Partial cloud-based evolving method for fault detection of HVAC system

Conference Paper · July 2018

DOI: 10.1109/FUZZ-IEEE.2018.8491478

3 authors, including:

Goran Andonovski
University of Ljubljana

Saso Blazic
University of Ljubljana

Partial cloud-based evolving method for fault detection of HVAC system

Goran Andonovski
Faculty of Electrical Engineering
University of Ljubljana, Slovenia
goran.andonovski@fe.uni-lj.si

Sašo Blažič
Faculty of Electrical Engineering
University of Ljubljana, Slovenia
saso.blazic@fe.uni-lj.si

Igor Škrjanc
Faculty of Electrical Engineering
University of Ljubljana, Slovenia
igor.skrjanc@fe.uni-lj.si

***Abstract*—In this paper we present an initial investigation of the evolving cloud-based method using partial density calculation for fault detection of HVAC system (Heating, Ventilation and Air Conditioning). Moreover, we investigate different approaches of choosing the most influential components when calculating the partial local density which is further used for evolving the model structure. The method is based on the simplified fuzzy rule-based-system AnYa where the antecedent part is presented by data clouds. The effectiveness of the proposed method is tested on a model of HVAC system and furthermore, different types of faults are investigated. The results were compared with the well established fault detection method DPCA (Dynamic Principal Component Analysis).**

I. Introduction

The Heating, Ventilation and Air Condition (HVAC) system presents an important part of todays modern and highly automated facilities. The main goal of the HVAC system is to control the indoor temperature and humidity at each room as well as to provide required air quality. According to [3] the energy consumption of the HVAC system accounts on average for $20-40\,\%$ of the industrial building's total energy consumption. Moreover this value is expected to rise by an average rato of $1.5\,\%$ per annum over the next 20 years [4].

A typical HVAC system consists of numerous sensors and actuators and all these signals are stored and processed in an online manner. All this signals could potentially cause a fault of the whole system which can further significantly increase the energy consumption. Beside this, faults directly influence on thermal comfort and air quality and also can generate equipment problems and unnecessary maintenance experiences. Therefore, an efficient monitoring and supervising of the signals (the whole system) is necessary to prevent such situations and to optimize the overall energy efficiency of the building [5].

This paper addresses the problem of fault detection (FD) of HVAC systems. In the literature we can find different model based and statistical approaches that tackle the FD problem of the HVAC system. Most of them are specifically focused on a single part of the system, such as air handling units (AHU) [3], [6], [7], [8], water chillers [9], [10], [11], [12], [13] and variable air volume (VAV) box [14]. The work in this paper is mostly related to the fault detection problem of the air handling unit but also takes into consideration several signals that are related to the water chillers plant.

In this paper we present an evolving cloud-based method for fault detection of HVAC systems. Actually, the paper investigates different approaches of partial density calculation which is used as mechanism for adding new clouds. The partial density is calculated taking into account the most influential components only. The components are chosen according to the variances (diagonal elements of the covariance matrix) of the data. Therefore, the method is data driven and all the computations are done online and in a recursive manner.

The proposed fault detection algorithm is based on the simplified fuzzy-rule-base (FRB) system AnYa [15]. AnYa system does not require any assumptions of the data distribution and it is based of the concept of data clouds. The previous work proposed in [16], [17] calculates the mean RDE using the time thresholds for detecting faults and normal conditions. This can be very dependent of the processes' time constant.

The presented concept is compared to the well-established fault detection method DPCA (Dynamic Principle Component Analysis) [18], [19], [20]. The method could be calculated in a recursive manner using a rank-one modification or Lanczos tridiagonalization of the covariance matrix [21].

The paper is organized as follows. In Section II the HVAC system is presented and some of the possible types of faults are described. Next, in Section III the cloud-based evolving fuzzy system is presented and further, in Section IV the fault detection procedure is described. In Section V the results of the proposed method are compared to the DPCA method. Finally in Section VI the conclusions and further developments are summarized.

II. Description of the HVAC process model

In this section the model of Heating, Ventilation and Air Condition (HVAC) process will be described. The general purpose of the HVAC system is to provide thermal comfort in the rooms and to achieve the required indoor air quality parameters. The HVAC system controls the temperature, the humidity and the pressure (actually the air flow) for each sector (room) that the system is responsible for. Such systems present an important segment of each modern and energy efficient building. They have been widely used in the residential and commercial buildings. According to [3] the contribution of buildings to overall energy consumption in developed countries is between $20\,\%$ to $40\,\%$ of which $40\,\%$

belongs to HVAC systems. Therefore, an efficient management and monitoring of these systems can significantly reduce the energy consumption.

The HVAC system used in this article is presented in Fig. 1 and consists of the following components (mechanical and electrical). First, the blinds for input and output air, which are open to $100\,\%$ when the system is turned on. There are also different types of filters (G3, F5 and F7) that remove the solid particulates such as dust, pollen, etc. The air flow is divided in four separate zones: outdoor, supply, return and exhaust air which are shown (see Fig. 1) with green, blue, yellow and orange color, respectively. The direction of the air flow is shown by the arrows in Fig. 1. We can see that the return air is not mixed with the fresh outdoor air due to the required indoor air quality standards.

The main, electrically controlled, parts of the system are: recuperator, heater, humidifier, cooler, supply and return fan (see Fig. 1). According to the outdoor air condition, and to some other requirements, the general control algorithm controls the openness of each element's actuator. Due to paper limitation we are not going to explain the control structure into detail but we will focus on the fault detection problem. In general, the recuperator, heater, cooler and the humidifier control the temperature and the humidity of the air. On other hand the supply and the return fan control the process of exchanging and replacing air in each room to provide high indoor air quality.

In [22] a model of the presented HVAC system was built for testing and developing new fault detection methods and for optimizing the control algorithm. The parameters of the model were tunned using the real data acquired from the real HVAC system. Therefore, as shown in [22] the model sufficiently represents the static and dynamic properties of the real process.

A. *Possible faults on HVAC system*

The list of signals that are available on the described HVAC system are presented in Table I. The signals are separated in two groups: actuators and sensors. The possible faults that can appear on the HVAC system can be influenced by the actuators and sensors as well. Some of the possible types of faults (also see Fig. 2) are:

1) *Communication error* - The real value of the signal is not the same as the value that the control algorithm sees (First subplot in Fig. 2).
2) *Signal offset* - The real value of the sensor/actuator is higher/lower for some offset value (Second subplot in Fig. 2).
3) *Signal drift* - The real signal starts drifting (Third subplot in Fig. 2).
4) *Step error* - The real signal is on its maximum or minimum value (Fourth subplot in Fig. 2).

III. Evolving fuzzy model

Fuzzy systems are general approximation tools for the modelling of non-linear dynamic processes. In this paper we use a fuzzy-rule-based (FRB) system with a non-parametric antecedent part presented by [15]. The main difference comparing to the classical Takagi-Sugeno [23] and Mambdani [24] fuzzy systems is the simplified antecedent part that relies on the relative data density.

TABLE I
List of signals on HVAC system.

	Description	Symbol	Range
ACTUATORS	Valve - Recuperator	V_r	0–100%
	Valve - Heater	V_h	0–100%
	Valve - Humidifier	V_{hm}	0–100%
	Valve - Cooler	V_c	0–100%
	Fan - Supply air	F_{sa}	0–100%
	Fan - Return air	F_{ra}	0–100%
	Blinds - Outdoor air	B_{oa}	0–100%
	Blinds - Exhaust air	B_{ea}	0–100%
SENSORS	Temperature - Outdoor air	T_{oa}	−40–80 °C
	Temperature - Supply air	T_{sa}	0–50 °C
	Temperature - Return air	T_{ra}	0–50 °C
	Temperature - Recuperation water	T_{rw}	−40–80 °C
	Temperature - Hot water for heater	T_{hw}	0–100 °C
	Temperature - Cold water for cooler	T_{cw}	0–50 °C
	Humidity - Outdoor air	RH_{oa}	0–100%
	Humidity - Supply air	RH_{sa}	0–100%
	Humidity - Return air	RH_{ra}	0–100%
	Pressure - Supply air	P_{sa}	0–300Pa
	Pressure - Return air	P_{ra}	0–300Pa

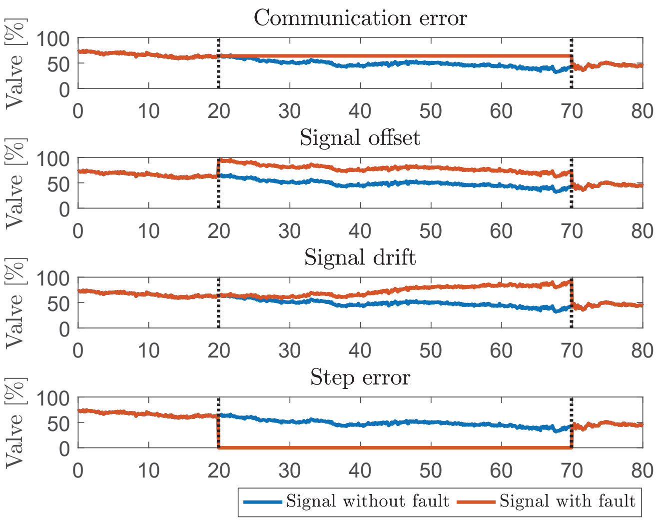

Fig. 2. Examples of possible types of faults that can appear on the actuators or sensors. The dotted black lines denote the start and the end of the faults.

A. *Cloud-based model structure*

The rule-based form of the i^{th} rule is defined as:

$$\mathcal{R}^i : \text{IF} \quad (\boldsymbol{x}_k \sim X^i) \quad \text{THEN} \quad y_k^i = f^i(\boldsymbol{x}_k) \qquad (1)$$

where $\boldsymbol{x}_k = [\boldsymbol{x}_k(1), \boldsymbol{x}_k(2), \ldots \boldsymbol{x}_k(m)]$ is m-dimensional input vector. The operator $\sim$ is linguistically expressed as '*is associated with*', which means that the current data $\boldsymbol{x}$ is related to one of the existing clouds X^i according to the

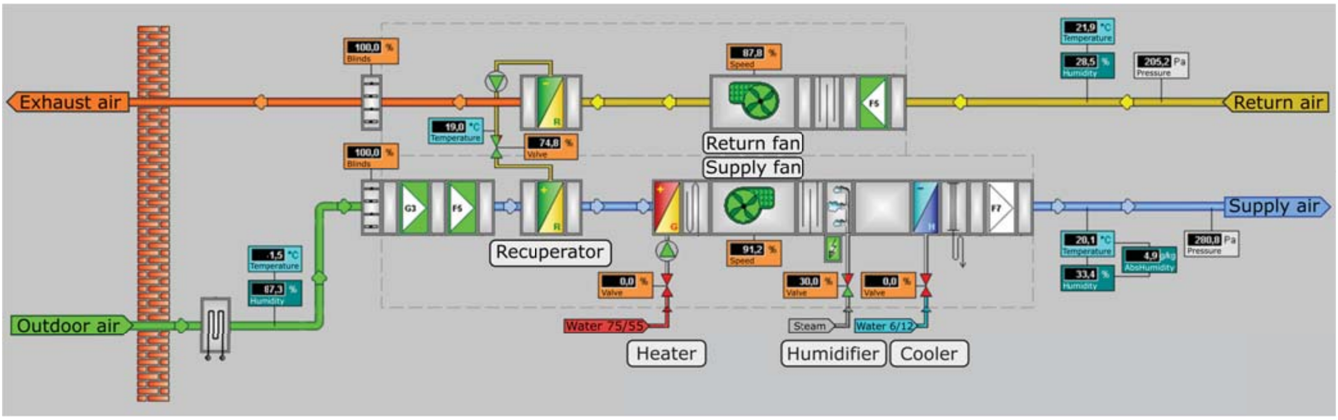

Fig. 1. HVAC system.

membership degree (the normalized relative density of the data):

$$\lambda_k^i = \frac{\gamma_k^i}{\sum_{j=1}^{c} \gamma_k^j} \quad i = 1, \ldots, c \tag{2}$$

where γ_k^i is the local density of the current data $\boldsymbol{x}_k$ with the i^{th} cloud. The local density is defined using the Cauchy kernel as follows [15]:

$$\gamma_k^i = \frac{1}{1 + \frac{\sum_{j=1}^{M^i} (\boldsymbol{x}_k - \boldsymbol{x}_j^i)^T \boldsymbol{A}^i (\boldsymbol{x}_k - \boldsymbol{x}_j^i)}{M^i}}, \; i = 1, \ldots, c \tag{3}$$

where M^i is the number of data points that belong to the i-th cloud. Equation (3) could be rewritten in the recursive form for easier implementation, as follows:

$$\gamma_k^i = \frac{1 + T^i}{1 + (\boldsymbol{x}_k - \boldsymbol{\mu}_{M^i}^i)^T \boldsymbol{A}^i (\boldsymbol{x}_k - \boldsymbol{\mu}_{M^i}^i) + T^i}, \; i = 1, \ldots, c \tag{4}$$

where $\boldsymbol{\mu}_{M^i}^i$ is vector of mean value (center) of the data that are part of the i^{th} cloud. The recursive form of $\boldsymbol{\mu}_{M^i}^i$ is calculated as follows:

$$\boldsymbol{\mu}_{M^i}^i = \frac{M^i - 1}{M^i} \boldsymbol{\mu}_{M^i-1}^i + \frac{1}{M^i} \boldsymbol{x}_k, \quad \boldsymbol{\mu}_0^i = \boldsymbol{x}_k \tag{5}$$

In (4) the matrix $\boldsymbol{A}^i \in \mathbb{R}^{m \times m}$ is an identity matrix or an inverse of the covariance matrix in case of Euclidean or Mahalanobis distance, respectively. In (4) the scalar T^i is calculated as follows:

$$T^i = \frac{M^i - 1}{M^i} trace(\boldsymbol{A}^i \boldsymbol{\Sigma}_{M^i}^i) \tag{6}$$

where $\boldsymbol{\Sigma}_{M^i}^i \in \mathbb{R}^{m \times m}$ is the covariance matrix of the i-th cloud and it is calculated as follows:

$$\boldsymbol{\Sigma}_{M^i}^i = \frac{1}{M^i - 1} \sum_{k=1}^{M^i} (\boldsymbol{x}_k^i - \boldsymbol{\mu}_{M^i}^i)(\boldsymbol{x}_k^i - \boldsymbol{\mu}_{M^i}^i)^T \tag{7}$$

The recursive way of calculating the covariance matrix (7) is explained in the following. Firstly, the un-normalized covariance matrix is computed as:

$$\boldsymbol{S}_{M^i}^i = \boldsymbol{S}_{M^i-1}^i + (\boldsymbol{x}_k - \boldsymbol{\mu}_{M^i-1}^i)(\boldsymbol{x}_k - \boldsymbol{\mu}_{M^i}^i)^T \tag{8}$$

and the covariance matrix is than obtained as

$$\boldsymbol{\Sigma}_{M^i}^i = \frac{1}{M^i - 1} \boldsymbol{S}_{M^i}^i \tag{9}$$

The final algorithm for recursive calculating of the covariance matrix (9) can be summarized using the following instructions [25]:

$$M^i \leftarrow M^i + 1 \tag{10}$$

$$\boldsymbol{\mu}_{M^i}^i \leftarrow \frac{M^i - 1}{M^i} \boldsymbol{\mu}_{M^i-1}^i + \frac{1}{M^i} \boldsymbol{x}_k \tag{11}$$

$$\boldsymbol{S}_{M^i}^i \leftarrow \boldsymbol{S}_{M^i-1}^i + (\boldsymbol{x}_k - \boldsymbol{\mu}_{M^i-1}^i)(\boldsymbol{x}_k - \boldsymbol{\mu}_{M^i}^i)^T \tag{12}$$

$$\boldsymbol{\Sigma}_{M^i}^i \leftarrow \frac{1}{M^i - 1} \boldsymbol{S}_{M^i}^i \tag{13}$$

where the states are initialized with $M^i = 0$, $\boldsymbol{\mu}_0^i = \boldsymbol{x}_1$ and $\boldsymbol{S}_0^i = 0$.

B. Partial local density

In the previous subsection III-A we mentioned that the matrix $\boldsymbol{A}^i$ can be an identity or a covariance matrix in case of Euclidean or Mahalanobis distance, respectively. We can multiply the matrix $\boldsymbol{A}^i$ with another identity matrix $\boldsymbol{Q} \in \mathbb{R}^{m \times m}$:

$$\gamma_{q,k}^i = \frac{1 + T^i}{1 + (\boldsymbol{x}_k - \boldsymbol{\mu}_{M^i}^i)^T \left[\boldsymbol{A}^i \boldsymbol{Q}\right] (\boldsymbol{x}_k - \boldsymbol{\mu}_{M^i}^i) + T^i}, \quad i = 1, \ldots, c \tag{14}$$

where diagonal elements in matrix $\boldsymbol{Q}$ are $q(j) \in \{0, \; 1\}$ for $j = 1, \ldots, m$. Using the new notation (14), we can directly choose the desired components for calculation of partial density. If some component is included into the calculation (14) then the binary variable has value $q(j) = 1$, otherwise $q(j) = 0$. We should note that also the calculation of (6) is changed as follows:

$$T^i = \frac{M^i - 1}{M^i} trace(\boldsymbol{A}^i \boldsymbol{Q} \boldsymbol{\Sigma}_{M^i}^i) \tag{15}$$

The number of components p could be a pre-defined design parameter which defines the number of the chosen components used into calculation (14) (the number of non-zero elements in matrix $\boldsymbol{Q}$). Another way of calculating the number of the most influential components is according to cumulative percent variation (CPV) measure [21]:

$$CPV_1 = \frac{\sum_{j=1}^{p} \sigma(j)}{trace(\boldsymbol{\Sigma})} \times 100\,\% \tag{16}$$

where $\sigma(j)$ is the j-th diagonal element of the covariance matrix $\boldsymbol{\Sigma}$, p is the number of the chosen components, m is the dimension of $\boldsymbol{x}$ and the rule $p \leq m$ holds. Usually, the number of components p is chosen when CPV reaches a predetermined limit of $95\,\%$ [21].

Another way of choosing the most influential components is according to the mean value of the variances:

$$CPV_2 = \frac{trace(\boldsymbol{\Sigma})}{m} \tag{17}$$

where m is the dimension of the input vector $\boldsymbol{x}$. Some component is used into account if its variance is higher than CPV_2.

C. *Evolving cloud-based method*

The evolving nature of the proposed method means that new clouds in the rule-base system (1) could be added if some criteria are satisfied. We should note that at the beginning of the experiment there are no pre-defined clouds. The first cloud with the properties $X_1^1 \in \{\boldsymbol{\mu}_1^1 = \boldsymbol{x}_1, S_1^1 = 0\}$ is defined with the first data point $\boldsymbol{x}_1$ received.

At each time stamp k all the partial densities $\gamma_{\delta k}^i$ between the current data $\boldsymbol{x}_k$ and the existing clouds X^i are calculated ($i = 1, \ldots, c$). The current active cloud is the one with the maximal partial density $\max_i \gamma_{\delta k}^i$ ("winner takes all"). If this value is lower than a pre-defined threshold γ_{max} ($\max_i \gamma_{\delta k}^i < \gamma_{max}$) then a new cloud is added. The default value of $\gamma_{max} = 0.93$ is chosen [26]. Further, to protect adding new clouds based on outliers, the evolving mechanism is frozen for $n_{add} = m+1$ samples after a new cloud is added. This mechanism is also useful for suitable initialization of the covariance matrix.

We have to note here that only the properties $\boldsymbol{\mu}_k^i$, and S_k^i of the currently active cloud (with maximal partial density) are updated and the properties of all the other clouds are kept constant.

IV. Fault detection procedure

In this section the fault detection procedure will be described. For this purpose three different data sets are necessary. The first data set describes the normal process operation (without faulty states, $F = 0$) while second data set contains faults ($F = 1$). The third data set is used for testing purposes and this data set contains areas of normal process operation as well as areas where there are faults. Therefore the proposed method is trained on the first two data set and its effectiveness is tested on the third data set. In the following subsection this is explained in detail.

A. *Learning/training phase*

As we said above, in the learning/training phase we use two types of data. The first one does not contain any faults and the second one contains faulty states. Running the Algorithm 1 for the first data set we discovered all the clouds that describe normal process operation, labeled as $X^i \in \{\boldsymbol{\mu}_k^i, A_k^i, F = 0\}$ (shortened notation $X_{F=0}^i$). Next, using the second dataset the discovered clouds are labeled as clouds which cover the faulty areas $X^i \in \{\boldsymbol{\mu}_k^i, A_k^i, F = 1\}$ (shortly notation $X_{F=1}^i$).

Algorithm 1 Pseudo algorithm for adding clouds.

Initialize: γ_{max}.
repeat ▷ *k is the running index*
  Input: $\boldsymbol{x}_k = [\boldsymbol{x}_k(1), \boldsymbol{x}_k(2), \ldots \boldsymbol{x}_k(m)]$
  if k==1 **then**
    Init.: $c = 1$
    Init.: $\boldsymbol{\mu}_1^i = \boldsymbol{x}_1$, and $S_1^i = 0$.
    Define: $X_1^i \in \{\boldsymbol{\mu}_1^i, S_1^i, F \in [0, 1]\}$
  else
    Compute $\gamma_{\delta k}^i$ using equation (14).
    if $\max_i(\gamma_{\delta k}^i) < \gamma_{max}$ **then** ▷ *Evolving law*
      New cloud is detected.
      Increment: c.
      Init.: $\boldsymbol{\mu}_1^i = \boldsymbol{x}_k$, and $S_1^i = 0$.
      Define: $X_k^i \in \{\boldsymbol{\mu}_1^i, S_1^i, F \in [0, 1]\}$
    else
      Associate sample $\boldsymbol{x}_k$ with X^i ($max_i \gamma_{\delta k}^i$)
      Update $\boldsymbol{\mu}_k^i$, and S_k^i, for the active cloud X^i
    end if
  end if
until End of data stream.

B. *Validation phase*

In the second stage of the fault detection method, the effectiveness of the method is tested on data set that contains normal operation areas and faults. The desired goal of this stage is to detect the current states correctly. We have to emphasize that after the learning phase we have acquire clouds that describe normal process operation $((X_{F=0}^i))$ and faults $((X_{F=1}^i))$. In the validation stage, for each data point $\boldsymbol{x}_k$ received, the partial local densities are calculated with all the acquired clouds from the previous stage ($X_{F=0}^i$ and $X_{F=1}^i$). The current data point $\boldsymbol{x}_k$ is classified as fault state if the following criteria is satisfied:

$$\max_i \gamma_{\delta k}^i(X_{F=0}^i) < \max_i \gamma_{\delta k}^i(X_{F=1}^i) \tag{18}$$

otherwise, the data point $\boldsymbol{x}_k$ is classified as normal process operation.

V. Experimental results

In this section we will present the experimental results of the proposed method. As we already mentioned in the previous sections the partial cloud-based data driven method was tested on model of HVAC system. The results were compared to established fault detection method DPCA [20], [18], [22]. For the statistical comparing of the methods, the following measures (True Positive Rate - TPR, True Negative Rate - TNR and Accuracy - ACC) were used:

$$TPR = \frac{TP}{total\ samples\ (f=1)} \times 100\,[\%] \tag{19}$$

$$TNR = \frac{TN}{total\ samples\ (f=0)} \times 100\,[\%] \tag{20}$$

$$ACC = \frac{TP + TN}{total\ samples} \times 100\,[\%] \tag{21}$$

where TP (true positives) and TN (true negatives) are the number of correctly detected faults and normal states, respectively; The last measure ACC is more general and takes into account both, TP and TN.

In this experiment four different faults were considered as presented in Table II. For each of this faults three data sets were acquired form the HVAC model [22]. First two data sets were used for learning purposes (see IV-A) and the third one was used for testing (see IV-B).

TABLE II
LIST OF THE FAULTS OF HVAC SYSTEM.

Fault no.	Signal description	Symbol	Fault type
$F1$	Temperature - Hot water	T_{hw}	Offset
$F2$	Valve - Heater	V_h	Step
$F3$	Valve - Cooler	V_c	Step
$F4$	Valve - Recuperator	V_r	Step

In the following Figs. 3, 4, 5, and 6 the results of the validation phase for each fault $F1$, $F2$, $F3$, and $F4$ are presented, respectively. The results on the figures are for the proposed partial cloud-based method. Firstly, the shaded areas in the figures show the actual area where a particular fault occurs. The blue line presents the maximal density of the clouds that describe normal process operation $\max_i \gamma^i_{\delta k}(X^i_{F=0})$ while the red line shows the maximal density of the faulty clouds $\max_i \gamma^i_{\delta k}(X^i_{F=1})$. According to the criterion (18) the current data is classified as fault or normal operation. The green and the red dots present the correctly and the wrongly detected data samples, respectively.

In the Tables III, IV and V the values of TPR, TNR, and ACC measures for each fault are shown, respectively. Moreover, the tables show the results of DPCA method and the different version (subsection III-B) of the proposed cloud based (CB) method. From the results shown in the tables we can consider that the cloud-based method is comparable with the DPCA method. As shown in Table III $CB_{default}$ method is better that DPCA method in three of four faults. While in Table IV the situation is opposite, which can be further summarized in table V where the comparing results are equal.

VI. Conclusion

In this paper we investigate an evolving cloud-based method which uses partial calculation of the local density for fault

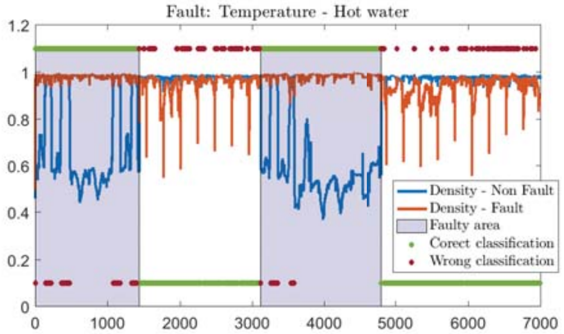

Fig. 3. Fault detection results for the fault $F1$.

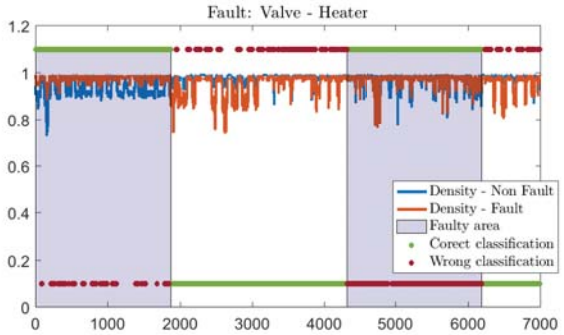

Fig. 4. Fault detection results for the fault $F2$.

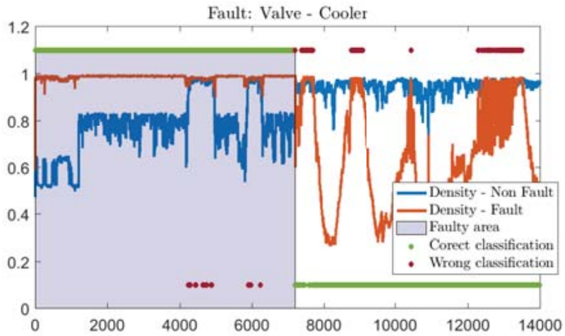

Fig. 5. Fault detection results for the fault $F3$.

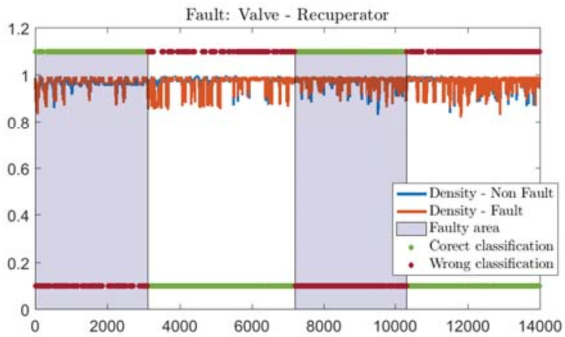

Fig. 6. Fault detection results for the fault $F4$.

TABLE III
TPR - TRUE POSITIVE RATE.

	F1	F2	F3	F4
$DPCA$ [22]	82.5	65.2	97.7	**71.6**
$CB_{default}$	**91.4**	**78.2**	**99.9**	62.6
CB_{CPV_1}	87.2	77.7	91.4	61.6
CB_{CPV_2}	86.4	78.1	90.2	60.1

TABLE IV
TNR - TRUE NEGATIVE RATE.

	F1	F2	F3	F4
$DPCA$ [22]	**99.2**	**79.0**	**98.4**	66.2
$CB_{default}$	83.5	77.2	77.1	**71.2**
CB_{CPV_1}	80.2	72.4	74.6	69.9
CB_{CPV_2}	83.0	73.8	70.2	63.0

TABLE V
ACC - ACCURACY.

	F1	F2	F3	F4
$DPCA$ [22]	**91.95**	73.35	**98.04**	67.90
$CB_{default}$	86.91	**77.60**	88.28	**68.55**
CB_{CPV_1}	82.53	72.03	82.11	65.21
CB_{CPV_2}	81.85	70.23	84.37	63.81

detection purposes. The most influential components were chosen according to the different criteria. The proposed procedure was tested on data acquired from HVAC model and the results were compared to the well established fault detection method DPCA. From the obtained results we can conclude that the cloud-based method which takes into account all the components could be competitive with the DPCA method. Unfortunately, introducing the partial density calculation does not improve the results, but this could be a question for further research. The main advantage of the partial calculation is reducing the number of components which further reduces the computational time.

REFERENCES

[1] J. Dong, Y. Wu, and G. H. Yang, "A New Sensor Fault Isolation Method for T-S Fuzzy Systems," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2437–2447, 2017.

[2] J. Dong and G. H. Yang, "Observer-Based Output Feedback Control for Discrete-Time T-S Fuzzy Systems With Partly Immeasurable Premise Variables," *EEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 1, pp. 98–110, 2017.

[3] K. Bruton, D. Coakley, P. Raftery, D. O. Cusack, M. M. Keane, and D. T. O'Sullivan, "Comparative analysis of the AHU InFO fault detection and diagnostic expert tool for AHUs with APAR," *Energy Efficiency*, vol. 8, no. 2, pp. 299–322, 2015.

[4] L. Pérez-Lombard, J. Ortiz, and C. Pout, "A review on buildings energy consumption information," *Energy and Buildings*, vol. 40, no. 3, pp. 394–398, 2008.

[5] K. Severson, P. Chaiwatanodom, and R. D. Braatz, "Perspectives on process monitoring of industrial systems," *IFAC-PapersOnLine*, vol. 28, no. 21, pp. 931–939, 2015.

[6] B. Fan, Z. Du, X. Jin, X. Yang, and Y. Guo, "A hybrid FDD strategy for local system of AHU based on artificial neural network and wavelet analysis," *Building and Environment*, vol. 45, no. 12, pp. 2698–2708, 2010.

[7] S. Wang and F. Xiao, "AHU sensor fault diagnosis using principal component analysis method," *Energy and Buildings*, vol. 36, no. 2, pp. 147–160, 2004.

[8] V. Reppa, P. Papadopoulos, M. M. Polycarpou, and C. G. Panayiotou, "A Distributed Architecture for HVAC Sensor Fault Detection and Isolation," *IEEE Transaction of Control Systems Technology*, vol. 23, no. 4, pp. 1323–1337, 2015.

[9] P. Wang, S. Member, and R. Gao, "On - Line Fault Detection and Diagnosis for Chiller System," in *IEEE International Conference on Automation Science and Engineering (CASE)*, Fort Worth, TX, USA, 2016, pp. 1313–1318.

[10] X. Xu, F. Xiao, and S. Wang, "Enhanced chiller sensor fault detection, diagnosis and estimation using wavelet analysis and principal component analysis methods," *Applied Thermal Engineering*, vol. 28, no. 2-3, pp. 226–237, 2008.

[11] A. Beghi, R. Brignoli, L. Cecchinato, G. Menegazzo, and M. Rampazzo, "A data-driven approach for fault diagnosis in HVAC chiller systems," in *2015 IEEE Conference on Control and Applications (CCA)*, 2015, pp. 966–971.

[12] G. Andonovski, G. Mušič, S. Blažič, and I. Škrjanc, "Evolving model identification for process monitoring and prediction of non-linear systems," *Engineering Applications of Artificial Intelligence*, vol. 68, no. October, pp. 214–221, 2018.

[13] S. M. Namburu, M. S. Azam, J. Luo, K. Choi, and K. R. Pattipati, "Data-driven modeling, fault diagnosis and optimal sensor selection for HVAC chillers," *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 3, pp. 469–473, 2007.

[14] W. H. Allen, A. Rubaai, and R. Chawla, "Fuzzy Neural Network-Based Health Monitoring for HVAC System Variable-Air-Volume Unit," *IEEE Transactions on Industry Applications*, vol. 52, no. 3, pp. 2513–2524, 2016.

[15] P. Angelov and R. Yager, "Simplified fuzzy rule-based systems using non-parametric antecedents and relative data density," in *Symposium Series on Computational Intelligence (IEEE SSCI 2011) - IEEE Workshop on Evolving and Adaptive Intelligent Systems (EAIS 2011)*, 2011, pp. 62–69.

[16] B. S. J. Costa, P. P. Angelov, and L. A. Guedes, "Real-time fault detection using recursive density estimation," *Journal of Control, Automation and Electrical Systems*, vol. 25, no. 4, pp. 428–437, 2014.

[17] ——, "Fully unsupervised fault detection and identification based on recursive density estimation and self-evolving cloud-based classifier," *Neurocomputing*, vol. 150, no. Part A, pp. 289–303, 2015.

[18] B. D. Ketelaere, M. Hubert, and E. Schmitt, "Overview of PCA-Based Statistical Process-Monitoring Methods for Time-Dependent, High-Dimensional Data," *Journal of Quality Technology*, vol. 47, no. 4, pp. 318–335, 2015.

[19] S. Yin, S. X. Ding, A. Haghani, H. Hao, and P. Zhang, "A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process," *Journal of Process Control*, vol. 22, no. 9, pp. 1567–1581, 2012.

[20] T. J. Rato and M. S. Reis, "Defining the structure of DPCA models and its impact on process monitoring and prediction activities," *Chemometrics and Intelligent Laboratory Systems*, vol. 125, pp. 74 – 86, 2013.

[21] W. Li, H. Yue, S. Valle-Cervantes, and S. J. Qin, "Recursive PCA for adaptive process monitoring," *Journal of Process Control*, vol. 10, pp. 471–486, 2000.

[22] Ž. Stržinar, "Modeliranje in zaznavanje napak v klimatskih sistemih," Master Thesis, University of Ljubljana, 2017.

[23] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-15, no. 1, pp. 116–132, 1985.

[24] E. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1–13, 1975.

[25] S. Blažič, P. Angelov, and I. Škrjanc, "Comparison of Approaches for Identification of All-data Cloud-based Evolving Systems," in *2nd IFAC Conference on Embedded Systems, Computer Intelligence and Telematics CESCIT 2015*, Maribor, Slovenia, 2015, pp. 129–134.

[26] G. Andonovski, P. Angelov, S. Blažič, and I. Škrjanc, "A practical implementation of Robust Evolving Cloud-based Controller with normalized data space for heat-exchanger plant," *Applied Soft Computing*, vol. 48, pp. 29–38, 2016.