Energy 265 (2023) 126337

Contents lists available at ScienceDirect

Energy

journal homepage: www.elsevier.com/locate/energy

An evolving learning-based fault detection and diagnosis method: Case study for a passive chilled beam system

Liping Wang [a,*], James Braun [b,c], Sujit Dahal [a]

[a] Civil and Architectural Engineering, University of Wyoming, 1000 E University Ave, Laramie, WY, 82071, USA
[b] School of Mechanical Engineering, Purdue University, 585 Purdue Mall, West Lafayette, IN, 47907, USA
[c] Center for High Performance Buildings, Ray W. Herrick Laboratories, Purdue University, 177 S Russell St, West Lafayette, IN, 47907, USA

ARTICLE INFO

Keywords:
Evolving learning
Fault detection and diagnosis
Feature selection
Passive chilled beam

ABSTRACT

Traditional fault detection and diagnosis (FDD) methods learn from training data obtained under limited operating conditions, after which they stop learning. In this study, we developed an evolving learning-based FDD method for HVAC systems, which learns as the performance of a building system and its components changes. Specifically, an evolving learning algorithm—growing Gaussian mixture regression—is used to construct both a data-driven model representing normal performance and a transfer function for fault diagnosis. The evolving learning-based FDD method was demonstrated for detecting and diagnosing common faults of passive chilled beam systems. We employ generalized performance indices, such as the deviations between predictions (expectations) and measurements, the differences between two parameters, and other features extracted from parameters. A novel feature selection method was developed for selecting fault signatures. An uncertainty threshold determining whether a performance index was within the range of normal operation influences false alarm rates. By increasing the uncertainty thresholds from zero to two standard deviations, false alarm rates for normal operations were reduced from 14.8% to 1.3% and the percentage of normal operation data categorized as an unknown operation was reduced from 25% to 0%. Eight known faults were detected and diagnosed with an accuracy of 100%. A new fault was first categorized as an unknown fault before evolving. After evolving the transfer function by updating the key parameters of the Gaussian components, the unknown fault was also accurately diagnosed. The evolving learning-based FDD method and novel feature selection method can be employed for detecting and diagnosing common faults of other systems or subsystems in the built environment.

1. Introduction

Buildings account for 40% of total energy consumption [1], 74% of electricity use [2], and 37% of greenhouse gas emissions in the U.S [3]. Therefore, environmental impacts and resource consumption associated with building operations are significant throughout the entire life cycle of buildings. Heating, ventilation and air conditioning (HVAC) systems account for about 67% of the total energy use in buildings. Despite national efforts toward improving system performance and sustainability, most HVAC systems in existing buildings do not operate close to their design efficiency due to equipment degradation, out-of-calibration sensors, or improper control operations. These issues can lead to high maintenance costs, occupant discomfort, and an energy use penalty of between 15% and 30% for commercial buildings [4].

Fault detection and diagnosis (FDD) for HVAC systems in buildings can enable early detection and identification of operational faults based on the analysis of measured behaviors. FDD is critical in achieving the goal of reducing energy waste in buildings caused by operational faults and ensuring the persistence of building commissioning. The FDD subject of conventional HVAC systems has been studied since the 1990s [5–8].

Traditional FDD approaches that are applied to existing buildings learn from training data obtained under limited operating conditions to detect and diagnose faults, after which they stop learning. The major challenge is that the training data available to create diagnostic algorithms do not include all possible operating conditions that the tested systems and components experience throughout their life cycles. It is expected that building system performance such as energy use changes over time due to natural wear, changing control sequences, equipment replacement or addition, or other external factors. This can lead to missed detections and false alarms when traditional FDD approaches are

* Corresponding author.
E-mail address: lwang12@uwyo.edu (L. Wang).

https://doi.org/10.1016/j.energy.2022.126337
Received 7 May 2022; Received in revised form 30 October 2022; Accepted 3 December 2022
Available online 6 December 2022

Nomenclature

Symbols

E	expectation operator [−]
k	number of components [−]
N	probability distribution function of a multi-variable Gaussian distribution [−]
S_f	change rates of supply water flow [gpm/hr]
t	sample index [−]
q	expected posterior [−]
w	updated weight coefficient [−]
π	weight coefficient [−]
β	learning rate [−]
ρ	learning factor [−]
ΔC	deviations between predicted and actual cooling rates [kW]
Δf	deviations between predicted and actual flow rates [gpm]
Δp:	deviations between loop pressure and its setpoint [psi]
ΔT	temperature deviations between temperature and its setpoint, or temperature differences [°F]
$\boldsymbol{d}$	multi-variable data [−]
$\widehat{f}$	fault diagnosis results [−]
$\boldsymbol{f}$	individual faults [−]
$\boldsymbol{F}$	fault matrix [−]
$\boldsymbol{H}$	a transfer function for fault diagnosis [−]
$\boldsymbol{i}$	a fault feature vector [−]
$\boldsymbol{P}$	parameter matrix [−]
$\boldsymbol{S}$	fault signature matrix [−]
$\boldsymbol{\mu}$	mean vector [−]
$\boldsymbol{\delta}$	covariance matrix [−]Prefix
cusum	cumulative sums over a 24-h moving window

Subscripts

j	the j^{th} component in a Gaussian mixture model
X	independent variables
Y	dependent variable
ZAT	zone air temperature
r-s:	between supply and return water temperature
sZAT	between supply water temperature and zone air temperature setpoints

applied [9,10]. Performance indices such as energy use require that the baseline models for predicting energy use evolve along with the dynamic changes in building systems and components. Furthermore, fault severity levels and fault symptoms may change over time. Given that the training set for FDD does not cover all operating conditions and fault scenarios, FDD algorithms for building HVAC systems need to evolve over time.

As opposed to FDD applied to conventional HVAC systems (e.g., variable air volume systems), FDD for high-performance HVAC systems has rarely been studied. To meet the goals of the net-zero energy building initiatives for commercial buildings, FDD technologies that can continuously ensure thermal comfort and energy efficiency are needed especially for the operation of high-performance HVAC systems that have the potential to achieve high indoor environmental quality and deep energy reduction through integrated system design and advanced operation for commercial buildings. These high-performance HVAC systems (such as chilled beam [11], radiant heating/cooling [12], and mixed-mode ventilation [13,14]) are complex in operation.

To address the research gaps discussed above, the objective of the study is to develop an evolving learning-based FDD method for HVAC systems, which learns as the performance of a building system and its components changes. Specifically, an evolving learning algorithm—growing Gaussian mixture regression—is used to construct both a data-driven model representing normal performance and a transfer function for fault diagnosis. A novel feature selection method was also developed for selecting fault signatures to enhance the robustness and transferability of the FDD algorithm. The evolving learning-based FDD method as well as the feature selection method were demonstrated for detecting and diagnosing common faults of passive chilled beam systems in this study.

The methodology of the evolving learning-based FDD and the dynamic updating mechanism for growing Gaussian mixture regression were presented in Section 2. A demonstration of the evolving learning-based FDD method is presented in Sections 4-7 for a passive chilled beam system. Section 3 describes fault implementation in the passive chilled beam system; Section 4 presents the development of an evolving learning-based GGMR model representing cooling rates of the PCB system under normal operation; Sections 5 and 6 present evaluation and selection of key performance indices for fault detection and diagnosis; Section 7 develops a GGMR model for fault diagnosis of the passive chilled beam system and presents diagnosis results for both known and unknown faults.

2. Methodology

2.1. Evolving learning-based fault detection and diagnosis

A schematic diagram of an evolving learning-based FDD method is illustrated in Fig. 1. The evolving learning-based approach is used to construct both a data-driven model representing normal performance and a transfer function $\boldsymbol{H}$ for fault diagnosis. The evolving learning-based FDD method includes two stages: "detection" and "diagnosis."

Detection: Performance indices for a system of interest were composed for the purpose of fault detection. These performance indices include the deviations between predictions (expectations) and measurements, the differences between two key parameters, and other features extracted from parameters (e.g., cumulative differences or changes of parameters over time). Furthermore, the performance indices are categorized into static (non-evolving) performance indices and evolving performance indices.

Key performance parameters of HVAC systems such as cooling/heating energy rates and air/water flow rates evolve with the changes in HVAC operating environments, building systems, and components over their life cycles. These key performance parameters can be predicted using evolving learning-based models [15]. The evolving performance indices are created using the predictions from these evolving learning-based models and their corresponding measurements. One example of the evolving performance indices is the differences between cooling/heating energy rates predicted from evolving learning method-based models and their measurements. Traditional data-driven-based modeling is limited by the training data which do not include all possible operating conditions for the system of interests. The evolving learning-based models overcome this limitation of the traditional data-driven-based modeling approach by updating key parameters over time.

The non-evolving performance indices do not include predictions from evolving learning-based models. Instead, they are built upon actual measurements and/or control operating parameters. An example of the non-evolving performance index could be the differences between supply water temperature and its setpoint. The expected ranges of non-evolving performance indices are static and should not change over time such as the acceptable deviations between controlled variables and setpoints. For detecting degradation faults such as coil fouling under which performance gradually changed over time, static performance indices should be used.

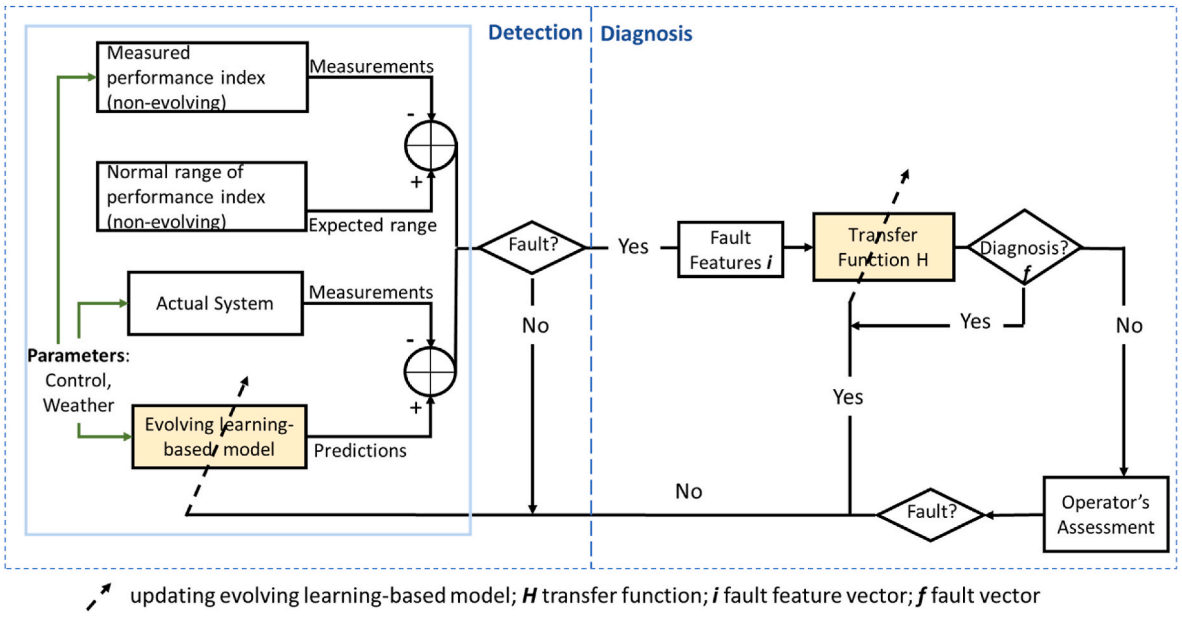

Fig. 1. Schematic diagram of evolving learning-based fault detection and diagnosis method.

For performance indices with the evolving feature, the baseline models for performance parameters such as predicted energy use would evolve along with the changes in building systems and components to represent normal operation over time. For non-evolving performance indices, the calculated index values are compared to the expected ranges under normal operation. If statistically significant deviations between the prediction (expected ranges) and the measurement of performance indices are observed, a fault, multiple faults, or an unseen normal operation is detected, and the diagnosis stage is initiated. Otherwise, the system is operating normally and the evolving learning-based model is updated with the measurement. Because the performance indices include both static (non-evolving) performance indices and evolving performance indices, the FDD method is applicable for different types of faults with either sudden or gradual performance changes.

Diagnosis: Finding a diagnosis transfer function $\boldsymbol{H}$ is essential to enable fault diagnosis. An evolving learning-based transfer function $\boldsymbol{H}$ is used to relate the fault feature vector $\boldsymbol{i}$ comprising various performance indices to the fault vector $\boldsymbol{f}$ based on experimental data. Given $\boldsymbol{H}$, faults can be identified based on the fault feature vector $\boldsymbol{i}$ by using $\widehat{f} = \boldsymbol{H}(i)$, where $\widehat{f}$ is the fault diagnosis results using integers representing individual faults. If no faults can be diagnosed for the measurements in the diagnosis state, the measurement is identified as a new state, which is either an unseen fault or an unseen normal operation. After a new state (either an unseen fault or unseen normal operation) is identified by the algorithm, a message about the new state will be sent to a building operator, and then the building operator will assess the new state and determine 1) whether the system is in a new normal operating mode or an unseen faulty operation, and 2) the root cause of the fault if it is an unseen faulty operation through measurement and verification. Once the new state is unveiled, the measurement of the new state is used to update the evolving learning-based model (normal operation) or the transfer function (faulty operation) based on the building operator's assessment.

2.2. Growing Gaussian mixture regression (GGMR)

The growing Gaussian mixture regression (GGMR) was used in this study to construct a model representing normal operation and the transfer function $\boldsymbol{H}$ as the evolving learning-based data-driven method. GGMR integrates a growing Gaussian mixture model (GGMM) [16] with a Gaussian mixture regression (GMR) [17]. GGMM updates the parameters of the Gaussian components through a recursive procedure, merges Gaussian components, and creates a new Gaussian component based on new measurements. GMR is then used to predict the value of the dependent variable given the observations of independent variables and GGMM parameters.

A flow chart of the GGMR algorithm is shown in Fig. 2. A Gaussian mixture model (GMM) assumes that the observed data are made up of a mixture of multiple Gaussian distributions, and is thus a parametric probability density function represented as a weighted sum of Gaussian component densities, as shown in Eq. (1).

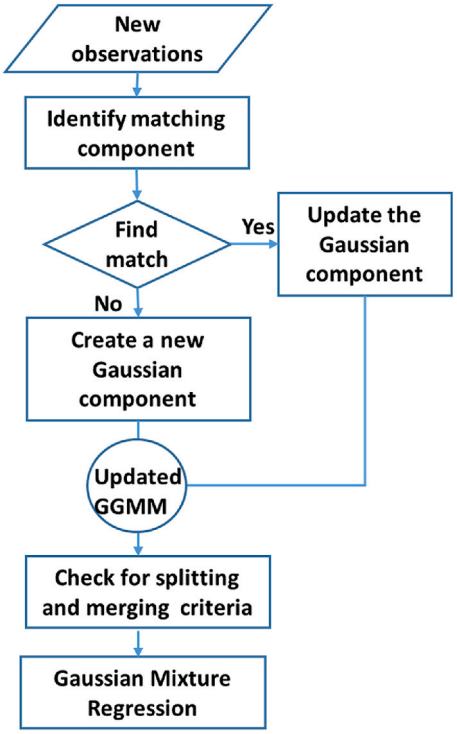

Fig. 2. Flow chart of GGMR algorithm.

$$prob = \sum_{j=1}^{k} \pi_j N(\boldsymbol{i}.\boldsymbol{f}|\boldsymbol{\mu}_j, \delta_j) \quad (1)$$

where π_j, $\boldsymbol{\mu}_j$, and δ_j denote the weight coefficient, mean vector, and covariance matrix of the *jth* component, respectively. $N(\boldsymbol{i}.\boldsymbol{f}|\boldsymbol{\mu}_j, \delta_j)$ represents the probability distribution function of a multi-variable Gaussian distribution. A growing Gaussian mixture model (GGMM) offers the evolution of both the GMM model structure and the parameters in real-time. The learning mechanism includes updating the parameters of the GMR, adding a new component to the existing model, as well as splitting and merging the Gaussian mixture model components.

First, GGMM identifies which Gaussian components will be updated based on the calculated probability density of each. The parameters of the component with the highest probability density (non-zero), including the weight coefficient π_j, mean vector $\boldsymbol{\mu}_j$, and covariance matrix δ_j are updated using a recursive filtering approach, as shown in Eqs. (2)–(4) [16].

$$\pi_j(t) = (1-\beta)\pi_j(t-1) + \beta q_j \quad (2)$$

$$\boldsymbol{\mu}_j(t) = (1-\rho_j)\boldsymbol{\mu}_j(t-1) + \rho_j \boldsymbol{d}(t) \quad (3)$$

$$\delta_j(t) = (1-\rho_j)\delta_j(t-1) + \rho_j(\boldsymbol{d}(t) - \boldsymbol{\mu}_j(t))(\boldsymbol{d}(t) - \boldsymbol{\mu}_j(t))^T \quad (4)$$

where $\boldsymbol{d}(t) = [\boldsymbol{i}(t), \boldsymbol{f}(t)]$, q_j is the expected posterior of the j th component, β is the learning rate, and ρ_j is the learning factor corresponding to updating $\boldsymbol{\mu}_j$ and δ_j.

Second, if the calculated probability density of all Gaussian components based on the new data is zero, this indicates that there is no existing Gaussian mixture component matching the new data. The GGMM then generates a new component and initializes its parameters [16].

Third, after the GGMM model is updated, merging and splitting criteria are checked to ensure the health of the updated model structure. When the volume of a Gaussian component — the determinant of its covariance matrix — exceeds a maximum threshold, the Gaussian component is split into two. The distance between two Gaussian components is measured by the Kullback-Leibler divergence (KLD) [18]. If the KLD of any two Gaussian components is smaller than a minimum threshold, the two components are merged.

Lastly, a Gaussian mixture regression (GMR) [17] is used to predict the dependent variable $\widehat{\boldsymbol{f}}(\boldsymbol{i})$ such as diagnosed faults represented by integers (0, 1, 2 …) given the observed independent variables such as performance indices described in Section 5 and the GGMM parameters of the test data. $\widehat{\boldsymbol{f}}(\boldsymbol{i})$ represents the prediction of the fault labels $\boldsymbol{f}(\boldsymbol{i})$. $\boldsymbol{w}_j(\boldsymbol{i})$ is the updated weight coefficients given the observed independent variables for the j th component, as shown in Eqs. (5) and (6).

$$\widehat{\boldsymbol{f}}_k(\boldsymbol{i}) = E(Y|X=\boldsymbol{i}) = \sum_{j=1}^{k} w_j(\boldsymbol{i})\left(\mu_{jY} + \delta_{jYX}\delta_{jXX}^{-1}(\boldsymbol{i} - \boldsymbol{\mu}_{jX})\right) \quad (5)$$

$$w_j(\boldsymbol{i}) = \frac{\pi_j N(\boldsymbol{i}; \boldsymbol{\mu}_{jX}, \delta_{jXX})}{\sum_{j=1}^{k} \pi_j N(\boldsymbol{i}; \boldsymbol{\mu}_{jX}, \delta_{jXX})} \quad (6)$$

3. Fault implementation

Living Lab 1 (W: 9.5 m × L:10.7 m) at Herrick Laboratories, Purdue University is an open office with a south-facing double-skin façade. The space can be served by a passive chilled beam system (PCB), a hydronic radiant slab system, and/or displacement ventilation from a variable air volume system (VAV). In this study, we implemented five types of common faults at different severity levels (nine individual faults in total) into the passive chilled beam (PCB) system of Living Lab 1 between September 15, 2021 and November 5, 2021. Trend data of the PCB system performance and environmental conditions under normal operation were recorded from April 1, 2021 to July 26, 2021. During the experiments, the radiant slab system was turned off, space heating was served by the VAV system, and space cooling was served by both PCB and VAV systems. Condensation is a common operating issue of chilled beam systems. However, it is avoidable by controlling the supply water temperature above space air dew point temperature via control sequences. To prevent any risk of condensation in Living Lab 1, the supply water temperature setpoint was set dynamically based on thermal loads but at least 1.7 °C (3 °F) above the dew point of the space air.

In this study, we focused on mechanical faults associated with actuators and sensors. The five fault types include stuck chilled beam control valves, stuck chilled water control valves of the heat exchanger, supply water temperature sensor offset, zone air temperature sensor offset, and stuck VFD controller at a constant pump speed. These common faults implemented into the PCB system including severity level, associated components, and labels used in this paper were summarized in Table 1.

Under normal operation, chilled beam valves and the chilled water valve were controlled using PI feedback loops that compare measured control variables and setpoints. For the stuck valve faults, constant signals (e.g., 0%, 50%, or 100%) were used to override the PI loop outputs. In addition to the actual (override) control valve positions, the output signals from PI loops for the control valves were also trended and were used for predicting PCB performance under normal operation.

For temperature sensor offset faults, a constant offset was added to the control sequence via the building control system to adjust readings from the temperature sensors such as supply water temperature or zone air temperature before the temperature variables were provided as inputs for various PI feedback controls.

The supply water pump speed was controlled using a PI loop based on a differential pressure setpoint (10 psi). For the fault of the stuck pump VFD controller at a constant speed, a constant percentage of full pump speed was used to override the PI loop outputs.

4. Baseline model

The baseline model for predicting cooling energy rates of the PCB system under normal operation was constructed for the development of evolving performance indices using the evolving learning-based algorithm—GGMR for fault detection. The selection of input parameters and key performance parameters in GGMR was discussed in a recent study [15]. PCB flow rates were first predicted by the GMR model and were then fed into the GGMR model representing normal operation as one of the input parameters.

We selected the set of inputs to include the zone air temperature (ZAT), average PCB control valve position, chilled water valve position, double façade temperature, and predicted flow rate. The Spearman's

Table 1
Common faults of a passive chilled beam system.

#	Fault Severity Level	Component	Fault Label
1	Stuck valve: 0%	Chilled beam control valve	CBV_0%
2	Stuck valve: 50%		CBV_50%
3	Stuck valve: 100%		CBV_100%
4	Stuck valve: 0%	HX chilled water control valve	CHV_0%
5	Temperature sensor offset: +9 °F	Supply water temperature sensor	SWT+9F
6	Temperature sensor offset: 9 °F		SWT-9F
7	Temperature sensor offset: +5 °F	Zone air temperature sensor	ZAT+5F
8	Temperature sensor offset: 5 °F		ZAT-5F
9	Stuck at a constant speed: 50%	VFD for supply water pump	VFD_50%

rank correlation coefficients were calculated between two independent variables. The maximum correlation coefficient is 0.197 between zone air temperature and flow rates and the other coefficients are in the range of 0 and 0.006. Three-month (April 1, 2021–June 30, 2021) trend data for the normal operation were used to train the baseline GGMR model. The optimum number of components was 30. The learning rate was kept at a constant of 0.01.

Trend data between July 1, 2021 and July 26, 2021 were used to test the GGMR model. Statistical analysis results for the prediction of cooling rates using the test data are a coefficient of variance of the root mean square error (CV-RMSE) of 20.7%, a mean bias error (MBE) of 0.5%, and an R-square of 0.909. A snapshot of the predicted and measured hourly PCB cooling rates for Living Lab 1 is shown in Fig. 3. The predicted hourly cooling rates agree well with the measured PCB cooling rates although there were some significant fluctuations throughout the day for hourly cooling rates.

5. Fault symptom analysis

In this section, various performance indices for capturing fault behaviors were created and analyzed for validating the implementation of each fault and for the preparation of feature selection in Section 6 *Feature Selection*. Symptoms or behaviors for each fault are presented in this section.

Trend data (5-min intervals) for zone environmental conditions and PCB performance data of Living Lab 1 were collected through the building control system. The 15 trended parameters through the building control system illustrated in Fig. 4 include 1) supply water pump speeds (VFD), 2) PCB supply water temperatures (SWT), 3) PCB return water temperatures (RWT), 4–6) control valve signals for north (CVN), central (CVC), and south (CVS) chilled beam banks, 7) differential pressure of the PCB loop (ΔP), 8) PCB system flow rates (Flow), 9) chilled water control valve signals at the heat exchanger side (CVHXC), 10) heating hot water control valve signals at the heat exchanger side (CVHXH), 11) zone air temperature (ZAT), 12) zone air relative humidity, 13) façade space temperature, 14) outdoor air temperature, and 15) outdoor air relative humidity. In addition to the 15 trended parameters, two parameters including the predicted PCB cooling rates and the predicted flow rates from GGMR models were also taken into consideration.

Various performance indices were constructed based on the deviations between baseline model predictions (expectations) and measurements, the relative differences between parameters, and/or extracted features of parameters such as cumulative differences or the change of parameters over time. They were evaluated and analyzed for various faults. The fault feature vector i is composed of the selected performance indices from Section 6 *Feature Selection*. They do not primarily rely on absolute values from a specific system. Therefore, these derived performance indices can improve the transferability of the FDD algorithms from one system to other.

The specific performance indices include 1) deviations between predicted and actual cooling rates ΔC, 2) deviations between predicted and actual flow rates Δf, 3) deviations between loop pressure and its setpoint Δp, 4) deviations between zone air temperature and its setpoint ΔT_{ZAT}, 5) temperature differences between supply and return water temperature $\Delta T_{r\text{-}s}$, 6) temperature differences between supply water temperature and zone air temperature setpoints ΔT_{sZAT}, and 7) change rates of supply water flow S_f (Eq. (7)). The parameter S_f is defined as the difference between the maximum flow rate $_{f_{max}}$ and minimum flow rate $_{f_{min}}$ occurring over a 24-h moving window divided by the period (24 h). The cumulative sums (*cusum*) of ΔC, Δf, Δp, ΔT_{ZAT}, $\Delta T_{r\text{-}s}$, and ΔT_{sZAT} over a 24-h moving window were also considered as key performance indices. The cooling rates and flow rates of the PCB system were predicted on an hourly time interval. Thus, ΔC, Δf, *cusum*_ΔC, and *cusum*_Δf were calculated for each hour. These parameters were synchronized with other performance indices on a 5-min time interval assuming semi-steady state conditions of cooling rates and flow rates within each hour for feature selection, fault detection, and diagnosis. Among all the performance indices, ΔC, Δf, *cusum*_ΔC, and *cusum*_Δf have evolving features requiring baseline models for predicted energy use and flow rates evolve along with the changes in building systems and components over time.

$$S_f = \frac{(f_{max} - f_{min})}{time\ period} \tag{7}$$

The performance indices, including ΔC, Δf, Δp, ΔT_{ZAT}, $\Delta T_{r\text{-}s}$ and ΔT_{sZAT}, characterize performance at a given time and are subject to short-term dynamic fluctuations, while the moving cumulative sums filter the shorter-term dynamics, and better represent performance trend over time. Under faulty operation, deviation of system performance from the normal operation may not be significant or even consistent for individual data points due to uncertainties in measurements and external factors while the cumulative sums show cumulative effects over time and amplify the effects of faulty operations.

There were 25 days of measurement data for normal operation and about two days of measurement data for each fault at a 5-min interval. The data were divided into training and testing data sets. The training data included 3456 data points (12 days) of normal operation and 288 data points (1 day) for each of the implemented faults. The testing data included 3744 data points (13 days) of normal operation and 288 data points (1 day) for each fault type. For each data point, cumulative indices were calculated based on the sum of the deviation over a moving 24-h period (288 data points for 5-min data). The cumulative indices for each data point of the second day under a specific fault can be easily calculated. To calculate the cumulative indices for each data point of the first day of the same fault, the second day was treated as the day before the first day for cumulative indices calculation. In other words, the data of the deviation of each data point on the second day were used as the

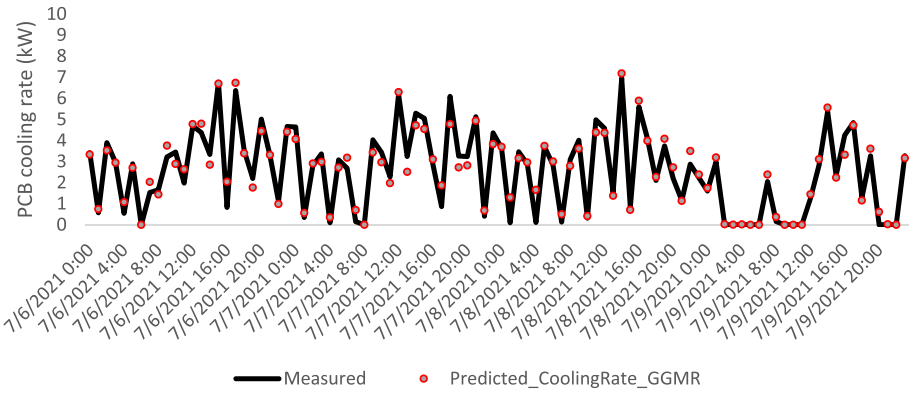

Fig. 3. The hourly PCB cooling rate prediction for testing data from Living Lab 1 measurement.

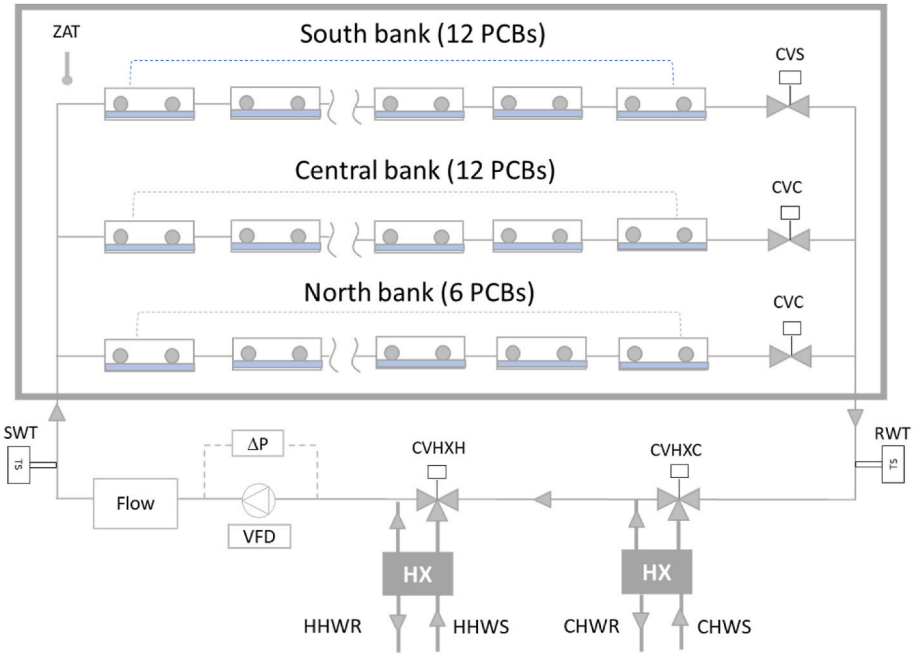

Fig. 4. The system diagram for Living Lab 1 passive chilled beam system with trended control and performance parameters. (VFD: supply water pump speeds; SWT: PCB supply water temperatures; RWT: PCB return water temperatures; CVN: control valve signals for north chilled beam bank; CVC: control valve signals for central chilled beam bank; CVS: control valve signals for south chilled beam bank; ΔP: differential pressure of the PCB loop; Flow: PCB system flow rates; CVHXC: chilled water control valve signals at the heat exchanger side; CVHXH: heating hot water control valve signals at the heat exchanger side; ZAT: zone air temperature).

data before the first day for cumulative indices calculation of the first day.

As an example, the cumulative sum performance index for flow rates (*cusum_Δf)* for all the operation scenarios is shown in Fig. 5 for all the training data including normal (3456 data points) and faulty operations (about 288 data points per fault).

The 24-h cumulative sums for each operation scenario stayed within a relatively narrow range except for the faults of CBV_50% and CHV_0%. This performance index has consistent and clear characteristics for every operation scenario. *cusum_Δf* is close to zero under normal operation. CBV_0%, CBV_50%, CHV_0%, and VFD_50% have large positive values of *cusum_Δf*. CBV_100% and ZAT-5°F have large negative values of *cusum_Δf*.

Fig. 6 shows the change rates of supply water flow rates for normal and faulty operation scenarios for all the training data including normal (3456 data points) and faulty operations (about 288 data points per fault). The larger the $_{sf}$ values, the more supply water flow rates fluctuate over time. It was found that $_{sf}$ was nearly zero for a stuck chilled beam valve fault (CBV_0%, CBV_50%, and CBV_100%).

Table 2 presents a qualitative summary of the behavior of the performance indices for normal operation and each implemented fault. In Table 2, all the performance indices except the change rates of supply water flow rate $_{sf}$ were presented as positive (+), negative (−), or zero (0) relative to the range of normal operation with the consideration of uncertainties. The uncertainty thresholds were discussed in Section 6.3 *Feature Selection Procedure*. The change rates of supply water flow $_{sf}$ are presented in Table 2 as absolute values. If any performance index for a specific fault showed inconsistent characteristics—sometimes positive (+), sometimes negative (−), or zero (0) in comparison with normal operation, the cell in Table 2 for the corresponding fault and performance index was left blank. A blank field indicates that the performance index should not be used as an indicator for diagnosing this specific fault because of the inconsistency of the behavior. It is also important to note that some performance indices, including Δp, ΔT_{ZAT}, $\Delta T_{r\text{-}s}$, and ΔT_{sZAT}, are all zeros and thus cannot be used to differentiate between normal operation and faulty operation.

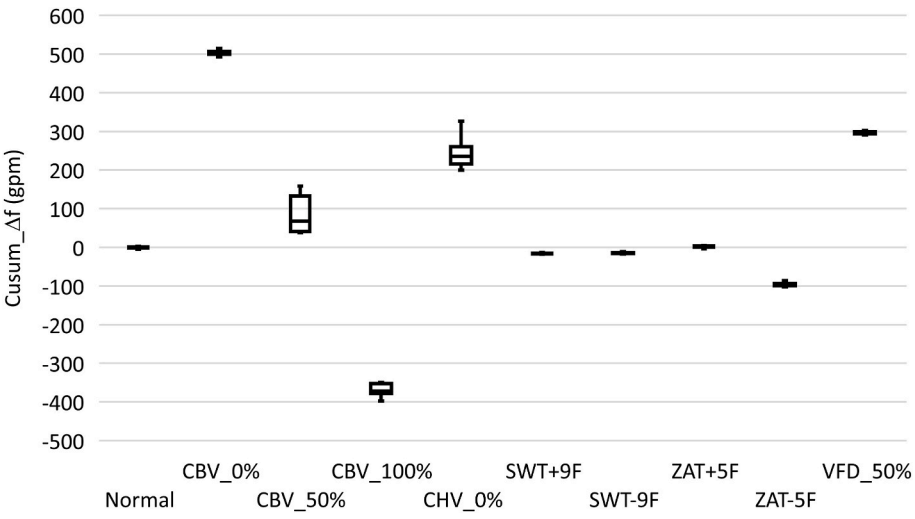

Fig. 5. *cusum_Δf* for normal operation and faulty operations (CBV_0%: chilled beam control valve stuck at 0%; CBV_50%: chilled beam control valve stuck at 50%; CBV_100%: chilled beam control valve stuck at 100%; CHV_0%: chilled water control valve stuck at 0%; SWT+9F: supply water temperature sensor offset +9 °F; SWT-9F: supply water temperature sensor offset -9 °F; ZAT+5F: zone air temperature sensor offset +5 °F; ZAT-5F: zone air temperature sensor offset -5 °F; VFD_50%: pump stuck at a constant speed 50%.).

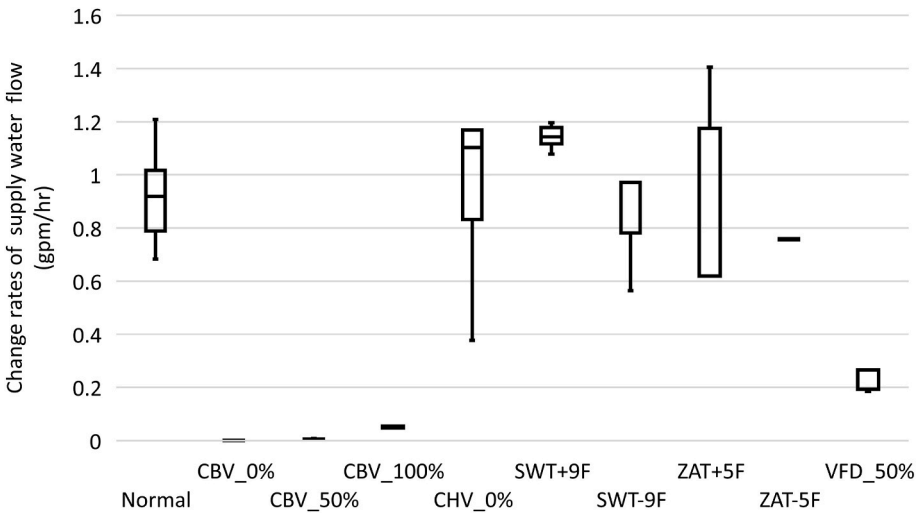

Fig. 6. Change rates of supply flow rates over 24 h for normal operation and faulty operations (CBV_0%: chilled beam control valve stuck at 0%; CBV_50%: chilled beam control valve stuck at 50%; CBV_100%: chilled beam control valve stuck at 100%; CHV_0%: chilled water control valve stuck at 0%; SWT+9F: supply water temperature sensor offset +9 °F; SWT-9F: supply water temperature sensor offset -9 °F; ZAT+5F: zone air temperature sensor offset +5 °F; ZAT-5F: zone air temperature sensor offset -5 °F; VFD_50%: pump stuck at a constant speed 50%.).

6. Feature selection

In this section, key performance indices created from Section 5 were selected and used to compose fault feature vector ***i***. A fault signature matrix ***S*** describing relationships between a set of parameters and the capability for diagnosing specific faults was developed using an optimization approach. Finding the best set of parameters for detecting and diagnosing common faults in HVAC systems can be defined as an optimization problem [19–21] that maximizes the level of diagnosability with analytical redundancy taken into consideration. The feature selection method presented in this paper was inspired by Fijany and Vatan's study [19], but was customized for this application.

6.1. Fault signature matrix

Fault signature matrix ***S*** is composed of two matrices: fault matrix ***F*** and parameter matrix ***P***. Fault matrix ***F***, parameter matrix ***P***, and fault signature matrix ***S*** have the same number of rows, representing the collections of performance features for faulty operations. Fault signature matrix ***S*** establishes a relationship between faulty operation scenarios in ***F*** and parameters in ***P***. Fault matrix ***F*** describes specific faulty operation scenario(s) with unsigned binary integers by assigning '1' to a specific column or columns representing specific faulty operation scenarios and '0' to the rest of columns in each row. The number of columns in fault matrix ***F*** represents the number of fault types to be diagnosed.

Parameter matrix ***P*** describes combinations of sensors/performance indices which can be used to diagnose faults in HVAC systems. These parameters include deviations between predictions (expectations) and measurements of a set of parameters (e.g., cooling rates, flow rates, temperatures) and/or extracted features of parameters (e.g., cumulative sums of the deviations, the change of parameters over time). ***P*** is an integer matrix with elements of '0' or a signed integer '-1/+1, −2/+2, …' If characteristics of a performance index are important for a specific fault or faults, the element in the parameter matrix ***P*** is a non-zero integer '-1/+1, −2/+2, …' If characteristics of a performance index are inconsistent over time, the element of the performance index is '0.'

For example, suppose we identified six parameters available for detecting common faults of one component in an HVAC system and there are five common faults. An example of the parameter matrix ***P***, the fault matrix ***F***, and the fault signature matrix ***S*** are shown as follows.

$$\boldsymbol{P}=\begin{bmatrix}0 & 1-1 & 0 & 0 & 0\\ 1 & 2 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\\ 0 & 1 & 0 & 1 & 0 & 0\\ 0 & 0 & 2 & 0 & 1 & 0\end{bmatrix};\ \boldsymbol{F}=\begin{bmatrix}1 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\end{bmatrix}$$

$$\boldsymbol{S}=[\boldsymbol{P},\boldsymbol{F}]=\begin{bmatrix}0 & 1-1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\ 1 & 2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0\\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1\end{bmatrix}$$

The first row of the fault signature matrix ***S*** indicates that a combination of the 2nd and 3rd parameters in the parameter matrix can be used to diagnose the first fault, a combination of the 1st and 2nd parameters in the parameter matrix are used to diagnose the second fault, the 6th parameter in the parameter matrix is used to diagnose the third fault, and a combination of the 2nd and 4th parameters in the parameter matrix is used to diagnose the fourth fault.

6.2. Optimal set of parameters to detect and diagnose faults

Finding optimal sets of parameters for detecting and diagnosing specific faults in HVAC systems can be defined as an optimization problem that attempts to maximize the level of diagnosability. The optimization approach attempts to minimize the number of parameters in the parameter matrix—with analytical redundancy taken into consideration—while maintaining the same level of diagnosability as with all possible parameters. We can formulate the problem of finding the optimal sets of parameters into an integer programming problem [19] as defined in equation (8).

The constraints of this integer programming optimization [19] include two conditions to be satisfied to keep the maximum level of diagnosability the same as with all possible parameters for the parameter matrix ***P***: 1) there is no all-zero row and 2) all rows are distinct. Integer programming is used to find a vector ***x*** that minimizes a linear function subject to linear constraints.

$$\min \boldsymbol{g}^T\boldsymbol{x}\ \ \text{Subject to}\ \boldsymbol{U}\bullet\mathbf{x}\le\mathbf{b} \tag{8}$$

In the defined integer programming problem, the element of a vector ***g*** is 1 if there are non-zero elements in the column of the parameter

Table 2
Characteristic table for performance indices of normal operation and faulty operation for PCB systems. (*ΔC*: deviations between predicted and actual cooling rates; *Δf*: deviations between predicted and actual flow rates; S_f: change rates of supply water flow; *Δp*: deviations between loop pressure and its setpoint; ΔT_{ZAT}: deviations between zone air temperature and its setpoint; $\Delta T_{r\text{-}s}$: temperature differences between supply and return water temperature; ΔT_{sZAT}: temperature differences between supply water temperature and zone air temperature setpoints; *cusum*: cumulative sums of *ΔC*, *Δf*, *Δp*, ΔT_{ZAT}, $\Delta T_{r\text{-}s}$, and ΔT_{sZAT} over a 24-h moving window.)

		ΔC	cusum_ΔC	Δf	cusum_Δf	S_f	Δp	cusum_Δp	ΔT_{ZAT}	cusum_ΔT_{ZAT}	$\Delta T_{r\text{-}s}$	cusum_$\Delta T_{r\text{-}s}$	cusum_ΔT_{ZAT}	$\Delta T_{r\text{-}s}$	cusum_$\Delta T_{r\text{-}s}$	ΔT_{sZAT}	cusum_ΔT_{sZAT}
1	Normal operation	0	0	0	0	abs(+)	0	0		0		0	0		0	0	0
2	Chilled beam control valve: stuck at 0%		+		+	abs(0)	0	–		0		0	0		–		+
3	Chilled beam control valve: stuck at 50%		0		+	abs(0)	0	+		0		0	0			0	0
4	Chilled beam control valve: stuck at 100%		–	–	–	abs(0)	0	–		0		0	0		0	0	0
5	HX chilled water control valve: stuck at 0%		+		+	abs(+)	0	–		0		0	0		0	0	+
6	Supply water temperature sensor offset: +9 °F		+	0	–	abs(+)	0	0		0		0	0		–	0	+
7	Supply water temperature sensor offset: 9 °F		–	0	–	abs(+)	0	0		0		0	0		+	0	–
8	Space temperature sensor offset: +5 °F			0	0	abs(+)	0	–		0		0	0			0	–
9	Space temperature sensor offset: 5 °F		+		–	abs(+)	0	–				0	0		0	0	+
10	Pump stuck at a constant speed: 50%	0			+	abs(+)	0	–		0		–	0		0	0	0

matrix $\boldsymbol{P}$ and otherwise 0; $\boldsymbol{x}$ is a binary vector to be found and determines the optimal sets of parameters; $\boldsymbol{U} = \begin{bmatrix} -M_1 \\ -M_2 \end{bmatrix}$; M_1 is the same as the parameter matrix $\boldsymbol{P}$; $\boldsymbol{M}_2$ is the matrix composed of the difference between a pair of rows in $\boldsymbol{M}_1$. Each row in $\boldsymbol{M}_2$ contains the absolute differences between each pair of rows $R_{ij} = |R_i - R_j|$ in matrix $\boldsymbol{M}_1$ if the two rows R_i, R_j in matrix M_1 are distinct, or is equal to any of the two rows R_i, R_j if R_i and R_j in matrix $\boldsymbol{M}_1$ are the same. To meet the two prerequisite conditions for the parameter matrix $\boldsymbol{P}$: constraints $\boldsymbol{M}_1 \bullet \boldsymbol{x} \geq \mathbf{1}$ (no all-zero rows) and $\boldsymbol{M}_2 \bullet \boldsymbol{x} \geq \mathbf{1}$ (all rows are distinct) are required where $\mathbf{1}$ is an all-one vector and $\boldsymbol{b}$ is a linear inequality constraint vector. In this optimization task, $\boldsymbol{b} = \mathbf{-1}$ is an all negative-one vector. Therefore, the constraint for this linear programming problem is $\begin{bmatrix} -\boldsymbol{M}_1 \\ -\boldsymbol{M}_2 \end{bmatrix} \bullet \boldsymbol{x} \leq -\mathbf{1}$.

The linear programming solution can be further customized to select more favorable parameters/sensors with the consideration of other factors such as cost and high fidelity besides diagnosability. The elements of the vector $\boldsymbol{g}$ corresponding to less favorable parameters or sensors can be multiplied to a large weighting factor (greater than one). Then the specific elements in the vector $\boldsymbol{g}$ become the weighting factor multiplied by the number of non-zero elements in the corresponding columns of the parameter matrix $\boldsymbol{P}$. Therefore, the parameters/sensors with large weighting factors have less chance to be selected.

6.3. Feature selection procedure

The feature selection method described in Sections 6.1 and 6.2 was used to select features for detecting and diagnosing common faults of PCB systems. All the performance indices were considered in the feature selection with training data. The training data contained 3456 data points (12 days) of normal operation and 288 data points (1 day) for each implemented fault except the fault for the stuck chilled water control valve at a fully closed position (CHV_0%). The CHV_0% fault was used to test the evolving feature of the FDD algorithm as a new "unknown" fault and therefore data for this fault was not used for training. There are four key steps in feature selection.

Step 1. The values of performance indices for different operation scenarios (either normal or faulty operations) are analyzed and used to create a signed integer matrix of fault "symptoms".

The values of performance indices under different operation scenarios (either normal or faulty operation) are compared to the range of those based on the training data set for normal operation with the consideration of uncertainties. The normal operating range of a performance index was defined in Eq. (9) with the consideration of uncertainties.

$$z \in \textit{normal operation when } \min - \alpha \times std(z) \leq z \leq \max + \alpha \times std(z) \quad (9)$$

where z represents a performance index; min and max represent the minimum and maximum of the performance index under normal operation based on training data; *std(z)* is the standard deviation of the performance index under normal operation; α is a coefficient for adjusting the threshold. $\alpha \times std(z)$ represents the uncertainties of the threshold.

Different non-zero integers are used to differentiate characteristics for a performance index. For example, '-1' is used for cases where the value of the performance index is smaller than the value of the minimum of the performance index for normal operation subtracting the uncertainty threshold; '+1' is used for cases where the value of the performance index is larger than the value of the maximum of the performance index for normal operation with the addition of the uncertainty threshold; '+2' is used to represent when the value of the performance index is in the range of the performance index for normal operation with the consideration of the uncertainty threshold. In contrast to Table 2, a non-zero integer '+2' instead of '0' is used to represent the

characteristics of the performance indices that fall within the normal operating range. A performance index can still be a useful signature when the performance index under a specific faulty operation consistently behaves similarly to normal operations. Such a performance index can be used to differentiate the specific fault from other faults under which the performance index is out of the normal operating range. In this study, the integer matrix was generated based on the performance indices for faulty operation relative to normal operation except for the performance index—the change rates of supply water flow $_{Sf}$. Absolute values of $_{Sf}$ were considered by assigning '+2' for values smaller than 0.1 and '1' otherwise. The fault matrix was based on the ground truth of different operation scenarios (either faulty operation or normal operation).

Step 2. Any duplicate fault symptoms for the same fault or any fault symptoms with limited occurrence (e.g., five times or less) should be removed so that unique fault symptoms with high frequency are left for each fault. In this context, a fault symptom represents a particular pattern of signed integers for performance indices established in step 1. For all the training data in the case study, 39 symptoms representing different operation behaviors for normal or faulty operations were left after this process. The 39 symptoms were grouped into nine operation scenarios (normal operation and eight faulty operations, with stuck chilled water valve 0% as a new unknown fault). For example, after the duplicated symptoms or symptoms with less occurrence were removed, the fault CBV_ 50% had seven different unique symptoms left as summarized in Table 3. The highlighted performance indices are those having consistent characteristics for all seven symptoms. The characteristics of performance indices including ΔC, Δf, and $\Delta T_{r\text{-}s}$ were inconsistent over time.

Step 3. This step involves the construction of the parameter matrix $\boldsymbol{P}$. Each row of the matrix $\boldsymbol{P}$ contains a performance "signature" for a particular fault or for normal operation. The rows are initialized using fault symptoms from step 2 and then evolve through the optimization process. In initializing a row for a particular fault scenario, a single signature is established using the fault symptom results from step 2 by replacing any inconsistent fault features with a '0'. A '0' represents a performance index that should not be considered as a fault feature during optimization. Applying this logic to the example of Table 3, a unique performance signature for CBV_50% was determined and is given in Table 4.

Step 4. This step involves the application of linear programming to determine an optimal parameter matrix $\boldsymbol{P}$ as described in Section 6.2. The optimization process eliminates performance indices that have fault features assigned a "0" (i.e., leads to inconsistent performance) or that aren't needed in producing unique performance signatures that differentiate between fault cases (i.e., needed for unique fault diagnoses). However, in order to provide redundancy, any performance index that has non-zero and non-identical fault features across the different faults is added as additional columns to the final parameter matrix. For the PCB system case study, the selected performance indices determined through optimization using the training data were $_{Sf}$, *cusum_Δp*, and *cusum_*ΔT_{sZAT} based on optimization. Then, *cusum_Δf* was added for redundancy since *cusum_Δf* was non-zero and non-identical across the different faults in the parameter matrix $\boldsymbol{P}$. The key signatures of the selected performance indices for the eight different faults are summarized in Table 5. The selected performance indices were used to compose fault feature vector $\boldsymbol{i}$.

7. Fault detection and diagnostic results

This section develops the transfer function $\boldsymbol{H}$ for fault diagnosis of the passive chilled beam system and presents diagnosed results for both known and unknown faults. The four fault features presented in Table 5 represent normal operation and the eight known fault scenarios as well as the normal operation. As described earlier, 5-min interval data for the case study system were divided into training and testing data sets. The training data included 3456 data points (12 days) of normal operation and 288 data points (1 day) for each of the implemented faults. For the chilled water control valve at a fully closed position (CHV_0%) fault, the 288 training data points were used for applying the evolving learning FDD training method. The testing data included 3744 data points (13 days) of normal operation and 288 data points (1 day) for each fault type. The testing data were evenly divided into two data sets with equal numbers of data points: testing data for FDD without evolving and testing data for evolving transfer function $\boldsymbol{H}$.

7.1. Fault detection and diagnostic results for known faults

The training data set for normal operation and the eight known faults were used to train a GMR model as the transfer function $\boldsymbol{H}$. The testing data for the FDD algorithm without evolving covers normal operation (1873 data points) and nine faults including CHV_0% (144 data points for each fault).

It is important to note that the uncertainty thresholds defined in Eq. (9), Section 6.3 for determining whether a performance index behaves normally or is out of normal operating range affect the testing results for normal operation. Testing results for normal operation at different thresholds are summarized in Table 6. As shown in Eq. (9), α is a coefficient for adjusting the threshold. $\alpha = 2$ represents that the uncertainties of performance indices are two standard deviations of the performance indices. As the uncertainties increase from zero to two standard deviations, the accuracy for testing results of normal operations improved; false alarm rates for normal operations were reduced from 14.8% to 1.3% and the percentage of normal operation data categorized as an unknown operation was reduced from 25% to 0%. However, if the uncertainties were increased substantially, individual faults lost their fault signatures and the diagnosis accuracy was reduced.

Table 7 shows fault diagnosis results of testing data under various faulty operations for the chilled beam system. For each fault, there were 144 data points in this testing. Testing data of individual faults except for the fault for the chilled water valve stuck at 0% (CHV_0%) were diagnosed with 100% accuracy. The fault diagnosis results are consistent with the three different uncertainty thresholds ($\alpha = 0$, 1.0, and 2.0). As

Table 3
List of 7 unique fault symptoms for chilled beam valve stuck at 50% (CBV_50%). (ΔC: deviations between predicted and actual cooling rates; Δf: deviations between predicted and actual flow rates; S_f: change rates of supply water flow; Δp: deviations between loop pressure and its setpoint; ΔT_{ZAT}: deviations between zone air temperature and its setpoint; $\Delta T_{r\text{-}s}$: temperature differences between supply and return water temperature; ΔT_{sZAT}: temperature differences between supply water temperature and zone air temperature setpoints; *cusum*: cumulative sums of ΔC, Δf, Δp, ΔT_{ZAT}, $\Delta T_{r\text{-}s}$, and ΔT_{sZAT} over a 24-h moving window.)

ΔC	cusum_ΔC	Δf	cusum_Δf	S_f	Δp	cusum_Δp	ΔT_{ZAT}	cusum_ΔT_{ZAT}	$\Delta T_{r\text{-}s}$	cusum_$\Delta T_{r\text{-}s}$	ΔT_{sZAT}	cusum_ΔT_{sZAT}
2	2	1	1	2	2	1	2	2	2	2	2	2
2	2	1	1	2	2	1	2	2	2	1	2	2
2	2	2	1	2	2	1	2	2	2	2	2	2
2	2	2	1	2	2	1	2	2	2	1	2	2
2	2	−1	1	2	2	1	2	2	2	2	2	2
2	2	−1	1	2	2	1	2	2	2	1	2	2
−1	2	2	1	2	2	1	2	2	2	2	2	2

Table 4
Performance signature for chilled beam valve stuck at 50% (CBV_50%)(*ΔC:* deviations between predicted and actual cooling rates; *Δf*: deviations between predicted and actual flow rates; S_f: change rates of supply water flow; *Δp*: deviations between loop pressure and its setpoint; ΔT_{ZAT}: deviations between zone air temperature and its setpoint; $\Delta T_{r\text{-}s}$: temperature differences between supply and return water temperature; ΔT_{sZAT}: temperature differences between supply water temperature and zone air temperature setpoints; *cusum*: cumulative sums of *ΔC*, *Δf*, *Δp*, ΔT_{ZAT}, $\Delta T_{r\text{-}s}$, and ΔT_{sZAT} over a 24-h moving window.).

ΔC	cusum_ΔC	Δf	cusum_Δf	S_f	Δp	cusum_Δp	ΔT_{ZAT}	cusum_ΔT_{ZAT}	$\Delta T_{r\text{-}s}$	cusum_$\Delta T_{r\text{-}s}$	ΔT_{sZAT}	cusum_ΔT_{sZAT}
0	2	0	1	2	2	1	2	2	2	0	2	2

Table 5
Key signatures of selected fault features with eight different known fault types. (*Δf*: deviations between predicted and actual flow rates; S_f: change rates of supply water flow; *Δp*: deviations between loop pressure and its setpoint; ΔT_{sZAT}: temperature differences between supply water temperature and zone air temperature setpoints; *cusum*: cumulative sums of *Δf*, *Δp*, and ΔT_{sZAT} over a 24-h moving window.)

	cusum_Δf	S_f	cusum_Δp	cusum_ΔT_{sZAT}
Normal operation	2	1	2	2
Chilled beam control valve: stuck at 0%	1	2	−1	1
Chilled beam control valve: stuck at 50%	1	2	1	2
Chilled beam control valve: stuck at 100%	−1	2	−1	2
Supply water temperature sensor offset: +9 °F	−1	1	2	1
Supply water temperature sensor offset: 9 °F	−1	1	2	−1
Space temperature sensor offset: +5 °F	2	1	−1	−1
Space temperature sensor offset: 5 °F	−1	1	−1	1
Pump stuck at a constant speed: 50%	1	1	−1	2

Table 6
Testing results for normal operation (1873 data points) for different thresholds.

α	False Alarm	Unknown	Accuracy
0	14.8%	25.0%	60.2%
1.0	4.5%	8.3%	87.2%
2.0	1.3%	0.0%	98.7%

shown in Table 7, the testing data for unknown fault CHV_0% were all diagnosed as unknown/unseen states.

Table 8 compares signatures of fault features for the nine individual faults including the unknown fault CHV_0%. Built upon Table 5, Table 7 shows that the signature for the fault CHV_0% is unique in comparison with other faults and thus this fault was categorized as an unknown fault with the testing data. If a fault is unknown, the fault diagnostic algorithm returns an unknown ID (e.g. "0") different from any ID for normal operation or existing faults which represents neither normal operation nor existing faults.

7.2. Detection and diagnosis of an unknown fault

The fault chilled water valve stuck at 0% (CHV_0%) was used to test the evolving learning-based transfer function ***H*** for fault diagnosis. Based on the testing results, without evolving, the fault was diagnosed as an unknown fault. Then 288 data points for the fault CHV_0% were used for evolving learning FDD training to update the transfer function ***H***. Another testing data set including normal operation (1871 data points) and nine faults including CHV_0% (144 data points for each fault) were used to test the evolving feature of the FDD algorithm.

Table 8
Signature comparison of fault features with nine faults. (*Δf*: deviations between predicted and actual flow rates; S_f: change rates of supply water flow; *Δp*: deviations between loop pressure and its setpoint; ΔT_{sZAT}: temperature differences between supply water temperature and zone air temperature setpoints; *cusum*: cumulative sums of *Δf*, *Δp*, and ΔT_{sZAT} over a 24-h moving window.)

	cusum_Δf	S_f	cusum_Δp	cusum_ΔT_{sZAT}
Normal operation	2	1	2	2
Chilled beam control valve: stuck at 0%	1	2	−1	1
Chilled beam control valve: stuck at 50%	1	2	1	2
Chilled beam control valve: stuck at 100%	−1	2	−1	2
Supply water temperature sensor offset: +9 °F	−1	1	2	1
Supply water temperature sensor offset: 9 °F	−1	1	2	−1
Space temperature sensor offset: +5 °F	2	1	−1	−1
Space temperature sensor offset: 5 °F	−1	1	−1	1
Pump stuck at a constant speed: 50%	1	1	−1	2
HX chilled water control valve: stuck at 0%	1	1	−1	1

Table 7
Fault diagnosis results of testing data under various faulty operations for the chilled beam system.

Testing Data Diagnosis Results	CBV_0% (144 data points)	CBV_50% (144 data points)	CBV_100% (144 data points)	CHV_0% (144 data points)	SWT+9F (144 data points)	SWT-9F (144 data points)	ZAT+5F (144 data points)	ZAT-5F (144 data points)	VFD_50% (144 data points)
CBV_0%	100%								
CBV_50%		100%							
CBV_100%			100%						
CHV_0%				0%					
SWT+9F					100%				
SWT-9F						100%			
ZAT+5F							100%		
ZAT-5F								100%	
VFD_50%									100%
Unknown				100%					

Notes: Unknown: unknown/unseen state; CBV_0%: chilled beam control valve stuck at 0%; CBV_50%: chilled beam control valve stuck at 50%; CBV_100%: chilled beam control valve stuck at 100%; CHV_0%: chilled water control valve stuck at 0%; SWT+9F: supply water temperature sensor offset +9 °F; SWT-9F: supply water temperature sensor offset -9 °F; ZAT+5F: zone air temperature sensor offset +5 °F; ZAT-5F: zone air temperature sensor offset -5 °F; VFD_50%: pump stuck at a constant speed 50%.

The best matching component and the minimum Mahalanobis distance of the Gaussian mixture regression were used to evaluate the effectiveness of the evolving learning feature of the FDD algorithm. The best matching component is defined as the Gaussian component corresponding to the maximum non-zero probability density of each data. Mahalanobis distance M_d (Eq. (10)) measures the distance between multivariate data and a Gaussian distribution.

$$M_{d,j} = \sqrt{(d-\mu_j)^T \delta_j^{-1} (d-\mu_j)} \tag{10}$$

where d is the multivariate data point, μ_j represents the mean of the jth Gaussian component, δ_j represents the covariance matrix of the jth component.

Figs. 7 and 8 illustrate the minimum Mahalanobis distance and best matching component for individual faults before and after evolving learning for the FDD algorithm, respectively. Besides the unknown ID returned from the FDD algorithm, measurements of an unknown fault are featured with no matching component and a large minimum Mahalanobis distance. Before the evolving learning, the fault CHV_0% has no matching component and a large minimum Mahalanobis distance among Gaussian components (538.5). After the evolving learning, covariance matrices of Gaussian components were updated. The best matching component (component 9) was identified for the fault CHV_0% and the minimum Mahalanobis distance was less than 1.0. The accuracy of FDD testing results using the evolving test data set was 100% for all the individual faults.

8. Conclusion

This study represents the first in creating and applying an evolving learning-based FDD method for detecting and diagnosing common faults for HVAC systems. We used a passive chilled beam system to demonstrate the evolving learning-based FDD method. The FDD algorithm presented in this work accurately detected and diagnosed known faults and successfully diagnosed a new unknown fault after evolving. The evolving learning-based FDD method overcomes the limitation of the traditional FDD methods—that learn from training data obtained under limited operating conditions, after which they stop learning—by updating key parameters of the baseline model and the transfer function over time.

The case study application of the method involved nine common faults for a passive chilled beam system that was implemented by

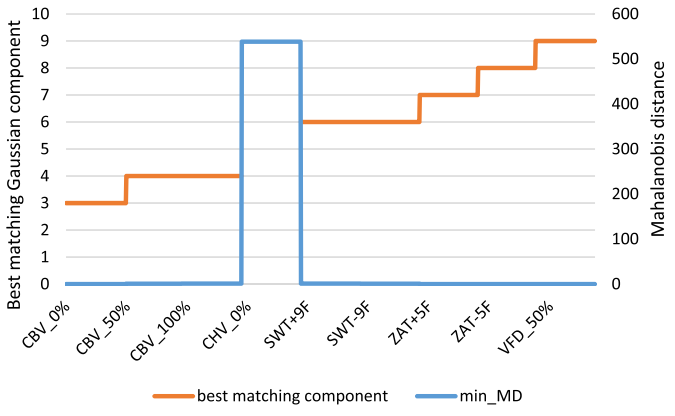

Fig. 7. The minimum Mahalanobis distance and best matching component for individual faults before evolving learning. (CBV_0%: chilled beam control valve stuck at 0%; CBV_50%: chilled beam control valve stuck at 50%; CBV_100%: chilled beam control valve stuck at 100%; CHV_0%: chilled water control valve stuck at 0%; SWT+9F: supply water temperature sensor offset +9 °F; SWT-9F: supply water temperature sensor offset -9 °F; ZAT+5F: zone air temperature sensor offset +5 °F; ZAT-5F: zone air temperature sensor offset -5 °F; VFD_50%: pump stuck at a constant speed 50%.)

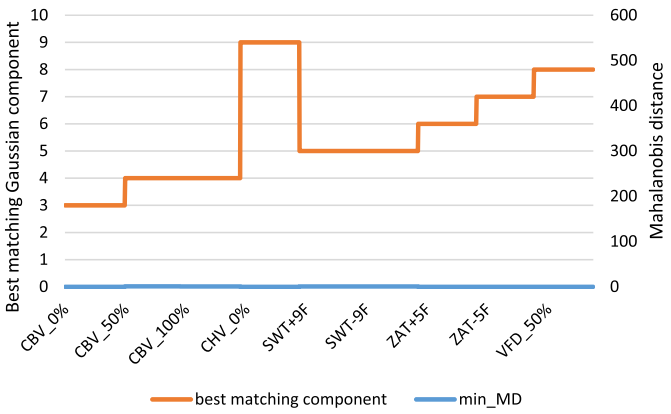

Fig. 8. The minimum Mahalanobis distance and best matching component for individual faults after evolving learning. (CBV_0%: chilled beam control valve stuck at 0%; CBV_50%: chilled beam control valve stuck at 50%; CBV_100%: chilled beam control valve stuck at 100%; CHV_0%: chilled water control valve stuck at 0%; SWT+9F: supply water temperature sensor offset +9 °F; SWT-9F: supply water temperature sensor offset -9 °F; ZAT+5F: zone air temperature sensor offset +5 °F; ZAT-5F: zone air temperature sensor offset -5 °F; VFD_50%: pump stuck at a constant speed 50%.)

overriding controls within the building control system. The performance of various operation scenarios for this system with the normal and faulty operations was trended for developing and testing the evolving learning-based fault detection and diagnosis algorithm. Eight of the nine faults were trained and tested as known fault types. These known faults were detected and diagnosed with an accuracy of 100%. The new fault was first tested without evolving and categorized as an unknown fault type. Measurements of an unknown fault are featured with the unknown ID returned from the FDD algorithm, no matching component, and a large minimum Mahalanobis distance with a value of 538.5. After evolving the transfer function by updating the key parameters of the Gaussian components, the best matching component was identified for the unknown fault, the minimum Mahalanobis distance was less than 1.0, and the unknown fault was diagnosed with 100% accuracy.

We developed a novel feature selection method for FDD that was demonstrated using the passive chilled beam system. To enhance the transferability of the FDD algorithm to other systems, we created and utilized generalized performance indices for fault discrimination, including deviations between predictions (expectations) and measurements, relative differences between parameters, and features extracted from other performance indices such as cumulative differences or changes of parameters over time.

An uncertainty threshold determining whether a performance index was within the range of normal operation influences false alarm rates. By increasing the uncertainty thresholds from zero to two standard deviations, false alarm rates for the normal operation were reduced from 14.8% to 1.3% and the percentage of normal operation data categorized as an unknown operation was reduced from 25% to 0%. However, if the thresholds are increased substantially, fault diagnosis accuracy is compromised due to the fact that individual faults lose their key signatures. Based on testing results, two standard deviations were selected as the uncertainty threshold to categorize performance indices into normal or faulty operations.

When a new state (either an unseen fault or unseen normal operation) is identified by the algorithm, the FDD algorithm requires interaction from an experienced building operator for the assessment and determination of the new state. The evolvement process needs external new input data. The algorithm requires sufficient data to evolve the transfer function and thus to successfully diagnose a new fault. In the future, both the evolving learning-based FDD method and the unique feature selection method can be applied for other high-performance HVAC system types as well as for conventional systems such as

variable air volume systems and rooftop units. The approach of creating performance indices for the passive chilled beam system can also be adopted for other HVAC systems. However, the set of performance indices and selected fault features should be unique for each type of HVAC system. The new FDD method was tested in a living laboratory in this study. Field implementations and continuous evaluation of the new FDD method should be carried out for demonstration and verification.

Credit author statement

Liping Wang: Conceptualization, Methodology, Formal analysis, Writing – original draft, Funding acquisition, James E. Braun: Conceptualization, Methodology, Writing – original draft, Sujit Dahal: Data curation.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Liping Wang reports financial support was provided by National Science Foundation.

Data availability

Data will be made available on request.

Acknowledgment

This study was supported by the National Science Foundation EPSCoR Research Infrastructure program under Grant No. 1929209. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] EIA. How much energy is consumed in U.S. residential and commercial buildings?. 2019. Available from: https://www.eia.gov/tools/faqs/faq.php?id=86&t=1.
[2] EIA. *Electricity-Data-Summary*; 2017. Available from: https://www.eia.gov/electricity/data.php#summary.
[3] EIA. *U.S.*. Energy-related carbon dioxide emissions. Available from: https://www.eia.gov/environment/emissions/carbon/pdf/2015_co2analysis.pdf; 2015.
[4] Katipamula S, Brambley MR. Review article: methods of fault detection, diagnostics, and prognostics for building systems–A review, Part 1. HVAC R Res 2005;11(1):3–25.
[5] Rossi TM, Braun JE. A statistical, rule-based fault detection and diagnostic method for vapor compression air conditioners. HVAC R Res 1997;3(1):19–37.
[6] Norford LK, et al. Demonstration of fault detection and diagnosis methods for air-handling units (ASHRAE 1020-RP). HVAC R Res 2002;8(1):41–71.
[7] Jin W, Li S. Tools for evaluating fault detection and diagnostic methods for air-handling units. ASHRAE RP-1312); 2012.
[8] Wang L, Haves P. Monte Carlo analysis of the effect of uncertainties on model-based HVAC fault detection and diagnostics. HVAC R Res 2014;20(6).
[9] Schein J. Results from field testing of embedded air handling unit and variable air volume box fault detection tools. Gaithersburg, MD: National Institute of Standards and Technology; 2006.
[10] Summers H, Hilger C. fault detection and diagnostic software. Pacific Gas and Electric Company; 2012.
[11] Chen C, et al. Operating characteristics and efficiencies of an active chilled beam terminal unit under variable air volume mode. Appl Therm Eng 2015;85:71–9.
[12] Hu R, Niu JL. Operation dynamics of building with radiant cooling system based on Beijing weather. Energy Build 2017;151:344–57.
[13] Wang L, Greenberg S. Window operation and impacts on building energy consumption. Energy Build 2015;92:313–21.
[14] May-Ostendorp PT, et al. Experimental investigation of model predictive control-based rules for a radiantly cooled office. HVAC R Res 2013;19(5):602–15.
[15] Wang L, Braun J, Dahal S. An evolving learning method —growing Gaussian mixture regression—for modeling passive chilled beam systems in buildings. Energy Build 2022:268.
[16] Bouchachia H, Vanaret C. Incremental learning based on growing Gaussian mixture models, vol. 2; 2011.
[17] Sung HG. Gaussian mixture regression and classification. PhD thesis.. Houston, Texas: Rice University; 2004.
[18] Hershey JR, Olsen PA. Approximating the Kullback leibler divergence between Gaussian mixture models. In: IEEE international conference on acoustics, speech and signal processing - icassp. vol. 7; 2007. 2007.
[19] Fijany A, Vatan F. New efficient algorithm for analyzing and optimizing the system of sensors. IEEE; 2006.
[20] Nejjari F, Sarrate R, Rosich A. Optimal sensor placement for Fuel Cell System diagnosis using BILP formulation. In: Control & automation (MED). 2010 18th Mediterranean Conference on; 2010.
[21] Bagajewicz M, Fuxman A, Uribe A. Instrumentation network design and upgrade for process monitoring and fault detection. AIChE J 2004;50(8):1870–80.