energies

MDPI

Article

FDD in Building Systems Based on Generalized Machine Learning Approaches

William Nelson [1] and Charles Culp [2,*]

[1] Department of Mechanical Engineering, Energy Systems Laboratory, Texas AM University, College Station, TX 77843, USA
[2] Department of Architecture, Energy Systems Laboratory, Texas AM University, College Station, TX 77843, USA
* Correspondence: cculp@tamu.edu

Abstract: Automated fault detection and diagnostics in building systems using machine learning (ML) can be applied to commercial buildings and can result in increased efficiency and savings. Using ML for FDD brings the benefit of advancing the analytics of a building. An automated process was developed to provide ML-based building analytics to building engineers and operators with minimal training. The process can be applied to buildings with a variety of configurations, which saves time and manual effort in a fault analysis. Classification analysis is used for fault detection and diagnostics. An ML analysis is defined which introduces advanced diagnostics with metrics to quantify a fault's impact in the system and rank detected faults in order of impact severity. Explanations of the methodology used for the ML analysis include a description of the algorithms used. The analysis was applied to a building on the Texas A&M University campus where the results are shown to illustrate the performance of the process using measured data from a building.

Keywords: fault detection; fault diagnosis; machine learning; building systems; HVAC

check for updates

Citation: Nelson, W.; Culp, C. FDD in Building Systems Based on Generalized Machine Learning Approaches. *Energies* **2023**, *16*, 1637. https://doi.org/10.3390/en16041637

Academic Editors: Zbigniew Leonowicz and Michał Jasiński

Received: 6 January 2023
Revised: 27 January 2023
Accepted: 1 February 2023
Published: 7 February 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

This paper discusses an automated process of detecting faults in building systems using machine learning (ML) analysis. The ML process includes a classification analysis to detect faults in the system and an experimental regression analysis to estimate the severity metric of the fault's impact on the system. An outline of the process for Boolean rule FDD is shown in Figure 1a. For comparison, the ML process is shown in Figure 1b.

One of the challenges of linking mechanical analysis with ML is nomenclature. ML models use datasets in matrix form. The matrix columns represent input features, which are quantities measured from the system, such as temperatures, flowrates, or other physical/nonphysical variables. The matrix rows, called data points, represent a set of feature values at a timestamp. The n input features of each data point can be mapped onto an n-dimensional space to create the dataset. In classification algorithms, the model output is a category, called a class in ML terminology. In regression algorithms, the model output is a continuous value. Outputs from ML models are commonly called predictions, referring to the model's use of input values to estimate an unknown output value. An abstract dataset structure is shown in Table 1.

ML models perform system analysis by using subsets of the system's features. Models detecting terminal box faults will use a subset of features measured from terminal box sensors. Models detecting faults in the cooling coil will use a subset of features consisting of measurements from cooling coil and chiller sensors. Using subsets of features which differ from those in rule-based FDD can change the resolution of the analysis and detect faults sooner. To calibrate all models, which is called a training process, a collection of data points will be used which were measured from the system during a defined period, called the baseline. A model calibrated to the baseline period will detect changes in system operation relative to the baseline period.

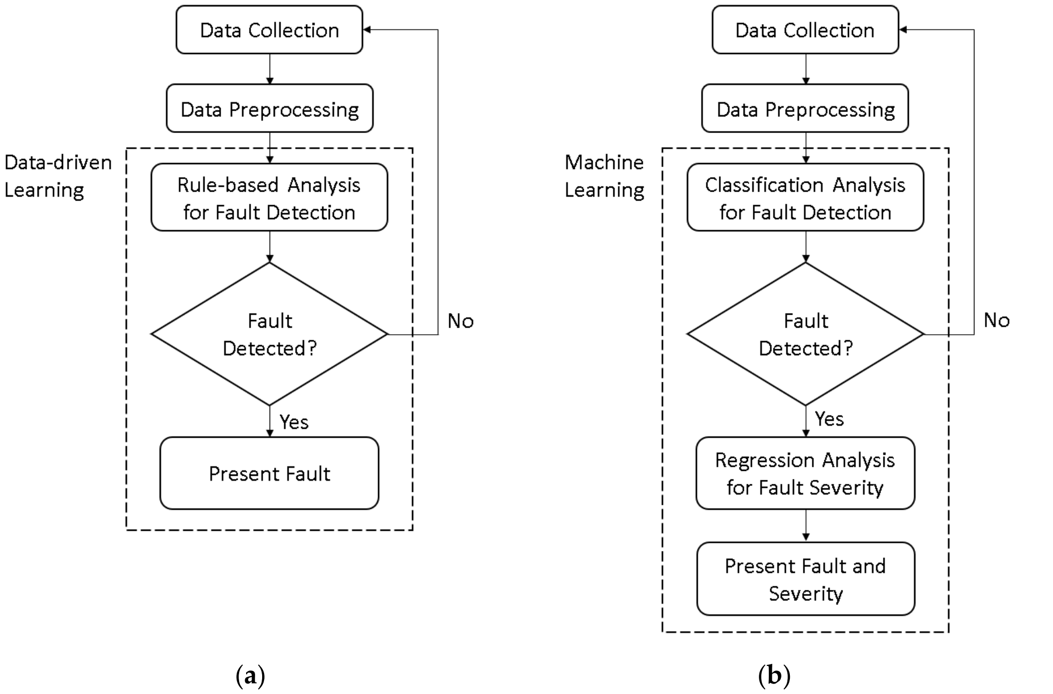


Figure 1. Automated Fault Detection and Diagnostics Process. (**a**) Original rule-based FDD process; (**b**) ML data-driven FDD process.

Table 1. Sample ML dataset.

	Feature 1	**Feature 2**	...	**Feature *n***	**Output**
Data point 1	x_{11}	x_{12}	...	x_{1n}	Y_1
Data point 2	x_{21}	x_{22}	...	x_{2n}	Y_2
Data point 3	x_{31}	x_{32}	...	x_{3n}	Y_3
...	...	...	...	...	...
Data point m	x_{m1}	x_{m2}	...	x_{mn}	Y_m

Granderson et al. [1] surveyed building operators and found that ML applications for automated fault detection and diagnostics (AFDD) in HVAC systems are underdeveloped in three major areas: generalizability, ease of use, and interpretability. A format has been established for this project which groups building data by category, such as coil command or damper position, with a common naming process. This paper describes a method that trains an ML model using categorized building data to identify changes in building behavior, which are then used to predict and classify faults in the system.

The data for this paper was collected from a Building Management System (BMS) on the Texas A&M University campus in College Station. A total of 6682 trends, with a year of data taken every 15 min, were collected from the BMS. The building may experience changes that occur with time constants less than the 15-min sampling rate. It is typical to use minimum on and off times for building equipment in the range of 5 min. The difference between the sampling rate and the command rate can cause the appearance of these measurements to look noisy. Another advantage of using thousands of samples is to add a smoothing effect for the ML predictions.

Current AFDD methods in commercial systems use rule-based methods [1]. House et al. developed the APAR ruleset to detect faults in air handling units (AHUs) [2]. The APAR ruleset consists of twenty-eight rules which detect faults during common operational states in AHUs and provide potential causes for each fault. PECI and Battelle reviewed commonly applied rules in buildings to evaluate their effectiveness [3]. Wang et al. implemented a residual-based, exponentially weighted moving average method with Boolean rules to detect faults in AHUs [4], which was able to achieve a positive detection rate of greater than 90% for single faults.

Rule-based FDD methods are limited in scope and flexibility. To define a rule, system variables which are correlated with the fault must be predetermined. A threshold must be configured manually to detect when those variables measure anomalous values. Thresholds will change between system components and should be reconfigured for each application.

ML approaches to AFDD have grown in popularity in recent years [5,6]. Bode et al. found that a balanced dataset, which contains an equal number of data points in faulty and baseline states, is important while training ML algorithms [7]. Several researchers have applied ML analysis to the ASHRAE RP-1043 dataset to detect faults [8–10]. These projects confirm the ability of ML models to detect faults when the scope of their analysis is a single dataset. Concerns remain regarding ML's capability to analyze datasets from multiple buildings with a single implementation [1]. Jang et al. developed a model to predict heating energy consumption in a building using long short-term memory (LSTM) models [11]. Their models were able to predict consumption with a 17.6% CVRMSE and 0.6% MBE.

This study used classification and regression for the analysis. The algorithms use first-principles equations to produce hybrid models. Tidriri reported several studies that show that the performance of data-driven methods is dependent on the training data, while the performance of model-driven methods is dependent on the mathematical model used in the analysis [12]. Tidriri proposed that a hybrid model, which combines data- and model-driven methods, can perform better than either would individually.

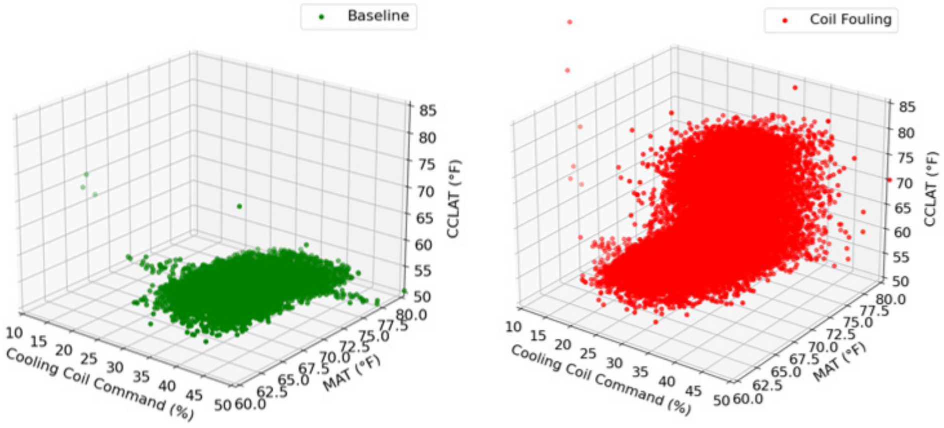
The ML applied in this study analyzes all mechanical equipment, such as air handlers or terminal boxes, in the system individually. A data format based on trend names, which groups trends by their category, is used to identify component trends for model training. System faults are defined using data categories which enables faults to be detected across different buildings and system configurations.

2. Technical Development

2.1. Classification

Classification algorithms are used to categorize a set of input values. This is achieved by comparing the similarities between input values from the baseline dataset to new input data. Datasets used for classification are constructed with a set of input variables, called features, and an output class, which is a categorical value dependent on the dataset's input variables. Types of faults in building systems are defined by their failure mode, represented by a categorical value in fault analytics. Classification algorithms generate a mapping between the feature space and each output. The feature space of a fault-free cooling coil is displayed in Figure 2a. The feature space for a cooling coil with fouling is displayed in Figure 2b.

The example cooling coil shown has three sets of input variables: a cooling coil valve command, a cooling coil leaving temperature, and a cooling coil inlet (mixed air) temperature. The two data categories, or output classes, in Figure 2 are fault-free coil data and coil-fouling data. For this example, the fault-free class is defined as the baseline. Each output class is paired with a set of input data, which consists of system data from the same year-long baseline period. The fault-free data points are measured directly from the system, without modifications or additional simulation. The coil-fouling points are generated using a first-principles equation with reduced chilled water flowrate values to produce a new cooling coil leaving temperature. In the fouled-coil dataset, this new temperature value replaces the measured cooling coil leaving temperature value from the baseline dataset. A second fault manifestation was simulated by increasing the cooling coil valve command, which may be measured from data if the building compensates for the fouled coil to meet the coil's temperature setpoint. One output class of data points is shown in each figure for clarity, though the two classes belong to the same dataset and will be analyzed together.



(**a**) (**b**)

Figure 2. Feature space of a cooling coil. (**a**) Baseline data points; (**b**) faulty data points.

Ongoing measurements from the BMS will be mapped to the feature space and evaluated with a probability of belonging to each class. The objective of the classification algorithm is to determine whether ongoing measurements belong to the baseline or coil fouling point category so that a determination can be made about the state of the system when those points were measured. These determinations are called predictions.

ML analysis predicts point categories for each time-dependent data point. In isolation, a single point's predicted category communicates information for one instant of time. Additional processing can be used to aggregate point predictions within a time period to generate time-based metrics such as the rate of degradation. For this paper, a metric was developed which aggregates thousands of point predictions over a desired time length, producing an overall metric representing the system state. This metric, Percent of Time, calculates the percent of timestamps which were predicted faulty and can be used to summarize system performance throughout the period. Percent of Time can be used to track the system degradation over time.

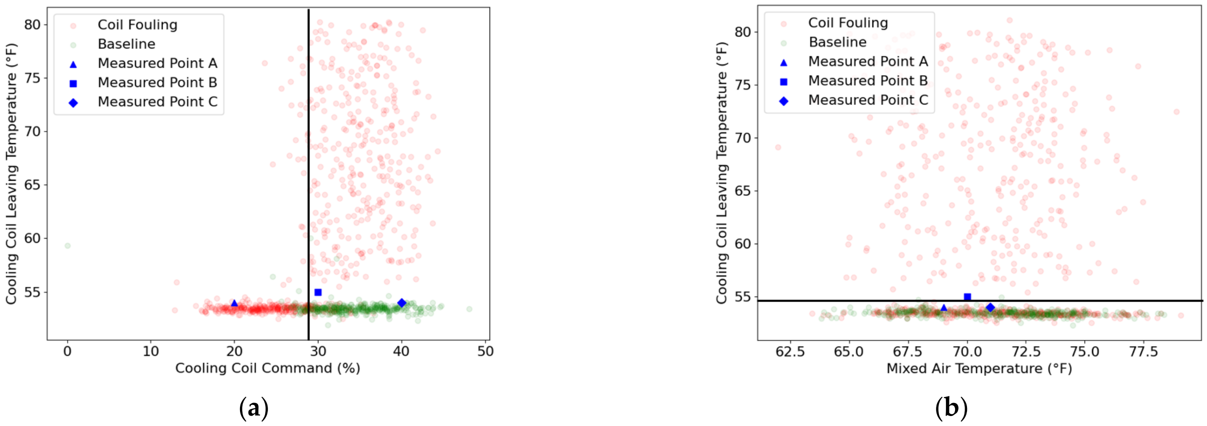
LightGBM Classification

LightGBM, which means Light Gradient Boosting Machine, is an open-source implementation of decision tree logic [13]. LightGBM performs well in structured datasets, which can be translated directly into a matrix form of rows and columns. Structured datasets, such as BMS trend data, are commonly stored on system databases. Decision trees divide the feature space into regions that are created as binary partitions [14]. Each region represents a unique prediction for the categorical output Y.

LightGBM is an evolution of the decision tree logic due to its inclusion of gradient boosting. Gradient boosting combines a group of learning models into a single learning model. The combined model is more powerful than individual models in the group. Standard implementations of the Gradient Boosted Decision Tree (GBDT) concept, such as XGBoost [15], struggle with extremely large datasets because their complexity is proportional to the number of features and data points. LightGBM automatically reduces the number of features and data points to improve computational efficiency.

The algorithm achieves this by combining two techniques: gradient-based, one-side sampling and exclusive feature bundling, which are explained in LightGBM [13]. Gradient-based, one-side sampling reduces the number of data points in the analysis. Exclusive feature bundling (EFB) reduces the total number of features in the dataset.

Figure 3 shows two decision boundaries calculated via the LightGBM algorithm using the data in Figure 2. The boundary locations are determined via an iterative process of moving the boundary across each axis and minimizing the sum of square errors after assigning the points in each region an output class.



(**a**) (**b**)

Figure 3. Sample decision boundaries for plots in Figure 3. (**a**) Cooling coil command vs. cooling coil leaving temperature; (**b**) mixed air temperature vs. cooling coil leaving temperature.

Measured points A, B, and C in Figure 3 were selected to illustrate the changes in output class probabilities between different regions of the feature space. The class probabilities of each measured point are tabulated in Table 2. Measured points A and C lie within clearly defined high-density regions of their respective class predictions, which produces a high prediction probability for each of those points. Measured point B, located between the two high-density regions, is predicted with a lower probability to each output class. The class probabilities are used to predict the point's categorization.

Table 2. Measured points in Figure 3 and their prediction probabilities.

Point Name	Baseline Probability	Coil Fouling Probability
Measured point A	0.003	0.997
Measured point B	0.7	0.3
Measured point C	0.998	0.002

In Figure 4, the prediction made by the decision tree is the class label representing baseline (zero) and coil fouling (one), the two categories of the dataset. Two boundaries defining class regions are shown for clarity in the visualization, though the full model may contain hundreds of boundaries. Figure 4 shows Figure 3a, with both prediction boundaries overlayed, and the corresponding predicted fault, represented by Y, for each region.

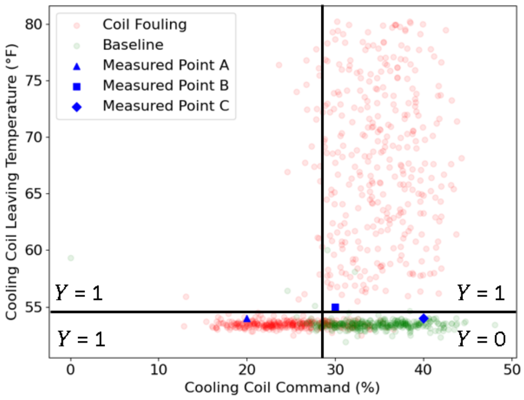


Figure 4. Prediction regions with corresponding prediction values.

2.2. Regression

While classification algorithms focus on output classes, regression algorithms generate a model to estimate an output using its relationship to the inputs and provide a measure of the data precision. Several researchers have used regression analytics to predict system consumption in different system states [16–20]. Regression model performance improves as features are added to the dataset. Features can be ranked according to their contribution to the model's output. Relative to higher-ranked features, the lower-ranked features can be removed with less added error to model predictions. To demonstrate feature weights, a regression model was trained using data generated via a 13-variable equation. Figure 5 shows the R^2 performance curve of a regression model as variable measurements are added to the model's list of features, sorted via the contribution to the R^2 value. The R^2 loss of removing feature 1 is 0.14, while the R^2 loss of removing feature 13 is 0.04. Feature 1 contributes over three times as much to minimizing the model's prediction error compared to feature 13.

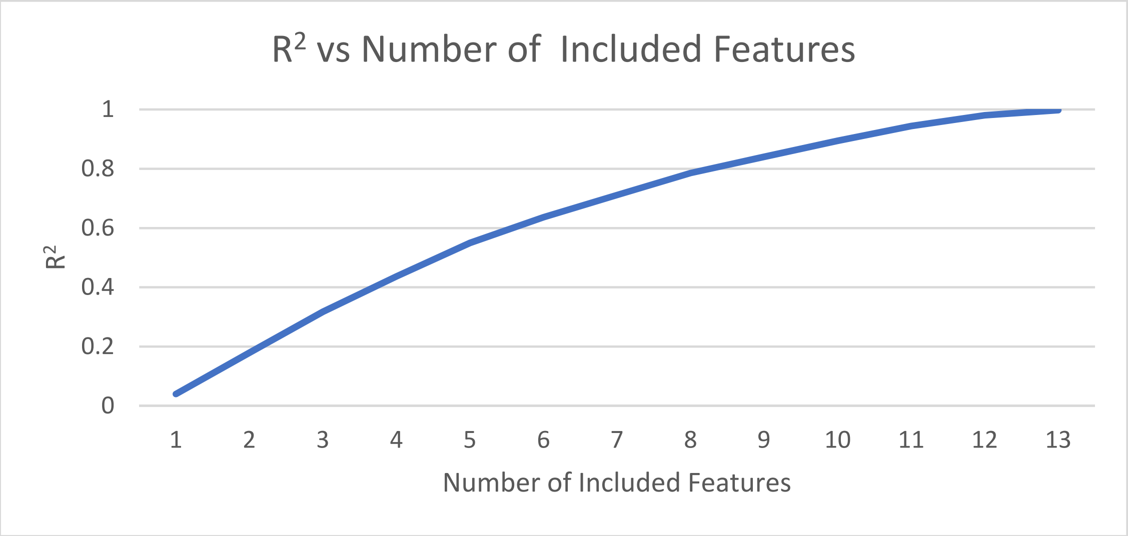


Figure 5. R^2 as a function of included features.

In a real building, measurements are interdependent. For example, the zone temperature of a room is dependent on the terminal box leaving air temperature and the volume of air being provided. Yan et al. used a principal component analysis (PCA) in a chiller model to rank the relative importance of each sensor reading for predicting the model's output [21]. The results were used to rank sensors as important to install or maintain so as to minimize model error.

A regression analysis can be used to create a model for noisy datasets. The models produced from this analysis are commonly used for predicting a system's output when given a set of system inputs. The output variable can be any measurement dependent on the inputs. The output of datasets used in this project's regression analysis is system energy consumption, which is dependent on variables including outside air temperature, internal loads, and system setpoints.

First-principles equations are used to calculate energy consumption from system measurements. At their simplest, the energy across any component is governed using Equation (1), where $\dot{m}$ is the mass flowrate of a substance (air or water), c_p is that substance's specific heat, and ΔT is the change in temperature across the component. Evaluating Equation (1) requires a set of sensor measurements, including a flowrate sensor to measure $\dot{m}$ and a pair of temperature sensors to measure T_{in} and T_{out}.

$$Q = \dot{m}c_p\Delta T = \dot{m}c_p(T_{out} - T_{in}) \tag{1}$$

Regression model forecasting can also be used to predict values in two date periods. Building system FDD defines these two periods as a baseline, which is normally one year, and the one year after the building's HVAC system has either been improved with commissioning or remained unchanged. Traditionally, only temperature measurements are used as the regressor variables for energy use. Adding a humidity measurement to the input dataset can improve the prediction.

While many faults in building systems will increase energy consumption, some faults will lower system consumption but adversely affect occupant comfort. If a high zone temperature consumes less cold air in conditioning, this fault would reduce energy use and negatively affect occupant comfort. Whether a fault reduces or increases energy consumption in the building, it represents a change in the system and should be investigated for unintentional effects on occupant comfort.

2.2.1. Advantages of Machine Learning

Developing physical models to represent buildings requires time and expert knowledge [22]. ML model training has advantages over manual calibration in its ability to automatically infer dependencies between measured and unmeasured variables. As a building ages, components and construction will degrade. Due to this degradation, unmeasured properties in the building will change, with duct leakages potentially increasing and coils potentially beginning to foul. System sensors to directly measure these properties may be missing, though their values can be estimated using related sensors and a physical model of the system.

Temperature-based economizer controls can be defined as a function of the outdoor air temperature, return air temperature, and outdoor air damper command, which can be monitored to detect faults in the economizer. The relationships between the system measurements will change in a malfunctioning economizer. For example, a faulty economizer may mix a higher fraction of outside air into the system in hot external conditions when the damper is expected to minimize outside air in the system according to ASHRAE Standard 62.1.

Manual calibration for physical models requires a series of parameter-tuning and re-evaluation steps to minimize model errors. Typically, a 10% error between the model predictions and measured values is considered acceptable for determining savings in commercial buildings. Model calibration in ML is an automatic grid-searching process, where dozens of tuning parameter configurations are defined and evaluated to minimize model errors. These parameters are properties of the algorithm: learning rate represents the step size between calibration iterations to minimize model loss, tolerance represents the amount of acceptable error in model predictions, and the model's depth represents the number of layers in a decision tree. The expert knowledge for ML training is an understanding of each tuning parameter and the consequences of changing their values, while expert knowledge for physical model calibration requires an understanding of the building systems and the first-principles equations that govern their operation.

2.2.2. Determining Unmeasured Relationships

Both ML and physical models rely on interdependencies to correlate the relationships between measured variables and unmeasured variables. A simple example has been developed to illustrate a ML algorithm's process of estimating an unmeasured variable.

Learning the interdependencies between measured and unmeasured variables allows ML to adapt its predictions at the component level, rather than the generalized rule-based FDD which analyzes all components under a single rule. ML does this by defining a feature space which can consist of 10 or more features independent of other components in the system. Each feature must be simulated according to its behavior after a fault is introduced to the system. For example, the simulation could modify the data to have a damper in a constant position which emulates a stuck damper. The defined feature space automatically associates dependencies between system variables to analyze the system.

Consider a terminal box model which is used to predict the total energy output from the terminal box output to the space. The terminal box is damper-controlled with airflow provided by the AHU's supply air fan. Relevant measured data from the system includes the supply air temperature, space temperature, terminal box damper command, supply duct static pressure, and supply air VFD command. The terminal box's energy impact is dependent on the temperature and flowrate of the air output to the zone as well as the space temperature in the zone, which is used to calculate ΔT in Equation (1). The supply air temperature is measured by the system and is included as a model input. The supply air flowrate is unmeasured in the system and must be inferred using dependent system measurements. Figure 6 illustrates the measurements in the system and the relationship between measured and unmeasured variables.

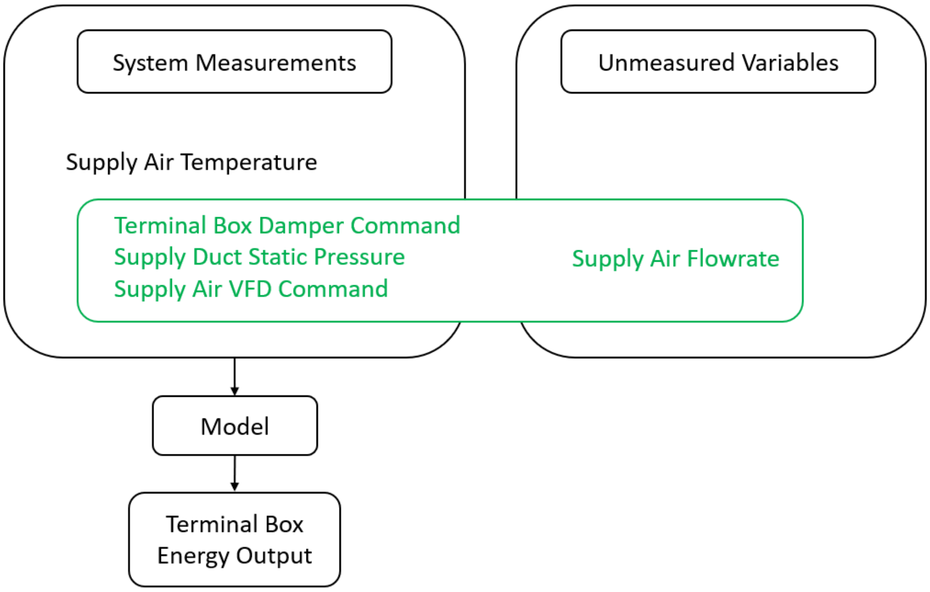


Figure 6. System measured and unmeasured variables.

Because the flowrate is directly dependent on the terminal box damper command, supply duct static pressure, and supply air VFD command, the value of the supply air flowrate can be inferred using those three measurements. The supply air flowrate is also indirectly dependent on space temperatures. Space temperatures failing to meet the setpoint may require a higher flowrate of supply air to condition zone loads. The supply air temperature is excluded from the supply air flowrate correlations in this simple model to reduce computational time because air densities had less than a 2% impact on the output variable. If higher precision models are required, the supply air temperature could be included.

Table 3 shows results from two models trained to predict the terminal box energy output. The Used For column contains variable names which have been color-matched to relevant features in the Features column. Model 1 used all three green dependent variables to learn and estimate the supply air flowrate. Model 2 used one green dependent variable to learn and estimate the supply air flowrate. The energy transfer for the supplied air is calculated using the first-principles energy calculation and is compared to each of the two model's energy outputs. The mean squared error (MSE) squares the value of all errors of model predictions. In this example, Model 2's MSE is over 13 times Model 1's MSE.

Table 3. Model performance using two sets of features to predict terminal box energy outputs.

Model	Features	Used for	Normalized MSE
Model 1	Terminal box damper command Supply duct static pressure Supply air VFD command	Air flowrate	1
	Supply air temperature Space temperature	Temperature	
Model 2	Terminal box damper command	Air flowrate	13.48
	Supply air temperature Space temperature	Temperature	

The lower performance of Model 2 is expected because of the fewer features used to learn and estimate the supply air flowrate. The supply duct static pressure and VFD command measurements are crucial in predicting the terminal box's energy output because of their correlation to the supply air flowrate. Using features which differ from those used in a rule-based FDD analysis to train an ML model can increase the resolution of the analysis to allow for earlier fault detection. For example, a rule-based analysis for space temperature faults typically analyzes only space temperature values. Including outdoor air temperature or time of day features in an ML model can provide additional context for fault diagnostics.

The example above demonstrates that, provided the relationship between an unmeasured variable and measured variables in the system are governed by a relationship which can be modeled, ML can automatically derive the relationships between unmeasured and measured variables. Physical models can also predict a building's measured behavior but require additional manual calibration and an expert knowledge of building systems to achieve similar results.

Figure 7 illustrates the training process of a baseline model of a building system. This process was used to train the models discussed in Table 3 with the addition of weather conditions. In the flowchart, the external weather and internal conditions represent measured variables from the system. Relationships between the variables and the model output are inferred in the training process.

A regression model calibrated to the baseline period estimates the baseline system's consumption; any faults or component degradation occurring after the baseline period will exist in the evaluation period for comparison. Comparisons between the weather-normalized baseline energy consumption, which is predicted by the ML model, and the measured energy consumption can show the improvement from energy efficiency measures or the energy impact of a fault in the system. Weather-normalized baseline evaluations are a normal measurement and verification (M&V) procedure, though physical models are typically used instead of ML models. The procedure is displayed in Figure 8.

To demonstrate the magnitude of the increase in energy consumption when various faults are introduced to a building, a simulated dataset has been created. The evaluation uses data measured from the Zachry building with a simulated cooling coil fouling fault using the procedures outlined in Section 2.1: the chilled water flowrate through the coil was reduced to produce a new fouled coil dataset which is compared to the baseline. The flowrate of the system has been normalized in this example dataset. The predicted difference in total system energy consumption between the two system states was 15%. Figure 9 plots the two consumption calculations and illustrates the magnitude of their difference. The blue series represents baseline consumption, and the orange series represents the system consumption with a fouled cooling coil. The difference between the two series is the change in energy consumption in the system due to the fault. Knowing the change in energy consumption due to each fault in the system enables a building operator to prioritize faults based on a calculated impact on the system, which is normally uncalculated in rule-based analytics.

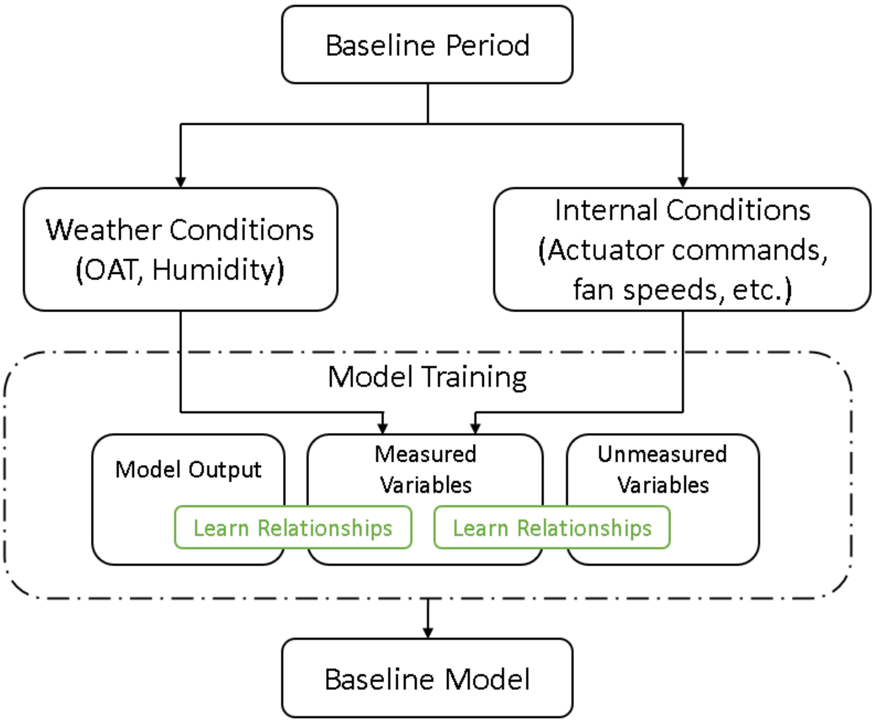


Figure 7. Training baseline regression model.

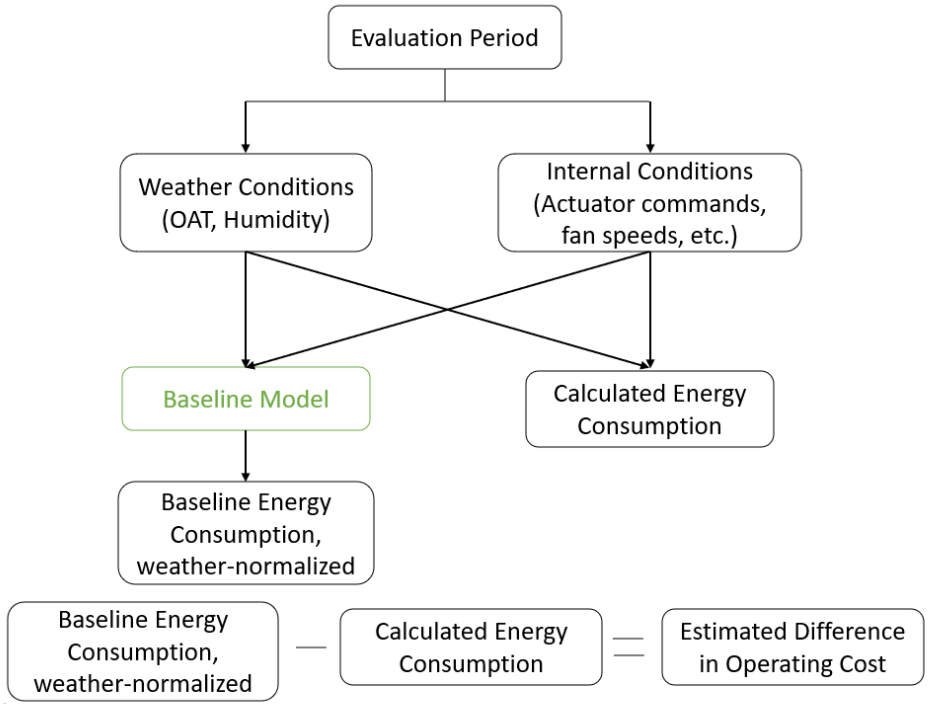

Evaluation Period
Weather Conditions (OAT, Humidity)
Internal Conditions (Actuator commands, fan speeds, etc.)
Baseline Model
Calculated Energy Consumption
Baseline Energy Consumption, weather-normalized
Baseline Energy Consumption, weather-normalized
—
Calculated Energy Consumption
=
Estimated Difference in Operating Cost

Figure 8. Evaluating data with the baseline model.

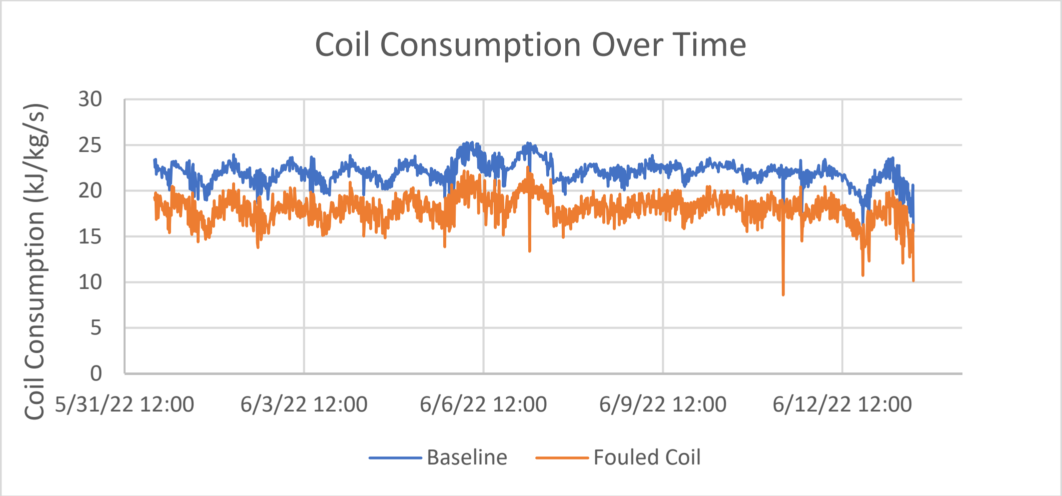


Figure 9. Consumption values in a cooling coil.

3. Performance Field Test

3.1. Detecting Faults

The previous sections described the procedures used to detect faults in building systems. The methods were developed to be generalized and applied to different buildings and configurations using categorized data in each building. This approach was used to address concerns found by Granderson that current commercial ML-based FDD is designed in a specialized way for individual buildings [1]. The following section describes the output from ML algorithms following the above procedures in an existing building.

3.2. Zachry Introduction

The Zachry Engineering Education Complex at Texas A&M University, shown in Figure 10, is a multipurpose building on the College Station campus, which opened for public use during the Fall semester in 2018. The construction reused the original building's concrete structure and expanded the area to a 525,000 square foot building containing classrooms, study rooms, restaurants, lobbies, and engineering laboratory spaces. Zachry is the largest building on campus and serves as the home of Texas A&M University's College of Engineering.

Figure 10. The Zachry building.

The building is conditioned by 20 air handling units and an additional 33 fan coil units. There are 468 terminal box units in the system. The entire building contains 6682 sensors trending unique data points. Building sensors have varying levels of reliability and can drift or fail completely. Though historically less reliable than temperature sensors, flowrate measurements may be used to calculate the total flow of energy through the system.

The Zachry building is open to the public from 7:30 AM until 10 PM on weekdays, 9:00 AM until 5:30 PM on Saturday, and 9:00 AM until 10:00 PM on Sunday. The HVAC equipment scheduling reflects these operational hours. Space temperatures in the building are controlled using unoccupied and occupied temperature setpoints which are triggered by occupancy sensors in each space. The ML models were trained to analyze space temperatures in the Zachry building using the procedures mentioned above. The models analyzed multiple zones to demonstrate the power of ML automated calibration using categorized data.

Abnormal Space Temperature

Abnormal temperatures are common in building systems and can impact occupant comfort and productivity. Space temperature data measured from the Zachry building is shown in Figure 11. The data shows several periods of high space temperature readings exceeding 80 degrees Fahrenheit in early evenings.

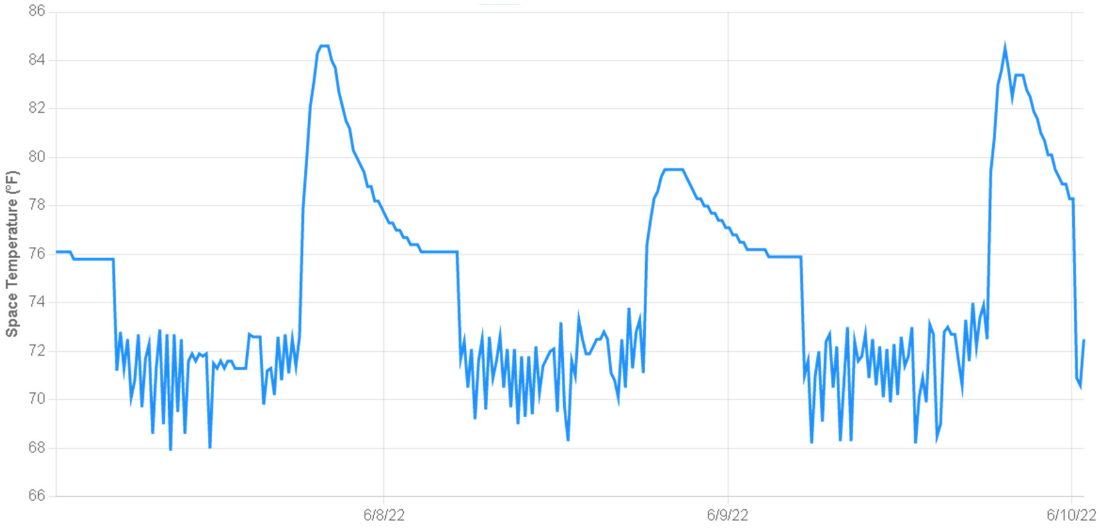


Figure 11. Space temperature data in a terminal box.

The ML algorithm detected abnormal space temperatures in the system using the Percent of Time metric. The detection rate was 55%; further analysis into the calculations illustrates the reliability of the model's output. Figure 12 visualizes the model's categorical output for each data point, where green cross points represent the baseline and red points represent abnormal space temperatures. Figure 12 shows that periods of abnormal behavior were detected during the night. When the predictions are inspected closely, abnormal space temperatures were detected consistently when the space temperature is above 74 degrees.

Further manual analysis into the space explains the high space temperatures. This classroom space faces northwest with a window to wall ratio of approximately 50–60%. Space temperatures rise in the zone during the evenings because the terminal box fans turn off before the sun has set. Solar energy is input to the room while the conditioning system is offline. The system is pre-cooled to 76 degrees Fahrenheit beginning at 4:00 am. When activity is sensed in this space, the setpoint is set to 72 degrees.

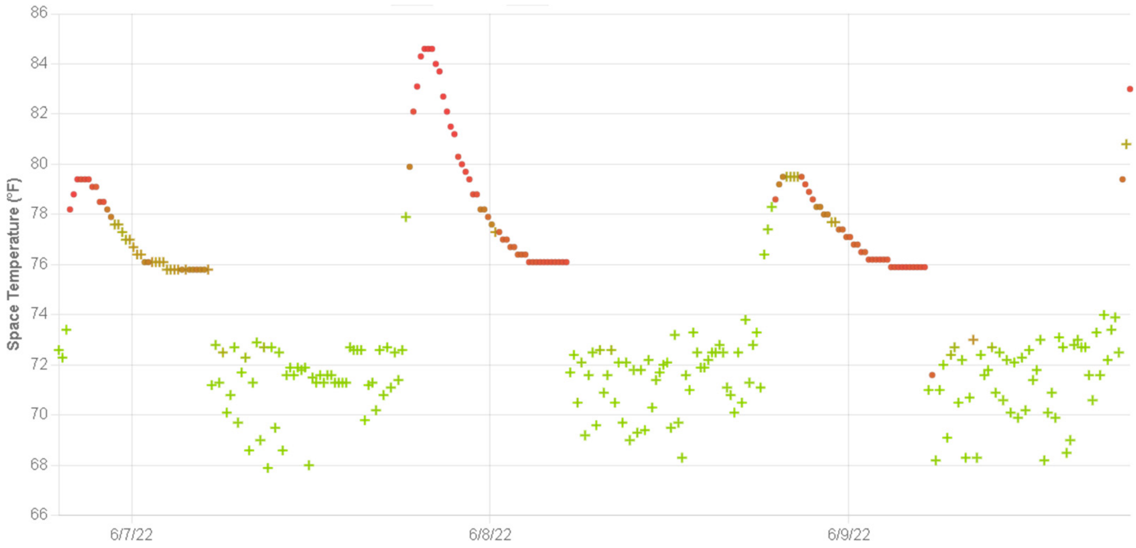


Figure 12. High space temperature detection in the evaluation period.

Figure 13 shows results from a different zone in the Zachry building using the same color scheme shown in Figure 12. Normal space temperatures in this zone are between 72- and 75-degrees Fahrenheit, which is approximately two degrees higher than the normal temperatures shown in Figure 12. Results in Figure 13 show abnormally low space temperature readings beginning at 71 degrees Fahrenheit, which is within the normal space temperatures defined in Figure 12.

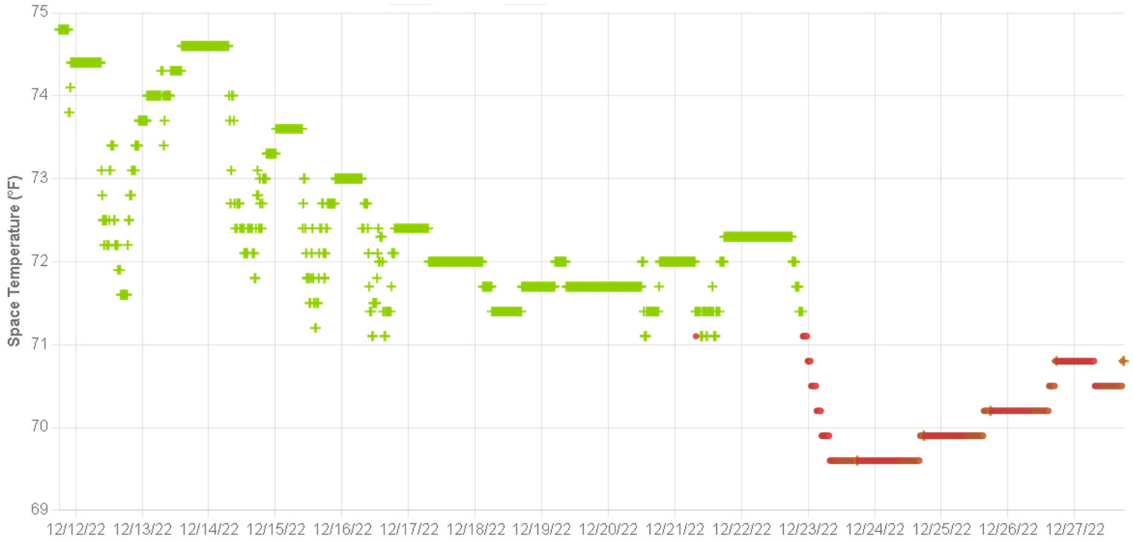


Figure 13. Fault detection in the evaluation period.

A strength of ML analysis is its ability to detect abnormal behavior relative to expectation. Figures 11 and 12 show results of a successful ML analysis but could be solved using a Boolean rule-based FDD analysis detecting space temperature values beyond a calibrated threshold value. ML's advantages become apparent when implementing a rule-based FDD system in an entire building, which may consist of hundreds of zones. Each zone requires a threshold calibrated to its normal temperature measurements, which is a manual process in rule-based FDD. ML models zones using a training process which can determine thresholds automatically.

Figure 14 illustrates normal space temperatures in 14 zones served by a single AHU in the Zachry building. Zones 1 and 2 are the zones in Figures 12 and 13, respectively. To determine each zone's normal temperature range, space temperatures were plotted for a 365-day period. The temperature range representing at least 75% of its operation during this period is defined as the zone's normal operational behavior.

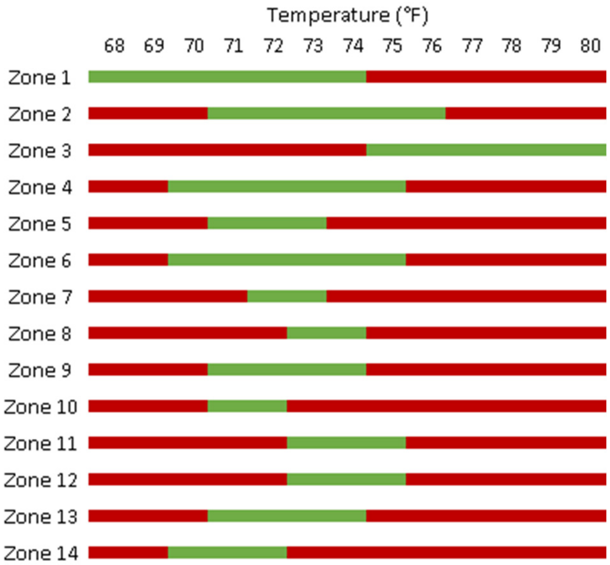


Figure 14. Normal space temperatures of zones in the Zachry building. Green and red regions represent normal and abnormal operating temperatures, respectively.

Rules are developed to fit either the union or intersection of normal operational behavior in zones. The union of normal operational behavior in Figure 14 is the range from 68 to 80 degrees Fahrenheit, defined by Zones 1 and 3. A rule defined in this way will misrepresent all regions of red in the figure. The intersection of normal behaviors in Figure 14 is an empty set.

As illustrated in Figure 14, unique rules should be defined for zones in the system to fully capture the normal operational behavior of each zone. Using multiple rules for the analysis can fully capture each zone's behavior but requires additional manual effort to configure. The ML in this project analyzes zones individually, which is equivalent to defining unique rules for each zone, by automatically determining appropriate thresholds in each zone. Extrapolating these results across hundreds of zones exemplifies the time savings when using ML.

As mentioned in Section 2.2, some fault severity metrics may communicate decreased consumption due to a fault. High space temperatures belong to this group because the hot zone is a symptom of under-conditioned air entering a space, where the lack of conditioning results in dollar and energy savings but reduced occupant comfort and a lower overall productivity in the space.

4. Conclusions

4.1. Future Work

Section 2.2.2 discussed ML's ability to automatically derive relationships between measured and unmeasured variables. These relationships can be used to automatically detect faults which require physical modeling and expert knowledge in rule-based FDD. Demonstrations of these benefits remain for future work.

A benefit of using ML to perform system analysis is its ability to automatically infer dependencies between measured and unmeasured variables. Faults related to those unmeasured variables, such as economizer controls failures or duct leakage, will be evaluated in future work.

4.2. Summary

The component analysis performed in this study was conducted using categorized data, which groups trends by their category. Building systems should implement standards for data access to reduce the preprocessing time for categorizing building data. These standards could eliminate the reliance on trend names for generalized datasets. The standards would also reduce manual processing required to implement ML projects. Where this project uses manual preprocessing to organize sensor measurements, a standardized system of data access can automate the process by organizing measurements as they are read from the BMS. Projects such as the Brick Schema and Project Haystack propose solutions toward this end but have yet to be widely adopted [23,24].

Preliminary case studies using the proposed procedure in Figure 1 show promising results. An automated system was used to determine installed system sensors and curate datasets to predict individual faults in the system. The algorithm was able to successfully detect abnormal space temperatures in multiple zones of the Zachry building without the manual effort required to calibrating the localized analysis.

A results format was developed which consolidates the complex results from ML analysis into a single intuitive metric. Scatter plots were used to visualize the severity of each violation throughout the evaluation period. An experimental fault severity metric was calculated for each fault to estimate the fault's impact on the system consumption.

Author Contributions: Investigation, W.N.; resources, W.N.; writing—original draft preparation, W.N.; writing—review and editing, C.C.; supervision, C.C.; project administration, C.C.; funding acquisition, C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded using internal funds from Texas A&M University's TEES Energy Systems Lab.

Data Availability Statement: Restrictions apply to the availability of these data. Data was obtained from Texas A&M Utilities and are available from the authors with the permission of Texas A&M.

Acknowledgments: Chris Dieckert from Texas A&M University provided the authors with access to the Texas A&M University data management system which was used to collect the data used in this project.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Granderson, J.; Lin, G.; Singla, R.; Mayhorn, E.; Ehrlich, P.; Vrabie, D. *Commercial Fault Detection and Diagnostics Tools: What They Offer, How They Differ, and What's Still Needed*; Lawrence Berkeley National Laboratory: Berkeley, CA, USA, 2018. [CrossRef]
2. House, J.M.; Vaezi-Nejad, H.; Whitcomb, J.M. An Expert Rule Set for Fault Detection in Air-Handling Units. *Ashrae Trans.* **2001**, *107*, 858.
3. Luskay, L.; Brambley, M.; Katipamula, S. *Methods for Automated AND Continuous Commissioning of Building Systems*; U.S. Department of Energy: Oak Ridge, TN, USA, 2003; pp. 6–8. [CrossRef]
4. Wang, H.; Chen, Y. A robust fault detection and diagnosis strategy for multiple faults of VAV air handling units. *Energy Build.* **2016**, *127*, 442–451. [CrossRef]
5. Zhao, Y.; Li, T.; Zhang, X.; Zhang, C. Artificial intelligence-based fault detection and diagnosis methods for building energy systems: Advantages, challenges and the future. *Renew. Sustain. Energy Rev.* **2019**, *109*, 85–101. [CrossRef]
6. Chen, J.; Zhang, L.; Li, Y.; Shi, Y.; Gao, X.; Hu, Y. A review of computing-based automated fault detection and diagnosis of heating, ventilation and air conditioning systems. *Renew. Sustain. Energy Rev.* **2022**, *161*, 112395. [CrossRef]
7. Bode, G.; Thul, S.; Baranski, M.; Müller, D. Real-world application of machine-learning-based fault detection trained with experimental data. *Energy* **2020**, *198*, 117323. [CrossRef]
8. Zhao, Y.; Wang, S.; Xiao, F. A statistical fault detection and diagnosis method for centrifugal chillers based on exponentially-weighted moving average control charts and support vector regression. *Appl. Therm. Eng.* **2013**, *51*, 560–572. [CrossRef]
9. Han, H.; Cao, Z.; Gu, B.; Ren, N. Pca-svm-based automated fault detection and diagnosis (afdd) for vapor-compression refrigeration systems. *HVAC R Res.* **2010**, *16*, 295–313. [CrossRef]
10. Tran, D.A.T.; Chen, Y.; Jiang, C. Comparative investigations on reference models for fault detection and diagnosis in centrifugal chiller systems. *Energy Build.* **2016**, *133*, 246–256. [CrossRef]

11. Jang, J.; Han, J.; Leigh, S.B. Prediction of heating energy consumption with operation pattern variables for non-residential buildings using LSTM networks. *Energy Build.* **2022**, *255*, 111647. [CrossRef]
12. Tidriri, K.; Chatti, N.; Verron, S.; Tiplica, T. Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: A review of researches and future challenges. *Annu. Rev. Control.* **2016**, *42*, 63–81. [CrossRef]
13. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Available online: https://github.com/Microsoft/LightGBM (accessed on 31 January 2023).
14. Hastie, T.T. *The Elements of Statistical Learning*, 2nd ed.; Springer: New York, NY, USA, 2017; Volume 27.
15. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. *arXiv* **2016**, arXiv:1603.02754. Available online: https://arxiv.org/abs/1603.02754 (accessed on 31 January 2023).
16. van Every, P.M.; Rodriguez, M.; Jones, C.B.; Mammoli, A.A.; Martínez-Ramón, M. Advanced detection of HVAC faults using unsupervised SVM novelty detection and Gaussian process models. *Energy Build.* **2017**, *149*, 216–224. [CrossRef]
17. Yan, K.; Shen, W.; Mulumba, T.; Afshari, A. ARX model based fault detection and diagnosis for chillers using support vector machines. *Energy Build.* **2014**, *81*, 287–295. [CrossRef]
18. Lee, W.Y.; House, J.M.; Kyong, N.H. Subsystem level fault diagnosis of a building's air-handling unit using general regression neural networks. *Appl. Energy* **2004**, *77*, 153–170. [CrossRef]
19. Karami, M.; Wang, L. Fault detection and diagnosis for nonlinear systems: A new adaptive Gaussian mixture modeling approach. *Energy Build.* **2018**, *166*, 477–488. [CrossRef]
20. Mulumba, T.; Afshari, A.; Yan, K.; Shen, W.; Norford, L.K. Robust model-based fault diagnosis for air handling units. *Energy Build.* **2015**, *86*, 698–707. [CrossRef]
21. Yan, K.; Ma, L.; Dai, Y.; Shen, W.; Ji, Z.; Xie, D. Cost-sensitive and sequential feature selection for chiller fault detection and diagnosis. *Int. J. Refrig.* **2018**, *86*, 401–409. [CrossRef]
22. Katipamula, S.; Brambley, M.R. Methods for Fault Detection, Diagnostics, and Prognostics for Building Systems—A Review, Part I. *HVACR Res.* **2005**, *11*, 3–25. [CrossRef]
23. Project Haystack. Available online: https://project-haystack.org/ (accessed on 6 December 2022).
24. Brick Schema. Available online: https://brickschema.org/ (accessed on 6 December 2022).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.