FADED: FAult DEtection and Diagnostic System for HVAC Sensors in Commercial Buildings

Altynay Smagulova
University of California, Merced
asmagulova@ucmerced.edu

Alberto E. Cerpa
University of California, Merced
acerpa@ucmerced.edu

ABSTRACT

Commercial building faults account for 0.3-1.8 quadrillion BTU of primary energy annually in the US. Effective fault detection and diagnosis (FDD) methods rely heavily on sensors as the primary source of data. However, sensor faults, which occur when sensors cease to function or produce erroneous readings, can lead to energy inefficiency and user discomfort. Thus, ensuring sensor reliability is crucial for maintaining energy efficiency and occupant comfort. In this paper, we introduce FADED, a Fault Detection and Diagnostic System for HVAC Sensors in Commercial Buildings. FADED differs from traditional FDD methods by deploying an *ephemeral and parallel infrastructure* to provide additional data points for detecting and identifying sensor faults at the Variable Air Volume (VAV) zone level. When the Facilities department receives complaints, technicians use a smartphone application integrated with airflow and temperature sensors to collect data through a simple walk-through. The collected data, combined with Building Monitoring System (BMS) readings, are analyzed using machine learning algorithms to identify and diagnose temperature and airflow sensor faults in VAV units. The system enhances sensor reliability by accurately classifying and quantifying faults, aiding technicians, conserving energy, and ensuring comfort. FADED utilizes real building data and achieves classification accuracies of 99.57% for zone temperature, 98.67% for supply temperature, and 99.01% for airflow faults, with minimal false alarms. It can also correct common sensor faults in real-time with low mean absolute error, paving the way for self-healing buildings. The proposed system was tested in a University building, successfully classifying operational sensors and identifying sensor faults in rooms with poor ventilation and high temperatures, demonstrating its practical utility.

CCS CONCEPTS

• **Computer systems organization** → **Sensors and actuators**; *Redundancy*; • **Computing methodologies** → **Supervised learning**; • **Networks** → Network reliability.

KEYWORDS

Fault detection, Machine learning, HVAC systems, Smart buildings

1 INTRODUCTION

The building sector accounts for more than 40% of primary energy consumption in the US, with about 50% of this energy used by heating, ventilation, and air conditioning (HVAC) systems [11]. HVAC systems ensure comfortable environments in commercial buildings by maintaining desired temperature levels through a network of sensors, actuators, and feedback controllers. Accurate sensor readings are essential for monitoring HVAC health and identifying potential issues. However, mass-produced and inexpensive sensors may fail to provide reliable information.

In this paper, we define a sensor fault as a situation where a sensor either stops functioning or generates inaccurate readings, such as random noise, bias, or drift caused by prolonged operation and challenging environments [24]. Sensor failures due to aging, flaws, or environmental factors hinder effective HVAC control, leading to increased energy consumption [42]. Commercial building faults account for 0.3-1.8 quadrillion BTU of primary energy annually in the US [37]. Effective fault detection and diagnosis (FDD) relies heavily on sensor reliability, which is crucial for conserving energy and improving occupant comfort in buildings. Consequently, there has been significant research interest in *sensor* fault detection and diagnosis for HVAC systems in recent years [18].

Currently, sensor faults are addressed by technicians manually. They analyze sensor data from a Building Management System (BMS). They may use additional reference sensors or diagnostic tools to inspect the HVAC system. The inspection, diagnostic, and calibration processes may take a few hours to a full day or longer. Generally, it depends on the competency of the technician and an insufficiently trained workforce impedes energy efficiency in buildings [10]. Particularly, inadequate skills and knowledge can lead to poor decision-making and misinterpretation of system conditions, resulting in recurrent failures and additional repair work, ultimately leading to higher costs [39]. Additionally, these labor-intensive methods can disrupt the work environment and prolong the procedure, potentially compromising occupant comfort [4].

This paper aims to develop a fault detection, diagnosis, (FDD) and reconstruction system for the Variable Air Volume (VAV) main sensors for each zone in a building that are used by the BMS to perform HVAC control, i.e. supply and zone temperature and airflow sensors using real data traces. The goal is to facilitate the maintenance crew's job when debugging zone issues with BMS faulty sensors. While the majority of FDD systems are based on analyzing BMS data, we propose to take a radically different approach by deploying an *ephemeral parallel sensing infrastructure*. The idea is not simply to compare sensor values between the BMS and FADED directly, but rather to test each zone, in a short time, and compare the two sensing data streams under *dynamic conditions* using data-driven techniques to detect and classify sensor faults.

To test the effectiveness of FADED, we introduce five different sensor faults of various severity in real production building sensor streams and test different multi-class machine learning classification models using real data from the BMS. Perhaps more importantly, once a fault is classified, we present a way to estimate the parameters of the sensor's error model. This enables us to have a full functional description of the fault, allowing us to reverse the process and estimate with high accuracy the correct sensor value as

if the sensor was not faulty. The implications of this are significant, as it can open new avenues of research on "self-healing" buildings, systems that can adapt to faulty components similarly to biological systems in nature. The utilization of fault parameterization is a commonly employed practice within the field of structural health monitoring [22]. By using real data from production buildings, we show that FADED is a state-of-the-art framework for sensor fault detection, diagnosis, and quantification that enables HVAC technicians to identify sensor faults, thereby optimizing energy usage in buildings and enhancing occupant comfort quality.

We would like to highlight the contributions of the paper:

1. We propose FADED, a novel fault detection, diagnosis, and quantification system that creates an ephemeral parallel sensing infrastructure in each zone to identify faults. FADED enables technicians to effortlessly gather sensor data during the investigation of a malfunctioning zone through a mobile application, streamlining the process, while also automating the analysis and decision-making procedures. No study, to our knowledge, has developed a similar end-to-end fault detection, diagnosis, and quantification system.

2. We use ML models for back-end fault detection, using a multi-class Light Gradient Boosting Machine (LGBM) model. The classifier is pre-trained using known sensor fault models and real data from buildings. It can detect common sensor fault types such as bias, drift, precision degradation, gain, and complete failure.

3. We show that FADED can estimate the parameter of the fault for the reconstruction and eventual fault correction in software. The findings from this study might be crucial for the development of "self-healing" buildings, where the sensor readings can be corrected in the BMS without the need to replace faulty sensors.

4. We test FADED by doing real-time experiments in a large production University building using commercial sensors and a real BMS system, verifying the operation of normal and faulty sensors and reporting the faults found in many misbehaving zones to the University facilities' crew.

2 RELATED WORK

In this section, we discuss some work done in FDD for HVAC systems. FDD locates the causes of operational failures as well as diagnoses such faults [8]. Condition-based, behavior-based, and outcome-based faults are the three types of faults defined in FDD. In our work, we emphasize the latter fault, which occurs when a measured outcome differs from a reference outcome [8]. Kim et al. [21] prioritize sensor faults as a top concern due to their significant energy impact and highlight the limited literature addressing them despite this emphasis. In this paper, we will discuss related works that use a data-driven approach to address sensor FDD [16].

Liu et al. [25] use the one-dimensional convolutional neural network (1-D CNN) and WaveCluster clustering analysis to extract features from raw data and analyze the extracted features respectively to identify temperature and water flow bias faults in Air Handling Unit (AHU) using simulated data. However, authors report that the WaveCluster clustering algorithm's limitations include its reliance on hyperparameters like grid density and outlier threshold, which can lead to suboptimal settings and inaccurate FDD results. On the contrary, FADED requires minimal hyperparameter tuning, enhancing its implementation efficiency and generalizability.

Ng et al. [32] uses the Bayesian method to detect bias faults in water flow and temperature sensors in a central chiller plant. The proposed method is tested on real and simulated data, it maintains functionality even when there is missing data. However, the proposed method relies on thermo-physical relationships among water temperatures in the piping system and, thus, fails when the same bias is simultaneously introduced to all sensors. The same issue will arise if we opt for the Bayesian method to detect VAV sensor faults. Additionally, modeling the dynamics in our case may necessitate occupancy counts in the zone, thus posing challenges.

In [28], the authors use physical relationships between all the sensors in ventilation units to generate a set of virtual sensors. They use both linear and nonlinear regression models and establish a confidence interval between the model error's range. The fault is detected when a data point deviates from the confidence interval. Virtual sensors are economical and can be a great substitute for expensive and impractical physical sensors [29]. Gao et al. [13] detect sensor failure by identifying the absolute deviation produced by contrasting the output of virtual sensors with the actual measured value. The virtual sensors are generated using the Long Short-Term Memory (LSTM) model. Conceptually, the main distinction of our approach is that we introduce real sensing redundancy into the system for direct comparison rather than using correlations in the system. Additionally, we control the data collection process with a human-in-the-loop. The HVAC technician changes BMS data to add variability to the data and thus obtains accurate results [41].

Extensive research has shown that traditional machine learning algorithms exhibit a high level of efficiency in detecting faults in different HVAC units. Additionally, they require fewer parameters and computational resources compared to transformers or neural networks. In [30], authors use a Support Vector Machine (SVM) to detect both sensor and actuator faults in air handling units. The performance of the model was verified on simulation data. Similarly, Zhang et al. [43] developed eXtreme Gradient Boosting (XGB) to detect and diagnose faults for the screw chillers. In [14], authors use the K-Nearest Neighbor (kNN) technique to detect drift faults in the chiller system. These three machine-learning techniques will be used as a benchmark for our system.

3 HOW DO HVAC SYSTEMS WORK?

Before proceeding to explain FADED, it is important to explain the basics of how an HVAC system controls the temperature and ventilation levels of a zone, and what sensors are used as input to the BMS system. The thermodynamics of a zone can be modeled using a resistor-capacitor system [41]. A differential equation is used to show the thermal model of a zone as in Eq. 1:

$$M_n \frac{dT_{z_n}}{dt} = (T_{s_n} - T_{z_n}) c_p m_{z_n} + Q_n + \frac{T_o - T_{z_n}}{R_{n,o}} \tag{1}$$

The left term represents zone temperature change, where T_{z_n} is the zone temperature and M_n is the zone's capacitance. The first term on the right side of the equation is the heat transfer from the HVAC system through the zone's supply vents and T_{s_n}, c_p and m_{z_n} are supply temperature, the specific heat capacity of air and mass airflow respectively. The second term, Q_n refers to the total load in the area, which is the total thermal load produced by people and equipment. The third term describes the inflow of heat from the outside (external to the building) into the zone and T_o and $R_{n,o}$

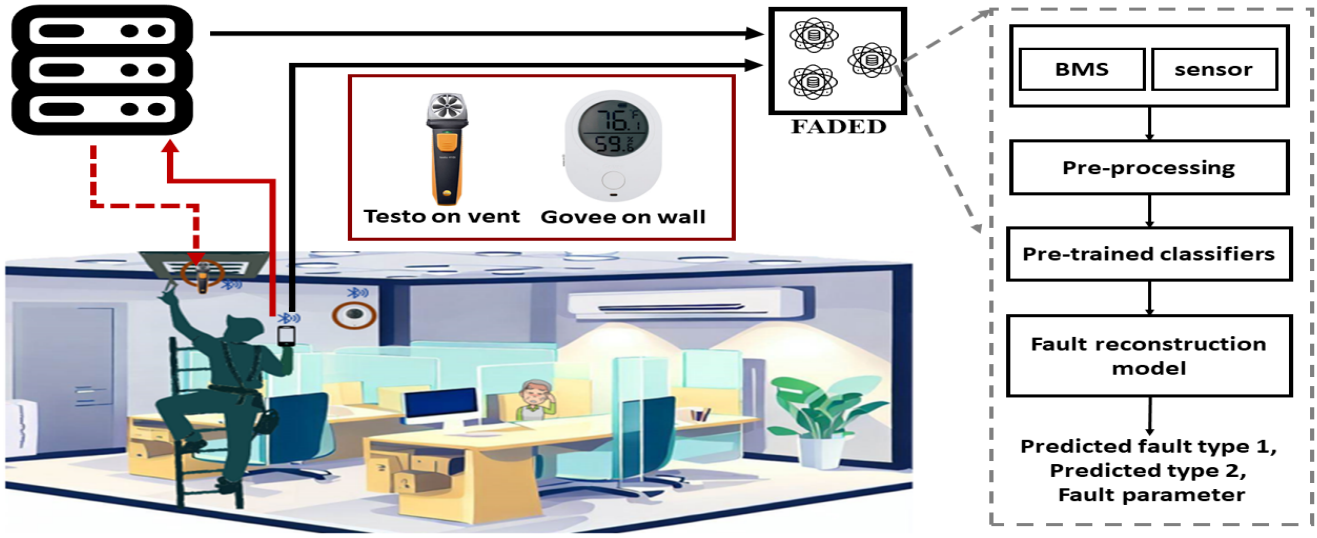

Figure 1: FADED System Overview

are outside temperature and thermal resistance of the walls and windows between the outside and zone environment. Note that this simplified model does not include the thermal diffusion that can happen with adjacent rooms that share a wall. This simplification is reasonable for buildings where the temperature differential will be no more than 5 °C in the worst-case scenario (usually less than this). Outside temperatures may have a much larger differential, which is why the last term of the Eq. 1 is included.

As can be seen in Eq. 1 zone temperature, supply temperature, and airflow are the main variables that affect the thermodynamics of the building. HVAC systems control the supply temperature and airflow to modify zone temperature in accordance with control rules and setpoint criteria. A BMS acts as the central nervous system and facilitates monitoring and control of the HVAC system in all zones. It gathers information from sensors, meters, and equipment to provide real-time insights into a building's performance and implements a control logic to keep the zones working within specified temperature and ventilation limits. When a building sensor fails to provide correct input information, the BMS may fail to provide adequate comfort and energy efficiency in buildings.

4 SYSTEM OVERVIEW

FADED connects commercial-off-the-shelf (COTS) sensors with a mobile phone, allowing technicians to collect sensor readings from a troubleshooting zone with a quick walk-through. It is important to note that walkabouts are an essential part of established HVAC maintenance and inspection protocols [3].

The basic operation is as follows. An HVAC technician enters the room and deploys temperature and airflow sensors (§ 4.1) to increase the information source of data as shown in Fig. 1. Here, the Testo410i sensor is installed on the vent to record supply temperature and supply airflow. The Govee sensor, on the other hand, is positioned on the wall to record zone temperature; however, it also offers flexibility to be installed on the desk or in the middle of the room. Airflow and temperature sensor readings are transmitted to the phone via Bluetooth. Meantime, the technician sends a command to the BMS web interface using his phone (depicted as the red solid line in Fig. 1). This command will suddenly change the supply temperature, change the airflow, and/or a combination of both to change the zone temperature. The idea is to introduce variability in the system to improve data quality and model accuracy when collecting data. The data is collected for 30 minutes at 5-minute intervals (i.e. at times 0, 5, 10, 15, 20, 25, and 30 minutes), so we have 7 readings for the FADED sensors and 7 readings for BMS sensors. These 14 readings are sent to the FADED framework (a server in the cloud implementing the processing pipeline), where they are used as window features for our machine-learning models after some pre-processing steps (§ 5.1), i.e. filling in the missing values, and applying a Min-Max scaler. We have separate pre-trained models for each sensor (supply temperature, zone temperature, and supply airflow) to detect, diagnose, and reconstruct faults. The pre-trained classifier outputs (§ 5.2) the first and second predicted fault type, i.e., the first and second most probable case. Technicians can further use this second prediction class to make an informed decision about the sensor condition. Note that one of the classes is Normal (i.e. no-fault). In case of fault, the data is transmitted to the pre-trained fault reconstruction model (§ 5.3) to estimate the parameter of the predicted fault. Finally, the technician receives the first and second most probable fault types with the fault's parameter values. If the fault is predicted normal we do not need a fault reconstruction step.

4.1 Hardware

Our hardware consists of a mobile phone used by the technician, running our mobile app, along with two wireless COTS sensors. As shown in Fig. 1, the Govee Bluetooth temperature sensor is deployed on the wall, desk, or in the middle of the room to measure the zone temperature, which ranges from -20 to 60°C. This sensor allows us to collect temperature data on demand from the phone.

Additionally, Fig. 1 depicts the Testo 410i anemometer vane smart probe. This sensor is attached to the supply vent or air diffuser and may require multiple installations if there are several vents in a zone. However, in commercial buildings, single VAV systems and vents are more common than multi-VAV setups [5]. The Testo sensor

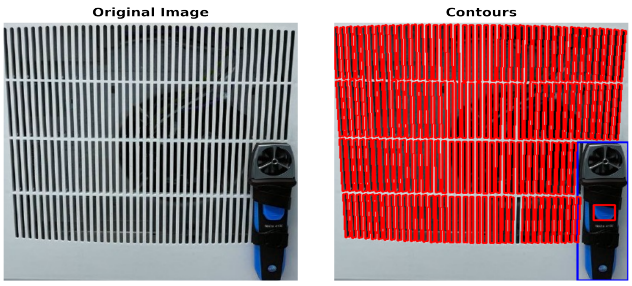

Figure 2: Vent Area Estimation

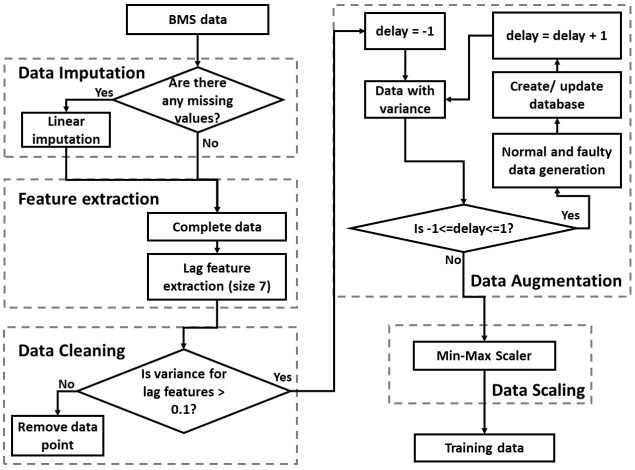

Figure 3: Training Data Generation

measures supply temperature (from -20 to 60°C) and air velocity (from 80 to 5,900 fpm) and also supports on-demand data collection. Traditional methods, such as deploying fixed zone-level sensors, are costly and pose significant integration challenges. Wireless deployments also face issues with system longevity and costs. In contrast, FADED sensors, overseen by humans and strategically reused, offer a cost-effective and durable solution, eliminating the need for extensive deployments.

4.2 Vent Area Estimation

Since our Testo 410i anemometer measures airspeed, we need the effective cross-section area of the vent being used to estimate flow. We developed a computer vision tool to facilitate duct area estimation using a cell phone. This tool estimates the area of the open vent with a Mean Percentage Absolute Error (MPAE) of 92.3%. Fig. 2 shows how the vent area was estimated using open holes in the vent. Here, we use our Testo 410i sensor as a reference object with a known size. The total area of the open holes is estimated in reference to the size of this object. The system uses grayscale conversion, Gaussian blurring, edge detection, morphological operations, and contour-detecting techniques to accurately find objects. The noise and intensity variations are removed by applying grayscaling and Gaussian blurring. Then, we use the Canny edge detection algorithm to detect edges in the blurred grayscale image. After that, we apply morphological operations such as dilation and erosion to reduce the noise and fill the missing gaps, thus enhancing contour detection. After detecting contours we remove the remaining noise using the thresholding method and find the reference object by clustering the contours. The proposed vent area estimation system was implemented using the *Open-CV* library.

Table 1: Fault Parameter Values

Fault Type	Zone temp (°C)	Supply temp (°C)	Airflow (*cfm*)
Noise (σ_1)	[0.1, 0.24]	[-4, 4]	[1, 2.75]
Bias (a)	[-5, 5]	[-5, 5]	[-65, 65]
Drift (b)	[-0.25, 0.25]	[-0.9, 0.9]	[-17, 17]
Gain (c)	[0.72, 1.28]	[0.72, 1.28]	[0.72, 1.28]
Complete failure (d)	[18, 36]	[18, 36]	[90, 270]
Precision degradation (σ_2)	[1.00, 2.35]	[3, 7.05]	[5.00, 27.50]

4.3 Data Selection

The American Society of Heating, Refrigeration, and Air-Conditioning Engineers (ASHRAE) offers a wide range of datasets to develop fault detection algorithms and evaluate their efficiency, such as ASHRAE RP-1312/1020/1043/1139 [9, 34, 36, 40]. However, these datasets are generally designed for actuator faults in HVAC systems and do not include the required sensor readings for each zone, such as supply airflow, supply temperature, and zone temperature. Therefore, ASHRAE datasets are not used for our sensor fault study. In this study, we use real BMS data from a University building for the past six months and real sensor readings. The building is a large production building with a single large plant floor of 7,735 m^2, with a single duct terminal reheat HVAC system that has 6 AHU loops and 154 zones in total.

4.4 Fault Modeling

As previously mentioned, it is hard to collect real faulty sensor data. Most of the studies available in the literature inject faults in real or simulated data as in [9, 34, 36, 40] and [15]. Moreover, these studies contain only bias faults for temperature sensors with a limited range of values (-4 °C, -2 °C, 2 °C, 4 °C). To the best of our knowledge, no other study has been done to detect different sensor faults and be sensitive to small fault error parameters. In general, it is difficult to discover and fix some faults due to the perceived complexity of measuring and modeling. We argue that such unrecognized faults may cause "sick building" syndrome (SBS), with many symptoms, including irritation in the eyes, nose, throat, headache, and fatigue, linked to indoor environmental and psychosocial conditions [17]. Some HVAC faults may be undetected and compromise the health and comfort of occupants failing to provide optimal ventilation [12] and temperature [33]. Such unrecognized faults not only cause health issues but also decrease the energy efficiency of buildings. 15% - 30% of energy wastage is due to HVAC failures in commercial buildings [6]. To address this in our work, we analyze a larger set of faults, including bias, drift, gain, precision degradation, and complete failure, which have been studied in [7, 19, 23, 26, 27, 44].

Fault types. Below, we provide a detailed description of the various types of faults. We introduced noise using a Gaussian distribution with a mean of zero and a variance of σ_1 to all faults, except for the precision degradation case. Here, we injected faults to *BMS readings* and X_t represent BMS reading discrete time steps t.

Bias. Bias fault occurs when sensor readings have slight variance and offset concerning the actual readings. The offset intensity is referred to as a (°C for temperature values and *cfm* for airflow values). Bias fault is defined as follows:

$$Bias = X_t + a + noise(N(0, \sigma_1)) \tag{2}$$

Drift. In a drift fault scenario, sensor readings exhibit deviations from actual values, often following a polynomial function. However, for simplicity, we will approximate this behavior with linear changes rather than polynomial adjustments. The rate at which the sensor readings deviate from the actual values is represented by the parameter b (°C for temperature values and *cfm* for airflow values) at each discrete sample recording t, where t takes on discrete values such as 0, 1, 2, etc. The drift fault can be described as follows:

$$Drift = X_t + b \times t + noise(N(0, \sigma_1)) \tag{3}$$

Gain. Gain is a multiplicative fault, and it affects the sensor readings by scaling them up or down. The parameter c (°C for temperature values and *cfm* for airflow values) represents the constant fault associated with this gain.

$$Gain = X_t \times c + noise(N(0, \sigma_1)) \tag{4}$$

Complete failure. In a complete failure, the sensor outputs a constant value and does not follow the trend in the actual sensor. Here, d (°C for temperature values and *cfm* for airflow values) represents the value at which the readings become stuck. The error model is defined as follows:

$$\text{Complete failure} = d + noise(N(0, \sigma_1)) \tag{5}$$

Precision degradation. Precision degradation refers to the addition of Gaussian noise with zero mean and random variance. Here, σ_2, the variance for precision degradation fault also incorporates the small noise and is quite larger than the σ_1, variance for noise. Precision degradation fault can be expressed as follows:

$$\text{Precision degradation} = X_t + N(0, \sigma_2) \tag{6}$$

These faults can be caused due to the failure of electronic components in the sensor, aging of the sensor, and other environmental factors. Fig. 4 illustrates how temperature sensors operate under different fault modes. Five faults have different patterns and behaviors. The error is constant for bias sensor fault, whereas in the case of drift the sensor measurements deviate and get worse over time. Moreover, in case of complete failure, the sensor output is different from actual values, while there is a slight noise in sensor readings for precision degradation. During a gain fault, the sensor reading deviates from actual readings with some multiplicative factor. Also, Fig. 4 shows that in certain scenarios, particularly when there is minimal variance in temperature changes, bias and gain faults yield comparable outcomes. Therefore, to effectively differentiate between these faults, we introduce variability to the system, as elaborated in (§ 4). Here, technicians adjust zone temperature, supply temperature, or supply airflow by issuing commands to the BMS via phone. Overall, sensor fault identification can vary in complexity depending on the nature and intensity of the faults.

Fault parameter values. Table 1 shows the range of fault parameter values and units of measurement. Each fault has ten parameters of different intensities within each range. The parameters for temperature and airflow sensors are determined by analyzing

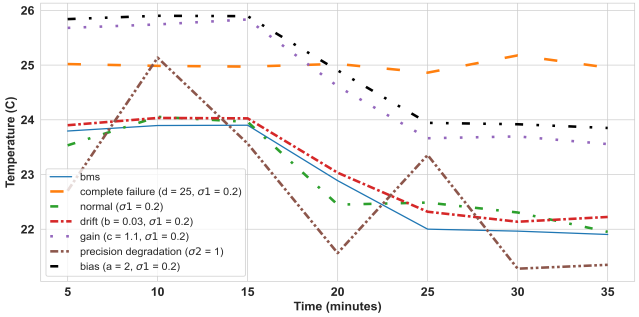

Figure 4: Behavior of different temperature sensor faults compared to a normal (non-faulty) sensor

sensor patterns from BMS data. As shown in Table 2, zone and supply temperature have different profiles. Zone temperature has a small range, whereas supply temperature can quickly increase to 51 $^{\circ}C$ when receiving a control signal from BMS to heat up the room. The temperature accuracy of the Govee H5072 and Testo 410i sensors are $\pm 0.3^{\circ}C$ [1] and $\pm 0.5^{\circ}C$ [2], respectively. The accuracy of the velocity sensor is ±(39.4 fpm + 2% of measured value), and the measurement is accurate within the range of 78.7 to 3937 fpm [2]. Thermistors, Precon type II, are used to monitor temperature in our building. The sensor has an accuracy of $\pm 0.36^{\circ}C$. So, to emulate these sensor inaccuracies we injected noise into our data. We added Gaussian noise with zero mean and different standard deviation parameters to improve the accuracy of our models (see Table 1, last row). Additionally, drift faults were injected with diverse starting points to model existing and new drift faults.

5 MODELING SCHEMES

This section details the machine learning classifier used for fault detection and classification, and the regressor used for fault parameter value estimation once a fault is detected and classified. We demonstrate their performance and compare FADED with the state-of-the-art data-driven FDDs.

5.1 Input Features and Training Data

Creating unified dataset. Real BMS data from a production building was used for training our fault detection data-driven models. HVAC sensor readings change gradually with time, and actuation rules are activated every 5 minutes, so there is no need to collect and process data every minute, and we collected data every 5 minutes. In this study, we combine the data streams from two different sources: the deployed FADED's sensor and BMS. We align the timestamps of the data points to create a unified dataset. The timestamps serve as a shared reference point to maintain temporal synchronization between related measurements from the BMS and FADED. So, we create separate datasets for zone temperature, supply temperature, and supply airflow by merging the two data streams. The temperature datasets encompass values ranging from 9.40°C to 51.60°C, while the airflow dataset ranges from 51 to 404 *cfm*. The pre-processing pipeline consists of four major steps: data cleaning, feature generation, scaling, and augmentation. Fig. 3 shows the training data preparation pipeline for each sensor.

Data imputation. Since this is a real building, we encountered a few instances of incomplete data due to sensor and BMS errors. The missing values were filled using the linear imputation technique.

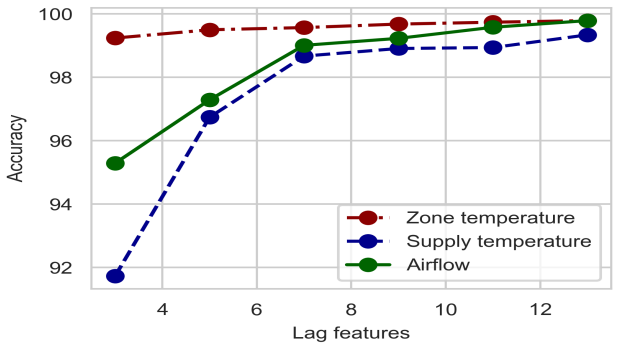

Figure 5: Model accuracy vs. the lag feature size

Linear imputation assumes that the relationship between variables is linear. Even though it is sensitive to outliers, linear imputation was chosen for handling missing data due to its alignment, simplicity, and ability to maintain data distribution and relationships. Other imputation techniques like k-Nearest Neighbors imputation may not be a good choice as they require more data. In [38], the resultant comprehensive linear model permits the realistic simulation of a building's thermal behavior at the granularity of individual zones, rendering it well-suited for utilization in Model Predictive Control (MPC) with a streamlined set of control variables, thus, zone temperature can be approximated with linear functions. When we apply step change to supply temperature and airflow, sensors' response to changes in the input variable exhibits a delayed and gradual adjustment due to hysteresis. This gradual response is akin to a low-pass filtering effect, which smooths out abrupt fluctuations in the sensors' output. As a result, over short time intervals, the sensors' behavior approximates linearity, meaning that their response appears as a linear transition rather than erratic jumps or oscillations [31]. This characteristic is crucial in applications where precise and stable measurements are required, as it minimizes rapid and undesirable variations in the sensor's readings.

Feature extraction. Lagged features in data capture temporal dependencies and trends, enabling models to learn from historical behavior by including past values. The time difference between each value is 5 minutes. Using both FADED's sensor and BMS readings helps to identify spatial dependencies and local variations. So, we extracted lag features to capture both temporal and local patterns in the specified time frame for FADED and BMS data. An optimal time frame was selected by analyzing the performance of the classification model at different lag window sizes. Fig. 5 illustrates how the accuracy of the classifier model (explained in § 5.2) changes with the increase in the number of features. There is a tradeoff between the number of lagged features included in the input vector and the time it takes to collect them. While including more features helps, this comes at a cost, since more features mean an HVAC technician running FADED will have to wait longer to get a determination per zone. The correct threshold may also depend on the accuracy assurances that a user may want. In our work, we can see that after 7 features that correspond to 30 minutes of readings, there is only a very small accuracy improvement for all sensor types. Thus, we have 14 features that describe the sequential dependencies in two-time series data (Building BMS and FADED sensor data). Let's denote the lag features for BMS as $B_1, B_2, \ldots, B_7$, and the lag features for sensors as $S_1, S_2, \ldots, S_7$. Then, the input vector U represents the lag features for both the BMS and FADED sensors and is defined as follows:

$$U = [B_1, B_2, \ldots, B_7, S_1, S_2, \ldots, S_7] \tag{7}$$

Table 2: Sensor Data Statistics

Sensor	Min	Max	Mean	Median	IQR
Supply temp (°C)	9.40	51.60	21.57	17.40	15.60
Zone temp (°C)	17.60	25.80	21.61	21.40	1.02
Airflow (*cfm*)	51.00	404.00	188.46	205.00	7.00

Data cleaning. Sensors have different change patterns as seen in Table 2. Supply temperature and airflow drastically increase and decrease, whereas zone temperature fluctuates around some value for a long duration of time. A higher interquartile range (IQR) means wider dispersion of data, whereas, a lower value means that data is clustered around the median. We selected time frames with higher variability to make our models accurate. We calculated the variance for each measurement using Eq. 8.

$$s^2 = \frac{1}{N-1}\sum_{t=1}^{N}(X_t - \bar{X})^2 \tag{8}$$

Here, X is the lag feature at sample t, $\bar{X}$ is the mean of the lag features, and N is the number of lag features. We have 7 lag features for both FADED sensors and BMS. We discarded data points where the variance of lag features is less than 0.1 to ensure high-quality training data. Model training and performance depend on effective feature engineering and data pre-processing in addition to the amount of the training dataset. In other words, the 'garbage-in, garbage-out' principle states that the quality of the data and the way it is prepared are equally important to the model's success.

Data augmentation. Zone temperature has a low variability score compared to supply temperature and airflow. After cleaning and discarding the features we had 690 data points for zone temperature and about 4000 data points for supply temperature and airflow. It is difficult to collect faulty data in real-time, so we artificially injected faults as it is done in [15].

In this study, we used BMS readings that were suggested as normal by the facility manager. We augmented our highly variable data by employing different fault types with various fault parameter values and adding some lags to the BMS data. The fault models used were explained in § 4.4. Overall, we got 4000 data points for zone temperature and 20000 data points for supply temperature and airflow. The training/validation and testing split was done by taking 75% of the dataset to train and validate the model, and 25% was used to test the models and report the system performance.

Data Scaling. Sensors have different ranges and sensitivity. In this study, we compare different benchmark algorithms used in related work, with some of them relying on distance, and using original features can result in unstable biased results. Scaling facilitates faster convergence of optimization algorithms. For each sensor, the scaler was fit on the training data to prevent information leakage and then applied to the whole data. In this study, we scaled our data using a Min-Max scaler. Min-Max scaler preserves the relationship between variables ensuring they fall within the range of [0, 1]. Unlike a Robust scaler, it keeps outliers, which are important for our fault detection analysis. Additionally, it does not require the normal distribution assumption like the Standard scaler.

Table 3: Comparison of separate and meta-classification models for faulty sensors

Model	Sensor	Accuracy (%)	DR (%)	FaR (%)
Separate	Zone temperature	**99.57**	99.83	**0.06**
	Supply temperature	**98.67**	**99.82**	**0.10**
	Supply airflow	**99.01**	**99.97**	**0.01**
Meta	Zone temperature	99.24	**99.97**	0.19
	Supply temperature	94.14	98.36	2.18
	Supply airflow	98.63	98.73	1.67

Table 4: Regression Model Performance Comparison

Model	Pearson correlation	RMSE
Linear	0.983	0.228
XGB Regressor	**0.998**	**0.188**

Also, in [35], among several techniques, the Min-Max scaler is identified as the most efficient, resulting in a 5-10% increase in accuracy scores across various algorithms.

$$X_{t\_scaled} = \frac{X_t - X_{min}}{X_{max} - X_{min}} \tag{9}$$

5.2 Fault Detection and Classification

FADED employs a multi-class Light Gradient Boosting Machine (LGBM) to detect and diagnose sensor faults. Our classification task encompasses six classes: bias, complete failure, drift, gain, precision degradation, and normal. We feed balanced data with 14 features to perform multi-class classification as shown in Eq. 7. For the output of the detection and classification model, we perform soft classification, resulting in a 6-class classification task with probabilities for bias, complete failure, drift, gain, precision degradation, and normal represented as p_b, p_{cf}, p_d, p_g, p_{pd}, p_n respectively. These probabilities represent the likelihood of belonging to each class. The output vector z is defined as follows:

$$z = [p_b, p_{cf}, p_d, p_g, p_{pd}, p_n] \tag{10}$$

This output vector represents the probabilities associated with each class for the given input sample. Each element of the vector corresponds to the probability of the sample belonging to the respective class. To provide the final detection and classification output, we select the class with the highest probability. The performance of the system is assessed in terms of accuracy, detection rate (DR), and false alarm rate (FaR) for each sensor. These parameters can be calculated from true positive (TP), true negative (TN), false positive (FP), and false negative (FN) classification cases. Eq. 11 shows the calculation for accuracy. It shows the portion of correctly classified cases. For an efficient model, we expect accuracy to be 1.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

In a multiclass classification scenario, accuracy measures the overall correctness of the model's predictions across all classes, while the detection rate focuses specifically on the model's ability to correctly identify faulty classes. To calculate DR we combined all the classes into *faulty* and *normal* instances. To ensure balanced representation we downsample the data associated with the *faulty* class to achieve parity with the *normal* class. The DR for our models is estimated as in Eq. 12. It shows the fraction of actual faults detected by our algorithm. We want it to be close to 1.

$$\text{DR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{12}$$

Eq. 13 estimates the FaR, the cases where sensors are falsely classified as faulty. A score closer to 0 means that the system is not misclassifying normal sensors as faulty. Here, we also use binary output for computation.

$$\text{FaR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{13}$$

Moreover, we implemented a single meta-classifier using scaled data from three sensors along with separate classifier models for each sensor. In order to build a single model, we balanced data by downsampling data for supply temperature and airflow to match the zone temperature's training size. Table 3 shows the results for these LGBM-based multi-class classifiers. Both models have high accuracy and DR and low FaR. However, separate models show higher accuracy for all three sensors. Particularly, it can classify supply temperature with an accuracy of 98.67% compared to the meta-model with 94.14%. Moreover, separate models make more accurate classifications for zone temperature and supply airflow. Also, it considerably decreases the false alarm rate for supply temperature and airflow by almost 10-20 times from 2.18% to 0.10% and 1.67% to 0.01% respectively. Overall, the classification performance of the models improves when we employ separate models for most of the sensors. Therefore, in FADED we use three separate models for our three sensors. Separate models help to better learn sensor behavior for our classification tasks.

5.3 Fault Reconstruction and Correction

Another contribution of this paper is the fault parameter value estimation and correction. For each fault detected by FADED classifier, we use regression models that are trained to find the model parameter for each fault type. The model is trained with un-scaled lag features and the parameters of the fault were used as label data. Each fault has its own pre-trained model for each sensor type. Table 4 illustrates a comparison of linear and XGB regression models. We use different regression models to find the parameter of the fault. As can be seen, the XGB Regressor is more accurate than the Linear Regression model and can capture more accurately the relationship between target and predicted values. It has a Pearson correlation of 0.998 compared to 0.983 from the Linear regression model. Most importantly, the XGB Regressor can reconstruct the fault parameter values with an RMSE of 0.188, whereas the linear regression makes less accurate estimations with an RMSE of 0.228. The HVAC technician can further use the predicted fault parameter value to assess the severity of the fault. The sensor can be replaced or if possible manually or automatically re-programmed in the BMS system to account for that fault. Knowing the full fault model, including the specific parameters of the fault is very significant, as this would allow the realization of self-healing buildings, similar to biological systems in nature that are capable of adapting and correcting to faults over time.

We use Pearson's correlation (P) and Root Mean Square (RMSE) to assess the performance of the reconstruction model. Pearson's

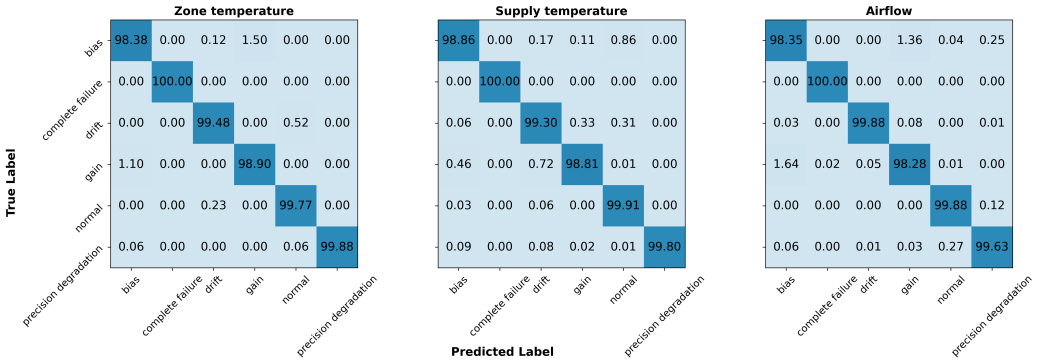

Figure 6: Confusion matrices for the zone temperature, supply temperature, and airflow.

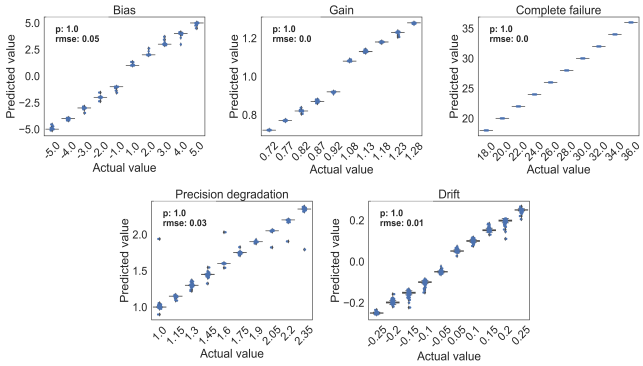

Figure 7: Fault reconstruction (zone temperature)

correlation is defined as follows:

$$P(X,Y) = \frac{\sum_{t=1}^{N}(X_t - \bar{X})(Y_t - \bar{Y})}{\sqrt{\sum_{t=1}^{N}(X_t - \bar{X})^2 \sum_{t=1}^{N}(Y_t - \bar{Y})^2}} \tag{14}$$

where N is the number of samples, t is discrete instance in time, X is the true parameters, Y is the predicted parameter and $\bar{X}$ and $\bar{Y}$ are mean values for true and predicted values respectively. The score ranges from -1 to 1 and shows a strong negative and positive linear correlation between two continuous variables. If the score is close to 0, it means that there is no correlation between true values and predicted values. RMSE estimates the differences between true values and predicted values. Particularly, it shows the average error value by measuring the square root of the mean of the squared differences between true and predicted values.

$$RMSE = \sqrt{\frac{\sum_{t=1}^{N}(X_t - Y_t)^2}{N}} \tag{15}$$

6 RESULTS AND ANALYSIS

In this study, we utilize three main types of data: real BMS data collected from a University building over the past six months, artificially generated fault-injected BMS data, and real sensor readings. To ensure robust training and testing of our model, we expand the dataset by augmenting the number of data points specifically for zone temperature from 690 to 4000, and for supply temperature and airflow from 4000 to 20000 data points as discussed in § 5.1.

For model evaluation and hyperparameter optimization, we employ a stratified 5-fold cross-validation technique. This approach ensures that each fold of the cross-validation maintains the same class distribution as the original dataset, thereby reducing bias and improving the reliability of our model evaluations. Additionally, we utilize Bayesian search for hyperparameter tuning, facilitated by the BayesSearchCV class from the *scikit-optimize* library. Bayesian optimization allows us to efficiently explore the hyperparameter space and identify optimal model configurations, enhancing the performance and generalization capabilities of our predictive model.

6.1 Performance Evaluation

Fig. 6 shows the confusion matrix for the classification of three sensors using FADED. The zone temperature and supply airflow have the best performance by correctly classifying almost all the test cases. Complete failure fault, precision degradation, and normal cases can be correctly classified almost with 100% accuracy. Bias and gain faults can be classified with an accuracy greater than 96%. Generally, these two faults are misclassified when there is not much variability in BMS and sensor readings, where the multiplicative gain and bias may have the same faulty effect. Also, normal cases can be confused with drift when the measurements drastically change in the given time frame due to changes made by the BMS control logic. Our test data contains small drift parameters, so it may be difficult to diagnose them due to noise and drastic fluctuations.

Fig. 7 shows the result of error reconstruction for zone temperature. As can be seen, complete failure and gain faults can be reconstructed accurately using this technique. Additionally, we find bias parameters with an average error of 0.05 $^{\circ}C$. Precision degradation and drift faults also have high correlation and lower average error values (0.03 $^{\circ}C$ and 0.01 $^{\circ}C$), but they still are in an acceptable range considering the temperature sensor sensitivity. It is worth noting that we are testing both new and existing drift faults. So, it can be difficult to estimate the actual drift parameter when you have different starting points.

Similarly, Fig. 8 shows the fault reconstruction results for the supply temperature. As it was mentioned in § 4.4, we used different fault parameter values for the supply temperature. The supply temperature has higher errors after reconstruction compared to the zone temperature. The variation in the results can be explained by the nature of these two sensors, the supply temperature changes rapidly within a short period, whereas, the zone temperature changes gradually over time. We have a high correlation for all the fault types except for precision degradation faults. However, we have more than 0.33 $^{\circ}C$ when we reconstruct bias and precision degradation

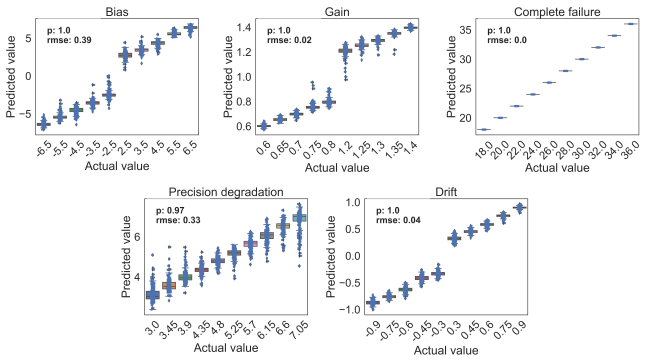

Figure 8: Fault reconstruction (supply temperature)

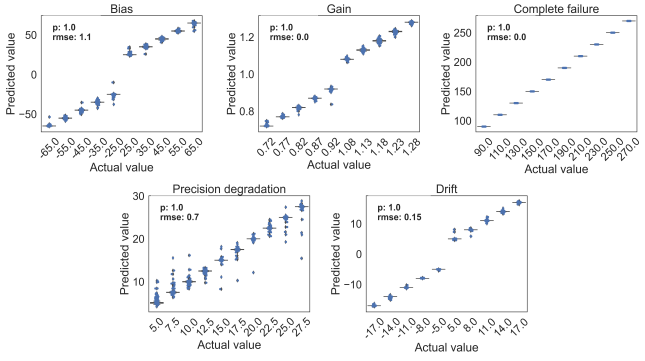

Figure 9: Fault reconstruction (airflow)

faults. The parameters of bias faults can be quantified with an error of 0.39 $^{\circ}C$. For precision degradation, the average error is 0.33 $^{\circ}C$, which still can be acceptable considering the dynamic nature of the supply temperature. The supply temperature varies from 9 $^{\circ}C$ to 52 $^{\circ}C$. So, the RMSE of 0.3 $^{\circ}C$ can be considered negligible.

Similarly, according to Fig. 9 the fault error model parameter values for the airflow sensor can be accurately estimated except for bias and precision degradation. Airflow sensors have different ranges, so the average error of 1 *cfm* still can be acceptable. Accurate estimation of fault parameter values can be difficult due to the dynamic nature of the sensor as in supply temperature.

Table 5 shows the performance of error parameter quantification for temperature sensors. Particularly, it describes how much we will be off from actual temperature if we adopt error reconstruction in terms of the temperature difference and temperature difference in percentage. The temperature difference is estimated using Mean Absolute Error (MAE) as shown in Eq. 16 and the temperature difference in percentage is calculated using Mean Absolute Percentage Error (MAPE) as shown in Eq. 17:

$$\text{MAE} = \frac{1}{N}\sum_{i=t}^{N}|X_t - Y_t| \tag{16}$$

$$\text{MAPE} = \frac{100}{N}\sum_{t=1}^{N}\left|\frac{X_t - Y_t}{Y_t}\right| \tag{17}$$

Zone temperature has a negligible MAE score for bias, drift, and gain faults, which is less than 0.2 $^{\circ}C$. However, the reconstruction of drift and precision degradation can result in a higher average error, introducing around 4% to 8% divergence from true non-faulty

Table 5: Temperature Fault Reconstruction Error

Fault	Zone ($^{\circ}C$)	Zone (%)	Supply (^{0}C)	Supply (%)
Bias	0.104	0.508	0.628	2.501
Complete failure	0.176	0.657	0.465	1.648
Drift	0.799	3.753	3.211	12.731
Gain	0.162	0.785	0.602	2.351
Precision degradation	1.647	8.024	4.811	23.132

Table 6: Airflow Fault Reconstruction Error

Fault	Supply (*cfm*)	Supply (%)
Bias	1.568	0.788
Complete failure	1.632	0.954
Drift	54.82	31.507
Gain	1.684	0.77
Precision degradation	30.353	15.848

values. On the other hand, quantification of drift and precision degradation can cause the offset of 3-5 $^{\circ}C$ after the reconstruction of supply temperature. On the contrary, bias, complete failure, and gain faults of supply temperature can be corrected with small errors.

Table 6 shows that after reconstructing the supply airflow sensor with predicted values we will experience minor alteration for bias, and complete failure and gain faults, which is less than 2 *cfm*. However, drift and precision degradation can cause alteration of 30-55 *cfm*. The MAE scores of these faults are very high causing a large offset from the true value despite quite a strong linear relationship between the actual and predicted values as shown in Fig. 9. Reconstructing the drift parameter is hard when there is high variability in lag features, where the difference between two lags is several hundredths or thousandth *cfm*. Additionally, different starting times, delayed data, and random noise exacerbate the drift parameter reconstruction for airflow sensors.

Overall, for bias, complete failure, and gain, we can predict error parameters accurately. However, other faults are hard to quantify due to their nature. For precision degradation, the number of readings is not sufficient to identify the mean of Gaussian noise. Moreover, the variability of the data impacts the accuracy of the reconstruction. For instance, supply airflow and temperature change drastically whereas zone temperature changes slowly over time. We may need more data for modeling supply temperature and airflow to capture their properties. Fig. 10 shows correctly classified fault cases for three sensors and their parameters. Fig. 11 shows common faults made by FADED. As was discussed previously, the algorithm fails to differentiate bias and gain when there is not much variance in the data. Also, the supply temperature fails to detect drift when there is a significant change in the data. In addition, the magnitude of the fault can also affect the classification results. Small precision degradation faults can be falsely classified as drift or bias with small magnitude. Overall, the system can accurately diagnose faults and quantify the parameters in most cases.

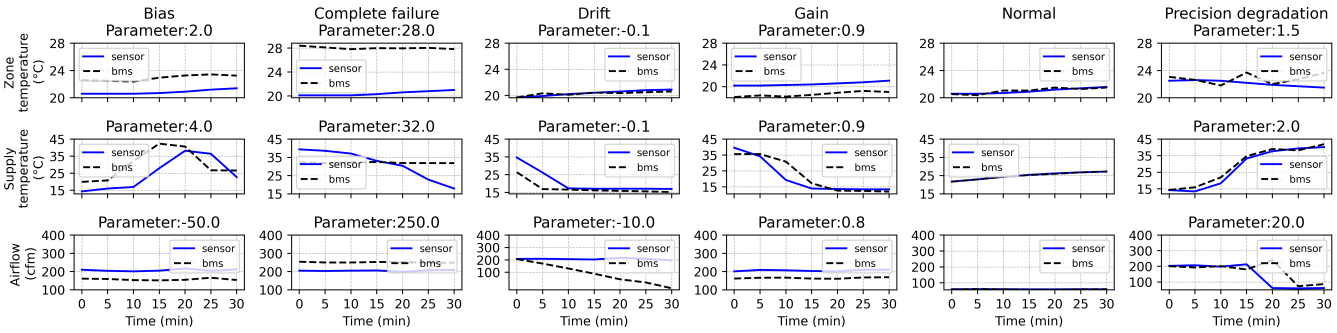

Figure 10: Correctly classified cases by FADED

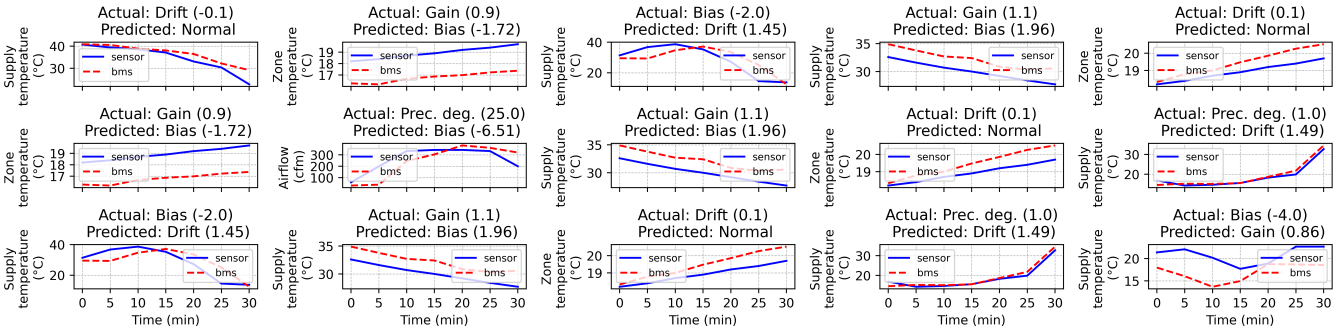

Figure 11: Misclassified cases by FADED

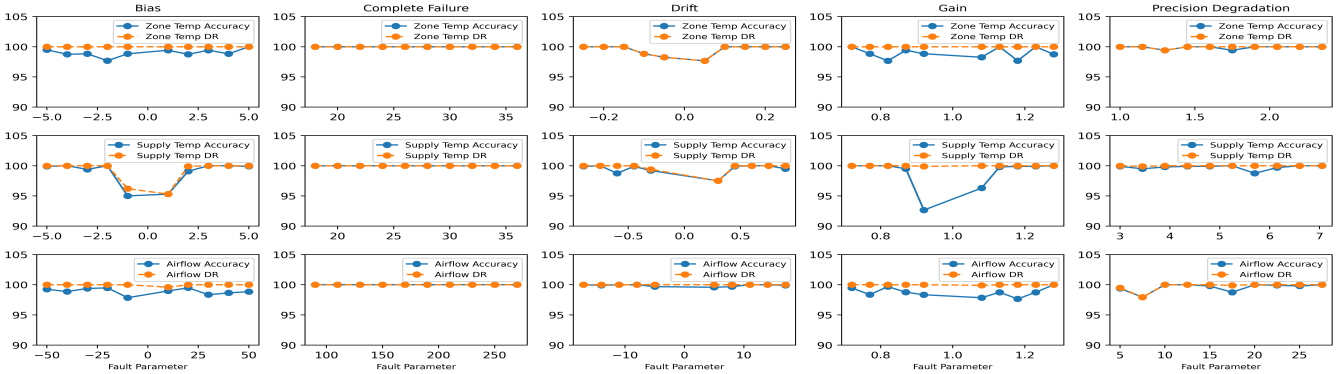

Figure 12: Sensitivity Analysis: accuracy for different fault parameter values for zone/supply temperature, and airflow sensors

6.2 Sensitivity Analysis

We explore how our models perform on real data for various faults with different fault parameter values, mentioned in § 4.4. Figure 12 shows the accuracy and detection rate of the models for different fault types and parameters. As can be seen, bias, complete failure, and gain faults are mostly correctly predicted as faulty and the detection rate is 100%. However, sometimes, bias faults with the parameters of -1 and 1 can be missed, and detection accuracy is around 95%. Considering the dynamic nature of supply temperature, $\pm 1^{\circ}C$ may be acceptable in real life. However, when the magnitude of the bias fault is greater or equal to 2, we can detect all fault cases with an accuracy of 100%. Similarly, small drift and precision degradation faults are classified as normal and the detection rate can be about 97%. Also, we can see that bias and gain faults are confused with each other, even though both of them are detected as faulty. Misclassifications can happen when there is low variability in the data and bias and gain faults output the same values at the same set of fault parameter values. Overall, the classification accuracy is approximately 97.5% for zone temperature and airflow sensors. We can observe non-negligible misclassifications for complete failure. Drifts are one of the complex fault types, and it is hard to detect faults when the drift rate is very small $\pm 0.05^{\circ}C$ and $\pm 0.1^{\circ}C$ for zone and supply temperatures. Overall, we can detect and classify faults with an accuracy of above 90% for each sensor in all cases.

6.3 Baseline comparison

This study compares our LGBM-based FADED system with three state-of-the-art machine learning FDDs mentioned in related work. These are XGB, kNN, and SVM-based algorithms used to detect

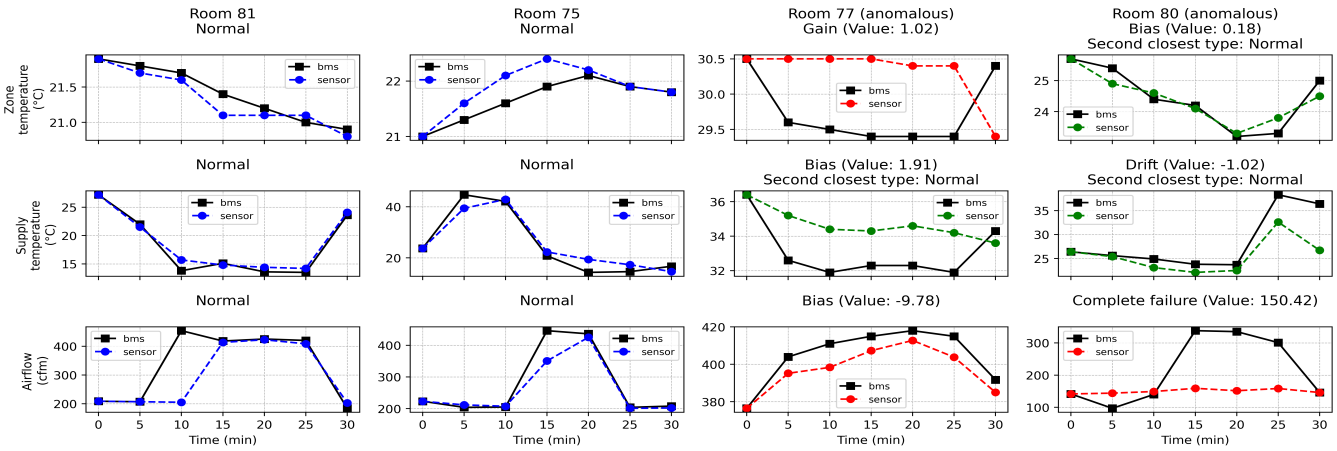

Figure 13: Real Experiments in a University Building

Table 7: Performance Comparison with other Schemes

Model	Accuracy (%)	DR (%)	FaR (%)
Zhang et al. (XGB) [43]	96.67	98.26	1.28
Gao et al. (kNN) [14]	86.60	95.62	2.10
Montazeri et al. (SVM) [30]	98.14	99.32	0.71
FADED (LGBM)	**99.08**	**99.87**	**0.06**

and diagnose sensor faults in different HVAC units. The hyperparameters of these benchmark algorithms are estimated using Bayesian Search as well. Table 7 shows accuracy, DR, and FaR for each algorithm. As it can be seen, FADED has a higher accuracy and detection rate, lower false alarm rate, and shows faster convergence compared to other algorithms for multiclass classification tasks while using real data traces from the BMS.

6.4 FADED Operational Experience

To close the loop, we used FADED to detect, diagnose, and reconstruct faults in 4 different zones in our building, hoping to detect, diagnose, and reconstruct potential sensor faults. Remarkably, the model that was trained using data from a single zone showed a significant degree of generalizability. This suggests that new zones could successfully use the same pre-trained models without further retraining or fine-tuning. These results highlight the generalizability of our methodology, facilitating its practical application across diverse zones within the building infrastructure.

Fig. 13 shows the results. Blue lines indicate normal, red failures, and green shows fault detected with the second option being normal. Room 81 and room 75 are assumed to be in normal operation as there were no registered complaints about those rooms. All the sensors in rooms 81 and 75 are classified as normal according to FADED. Room 77 and Room 80 were determined to be faulty by the facility manager. In particular, Room 77 was very hot despite the facility manager's and HVAC technician's efforts to control it. During experimentation, we also had problems with gathering quantitative data for our test. The HVAC system was not responding to our step functions when we forcefully made room to cool down or heat up to a particular temperature. Other rooms reacted to that command either by changing the supply temperature and/or supply airflow. But in Room 77 these sensor values experienced slight fluctuations only. We detected some bias faults for supply temperature and airflow and a small gain fault for zone temperature. However, from our observation, the problem might be more complex and involve actuator faults (we reviewed the control logic and sequence of operations, and they were all correct). On the other hand, Room 80 was having some ventilation problems. FADED detected a complete failure fault in the airflow sensor with the parameter of 150.42 *cfm*. Zone temperature is predicted as having a bias of 0.18 $^{\circ}C$, whereas supply temperature is predicted as having drift with the magnitude of -1.02 $^{\circ}C$. However, for both zone and supply temperatures, the second closest predicted class is normal. So given the predicted type, the fault parameter value estimated, and knowing the second closest predicted type might be useful in real-world scenarios and help HVAC technicians to make informed decisions.

6.5 Discussion

Model selection was done by trial and error. We tried different modeling techniques and we picked LGBM for classification and XGBRegressor for fault reconstruction. While other techniques like deep neural networks may provide slightly better results, they are more difficult to train and they are less interpretable when compared to FADED [20]. The margin for improvement is very small since our classification error is very close to nil, so we are quite satisfied. Feature engineering is a critical aspect of improving performance in any data-driven scheme. In our case, the main decision was to curtail the number of lagged features used in the input vector. While a larger number of features can help improve accuracy, the time it takes to collect more data while FADED is deployed is costly in terms of employee time. We settled for 7 samples (30 minutes), which could be revised depending on the accuracy/time tradeoff. This is one of FADED's most salient features. Facilities crews were quite impressed that we were able to find sensor errors in only a half hour. Larger scale usability studies have been left for future work. A piece of FADED that we may have overengineered is the computer vision algorithms used to do vent area estimation. While the number of supply vents and diffusers in the market is large, it is finite. Perhaps a simpler solution would have been to classify the type of vent using a small database of vents since the cross-sectional flow area for each vent is fully documented by

the manufacturers. We have left this for future work. FADED fails when the reference sensors themselves become unreliable or faulty. Therefore, technicians should ensure these redundant sensors are not faulty before using them, but this should be doable as they are mobile sensors and not embedded in the building infrastructure.

7 CONCLUSION

HVAC sensor faults hurt user comfort and energy efficiency, thus, it is essential to identify and quantify them. The proposed approach uses hardware redundancy, and machine learning models to facilitate labor-intensive, time-consuming, and expertise-dependent fault detection and diagnosis process. FADED has less than 0.1% FaR for zone temperature and airflow sensors. Also, it can classify real zone temperature, supply temperature, and airflow faults with an accuracy of 99.57%, 98.67%, and 99.01%. Moreover, FADED can contribute to the development of "self-healing" buildings, where the parameters in the BMS are automatically adjusted to compensate for sensor faults to ensure optimal operation of the HVAC system. RMSE for predicted fault parameter values is acceptable considering the sensitivity of the sensors. The proposed method was tested on a real building and it detected the existing temperature and airflow sensor faults in the VAV unit of malfunctioning rooms.

8 ACKNOWLEDGEMENTS

We would like to thank anonymous reviewers for their constructive comments and helpful suggestions. This material is based upon work partially supported by a Seed Fund award from the Center of Information Technology in the Interest of Society (CITRIS). .

REFERENCES

[1] [n. d.]. Govee H5072 Bluetooth Thermometer Hygrometer User Manual. https://manuals.plus/govee/govee-h5072-bluetooth-thermometer-hygrometer?expand_article=1. Accessed: 2023-09-20.
[2] [n. d.]. testo 410i - Vane anemometer wireless Smart Probe. https://www.testo.com/en-US/testo-410i/p/0560-1410. Accessed: 2023-09-20.
[3] ANSI/ASHRAE/ACCA. 2018. Standard Practice for Inspection and Maintenance of Commercial Building HVAC Systems.
[4] Antoniadou and et al. 2017. Occupants' thermal comfort: State of the art and the prospects of personalized assessment in office buildings. *Energy and Buildings* 153 (2017), 136–149.
[5] ASHRAE. 2020. ASHRAE Handbook: HVAC Systems and Equipment.
[6] Basarkar. 2011. Modeling and simulation of HVAC faults in EnergyPlus. (2011).
[7] Chen and et al. 2010. Fault detection, diagnosis and data recovery for a real building heating/cooling billing system. *Energy Conversion and Management* 51, 5 (2010), 1015–1024.
[8] Chen and et al. 2022. What's in a Name? Developing a Standardized Taxonomy for HVAC System Faults. (2022).
[9] Comstock and et al. 2002. Fault detection and diagnostic (FDD) requirements and evaluation tools for chillers. *West Lafayette, IN: ASHRAE* (2002).
[10] Crabtree and et al. 2012. Preparing building technicians for energy efficient operations of high performance facilities. *ACEEE* (2012).
[11] Buildings Energy Data. 2011. Energy efficiency and renewable energy. *US department of energy* 292 (2011).
[12] Fisk. 2009. Quantitative relationship of sick building syndrome symptoms with ventilation rates. (2009).
[13] Gao and et al. 2019. A novel chiller sensors fault diagnosis method based on virtual sensors. *Sensors* 19, 13 (2019), 3013.
[14] Gao and et al. 2022. Sensor drift fault diagnosis for chiller system using deep recurrent canonical correlation analysis and k-nearest neighbor classifier. *ISA transactions* 122 (2022), 232–246.
[15] Hong and et al. 2022. A novel CNN-TCN-TAM classification model based method for fault diagnosis of chiller sensors. In *IEEE CCC*.
[16] Hosseini and et al. 2021. Knowledge Discovery by Analyzing the State of the Art of Data-Driven Fault Detection and Diagnostics of Building HVAC. *CivilEng* 2, 4 (2021), 986–1008.
[17] Hou and et al. 2021. Associations of indoor carbon dioxide concentrations, air temperature, and humidity with perceived air quality and sick building syndrome symptoms in Chinese homes. *Indoor Air* 31, 4 (2021), 1018–1028.
[18] Hu and et al. 2016. Sensitivity analysis for PCA-based chiller sensor fault detection. *International journal of refrigeration* 63 (2016), 133–143.
[19] Jana and et al. 2022. CNN and Convolutional Autoencoder (CAE) based real-time sensor fault detection, localization, and correction. *Mechanical Systems and Signal Processing* 169 (2022), 108723.
[20] Ke and et al. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *NIPS*.
[21] Kim and et al. 2018. A review of fault detection and diagnostics methods for building systems. *Science and Technology for the Built Environment* 24, 1 (2018).
[22] Kullaa. 2013. Detection, identification, and quantification of sensor fault in a sensor network. *Mechanical Systems and Signal Processing* 40, 1 (2013), 208–221.
[23] Li and et al. 2016. A sensor fault detection and diagnosis strategy for screw chiller system using support vector data description-based D-statistic and DV-contribution plots. *Energy and Buildings* 133 (2016), 230–245.
[24] Li and et al. 2019. Handling incomplete sensor measurements in fault detection and diagnosis for building HVAC systems. *IEEE Transactions on Automation Science and Engineering* 17, 2 (2019), 833–846.
[25] Liu and et al. 2019. Sensor fault detection and diagnosis method for AHU using 1-D CNN and clustering analysis. *Comput. intelligence and neuroscience* (2019).
[26] Luo and et al. 2019. Development of clustering-based sensor fault detection and diagnosis strategy for chilled water system. *Energy and Buildings* 186 (2019).
[27] Malluhi and et al. 2022. Enhanced Multiscale Principal Component Analysis for Improved Sensor Fault Detection and Isolation. *Sensors* 22, 15 (2022), 5564.
[28] Mattera and et al. 2018. A method for fault detection and diagnostics in ventilation units using virtual sensors. *Sensors* 18, 11 (2018), 3931.
[29] Mohamed and et al. 2016. SBDaaS: Smart building diagnostics as a service on the cloud. In *IEEE IGBSG*.
[30] Montazeri and et al. 2020. Fault detection and diagnosis in air handling using data-driven methods. *Journal of Building Engineering* 31 (2020), 101388.
[31] Morris and et al. 2012. *Measurement and instrumentation: theory and application*. Academic Press.
[32] Ng and et al. 2020. Bayesian method for HVAC plant sensor fault detection and diagnosis. *Energy and Buildings* 228 (2020), 110476.
[33] Norbäck and et al. 2008. Sick building syndrome in relation to air exchange rate, CO 2, room temperature and relative air humidity in university computer classrooms: an experimental study. *International archives of occupational and environmental health* 82 (2008), 21–30.
[34] Norford and et al. 2000. Final report of ASHRAE Research Project 1020-RP: Demonstration of fault detection and diagnosis in real a building. *Massachusetts Institute of Technology and Loughborough University* (2000).
[35] Raju and et al. 2020. Study the influence of normalization/transformation process on the accuracy of supervised classification. In *IEEE ICSSIT*.
[36] Reddy and et al. 2001. Development and comparison of on-line model training techniques for model-based FDD methods applied to vapor compression chillers: Evaluation of mathematical models and training techniques. *Final Report, ASHRAE Research Project* 1139 (2001).
[37] Roth and et al. 2005. Energy impact of commercial building controls and performance diagnostics: market characterization, energy impact of building faults and energy savings potential. *Prepared by TAIX LLC for the US Department of Energy. November. 412pp (Table 2–1)* (2005).
[38] Sturzenegger and et al. 2012. Semi-automated modular modeling of buildings for model predictive control. In *ACM BuildSys*.
[39] Syazwan and et al. 2023. Identification of competency supervisor and technician heating ventilating and air conditioning (HVAC) maintenance in oil and gas industry Malaysia. In *AIP*.
[40] Wen and et al. 2012. RP-1312–Tools for evaluating fault detection and diagnostic methods for air-handling units. *ASHRAE, Tech. Rep, Tech. Rep* (2012).
[41] Winkler and et al. 2020. Office: Optimization framework for improved comfort & efficiency. In *ACM/IEEE IPSN*.
[42] Yoon and et al. 2011. Residential heat pump heating performance with single faults imposed. *Applied thermal engineering* 31, 5 (2011), 765–771.
[43] Zhang and et al. 2021. Fault detection and diagnosis for the screw chillers using multi-region XGBoost model. *Science and Technology for the Built Environment* 27, 5 (2021), 608–623.
[44] Zhang and et al. 2022. Evaluate the impact of sensor accuracy on model performance in data-driven building fault detection and diagnostics using Monte Carlo simulation. In *Springer Building Simulation*.