of thermal comfort as a benchmark for controller performance. *Automation in Construction* 43, 98–109.
- [15] Hong, D., 2018. 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.
- [16] Hong, D., Cai, R., Wang, H., Whitehouse, K., 2019. 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, pp. 203–212.
- [17] Hong, T., Li, H., Macumber, D., Fleming, K., Wang, Z., 2020a. Generation and representation of synthetic smart meter data. *Building Simulation*.
- [18] Hong, T., Wang, Z., Luo, X., Zhang, W., 2020b. State-of-the-art on research and applications of machine learning in the building life cycle. *Energy and Buildings* 212, 109831.
- [19] Huchtkoetter, J., Reinhardt, A., 2019. A study on the impact of data sampling rates on load signature event detection. *Energy Informatics* 2, 471.
- [20] International Energy Agency, 2013. Transition to Sustainable Buildings. OECD.
- [21] Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A., 2019a. Deep learning for time series classification: a review.

- Data Mining and Knowledge Discovery 33, 917–963.
- [22] Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D.F., Weber, J., Webb, G.I., Idoumghar, L., Muller, P.A., Petitjean, F., 2019b. Inceptiontime: Finding alexnet for time series classification.
- [23] Karim, F., Majumdar, S., Darabi, H., 2019. Insights into lstm fully convolutional networks for time series classification. *IEEE Access* 7, 67718–67725.
- [24] Karim, F., Majumdar, S., Darabi, H., Harford, S., 2018. Multivariate lstm-fcns for time series classification.
- [25] Kazmi, H., Munné-Collado, Í., Mehmood, F., Syed, T.A., Driesen, J., 2021. Towards data-driven energy communities: A review of open-source datasets, models and tools. *Renewable and Sustainable Energy Reviews* 148, 111290.
- [26] Klein, L., Kwak, J.y., Kavulya, G., Jazizadeh, F., Becerik-Gerber, B., Varakantham, P., Tambe, M., 2012. Coordinating occupant behavior for building energy and comfort management using multi-agent systems. *Automation in Construction* 22, 525–536.
- [27] Koh, J., Balaji, B., Akhlaghi, V., Agarwal, Y., Gupta, R., 2016. Quiver: Using control perturbations to increase the observability of sensor data in smart buildings.
- [28] Lange, H., Johansen, A., Kjærgaard, M.B., 2018. Evaluation of the opportunities and limitations of using ifc models as source of building metadata, in: Ramachandran, G.S., Batra, N. (Eds.), *BuildSys'18*, The Association for Computing Machinery. pp. 21–24.
- [29] Li, S., Hong, D., Wang, H., 2020. Relation inference among sensor time series in smart buildings with metric learning. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 4683–4690.
- [30] Modelica Association, 2017. *Modelica - a unified object-oriented language for systems modeling: Language specification - version 3.4*.
- [31] Müller, D., Lauster, M., Constantin, A., Fuchs, M., Remmen, P., September 2016. Aixlib - an open-source modelica library within the iea-ebc annex 60 framework, in: *BauSim 2016*, pp. 3–9.
- [32] Nordhaus, W., 2019. Climate change: The ultimate challenge for economics. *American Economic Review* 109, 1991–2014.
- [33] Oliphant, T.E., 2006. *A guide to NumPy*. volume 1. Trelgol Publishing USA.
- [34] Pan, Y., Zhang, L., 2021. Roles of artificial intelligence in construction engineering and management: A critical review and future trends. *Automation in Construction* 122, 103517.
- [35] Pang, Z., Chen, Y., Zhang, J., O'Neill, Z., Cheng, H., Dong, B., 2020. Nationwide hvac energy-saving potential quantification for office buildings with occupant-centric controls in various climates. *Applied Energy* 279, 115727.
- [36] Pearson, R.K., Neuvo, Y., Astola, J., Gabbouj, M., 2016. Generalized hampel filters. *EURASIP Journal on Advances in Signal Processing* 2016, 115.
- [37] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É., 2011. Scikit-learn: Machine learning in python.
- [38] Pritoni, M., Bhattacharya, A.A., Culler, D., Modera, M., 2015. 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. (Eds.), *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, ACM. pp. 133–136.
- [39] Privara, S., Cigler, J., Váňa, Z., Oldewurtel, F., Sagerschnig, C., Žáčková, E., 2013. Building modeling as a crucial part for building predictive control. *Energy and Buildings* 56, 8–22.
- [40] Prud'hommeaux, E., Seaborne, A., 2007. Sparql query language for rdf.
- [41] Remmen, P., Lauster, M., Mans, M., Fuchs, M., Osterhage, T., Müller, D., 2018. Teaser: an open tool for urban energy modelling of building stocks. *Journal of Building Performance Simulation* 11, 84–98.
- [42] Ruiz, A.P., Flynn, M., Large, J., Middlehurst, M., Bagnall, A., 2021. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery* 35, 401–449.
- [43] Stinner, F., Kornas, A., Baranski, M., Müller, D., 2018. Structuring building monitoring and automation system data, in: REHVA (Ed.), *The REHVA European HVAC Journal - August 2018*. REHVA Journal, pp. 10–15.
- [44] Stinner, F., Raßpe-Lange, L., Baranski, M., Müller, D., 2019a. Takeshi: Application of unsupervised machine learning techniques for topology detection in building energy systems. *Journal of Physics: Conference Series* 1343, 012041.
- [45] Stinner, F., Wiecek, M., Kümpel, A., Baranski, M., Müller, D., 2021. Automatic digital twin data model generation of building energy systems from piping and instrumentation diagrams, in: *Proceedings of ECOS 2021*.
- [46] Stinner, F., Yang, Y., Schreiber, T., Bode, G., Baranski, M., Müller, D., 2019b. Generating generic data sets for machine learning applications in building services using standardized time series data, in: Al-Hussein, M. (Ed.), *Proceedings of the 36th International Symposium on Automation and Robotics in Construction (ISARC)*, International Association for Automation and Robotics in Construction (IAARC).
- [47] Stinner, F., Llopis, B., Kümpel, A., Müller, D., 2021. Deep learning supervised topology detection. URL: <https://github.com/RWTH-EBC/Deep-learning-supervised-topology-detection>. [Github repository].
- [48] Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A., 2017. Inception-v4, inception-resnet and the impact of residual connections on learning, in: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI Press. pp. 4278–4284.
- [49] Taguchi, G., Konishi, S., 1987. *Orthogonal Arrays and linear graphs: Tools for quality engineering*. American Supplier Institute, Dearborn, Mich.
- [50] Wang, W., Brambley, M.R., Kim, W., Somasundaram, S., Stevens, A.J., 2018. Automated point mapping for building control systems: Recent advances and future research needs. *Automation in Construction* 85, 107–123.
- [51] Wang, Z., Hong, T., 2020. Reinforcement learning for building controls: The opportunities and challenges. *Applied Energy* 269, 115036.
- [52] Wang, Z., Yan, W., Oates, T., 2017. Time series classification from scratch with deep neural networks: A strong baseline, in: *IJCNN 2017*, Ieee. pp. 1578–1585.
- [53] Yang, S., Wan, M.P., Chen, W., Ng, B.F., Dubey, S., 2020. Model predictive control with adaptive machine-learning-based model for building energy efficiency and comfort optimization. *Applied energy*, 271, 115147. *Applied Energy* 271, 115147.
- [54] Zhou, X., 2010. A plug and play framework for an HVAC air handling unit and temperature sensor auto recognition technique.