Hybrid Model-based and Data-driven Fault Detection and Diagnostics for Commercial Buildings

Stephen Frank, Michael Heaney, Xin Jin, and Joseph Robertson,
National Renewable Energy Laboratory
Howard Cheung, Hong Kong Polytechnic University
Ryan Elmore, University of Denver
Gregor Henze, University of Colorado

ABSTRACT

Commercial buildings often experience faults that produce undesirable behavior in building systems. Building faults waste energy, decrease occupants' comfort, and increase operating costs. Automated fault detection and diagnosis (FDD) tools for buildings help building owners discover and identify the root causes of faults in building systems, equipment, and controls. Proper implementation of FDD has the potential to simultaneously improve comfort, reduce energy use, and narrow the gap between actual and optimal building performance. However, conventional rule-based FDD requires expensive instrumentation and valuable engineering labor, which limit deployment opportunities. This paper presents a hybrid, automated FDD approach that combines building energy models and statistical learning tools to detect and diagnose faults noninvasively, using minimal sensors, with little customization. We compare and contrast the performance of several hybrid FDD algorithms for a small security building. Our results indicate that the algorithms can detect and diagnose several common faults, but more work is required to reduce false positive rates and improve diagnosis accuracy.

Introduction

Commercial buildings often experience faults: improper operation of equipment and controls that produce undesirable behavior in building systems. Building faults waste energy, decrease occupants' comfort, and increase operating costs. Faults in buildings are pervasive and excessive; therefore, detection and correction of building faults represent an enormous opportunity for energy savings. In the United States alone, faults waste an estimated 0.3 to 1.8 quadrillion British thermal units (quads) (88 to 528 terawatt-hours [TWh]) of primary energy, or between 2% and 11% of total commercial building sector energy consumption (Roth et al. 2005).

Fault detection and diagnosis (FDD) is the process of detecting operational faults and identifying their root causes. Automated fault detection and diagnosis (AFDD) tools perform FDD in software with minimal human intervention. Properly implemented, AFDD enables building operators to simultaneously improve comfort and reduce energy use, closing the gap between actual and optimal building performance. Multiple case studies have validated the benefits associated with well-implemented AFDD and other energy information systems (Capehart and Brambley 2015; Granderson and Lin 2016).

Despite these benefits and the rapid expansion of AFDD products available in the marketplace, significant barriers to cost-effective AFDD persist, limiting adoption (Capehart and Brambley 2015). Successful AFDD implementations in buildings must achieve high accuracy at low cost; otherwise, the financial benefits of increased energy and operational efficiency do not outweigh the implementation cost. In many cases, the engineering labor and instrumentation costs required to configure, deploy, and maintain AFDD software are prohibitively large, particularly for operators of small commercial buildings. Meanwhile, high false positive rates and misidentified root causes often overwhelm building operators with frivolous or misleading alerts for inconsequential problems, wasting investigative resources and decreasing consumer confidence in AFDD.

FDD methods may be divided into three broad categories: rule-based, model-based, and data-driven (Katipamula and Brambley 2005). Rule-based FDD[1] compares measured building performance to a set of relational rules that describe proper building operation; violation of a rule triggers a fault. Because rules are conceptually simple and easy to implement in software, most AFDD products presently on the market are rule-based. Properly implemented, rule-based FDD can be extremely powerful and highly effective, at both the equipment scale (Zhao, Wen, and Wang 2015) and the whole-building scale (Accenture Consulting and Granderson 2015). However, the development and tuning of effective rules is time-consuming and requires considerable engineering expertise. Moreover, rules created for one building rarely apply to another without modification. Meanwhile, sparse and low-quality instrumentation limits which rules are feasible to deploy and decreases the value of the results obtained. In many buildings, poor instrumentation and poorly tuned rules are the norm rather than the exception.

Model-based FDD uses a model to make predictions about building energy consumption and identifies faults via statistical analysis of the discrepancy between predictions and actual measurements. Model-based methods can achieve high accuracy, but the accuracy achieved depends heavily on the quality of the model (Katipamula and Brambley 2005). Development and tuning of the building model is therefore a major cost consideration. Both complex and simplified physical models have been successfully applied to FDD for buildings (O’Neil et al. 2014; Henze et al. 2015).

Data-driven FDD methods mathematically relate measured inputs to measured outputs and detect faults if the input/output relationship changes significantly. Capozzoli, Lauro, and Khan (2015) provide a recent example of several such methods applied to commercial buildings. Data-driven methods require little information about the underlying building systems, which decreases reliance on engineering expertise. However, data-driven methods require a large amount of training data and are limited in extrapolation beyond the training data range. In addition, data-driven algorithms have no reference for proper operation apart from the training data and therefore cannot detect faults already present in the training data.

In this paper, we present a hybrid, whole-building AFDD approach that combines energy modeling with data-driven analytics. Whole-building AFDD analyzes energy performance holistically, at the building scale, rather than for individual equipment; it therefore requires fewer sensors and less configuration than traditional rule-based methods. By reducing the engineering labor associated with AFDD setup and data integration, this hybrid approach has the potential to drive down implementation costs, enabling cost-effective FDD even for small buildings. Simulation-based trials with a small security building demonstrate the promise of the approach.

Algorithmic Approach

Figure 1 summarizes our proposed hybrid AFDD algorithm. Conceptually, it consists of two distinct stages: fault detection and fault diagnosis. The fault detection engine compares

[1] Katipamula and Brambley (2005) place rule-based FDD within the broader category of “qualitative model-based FDD,” which also includes qualitative physics-based modeling approaches.

measured building performance (typically, interval energy consumption data) with expected performance using a statistical model constructed from historical measurements, weather history, and, if available, a whole-building physics-based model.[2] Significant deviation between measured and expected performance indicates a fault. Given a detected fault, the fault diagnosis engine classifies the type of fault using data-driven models constructed from a large database of simulated fault behavior.

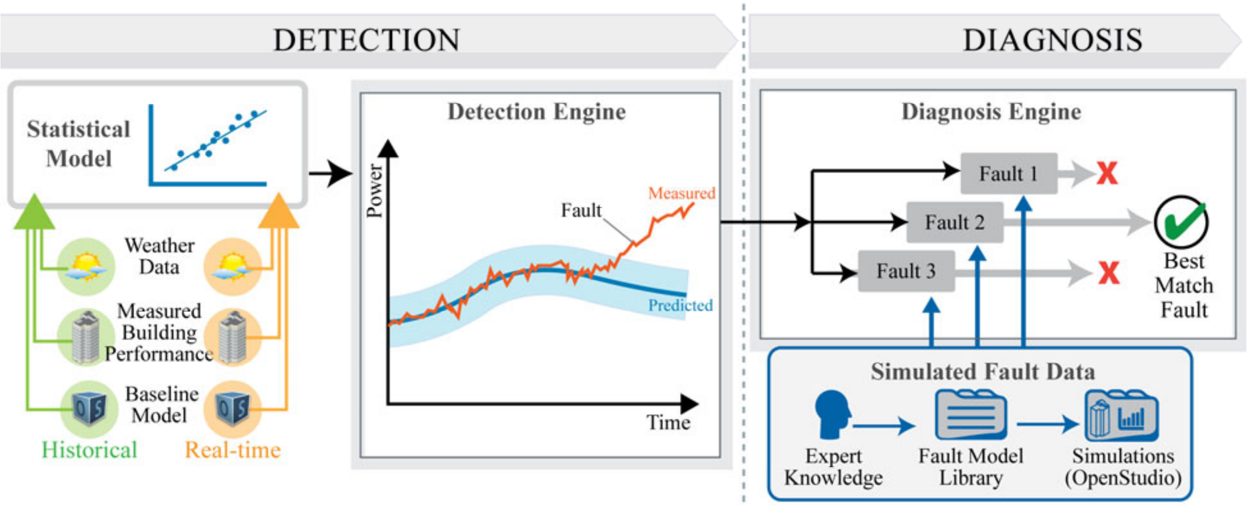

Figure 1. An illustration of the proposed hybrid AFDD process.

Data-driven algorithms require large and comprehensive training data sets, but comprehensive measured data for faults are rarely available. The hybrid algorithm addresses this weakness by leveraging a pre-simulated database of modeled faults to provide rich training data. To create the training data, we superimposed realistic physics-based fault models on a physics-based whole-building energy model for the actual building (or a sufficiently similar building). Both the whole-building energy model and the fault models use the EnergyPlus energy simulation software (DOE 2013) and the OpenStudio development environment (NREL 2014). The fault model library includes both control and equipment faults. To model each fault, we applied first principles and empirical analysis to transform the underlying fault mechanism into altered building control or equipment performance parameters, as described by Cheung and Braun (2015). We then simulated each building under each individual fault using multiple years of weather data to derive time series data of fault impact for use in training the detection engine.

For the fault detection engine, we explored two broad types of algorithms: regression methods and machine learning classification methods. In both cases, historical measurements and, if available, modeled whole-building performance using actual meteorological year (AMY) weather data provided training data for establishing baseline or expected behavior. The regression methods flag abnormal deviation from the expected behavior as a fault, while the machine learning methods compare the observed data to the baseline and simulated fault data to perform a direct classification of the observed behavior as faulted or un-faulted.

For the subsequent fault diagnosis step, we employed machine learning methods to perform fault classification. Each method constructs a classification or clustering model from

[2] An existing energy model may be available from new construction design, a deep retrofit, or an energy audit. If an energy model is unavailable, the detection step becomes purely data-driven, but the diagnosis step will fall back on fault signatures modeled for a similar generic building.

simulated fault data, then attempts to classify new faults into one of the known fault categories. In the end, the AFDD algorithm outputs the most probable class of fault, which the building operator can use to initiate corrective action.

Fault Modeling

To create the fault model library required for the hybrid AFDD algorithm's diagnosis phase, we used three well-established approaches: physical modeling, semi-empirical modeling, and empirical modeling. Physical modeling is the modeling of faults by changing the inputs of the building simulation with the fault level directly; semi-empirical modeling involves changing the simulation inputs based on a simplified fault mechanism; and empirical modeling represents faults by changing the simulation outputs of a building component model using empirical models without directly considering the underlying fault mechanism. The models used for the proposed algorithm are described by Cheung and Braun (2015), which includes models developing using all three methods for air conditioner faults, chiller faults, fan motor faults, and other faults.

Figure 2 illustrates the impact to building energy consumption for an example of an economizer damper fault. A damper stuck in the open position introduces large amounts of unconditioned air to the space. In the winter, the extra cold air increases natural gas consumption; in the summer, the extra hot air increases electricity used for cooling. The AFDD algorithm leverages these performance data to train fault diagnosis models.

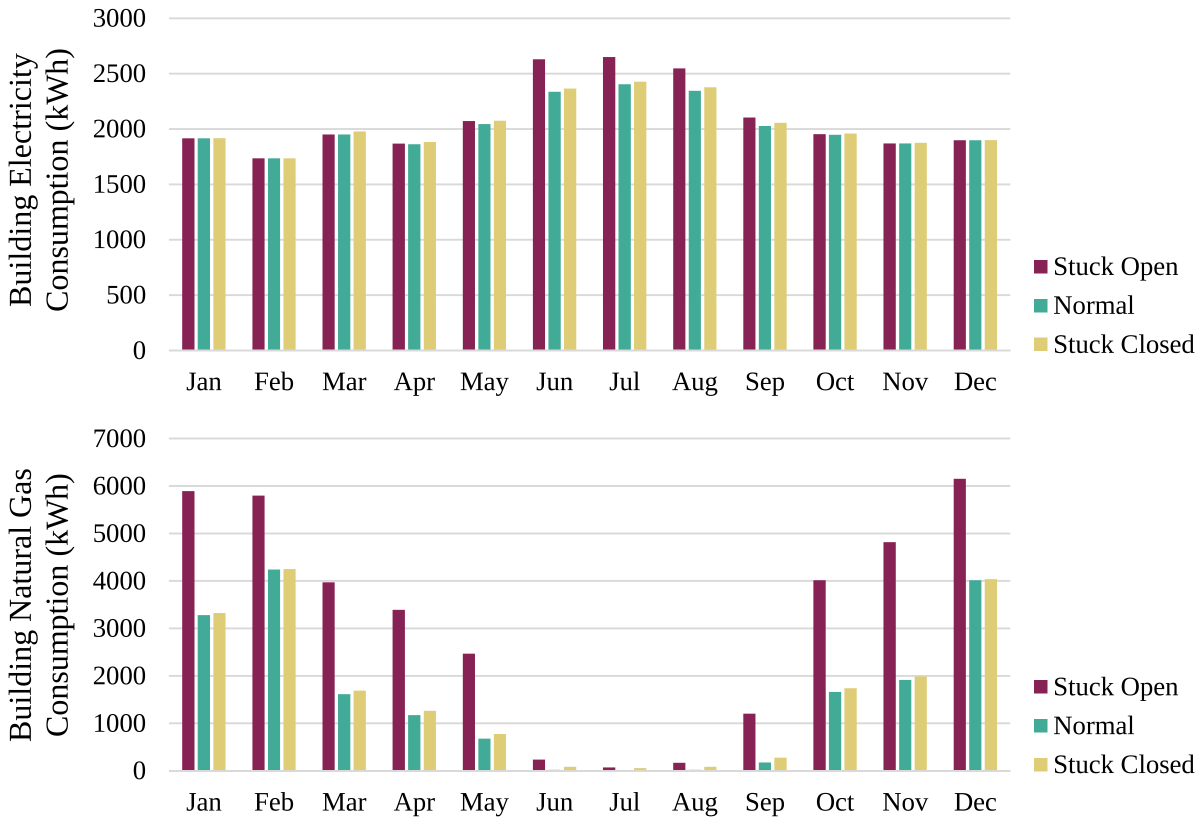

Figure 2. Impact of a stuck economizer outside air damper on building energy consumption.

Fault Detection

For the fault detection engine, we tested two different regression methods and five different machine learning methods (Table 1). The two regression methods are variants of multiple linear regression (MLR). The five machine learning methods represent a variety of black-box clustering approaches that have been successfully applied to FDD in other fields (Fernández-Delgado 2014). For all methods, the only data required are whole-building energy consumption interval data (hourly or 15-minute) and local weather data.

Table 1. Fault detection methods

Type	Method Name	Abbreviation
regression	ordinary least squares	OLS
regression	quantile regression	QR
machine learning	k nearest neighbor	*k*-NN
machine learning	naive Bayes	NB
machine learning	classification and regression tree	CART
machine learning	random forest	RF
machine learning	support vector machine	SVM

Ordinary Least Squares

MLR analysis is the de facto standard statistical method for modeling the linear relationships between two or more predictor (or independent) variables and one response (or dependent) variable. MLR generally uses the ordinary linear least square (OLS) method to minimize the sum of squared errors between the predicted and actual responses (Hutcheson 2011). Estimating prediction intervals is well established and OLS software is readily available.

Although fairly robust, OLS requires certain assumptions. In particular, the error estimated for each predictor variable assumes uniform variance with respect to each predictor variable. If this is not the case, the prediction intervals could be incorrectly estimated. Thus, the prediction from the resulting model can be heavily influenced by extreme outliers.

In the fault detection application as implemented, the algorithm fits a separate MLR model of energy consumption (electricity or gas) for each calendar quarter in the year using the outdoor air drybulb temperature and outdoor air relative humidity as predictor variables. Polynomial terms for outdoor air drybulb temperature are included to model the non-linear behavior. The day of the week and month of the year are included as time series categorical predictor variables to account for the additional changes in the energy use that cannot be predicted by the weather variables alone.[3]

To refine the model, we implemented a stepwise regression procedure. Stepwise regression refines an OLS model using a combination of backward elimination and forward selection processes. The procedure adds predictor variables one by one as it applies OLS, beginning with the variable most highly correlated with the response. As variables enter the model, stepwise regression tests each for significance using the Student-*t* statistic; any insignificant predictors are removed. Throughout this process, Akaike's information criterion

[3] For simplicity, we excluded holidays from the analysis approach and from the simulation trials. However, it is straightforward to include holiday schedule as a categorical variable in the regression analysis.

(AIC), which expresses the balance between model prediction performance and model complexity, is computed for each new model. The procedure terminates when no new trial model obtained by adding or subtracting a variable yields a lower AIC.[4]

Once the baseline model is constructed, we apply it for fault detection. For each new day, the algorithm determines whether the daily average energy consumption falls within the prediction interval. Any observation of energy use that falls outside the prediction interval is considered a fault for that day.[5]

Quantile Regression

Quantile regression (QR) analysis is used in statistics and econometrics. Whereas OLS results in estimates that approximate the conditional mean of the response variable from the predictor variables, QR estimates either the conditional median or other quantiles of the response variable. Unlike OLS, which requires only numerical linear algebra, QR requires solving a linear programming problem. Compared to estimates obtained from OLS, QR estimates are more robust against outliers in the response measurements. Yet the main attraction of quantile regression is that different measures of central tendency and statistical dispersion are available to define arbitrary thresholds of frequency of occurrence deemed to represent fault-free behavior.

In this work, user-specified upper and lower quantiles on the energy consumption define the fault detection limits. The QR implementation shadows the OLS implementation. First, the algorithm fits one QR model for each calendar quarter using the same predictor variables as for OLS. Then, the algorithm determines whether the daily average energy consumption falls between the specified upper and lower quantiles; any value outside this range indicates a fault.

Machine Learning

Machine learning uses pattern recognition techniques to construct algorithms that learn from and make predictions about data (Bishop 2007). When data with example inputs and desired outputs are provided, the algorithm is trained in a “supervised” manner and is able to make predictions about new data. This is known as supervised learning, which is well-suited for FDD problems when training data are available. Machine learning is applicable for both fault detection and fault diagnosis; the difference lies in whether the output classification is a binary (faulted or un-faulted) or is a match to a specific type of fault.

Each machine learning method involves three steps: data filtering, feature extraction, and pattern classification. Because building energy consumption is a function of load conditions such as weather and occupancy, the algorithm requires training data that have similar weather conditions and occupancy schedule as the observed data. The data filtering step selects a subset of the training data with similar ambient temperature, relative humidity, and day of the week.

For both the training and testing stages, the feature extraction step partitions the interval data into individual days and computes six key statistical features for each day: mean, median, maximum, variance, skewness, and kurtosis. The algorithm extracts each feature from the whole-building hourly electricity and natural gas consumption data, for a total of 12 characteristic features. These features represent the load shape in a compact yet informative way.

[4] We also explored the least absolute shrinkage and selection operator (LASSO) regression method for model construction, but found it unnecessary given the small number of predictor variables.

[5] Lower than expected energy consumption may seem advantageous, but could also indicate equipment failure or improper controls operation. Therefore, we consider such observations probable faults.

Following feature extraction, the algorithm constructs a pattern classifier based on the training features and the corresponding fault types; this classifier is used to predict the types of incoming observed faults. Many pattern classifiers have been developed (Fernández-Delgado 2014); we used the following well-known classifiers in this study: k-nearest neighbors (*k*-NN), naive Bayesian (NB), classification and regression tree (CART), random forest (RF), and support vector machines (SVM).

Fault Diagnosis

Following detection of a fault, the next step is to identify the fault type based on the available data, which is known as fault diagnosis. The machine learning methods described in the previous section can also be used for fault diagnosis. The detection stage is a binary classification problem (faulted or un-faulted), while the diagnosis stage is a multi-class problem. For fault diagnosis, the pattern classifier may follow fault detection, or fault detection and diagnosis may be performed in a single monolithic step. For this paper, we tested seven combined FDD algorithm variants (Table 2).[6]

Table 2. Combinations of fault detection and diagnosis methods examined

Approach	Detection Method	Diagnosis Method
MLR for detection, standalone pattern classifier for diagnosis	OLS	CART
	OLS	RF
Machine learning for both detection and diagnosis	*k*-NN	*k*-NN
	NB	NB
	CART	CART
	RF	RF
	SVM	SVM

Results

To explore the relative merits of the various FDD approaches, we applied each method to simulated baseline and faulty energy consumption data obtained from a model of a small security building located in Golden, Colorado, USA. Simulated data provide a “best case” scenario that is useful for comparing the performance of alternative algorithms, which was a primary goal of the research. Testing with simulated data rather than measured data provides several benefits. It:

- Eliminates potential sensing and data quality issues in the test data
- Allows multiple years’ worth of data to be tested in a short period of time
- Offers completely reliable ground truth knowledge of fault states
- Provides a much richer set of faults than is typically available from a real building.

However, the converse of these advantages is that the algorithms as presented are not yet validated in a real-world operating environment, with all the corresponding nuance, noise, and uncertainty. Testing with a real building is the next logical step and is planned as future work.

[6] As will be seen in the results section, we found QR to produce significantly higher false positive rates than OLS when used for fault detection. Therefore, we did not test a full FDD method that used QR in the detection step.

Building Model

We tested the FDD algorithms using a calibrated model of an 880 ft^2 (82 m^2), single-story security building located in Golden, Colorado, USA. We selected this model because it represents an actual building with known sequences of operation and because the model had previously been calibrated using measured utility data. The building contains one thermal zone and one plenum zone. The thermal zone is conditioned by a split air conditioner which provides heating using an 8.3-kW gas furnace and cooling using a 17-kW direct expansion coil. To expand the range of possible faults, we modified the model to include an economizer for free cooling and ventilation.

We simulated the building under both normal (un-faulted) and faulted building operation. For testing, we selected a set of seven common HVAC equipment and controls problems found in commercial buildings (Table 3). This is not a comprehensive set of faults applicable to the building model used for the test. However, we chose to limit the faults under consideration in order to concentrate on algorithm development and initial performance evaluation. Future work will include expanding the algorithm to consider more faults types.

Table 3. Description of faults modeled for algorithm comparison tests

Fault	Description	Fault Level Control Variable
Duct fouling	Increase of flow resistance and pressure drop across air ducts due to dust accumulation at heat exchangers or in air filters	Percentage increase of air duct pressure drop, following the model in Cheung and Braun (2015)
Thermostat bias	Fixed bias in thermostat reading of the zone air temperature	Bias from the real air temperature, following the model in Barsarkar et al. (2011)
Economizer outside air temperature sensor bias	Fixed bias at the temperature sensor readings for economizer damper control: return air or outside air, depending on the fault	Bias from the real air temperature, following the model in EnergyPlus version 8.1
Economizer return air temperature sensor bias		
Economizer damper stuck fault	Inability of the economizer controller to vary outdoor air intake because the damper is stuck at a fixed position.	Damper fixed percentage open, following the model in Barsarkar et al. (2011)
Blower fan motor efficiency degradation	Degraded efficiency of fan motor as a result of bearing fault and stator winding fault	Percentage reduction in fan motor efficiency, per the model in Cheung and Braun (2015)
Excessive infiltration	Deterioration of building envelope, increasing infiltration airflow to the building	Percentage increase of air infiltration to the building, modeled as an increase in airflow through the walls

Training and Testing Data Sets

To test the FDD algorithms, we constructed separate training and testing data sets, both based on simulation. The training data are required to train the data-driven fault diagnosis algorithms; thus, construction of a similar training data set would be required for any real-world deployment of the hybrid AFDD algorithm. The simulated testing data, on the other hand, serve as a proxy for measured fault data—data that ultimately will be collected from an actual building in near-real-time using whole-building energy meters.

To create the training data, we simulated both the baseline model and the seven faults at a variety of fault levels (Table 4). Imposing the fault models with fault levels in Table 4 creates 40 fault scenarios. Together with the baseline, these scenarios were simulated using 2012 weather data for Golden, Colorado to create 41 full-year time series. The results were divided into 14,965 daily simulation results to serve as training data for the FDD algorithms.

Table 4. Fault levels imposed on the building model to construct the training data set

Fault	Location	Fault Levels
Duct fouling	Split AC blower fan	Increased pressure drop across air ducts by 10%, 20%, 30%, 35%
Thermostat bias	Thermal zone	−5K, −3K, −2K, −1K, +1K, +2K, +3K, +5K
Economizer outside air temperature sensor bias	Economizer	−5K, −3K, −2K, −1K, +1K, +2K, +3K, +5K
Economizer return air temperature sensor bias	Economizer	−5K, −3K, −2K, −1K, +1K, +2K, +3K, +5K
Economizer damper stuck fault	Economizer	Opening stuck at 0%, 25%, 50%, 75%, 100%
Blower fan motor efficiency degradation	Split AC blower fan	Decreased efficiency by 10%, 20%, 30%, 40%
Excessive infiltration	Thermal zone	Increased infiltration airflow by 10%, 20%, 30%

We applied a similar procedure to produce a data set for testing but with a randomized set of fault levels and a different weather year (2011), which yielded 10,585 daily simulation results. In the absence of information regarding the relative prevalence of each fault type, we elected to sample the data such that each fault model was imposed on an equal number of simulated days.

The simple presence of an active fault model does not necessarily produce faulty behavior in the building. For example, if the economizer damper is stuck closed when weather conditions dictate it should be closed, no fault is present. Instead, we defined faulty behavior as any behavior that produces a 5% deviation (minimum 100 W) from the simulated baseline daily energy consumption (either electricity or natural gas).[7] Using this definition for a fault, the test data contained 4,805 days—roughly 45% of the samples—exhibiting faulty behavior.

Algorithm Performance

To test the relative merits of the algorithms, we implemented each algorithm in the R statistical computing environment (R Core Team 2016) using various open source packages

[7] Herein is the advantage of using simulated data: the deviation from the baseline for each fault is known exactly.

(Koenker 2015; Kuhn 2016; Liaw and Wiener 2002; Meyer et al. 2015; Therneau, Atkinson, and Ripley 2015). All methods had access to only two weather predictor variables (outside air dry bulb temperature and relative humidity) and two response variables (whole-building electricity and natural gas energy consumption). Most algorithms also employed auxiliary categorical variables calculated from the time data, such as the month of the year and day of the week.

The test procedure involved two steps. First, we provided each FDD algorithm with the baseline and fault data in the training data set to allow model construction. Second, we tested the FDD algorithm against each of the daily simulations in the testing data set. For each day, the algorithms output two pieces of information: a classification of the behavior as faulted or un-faulted and, if a fault was present, a prediction of the fault type. We compared these results against the ground truth known from the testing data set to compute the true positive rate (TPR), false positive rate (FPR), and frequency of diagnosis statistics for each algorithm.

Figure 3 displays the receiving operating characteristic (ROC) curve for the seven fault detection methods; the ROC curve plots TPR against FPR. For the MLR methods, the plot shows a curve derived by varying the confidence level used to determine the control limits for fault detection. For example, a confidence level of 90% indicates that 90% of normal days would theoretically fall in "un-faulted" category. For the machine learning methods, the plot shows the binary classification results from a single trial. The gray diagonal line represents the theoretical performance of a "random chance" detection algorithm, as would occur if the algorithm simply flipped a virtual coin to classify each input day as faulted or un-faulted.

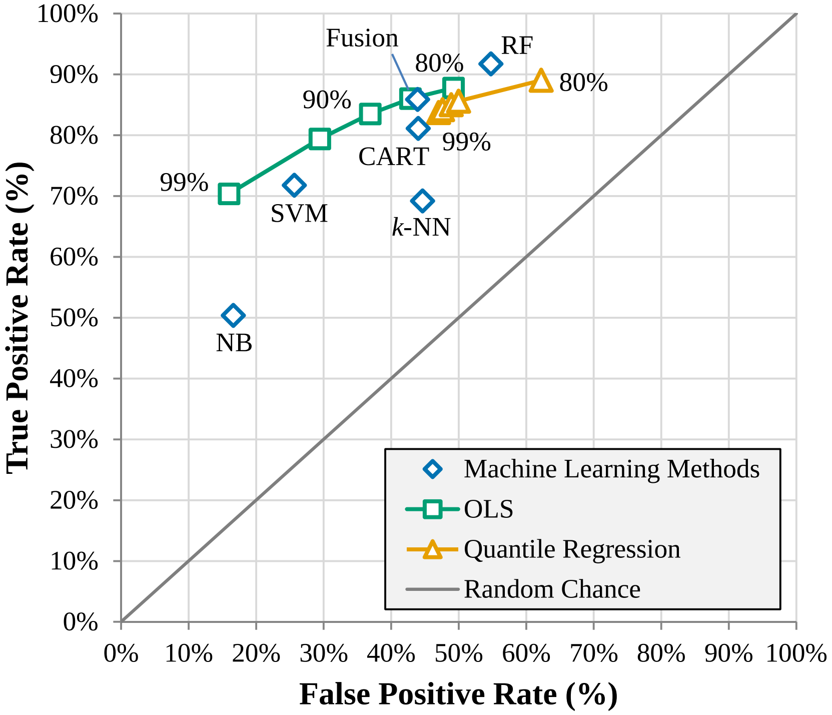

Figure 3. ROC curve for the fault detection methods tested. For OLS and QR, the annotations indicate the progression of confidence levels used to obtain the fault detection control limits.

Of all the detection methods, OLS provided the best balance of TPR and FPR. QR provided marginally higher TPR than OLS for the same confidence levels, but also a significantly larger FPR. In trials, we observed that the QR method resulted in very tight control limits and consistently experienced numeric difficulty when fitting the outer quantiles. We hypothesize that the highly structured and non-random nature of simulation data may be the primary cause for this behavior. Therefore, QR should not be discounted entirely based on these results, as it has been known to perform well when used with real-world data (Henze et al. 2015). The machine learning methods collectively performed slightly more poorly than OLS for this building, which may indicate that the selected features were not sufficiently discriminative. Feature selection or the addition of more features may improve performance; this is future work.

All detection methods had higher FPR than is desirable. For the MLR methods, this may be because the control limits indicate bounds much tighter than the 5% threshold we set as a definition of faulted behavior—which results in flagging "faults" with only a very minor energy impact. This may again be an artifact of highly structured simulation data, but it could also occur with well-behaved real buildings. One possibility for reducing FPR is to introduce a secondary significance check that examines the percentage difference from the expected energy consumption and avoids flagging days that fall within the 5% threshold; similar desensitization is possible for machine learning algorithms by tuning the input parameters. This type of tuning is analogous to finding the correct thresholds in rule-based AFDD, but the model-based approach has a natural advantage: sensitivity is intuitively tied directly to energy and therefore to cost.

Figure 4 compares the frequency of diagnosis of the seven FDD methods for each fault. For the combined MLR/machine learning methods, we present the results that used OLS at a confidence interval of 95%, which provided the best tradeoff between TPR and FPR.

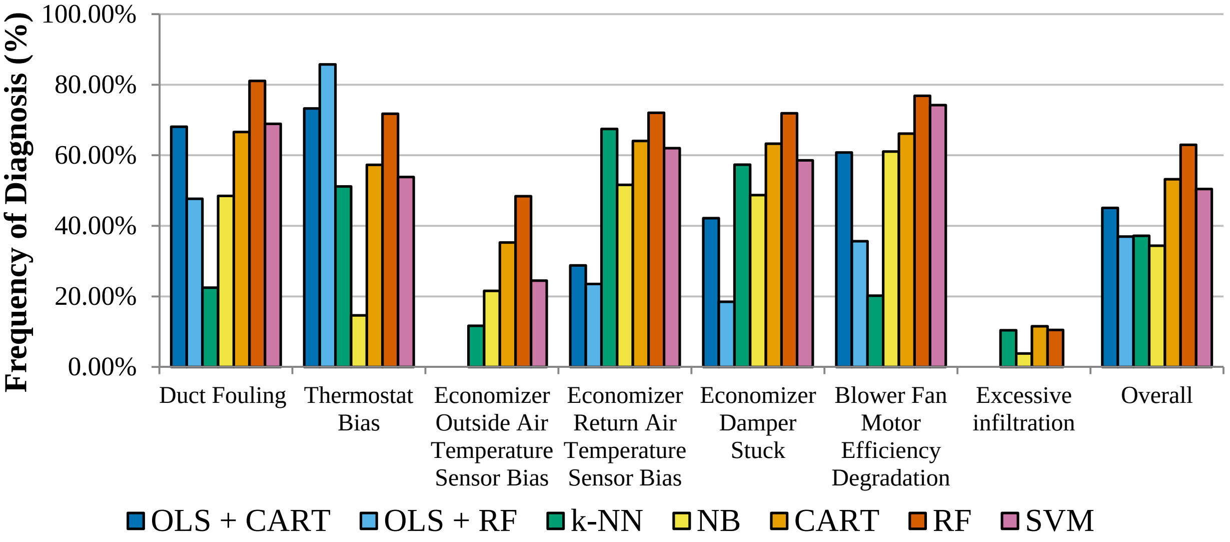

Figure 4. Frequency of correct diagnosis of each fault type for the seven FDD methods tested.

Diagnosis results varied by fault. All methods consistently misclassified the economizer outside air temperature sensor bias and excessive infiltration faults. Thermostat bias had the highest diagnosis performance overall but was the most common incorrect identification for other faults. For example, nearly all excessive infiltration faults were misclassified as a thermostat bias. These results hint that the underlying fault signatures may be very similar at the level of whole-building energy consumption. The similarity of signatures in turn suggests that other simple metrics, such as an estimate of unmet hours based on internal temperature data, may improve diagnosis accuracy—at the expense of additional instrumentation. Future work will explore fault signatures to discover differences that can be exploited in the fault classification process.

Conclusion

When trained using simulated fault data, both regression and machine learning methods show great potential for whole-building FDD using minimal input data. The proposed hybrid approach has unique advantages with respect to pure rules-based, model-based, and data-driven approaches. Unlike rules-based approaches, the hybrid approach can diagnose faults using only whole-building data. Hybridization with data-driven regression and pattern-recognition algorithms allows the use of pre-simulated data for fault diagnosis, rather than real-time simulation of faults as might be required in a pure model-based approach. Finally, the hybrid approach overcomes the excessively long training period that is often required by pure data-driven approaches by using modeled fault data for training.

Nevertheless, the proposed hybrid approach faces challenges. In our initial trials, detection rates were high, but false positive rates were also greater than desired. Future work should explore algorithm tuning and heuristic methods for reducing false positives, the cost/benefit tradeoff of making additional variables available to the detection and diagnosis algorithms, and the performance of the algorithms for more buildings, both simulated and real. Benchmarking of model-based algorithms against conventional rule-based FDD using controlled real-world tests would also improve understanding of the tradeoff between instrumentation cost and FDD accuracy for each approach.

Ultimately, the proposed hybrid approach has the potential to drive down FDD implementation costs, enabling cost-effective FDD with high accuracy for a wide range of commercial buildings. However, despite the projected reduction in deployment cost compared to rule-based FDD, it is unclear whether sufficiently accurate physics-based models can be obtained at a sufficiently low cost to make the approach financially viable. Automatic generation of high-quality physics-based models may overcome this barrier; we suggest this as future work.

References

Accenture Consulting Staff and J. Granderson. 2015. “Energy-Smart Buildings Demonstrating How Information Technology Can Cut Energy Use and Costs of Real Estate Portfolios.” In *Automated Diagnostics and Analytics for Buildings*. Lilburn, GA: The Fairmont Press, Inc.

Basarkar, M., X. Pang, L. Wang, P. Haves, and T. Hong. 2011. “Modeling and Simulation of HVAC Faults in EnergyPlus.” In *Proceedings of Building Simulation 2011*. 2897–2903. Sydney, Australia: IBPSA.

Bishop, C. 2006. *Pattern Recognition and Machine Learning*. New York, NY: Springer.

Brambley, M., D. Hansen, P. Haves, D. Holmberg, S. McDonald, K. Roth, and P. Torcellini. 2005. *Advanced Sensors and Controls for Building Applications: Market Assessment and Potential R&D Pathways*. Richland, WA: Pacific Northwest National Laboratory. http://www.pnl.gov/main/publications/external/technical_reports/pnnl-15149.pdf

Capehart , B., and M. Brambley. 2015. “Introduction,” In *Automated Diagnostics and Analytics for Buildings*. Lilburn, GA: The Fairmont Press, Inc.

Capozzoli, A., F. Lauro, and I. Khan. 2015. “Fault Detection Analysis Using Data Mining Techniques for a Cluster of Smart Office Buildings.” *Expert Systems with Applications*. 42: 4324–4338.

Cheung, H., and J. Braun. 2015. *Development of Fault Models for Hybrid Fault Detection and Diagnostics Algorithm*. Golden, CO: National Renewable Energy Laboratory. http://www.nrel.gov/docs/fy16osti/65030.pdf

DOE (U.S. Department of Energy). 2013. *EnergyPlus Energy Simulation Software, Version 8.1*. Washington, DC: U.S. Department of Energy. https://energyplus.net/

Fernández-Delgado, M., E. Cernadas, S. Barro, and D. Amorim. 2014. “Do We Need Hundreds of Classifiers to Solve Real World Classification Problems?” *Journal of Machine Learning Research*, 15: 3133–3181.

Granderson, J., and G. Lin. 2016. “Building energy information systems: synthesis of costs, savings, and best-practice uses.” *Energy Efficiency*. Online: http://link.springer.com/article/10.1007/s12053-016-9428-9

Henze, G., G. Pavlak, A. Florita, R. Dodier, and A. Hirsch. 2015. “An Energy Signal Tool for Decision Support in Building Energy Systems.” *Applied Energy*. 138: 51–70.

Henze, G., S. Pless, A. Petersen, N. Long, and A. Scambos. 2015. “Control Limits for Building Energy End Use Based on Frequency Analysis and Quantile Regression.” *Energy Efficiency*, 8(6): 1077–1092.

Hutcheson, G. 2011. “Ordinary Least-Squares Regression.” In L. Moutinho and G. Hutcheson, *The SAGE Dictionary of Quantitative Management Research*. Thousand Oaks, CA: SAGE Publications Ltd. 224–228. https://datajobs.com/data-science-repo/OLS-Regression-%5BGD-Hutcheson%5D.pdf

Katipamula, S., and M. Brambley. 2005. “Methods for Fault Detection, Diagnostics, and Prognostics for Building Systems – A Review, Part I.” *HVAC&R Research*. 11(1): 3–25.

Koenker, R. 2015. “quantreg: Quantile Regression.” R package version 5.19. http://CRAN.R-project.org/package=quantreg

Kuhn, M. 2016. “caret: Classification and Regression Training.” R package version 6.0-64. http://CRAN.R-project.org/package=caret

Liaw, A., and M. Wiener. 2002. “Classification and Regression by randomForest.” *R News*. 2(3): 18–22. https://cran.r-project.org/doc/Rnews/Rnews_2002-3.pdf

Meyer, D., E. Dimitriadou, K. Hornik, A. Weingessel and F. Leisch. 2015. “e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien.” R package version 1.6-7. http://CRAN.R-project.org/package=e1071

NREL (National Renewable Energy Laboratory). 2014. *OpenStudio, Version 1.5.0*. Golden, CO: National Renewable Energy Laboratory. https://www.openstudio.net/

O'Neill, Z., X. Pang, M. Shashanka, P. Haves, and T. Bailey. 2014. “Model-Based Real-Time Whole Building Energy Performance Monitoring and Diagnostics.” *Journal of Building Performance Simulation*, 7(2): 83–99.

R Core Team. 2016. “R: A Language and Environment for Statistical Computing.” Vienna, Austria: R Foundation for Statistical Computing. http://www.R-project.org/

Roth, K., D. Westphalen, M. Feng, P. Llana, and L. Quartararo. 2005. *Energy Impact of Commercial Buildings Controls and Performance Diagnostics: Market Characterization, Energy Impact of Building Faults and Energy Savings Potential*. Cambridge, MA: TIAX LLC.

Therneau, T., B. Atkinson, and B. Ripley. 2015. “rpart: Recursive Partitioning and Regression Trees.” R package version 4.1-10. http://CRAN.R-project.org/package=rpart

Zhao, Y., J. Wen, and S. Wang. 2015. “Diagnostic Bayesian Networks for Diagnosing Air Handling Units Faults – Part II: Faults in Coils and Sensors.” *Applied Thermal Engineering*. 90: 145–157.