BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 05/10 pp170–184
Volume 6, Number 2, June 2023
DOI: 10.26599/BDMA.2022.9020015

Semi-Supervised Machine Learning for Fault Detection and Diagnosis of a Rooftop Unit

Mohammed G. Albayati†, Jalal Faraj†, Amy Thompson, Prathamesh Patil, Ravi Gorthala, and Sanguthevar Rajasekaran*

Abstract: Most heating, ventilation, and air-conditioning (HVAC) systems operate with one or more faults that result in increased energy consumption and that could lead to system failure over time. Today, most building owners are performing reactive maintenance only and may be less concerned or less able to assess the health of the system until catastrophic failure occurs. This is mainly because the building owners do not previously have good tools to detect and diagnose these faults, determine their impact, and act on findings. Commercially available fault detection and diagnostics (FDD) tools have been developed to address this issue and have the potential to reduce equipment downtime, energy costs, maintenance costs, and improve occupant comfort and system reliability. However, many of these tools require an in-depth knowledge of system behavior and thermodynamic principles to interpret the results. In this paper, supervised and semi-supervised machine learning (ML) approaches are applied to datasets collected from an operating system in the field to develop new FDD methods and to help building owners see the value proposition of performing proactive maintenance. The study data was collected from one packaged rooftop unit (RTU) HVAC system running under normal operating conditions at an industrial facility in Connecticut. This paper compares three different approaches for fault classification for a real-time operating RTU using semi-supervised learning, achieving accuracies as high as 95.7% using few-shot learning.

Key words: semi-supervised machine learning; fault classification; fault detection and diagnostics; heating, ventilation, and air-conditioning; data-driven modeling; energy efficiency

• Mohammed G. Albayati and Amy Thompson are with the Department of Mechanical Engineering, School of Engineering, University of Connecticut, Storrs, CT 06269, USA. E-mail: mohammed.albayati@uconn.edu; amy.2.thompson@uconn.edu.
• Jalal Faraj and Sanguthevar Rajasekaran are with the Department of Computer Science and Engineering, University of Connecticut, Storrs, CT 06269, USA. E-mail: jalal.faraj@uconn.edu; sanguthevar.rajasekaran@uconn.edu.
• Prathamesh Patil and Ravi Gorthala are with the Department of Mechanical and Industrial Engineering, Tagliatela College of Engineering, University of New Haven, West Haven, CT 06516, USA. E-mail: ppati1@newhaven.edu; rgorthala@newhaven.edu.
† Mohammed G. Albayati and Jalal Faraj contributed equally to this article.
* To whom correspondence should be addressed.
Manuscript received: 2022-06-16; accepted: 2022-06-30

1 Introduction

Heating, ventilation, and air-conditioning (HVAC) systems account for 30% of the energy consumption in U.S. commercial buildings annually. Rooftop units (RTUs), which serve various-sized commercial buildings, are one of the major contributors to energy waste. According to the U.S. Department of Energy, RTUs serve about 60% of U.S. commercial buildings. Inefficient unit operation is common due to faults introduced during the installation of the units or that can occur during operation. These faults can result in $900 to $3700 worth of energy waste per unit annually[1]. Faults in RTUs are categorized into two main types: (a) hard faults, also called hard failures, which cause the RTU to stop functioning and (b) soft faults, which can decrease

the performance of the RTU until a hard failure occurs. One advantage of a fault detection and diagnosis (FDD) system is that it can detect and diagnose soft faults before hard failures occur[2]. A common approach to detecting faults in HVAC systems is to collect real-time operating characteristic data using a sensor network and analyze the collected data to determine if faults are present and the type of fault. By using FDD approaches, building owners can conduct early hardware repair or re-program control software to prevent eventual hard faults or to lower energy cost due to suboptimal operation.

FDD tools and methods have been developed extensively using laboratory data[2–6]. As an example, in their laboratory study, Braun and Yuill[3] developed a methodology to assess the FDD protocols for air conditioning devices. They fed the FDD protocol with different sets of experimental data under differing fault conditions and observed the responses. The methodology was found to perform poorly, resulting in identifying up to 51% of faults where no faults were present, 26% misdiagnosing faults, and 32% not detecting faults where the faults were present. However, only a few recent FDD for HVAC studies have been performed using datasets collected from the field, capturing typical HVAC operating conditions during typical building use. The laboratory set-up can get close to mimicking an actual system, but it does not fully depict the unit behavior under typical operating conditions due to multiple types of uncertainties.

In their field study, Wall and Guo[6] studied six different market-ready FDD tools (software-based from a building energy management system (BEMS)) at six different buildings located in Australia for different commercial building types including offices, airports, museums, hospitals, and laboratories to demonstrate benefits, capabilities, and value of the FDD tools. The results for each building were reported in terms of energy reduction, improved occupant comfort, maintenance issues, and site energy intensity. A combination of rule-based fault detection and first principles of thermodynamics was developed in Ref. [4]. It identifies the selected faults and estimated energy savings for an air handling unit (AHU) using the real operating data collected under normal operating conditions from a BEMS during a field study. The savings were estimated for identified faults by comparing the energy usage before and after the faults were repaired. Granderson et al.[5, 7] developed an automated FDD characteristic framework to better understand current automated FDD technologies and tools by defining their main characteristics.

Kim and Katipamula[8] surveyed FDD methods since 2004 and classified them into three categories: process history based, qualitative model based, and quantitative model based, and assessed the strengths and weaknesses of common FDD methods. Machine learning (ML) models were built by Robinson et al.[9] to predict energy consumption in commercial buildings. Researchers trained the models on national data from the Commercial Buildings Energy Consumption Survey and validated them with the New York City Local Law 84 energy consumption dataset. Their methodology depends on five commonly available building and climate features, and their best performing ML model was found to be gradient boosting regression.

Supervised ML methods for FDD for HVAC systems are efficient if the collected dataset is labeled and is balanced; however, data collected during field studies is usually not labeled and imbalanced. Researchers have developed unsupervised ML approaches to handle imbalanced data. An unsupervised ML framework was developed by Yan et al.[10], based on the generative adversarial network (GAN), to generate new faulty data using a few faulty data points from the original dataset for an AHU. The promise was to re-balance the data using a few faulty data points and then use the supervised approaches to detect and diagnose AHU faults. Another ML method was developed by Yan et al.[11] to better deal with the imbalanced training data problem for FDD for AHUs where a few faulty samples exist with many normal samples. This method illustrates the use of semi-supervised ML, specifically semi-supervised support vector machines (SVMs), to handle imbalanced training datasets as well as the required faulty samples to accurately predict AHU faults. The accuracy of their method was 80–89% with a training dataset of 8000 normal samples and 30 faulty samples for each fault. New samples were artificially generated to balance the minority classes or faults using the Synthetic Minority Over-sampling Technique (SMOTE)[2, 12]. Imbalanced data can significantly influence the fault classification accuracy. Yan et al.[13] implemented the conditional Wasserstein GAN algorithm to re-balance the training dataset for a chiller automated FDD system so that supervised ML methods can then be applied. GAN can be applied to randomly increase the diversity of training data. It has been shown that using GAN to generate faulty samples is an efficient approach to enrich training

dataset; however, selecting high quality synthetic fault samples is a significant factor on the accuracy of the automated FDD methods. Yan[14] investigated the use of the variational auto-encoder and the Anomaly with GAN to control the data generation and selecting the high-quality synthetic samples for chiller HVAC systems. Their method was found to outperform traditional FDD methods.

There are many attempts to promote ML for detecting and diagnosing some common RTU faults; however, these studies are limited by data availability and ML methods. Multi-class classification ML methods were developed by Ebrahimifakhar et al.[15] for detecting and diagnosing seven common RTU faults, using simulation data with fifteen input variables to train and test three classification methods: k-nearest neighbor (k-NN), logistic regression, and random forests. Out of the three classification methods, the logistic regression method performed the best with 93.6% accuracy. A data-driven FDD for RTU method was developed by Ebrahimifakhar et al.[2] using simulated data with fifteen input variables to detect seven common RTU faults utilizing seven statistical ML classification methods. The overall accuracy showed SVM as the best classifier with an accuracy of 96.2% and linear discriminant as the worst classifier with an accuracy of 76.2%. SMOTE was used to balance the minority classes, resulting in a better performance of classifying the minority class. The multiscale convolutional neural networks FDD for AHU approach was proposed by Cheng et al.[16], which improves the ability of feature extraction using three different scale kernels, thus improving the diagnostic performance of the proposed method. Their proposed method outperformed other data driven FDD approaches but did not address the issue of imbalanced data. A deep learning fault diagnostic method was proposed by Lee et al.[17] to improve the operational efficiencies of AHU and although the model achieved 95.16% accuracy using the simulated data under ideal conditions, it was not validated with real AHU data gathered under normal operation conditions. Another supervised ML FDD method was developed by Wang et al.[18] using a two-layer random forest based FDD to isolate the simultaneous faults in variable air volume systems and the developed method was validated using real operation data, with the lowest accuracy found to be 80.1%. Chintala et al.[19] developed an FDD algorithm for residential air-conditioning systems using a Kalman filter model using already existing data points for indoor and outdoor air temperatures. The algorithm was tested on EnergyPlus simulated data and was found to perform at low fault classification accuracies: 40% for airflow and undercharge faults, and 70% for duct leak faults.

The goal of fault detection for HVAC systems is to detect faults in real time, i.e., within minutes of a fault occurring, so that diagnosis and fault repair can be conducted to reduce unnecessary energy consumption and ensure comfortable indoor environments. ML algorithms have been applied to detect faults in complex operating systems because these algorithms can run fast and repetitively on one-minute interval data, with a short training period. Supervised ML approaches developed for fault detection in laboratory conditions[2, 15] may not work well for HVAC systems running under normal operating conditions. This is because the supervised ML approaches need every data point to be labeled, which is typically not possible under normal operating environments where fault conditions are not controlled. Unsupervised ML approaches can be built without labeled data points (only row data); however, there is no way to verify and validate the accuracy of the ML model and whether faults are identified accurately using this approach. A semi-supervised data-driven ML approach can be applied to FDD for detecting some common faults in packaged RTU systems based upon datasets collected in the field under normal operating conditions. Semi-supervised ML works by combining an unsupervised ML method with a supervised ML method[20]. It is highly effective in building a more robust model that uses a small number of labeled training data points collected during short training period to predict outcomes based upon many unlabeled test data points (post training period during normal operation). However, it comes with higher computational cost in comparison with supervised learning[21]. Because labeled data is expensive and labor intensive, many researchers have investigated the use of semi-supervised FDD approaches; however, the accuracy of these models can be hard to predict when the test data is unlabeled. Semi supervised neural network FDD was developed in Ref. [22] for AHU and tested using operational data. The authors stated that their approach can not only diagnose faults with limited labeled data, but also detect unseen faults. Li et al.[23] developed a semi-GAN fault diagnosis for a chiller that extracts information from unlabeled data with limited labeled training data. They trained and tested their model with experimental data and their models illustrated a potential for fault diagnosis of 84% accuracy with 84

labeled data points. Their method is limited because it cannot diagnosis simultaneously occurring faults. The authors stated improvement in accuracy with increasing unlabeled data size. The proposed semi-supervised data driven ML approach can be used to develop a realistic and robust FDD algorithm since it can be applied during normal operating conditions. The previously developed ML approaches can handle one fault at a time, but this is not applicable for an RTU operating in the field where faults can occur simultaneously, so this aspect adds complexity to the fault detection and classification problem.

This research proposes a novel approach to apply semi-supervised ML to datasets collected from an operating system in the field to develop better FDD methods. The study data was collected from an RTU running under normal operating conditions[24]. Two different approaches are investigated for fault classification using semi-supervised learning with the goal of minimizing labeled training data points. Accuracy can be measured by calculating the percentage of correct predictions made by the model on the test dataset. Since the test data points are labeled, the prediction accuracy is determined by comparing the labels of the predicted test data points with the actual labels of the corresponding test data points. A tradeoff is proposed to determine the best method that achieves the higher fault prediction accuracy for each fault type.

2 RTU Specifications and Considered Faults

The datasets are collected from a two-stage Trane RTU serving 207 000 square feet (19 230.93 m^2) of enclosed and conditioned industrial space, located in Connecticut. Table 1 lists the RTU specification.

Even though there is a lengthy list of possible RTU faults that could be evaluated, the focus of the paper is on four faults while demonstrating the developed ML approaches. The reason for selecting only these four faults is based upon confidence in the fault intensity calculation, detailed below, to accurately identify these four faults. In the future, more faults could be added as more labeled data points for the other faults become available. These four faults are as follows.

- Refrigerant undercharge (UC). This is one of the most common faults in an RTU and it indicates either there is a leak in the refrigerant line, or that the unit has not been charged according to the manufacturer's specifications. Fault intensity of refrigerant charge (FI_{ch}) was calculated according to the method presented by Albayati et al.[24], which was originally developed using a virtual refrigerant charge sensor approach[25–27]. This approach is easier to implement for field study and requires data acquired from surface-mounted temperature sensors. The calculated FI_{ch} is then used to label the refrigerant undercharge fault from the dataset. In this case, FI_{ch} of -0.2, which indicates 20% refrigerant undercharge, was considered the threshold for the undercharge fault. This threshold value was chosen according to the RTU specifications and previous studies on the impact of the FI_{ch} on the coefficient of performance (COP) of air conditioning units. For example, for a fixed orifice (FXO) unit operating at 20% undercharge and an Air-Conditioning, Heating, and Refrigeration Institute (AHRI) A rating condition, the COP is 84.7% of the nominal COP value[24,28,29].
- Refrigerant overcharge (OC). This fault is less common than the UC fault and has less effect on the RTU performance. This type of fault is usually caused by a technician's inexperience or error when charging the system. This is mostly because either the RTU is an old unit and no manufacturer manual exists, so the technician will usually estimate the refrigerant amount. Another instance that can lead to this fault is if there is a small leak which is hard to detect, this leads the technician to overcharge the RTU for longer operation time. Just like UC, OC is negatively affecting the compressor performance and consequently causes an

Table 1 Rooftop unit specification.

Specification	Value
Size	35 kW
Refrigerant	R22
Refrigerant charge	3.3 kg(Circuit 1); 2.4 kg(Circuit 2)
Number of compressors	2
Compressor type	Scroll
Expansion device	Fixed orifice
Nominal (max) airflow	113.3 m^3/min
Gross cooling capacity	3.3695 kW
ARI net cooling capacity	3.3402 kW
Number of fans	2 (1 outdoor and 1 indoor)
Fan power	0.56 kW; 2.2 kW
Unit power	10.96 kW
Energy efficiency ratio	10.4
Seasonal energy efficiency ratio	11.75
Coefficient of performance	3.05
Voltage	208/230 V(3 phase, 60 Hz)

RTU to not perform at the nominal efficiency level. FI_{ch} was calculated according to the previously developed methods[24–27] and was used to label OC faulty data points. The threshold of FI_{ch} was chosen to be $+0.2$ based on the RTU specification and previously published works on the impact of the FI_{ch} on the COP of air conditioning units. For an FXO unit operating at 20% overcharge under an AHRI A rating condition, the COP is 94.9% of the nominal COP value[24,28,29].

• Condenser fouling (CA). Any reduction in the airflow due to the condenser fouling leads to a reduction in the heat rejection from the condenser coils to the surroundings. This is mainly because fouling works as an insulator to prevent heat transfer and this indicates the need for cleaning. Fault intensity of condenser fouling (FI_{CA}) was calculated based on the method presented by Albayati et al.[24] and Yuill et al.[30]. The method utilizes the actual airflow rate (AFR) and the nominal AFR through the condenser to calculate the presence of the fault and its intensity. The generalization relationship developed by Mehrabi et al.[28] was found to be useful to determine the threshold value of FI_{CA}, assuming the AHRI A rating test condition for an FXO air condition unit. The FI_{CA} threshold value was chosen to be –0.4 which impacts the COP[28], reducing the air conditioner unit performance to 81.5% of the nominal COP value.

• Evaporator fouling (EA). This fault is caused by a reduction in airflow across the evaporator which reduces the efficiency of the RTU. The fault intensity of evaporator fouling (FI_{EA}) was calculated[24,30]. The calculation method utilizes the actual and nominal supply AFR through the evaporator to determine the presence of the fault and to calculate its severity. The FI_{EA} is used to identify the EA fault by comparing it with a preset threshold value. The system was considered faulty if the FI_{EA} value was less than the threshold. The threshold value was selected to be -0.4 based on the fault impact on the COP of the RTU, assuming the AHRI A rating test condition for an FXO unit. The air conditioning unit performs 96.9% of the nominal COP with a $-0.4\,FI_{EA}$ value[28].

• No fault (NF). No fault represents the data points where none of the faults listed above are present. This was an important category to include since the data is collected from an RTU running under normal operating conditions, with no control over what fault to include or exclude. There are some instances in our dataset where no fault, only one fault, or more than one fault was present at the same time.

3 Preliminary Study

This section describes the dataset used to build the ML models and feature selection process.

3.1 Lab-in-the-field vs. traditional lab setting

The data for this study was collected from a "lab-in-the-field" where the researchers installed 20+ sensors on a packaged HVAC unit serving an actual operating warehouse in Connecticut. Because the HVAC unit was in use by the warehouse, the research team could not inject known faults into the HVAC unit. The data collected represents states and behavior of the HVAC unit that may or may not contain faults, which occurred over a specific time span of hours during the summer peak operating period in Connecticut. The research team identified whether certain faults occurred by calculating fault intensity (FI) values based upon data collected by the sensors and based upon usage of FI threshold values proven reasonable by previous studies. The collection of data under this scenario is referred to in the paper as "normal operating conditions".

3.2 Data description and pre-processing

The detailed methodology used to collect and store the study data is described by Albayati et al.[24] The sensors and data logger recorded the data continuously with one-minute intervals for each input variable (feature). There are two datasets used for this study. The first dataset includes a total of 4284 observations (three days' worth of data); however, only 3336 observations, representing faulty and unfaulty data with a total of 30 input variables (features), were used after excluding the observations where the RTU was off. The second dataset has the same input variables (features) with 2873 observations (two days' worth of data), but only 2099 observations are considered after omitting the system-off observations. The second dataset differs from the first dataset as it is assumed that only one fault occurs per observation in the second dataset. Both datasets shown in Table 2 are for the same RTU, collected using the same sensors, and have the same input variables, or features, listed in Table 3 and detailed in the dataset[31].

Figure 1a shows where the air-side temperature and relative humidity sensors are located in the return duct, supply duct, mixed air section, outdoor section, and the

Table 2 Characteristics of Datasets 1 and 2.

Dataset name	# observations	Days of data	Multiple faults?
Dataset 1	3336	3	Yes
Dataset 2	2099	2	No

Table 3 List of measured input variables (features) for the RTU that used to develop the proposed FDD methods.

Feature	Unit	Meaning	Feature	Unit	Meaning
$\text{Power}_{\text{main}}$	W	Main power	$T_{\text{air, cond}}$	℃	Condenser exiting air temperature
$\text{Power}_{\text{fan}}$	W	Fan power	$T_{\text{air, evap}}$	℃	Evaporator exiting air temperature
$\text{Power}_{\text{comp1}}$	W	Circuit 1 compressor power	$T_{\text{air, econ}}$	℃	Economizer air temperature
$\text{Power}_{\text{comp2}}$	W	Circuit 2 compressor power	$\text{RH}_{\text{air, econ}}$	%	Economizer air relative humidity
P_{suc1}	kPa	Circuit 1 suction pressure	$T_{\text{air, ret}}$	℃	Return air temperature
P_{suc2}	kPa	Circuit 2 suction pressure	$T_{\text{air, ret, avg}}$	℃	Return air average temperature
T_{suc1}	℃	Circuit 1 suction temperature	$\text{RH}_{\text{air, ret}}$	%	Return air relative humidity
T_{suc2}	℃	Circuit 2 suction temperature	$T_{\text{air, sup}}$	℃	Supply air temperature
P_{dis1}	kPa	Circuit 1 discharge pressure	$\text{RH}_{\text{air, sup}}$	%	Supply air relative humidity
P_{dis2}	kPa	Circuit 2 discharge pressure	$T_{\text{air, sup, avg}}$	℃	Supply air average temperature
T_{dis1}	℃	Circuit 1 discharge temperature	$\text{RH}_{\text{air, sup, avg}}$	%	Supply air average humidity
T_{dis2}	℃	Circuit 2 discharge pressure temperature	$T_{\text{air, ret, avg}}$	℃	Supply air average temperature
T_{EL1}	℃	Circuit 1 temperature after expansion device	$\text{RH}_{\text{air,i}}$	%	Indoor air relative humidity
T_{EL2}	℃	Circuit 2 temperature after expansion device	$T_{\text{air,o}}$	℃	Outdoor air temperature
AFR	m^3/min	Airflow rate	$\text{RH}_{\text{air,o}}$	%	Outdoor air relative humidity

indoor thermal zone. The high accuracy RTD sensors denoted 2 and 4 are respectively located within the return and supply air sections. The temperature and relative humidity (T/RH) duct probes denoted 1, 3, and 5 are in the return, supply, and mixed air sections. The indoor and outdoor sensors denoted 7 and 6 are respectively located within the thermal zone that the RTU serves, and near the economizer located outdoors. Figure 1b depicts the location of pressure and temperature instruments. Pressure transducers identified by 9 and 10 locate the discharge and suction pressure sensors, while sensors 11, 12, and 13 denote the surface-mounted thermocouple locations on the discharge, suction, and liquid line respectively. Sensors located at 14 and 15 indicate thermocouples located on the air stream of the evaporator and condenser. In addition to refrigerant side and air side instruments, power measuring instruments are utilized to measure main, compressor, and the evaporator fan power. The configuration below depicts a single cooling stage, so for the two cooling stages the same instrumentation method is repeated on the second cooling stages.

Unlike laboratory experiments where the researcher controls what data to include, this was not possible since the project team was collecting the data from an RTU located in the field running under normal operating conditions. Therefore, P_{fan}, P_{comp1}, P_{comp2} were used

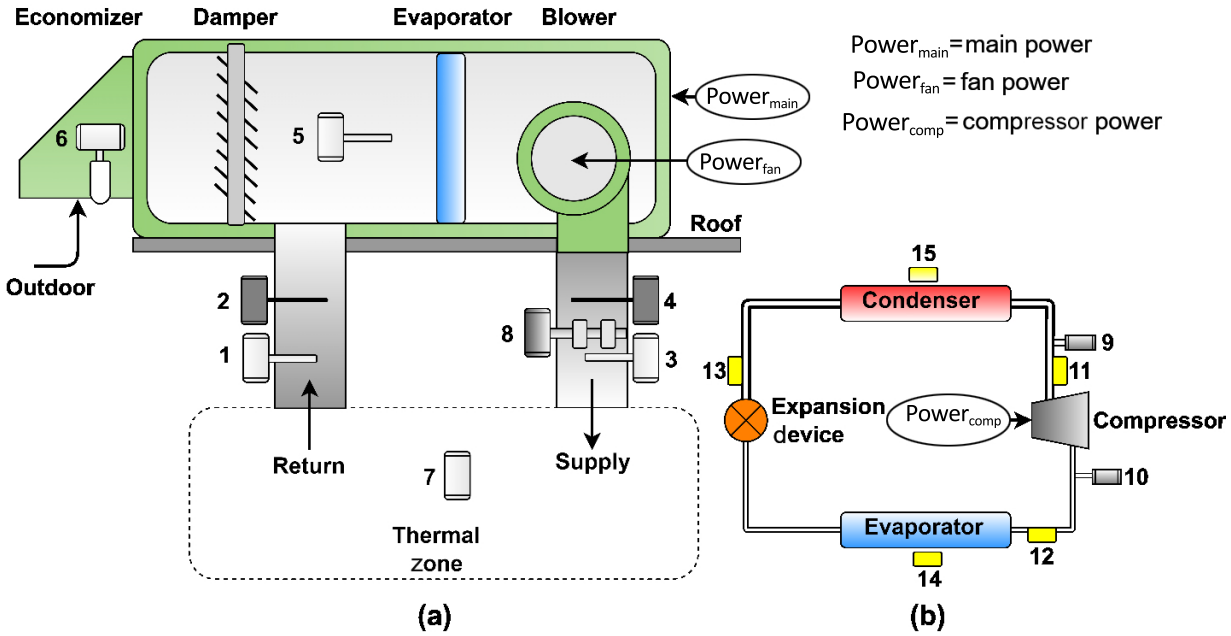


Fig. 1 (a) A schematic of an RTU with labels indicating air-side sensor locations; (b) A schematic of the vapor compression cycle with labels indicating refrigerant side pressure and temperature instrument locations.

to exclude the observations where the RTU was off. This was a crucial step to eliminate the probability of misdiagnosing the unit-off data points. Then, both datasets were labeled based on pre-identified threshold values of FI for each fault as illustrated in Section 2 and summarized in Table 4. The threshold FI values were chosen based on their effects on the unit performance as illustrated in Refs. [24, 28, 29].

Interestingly, the majority of the data points in the labeled datasets are faulty with at least one fault presented per observation. This was helpful for training the model since the abundance of faults made it easier to train the model to classify them. Additionally, the dataset has many instances where there are multiple faults per observation. This is an advantage for our model since it performs multi-fault detection, which is desired when study datasets are collected from field study.

Imbalanced datasets can pose a major challenge to the performance of a ML model[2, 10, 11, 15], however, using our robust semi-supervised ML methods, the imbalance was addressed while minimizing the model error. Table 5 depicts the value counts of the fault labels in both datasets, while Table 6 shows percentages of observations of Dataset 1 for individual faults as well as the combination of faults. The sum of the faulty and non-faulty observations exceeds 3336 observations for Dataset 1 because more than one fault was present per observation.

Table 4 List of faults with FI threshold.

Fault type	FI threshold
UC for circuit 1 and 2 (UC_{C1} and UC_{C2})	−0.2
OC for circuit 1 and 2 (OC_{C1} and OC_{C2})	0.2
CA	−0.4
EA	−0.4

Table 5 List of faults with fault counts for datasets.

Dataset name	# observations	# UC_{C2}	# OC_{C1}	# CA	# EA	# NF
Dataset 1	3336	1577	1032	2013	607	171
Dataset 2	2099	837	475	451	237	99

Table 6 Fault labels and observation percentages of Dataset 1.

(%)

Fault class	Percentage of observation
UC_{C2}	4.9
OC_{C1}	12.3
CA	0.1
EA	15.9
NF	5.2
EA + CA	0.1
EA + CA + OC_{C1}	0.5
CA + OC_{C1}	17.2
CA + UC_{C2}	42.1
EA + CA + UC_{C2}	0.6
EA + UC_{C1}	1.1

Similar ML methods for FDD approaches have linked imbalanced datasets (due to the lack of faulty data) to poor model performance[2, 10, 11, 15]. Our approach has been rigorous to this challenge and the results for the proposed semi-supervised ML approach are promising.

3.3 Feature selection

Feature selection was employed to improve model performance. A confusion matrix and a classification accuracy table were generated to determine the model accuracy with and without feature selection. The accuracy of the model was assessed by 10-fold cross validation, and feature selection was performed using the correlation matrix of the Pearson correlation filtering method. This filtering method works by calculating the linear relationship between the independent variables (input features) and dependent variable (output or label) and has a correlation coefficient between −1 and 1. For each dependent variable, which represents fault class in this study and has a numerical value of FI, the correlation matrix was investigated to select a highly correlated set of independent variables (features). 24 corelated features were selected considering all fault classes ("labels"). This resulted in the highest increase in accuracy, and this new subset of features was used for both datasets. The 24 features selected by the feature selection method along with the four selected faults and NF were used to build the semi-supervised ML methods. Reducing the number of features results in a reduced cost for data collection, since the proposed model can achieve high accuracy using a subset of the original 30 features.

4 Methodology

This section outlines the developed ML FDD approaches for HVAC rooftop system. First, SVM, which is a supervised learning algorithm, was trained on the dataset. This supervised ML method served as a baseline, so that the performance of the other methods developed in this paper could be compared against it. Then, two novel semi-supervised ML approaches were developed. Since this paper includes multiple methods, there was a need to compare the performance of these methods; therefore, a tradeoff was used to select the best performing method for each fault type. Table 7 shows the training and test datasets for Methods 1, 2, and 3. Methods 1 and

Table 7 Description of training and test datasets for the developed ML methods.

Method	Data type	Description
Method 1	Training data	25–30 observations (labeled data) selected randomly for each fault class
(Dataset 1)	Test data	The remaining observations for each fault class
Method 2	Training data	25–30 observations (labeled data) selected randomly for each fault class + k-NN data points.
(Dataset 1)	Test data	The remaining observations for each fault class
Method 3	Training data (labeled data)	25–30 observations (labeled data) selected randomly for each fault class + k-NN data points.
(Dataset 2)	Test data	The remaining observations for each fault class

2 are applied on Dataset 1, while Method 3 uses Dataset 2. For each method, 25–30 observations were selected randomly for each fault class to form the training datasets, ensuring that both classes (faulty and non-faulty) were sampled (not necessarily uniformly). The data points in the selected observations for the training datasets are labeled based upon the methodology explained in Section 2. The remaining observations for each fault were used as the test datasets. The following subsections illustrate all the methods in detail.

4.1 Supervised learning

In this study, SVM was used to train on the training dataset and predict the test dataset. SVM was selected out of the other supervised methods because it performed the best with an overall accuracy of 96.2% in a related study[2]. SVM is a supervised ML algorithm that can solve both linear and nonlinear problems. This method works by taking the dataset as an input and outputting a line, or a set of lines for multi-classification that separates the classes within the data. It starts by first drawing a line that acts as a generalized separator, then SVM finds the closest points from each class to the separating line and calls these data points support vectors. The distance between the separating line and the support vectors is called the margin, and the goal of SVM is to maximize this margin until an optimal hyperplane is found. Furthermore, SVM tries to define a decision boundary between the classes that is as wide as possible (largest margin), which is performed to minimize misclassification. Additionally, SVM can handle linearly inseparable models by increasing the dimensionality of the model until it becomes linearly separable. Once this is achieved the decision boundary is then projected back to the original dimension. The process of finding the optimal transformation in the model is achieved by selecting the kernel in the SVM algorithm. A kernel calculates the dot product of two vectors within the data to measure the correlation between them. One of the challenges of using SVM is selecting the right kernel function[32]. The SVM model used in this paper utilized the default tuning parameters; however, the C parameter was the only parameter that was tuned. This parameter represents the tradeoff between a smooth decision boundary and classifying training points correctly. As C increases, the model classifies more training points correctly. However, when specifying the C parameter, one must consider that a high value can lead to overfitting, since the model will not be general enough. Additionally, large values of C increase the penalty on SVM for misclassification.

Method 1: Support vector machine. As mentioned before SVM is a supervised learning algorithm. This differs from unsupervised learning algorithms in that the algorithm can see the labels of the training data points. Therefore, supervised learning is ideal for classification, especially when the true labels are necessary[33]. The SVM method used in this study was composed of five binary classifiers. The output 0 corresponds to “no fault present”, and the output 1 corresponds to “fault present”.

For each classifier we only explored one specific fault at a time. The training and test datasets were created based upon the methodology explained previously in Section 4 and shown in Table 7. As illustrated in Fig. 2, the training dataset was used to train the SVM classifier. After training the classifier, the SVM model was used to predict the fault class for each observation in the test dataset. This process was then repeated for all the fault classes, and the accuracy of this method was calculated by taking the average of the accuracies of the five classifiers.

4.2 Semi-supervised learning

After training the supervised ML model using SVM, the next step was to develop a semi-supervised ML approach. The semi-supervised ML approach is desired in ML applications such as FDD because labeled data is difficult and expensive to obtain. A small number of labeled training data points is desirable for two reasons. Based on normal operating conditions in the building, some RTUs operate only for a few minutes because the other RTUs in the building are taking care of the cooling

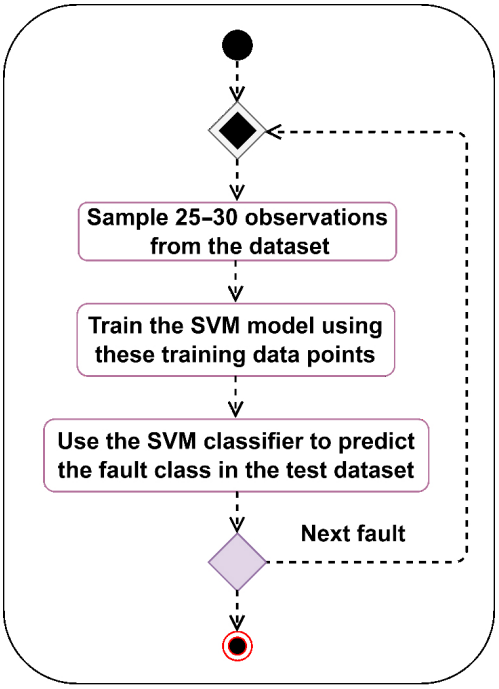


Fig. 2 Method 1: support vector machine.

load; therefore, it is hard to gather many data points out of these RTUs. For a laboratory test set up, it will be expensive to run the experiment multiple times until enough data is collected.

This paper presents two semi-supervised ML models. In both models the supervised learning method used is SVM. The two models differ in the unsupervised learning method. The first model uses a novel unsupervised k-NN labeling approach. This method can be applied to any dataset that contains single or multiple faults at once. The second model uses clustering and can be applied to datasets that contain one fault at a time (i.e., no more than one fault per observations).

Semi-supervised learning using unsupervised k-NN labeling. This unsupervised classification method is a novel ML classification approach. This approach utilizes the k-NN algorithm to label the k-closest data points to the few training data points that are given (using Euclidean distances). This method expands the training dataset, which results in a higher model accuracy. This approach is ideal for situations where only a few data points are labeled. These few labeled data points are used as training data points, and for each one of these training data points we find the closest k-points to it, and label them with the same label as the training data point (since it is known to us). This approach enhances the model performance, especially when it is paired with clustering (refer to Method 3).

Semi-supervised learning using clustering. The second semi-supervised learning method used in this paper utilizes clustering to group the data points into clusters. There are various clustering techniques that can do this (k-means, k-medoids, etc.). Clustering of the data is beneficial to us since the data points are partitioned into subgroups, each subgroup containing data points that are similar to each other. Similarity is usually measured by distance. Therefore, data points that are closest to each other will most likely belong to the same cluster. There are five clusters representing the four faults, and the non-faulty class in the dataset. Furthermore, different clustering techniques have different objectives. In this work we utilized k-medoids clustering techniques. The objective of this clustering method is to produce clusters that minimize the sum of dissimilarities between a given data point, and the cluster center it is assigned to Ref. [34] (In k-medoids clustering centers are actual data points).

Method 2: Combination of SVM and unsupervised learning of the *k*-NN labeling. In this method (illustrated in Fig. 3), the training dataset was expanded by applying a novel unsupervised ML method. Nonetheless, the combination of the supervised SVM and the unsupervised k-NN methods makes this method a semi-supervised ML method. This method works by measuring the distances of the closest unlabeled data points in the dataset to the training data points (labeled data), then the k-closest data points to each training data point were labeled with the same label as the training data point. Each classifier in this method explores one specific fault at a time. This method can classify more than one fault in the same observation, by combining the classifications of all the binary classifiers together. Method 2 involves four main steps: (1) the training and test datasets were created based upon the methodology explained previously in Section 4 and shown in Table 7; (2) the sampled data points in the training dataset were then used to label their k-closest neighbors; this resulted in an increase in the size of the training dataset; (3) the new training dataset, which is composed of the sampled data from the original training dataset and the output from step (2), was used to train the SVM classifier; (4) after training the classifier, the SVM model was used to predict the fault class for each observation in the test dataset. This process was repeated for all the fault classes, and the accuracy of this method was calculated by taking the average of the accuracies of the five classifiers.

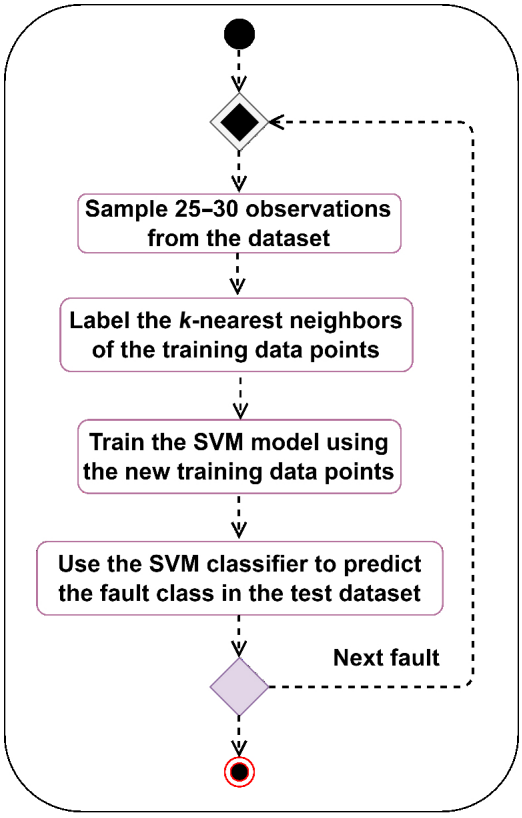


Fig. 3 Method 2: Combination of SVM and unsupervised learning of the k-NN labeling.

Method 3: Combination of SVM, clustering, and unsupervised learning of the k-NN labeling. In this method (illustrated in Fig. 4), the sampled training dataset was expanded by applying a novel unsupervised ML method. Before applying this method, the search space for the labeling of the new data points was constrained by applying unsupervised clustering, using k-medoid. This constrained the unsupervised labeling of the k-NN to only label the closest k-points to each training point that falls within the same cluster. This method was only applied to the multiclass problem, where it is assumed that each observation only contained one fault. This was the case for the Dataset 2 as illustrated in Table 2. Method 3 involves five main steps: (1) the training and test datasets were created based upon the methodology explained previously in Section 4 and shown in Table 7; (2) the data was clustered using an unsupervised clustering method; (3) for each cluster (corresponding to a fault), the sampled data points in the training dataset were then used to label their k-closest neighbors within the same cluster, this resulted in an increase in the size of the training dataset; (4) this new training dataset, which is composed of the sampled data points from the original training dataset and the output from step (3), was used to train the SVM multi-class classifier; (5) after training the classifier, the SVM model was used to predict the fault class for each observation in the test dataset.

4.3 Tradeoff between Methods 1 and 2 for higher fault classification accuracy

Since multiple data-driven ML FDD methods were developed, the need for a tradeoff between these methods was necessary. Therefore, this tradeoff method (illustrated in Fig. 5) was developed to select the best performing method for each fault type for Methods 1 and 2. Method 3 was not considered since it cannot classify faults simultaneously. First, for each of the five fault categories, the data was split into a training dataset and test dataset based upon the methodology explained previously in Section 4 and shown in Table 7, which were used for both Methods 1 and 2. Next Methods 1 and 2

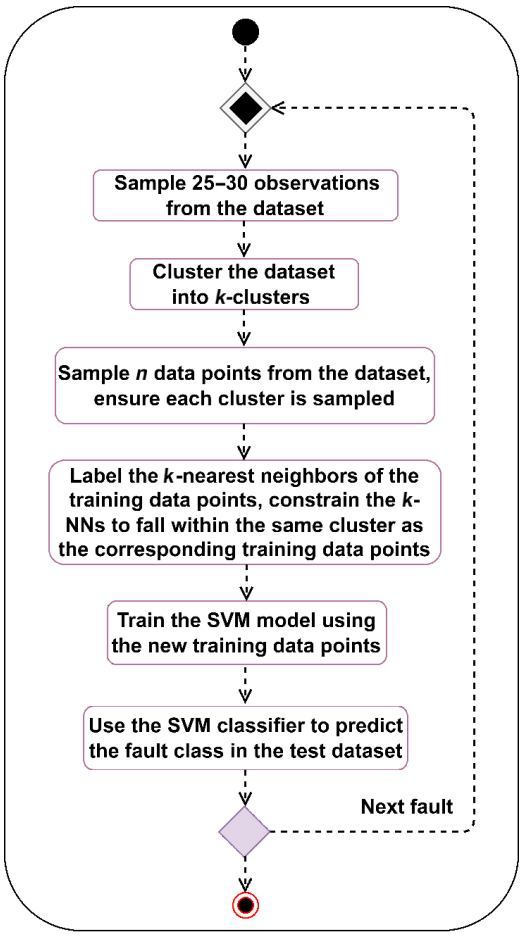


Fig. 4 Method 3 process: Combination of SVM, clustering, and unsupervised learning of the k-NN labeling.

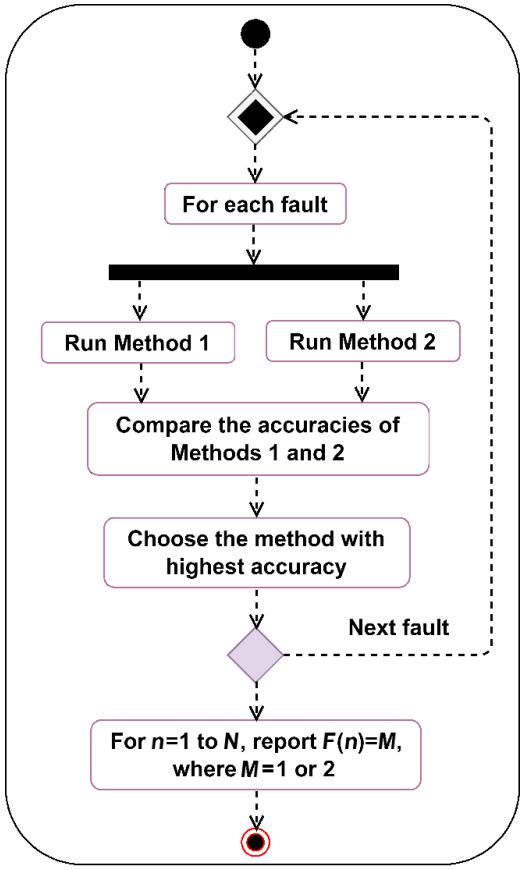


Fig. 5 Tradeoff method, N is number of faults an M is either Method 1 or 2.

were applied using the training and test datasets, and the accuracies of the two methods were compared by fault class. The method with the highest accuracy was chosen for each fault class. Using this method, the accuracy of this method is equivalent to the highest accuracy of either Method 1 or Method 2 by fault class.

5 Result and Discussion

This section highlights the recorded accuracies for each of the three methods that were implemented in this study for both datasets. Method 1 was used as a baseline to compare the accuracies of the other methods, while the tradeoff method was used to select a method (Method 1 or Method 2) with the highest accuracy for each fault.

5.1 Method 1: Support vector machines

The binary classification method is a supervised ML method that uses SVM to train on 25–30 sampled observations for each fault (for a total of five classifiers). The average accuracy of this classification method was calculated to be 93.5% by taking the average of the five accuracies. Table 8 shows the accuracies for each fault class as well as the NF class, and the overall averaged accuracy. This method performed very well for the refrigerant fault for circuit 2; however, the accuracy of NF is low. The accuracy of NF is highly impacted by the imbalanced data as illustrated in Table 5. This is expected since the Dataset 1 has only 171 observations labeled as NF. No attempts were made using GAN or any other techniques developed in Refs. [2, 10, 11, 15] to deal with imbalanced data since the focus of this work is semi-supervised ML methods, where not all data points are labeled.

Table 8 Fault accuracy of Method 1.

Fault	Accuracy (%)
UC_{C2}	99.9
OV_{C1}	95.4
CA	93.9
EA	97.5
NF	80.6
Average	**93.5**

5.2 Method 2: Combination of SVM and unsupervised learning of the k-NN labeling

The following binary classification method is a semi-supervised ML method that uses unsupervised labeling of the k-NN to expand the training dataset. Then SVM was used to train on the new training dataset for each fault (for a total of five classifiers). Like Method 1, this method uses 25–30 training observations, then using unsupervised labeling of the k-NN, a new set of data points are added to the training dataset. The average accuracy of this method was calculated by taking the average accuracies of the five accuracies, which was 94.9%. Table 9 shows the accuracies by fault class. Method 2 shows a higher average accuracy than Method 1; however, the individual fault accuracies are slightly lower for UC_{C2}, OC_{C1}, and EA. The big improvement in comparison to Method 1 is the NF accuracy. Method 2 has an NF accuracy of 91.8%, while the NF accuracy of Method 1 is 80.6%. This is very promising and means this method can handle the

Table 9 Fault accuracy of Method 2.

Fault	Accuracy (%)
UC_{C2}	99.8
OV_{C1}	92.3
CA	93.9
EA	96.7
NF	91.8
Average	**94.9**

imbalanced dataset problem with no further techniques to generate new NF data points.

5.3 Method 3: Combination of SVM, clustering, and unsupervised learning of *k*-NN labeling

This semi-supervised ML method works well when there is only one fault per data point, and so was applied to Dataset 2. This is rarely possible for an RTU running under normal operating conditions due to having no control over fault behavior; however, it is an efficient method when only a small number of labeled data points are available. As illustrated in Fig. 6, the accuracy of this method is highly dependent on the k value in k-NN. The highest average accuracy was achieved using $k = 50$. A confusion matrix, shown in Figs. 7 and 8, is used to investigate where this method is misclassifying the five fault categories. The confusion matrix shows the correct predictions in the diagonal elements and incorrect predictions in the off-diagonal elements. The OC_{C1} and UC_{C2} are predicted with 100% accuracy, while there were some misclassifications for CA, EA, and NF. For CA, six of the test data points are classified

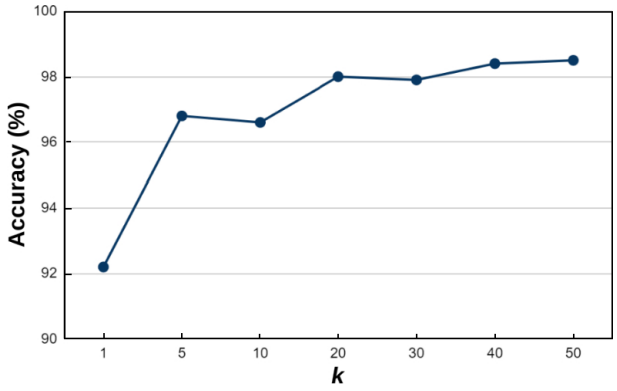


Fig. 6 Accuracy of Method 3 as a function of k (k-medoids).

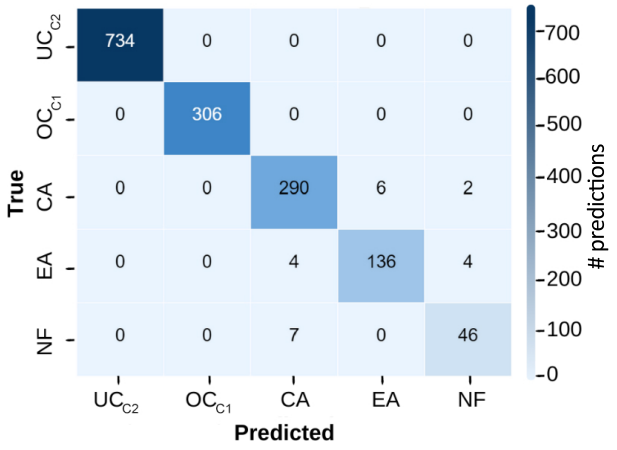


Fig. 7 Confusion matrix for Method 3 without normalization.

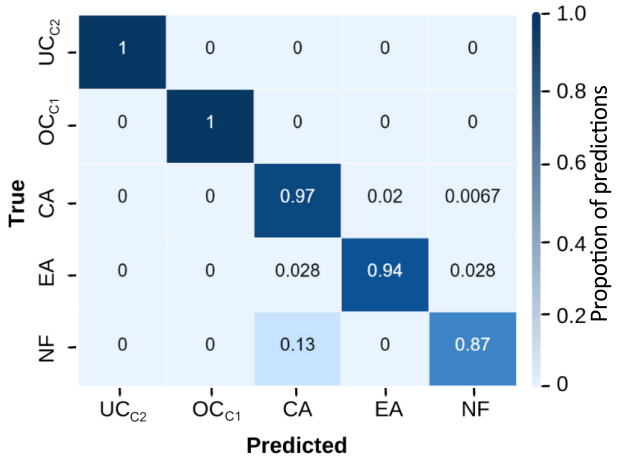


Fig. 8 Normalized confusion matrix for Method 3.

as EA and two as NF, while for EA, four of the test data points are classified as CA and four as NF. This method also misclassified seven of the NF testing data points as CA. Although the NF accuracy with 25–30 labeled observations is low (87% shown in Fig. 8) in comparison to the other faults, the overall accuracy of this method is promising when compared to results of other studies with even larger numbers of labeled data points. For instance, Ref. [23] reported only an 84% accuracy with 84 labeled data points. This method can be further improved if the data imbalance is addressed by using methods such as GAN or oversampling of the minor class.

5.4 Tradeoff method

In this method, the accuracy is calculated by using Methods 1 and 2 and including the method that achieves a higher accuracy for each specific fault. Accuracies of this method for each fault class are listed in Table 10. The tradeoff is a good technique to use when there are two or more methods that can achieve higher accuracy when alternatively combined with each other. The method with the highest accuracy for each fault is chosen to be the model of choice for FDD for that fault. The average accuracy achieved by tradeoff method was 95.7%.

Table 10 Tradeoff method accuracy.

Fault	Selected method	Accuracy (%)
UC_{C2}	Method 1	99.9
OC_{C1}	Method 1	95.4
CA	Method 2	93.9
EA	Method 1	97.5
NF	Method 2	91.8
Average		**95.7**

6 Conclusion, Limitations, and Future Work

In this paper we have presented different ML methods for FDD for HVAC systems, for decision making when only a small number of labeled data points exist. Datasets can be expensive and difficult to obtain, and if they are available, labeling them is even harder and requires an in-depth knowledge of RTU behavior and thermodynamic principles. In this work, one supervised learning method and two semi-supervised learning methods were developed. The focus of this paper is semi-supervised ML methods to address the lack of labeled data points; however, a supervised ML method was also developed as a baseline for comparison purposes. All the developed methods were able to classify all seven considered faults categories (UC_{C1}, OC_{C1}, UC_{C2}, OC_{C2}, CA, EA, NF); however, only five classes are identified and analyzed because there was no instance in the datasets for the UC_{C1} and OC_{C2} faults. The average fault classification accuracy of the supervised ML method for the baseline method was high (93.5%); however, the minority class (NF) classification accuracy was low (80.6%) because of the data imbalance. This low fault classification accuracy can be addressed in future studies by utilizing oversampling techniques on the minority class. A combination of SVM and a novel unsupervised ML technique that utilizes k-NN labeling (Method 2) was developed. This method is very promising, as it shows a high average accuracy (94.9%) even with a few labeled data points and it can predict multiple faults in the same data point. This method also shows encouraging results for dealing with imbalanced datasets without the need for additional techniques to generate new data points to balance all classes. A combination of SVM, clustering, and unsupervised learning of k-NN labeling (Method 3) was developed. This method is limited to a scenario where only one fault at a time is present in the dataset; however, it is a powerful approach to deal with limited labeled data points. The highest average accuracy was achieved using k-NN with $k = 50$. Interestingly, all OC_{C1} and UC_{C2} testing data points are correctly predicted, while there were a few data points that were misclassified for CA, EA, and NF. Even though the imbalanced dataset challenge can be handled by using different techniques, the main drawback of this method is the presence of multiple faults in the same observation. Finally, a tradeoff method was developed to select between Methods 1 and 2 for each fault type. Rather than looking at the overall accuracy of each method, this method looks at the accuracy of each individual classifier (one classifier for each fault class). This is useful when it is necessary to select between different methods (SVM or a combination of SVM and unsupervised ML of k-NN labeling) for each classifier, to achieve better predictions, and an overall higher average accuracy.

The developed methods in this paper perform best when used for RTUs with similar system specifications (e.g., refrigerant, number of compressors, compressor type, expansion device, number of fans, etc.). These methods have not been tested on datasets collected from differing RTU types. Future work could focus on whether these methods can be generalized to other RTU types or datasets with multiple different types of RTUs. Another limitation of this work is the list of considered faults. A total of six faults for both circuits were only considered since the aim of this project was to develop a high accuracy and robust ML FDD methods given only a few labeled observations. However, more faults can be added with the availability of more fault labels. The proposed methods in this work require only a few labeled datapoints. Reliable data labels are important to build a robust model to correctly predict faults. The thresholds of fault severity are estimated based on the previous literature and used to label the datasets for this study; however, the datasets used in this paper are composed of real data collected from the RTU operating under typical operating conditions and are different from the data used in the literature (simulated data). Intensive research has been performed to produce generalization effects of FI values using simulation or experimental setup; however, more work is needed to verify the fault severity on COP or other performance metrics for a real air conditioning unit in the field. The application of the work presented in this paper has a high potential to reduce lifecycle costs for HVAC systems. Building owners and managers can hire a technician to validate the soft faults classified by the models developed in this paper rather than wait for a hard, more expensive fault to occur and incur higher energy costs due to suboptimal operation.

Acknowledgment

This work was supported in part by the US Department of Energy (No. DE-EE0008189) and the National Science Foundation (Nos. 1743418 and 1843025).

References

[1] What's on Your Roof? Rooftop Unit (RTU) Efficiency Advice and Guidance from the Advanced RTU Campaign, https://www.energy.gov/eere/buildings/articles/what-s-your-

roof-rooftop-unit-rtu-efficiency-advice-and-guidance-advanced, 2021.

[2] A. Ebrahimifakhar, A. Kabirikopaei, and D. Yuill, Data-driven fault detection and diagnosis for packaged rooftop units using statistical machine learning classification methods, *Energy Build.*, vol. 225, p. 110318, 2020.

[3] J. Braun and D. Yuill, FDD Evaluator 0.1, Ray W. Herrick Laboratories-Purdue University, West Lafayette, 2013.

[4] S. Deshmukh, S. Samouhos, L. Glicksman, and L. Norford, Fault detection in commercial building VAV AHU: A case study of an academic building, *Energy Build.*, vol. 201, pp. 163–173, 2019.

[5] J. Granderson, R. Singla, E. Mayhorn, P. Ehrlich, D. Vrabie, and S. Frank, *Characterization and Survey of Automated Fault Detection and Diagnostics Tools*. Berkeley, CA, USA: Lawrence Berkeley National Laboratory, 2017.

[6] J. Wall and Y. Guo, *Evaluation of Next-Generation Automated Fault Detection & Diagnostics (FDD) Tools for Commercial Building Energy Efficiency*. CSIRO Energy for Low Carbon Living CRC, 2018.

[7] J. Granderson, G. Lin, R. Singla, E. Mayhorn, P. Ehrlich, D. Vrabie, and S. Frank, *Commercial Fault Detection and Diagnostics Tools: What They Offer, How They Differ, and What's Still Needed*. Berkeley, CA, USA: Lawrence Berkeley National Laboratory, 2018.

[8] W. Kim and S. Katipamula, A review of fault detection and diagnostics methods for building systems, *Sci. Technol. Built Environ.*, vol. 24, no. 1, pp. 3–21, 2018.

[9] C. Robinson, B. Dilkina, J. Hubbs, W. Zhang, S. Guhathakurta, M. A. Brown, and R. M. Pendyala, Machine learning approaches for estimating commercial building energy consumption, *Appl. Energy*, vol. 208, pp. 889–904, 2017.

[10] K. Yan, J. Huang, W. Shen, and Z. Ji, Unsupervised learning for fault detection and diagnosis of air handling units, *Energy Build.*, vol. 210, p. 109689, 2020.

[11] K. Yan, C. W. Zhong, Z. Ji, and J. Huang, Semi-supervised learning for early detection and diagnosis of various air handling unit faults, *Energy Build.*, vol. 181, pp. 75–83, 2018.

[12] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321–357, 2002.

[13] K. Yan, A. Chong, and Y. Mo, Generative adversarial network for fault detection diagnosis of chillers, *Build. Environ.*, vol. 172, p. 106698, 2020.

[14] K. Yan, Chiller fault detection and diagnosis with anomaly detective generative adversarial network, *Build. Environ.*, vol. 201, p. 107982, 2021.

[15] A. Ebrahimifakhar, D. Yuill, and A. Kabirikopaei, Application of machine learning classification methods in fault detection and diagnosis of rooftop units, in *Proc. 18^{th} Int. Refrigeration and Air Conditioning Conf.*, West Lafayette, IN, USA, 2021, p. 2137.

[16] F. Cheng, W. Cai, X. Zhang, H. Liao, C. Cui, Fault detection and diagnosis for air handling unit based on multiscale convolutional neural networks, *Energy Build.*, vol. 236, p. 110795, 2021.

[17] K. P. Lee, B. H. Wu, and S. L. Peng, Deep-learning-based fault detection and diagnosis of air-handling units, *Build. Environ.*, vol. 157, pp. 24–33, 2019.

[18] H. Wang, D. Feng, and K. Liu, Fault detection and diagnosis for multiple faults of VAV terminals using self-adaptive model and layered random forest, *Build. Environ.*, vol. 193, p. 107667, 2021.

[19] R. Chintala, J. Winkler, and X. Jin, Automated fault detection of residential air-conditioning systems using thermostat drive cycles, *Energy Build.*, vol. 236, p. 110691, 2021.

[20] I. Cohen, F. G. Cozman, N. Sebe, M. C. Cirelo, and T. S. Huang, Semisupervised learning of classifiers: Theory, algorithms, and their application to human-computer interaction, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 12, pp. 1553–1566, 2004.

[21] M. S. Mirnaghi and F. Haghighat, Fault detection and diagnosis of large-scale HVAC systems in buildings using data-driven methods: A comprehensive review, *Energy Build.*, 229, p. 110492, 2020.

[22] C. Fan, X. Liu, P. Xue, and J. Wang, Statistical characterization of semi-supervised neural networks for fault detection and diagnosis of air handling units, *Energy Build.*, vol. 234, p. 110733, 2021.

[23] B. Li, F. Cheng, X. Zhang, C. Cui, and W. Cai, A novel semi-supervised data-driven method for chiller fault diagnosis with unlabeled data, *Appl. Energy.*, vol. 285, p. 116459, 2021.

[24] M. Albayati, R. Gorthala, A. Thompson, P. Patil, and A. Hacker, Bringing automated fault detection and diagnostics tools for HVAC&R into the mainstream, *J. Eng. Sustain. Build. Cities*, vol. 1, no. 3, p. 030902, 2020.

[25] W. Kim and J. E. Braun, Performance evaluation of a virtual refrigerant charge sensor, *Int. J. Refrig.*, vol. 36, no. 3, pp. 1130–1141, 2013.

[26] W. Kim and J. E. Braun, Extension of a virtual refrigerant charge sensor, *Int. J. Refrig.*, vol. 55, pp. 224–235, 2015.

[27] H. Li and J. E. Braun, Development, evaluation, and demonstration of a virtual refrigerant charge sensor, *HVAC&R Res.*, vol. 15, no. 1, pp. 117–136, 2009.

[28] M. Mehrabi and D. Yuill, Generalized effects of faults on normalized performance variables of air conditioners and heat pumps, *Int. J. Refrig.*, vol. 85, pp. 409–430, 2018.

[29] M. Mehrabi and D. Yuill, Generalized effects of refrigerant charge on normalized performance variables of air conditioners and heat pumps, *Int. J. Refrig.*, vol. 76, pp. 367–384, 2017.

[30] D. P. Yuill and J. E. Braun, Evaluating the performance of fault detection and diagnostics protocols applied to air-cooled unitary air-conditioning equipment, *HVAC&R Res.*, vol. 19, no. 7, pp. 882–891, 2013.

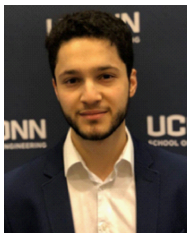
[31] A. Thompson, R. Gorthala, M. Albayati, and P. Pati, RTU-HVAC real-time operating data from unit in field, https://data.mendeley.com/datasets/9h6gpbhj5k/1, 2021.

[32] L. M. R. Baccarini, V. V. R. E Silva, B. R. de Menezes, and W. M. Caminhas, SVM practical industrial application for mechanical faults diagnostic, *Expert Syst. Appl.*, vol. 38, no. 6, pp. 6980–6984, 2011.

[33] V. N. Vapnik, *The Nature of Statistical Learning Theory*, New York, NY, USA: Springer, 2000.

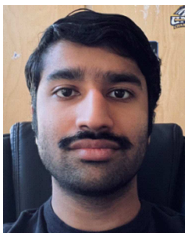
[34] P. Arora, Deepali, and S. Varshney, Analysis of K-means and K-medoids algorithm for big data, *Proced. Comput. Sci.*, vol. 78, pp. 507–512, 2016.

Mohammed G. Albayati received the BS degree from the University of Tikrit, Iraq in 2008, the MS degree in mechanical engineering from the University of New Haven, USA in 2016, and the Graduate Certificate in advanced system engineering from the University of Connecticut, USA in 2020. He is currently pursuing the PhD in mechanical engineering and works as a graduate research assistant at the Institute for Advanced Systems Engineering, the University of Connecticut. His research interests include model-based systems engineering with applications in aerospace and manufacturing. He also conducts research related to building energy efficiency and fault detection and diagnostics for building HVAC systems. During his study at the University of New Haven, he was awarded the 2016 Mechanical Engineering Award for Superior Academic Performance. From 2008–2013 and 2017–2019, he served as a project engineer at North Refineries Company in Iraq. He currently is a member and serves as the president of the INCOSE student division at the University of Connecticut.



Jalal Faraj received the BS degree in chemical engineering from the University of Connecticut in 2020, and the MS degree under the guidance of Dr. Sanguthevar Rajasekaran in computer science and engineering (CSE) Department from the University of Connecticut in 2021. He is currently working as a data science analyst at Accenture.

Amy Thompson received the BS degree in industrial engineering in 2001, the MS degree in manufacturing engineering in 2004, and the PhD in industrial and systems engineering from the University of Rhode Island in 2016. Since 2017 she has served as the associate director for the Institute for Advanced Systems Engineering and the associate professor-in-residence of systems engineering at the University of Connecticut. She also serves as the director of the SmartBuildings CT program and the assistant director for the Department of Energy Industrial Assessment Center at the University of Connecticut. Prior to joining the University of Connecticut, she served as a faculty member in systems engineering at the University of New Haven where she coordinated the BS degree in systems engineering. From 2014 to 2017 she was the owner and president of Paguridae LLC which provided operations and sustainability consulting to the utility industry. Her current research interests include the application of model-based systems engineering for the design and optimization of complex systems, fault detection and diagnostics (FDD) for HVAC-R systems; design of smart manufacturing systems, facilities, and buildings; supply chain design; and undergraduate, graduate, and online systems engineering education. She is a recipient of the US EPA Environment Merit Award in 2017.



Prathamesh Patil received the BS and MS degrees in mechanical engineering from the University of New Haven, USA respectively in 2019 and 2020. He is currently pursuing the PhD degree in mechanical engineering from the University of New Haven. His research interests include the development of energy efficient building energy technologies, building HVAC systems, thermal storage, renewable energy, mechatronics, and product development. He conducts research in applying machine learning techniques for fault detection and diagnostics for HVAC systems.

Ravi Gorthala received the MS and PhD degrees in mechanical engineering from the University of Mississippi in 1987 and 1992, respectively. He has more than 20 years of industry experience in the areas of energy efficiency and renewable energy. He joined University of New Haven in 2012 as an associate professor and served as the chair of Departments of Mechanical and Industrial Engineering for 4 years.

Sanguthevar Rajasekaran received the MEng degree in automation from the Indian Institute of Science, India in 1983, and the PhD degree in computer science from Harvard University, USA in 1988. Currently, he is the head of the Computer Science and Engineering (CSE) Department, board of trustees distinguished professor, and United Technologies Corporation (UTC) chair professor of CSE at University of Connecticut. Before joining University of Connecticut, he has served as a faculty member at Computer & Information Science & Engineering (CISE) Department of the University of Florida and Computer and Information Science (CIS) Department of University of Pennsylvania. During 2000–2002, he was the chief scientist for Arcot Systems. His research interests include big data, bioinformatics, algorithms, data mining, randomized computing, and HPC. He has published over 350 research articles in journals and conferences. He has co-authored two texts on algorithms and co-edited six books on algorithms and related topics. He has been awarded numerous research grants from such agencies as NSF, NIH, DARPA, Industry, and DHS (totaling more than $20 million). He is a fellow of the IEEE, AAAS, AAIA, and AIMBE. He is also an elected member of the Connecticut Academy of Science and Engineering.