Proceedings of the 2021 Winter Simulation Conference
S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo and M. Loper, eds.

A WORKFLOW FOR DATA-DRIVEN FAULT DETECTION AND DIAGNOSIS IN BUILDINGS

Joseph Boi-Ukeme
Gabriel Wainer

Department of Systems and Computer Engineering
Carleton University
1125 Colonel By Drive
Ottawa, ON, K1S 5B6, CANADA

ABSTRACT

The evolution of smart buildings has been driven by technological advancement in building control systems, which made building systems more fragile and prone to faults. Buildings now generate an enormous amount of data, and the collection and analysis of such data is useful for detecting faults. However, fault detection approaches are not optimal in buildings due to several challenges. Data for fault detection is not readily available because of poor data collection practices and often when data is available, the data quality is inadequate to have useful models for fault detection and diagnosis (FDD). We propose a workflow for data-driven fault detection and diagnosis to deal with some of these challenges. The workflow incorporates a data collection framework and recommends the best data-driven modeling practices to improve data quality and model performance.

1 INTRODUCTION

Technological advancements in sensing and control systems have improved the quality of systems installed in buildings, which now can react to their environment, which ultimately increases their ease of use and efficiency (Garcia et al. 2017). The benefits of smart buildings come at a cost: the underlying systems that drive these innovations are complex and fragile, thereby being more prone to faults and failure (Viktoros et al. 2020). In our research we are interested in buildings equipped with sensors, actuators, and control systems; we will call these systems *building systems*.

The numerous advantages of control systems for buildings can only be enjoyed when building systems are reliable and perform as expected. However, building systems are affected by several conditions that can lead to faults, including physical damage, deterioration, inadequate maintenance, and poor quality of the materials used (Rajkumar et al. 2010). These conditions are more prominent when old buildings are retrofitted or when buildings are subject to harsh operating conditions like inclement weather and occupancy. These faults, if left unchecked, can lead to failure. Failure in certain components can have serious effects, for example, if the heating system of a building fails during the peak of winter.

During their lifetime, control systems for buildings generate significant amounts of data, including data obtained from building design, operation, and reports from technicians and building occupants. Irrespective of the source of the data, they can provide enormous amounts of information that, when analyzed properly, reflect the actual operation of the building (Lazarova-Molnar et al. 2016). This knowledge can also help us detect faults and diagnose faults through a process called Fault Detection and Diagnosis (FDD).

Although we can collect data from buildings, such data collected might not be good enough for FDD. Also in many cases, the FDD process is challenging even with adequate data. One of the reasons for this is the variation in building systems that ultimately affects the data collection strategy and data quality. We would generally have a different data strategy for old buildings, non-digital control systems, and multiple decentralized control systems when compared with modern-day buildings.

Although FDD is a well-established domain, data collection strategies are still a major concern. There are very few resources available in literature on how to properly collect data for FDD in buildings. Data

978-1-6654-3311-2/21/$31.00 ©2021 IEEE

collection for fault detection suffers from poor planning, passive data collection, improper fault validation, and poor metadata modeling. This leads to poor data quality which ultimately affects the accuracy and validity of any data-driven FDD strategy employed (Lazarova-Molnar et al. 2016). Also, there is a lot of potential for the improvement of the data-driven modeling process by incorporating simulation in the workflow. Simulation can help us deal with some of the challenges in data collection and modeling discussed above.

Considering these facts, we introduce a workflow for Data-Driven FDD in buildings that considers the different aspects of data collection, incorporates simulation and best practices for building models for FDD in smart buildings. The workflow is developed by using data from real buildings and then defined in a generic fashion such that it can be applied to similar systems.

2 BACKGROUND AND RELATED WORK

Faults are abnormal conditions that could cause errors that could result in failure. Faults could be caused by internal or external conditions within the building system, or, for instance, by extreme environmental conditions which are external to the building system in question. If the faults occur without causing errors, they are called *passive faults* and if they lead to errors, they are *active faults* (Shu et al. 2016; Angelopoulos et al. 2020). There have been numerous efforts to classify the types of faults that affect building systems with different authors coming up with different ways of classifying faults. However, for the purpose of this work, since major aspects of building systems include sensing, actuating, and control, we will consider faults in these subsystems.

Faults in *sensors* occur for several reasons, including aging, battery exhaustion, physical damage, and noise (Raposo et al. 2017). These faults can affect performance within a sensor network. Studies have reported multiple temporary failures in sensor readings within the sensor network (Sharma et al. 2010). Sensor faults typically prevent us from properly capturing changes in physical processes.

Actuator faults are important because they can cause the entire system to fail. They are also difficult to deal with because it is hard to have actuator redundancy. A modern actuator is a dynamic system by itself and faults within actuators can be classified as like a dynamic system (Zhan et al. 2018 and Bartys et al. 2006). There are different types of actuators: **hydraulic**, in which hydraulic power from a liquid is used to facilitate the mechanical process of switching, **pneumatic**, which are like the hydraulic but using compressed air in place of the liquid, **electrical**, which convert electric power to mechanical torque or light, and **mechanical** actuators, which convert rotating motion to linear motion. These actuator types have different parts, and the actuator faults can be classified according to which part is responsible for the fault (Ponomareva et. al. 2006, Cura et al. 2003).

The behavior within a control system is affected by the design as well as the logical results and time of execution. For a building system, the correct service for a control system means that the design parameters are correct, and the behavior is correct, meaning it has expected logical results at the expected physical time of execution. Any deviation from a correct service is a fault (Lee 2015). Control system faults can occur at different phases during the life of the control systems system; they could occur at the development phase due to software flaws, hardware errors, or production defects. They could also occur during the operation phase as physical faults or faults because of the interaction of the building with the environment (any entity external to the building) (Aivizienis et al. 2004)

When a fault occurs, the optimum operation of the building system cannot be guaranteed. There is usually a degradation in performance even if it does not interfere with the system operation. Therefore, it is important to detect faults quickly to prevent failure. FDD methods are developed to detect faults, identify their cause, severity, and consequences. Over the last decade, there has been a lot of work done in FDD (Lazarova-Molnar et. al. 2016)

FDD in building systems is a particularly challenging problem for several reasons. First, many of the processes within buildings are not adequately monitored. Second, there is significant fault propagation within building systems that complicate the diagnosis aspect of FDD. Finally, buildings provide a wide range of functions and the systems are subject to various configurations and environmental conditions. This typically leads to large variations that further complicates the FDD process (Reppa et al. 2016)

To address some of the challenges outlined above, several researchers have come up with different methods for FDD in buildings, which can be classified as *Model-Based*, *Hardware-Based*, or *Data-driven* methods within the building system (Boi-Ukeme et al. 2020).

Model-based FDD methods are designed to use a model to predict the output of a fault-free system and then these outputs are constantly compared with the output of the system. Whenever there is a significant deviation in the output, a fault condition is raised. Hardware-based methods use dedicated hardware within the building system to detect faults. Data-driven methods rely on data obtained from the building system to detect faults. This is achieved by using algorithms to benchmark normal system performance and find discrepancies in the data when compared to the benchmark (Zhao et al. 2020).

Our research focuses on data-driven methods, and we will introduce a workflow for these methods. Data-Driven FDD methods apply various approaches to study patterns inherent in the data to detect faults. They can be either qualitative, (for example, fuzzy logic or pattern recognition), or quantitative (for example, statistical methods or neural networks). When compared to other methods, Data-driven FDD methods are easier to apply in practice because they neither require significant knowledge of the systems nor the use of physical models. They simply rely on studying the relationship between input data (predictors) and output data to detect faults (Zhao et al. 2020; and Li and Brown 2007). Data-driven methods use different data mining methods, which can be classified as supervised or unsupervised. Supervised data mining methods aim to learn complex relationships between multiple variables. Generally, they are used for predictive or classification tasks. Unsupervised data mining methods are used to detect structure, relationships, and connections. It can discover hidden patterns within the data. Supervised methods are accurate but difficult to explain while unsupervised methods do not require ground truth information but are overly sensitive to data. For building systems, unsupervised methods discover an enormous amount of knowledge within the data most of which is not useful for FDD. Also, when compared with the supervised methods, the unsupervised methods are not good at finding complex relationships between multiple features. This makes the supervised methods more appropriate for our research: building systems have multiple related variables (Mimaghi and Haghighat 2020).

Data-driven FDD in buildings has been studied in the past decade; while the work done addresses relevant issues regarding FDD, there is still room for improvement. Most of the work in this area focuses on applying different data mining approaches to specific building systems, and many of them focus on building systems with significant energy consumption (Zhao et al. 2020). Although the studies so far have been useful for the improvement of FDD methods, there is no generic workflow to guide the process of FDD in building systems. Also, there are many methods for Data-Driven FDD such that the choice of the best method is not always obvious. Not all building systems benefit from a data-driven approach, thus a workflow could help to identify candidates for FDD and guide the selection of methods. Finally, there is little work done regarding data collection for FDD, and it is clear from different researchers that the quality of the FDD methods greatly depends on the quality of the data used for FDD. For example, the data of fault ground truth always yields an unbalanced dataset with the fault-free data. Simulation data can significantly help improve the data quality for FDD (Zhao et. al 2020; and Lazarova-Molnar et al. 2016)

3 A WORKFLOW FOR DATA-DRIVEN FDD

The workflow presented in this section was developed by applying standard data mining methods to datasets obtained from a building system, the datasets studied include simulated data, experimental data, and actual data from buildings. In applying these methods, we observed certain gaps and peculiarities for FDD that are not captured in the standard data mining process, hence this workflow is proposed to deal with these gaps. The proposed workflow is presented in Figure 1.

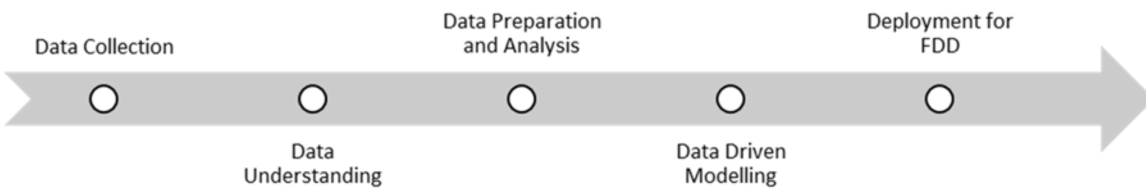

Figure 1: Data-driven FDD workflow.

The workflow shows a five-step process for Data-driven FDD. The steps include data collection from the building system of interest, data understanding process, data preparation and analysis, data modeling, and deployment of the models for FDD. The direction of the arrow shows the typical sequence of these steps, although in some cases you can go back to a previous step to make changes that impact the current step. We start by collecting data from the building that meet the requirements for FDD, then we apply some techniques to understand the data. After data understanding is completed, we would proceed to data preparation and Analysis which would help us prepare the data set for FDD and guide us with the selection of the right modeling approach. The next step in the process is the development of the data-driven models after which the models can be deployed to detect faults in real buildings. In this workflow, we are not concerned with deployment because the deployment strategy will depend on the specific building system use case.

At first glance, this workflow looks like a standard data mining workflow. We will now describe each of the components within the workflow, highlighting the steps needed for proper FDD in buildings.

3.1 Data Collection

As we discussed in the background, there is a drive to collect data from buildings, and the data collected can be useful for FDD. For this data to be good enough for FDD, we have developed an approach and a set of requirements that the data must meet. The way the data is collected will depend on the building and the type of information available. The data collection framework gives a summary of how to collect data.

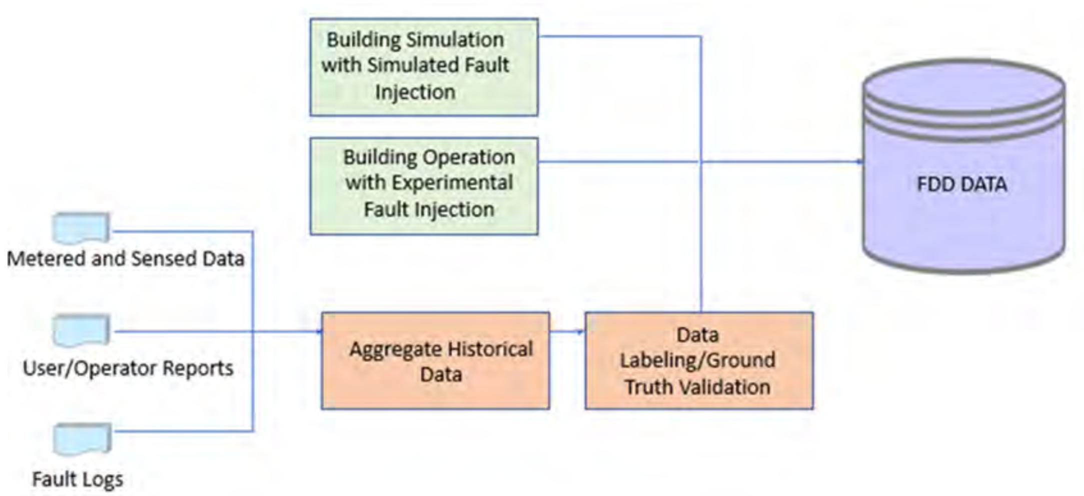

Figure 2: Data collection framework.

From Figure 2 we can see that the data can be from a simulation, experiment, or the actual operation of the building. The idea that smart buildings are designed to be efficient makes it difficult to collect many data points for faults during operation. In general, building operation data does not have enough data points to show the range of faulty conditions, this is because the building systems are designed to be reliable and sometimes have not been operated for long enough periods to see the various fault types and intensities. Simulation is important regardless of whether or not we have operational data because it helps us generate the required FDD data or compensate for the inadequacies in the data collected from the other sources. There are many tools available with standard methods that can help us simulate these conditions and generate adequate data for developing FDD models. This shows how data from simulation and experiments can help. From simulation and experimentation, we can force the system or model of the system to work in different conditions and collect fault data which will be useful in developing data-driven models. In an ideal situation, we would have models of the system, and physical models where we can run experiments to collect data for FDD, however, this is not always the case. It may be an old building that has been retrofitted with smart systems. In this case, we would need to aggregate the data from different sources including metered and sensed data, user/operator reports, fault logs, and other available methods. Once the data is aggregated, there would need to be a detailed process for data labeling and ground truth validation, this is a tedious process but is needed to be sure the data obtained meets the requirement for FDD. Regardless of how the data is collected and what means are available, FDD data should meet the following requirements:

1. **Data Description:** The data collected should be clearly described. Proper data description would ensure that historical data can be used for future FDD strategies in the building. Elements that make for proper description include:
    a. Information on how the data was collected.
    b. What information we can obtain from the data.
    c. Metadata to describe what data represents, the correct unit of measure, and the time stamp.
2. **Data Collection Frequency and Duration:** The data for FDD should be collected frequently enough to ensure faults are not missed and long enough to ensure we capture the operating range of the building system. For example, for temperature data in a building, we would need to ensure we have measurements during the different seasons in the year.
3. **Datapoints** should obey standard rules (for example, the number of data points should be at least ten times the number of features). Also, the features present in the dataset should be relevant to the fault to be investigated.
4. **Fault Input scenarios** should be clearly defined with types and time. This includes fault types, fault intensities, fault injection mechanisms, measurement frequency, and fault duration.
5. **Target:** The ground truth information between fault free and faulted data should be known

3.2 Data Understanding

After the FDD data that meets the requirements have been collected, the next step in the workflow is data understanding. The objective of this step is to understand the attributes of the data and clearly identify key features that would help us make informed decisions regarding data-driven modeling for FDD. This step is important to avoid unexpected problems in the next phase (data pre-processing and analysis), which is usually the most time-consuming step in the workflow.

For proper data understanding, several questions need to be answered about the datasets available. The answers to the questions of the data understanding step would help us understand the data from its description, identify the features and target, decide whether the data is relevant for FDD and if there are any ethical concerns or other similar issues with developing the models. At the end of this stage, we can decide whether it makes sense to adopt a data-driven approach.

3.3 Data Pre-processing and Analysis

After understanding the data, the next step is pre-processing and analysis. In this step, we start with pre-processing where we would ensure that the raw data is cleaned and ready for modeling. Then we would proceed to analysis where we can explore the different variables in our data. We would be able to discover missing, noisy, or inconsistent data. We would also find the distribution of each variable and how they relate to each other. Depending on the way the dataset is collected, there are several activities to be done at this stage.

This exploratory analysis is particularly important for buildings because the way building systems are designed, the various features are closely related. For example, the temperature in the room is related to the speed of the fan in the air handling unit and many other features. By doing the univariate and bivariate analysis, we can see what the features of interest are and how they relate or correlate with each other.

The knowledge obtained from this step would guide our decision on what types of models to use in the modeling stage.

3.4 Data-Driven Modeling

After the data pre-processing and analysis are done, we would have an idea of the categories of models that would be good for the data set. The choice of an exact model would be decided iteratively, meaning, we would try different models within the category and select the best one based on certain metrics. The following steps summarize the activities in the modeling phase.

1. Specify the target and the inputs (predictors)
2. Partition the data into the training and testing set. The training set is a subset to train the model while the test set is a subset to test the trained model.
3. Correct for Imbalance: As we saw from the discussions in the preceding section, FDD data is usually unbalanced, and we would correct that. This can be achieved by collecting more data, resampling the dataset, or generating synthetic samples through simulation. Generating synthetic samples through simulation is gradually becoming popular. For example, there has been some work done in generating synthetic samples using a discrete event simulator called SimPy (Kenny et al. 2021). It can also be corrected in the modeling stage by using a penalized model or by evaluating model performance using a different metric other than accuracy. This is because unbalanced datasets usually affect model accuracy.
4. Choose the model(s) iteratively and according to the FDD objective. For example, for a flag target, we would need a classification model.
5. Evaluate models based on certain metrics and select the best performing model. Some metrics include accuracy of fault detection, confusion matrix (Detection of false positives or false negatives).

4 A WORKFLOW FOR DATA-DRIVEN FDD – CASE STUDY

The case study presented in this section uses datasets obtained for the Air Handling Unit (AHU) of a building. AHUs are systems that are used to circulate air within the building. It is used to supply and transmit fresh air at the correct temperature, this is achieved through temperature control, oxygen regulation, air conditioning, and humidity regulation. They are usually found underground on the roof or the floor of a building. Most units have a duct that pulls used air from the room to the AHU, which a fan releases into the atmosphere (Afroz et al. 2018).

4.1 Data Collection

The dataset was created by a team in the Pacific Northwest National Laboratory (PNNL) using a building model. The building envelope model was simulated using the EnergyPlus tool (Crawley et al 2001) while the Heating Ventilation and Cooling (HVAC) system model was simulated using Dymola (Bruck et al 2002). The facility is equipped with three AHUs. AHU-1 serves the common areas of the building. The remaining AHUs serve the A-and B-Test Systems. AHU-A and B are identical, with each AHU serving four zones. Of the four zones, three have external exposures and one sees only internal conditions. The A and B zones are mirror images. The zones are identical in terms of construction, exposure, and thermal load. The focus for the experiments and simulation is on AHU-A (Lin et al 2020). The dataset provides information about the main components of the AHU-A test system. They include:

- Air supply fan and the return air fan,
- Preheating, cooling, and heating coils,
- Heating and cooling control valves.
- Recirculating air, exhaust, and outside air (OA) dampers.
- A duct for transferring air to and from an air-conditioned space.
- There is a target field indicating a fault in the OA temperature bias.

The dataset fulfills the requirements as follows:

1. For metadata, every column is properly labeled, and the unit of measure is clearly defined in the data description. There is a date-time column showing the time stamp for each record.
2. The fault of interest is OA temperature bias, and the data was collected every two (2) minutes for 1 year, on inspection, the data appears to cover the operating range of the systems involved.
3. We have 272160 records and 18 columns which fulfills the requirement for the number of fixtures and records. Dataset also has relevant features and properly defined units of measure.
4. The input scenario for the fault injection process was described in the data collection document. Faults were injected and left to last for 12 hours. The fault intensity is described in Figure 3.

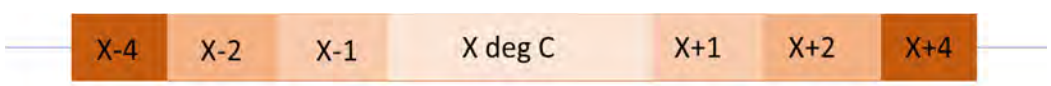

Figure 3: Fault intensities.

   The middle value in Figure 3 (X deg C) is the correct temperature reading. The boxes to the right and left of it represent deviations from the correct reading and are considered as faulty values. OA temperature bias faults start with a 1 deg C difference in the measured value. There are also higher bias values with a difference of 2 deg C and 4 deg C, as shown in Figure 3. The fault values vary in intensity. The higher the deviation from the correct reading the higher the fault intensity. In this example, there are three levels of fault intensity considered. They are deviations of 1, 2, and 4 deg C respectively with 1 deg C having the least intensity and 4 deg C having the most intensity.
5. Whenever there was a temperature bias, the target data was defined as 1 and when there was no fault it was defined as 0. The target ground truth is known, and the target is a digital value.

4.2 Data Understanding

To understand the data, we analyzed the various columns for their data type and statistics. Table 1. Summarizes the dataset.

From Table 1, we can see that the dataset has 18 columns and 272160 records. The data consists of some continuous measurements and some flag measurements. During the data understanding process, we observed that some measurements were not properly categorized. This was noted and would be addressed in modeling.

Table 1: Dataset summary.

Field	Sample Graph	Measurement	Min	Max	Mean	Std. Dev	Skewness	Unique	Valid
AHU: Supply Air Temperature		Continuous	44.950	72.030	53.762	2.546	2.352	--	272160
AHU: Supply Air Temperature Set Point		Continuous	55.040	55.040	55.040	0.000	--	--	272160
AHU: Outdoor Air Temperature		Continuous	-14.260	90.140	48.358	20.154	-0.712	--	272160
AHU: Mixed Air Temperature		Continuous	22.610	78.720	65.000	7.410	-0.132	--	272160
AHU: Return Air Temperature		Continuous	54.470	87.870	71.454	3.702	-0.681	--	272160
AHU: Supply Air Fan Status		Flag	0	1	--	--	--	2	272160
AHU: Return Air Fan Status		Flag	0	1	--	--	--	2	272160
AHU: Supply Air Fan Speed Control Signal		Flag	0	1	--	--	--	2	272160
AHU: Return Air Fan Speed Control Signal		Flag	0	1	--	--	--	2	272160
AHU: Outdoor Air Damper Control Signal		Flag	0	1	--	--	--	2	272160
AHU: Return Air Damper Control Signal		Flag	0	2	--	--	--	2	272160
AHU: Cooling Coil Valve Control Signal		Flag	0	0	--	--	--	1	272160
AHU: Heating Coil Valve Control Signal		Flag	0	1	--	--	--	2	272160
AHU: Supply Air Duct Static Pressure Set Point		Flag	0.040	0.040	--	--	--	1	272160
AHU: Supply Air Duct Static Pressure		Flag	0	0	--	--	--	1	272160
AHU: Supply Air Duct Static Pressure		Flag	0	0	--	--	--	1	272160
Occupancy Mode Indicator		Flag	0	1	--	--	--	2	272160
Fault Detection Ground Truth		Flag	0	1	--	--	--	2	272160

4.3 Data Pre-processing and Analysis

The data provided did not require much cleaning. There were no missing values or duplicated entries. The only data cleaning practice that was applied was to convert the Datetime column data type from string to DateTime datatype and then to ensure uniformity in the column names.

4.3.1 Univariate Analysis

Histogram plots, bar graphs, measures of central tendency, and dispersion were used to perform univariate analysis column by column on the AHU dataset. It was observed that there is an imbalance in the dataset since class 1 (faults detected) exceeds class 0 (no faults) by a large number as shown in Figure 4. This needs to be dealt with in the modeling stage, otherwise, the models will not perform well.

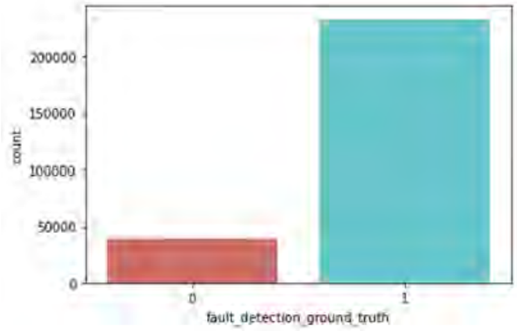

Figure 4: Fault ground truth.

Non-categorical columns which consisted of temperature data were observed to have higher variances while categorical columns have lower variances. The constant set values have the least variances since these values remain all through each data point.

4.3.2 Bivariate Analysis

Here, the correlation coefficients were used to observe any linear correlations that existed within the variables in the data. The Variance Inflation Factor (VIF) scores were used to detect this multicollinearity. VIF is the inverse of the correlations. High VIF readings (above 5) shows there is high multicollinearity. Table 2. Shows the VIF scores for the variables.

Table 2: Feature correlation.

Variables	VIF
Supply Air Temperature	1.425416
Supply Air Temperature Setpoint	0.862552
Outdoor Air Temperature	3.735444
Mixed Air Temperature	8.014675
Return Air Temperature	3.460612
Supply Air Fan Status	inf
Return Air Fan Status	inf
Supply Air Fan Speed Control Signal	inf
Return Air Fan Speed Control Signal	inf
Outdoor Air Damper Control Signal	4.357522
Return Air Damper Control Signal	1.000012
Cooling Valve Control Signal	1.364601
Heating Coil Valve Control Signal	1.005016
Supply Air Duct Static Pressure Setpoint	0.014861
Supply Air Duct Static Pressure	45.70911
Occupancy Mode Indicator	5.975188

From Table 2, there are notable variables with VIF scores above 5. This shows that there was multicollinearity in the data variables. With this type of multicollinearity, we have two options: we can either remove some features that are highly correlated and then still try a regression approach or treat it as a classification problem where the multicollinearity would not be a cause for concern. We would try both models and compare the results.

4.4 Data-Driven Modeling

From the data analysis and pre-processing, we observed that the FDD problem for this case study is a classification problem. For this example, we trained two models to detect faults. The models are Logistic Regression (LR) and K- Nearest Neighbors (KNN). The workflow shown in Figure 5 describes the modeling process.

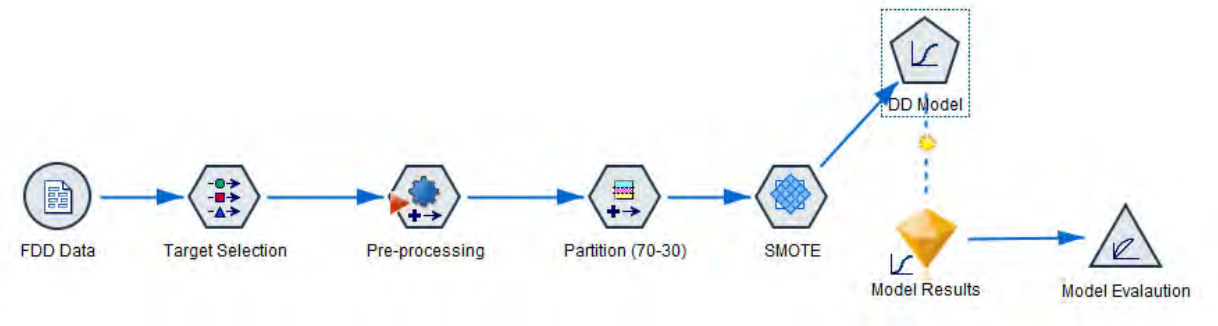

Figure 5: Modeling workflow.

From Figure 5, we start by importing the FDD data that meets the requirement defined in the data collection step, then we select the target column (fault ground truth); the next step is to clean the data in the pre-processing node. To partition the data, we used the 70-30 rule where 70% of the data was used to train the model while 30% of the data was used to test the models. The next step was to correct for imbalance. To do this, we use Synthetic Minority Oversampling Technique (SMOTE) to balance the data. The models were then trained iteratively using Scikit-learn which is an open-source python library. The results were collected, and the model was evaluated using the test data and the results are presented in section 5.

5 DISCUSSION OF RESULTS

In this section, we present and compare the results obtained from both models. In Figure 6, we checked the feature importance towards detecting the target.

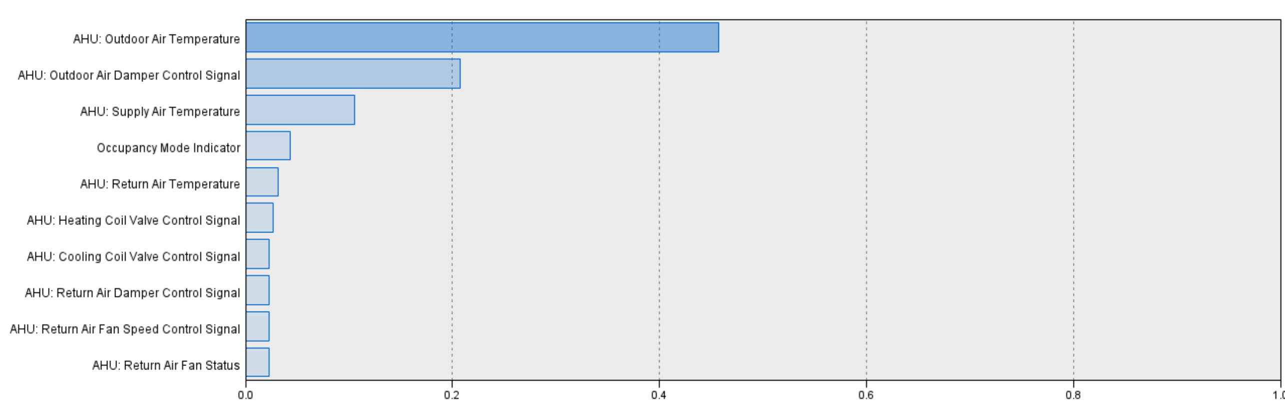

Figure 6: Feature importance.

We observed that the top three features that contribute towards predicting the target are OA temperature, OA damper control signal, and the AHU supply air temperature. The result for the performance of the two models is presented in Table 3 and Table 4.

Table 3: Logistic regression results.

Partition	Testing	
Correct	70,155	85.77%
Wrong	11,639	14.23%

Table 3 shows the model performance for the LR model. We evaluated the logistic regression model using the test data. The model predicted the test data with an accuracy of 85.77%. This was the best result after features that exhibited high multicollinearity were dropped as predictors. Similarly, we evaluated the KNN model using the test data and the results are shown in Table 4.

Table 4: KNN model results.

Partition	Testing	
Correct	78,996	96.58%
Wrong	2,798	3.42%

The model predicted the test data with an accuracy of 96.58%. The KNN model outperformed the LR model.

6 CONCLUSION AND FUTURE WORK

In this paper we presented a workflow for data-driven FDD in building systems, we went further to show the usability of the workflow by applying it to a simplified case study.

At first glance, the workflow would appear like a typical data mining workflow. While this workflow emanates from the standard, we ensured that we highlighted the important aspects of the data mining process that need to be paid attention to for FDD in buildings. FDD in a building should not be a random event, it should be carefully planned from the building design to its operation. The data collection strategy should start early and be done properly to ensure that the data collected is useful for FDD. The importance of simulated data and simulated fault injection was also highlighted in the data collection process. In the process of developing the workflow, it was observed that collecting ground truth information can be tedious, but it is important for FDD, which buttresses the point that consistency in modeling, simulation, and deployment is important in the data collection process. Fault detection data also suffers from a huge imbalance between faulty data and fault-free data, and it is important to correct this during data-driven modeling. Simulation is particularly useful for handling imbalance through the use of synthetic data.

Also, modeling decisions should not be random, they should be decided based on evidence obtained from studying the data. Detailed Data Understanding and Exploration can provide useful insight and guide the selection of data-driven models for FDD. For example, in this case, study, the target is a label (digital) therefore it is a classification problem it was also observed that a regression model would perform poorly because of multicollinearity in the features.

In our future work, we would apply the workflow to field data and other fault scenarios to evaluate the usability of the workflow. This will help us find areas of potential improvement in the workflow. Also, we would study and compare models developed with field data with those developed with data that include simulation data. This would help us find strategies to combine simulation data and field data within the same FDD workflow.

REFERENCES

Afroz, Z., Shafiullah, G.M., Urmee, T. and Higgins, G., 2018. "Modeling techniques used in building HVAC control systems: A review." *Renewable and sustainable energy reviews*, 83:64-84.

Angelopoulos, A., Michailidis, E.T., Nomikos, N., Trakadas, P., Hatziefremidis, A., Voliotis, S. and Zahariadis, T., 2020. "Tackling faults in the industry 4.0 era—a survey of machine-learning solutions and key aspects." *Sensors,* 20(1):109.

Avizienis, A., Laprie, J.C., Randell, B. and Landwehr, C., 2004. "Basic concepts and taxonomy of dependable and secure computing." *IEEE transactions on dependable and secure computing*, 1(1):11-33.

Bartyś, M., Patton, R., Syfert, M., de las Heras, S. and Quevedo, J., 2006. "Introduction to the DAMADICS actuator FDI benchmark study." *Control engineering practice*, 14(6):577-596.

Boi-Ukeme, J., Ruiz-Martin, C. and Wainer, G., 2020, December. "Real-Time Fault Detection and Diagnosis of CPS Faults in DEVS". *In 2020 IEEE 6th International Conference on Dependability in Sensor, Cloud and Big Data Systems and Application (DependSys)*, December 14th -16th, Fiji, Fiji Islands, 57-64.

Brück, D., Elmqvist, H., Mattsson, S.E. and Olsson, H., 2002, "March. Dymola for multi-engineering modeling and simulation". In *Proceedings of Modelica*, March 18th -19th, Oberpfaffenhofen, Germany, 551-558.

Crawley, D.B., Lawrie, L.K., Winkelmann, F.C., Buhl, W.F., Huang, Y.J., Pedersen, C.O., Strand, R.K., Liesen, R.J., Fisher, D.E., Witte, M.J. and Glazer, J., 2001. EnergyPlus: creating a new-generation building energy simulation program. *Energy and Buildings*, *33*(4):319-331.

Cura, V.O.D., Cunha, F.L., Aguiar, M.L. and Cliquet Jr, A., 2003. "Study of the different types of actuators and mechanisms for upper limb prostheses." *Artificial organs*, 27(6):507-516.

García, Ó., Alonso, R.S., Prieto, J. and Corchado, J.M., 2017. "Energy efficiency in public buildings through context-aware social computing". *Sensors*, 17(4):826.

Kenny, E., Hassanzadeh, H., Khanna, S., Boyle, J., and Louise, S., 2021. "Patient Flow Simulation Using Historically Informed Synthetic Data". In *Healthier Lives, Digitally Enabled*, edited by M. Merolli, C. Bain and L.K. Schaper, 32-37. Amsterdam: IOS Press.

Lazarova-Molnar, S. and Mohamed, N., 2016. "Challenges in the Data Collection for Diagnostics of Smart Buildings". In *Information Science and Applications (ICISA)*,February 15th -18th, Ho Chi Minh, Vietnam, 941-951.

Lazarova-Molnar, S., Shaker, H.R. and Mohamed, N., 2016, March. "Fault detection and diagnosis for smart buildings: State of the art, trends, and challenges". In *2016 3rd MEC International Conference on Big Data and Smart City,* March 15th -16th, Muscat, Oman, 1-7.

Lee, E., 2015. "The past, present, and future of cyber-physical systems: A focus on models". *Sensors*, 15(3):4837-4869.

Li, H., and Braun, J.E., 2007. "A methodology for diagnosing multiple simultaneous faults in vapor-compression air conditioners". *HVAC&R Research*, 13(2):369-395.

Lin, G., Kramer, H. and Granderson, J., 2020. "Building fault detection and diagnostics: Achieved savings, and methods to evaluate algorithm performance". *Building and Environment*, 168:106505.

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

Ponomareva, E., 2006. "Hydraulic and pneumatic actuators and their application areas". *Mechatronics-Foundations and Applications,* 12-19.

Rajkumar, R., Lee, I., Sha, L. and Stankovic, J., 2010. "Cyber-physical systems: the next computing revolution" In *Design automation conference,* June 13th -18th, Anaheim California, 731-736.

Raposo, D., Rodrigues, A., Silva, J. S., & Boavida, F. 2017. "A taxonomy of faults for wireless sensor networks". *Journal of Network and Systems Management*, 25(3), 591-611.

Reppa, V., Polycarpou, M.M. and Panayiotou, C.G., 2016. "Sensor fault diagnosis". *Foundations and Trends in Systems and Control*, 3 (1-2):1-248.

Sharma, A.B., Golubchik, L., and Govindan, R., 2010. "Sensor faults: Detection methods and prevalence in real-world datasets". *ACM Transactions on Sensor Networks,* 6(3):23.

Shu, Y., Ming, L., Cheng, F., Zhang, Z., and Zhao, J., 2016. "Abnormal situation management: Challenges and opportunities in the big data era". *Computers & Chemical Engineering,* 91:104-113.

Viktoros, S.A., Michael, M.K. and Polycarpou, M.M., 2020, August. "Compact Fault Dictionaries for Efficient Sensor Fault Diagnosis in IoT-enabled control systems". In *2020 IEEE International Conference on Smart Internet of Things,* August 14th -16th, Beijing, China, 236-243.

Zhang, M., Li, Z.T., Dahhou, B. and Cabassud, M., 2018. "Root Cause Analysis of Actuator Fault". In *Actuators,* edited by C. Volosencu, 131, Rijeka: IntechOpen.

Zhao, Y., Zhang, C., Zhang, Y., Wang, Z., and Li, J., 2020. "A review of data mining technologies in building energy systems: Load prediction, pattern identification, fault detection, and diagnosis". *Energy and Built Environment,* 1(2):149-164.

AUTHOR BIOGRAPHIES

JOSEPH BOI-UKEME is pursuing a Ph.D. in Electrical and Computer Engineering at Carleton University where he researches on Cyber-physical Systems. His email address is joseph.boiukeme@carleton.ca.

GABRIEL WAINER is a Professor at the Department of Systems and Computer Engineering at Carleton University. He is a Fellow of the Society for Modeling and Simulation International (SCS). His email address is gwainer@sce.carleton.ca.