Anomaly detection in district operation systems

Gerrit Bode, Ansas Bogdan, Marc Baranski, Dirk Müller
RWTH Aachen University, E.ON Energy Research Center,
Institute for Energy Efficient Buildings and Indoor Climate,
Mathieustraße 30, 52072 Aachen, Germany

Abstract

Modern buildings, as well as sites and even city districts, accumulate large amounts of data collected by monitoring and energy management systems. With the integration these data systems into more and more buildings and districts, the amount of data available to the operators of these districts increases dramatically. At a certain point, the check the plausibility of every single measurement and set point collected in the system must be supported by automation. In this paper, we present the application of an anomaly detection algorithm in a real-life district operation scenario. The algorithm is trained with synthetic sensor data to create representations of time series data. These representations are collected for every sensor during normal operations. If a newly calculated representation is too far from the existing data, an alarm is raised. We applied the algorithm to data collected in a district with multiple building types and energy systems in Munich. A failure event in a meeting room in one of the buildings is more closely investigated, and the algorithm sucessfully detects the problem. The developed system is able to be deployed in the application scenario with little configuration effort and provides valuable insights to the maintainable teams.

Introduction

Smart buildings and connected devices as well as closely monitored district energy systems produce a large amount of data every second. With new technologies like the Internet-of-Things (IoT), the amount of data will likely continue to increase (Bode et al., 2019).

Operators tasked with the supervision of these district systems may, at some point, become overwhelmed with the task of sighting, checking and evaluating each of the collected time series. Increasing the staff is, however, seldom economically viable. Nonetheless, close monitoring of these systems is important. Studies have reported that up to 50 percent of the energy consumed in the building sector is due to unwanted or erroneous behaviour of the energy systems (Piette et al., 2001; Katipamula and Brambley, 2005a,b; Yu et al., 2014). These high costs are often associated with faults that are not detected until the next regularly scheduled maintenance interval. This can be because the impact on the costumer of the energy service is either unnoticeable to the costumer, or mitigated by other measures. However, this mitigation causes increased financial or ecological cost (Turner et al., 2017).

Automated fault detection and diagnosis systems provide a promising solution to this problem. Within the past years, the amount of proposed systems and algorithms has increased drastically. Among the most used methods are many applications from the field of machine learning. Machine learning is currently regarded as the most viable approach to the problem, because it is specifically designed to work with and benefit from large amounts of data. However, few of these algorithms have transitioned into real-world applications. This is commonly credited to the lack of properly labelled and complete training data for supervised learning processes (Kim et al., 2018; Zhao et al., 2019; Bode et al., 2020).

In supervised machine learning, a set of data with known desired outcomes, i.e. whether a fault is present, is used as a training set for a model. Without this data, many of the algorithms cannot be adapted to real-world use cases, as they are fitted to their respective experimental or simulation-based training data. To circumvent this problem, unsupervised techniques can be used for the training of the algorithm. In unsupervised learning, the algorithm tries to find structures in unstructured data without any further input in form of training labels. A common example are clustering algorithms.

It is not always possible to clearly distinguish between supervised and unsupervised approaches, and both approaches can be combined within a system. For example, Yan et al. (2020) use unsupervised algorithms to enhance their training data for supervised learning to compensate insufficient amount of training data.

Unsupervised learning can be used for fault detection and diagnosis, especially anomaly detection.

Anomaly detection is, at its core, a basic classification problem: normal and not normal (Sommer and Paxson, 2010). By learning from the existing measured data from a system, the models are able classify new measurements from the system and raise an alert if they do not fit into what is considered normal plant behaviour. In other applications and fields, for example in computer science, large-scale applications of machine learning anomaly detection already exist.

For the use in district energy systems, any anomaly detection system should meet the following criteria, so that the system can be applied in the district without exceeding personnel capabilities or computational limits.

- Due to the vast number of data points, the system must be applicable without manual configuration to any data point.
- The training effort per data point must be limited to limit overall resource usage.
- In any real-life system, especially during commissioning, faults and off-normal behaviour will occur. The algorithm must therefore be able to correctly train with training data that encompasses a number of anomalies.
- In order to avoid technicians checking false positive detections, the system should revert back to normal state if an anomaly is not persistent over a longer period of time.

In this paper, we present a real-world application of an unsupervised algorithm for the detection of anomalies in a district energy system. In the following section, we first describe the approach to detecting a time series anomaly. We subsequently describe the supervisory system we implement to apply this detector to the field test district and evaluate the performance using an example anomaly.

Method

The largest computational effort in machine learning applications is the fitting of the model to the training data set. To meet the criteria mentioned in the previous section, we need to avoid having to train a model for each time series in the district. Therefore, we trained an artificial neural network (ANN) to not directly detect anomalies, but rather to compute compresssed representations of the time series, also called embedding. We chose an ANN. because they have been used for similar encoding tasks with great sucess in the past. Then, each time series is analysied and a set of known embeddings is created. If an anomaly occurs, the representation of the time series will change, and no longer fit into the set of known representations. To achieve this, the network is trained to produce an embedding of a time series into the Euclidean space where the similarity between two signals is the distance betweens their embeddings. This is achieved by using the Triplet Loss according to Schroff et al. (2015)

$$L = [\sum_{i}^{\mathrm{N}} \|f_i^a - f_i^p\|^2 - \|f_i^a - f_i^n\|^2 + \alpha] \qquad (1)$$

where f_i^a is the embedding of the anchor, f_i^p is the embedding of a positive sample (similar to the anchor), f_i^n is the embedding of a negative sample (different from the anchor) and α is a margin between positive and negative pairs.

To create the set of known representations, a number of embeddings for the historic data of a sensor is computed with a sliding window. These embeddings are clustered using k-means and the centroids of that clustering are used as the representations for that specific sensor. An anomaly is defined as:

$$\min\{\|e - f\|^2 \mid f \in F\} > t \qquad (2)$$

where e is the embedding of the current signal, F is the set of fingerprints and t is a threshold for determining whether or not two embeddings are similar.

Network Architecture

The network is based on the Multi-Scale Convolutional Neural Network according to Cui et al. (2016) to capture the behaviour of the time series on multiple scales. A graphical representation of the network architecture is depicted in Figure 1.

The network was trained to embed 128 time steps into a 32-dimensional Euclidean space using

- down-sample strides of $[2, 4, 8, 16]$,
- smoothing windows of $[8, 16, 32, 64]$,
- a pooling factor of 4, 256 channels,
- a local filter width of 2,
- a full filter width of 2 and
- a full pool size of 4.

Instead of the sigmoid layer at the end to assign a label to the time series, we used a linear layer followed by 12 normalizations to compute the embedding.

Training

Usually the training of machine learning models requires sufficient data from the application. Models that are trained directly on this data are then specific to it, meaning that they have to be re-trained when transfered to other applications, for example other sensor types. A major advantage of our approach is that the network is not trained to detected the anomaliy, but to calculate a represenation of the time series, which can then be compared to representations considered normal for a specific sensor. There is actually no need to use any actual sensor data during the training. Instead, we simply generated time series with the added benefit that it is comparatively easy to ensure that the anchor and the positive samples are indeed similar while the anchor and the neg-

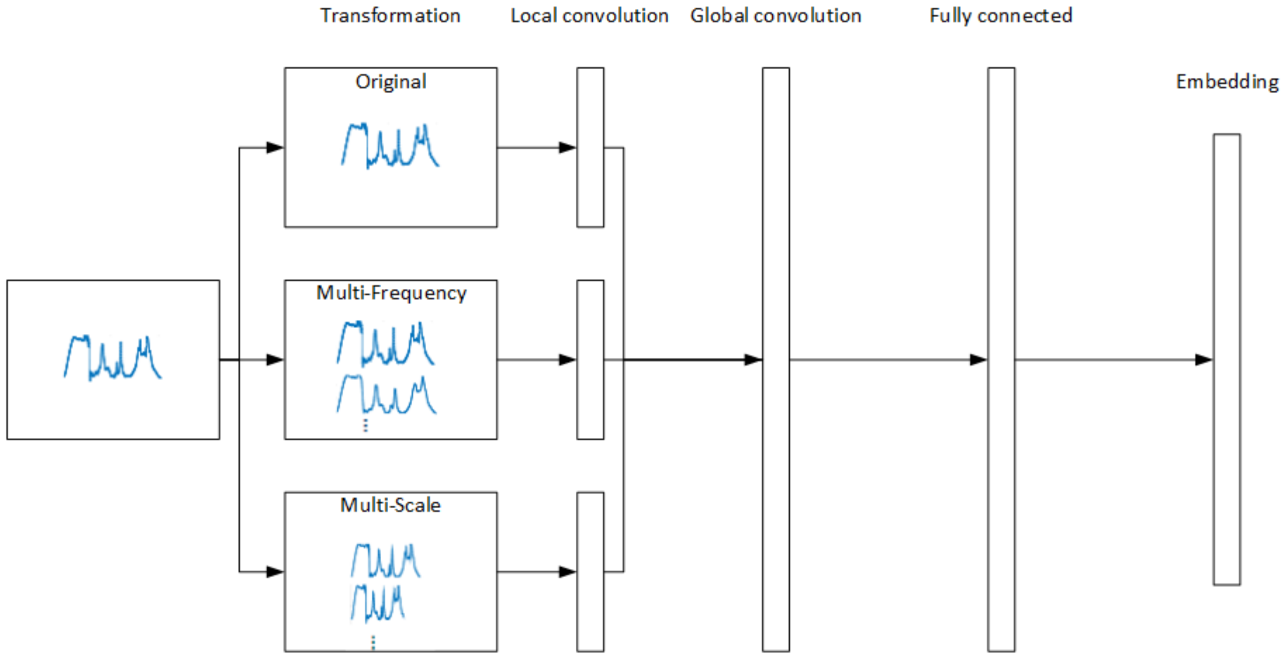

Figure 1: Architecture of MCNN modelled after Cui et al. (2016) to generate embeddings from time series data

ative samples are dissimilar. Generating the training data also ensures that we have a sufficient amount samples to prevent over-fitting. The trained model can then be reused for all desired time series without the need for, computational expensive, re-training.

We performed the training using Triplet Loss (Schroff et al. (2015)) with an alpha value of 0.2 for around 1 hour on a single NVIDIA Tesla V100 GPU, during which the network was shown around 41 million triplets.

Fingerprinting

One important difference between a real-life sensor and the generated training samples is that a sensor may have different behaviour in different situations, e.g. changing weather conditions or switching between heating and cooling mode. Therefore, it is not possible to use just a single embedding, but instead it is necessary to extract embeddings for all normal situations. On the other hand, we want to minimize the number of embeddings, because the larger the set gets, the more difficult comparing new data to it becomes. To this end, we employ k-means clustering to reduce the number of embeddings by creating cluster, which the embeddings fit into. We chose k-means clustering, because both the Triplet Loss and k-means try to minimize the squared Euclidean distance. Algorithm 1 shows how a suitable k is chosen by iteratively increasing k until the error ratio in the historical data is below a chosen threshold. This threshold should be chosen according to the quality of the historical data.

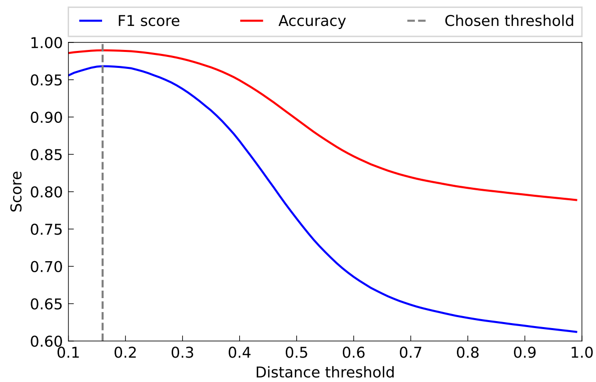

Figure 2: The development of the prediction accuracy and F1 score when the distance threshold setting is varied

Training results

After training, the network was tested against a validation data set to determine an optimal value for the threshold parameter. Figure 2 shows the performance of the alogirhtm expressed by the accuracy and F1-score.

The accuracy for single pairs is at 98.9%. In Figures 3 and 4, we can see the distances between positive pairs (pairs that were selected from the same time series) and negative pairs (pairs that were selected from different time series). We can see that most pairs are quite far below the threshold for the positive pairs, or quite high above the threshold for the negative pairs.

Data: embeddings, threshold, max error
Result: fingerprints
$k \leftarrow 1$
repeat
  $k - means(embeddings, k)$
  $fingerprints \leftarrow cluster\ centers$
  $distances \leftarrow \{min\, \|e - fingerprints\|^2 \mid e \in embeddings\}$
  $error \leftarrow |\{d > threshold \mid d \in distances\}| \div |embeddings|$
  $k \leftarrow k + 1$
until $error \leq max\ error$;

Algorithm 1: Computing the (number of) fingerprints by iteratively increasing k until the amount of anomalies in the historical data is below a chosen value

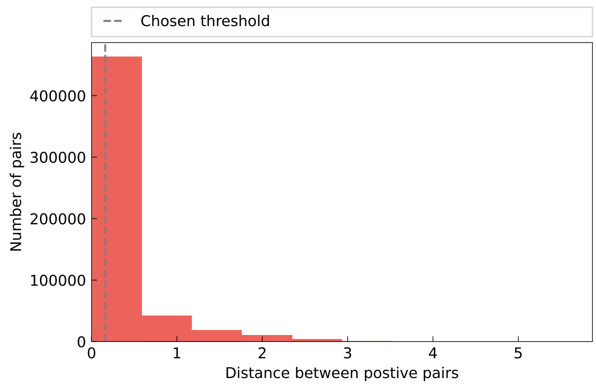

Figure 3: Pairwise distances between postive pairs in relation to the threshold value from Figure 2

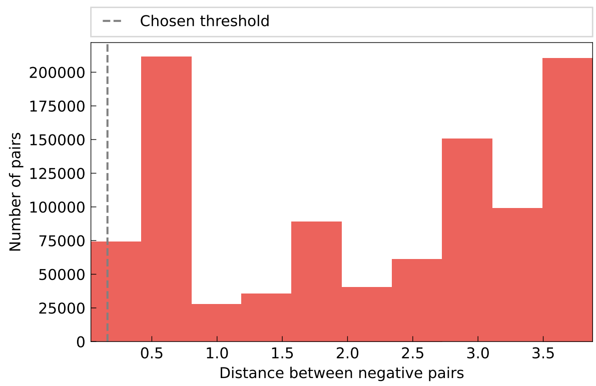

Figure 4: Pairwise distances between negative pairs in relation to the threshold value from Figure 2

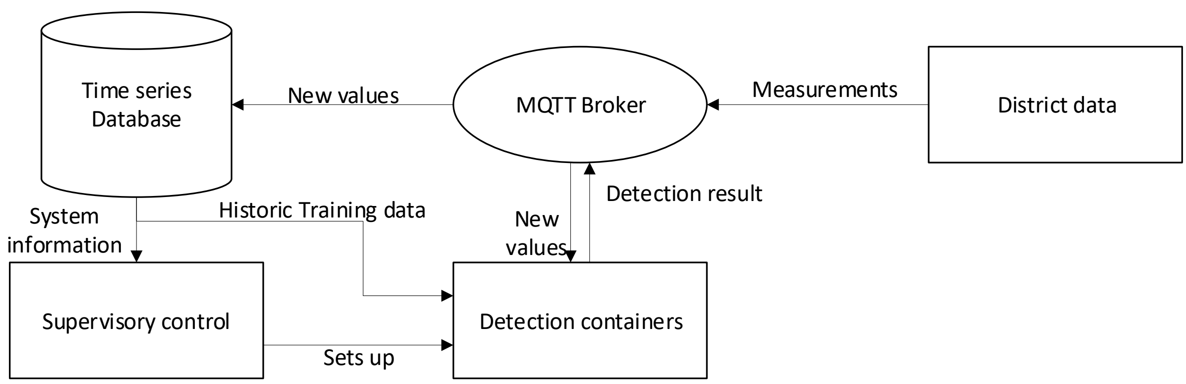

Figure 5: The schematic architecture of the supervisory detection system coordinating the roll-out of the anomaly detectors

Supervisory detection system

In order to roll out the system to large quantities of data points in a system, the deployment of the system has to be automated. For this, we developed an automated roll-out set-up for the anomaly detectors. Figure 5 shows the general structure of the system. In our project, the data is stored in a time series database that is accessible via an HTTP API (see section). The API provides a list of all accessible data points that are logged in the system. Using this list, the detection system can create an instance of the anomaly detector for each data point. For each data point, the extraction process for the embeddings is run with the available historical data. To exclude obvious errors, NaN values are dropped from the time series beforehand. This creates a discontinuity in the data but since the model expects a certain amount of errors, this influences the resulting model performance less then NaN values in the training data. The resulting fingerprints are passed on to the detection process.

The actual anomaly detection process is run inside a docker container for better handling of the needed libraries and dependencies and for use in generic cloud infrastructures. The detection is run for each data point in sequence. If the supervisory system detects that the frequency of anomaly scans for a data point is below the desired threshold frequency, in our case 60 seconds, the list of data points is split and distributed to two (or more) docker containers, so that detection will run in parallel.

The results of the detection process are pushed as an MQTT message back to the central broker and stored in the data base. Since an alarming system is already implemented at monitoring system level, we use it to inform the operator about anomalies. The database also supports averaging functions for time series. This allows us to select a time span over which the detection result should be averaged, and only raise an alarm if this average exceeds the threshold. This greatly reduces the amount of false alarms, as non persistent detections are averaged out (Adetola et al., 2016).

Field test

District description

The application case is a city district in Munich called the Werksviertel. This former industrial district is currently undergoing major reconstruction and remodelling of the old industrial plants and construction of new buildings into a modern multi-use district featuring offices, a housing area, bars, sport centres and musical and concert halls. The district will be supplied with electricity, heating and cooling by local district networks, fed primarily by a local energy hub. Energy provision units include two large combined heat and power plants, a gas boiler, adsorption chillers and cooling towers. Additionally, the buildings will be supplied by various local geothermal heat pumps and solar plants. Within the buildings, a multitude of air handling units, floor heatings, radiators etc. deliver the energy to the occupant. For an optimal operation of the district energy systems, all these systems need to be operated at beneficial states.

Not all electricity can be generated locally, so the district is also connected to the higher grid level of the city of Munich. Transactions like purchasing or selling energy to this grid must be predicted in advance, and divergence from these predictions can become costly. The gas supply is a limiting factor in the amount of heat that can be provided to the buildings. This means that the company running the energy hub needs to be aware of the state of the whole district at any time and needs to ensure that this information is correct.

The data flow and system architecture, to which the supervisory detection system will be attached, is depicted in figure 6. The information of the district is collected from each building and the energy hub using local logging devices. The logging devices are produced by several different manufactures, and the data is stored in different data bases. This data is either sent to the cloud via the MQTT protocol or queried via APIs and stored in a central time series data base. The time series and all meta information can be accessed via an HTTP API (Brümmendorf et al., 2019), that is used for the detection system in this paper.

Given the enourmous amount of data points and data collected each second, it becomes vital that this information can be processed in an automated manner. For this, the described detection system is deployed, and interaction is then limited to alerts rasied when an anomaliy is detected. In the results, we focus on two occurrences of anomalies within the district, that were subsequently further investigated. The detection system allowed to pinpoint the problem from the district perspective down to a much smaller system. For our field test, we trained the anomaly detectors with ninety days of collected data.

Field test results

First, we investigate an anomaliy of the air handling unit supplying a meeting room by taking a look at the CO_2 concentration measured within the room. Figure 7 shows the CO_2 concentration for the investigated day. When the first meeting in the room took place, the CO_2 concentration drastically increased, after which it is lowered again by opening the windows when the low air quality was noticed by the occupants. The results of the detection algorithm are also shown. As desired, the algorithm detects a high probability of an anomaly over the course of the day, and returns to normal once the problem is no longer present, i.e. the CO_2 concentration is back to normal.

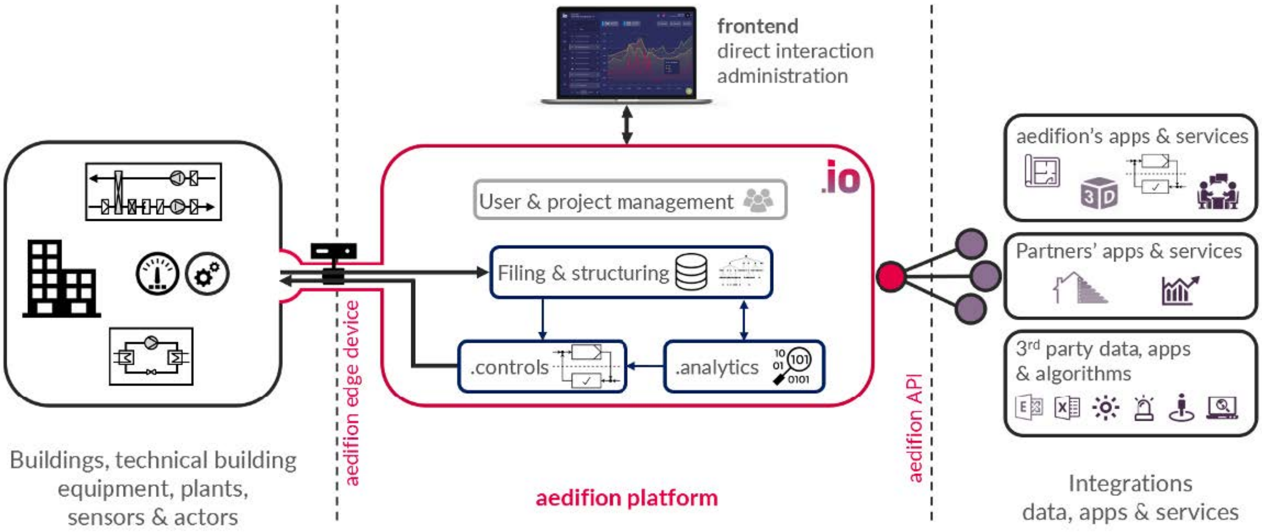

Figure 6: The architecture of the monitoring platform that is used in the field test according to Brümmendorf et al. (2019)

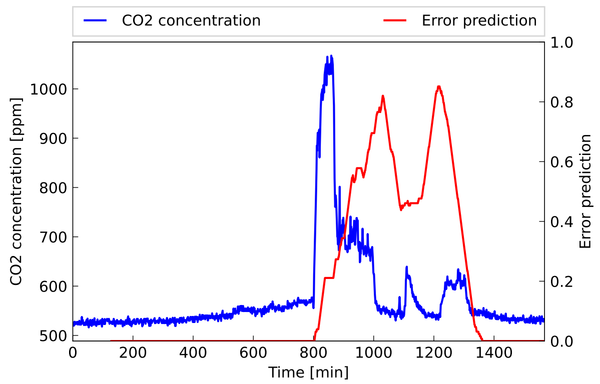

Figure 7: CO_2 concentration in a meeting room during a fault in the ventilation system and the corresponding output of the anomaly detector.

The second investigated anomaly is a malfunction of a meeting room control panel, in which the panel's temperature sensor was accidentally disconnected. Due to a questionable design decision in the programming of the panel, this led to overriding the measurement data channel with the desired temperature rather than raising an error. Measured and desired temperature values were then the same, so temperature control of the room was not activated, causing the room's temperature to reach undesired values.

The chain of reporting the error in the district is very long, starting with the tenant reporting to the building manager, who reports it on to the technical staff, who then need to investigate the source and consult manufacturers for futher informations. With the time needed required interactions and the manual investigation of data and equipment, the location of the fault took several weeks after its occurrence.

Figures 8-11 show the results for four selected time series. In figure 8, the sensor in question is shown. The anomaly detector is able to properly detect the problem with the signal when it occurs. It also detects abnormal behaviour in the day leading up to the fault.

Likewise, all other signals show some sort of anomalous values in the time before the fault. Figure 9 presents the control output of the panel. As expected, when the sensor was disconnected and the measured value was overridden with the set point, the control action vanishes immediately. This is, however, not marked as an anomaly, because this is also the behaviour if the room control is intentionally switched off.

Figures 10 and 11 show the calculated temperature set point for the air temperature and the opening of the exhaust air duct damper. Both time series show anomalies in the pre-fault time, but return to normal after the fault occurs. For the maintenance personnel, this provides valuable insight, i.e. that these systems functions properly and the search for the fault should be conducted elsewhere.

In this example, the anomaly detection system was able to swiftly and clearly point to the problem in the meeting room, likely even before the occupants noticed problems with the air temperature.

Unfortunately, it was not possible to determine the exact cause of the sensor disconnection. Given the anomalous behaviour of both the sensor values and the damper control, it could, however, point to some work being carried out on either the panel itself or the ventilation system. This could, in theory, be matched with staff work records, but these are not available due to tenant contracts.

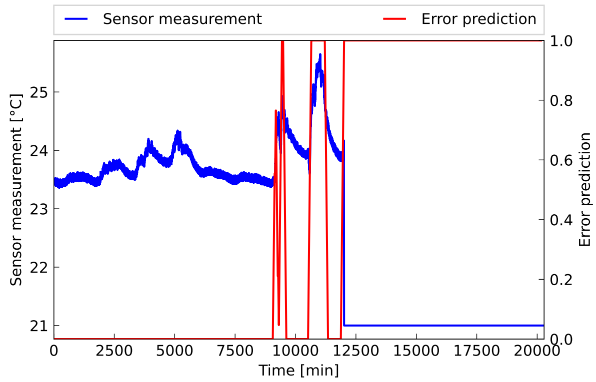

Figure 8: The measurement of the sensor in the room.

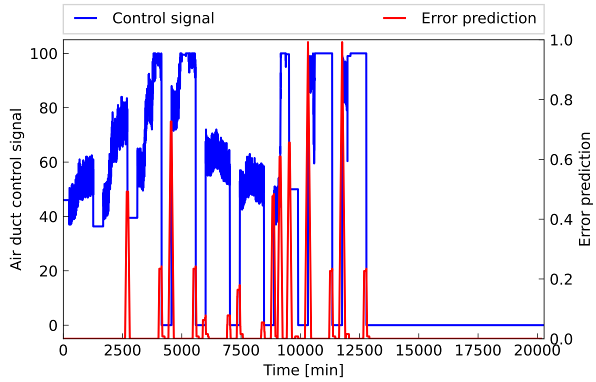

Figure 9: The control signal for the air inflow into the room as calculated by the controller.

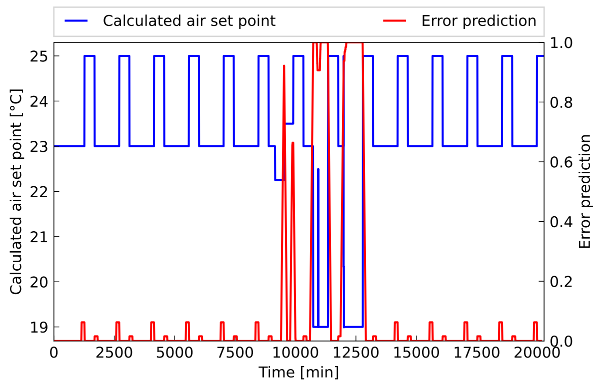

Figure 10: The measurement of the air inflow temperature set point as calculated by the control system.

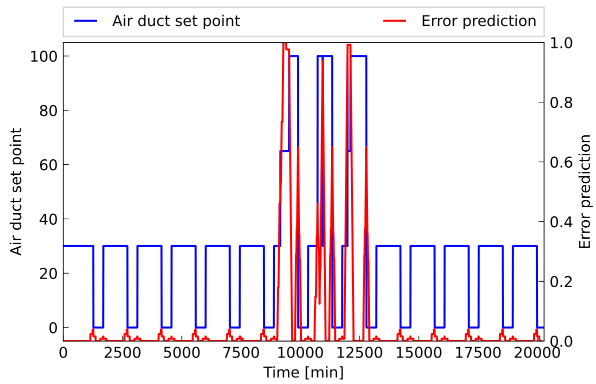

Figure 11: The actual set point response of the opening of the air flow duct damper.

Conclusion

In this paper, we described an anomaly detection algorithm and applied it to a detection system set-up that is suitable for application in building and district energy systems.

The supervisory system we describe is able to apply the anomaly detection algorithm to all data points in the district, without the need for further manual interaction or configuration. We investigated two examples for anomalies that were present in the available data. In both cases, the system detected the anomalous behaviour as desired. With the system, the operators of the district would have been able to react swiftly to the fault and reduce the impact on the occupant as a costumer and reduce the time spent in search of the fault's source.

Initial pre-training of the models with synthetic data severely reduces the training effort and time needed to adapt the models to a certain data point. However, the initial extraction of the embeddings remains computationally expensive and should be conducted on larger cloud resources. The trained model is suitable to run on local machines or clouds.

Since the extraction process for the embeddings can be configured beforehand to respect a certain percentage of allowed deviation, this allows considering historic time series that are not fault-free for the training process. The required amount of training data is comparatively low, because many characteristics are already captured in the pre-trained model and do not need to be captured be the extraction process. However, any behaviour deviating from the training data will be considered anomalous, even if it represents correct plant behaviour, so a minimum amount of data should encompass all relevant operation strategies and settings.

The system is limited by its inability to react to planned changes like refurbishment. Further, it cannot be used during commissioning, as a minimum amount of data is required. As the system expects a certain amount of faults in the training data, it cannot distinguish betweens faults and rare, but normal events in this data.

Further research should further consider reducing the effort for the extraction of the embeddings. This could be done by grouping similar time series or investigating the possibility of using embeddings from a typical sensors for all sensors of that type. There is also a need to consider and investigate correlations between sensors as anomalies are contextual, i.e. a value itself might not be problematic, but abnormal with regard to other values of the system (Hundman et al., 2018). For example, automatically found connections between time series and model representations extracted from meta data could be included according to (Stinner et al., 2019).

Acknowledgments

We gratefully acknowledge the financial support provided by the BMWi (Federal Ministry for Economic Affairs and Energy), promotional reference 03SBE006A. We would further like to thank our project partners werkkraft GmbH.

References

Adetola, V., S. Bengea, K. Kang, A. Kelman, F. Leonardi, P. Li, T. Lovett, S. Sarkar, and S. Vichik (2016). Model Predictive Control and Fault Detection and Diagnostics of a Building Heating, Ventilation, and Air Conditioning System. In *3rd International High Performance Buildings Conference at Purdue 2014*. Red Hook, NY: Curran Associates Inc.

Bode, G., M. Baranski, M. Schraven, A. Kümpel, T. Storek, M. Nürenberg, D. Müller, A. Rothe, J. H. Ziegeldorf, J. Fütterer, and B. Scheuffele (2019). Cloud, wireless technology, internet of things: the next generation of building automation systems? *Journal of Physics: Conference Series 1343*, 012059.

Bode, G., S. Thul, M. Baranski, and D. Müller (2020). Real-world application of machine-learning-based fault detection trained with experimental data. *Energy*, 117323.

Brümmendorf, E., J. H. Ziegeldorf, and J. P. Fütterer (2019). IoT platform and infrastructure for data-driven optimization and control of building energy system operation. *Journal of Physics: Conference Series 1343*, 012040.

Cui, Z., W. Chen, and Y. Chen (2016). Multi-Scale Convolutional Neural Networks for Time Series Classification.

Hundman, K., V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom (2018). Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding. In Y. Guo and F. Farooq (Eds), *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD '18*, New York, New York, USA, pp. 387–395. ACM Press.

Katipamula, S. and M. Brambley (2005a). Review Article: Methods for Fault Detection, Diagnostics, and Prognostics for Building Systems—A Review, Part I. *HVAC&R Research 11*(1), 3–25.

Katipamula, S. and M. Brambley (2005b). Review Article: Methods for Fault Detection, Diagnostics, and Prognostics for Building Systems—A Review, Part II. *HVAC&R Research 11*(2), 169–187.

Kim, J., S. Schiavon, and G. Brager (2018). Personal comfort models – A new paradigm in thermal comfort for occupant-centric environmental control. *Building and Environment 132*, 114–124.

Piette, M. A., S. K. Kinney, and P. Haves (2001). Analysis of an information monitoring and diagnostic system to improve building operations. *Energy and Buildings 33*(8), 783–791.

Schroff, F., D. Kalenichenko, and J. Philbin (2015). FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823. IEEE.

Sommer, R. and V. Paxson (2010). Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. In *2010 IEEE Symposium on Security and Privacy*, Los Alamitos, Calif., pp. 305–316. IEEE Computer Society.

Stinner, F., Y. Yang, T. Schreiber, G. Bode, M. Baranski, and D. Müller (2019). Generating Generic Data Sets for Machine Learning Applications in Building Services Using Standardized Time Series Data. In M. Al-Hussein (Ed), *Proceedings of the 36th International Symposium on Automation and Robotics in Construction (ISARC)*, Proceedings of the International Symposium on Automation and Robotics in Construction (IAARC). International Association for Automation and Robotics in Construction (IAARC).

Turner, W., A. Staino, and B. Basu (2017). Residential HVAC fault detection using a system identification approach. *Energy and Buildings 151*, 1–17.

Yan, K., J. Huang, W. Shen, and Z. Ji (2020). Unsupervised learning for fault detection and diagnosis of air handling units. *Energy and Buildings 210*, 109689.

Yu, Y., D. Woradechjumroen, and D. Yu (2014). A review of fault detection and diagnosis methodologies on air-handling units. *Energy and Buildings 82*, 550–562.

Zhao, Y., T. Li, X. Zhang, and C. Zhang (2019). Artificial intelligence-based fault detection and diagnosis methods for building energy systems: Advantages, challenges and the future. *Renewable and Sustainable Energy Reviews 109*, 85–101.