APPLICATION OF AUTOENCODERS ON MULTIVARIATE ANOMALY DETECTION IN BUILDING AUTOMATION SYSTEMS WITH VARIABLE SELECTION BASED ON SEMANTIC METADATA OF THE FACILITY

Miloš Šipetić*[1], Michael Schöny[1], Jerik Catal[1]

[1]Austrian Institute of Technology, Department of Energy, Vienna, Austria

*Corresponding Author: Sipetic.Milos@ait.ac.at

ABSTRACT

We describe the application of autoencoder neural network on building automation system (BAS) and IoT sensor data enriched with semantic metadata.
Datapoints to use for each anomaly detector are specified by querying the semantic representation of the facility and the accompanying IoT network using system, subsystem, connectedness or spatial criteria. Additional criteria can be used depending on the richness of the underlying representation and may include geometrical (core or perimeter zones, horizontal or vertical adjacency, space size), control strategy and similar information. List of used datapoints is further expanded with weather data such as temperature, humidity and wind speed, as well as synthetic datapoints obtained by lagging existing datapoints, measuring intervals between readings or calculating aggregations or derivations. Input data is cleaned of erroneous measurements and used for training autoencoder anomaly detectors dealing with specific subsystems. Individual autoencoder models are trained on data belonging to different systems and of differing spatial characteristics. We show that described method scales well to facility-wide generation of anomaly detection models. Reconstruction error threshold for detecting 10 anomaly groups within the test dataset reaches between 0.12 and 0.26 depending on the building system and number of input variables involved. Quality and normality of input data remains a requirement for training anomaly detection models based on unsupervised algorithms. Validation possibilities with existing facility data, and data sanitation techniques required to avoid saturation of output with false positives are evaluated. Approaches at incorporating operator feedback on detected anomalies are also discussed.

1 INTRODUCTION

Anomaly detection refer to the process of detecting data instances that significantly deviate from the majority of data instances (Pang *et al.*, 2021). Anomalies can disturb the operation of the system, reduce its efficiency, or completely prevent it from working. Liu *et al.*, 2020 classify anomalies into point anomalies and context anomalies, where point anomalies correspond to single-measurement anomalous values of an individual datapoint, while context anomalies occur over a period of time and can relate to multiple datapoints from the observed system. Machine learning approaches to anomaly detection include unsupervised, semi-supervised and supervised methods.

Autoencoders (Michelucci, 2022) are commonly used unsupervised machine learning techniques applied to anomaly detection. One of the main benefits of using them for anomaly detection is that they can be trained with data containing no anomalies. This is beneficial as anomalies happen rarely, and even more rare are facility or building automation system datasets containing labelled anomalies.

Araya *et al.*, 2016 propose a contextual anomaly detection framework and within it provisions to include temporal and spatial context. Tziolas *et al.*, 2022 investigated application of autoencoders on

https://doi.org/10.52202/077185-0060

industrial datasets. Unsymmetrical autoencoders were researched in Sun *et al.*, 2016 and showed improvement over symmetrical variants.
Linked data is increasingly used to connect knowledge about various domain and aid the data analysis in automation workflows. Specifically for domains related to the topic of this paper, numerous ontologies have been proposed that help with modelling buildings, such as BOT (Janowicz *et al.*, 2021) or IFCOWL (Pauwels and Terkaj, 2016), modelling building systems (TUBES (Pauen *et al.*, 2020), SAREF (Daniele, den Hartog and Roes, 2015)), automation control logic (Schneider, 2019) or combination thereof such as Brick (Balaji *et al.*, 2018) or Saref4Bldg (Poveda-Villalon and Garcıa-Castro).
Previous work in Gaida *et al.*, 2018 and Petrushevski *et al.*, 2017 explored use of semantically modelled automation system to perform automated data analysis and optimization. Work has also been done on integration of semantic knowledge base to support data analytics workflows in Šipetić *et al.*, 2020.
We build upon work on building and system ontologies, anomaly detection and autoencoders to propose usage of knowledge base representation of the facility and its systems to obtain a context for creating anomaly detectors targeted at specific systems and subsystems.

2 DEMONSTRATION SITE AND USE CASES

For the purpose of data collection, plant monitoring and model evaluation, FUTUREbase, a four-story office and laboratory building in Vienna, Austria, served as testbed (Figure 1). The building itself can be heated or cooled by a groundwater heat pump. A well pump system with two submersible pumps works as the main heat source. Daily and annual amounts of groundwater extraction and their return temperature to the ground are legally specified and need to be observed. The building consists of six floor levels, ranging from a basement floor up to the fourth floor. The entire HVAC[1] supply equipment, laboratory infrastructure and server rooms are located on the two lower levels. Offices and meeting rooms are placed in the upper floors which in turn are divided into HVAC zones. Each of these HVAC zones distribute heating or cooling energy for a selected group of rooms to control the indoor air temperature conditions via a thermal concrete core activation (CCA) of the ceiling. Concrete core temperature is regulated by a variable mass flow control within the hydraulic loops. Their supply mode can be easily switched from heating to cooling using a distribution valve.

Figure 1 - *Demonstration site FUTUREbase*

Initial investigations revealed that the CCA often alternated during summer and transition periods. This led to an idea to define an AI-supported detection of anomalous behavior in HVAC systems through

[1] Heating, Ventilation and Air Conditioning

https://doi.org/10.52202/077185-0060

the analysis of monitoring data. The goal of the use case was to identify potential misbehavior of data points from the BAS and visualize detected instances on a dashboard for further inspections. For this, use case requirements were specified. Here, pre-conditions like external and operational data as well as process related information were listed. Table 1 shows the data sources of mentioned pre-conditions.

***Table 1** - Use case pre-conditions and their data sources*

pre-conditions	data sources
External data	• IoT sensor data (temperature, humidity, motion, etc)
Operational data	• Sufficient amount of previously collected monitoring data for training of anomaly detection models • Milestone from which to check for anomalous or missing readings • Inclusion of operator feedback in further iterations • Incident list
Process related information	• Relationships of datapoints: naming scheme, other available documentation of the control system

All monitoring data from the BAS was transferred to a cloud platform, where almost 3.5 years of continuous data was collected. This data was used for training and evaluation of anomaly detectors.
Furthermore, an IoT sensor network was installed in the first and third floor to collect indoor environmental data of investigated zones / rooms (room air temperature, relative humidity, VOC, door and window contact, occupancy, brightness). The ZigBee communication protocol was selected for the IoT sensor network due to the wide availability of sensors and gateway equipment, simple deployment and data collection. ZigBee was especially convenient due to its adaptability and mesh networking. In a star network, all networked devices communicate with each other via a central node. In contrast, the ZigBee mesh allows arbitrary interconnections between edge (battery powered) and repeater (main-powered) devices.
Two Raspberry Pi 4 devices served as data loggers on each floor and were combined with ConBee II stick devices acting as gateways. The open-source software Home Assistant (Assistant, no date) ran as operating system on the Raspberry Pis. Associations of sensors to the network and their configuration were carried out via the Home Assistant interface. In Figure 2, the network area of installed IoT sensors for floor 1 is shown. Plugs and similar devices are main-powered and act as repeaters within the Zigbee network, effectively extending the reach of the network. This network topology reduces sensor failure because each battery-powered sensor can connect to any repeater within range if necessary. Network is also self-healing as multiple paths for a signal are possible, when one of the repeaters drops out of network, the network is able to reconfigure itself as long as there are viable routes and no disjoint subnetworks.

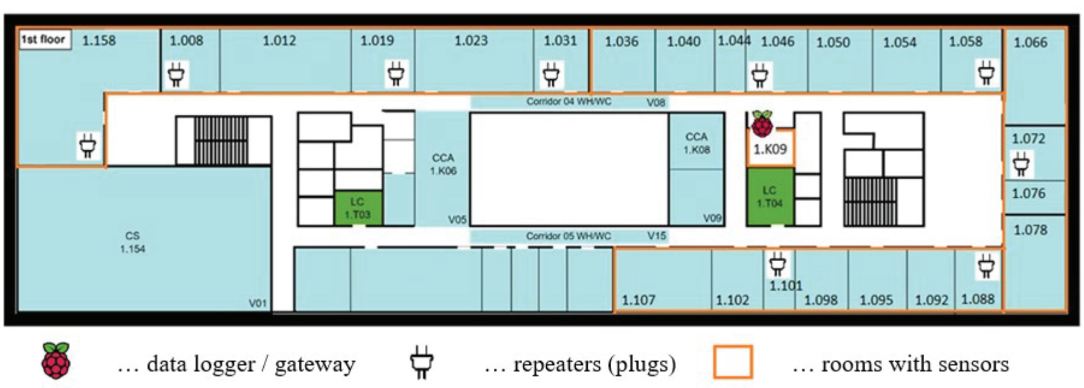

***Figure 2** – Map of installed IoT-equipment on the first floor of demonstration building*

https://doi.org/10.52202/077185-0060

3 METHODOLOGY

Where metadata model is presented, and machine learning workflow is broken down in steps with clarifications related to each step.

3.1 Facility model

As part of the project the Brick ontology was used to structure knowledge about the topology of the building, its HVAC subsystems and data points. Brick is an open-source ontology-based unified metadata schema which semantically describes physical, logical and virtual entities and their relationships. The standardized ontology is expandable and ensures a flexible data model that can be easily integrated into existing tools and databases. RDF (Resource Description Framework) represents logical statements about arbitrary things as the basis for semantic web technology and was designed by the W3C as a standard for formulating metadata. The general structure of an RDF data model consists of a subject that is logically connected to an object via a predicate. This gives the opportunity to provide machine-interpretable building data for the autoencoders workflow. Three sub-models were modeled for the building data model (knowledge database) as follows: the building, the BAS of heating / cooling supply / delivery for relevant areas (main service room & offices) and the IoT sensor network. The following figures show the basic structure of BAS and IoT sensor model. For a better understanding of the individual classes, color coding is shown in Figure 3:

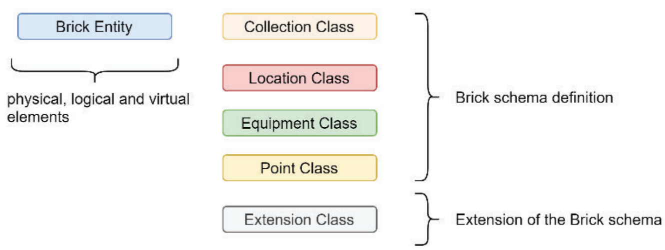

Figure 3 *- Color code for Brick schema*

Figure 4 outlines a part of the data model for the BAS and HVAC plant. Sub-system HK01 represents the main heat / cold supply system. HK01 is feeding O1HK which is located in Floor_U1 and contains O1HK_V02, a water system for concrete core activation. Within every sub-system entity like equipment (chiller, pumps, etc.), points (sensor types, etc.) exist and are somehow related with each other. For example, O1HK_V02 compose two regulating valves to control the volume flow of the heating (O1HK_V02_BTA_HV) or cooling (O1HK_V02_BTA_KV) circuit. These valves in turn include data points like a valve position command.

An overview of the data model structure regarding the IoT sensor network is shown in Figure 5. Here, the installed sensor devices for an office room are illustrated. An HVAC zone which is a logical entity to connect several rooms together, arranges meeting rooms and offices. In these offices are several different IoT sensor types (temperature, humidity, contact, illuminance) placed. The sensors itself contain the meta data of their unit.

https://doi.org/10.52202/077185-0060

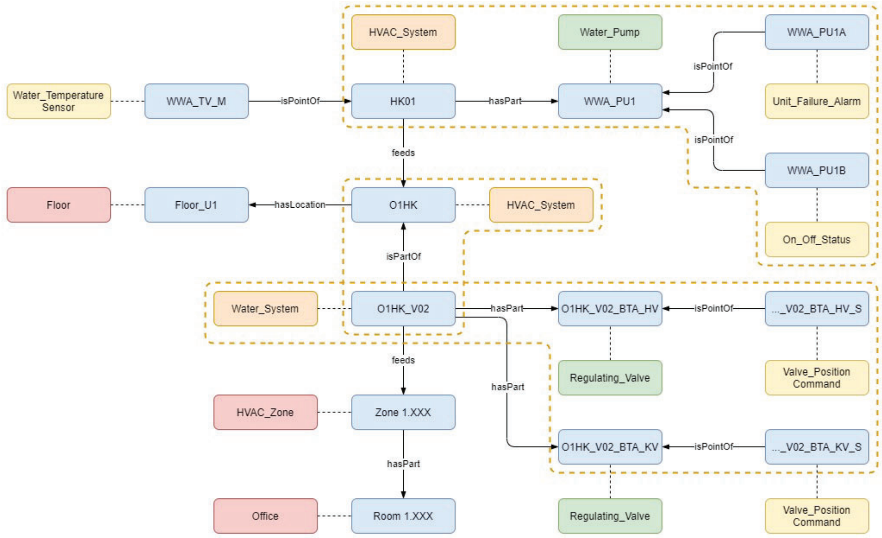

Figure 4 *- HVAC sub systems*

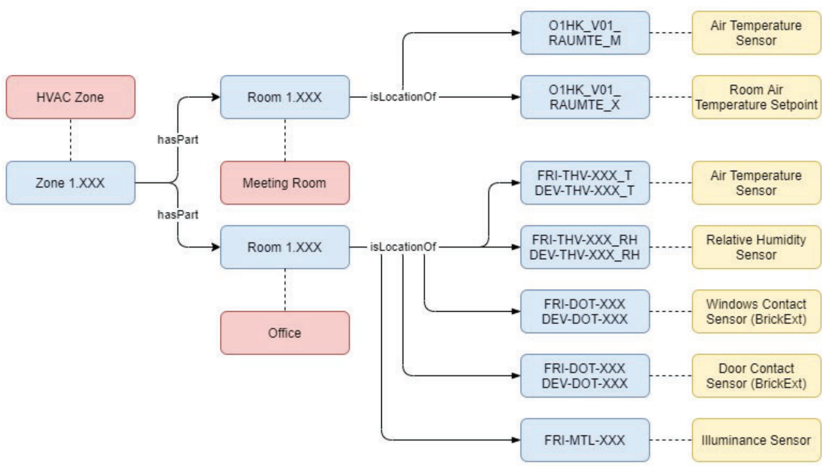

Figure 5 *- IoT sensors in rooms*

3.2 Machine learning workflow

Workflow shown on Figure 6 depicts progression of steps in the machine learning workflow.

First step involves selection of relevant datapoints that represent the context. Approaches to determining contextually relevant datapoints may include but are not limited to spatial adjacency and system connectedness. For the analysis of demonstration building's systems, we use the datapoints of the system itself and datapoints of its most closely connected subsystems. We query the facility knowledge base using the SPARQL query language (Quilitz and Leser, 2008).

To collect all the datapoints of a system we use query:

```
SELECT ?sub ?point
WHERE {
   ?point brick:isPointOf ?sub
}
```


https://doi.org/10.52202/077185-0060

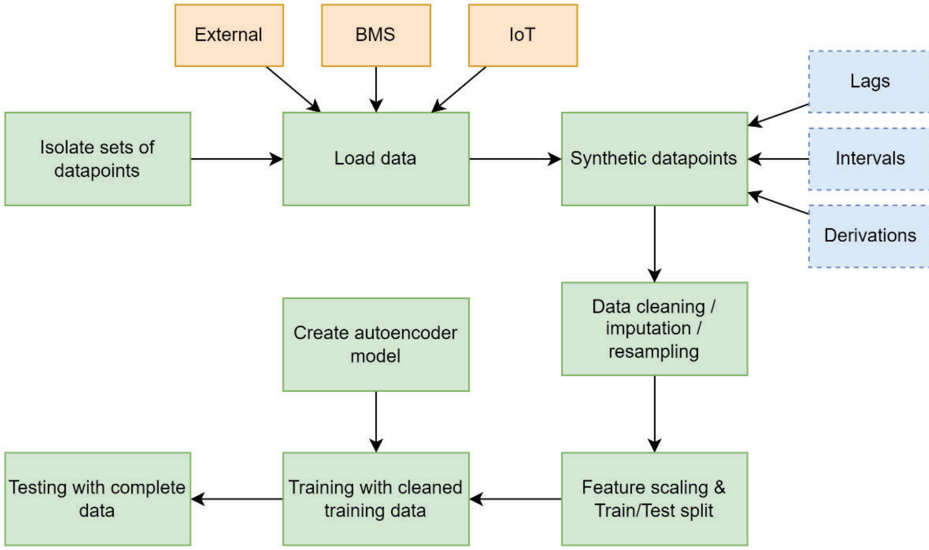

Figure 6 - *Workflow for data processing and anomaly detector training*

and to collect all the datapoints of all subsystems of a system we use query:

SELECT ?subsub ?point
WHERE {
?subsub brick:isPartOf ?sub .
?point brick:isPointOf ?subsub
}

Both queries are parametrized by replacing *?sub* placeholder with a reference to a specific system or a subsystem, and are executed iteratively over all systems of interest.

In case of system O1HK_V02, the returned datapoints are: O1HK_V02_BTA_TE_M, O1HK_V02_BTA_HV_S, O1HK_V02_BTA_KV_S, that represent temperature of the concrete core activation system, and settings of controllable valves of the heating and cooling, respectively.

In the next step data is collected from different internal (BMS and IoT network) and external (weather, forecast, calendar) systems (Figure 8). Next, synthetic datapoints are added. Synthetic datapoints may be simply lagged (time delayed) values of selected datapoints, but they may also include values derive from single or multiple datapoints. Derivation methods include, but are not limited to, average, median, minimum, maximum or total change of values for periods of arbitrary duration from the past (e.g. hourly, daily or weekly average for the past hour or 1, 2, 12, 24, 72 hours prior). Furthermore, as data collection is event-based and new events are arriving in non-deterministic intervals it is beneficial to calculate time intervals since last event for each selected datapoint. That would allow the anomaly detector to also detect problems with data collection itself (i.e. no data from sensor for the last two days).

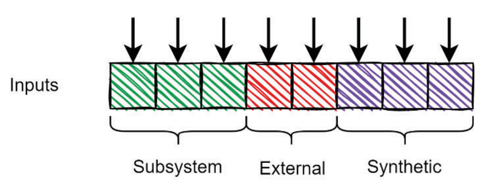

Figure 7 - *inputs to the neural network is composed out of multiple data sources. System/subsystem data includes datapoints collected by querying the system topology. External datapoints are collected from external data sources, while synthetic datapoints are dynamically generated rather than sensed.*

https://doi.org/10.52202/077185-0060

Data then needs to be cleaned of problems such as erroneous measurements, such as “stuck” values, instantaneous minimum, maximum or default readings. When running the system in the production, data will not be cleaned prior to running them through the anomaly detector, but in order to prevent detector to learn common data recording problems as normal, any segments with values outside statistical error threshold are removed. To identify such readings, we use the z-score (normal score) statistical measure. Normal score measure quantifies reading’s distance to the mean of a dataset. We calculate the normal score of each datapoint and remove datapoints where abs(zscore)>3, with the assumption that such values are erroneous.

As data collection is event based, and number of inputs is fixed, zero-order hold method is used to infer value of other data points when there is a new event from one of the trended data points. This means that last known state of values for all the relevant datapoints is retained apart from ones changed by the latest event. Synthetic datapoint needs to be updated whenever one of the datapoints that is used for its calculation is updated. For example, synthetic datapoints measuring time since last update of the datapoint need to be updated every time there is a change to any other datapoint. Figure 8 shows how values for all datapoints are inferred when new event happens. Datapoint 1 has a new value at timestep 0, 2 and 5, while datapoint2 has new values at timestep 1 and 4. Interval since last received value for datapoint 1 linearly increases every timestep, and resets to 0 when new event involving datapoint 1 occurs.

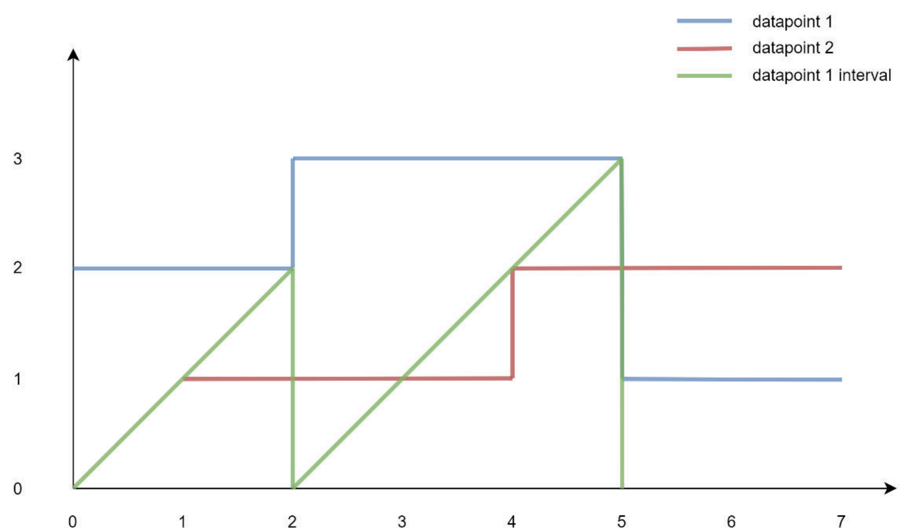

Figure 8 *- Calculation of synthetic datapoints in relation to behavior of primary datapoints; datapoint 1 interval changes every time a change occurs for any other datapoint. When event is received for the source datapoint, interval datapoint is reset to zero*

All the data points with little or no changes for the whole data collection period are ignored.

Data is split in distinct sets for training and validation in relation. 80% of the dataset is used to train the model and 20% is used for testing. Scaling parameters for scaling each time series to [0,1] range are calculated and applied to training and testing dataset to prevent any of the datapoints having outsized influence on the training process.

Once data is prepared, autoencoder models are created and trained. Resulting anomaly detectors are tested on a complete dataset, results are saved and stored for analysis.

3.3 Autoencoder network architecture

Network architecture is based on the classical symmetrical autoencoder concept as shown in Figure 9.

https://doi.org/10.52202/077185-0060

Incoming data is fed into the network on the input layer Le_1 (encoder layer 1). Size of the input layer corresponds to the number of primary, external and synthetic datapoints. Input layer is followed by an n-1-layer deep series of densely connected layers (Le_2-Le_n), ending with a encoded representation layer (Lc). This is followed by a series of decoder layers (Ld_n-Ld_1), with a final one providing outputs of the network. Outputs represent the reconstruction of the autoencoder's inputs. ReLu[2] is used as an activation function for all the layers. All tensors are initialized with a random normal function which is a part of the Keras library (Chollet et al., 2015) that provides a high-level API for setting up deep learning workflows. Biases are initialized with zeros.

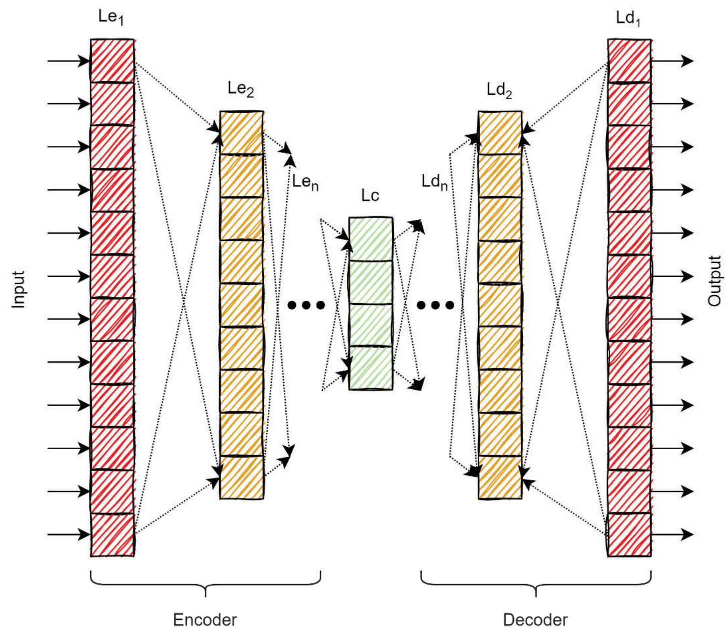

Figure 9 *- Schematic representation of symmetric autoencoder architecture with variable number of internal layers*

3.4 Hyperparameter optimization criteria

We set up a hyperparameter tuning process that optimizes hyperparameters. We consider the following hyperparameters: number of internal layers, learning rate, and factor for L2 regularization. Number of internal layers was limited to a set [1,3,5,7] (including the encoding layer). Learning rate can take values from the set [0.01, 0.001, 0.0001], while L2 factor is selected from the set [0.01, 0.001, 0.0001]. Hyperparameter optimization is performed using a hyperband tuner (Li *et al.*, 2018), a part of a Keras library.

3.5 Grouping of anomalies

Anomalies can occur as point anomalies or groups of related anomalous readings. We introduce a notion of anomaly group where one group can contain one or more anomalous points, and any anomaly point within a group is not farther than a chosen time delta from closest anomaly point within the group. We select 2 hours as our time delta. Grouping is especially important for views meant to be presented to users for feedback.

[2] Rectified linear unit

https://doi.org/10.52202/077185-0060

4 RESULTS

Table 2 shows best reconstruction errors and respective model hyperparameters for each combination of system and lag used. Interestingly, all of the selected models have only one internal layer, that is in fact the encoding layer. L2 factor of 0.001 is selected in models with smaller input sizes (O1HK_V01 and O1HK_V02). In all the other models L2 factor of 0.0001 yielded best results. Larger learning rate of 0.01 seem to be preferred by models of smaller encoding sizes, with larger encoding sizes favoring smaller learning rates of 0.001. Findings from the hyperparameter space optimization will be used if further research.

Table 2 *- Reconstruction error threshold and properties of best models of each category (system, lags)*

System	Lags	Input size	Encoding size	Internal layers	Learning rate	L2 factor	Threshold
O1HK_V01	0	13	5	1	0.01	0.001	0.18
O1HK_V01	1	17	5	1	0.01	0.001	0.25
O1HK_V02	0	13	4	1	0.01	0.001	0.23
O1HK_V02	1	17	5	1	0.01	0.001	0.26
O1HK_V03	0	17	7	1	0.01	0.0001	0.22
O1HK_V03	1	23	10	1	0.01	0.0001	0.21
O1HK_V04	0	29	12	1	0.001	0.0001	0.12
O1HK_V04	1	41	16	1	0.001	0.0001	0.15
O1HK_V06	0	19	7	1	0.01	0.0001	0.17
O1HK_V06	1	26	10	1	0.01	0.0001	0.13
O1HK_V07	0	21	9	1	0.001	0.0001	0.13
O1HK_V07	1	29	10	1	0.01	0.0001	0.12
O1HK_V10	0	19	8	1	0.001	0.0001	0.12
O1HK_V10	1	26	10	1	0.01	0.0001	0.12
O1HK_V11	0	21	7	1	0.001	0.0001	0.12
O1HK_V11	1	29	13	1	0.001	0.0001	0.16
O1HK_V12	0	19	7	1	0.001	0.0001	0.15
O1HK_V12	1	26	8	1	0.001	0.0001	0.13
O1HK_V13	0	21	8	1	0.001	0.0001	0.12
O1HK_V13	1	29	13	1	0.001	0.0001	0.12
O1HK_V14	0	21	9	1	0.001	0.0001	0.12
O1HK_V14	1	29	11	1	0.001	0.0001	0.12

Encoding dimension is the most important model parameter and influences the data reconstruction ability of the model. Threshold value from table is calculated as mean reconstruction error which results in 10 groups of anomalies (with groups defined as in section 3.5).

In more than half examples (14 out of 22), best reconstruction threshold was not reached by the model with largest encoding layer as would be expected. We calculated encoding size to input size ratio, and we plot thresholds of all trained models in Figure 10. Different systems are represented on the x-axis. Circle sizes indicate thresholds (smaller is better), while color indicates whether respective model input includes datapoint lags. Top of the plot (encoding to input ratio is higher) represents models where with looser compression, and expected better reconstruction, while the bottom of the plot shows models with more constrained representation, and consequently worse reconstruction.

https://doi.org/10.52202/077185-0060

5 DISCUSSION

We have shown that using semantic model of the facility helps with setting up training process by simplifying selection of relevant datapoints for each anomaly detector in a systematic way. When another approach for variable selection is devised, it is simple to reuse it on other subsystems, systems, zones or even buildings, as long as identical or similar modelling approach is used.

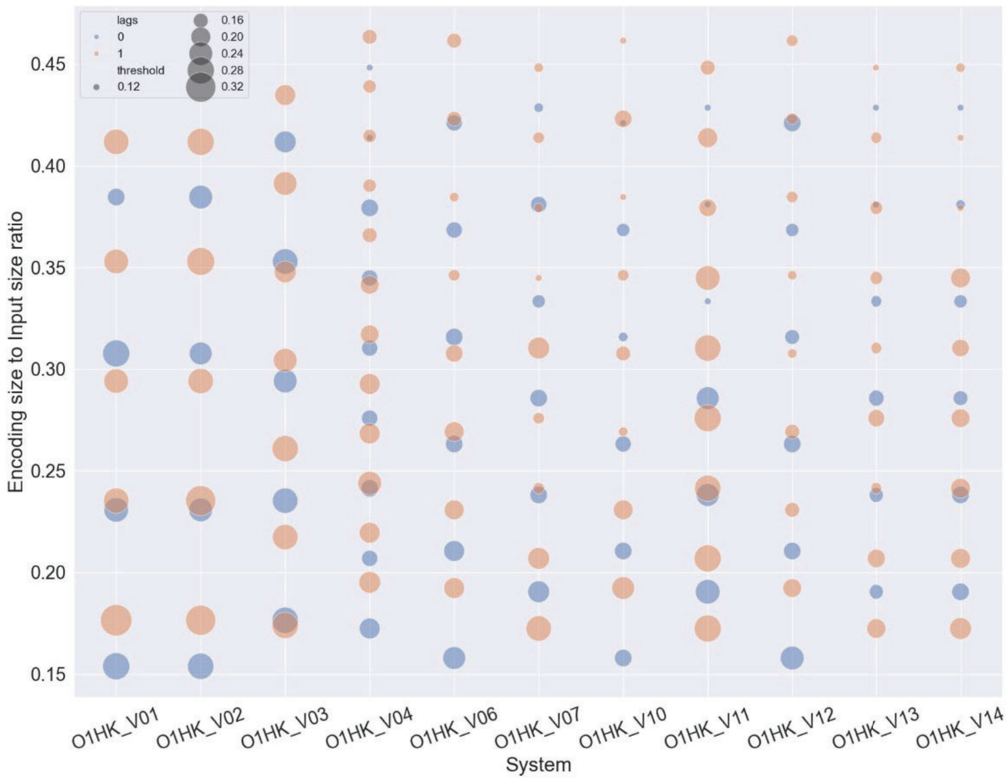

***Figure 10** - Reconstruction error thresholds resulting in 10 detected anomaly groups. Highest encoding to input size ratio does not always result in a best error threshold.*

Figure 11 shows snapshot from a dashboard created to explore trained models and their performance on the dataset. General workflow consists of selecting the desired system, after which the model dropdown box gets populated with a list of models trained for that system. After the model is selected, the whole dataset is evaluated using this model. Additionally, evaluation is saved for future use. Then system calculates the threshold required to detect 10 anomaly groups, and populates the rightmost dropdown box with a list of anomaly groups. Upon clicking on one of the groups, it gets visualized in the plot below. Plot is also interactive, so it is possible to turn on and off visualization for individual datapoints, as well as navigate the plot. Threshold can also be manually changed which causes anomaly group recalculation and update in the corresponding dropdown box.

Main deficiency of the proposed approach is requirement for the existence of high-quality data and contextual metadata. Furthermore, in case of facility changes, care must be taken to detect changes relevant for individual anomaly detectors and retrain them. Changes may be related to structure (adding, removing or rearranging equipment), or settings (setpoints, control algorithms). Depending on the scope of changes, it may be necessary to retrain the relevant models. In case of adding new datapoints, anomaly detectors affected by these datapoints may need to be retrained. To properly take new datapoints into consideration, a sufficient amount of data needs to be collected first.

https://doi.org/10.52202/077185-0060

Next step for the development is including feedback from facility manager about quality of detected anomalies. The desired feedback includes list and characterization of false positives and false negatives. To that end, to prevent future false negatives, either connecting directly to the building incident book or otherwise integrating the information about system failures would help when retraining the anomaly

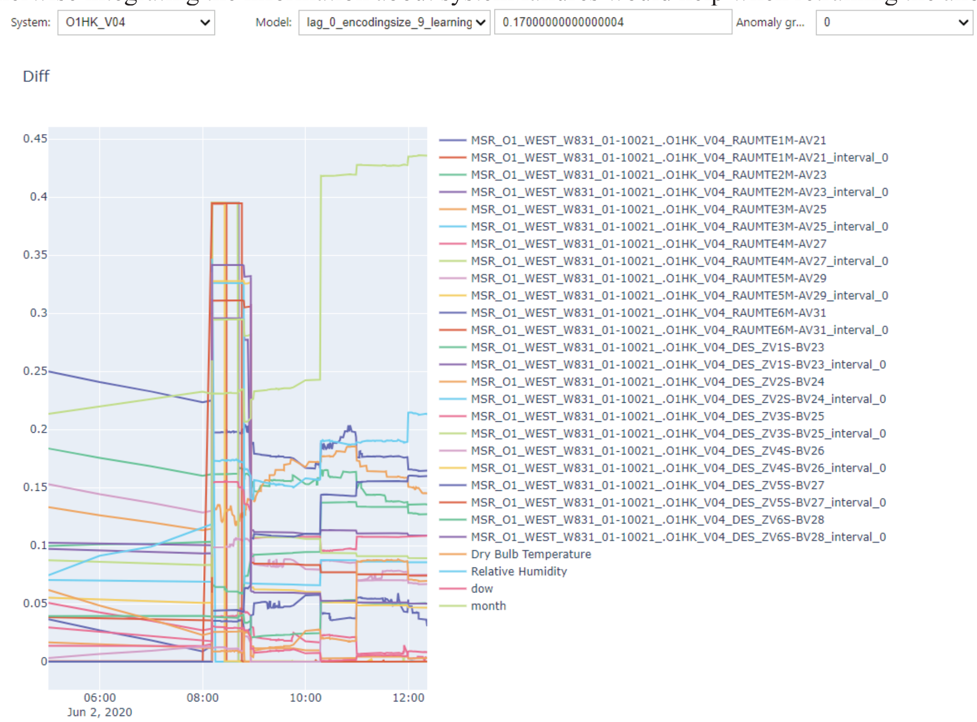

Figure 11 *- Anomaly detection dashboard enabling exploration of anomalies detected with each of the trained autoencoders. Model with encoding size of 9 of system O1HK_V04 is selected*

detectors. To prevent the false positives, either raising the threshold or putting more weight on problematic data segments should be performed to address specific issues. As seen, in some cases the anomaly detector does not reconstruct the input data well, and base threshold remains high. As this affects some systems significantly more than others, the affected systems will be examined closely to identify reasons for weak performance and improve it.

REFERENCES

Araya, D.B. *et al.* (2016) 'Collective contextual anomaly detection framework for smart buildings', in *2016 International Joint Conference on Neural Networks (IJCNN)*. *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 511–518. Available at: https://doi.org/10.1109/IJCNN.2016.7727242.

Assistant, H. (no date) *Home Assistant*, *Home Assistant*. Available at: https://www.home-assistant.io/ (Accessed: 15 March 2024).

Balaji, B. *et al.* (2018) 'Brick : Metadata schema for portable smart building applications', *Applied Energy*, 226, pp. 1273–1292. Available at: https://doi.org/10.1016/j.apenergy.2018.02.091.

Chollet, F. and others (2015) 'Keras'. Available at: https://keras.io.

Daniele, L., den Hartog, F. and Roes, J. (2015) 'Created in Close Interaction with the Industry: The Smart Appliances REFerence (SAREF) Ontology', in R. Cuel and R. Young (eds) *Formal Ontologies Meet Industry*. Cham: Springer International Publishing, pp. 100–112. Available at: https://doi.org/10.1007/978-3-319-21545-7_9.

Gaida, S. *et al.* (2018) 'Ontology-Based Optimization of Building Automation Systems', in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*. *IECON 2018 - 44th Annual Conference of the*

https://doi.org/10.52202/077185-0060

IEEE Industrial Electronics Society, Washington, DC: IEEE, pp. 819–825. Available at: https://doi.org/10.1109/IECON.2018.8591703.

Janowicz, K. *et al.* (2021) 'BOT: The building topology ontology of the W3C linked building data group', *Semantic Web*, 12(1), pp. 143–161. Available at: https://doi.org/10.3233/SW-200385.

Li, L. *et al.* (2018) 'Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization', *Journal of Machine Learning Research*, 18(185), pp. 1–52.

Liu, Y. *et al.* (2020) 'Anomaly detection based on machine learning in IoT-based vertical plant wall for indoor climate control', *Building and Environment*, 183, p. 107212. Available at: https://doi.org/10.1016/j.buildenv.2020.107212.

Michelucci, U. (2022) 'An Introduction to Autoencoders'. arXiv. Available at: https://doi.org/10.48550/arXiv.2201.03898.

Pang, G. *et al.* (2021) 'Deep Learning for Anomaly Detection: A Review', *ACM Computing Surveys*, 54(2), pp. 1–38. Available at: https://doi.org/10.1145/3439950.

Pauen, N. *et al.* (2020) 'Integrated representation of building service systems: topology extraction and TUBES ontology', *Bauphysik*, 42(6), pp. 299–305. Available at: https://doi.org/10.1002/bapi.202000027.

Pauwels, P. and Terkaj, W. (2016) 'EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology', *Automation in Construction*, 63, pp. 100–133. Available at: https://doi.org/10.1016/j.autcon.2015.12.003.

Petrushevski, F. *et al.* (2017) *Semantic Building Systems Modeling for Advanced Data Analytics for Energy Efficiency*. Available at: https://doi.org/10.26868/25222708.2017.161.

Poveda-Villalon, M. and Garcıa-Castro, R. (no date) 'Extending the SAREF ontology for building devices and topology'.

Quilitz, B. and Leser, U. (2008) 'Querying Distributed RDF Data Sources with SPARQL', in S. Bechhofer et al. (eds) *The Semantic Web: Research and Applications*. Berlin, Heidelberg: Springer, pp. 524–538. Available at: https://doi.org/10.1007/978-3-540-68234-9_39.

Schneider, G., Pauwels, P. and Steiger, S. (2017) 'Ontology-Based Modeling of Control Logic in Building Automation Systems', *IEEE Transactions on Industrial Informatics*, 13, pp. 3350–3360. Available at: https://doi.org/10.1109/TII.2017.2743221.

Schneider, G.F. (2019) *Semantic Modelling of Control Logic in Automation Systems - Knowledge-Based Support of the Engineering and Operation of Control Logic in Building and Industrial Automation Systems*. Available at: https://doi.org/10.5445/IR/1000092405.

Šipetić, M. *et al.* (2020) 'Design and integration of the project-specific ontology for data analytics support', in *Proceedings of the 8th Linked Data in Architecture and Construction Workshop*. *Linked data in Architecture and Construction 2020*, Dublin, Ireland (virtually hosted): ceur-ws, pp. 50–63.

Sun, Y. *et al.* (2016) 'Learning a good representation with unsymmetrical auto-encoder', *Neural Computing and Applications*, 27(5), pp. 1361–1367. Available at: https://doi.org/10.1007/s00521-015-1939-3.

Tziolas, T. *et al.* (2022) 'Autoencoders for Anomaly Detection in an Industrial Multivariate Time Series Dataset', in *ITISE 2022*. *ITISE 2022*, MDPI, p. 23. Available at: https://doi.org/10.3390/engproc2022018023.

ACKNOWLEDGEMENT

The work presented in this paper is part of the mAIntenance project (FFG Project no. FO999886903) funded under the program 'Stadt der Zukunft'.

https://doi.org/10.52202/077185-0060