DENSITY-BASED CLUSTERING ALGORITHM FOR FAULT DETECTION AND IDENTIFICATION IN HVAC SYSTEMS

Gesa Angelika Benndorf, Nicolas Réhault

Fraunhofer Institute for Solar Energy Systems ISE, Freiburg, Germany

ABSTRACT

The operation of building services like heating, ventilation and air conditioning systems (HVAC systems) is often vitiated by faults and suboptimal states. Such malfunctioning can be overcome by introducing a monitoring system with automated fault detection and diagnosis based on measurement data.

Here, we propose a method for the automatic detection and identification of faults in HVAC systems, which is based on a clustering algorithm. We illustrate the proposed method using simulation data from a simple model, where faults have been implemented artificially, and show that the approach performs well with respect to fault detection and that it provides additional valuable information to enable fault diagnostics.

Gebäudetechnische Systeme wie Heizungs- und Lüftungsanlagen unterliegen oftmals fehlerhaftem oder suboptimalem Verhalten. Derartige fehlerhafte Betriebsweisen können mit Hilfe eines Monitoringsystems und einer auf Messdaten basierenden automatisierten Fehlererkennung und -diagnose behoben werden.

Im Folgenden wird eine Methode zur automatisierten Fehlererkennung und -identifikation vorgestellt, die auf einem Clustering-Algorithmus basiert. Wir illustrieren die Verwendung der beschriebenen Methode an Hand von Simulationsdaten eines einfachen Anlagenmodells, wobei die Simulationsdaten künstlich implementierte Fehler enthalten. Wir zeigen, dass der Ansatz sich gut zur Fehlererkennung eignet und dass er zudem wertvolle Informationen für eine Fehlerdiagnose liefert.

INTRODUCTION

Buildings are known to be responsible for about 40% of the energy consumption in developed countries, where approximately half of the energy is consumed in heating, ventilation and air conditioning (HVAC) (Pérez-Lombard et al., 2008). In addition, recent studies show that 20-30% of energy can be saved by adjusting HVAC facilities (Bruton et al., 2013; Ginestet et al., 2008). Therefore, increasing attention has to be dedicated to the energy-efficient operation of buildings to tackle the huge saving potential remaining in it. Often, these adjustments do not imply an extensive optimization of the systems controls, but comprise rather simple modifications, like e.g. changing set values or implementing correct set-backs. In many cases, it even suffices to detect and correct improper behavior, which does not correspond to the foreseen operation of the building system, to save a significant amount of energy and respective costs.

In order to be able to detect faults and inefficiencies in HVAC systems automatically and thus provide a cost-efficient and continuous energy monitoring, adequate routines for fault detection and diagnosis (FDD) that analyze the data accumulated in Building Automation Systems (BAS) have to be developed. Work carried out within the framework of Annex 25 and Annex 34 significantly fostered this development (Hyvärinen and Kärki, 1996; Dexter and Pakanen, 2001) and recent reviews aimed at providing an overview of existing methods and their characteristics (Katipamula and Brambley, 2005; Bruton et al., 2013).

These studies reveal that there is a discrepancy between rule-based expert systems, like the so-called APAR for air handling units (Schein et al., 2006; Castro and Vaezi-Nejad, 2005), and methods from machine learning: while expert systems use to perform well in both, fault detection and diagnosis tasks, available methods from machine learning, like regression models, clustering methods, support vector machines or artificial neural networks are typically relatively easy to set up for fault detection tasks, but are hardly applicable for diagnosis. As, on the other hand, expert systems rely on detailed information and knowledge about every specific building and its HVAC system, the usage of pure expert systems for FDD can be time-consuming and less practicable whith growing system complexity. Therefore, increasing attention is being payed on supervised learning techniques, which need little expert knowledge and are therefore more widely applicable. The challenge is, to make well-known methods for data mining and classification suitable, not only for detecting, but also for diagnosing faults in HVAC systems.

In this paper, we use a density-based clustering algorithm, which is a well-known method for data mining, for fault detection in HVAC systems and demonstrate, how this method can be extended to be capable of identifying possible reasons for a fault, without explicitly training the faults. In the training phase, clusters are defined such that all fault-free data points belong to clusters. In the application phase, all data points, which do not fall into the previously defined clusters, are assumed to be faulty. Based on the analysis of the perturbations of faulty data points, the method allows for each detected faulty data point to identify which of the input variables is most likely to be responsible for the fault and thus to provide a diagnostic. We illustrate the proposed FDD method using simulation data from a simple model.

In the following section, we describe the method in detail. Then, we describe the application of the method, where we first refer to the used simulation model and the implemented faults and afterwards present and discuss the results of the FDD method for the simulation model. In the last section, we conclude and give some suggestions and comments for future work.

METHOD

The basic idea of the fault detection method, described in the following, is, in a first step, to distribute a set of non-faulty data points into clusters. This implies that "similar" data points from this set (e.g. data points corresponding to the same control regime or data points corresponding to the same temperature range) are likely to belong to the same cluster. The first step is called the *training phase*. The number of initial clusters depends on the data structure and on the chosen cluster parameters. Then, in a second step, new data points are assigned to the previously defined clusters if their attributes are "similar" to the data points already belonging to that cluster. If a new data point does not fit into any of the existing clusters, it is called an *outlier* and marked as faulty. The notion of similarity depends on the chosen clustering algorithm and the respective clustering parameters and will be discussed later. This second step is called the prediction phase or *application phase*.

Clustering algorithm

There exist a number of different clustering algorithms, extensively reported in the literature (Hartigan and Wong, 1979; Ester et al., 1996; Sibson, 1973; Steinbach et al., 2000). The three most well-known types of clustering are based on hierarchical algorithms, k-means or density-based clustering. Here, we choose a density-based clustering algorithm, as it was shown to perform best when cluster sizes vary significantly and / or cluster shapes are far from being spheroids (Ester et al., 1996). The density-based clustering algorithm follows, in simple words, two basic rules:

1. *Core points* of a cluster are those points which contain at least n data points within a distance of ϵ.
2. *Border points* of a cluster are those points which have at least one core point within a distance of ϵ.

Core and border points form the clusters. Points, which do not meet the requirements above, do not belong to any cluster. An iterative algorithm (described in Ester et al. (1996)) provides a consistent assignment of each data point according to the above mentioned rules. This algorithm is known as DBSCAN and is here used in form of an implementation in the R-package "fpc" (Hennig, 2014).

For applications, it is essential to choose the clustering parameters n and ϵ adequately. There exist heuristics which determine the parameters for a given data set. However, in our application, we use the clustering parameters as tuning parameters in order to obtain a classification method, which is on one hand widely applicable and on the other hand adjusted to the specific situation. Below, we describe how the clustering parameters are determined, when using the density-based clustering algorithm for fault detection.

Fault detection

Fault detection via clustering simply consists in predicting cluster memberships for new data points, based on the reference data points from the training interval. Data points which belong to clusters are assumed to be fault-free, while data points which do not belong to any cluster (outliers) are supposed to be faulty.

In order to assess the performance of the detection algorithm, common measures are calculated for the application interval: The *true positive rate* tp, also known as *sensitivity*, denotes the fraction of all faulty data points which were classified as faulty. The *false positive rate* fp denotes the fraction of all fault-free data points, which were classified as faulty. The term $1 - fp$, i.e. the fraction of all fault-free data points, which were classified as fault-free, is also known as *specificity*. If the amount of fault-free and faulty events is not equal, as it is typically the case, the *precision* pr of the method can be of major importance. The precision denotes the fraction of all data points, classified as faulty, which are faulty. Finally, the *accuracy* ac is defined as the total number of correctly identified data points (faulty and fault-free) divided by the total number of data points.

Estimation of clustering parameters

In order to find suitable values for n and ϵ with respect to the given data set, we introduce the *test phase*. In the test phase, a data set is used, which includes non-faulty and faulty data points and which is correctly classified. This means, for each data point from the test set it is known, whether it is faulty or not. This information can, for example, be provided with the help of expert knowledge or rule-based approaches.

We start with an arbitrary setting for n and ϵ and apply the density-based clustering algorithm to the fault-free data points in the training interval. Then, the quality of that choice of clustering parameters is evaluated by comparing the respective predictions for the test set with the real classification, i.e. to the true labels of the test data (faulty, non-faulty). Then, the clustering parameters are varied and the procedure is repeated. This procedure gives rise to a set of points in the plane spanned by sensitivity and specificity. This kind of representation of the performance of a classification algorithm comes from signal detection theory and is called ROC curve or ROC analysis, where ROC stands for receiver operating characteristic (Fawcett, 2006). From this set of possible settings, the best clustering parameters with respect to their prediction performance are chosen as the ones which maximize the sum of sensitivity and specificity.

If we keep $n = n_0$ fixed and vary ϵ in a given range $\epsilon \in [\epsilon_{min}, \epsilon_{max}]$ with step size s, the procedure is the following:

1. set $\epsilon = \epsilon_{min}$.
2. while $\epsilon < \epsilon_{max}$:
   - apply clustering algorithm to training interval.
   - predict labels for test interval.
   - compare predicted labels with true classification: calculate and store true positive rate tp and false positiv rate fp.
   - set $\epsilon = \epsilon + s$.
3. choose ϵ such, that $1 - fp + tp$ maximal.

Normalization

All data points are normalized with respect to the respective training interval and test interval. Given a data matrix $\mathbf{X}$, where rows correspond to observations and columns to variables, each entry x_{ij} is normalized in the following way:

$$\overline{x_{ij}} = \frac{x_{ij} - \mu_j}{\sigma_j}. \tag{1}$$

Here, μ_j is the sample mean of variable j in the training interval and test interval and σ_j is the respective standard deviation. If the standard deviation in the respective interval is zero, σ_j is set to some small, non-zero value (here: 0.1) in order to avoid singularities.

The reason for this choice of normalization is to adjust all variables at a common relative range. Thus, each data point is characterized by its distance, in standard deviations, to the sample mean of the training and test intervals. This measure is also known as *z-score* in the literature.

Fault responsibility

As a first step towards fault diagnosis, the clustering method allows to identify the variable or the set of variables which is most likely to be responsible for the detected fault. This is done in the following way: Given a data matrix $\mathbf{X}$, where rows $\{x_1, \cdots, x_l\}$ correspond to the training phase, rows $\{x_{l+1}, \cdots, x_m\}$ correspond to the test phase and rows $\{x_{m+1}, \cdots, x_n\}$ correspond to the application phase. We consider a data point x_k, with $m < k \leq n$, which was labeled as faulty by the clustering algorithm. In order to analyze which variable (or which set of variables) is most likely to be responsible for the fault, we take a random sample of d data points from the first l rows of the data matrix, i.e. from the training phase. Then, we replace the components of the faulty data point x_k by components of the d fault-free data points and check for each of the - in this way obtained - new data points $\tilde{x}_k$, whether it falls into any of the clusters.

The replacement can either concern single variables or sets of variables. In the first case, each component of x_k is replaced individually by the respective component from the d sample data points. In this way, faults can be identified, which are related to single variables corresponding to single sensors or actuators. In the second case, several components of x_k are replaced simultaneously by the respective components from the d sample data points. In this case, faults can be identified, where several variables are involved.

For each variable or set of variables, the number of times, where the replacement was "successful", i.e. where the new data point $\tilde{x}_k$ falls into one of the clusters, is related to the total number of replacements, d. This fraction, we call the *responsibility* r. If the responsibility is close to one, the respective variable or set of variables is likely to be responsible for the fault. In other words, if the responsibility r_i of variable i is one or close to one, it means that by only changing variable i, the faulty data point falls with high probability into a cluster and therefore variable i is likely to cause the fault.

Depending on the number of variables (N), the fault identification procedure can take some time if checking all possible combinations of variables, which yield $2^N - 1$ different subsets in total. Therefore, if available, additional information about dependencies between variables can be used in order to reduce the number of subsets which are to be checked and thus reduce the computational time. Alternatively, as our goal is to rely on as little expert knowledge as possible, one can simply introduce an upper limit for the size of the considered subsets. This makes sense, because most of the typical faults do not affect more than 5-8 sensors. Another way to speed-up the fault identification procedure is to reduce the number of samples d or to reduce the number of data points, which are actually checked for responsibility: instead of checking each data point in a consecutive sequence of, say, 100 detected faults, one would e.g. rather check only each tenth data point.

The fault identification or responsibility analysis allows to assess,

1. the trigger for a detected fault.
2. whether a detected fault is likely to be "false positive" (For examples, when outside air temperature shows maximal responsibility, this should be taken as a hint to check whether the original operation regime, which the algorithm was trained for, still

applies. Otherwise, a new training and testing period should follow.).

3. whether the variables with most responsibility coincide with the variables appearing in rule-based methods or other approaches (e.g. decision trees).

APPLICATION

In this section, we apply the described clustering algorithm to simulation data. The data is obtained from a simple simulation model, involving a room, modeled as a single zone (DIN EN ISO 13790, 2008-09), which is heated by a radiator. The heat is generated by a (fluctuating) heat source with a given mean supply temperature. A sketch of the considered model is shown in Fig. 1. The set mass flow, passed to the valve signal, serves as control for the room temperature and is determined as follows:

$$\dot{m} = \dot{m}_0 / \left(1 + e^{10\cdot(T_{room} - T_{set})}\right). \qquad (2)$$

Here, $\dot{m}_0$ denotes the nominal mass flow rate, T_{room} the actual room temperature and T_{set} the set value for the room temperature.

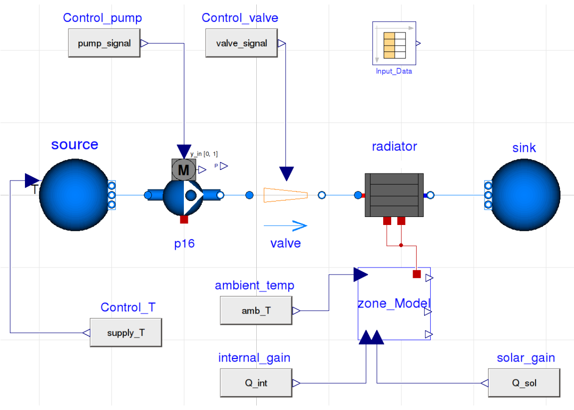

Figure 1: Snapshot of the simulation model (Modelica/Dymola) which was used to test the clustering algorithm. Faults were implemented artificially.

The time interval of the considered simulation data is one hour; the simulation period is 2844 hours (approx. 118 days), starting from first of January.
Analyzed were those model outputs which are likely to be available otherwise from monitoring of a real building. The considered variables are: status signal of the pump (boolean), supply temperature for the water circuit, room temperature, outside air temperature, set value for the room temperature, flow temperature, return temperature, and valve signal (set value for the mass flow rate).

Fault types

In order to test the applicability and performance of the proposed method, we artificially implemented faults into the simulation model. In the following, we analyze and discuss to which extent the simulated faults can be detected and identified using the clustering algorithm. We implemented four different types of faults, resembling faults likely to appear in real building operation:

1. Set temperature too high, namely 24°C, 22°C and 23°C (normal: 20 °C).
2. Temperature sensor for zone temperature disturbed, measuring actual zone temperature +2 K,-1,5 K, +2,1 K and -1 K.
3. Mean supply temperature too low, namely 293±1 K, 295±1 K and 297±1 K (normal: 310 ± 1 K).
4. Pump switched off/broken, i.e. pump status signal is 0.

In total, 360 of 2844 data points were faulty. The occurrences of the different types of faults in the whole time interval are shown in Fig. 2. 120 fault-free data points (5 days) were used as training interval for the clustering algorithm. The test interval consists of the following 120 data points. Training and test intervals are indicated by vertical (dashed) lines in Fig. 2. The performance of the method was evaluated on the remaining 2604 data points (appr. 16 weeks).

Fault detection

The results of the fault detection algorithm are shown in Fig. 3. Green symbols correspond to data points, which belong to clusters, and are thus classified as fault-free, while red symbols correspond to outliers and are thus classified as faulty. Additionally, the size of the red symbol indicates the responsibility of a given variable for the respective fault.
The ROC analysis in the test phase yields $\epsilon = 1.4$, when choosing $n = 3, \epsilon_{min} = 0.2, \epsilon_{max} = 10, s = 0.2$.
The measures of performance in the application phase yield $tp = 1.0,\ fp = 0.25,\ ac = 0.79$ and $pr = 0.38$. The value of the true positive rate shows that all faults were correctly detected. The significant false positive rate is mainly due to the last part of the considered time interval, where the ambient temperature and accordingly the room temperature rise beyond their respective ranges in the training interval. This implies a different operation regime, which was not present in the training phase and therefore leads to the detection of faults, where actually normal operation occurs. In order to avoid the detection of many false positive events, training should be repeated when environmental conditions change significantly or the operation mode switches.

Fault responsibility

Analyzing the responsibilities of the individual sensors and the subsets of sensors for the detected faults can give an indication for the reason of a fault, i.e. the fault type. Here, we compare the estimated responsibilities from the clustering method with the actual fault reasons. We choose $d = 20$ for the number of samples and consider subsets up to a maximal size of 5 for the responsibility analysis, following the procedure described above. The total number of subsets

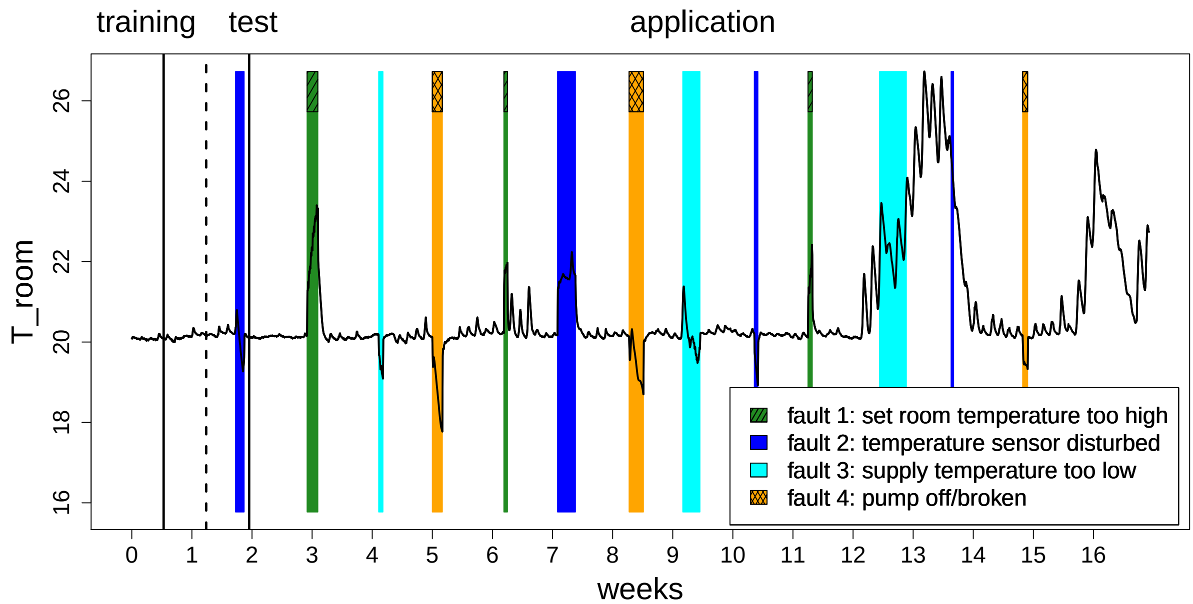

Figure 2: Shown is, for illustration, one variable of the simulation data, the room temperature. Training, test and application intervals are separated by solid and dashed vertical lines. There are four different (recurring) faults implemented into the model, highlighted by the color code. The training interval contains only fault-free data points. Test and application intervals contain faults and fault-free data. The test interval is used to determine the cluster parameter ϵ. Performance of the clustering algorithm is evaluated on the application interval.

(including single variables, i.e. size-1-subsets) thus yields 218. In order to make the results of the responsibility analysis easier to understand and to visualize, we project the results for the 218 different subsets to our 8 input variables in the following way: First, we divide the responsibility corresponding to each subset by the mean of all responsibilities for a given data point. This rescaling procedure allows for studying the relative contribution of each subset to a specific fault, rather than the absolute impact. Then, for each variable N, we determine those subsets, which involve N and take, for each faulty data point, the sum over all rescaled responsibilities of those subsets. Finally, we divide the result by the number of all subsets involving N.

The in this way obtained responsibilities determine the size of the red symbols in Fig. 3. In order to compare the obtained responsibilities with the true reasons for the faults, Fig. 4 shows the resulting responsibilities in the whole considered time interval in terms of a color code, where dark gray areas indicate high responsibility and light gray areas indicate low responsibility. For comparison, the true fault types are shown.

The first observation, comparing the true class in Fig. 4 to the detected faults and responsibilities, is, that same fault types correspond to similar responsibility patterns. This is actually a requirement for a valid fault identification routine. The second thing is, that the room temperature is affected by all types of faults and therefore shows high responsibility almost for every detected fault. This means, one has to analyze the responsibilities of the remaining variables in order to be able to distinguish between different fault types.

Let us have a closer look at the different responsibility patterns, starting from the first fault in the application interval (around week 3): Apart from the room temperature, the valve signal, the return temperature and the set temperature are affected. As the set temperature can only change due to some external influence, it seems plausible, that it is the actual reason for the fault. Assuming, that the set temperature was increased, the valve will open up to cover the increasing heat demand and the return temperature will increase due to a higher mass flow rate. The reverse effect would be observed for a decreased set temperature. All variables, which are affected by a changing set temperature show high responsibilities, which indicates that the set temperature is the actual reason for the fault. This is in agreement with the true type of the fault (type 1).

The next fault (around week 4) shows high responsibilities for the valve signal, the flow temperature and the supply temperature. Again, one can argue, that a change in the supply temperature can only be caused externally, which leads to the conclusion, that the supply temperature could be the reason for the fault and the flow temperature and the valve signal are affected by this change in the supply temperature. It is, indeed, simple to explain the effect on the flow temperature and the valve signal: the flow temperature is directly correlated with the supply temperature and therefore immediately changes when the supply temperature changes. The valve signal reacts on the in-

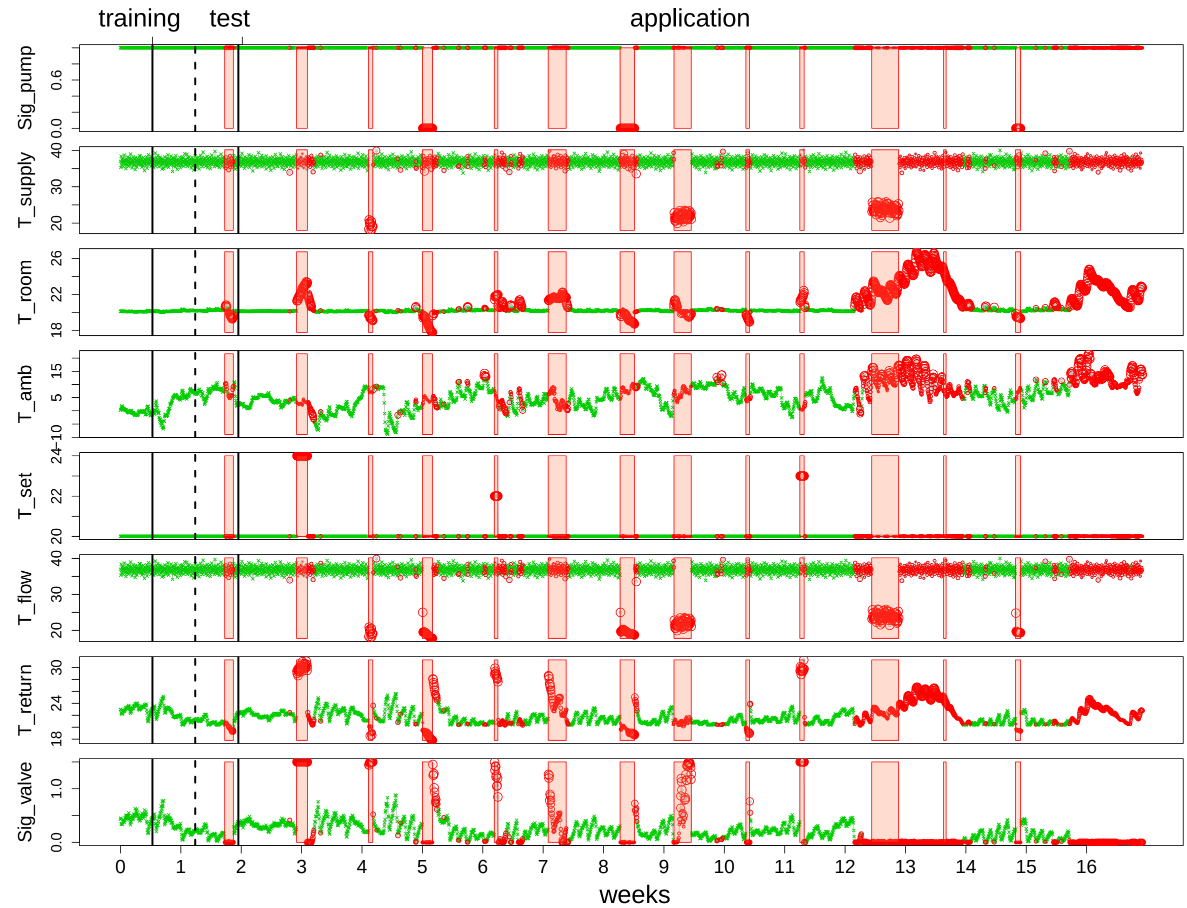

Figure 3: Timeseries of all variables. Training, test and application intervals are indicated by vertical lines. The results of the clustering method are captured by the color code: green symbols correspond to data falling into a cluster and hence being identified as non-faulty. Red symbols correspond to data falling into no cluster and therefore being identified as faulty. Additionally, the size of the red symbols indicates the responsibility of the respective variable for a given fault. Red areas correspond to time intervals where true faults occurred (compare Fig. 2).

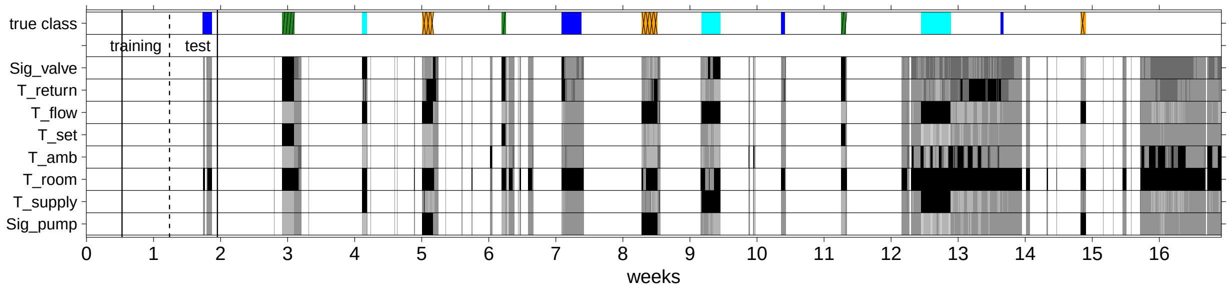

Figure 4: Fault responsibilities for all variables. Light gray areas correspond to faulty intervals, where minor responsibility could be identified for the respective variable. White areas correspond to fault-free intervals. Dark gray areas correspond to high responsibility of the respective variable for a detected fault. The visible patterns can be related to the true faults (compare Fig. 2) and their causes.

creased/decreased heat supply and accordingly regulates the mass flow. These observations coincide well with the type of the fault, namely type 3.

The next fault (around week 5) shows high responsibilities for the flow and return temperatures and the pump signal. In this case, it is quite obvious that the fault is caused by the pump, which leads to a decrease of both, flow and return temperature and a decrease of the difference between the two. This is also in agreement with the fault type 4.

Then, between weeks 6 and 7, we have another occurrence of fault 2, which can be told from the similar responsibility profile to week 3. Additionally, we find in this region some false positive events, which are partly due to the outside air temperature and can be partly not further identified.

Around week 7, we have a responsibility pattern, which barely shows any peaks apart from the room temperature. One can, however, distinguish some contribution from the valve signal and the return temperature. This responsibility pattern matches the pattern of fault 2, despite the responsibility of the set temperature, which is missing here. From this, we can deduce, that a similar effect to changing the set temperature must have taken place. Another variable, which influences the valve signal, as it can be seen from equation (2), is the room temperature. Typically, when the room temperature changes, the valve should react accordingly. If the induced effect on the valve is disproportional, this can be an indicator for a disturbed temperature sensor, as it is, indeed, the case here.

The next four faulty regions can be analyzed in the same way as described before. After week 12 we see a significant responsibility of the ambient temperature. This is an indicator for false positive events and should be a hint for starting a new training period, as the conditions, which were present in the training interval are not guaranteed to be fulfilled any more. The responsibility patterns of the true positive events, which occur in this period, are also affected by the changed environmental conditions: For example, the valve signal does not show any significant contribution to the pattern, as previously. This can be explained by the end of the heating period, where the valve is permanently closed and not affected by any faults concerning the heating circuit.

As mentioned earlier, the responsibility analysis allows not only to identify possible reasons for the detected faults, but also serves as an indicator for false positive events. In particular, when the outside temperature shows significant responsibility, it is likely that a detected fault actually corresponds to normal operation, but the operation mode might have changed. A post-processing routine can make use of this observation and reduce the number of false positive events by checking the respective responsibilities first and remove faults which seem to be caused only by varying environmental conditions.

CONCLUSION

In this paper we introduced a method for FDD in HVAC systems which makes use of a density-based clustering algorithm. All data points, which do not fall into the clusters defined in the training phase are considered to be faulty. For each faulty data point, corresponding responsible variables can be identified by analyzing the characteristics of the “disturbances” of a selected faulty data point.

The results obtained by applying the clustering approach to simulation data indicate, that the described method is in principal useful for detecting faults in HVAC systems. The sensitivity of 100% indicates that all faults are detected. Moreover, a closer analysis of the responsibilities of specific variables for the detected faults reveals plausible relationships between the most likely causes according to the FDD routine and the actual reasons for the faults.

A main disadvantage inherent to all methods, which are constructed on the basis of fault-free data, is a relative high false-positive rate or, alternatively, the need for a frequent repetition of the training procedure. In the example considered here, 25% of the normal operation was found to be faulty when training, test and application intervals were chosen to be 5 days, 5 days and 16 weeks, respectively. Most of these false positive events occur close to the end of the application interval, when the environmental conditions differ significantly from the training situation. A post-processing routine, which eliminates predicted faults, which are unlikely to be true faults (for example due to the respective responsibilities) can further improve the outcome.

Another aspect, which has to be taken into account with this type of fault detection method is the sensitivity of the outcome with respect to the training situation. As all data, which is used for training is considered to be fault-free, and, conversely, all data which differs significantly from the training data is considered to be faulty, this initial set of data has to be chosen very carefully. If, for example, transient behavior or noise is present at the first place, in the training situation, this means that similar behavior is likely to be classified as fault-free also later, in the application phase. On the contrary, if transient behavior and noise are absent in the training phase, a later occurence of transient behavior and noise can lead to false positive classifications.

In order to check the applicability and robustness of the method in real-world situations, a realistic test on monitoring data from building facilities is indispensable. Furthermore, fine-tuning of some aspects is yet to be done and the performance of this method has to be compared to alternative approaches. A promising approach for the development of an extensive FDD methodology is the combination of outlier-detection methods with rule-based approaches.

ACKNOWLEDGEMENTS

This work was conducted in the framework of the EC-funded H2020 project HIT2GAP under grant agreement No. 680708.

NOMENCLATURE

FDD = fault detection and diagnosis
HVAC = heating, ventilation and air conditioning
APAR = air handling unit performance assessment rules
DBSCAN = density-based spatial clustering of applications with noise (Ester et al., 1996)
ROC = receiver operating characteristic

References

Bruton, K., Raftery, P., Kennedy, B., Keane, M. M., and OSullivan, D. T. J. 2013. Review of automated fault detection and diagnostic tools in air handling units. *Energy Efficiency*.

Castro, N. S. and Vaezi-Nejad, H. 2005. Cite-ahu, an automated commissioning tool for air-handling units. In *National Conference on Building Commissioning*, pages 4–6.

Dexter, A. and Pakanen, J. 2001. *Demonstrating Automated Fault Detection and Diagnosis Methods in Real Buildings*. International Energy Agency, Energy Conservation in Buildings and Community Systems, Annex 34: Computer-aided Evaluation of HVAC System Performance, Symposium 217. Technical Research Centre of Finland (VTT), Espoo.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231.

Fawcett, T. 2006. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874.

Ginestet, S., Marchio, D., and Morisot, O. 2008. Evaluation of faults impacts on energy consumption and indoor air quality on an air handling unit. *Energy and Buildings*, 40(1):51–57.

Hartigan, J. A. and Wong, M. A. 1979. Algorithm as 136: A k-means clustering algorithm. *Applied statistics*, pages 100–108.

Hennig, C. 2014. *fpc: Flexible procedures for clustering*. R package version 2.1-9.

Hyvärinen, J. and Kärki, S. 1996. *Building Optimization and Fault Diagnosis Source Book*. Energy Conservation in Buildings and Community Systems Programme. Annex 25. Real Time Simulation of HVAC Systems for Buildings Optimisation, Fault Detection and Diagnostics. Technical Research Centre of Finland (VTT), Espoo.

Katipamula, S. and Brambley, M. R. 2005. Review article: methods for fault detection, diagnostics, and prognostics for building systemsa review, part i. *HVAC&R Research*, 11(1):3–25.

Pérez-Lombard, L., Ortiz, J., and Pout, C. 2008. A review on buildings energy consumption information. *Energy and buildings*, 40(3):394–398.

Schein, J., Bushby, S. T., Castro, N. S., and House, J. M. 2006. A rule-based fault detection method for air handling units. *Energy and Buildings*, 38(12):1485–1492.

Sibson, R. 1973. Slink: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34.

Steinbach, M., Karypis, G., Kumar, V., et al. 2000. A comparison of document clustering techniques. In *KDD workshop on text mining*, volume 400, pages 525–526. Boston, MA.