

A feature extraction framework with entropy on graphs for cross-dataset building fault detection

Jiajing Huang, Abhidnya Patharkar, Teresa Wu, Jin Wen, Zheng O'Neill and K.Selcuk Candan

Abstract— Simulation is commonly adopted in developing building automated fault detection and diagnosis (AFDD) strategies. However, simulations often fall short in accurately representing real-world scenarios, which hinders the efficacy of models trained on such data for identifying faults in actual buildings. To tackle this challenge, we present a new approach for feature extraction that leverages entropy obtained from graph structures. These structures are constructed based on features that can distinguish between normal and faulty conditions. This method includes acquiring graph structures from simulated data, extracting their entropies as features to train AFDD models. Then, the process of obtaining entropies from graphs is replicated for real building data, and the trained AFDD model is applied to conduct tests on them. Empirical findings illustrate that our suggested approach enables fault detection in real-world scenarios, even when the model is trained with simulated data. The features extracted by our proposed approach surpass the baseline, which consists of GNN embedded features, in terms of fault detection performance. Therefore, we infer that our method has the potential to take advantage of simulation for real building fault detection.

Index Terms— Failure Detection and Recovery; Big-Data and Data Mining; AI-Based Methods

I. INTRODUCTION

Buildings are complex with multiple subsystems and controllers, and the most critical ones are heating, ventilation and air conditioning (HVAC) systems. According to 2022 Global Status Report [1], buildings are responsible for approximate 135 exajoule operational energy demand and 10 Gross tonnage energy-related carbon dioxide emission in 2021 [1]. HVAC systems usually suffer from faults, including sensor failure, equipment failure and malfunctioning operations, which can lead to uncomfortable indoor environments, poor air quality, high maintenance costs, and excessive energy waste. It is reported that energy waste [2] caused by the faulty conditions in buildings accounts for almost 30% of total energy consumption [3]. Therefore, automatic fault detection and diagnosis (AFDD), a process including fault detection, identification, and isolation, is critical to ensure building operational reliability. Studies have shown that AFDD is able to achieve annual 10% median energy savings with two-year simple payback periods in the United States [4].

AFDD methods can be either qualitative model-based, quantitative model-based, or process history-based [5]. The qualitative and quantitative methods, including rule-based and physics-model approaches, are well-understood by building engineers and researchers due to their interpretability from a building physical domain. Nevertheless, these approaches are constrained by issues like limited adaptability across various systems and substantial expenses linked to implementation because of the necessity to tailor physics-based models, regulations, and thresholds for each building system. Consequently, they have not gained widespread acceptance in the architectural industry market [6]. The process-history ones, including data-driven and machine learning approaches, address the challenges of scalability and high implementation costs. Due to the progress in data science and machine learning, as well as the growing adoption of building automation systems (BAS) and sensor technologies, data-driven AFDD has become prominent. Different from those methods typically integrated into off-the-shelf AFDD products, data-driven AFDD techniques do not rely on prior expert knowledge and heuristics, thus offering the possibility of reducing costs while maintaining high accuracy in fault detection and diagnosis [7]. In recent studies, there has been a growing emphasis on data-driven AFDD approaches, both at the component and whole-building scales. For instance, Zhang et al. [8] study how data-driven AFDD influence the economic implications of installing new sensors; Li et al. [9] study data-driven AFDD on the vulnerabilities in BAS and strategies for cyber resilient control. The success of these data-driven methods significantly depends on the quality of the training data, underscoring the importance of high-quality training data for the efficacy of AFDD tools.

Most data-driven AFDD strategies are developed and assessed using simulated data, primarily because obtaining and analyzing real building data is challenging. Due to the ease of use and lower costs, simulations have thus been widely applied in building research. For example, Kang et al. [10] apply simulation modeling to predict optimal control for ice-based thermal energy storage systems in commercial buildings. Ye et al. [11] provide insights on the applications of simulated system modeling for grid-interactive efficient building detection strategy. Malkawi et al. [12] propose a simulation-based architecture to monitor building normal operations. Yet AFDD strategies trained by simulations may not achieve satisfactory effectiveness when directly applied to real

*Research supported by National Science Foundation.

Jiajing Huang, Abhidnya Patharkar, Teresa Wu and K. Selcuk Candan are with the School of Computing and Augmented Intelligence, Arizona State University, Tempe, AZ 85281, USA. jhuan177@asu.edu, apathark@asu.edu, teresa.wu@asu.edu, candan@asu.edu

Jin Wen is with the Department of Civil, Architectural and Environmental Engineering, Drexel University, Philadelphia, PA 19104, USA. jinwen@drexel.edu

Zheng O'Neill is with the J. Mike Walker '66 Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843, USA. zoneill@tamu.edu

Please address all correspondence to Jiajing Huang.

building systems, mainly because measurements from simulation system data may still contain different information from real-world scenarios, as concluded in [13]. As simulations rely on real-world data and are thus bound by physical laws, we assume that the relationships among various building components may remain consistent between real-world and simulated datasets. This assumption prompts us to explore leveraging these relationships to enhance the AFDD system. Consequently, we propose a method of extracting these relationships by training the model on simulated data and applying these learned relationships to real data, utilizing graph techniques for this purpose. Graph techniques utilize a collection of interconnected elements to represent connections and address issues in intricate systems, making them suitable for capturing relationships from simulated data and utilizing them for real data analysis in this context. The graph comprises nodes representing features and edges symbolizing the connections between them, reflecting relationships or interdependencies. In this scenario, we generate the graph structures using simulated data by considering the relationships between various attributes. The AFDD model is then trained using the entropies obtained from these configurations. This process of constructing graphs is iterated with actual data to calculate entropies, and ultimately, the model trained on simulated data entropies is tested using the entropies from the real data to make predictions. This method illustrates how graph structures can enhance the application of knowledge from simulated data to make predictions on real data, thereby enhancing the predictive accuracy of models trained on simulated data and evaluated on real data.

The study thus introduces an innovative method for feature extraction from graphs to facilitate the implementation of cross-dataset AFDD strategies for HVAC systems. Initially, we detect important pairwise feature relationships within simulated building data and utilize these relationships as connections to create multiple graphs for further analysis. Once the graph structures are established, corresponding entropies are extracted using the eigen-entropy technique, developed in a previous investigation [14]. Subsequently, the same entropy extraction process is carried out for real data. Ultimately, data-driven AFDD models are trained using simulation data entropies as features and then evaluated on real building data using the corresponding entropies. The rest of the paper is organized as follows. Methodology is detailed in Section II. Experiments and results are presented in Sections III, and IV, respectively. Conclusions and future work are drawn in Section V.

II. METHODOLOGY

In this section, we will introduce the graph structured data and eigen-entropy (EE). Next, we will outline the procedure for generating graphs and extracting associated EEs as attributes for both simulated and actual building datasets.

A. Graph structure

In graph theory, a graph is a mathematical structure used to present pairwise relationship between objects. A graph, $G(V, E)$, is composed of vertices (V) and edges (E) that connect pairs of vertices. Typically, given a graph with m vertices (e.g., in the context of building, m features), it can be represented by

a m by m adjacency matrix, $\mathbf{r}$, where elements of the matrix indicate edges, as illustrated in Fig. 1.

$$\begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ r_{21} & r_{22} & \cdots & r_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mm} \end{pmatrix}$$

Fig. 1: An example of graph adjacency representation

As discussed earlier, when components in a building system are viewed as vertices and pairwise interactions among components are viewed as edges. The interactions between pairs of building components are depicted using an adjacency matrix, which illustrates the coupling effects. These pairwise interactions correspond to the correlation coefficients among the features.

B. Eigen-entropy

Eigen-Entropy (EE) [14] is an information entropy derived on eigenvalues extracted from the correlation magnitude matrix of multivariate data. Given a building AFDD dataset with n samples and m features, the corresponding correlation magnitude matrix on feature space is defined as

$$\mathbf{C}^* = \begin{pmatrix} 1 & c_{12}^* & \cdots & c_{1m}^* \\ c_{21}^* & 1 & \cdots & c_{2m}^* \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1}^* & c_{m2}^* & \cdots & 1 \end{pmatrix} \quad (1)$$

where c_{jk}^* is the absolute value of correlation coefficient (correlation magnitude) between feature j and k , $c_{jk}^* \geq 0$.

EE is defined as

$$EE = - \sum_{i=1}^m \frac{\lambda_i}{m} \log \frac{\lambda_i}{m} \quad (2)$$

where λ_i is the eigenvalue extracted from $\mathbf{C}^*$. In other words, Eigen-entropy is utilized to extract features that represent correlations, as it has been demonstrated to be a successful method for capturing relationships between features.

In the building domain, the interconnections between components are often assessed through correlations [15]. Subsequently, the graph's adjacency matrix resembles the correlation magnitude matrix in case of EE. While the correlations are based on simulated data and represented through edges, it is anticipated that the correlation values may not be precisely identical due to potential noise in real data. However, the general trends of the significant correlations are expected to remain consistent as both the simulated and real building data adhere to physical principles, which is validated by domain knowledge [13]. Therefore, we favor employing an EE that illustrates the correlation patterns over utilizing direct correlation values obtained from simulated data. This enables the use of patterns observed in features derived from simulated data to analyze trends in real data, rather than relying solely on correlation values.

C. Proposed method

Fig. 2 illustrates the flowchart of conducting simulation-to-real AFDD analysis with the method we have put forward. The objective is to establish a graphical structure from both

simulated and actual data, extract the relevant EE values as features to facilitate cross-dataset analysis, specifically training on simulated data and evaluating on real data. Thus, during the training stage, our emphasis is on simulation data.

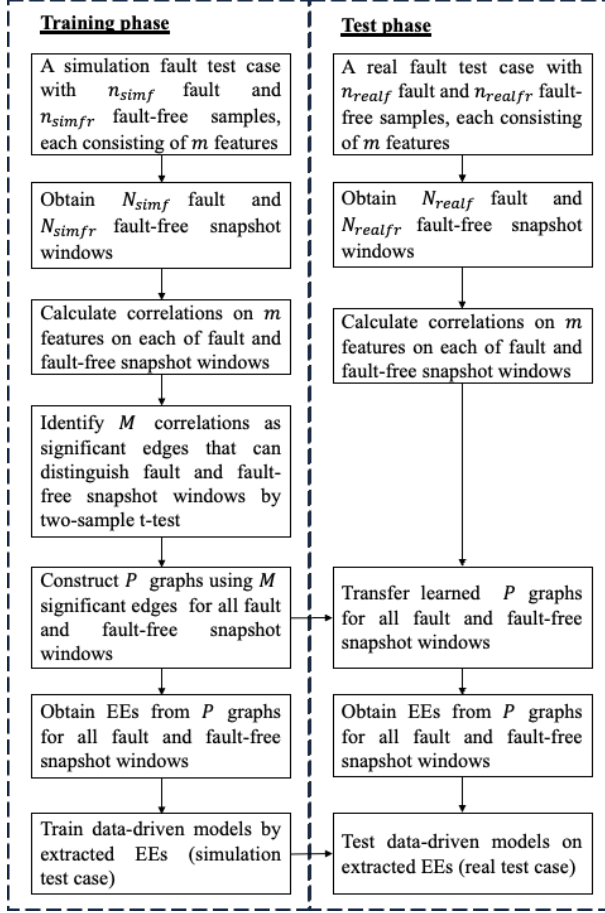


Fig. 2: Flowchart of the proposed method

The process is outlined as follows. Initially, we identify correlations among features that can be used to construct a graph structure. This involves evaluating the discriminatory power of these correlations in distinguishing between faulty and fault-free conditions. The method for evaluating this is elaborated on later in this section. First, let's examine the data: it consists of time series data for multiple features corresponding to both faulty and fault-free datasets. To accommodate the temporal aspect, we capture snapshots of these datasets at regular intervals defined by a window size, denoted as W . Suppose we have n_{simf} samples for the simulated faulty dataset and n_{simfr} samples for the simulated fault-free dataset. For each snapshot, we extract N_{simf} continuous samples (where $N_{simf} = n_{simf}/W$) for the faulty dataset and N_{simfr} samples (where $N_{simfr} = n_{simfr}/W$) for the fault-free dataset. Assuming there are m features in total, we compute feature correlations by evaluating each pair of features using samples from each snapshot window. These correlations are computed for every snapshot window across all features, resulting in N_{simf} or N_{simfr} correlation values for each pair of features. Following this, we conduct a two-

sample t-test to identify M significant feature correlations that effectively distinguish between faulty and fault-free conditions. These significant correlations form the edges for constructing graphs. Once these significant correlations (M in total) are determined, we proceed to construct graphs for each snapshot window in both faulty and fault-free datasets. The construction process is detailed as follows: for each snapshot window, we select one feature with identified significant correlation as the starting node. The significant correlations of this feature with other features, determined in the previous step, serve as edges, while the correlated features become additional nodes. This process is repeated for each significant feature in each snapshot window. Features with only two significant correlations are omitted since a minimum of three correlations is required to construct a graph.

Following the procedure mentioned above, for a given set of M edges, we obtain (say P) number of graphs for every simulation snapshot, and extract corresponding EEs as features. In the test phase, we follow the same procedure indicated above to extract EEs. As a result, we have EEs generated for actual building data and simulated data. The EEs from the simulation are then employed to train a data-driven model, which is subsequently tested using the EEs from the real data. This method allows for the application of training EEs obtained from simulated data to real data by deducing the EEs of the real data, thus enabling cross-dataset AFDD.

III. EXPERIMENTS

A. Experimental datasets

Simulation building datasets used for training in this study are generated by Lawrence Berkeley National Laboratory (LBNL) [16] from a single-duct variable-air-volume (VAV) air handling unit (AHU) virtual testbed, which provides heating and cooling to the middle floor of a three-story DOE large office reference building. Fig. 3 illustrates this building floor layout. As can be noticed, the conditioned floor space consists of a single interior zone and four perimeter zones, where AHU distributes conditioned air by five VAV boxes.

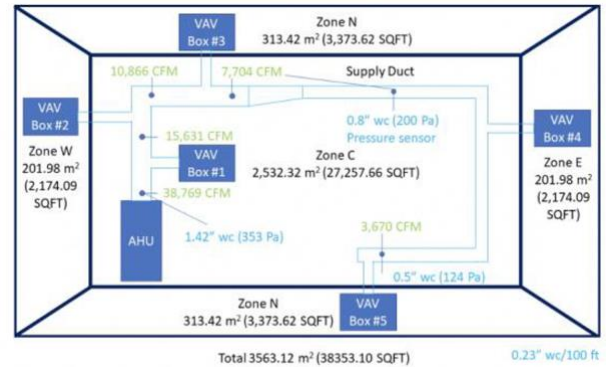


Fig. 3: Simulation virtual testbed floor layout (adapted from [16])

Real building datasets used for test are generated from the ASHRAE 1312 research project [17]. These datasets are collected from a laboratory building that is set up like a small office building whose layout is shown in Fig 4. As can be

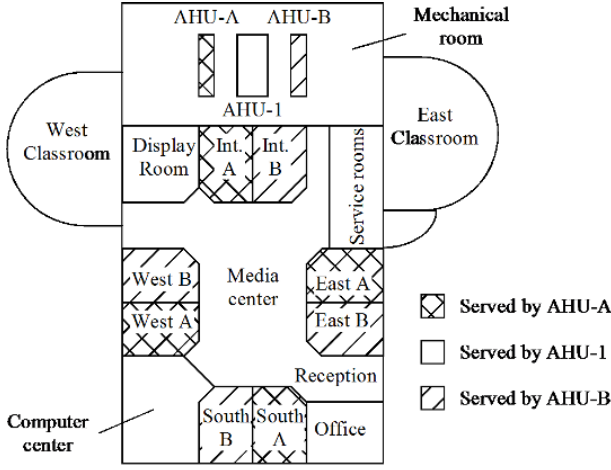


Fig. 4: Energy resource station (ERS) setup (adapted from [17])

observed, the building consists of HVAC systems with two VAV boxes, each serving 4 different rooms, and the design of the test facility is intended to have each AHU serving room with nearly identical loads. Each HVAC system serves rooms facing east, west, south, and one interior room. During the study, System A (AHU-A and all A rooms) is artificially injected with various commonly occurring faults while System B (AHU-B and all B rooms) is continuously operated in a fault-free state.

In this instance, we are examining data related to four distinct faults, with three related to the cooling coil valve (CCV) and the remaining one associated with the outdoor air damper (OAD). It is essential that the fault types remain consistent between the real and simulated data for cross-data analysis, as altering the fault type can potentially result in changes to the adjacency matrix, and also the corresponding EE values. We generate snapshot windows for every fault test case, for both simulation, and actual building data, comprising of both faulty and fault-free instances, each with a window size of 30 ($W = 30$) as recommended in [18]. The number of snapshot windows for each dataset is detailed in Table I. Additionally, Table II outlines the features utilized to identify important edges for constructing graphs.

TABLE I: Summary of the number of snapshot windows for each fault case

Case No.	Training datasets (LBNL simulation)			Test datasets (ASHRAE real building data)		
	Fault case	# snapshot		Fault case	# snapshot	
1	CCV	N_{simf}	5.7k	CCV	N_{realf}	20
	25% Open	N_{simfr}	5.7k	15% Open	N_{realfr}	20
2	CCV	N_{simf}	5.6k	CCV	N_{realf}	20
	50% Open	N_{simfr}	5.7k	65% Open	N_{realfr}	20
3	OAD	N_{simf}	5.7k	OAD	N_{realf}	20
	10% Open	N_{simfr}	5.7k	0% Open	N_{realfr}	20
4	OAD	N_{simf}	3.9k	OAD	N_{realf}	20
	75% Open	N_{simfr}	5.7k	45% Open	N_{realfr}	20

* N_{simf} : simulation fault snapshot windows; N_{simfr} : simulation fault-free snapshot windows; N_{realf} : real fault snapshot windows; N_{realfr} : real fault-free snapshot windows.

TABLE II: Summary of original sensor features

Feature index	Feature name	Feature description
1	SF-WAT	AHU supply air fan power
2	MA-TEMP	AHU mixed air temperature
3	OA-TEMP	AHU outdoor air temperature
4	RA-TEMP	AHU return air temperature
5	RA-DMPR	AHU return air damper position

6	SA-TEMP	AHU supply air temperature
7	SF-SPD	AHU supply air fan speed
8	RF-SPD	AHU return air fan speed
9	OA-DMPR	AHU outdoor air damper position
10	CHWC-VLV	AHU cooling coil valve position
11	RF-WAT	AHU return air fan power

B. Evaluation

We utilize two classification models, namely decision tree (DT) and random forest (RF), in order to classify derived EEs. These models are widely employed in machine learning to develop data-driven AFDD for identifying fault symptoms.

The next objective is to assess whether utilizing EEs outperforms the utilization of other features extracted from the graphs. To achieve this, we employ a baseline technique known as GNNs. We extract features using GNNs from the graphs we have created and then contrast the classification effectiveness achieved with our approach against that achieved with GNNs.

According to [19], four metrics, AUC, recall, and precision, F-measure (F) are commonly used as data-driven AFDD performance evaluations. The recall, precision, and F-measure are calculated as:

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (4)$$

$$F = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (5)$$

The samples in our study represent the feature values (extracted EE values or GNN features) of various snapshot windows. We define true positive (TP) as the count of faulty snapshot windows correctly recognized; true negative (TN) as the count of fault-free snapshot windows correctly recognized; false positive (FP) as the count of fault-free snapshot windows incorrectly identified as faulty; false negative (FN) as the count of faulty snapshot windows incorrectly identified as fault-free.

Recall is thus defined as the number of correctly identified faulty samples over total number of faulty samples. Precision is defined as the number of true faulty samples over total number of predicted faulty samples. F-measure is a combined metric derived from recall and precision. Three metrics emphasize more on detecting true faulty samples.

AUC, or area under the curve, is to measure the performance of a classifier [20], representing the degree of separability archived by the classifier. The value of AUC ranges from 0 to 1, and if AUC is equal to or lower than 0.5, it indicates no or poor ability of the classifier to distinguish two classes [20]; if $AUC > 0.6$, then it is said to be acceptable discrimination [21].

IV. RESULTS

Table III presents the count of features derived from GNN embedding and our proposed technique (EE) for each test

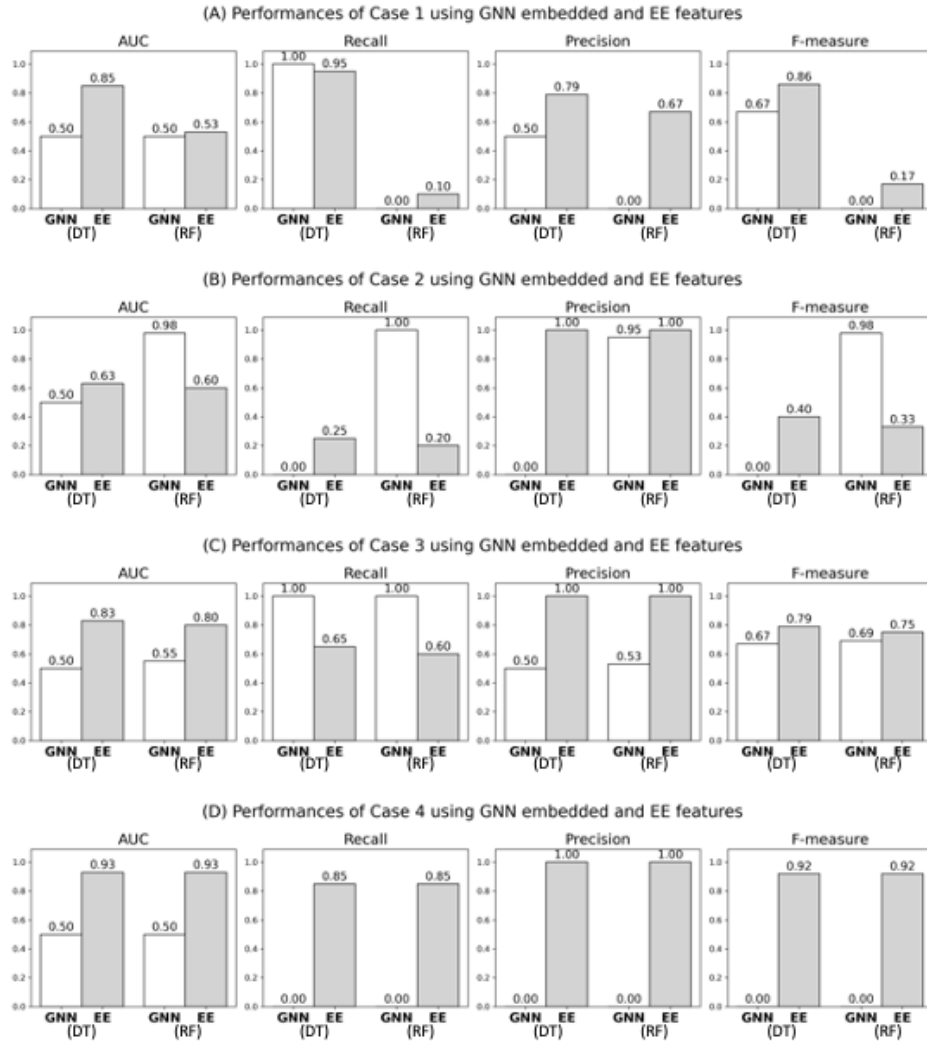


Fig. 5: Performance of Decision Tree (DT) and Random Forest (RF) using GNN embedded and EE features

instance. It is notable that the GNN embedding method produces a greater number of features compared to our proposed method across all fault test cases, despite being derived from identical graphs. The difference can be explained by the fact that GNN is a type of deep learning model, typically necessitating a larger amount of data for drawing conclusions.

TABLE III: Number of extracted features by two methods

Case No.	GNN embedded	Proposed method (EE)
1	16	11
2	88	11
3	32	11
4	120	11

Fig. 5 illustrates the detection performance achieved by DT and RF utilizing GNN embedded features and EE features respectively. For DT training (i.e., GNN(DT) and EE(DT) in each subplot), it is evident that employing EE features results in AUC values exceeding 0.60 for all four test cases. Conversely, the utilization of GNN embedded features does not yield satisfactory AUC values. Specifically, in case 1, our

proposed approach achieves an AUC of 0.85, Recall of 0.95, Precision of 0.79, and F-measure of 0.86, whereas the GNN embedded method attains an AUC of 0.50, Recall of 1.00, Precision of 0.50, and F-measure of 0.67. In case 2, our method reaches an AUC of 0.63, Recall of 0.25, Precision of 1.00, and F-measure of 0.40, while the GNN embedded method obtains an AUC of 0.50, Recall of 0.00, Precision of 0.00, and F-measure of 0.00. Moving on to case 3, our method achieves an AUC of 0.83, Recall of 0.65, Precision of 1.00, and F-measure of 0.79, in contrast to the GNN embedded method which results in an AUC of 0.50, Recall of 1.00, Precision of 0.50, and F-measure of 0.67. Lastly, in case 4, our proposed method attains an AUC of 0.93, Recall of 0.85, Precision of 1.00, and F-measure of 0.92, while the GNN embedded method reaches an AUC of 0.50, Recall of 0.00, Precision of 0.00, and F-measure of 0.00.

For RF training (i.e., GNN(RF) and EE(RF) in each subplot), it is evident that employing EE features can achieve an AUC > 0.60 for all test cases except case 1, whereas using GNN embedded features does not exhibit satisfactory AUC for all test cases except case 2. Specifically, in case 1, our

proposed approach attains an AUC of 0.53, Recall of 0.10, Precision of 0.67, and F-measure of 0.17, while the GNN embedded method achieves an AUC of 0.50, Recall of 0.00, Precision of 0.00, and F-measure of 0.00. For case 2, our proposed method achieves an AUC of 0.60, Recall of 0.20, Precision of 1.00, and F-measure of 0.33, whereas the GNN embedded method demonstrates strong performance with an AUC of 0.98, Recall of 1.00, Precision of 0.95, and F-measure of 0.98. Moving on to case 3, our proposed method achieves an AUC of 0.80, Recall of 0.60, Precision of 1.00, and F-measure of 0.75, while the GNN embedded method obtains an AUC of 0.55, Recall of 1.00, Precision of 0.53, and F-measure of 0.69. Finally, in case 4, our proposed method attains an AUC of 0.93, Recall of 0.85, Precision of 1.00, and F-measure of 0.92, whereas the GNN embedded method achieves an AUC of 0.50, Recall of 0.00, Precision of 0.00, and F-measure of 0.00.

To sum up, either DT or RF models trained using EE features can detect faults in a larger number of cases with higher AUC values ($AUC > 0.60$) compared to those trained using GNN embedded features. This suggests that the proposed approach is effective in facilitating cross-dataset building fault detection.

V. CONCLUSIONS AND FUTURE WORK

In this study, a new method for extracting entropy features from graph-structured data is introduced to facilitate the development of cross-dataset building AFDD. The approach involves calculating EE values from the graph structures of both simulated and real datasets. Subsequently, machine learning models for AFDD (e.g., decision tree and random forest) are trained using the EE values from simulated data and then tested on EE values from the real data. To assess the effectiveness of the proposed method, four distinct fault scenarios (consistent between simulated and real datasets) are examined under faulty and fault-free conditions. The experimental findings indicate that the features extracted by our method from simulation data can notably enhance fault detection performance in real-world building fault scenarios. Additionally, we utilize GNN embedded features as a baseline for comparison with our algorithm in the same classification task. Our algorithm outperforms the baseline in most of the fault cases, showcasing the effectiveness and generalizability of our approach for analyzing building HVAC systems across different datasets.

It is important to note that this study focuses solely on two common types of faults in building HVAC systems, with future plans to explore a broader range of fault types, including those affecting return fans or supply fans.

ACKNOWLEDGMENT

This research is supported by funds from the National Science Foundation award under the grant number 2309030 entitled "PIRE: Building Decarbonization via AI-empowered District Heat Pump Systems".

REFERENCES

- [1] United Nations Environment Programme, "2022 Global Status Report for Buildings and Construction: Towards a Zero-emission, Efficient and Resilient Buildings and Construction Sector," Nairobi, 2022.
- [2] L. Pérez-Lombard, J. Ortiz and C. Pout, "A Review on Buildings Energy Consumption Information," *Energy Build.*, vol. 40, no. 3, pp. 394–398, 2008.
- [3] Energy Conservation in Buildings and Communities Programme, "Real Time Simulation of HVAC Systems for Building Optimisation, Fault Detection and Diagnostics," International Energy Agency, Paris, France, Rep. No.: IEA ECBCS Annex 25, 1996.
- [4] H. Kramer, G. Lin, C. Curtin, E. Crowe, and J. Granderson, "Building analytics and monitoring-based commissioning: Industry practice, costs, and savings," *Energy Effic.*, vol. 13, pp. 537–49, 2020.
- [5] S. Katipamula and M.R. Brambley, "Review article: methods for fault detection, diagnostics, and prognostics for building systems—a review, part I," *HVAC&R Res.*, vol. 11, no. 1 pp. 3–25, 2005.
- [6] S. Frank, X. Jin, D. Studer and A. Farthing, "Assessing barriers and research challenges for automated fault detection and diagnosis technology for small commercial buildings in the United States," *Renew. Sust. Energ. Rev.*, vol. 98, pp. 489–499, 2018.
- [7] Z. Chen, Z. O'Neill, J. Wen, O. Pradhan, T. Yang, X. Lu, et al., "A review of data-driven fault detection and diagnostics for building HVAC systems," *Appl. Energy*, vol. 339, pp. 121030, 2023.
- [8] L. Zhang, M. Leach, J. Chen and Y. Hu, "Sensor cost-effectiveness analysis for data-driven fault detection and diagnostics in commercial buildings," *Energy*, vol. 263 pp. 125577, 2023.
- [9] G. Li, L. Ren, Y. Fu, Z. Yang, V. Adetola, J. Wen, et al. "A critical review of cyber-physical security for building automation systems," *Annu. Rev. Control*, vol.55, pp. 237-254, 2023.
- [10] X. Kang, X. Wang, J. An and D. Yan, "A novel approach of day-ahead cooling load prediction and optimal control for ice-based thermal energy storage (TES) system in commercial buildings," *Energy Build.* vol. 275, pp. 112478, 2022.
- [11] Y. Ye, C. A. Faulkner, R. Xu, S. Huang, Y. Liu, D. L. Vrabie, J. Zhang, and W. Zuo, "System modeling for grid-interactive efficient building applications," *J. Build. Eng.*, vol. 69, pp. 106148, 2023.
- [12] A. Malkawi, S. Ervin, X. Han, E. X. Chen, S. Lim, S. Ampanavos, and P. Howard, "Design and applications of an IoT architecture for data-driven smart building operations and experimentation," *Energy Build.* vol. 295, pp. 113291, 2023.
- [13] J. Huang, J. Wen, H. Yoon, O. Pradhan, T. Wu, Z. O'Neill, and K.S. Candan, "Real vs. simulated: questions on the capability of simulated datasets on building fault detection for energy efficiency from a data-driven perspective," *Energy Build.*, vol. 259, pp. 111872, 2022.
- [14] J. Huang, H. Yoon, T. Wu, K.S. Candan, O. Pradhan, J. Wen, and Z. O'Neill, "Eigen-Entropy: A metric for multivariate sampling decisions," *Inf. Sci.*, vol. 619, pp. 84-97, 2023.
- [15] W. Liang, M. Lv, and X. Yang, "Development of a Physics-Based Model for Analyzing Formaldehyde Emissions from Building Material under Coupling Effects of Temperature and Humidity," *Build. Environ.*, vol. 203, pp. 108078, 2021.
- [16] J. Granderson, G. Lin, Y. Chen, A. Casillas, P. Im, S. Jung, K. Benne, J. Ling, R. Gorthala, J. Wen, Z. Chen, S. Huang, and D. Vrabie, "LBNL Fault Detection and Diagnostics Datasets," United States. [Online]. Available: <https://dx.doi.org/10.25984/1881324>
- [17] J. Wen and S. Li, "Tools for evaluating fault detection and diagnostic methods for air-handling units," ASHRAE Research Project, 2011.
- [18] Y. Chen, J. Wen and L. J. Lo, "Using weather and schedule based pattern catching and feature based PCA for whole building fault detection — part I development of the method," *ASME J. Eng. Sustain. Build. Cities*, vol. 3, no. 1, pp. 011001, 2022.
- [19] G. Lin, H. Kramer and J. Granderson, "Building fault detection and diagnostics: achieved savings, and methods to evaluate algorithm performance," *Build. Environ.*, vol. 168, pp. 106505, 2020.
- [20] D. K. McClish, "Analyzing a portion of the ROC curve," *Med. Decis. Making*, vol.9, pp. 190-195, 1989.
- [21] S. Yang and G. Berdine, "The receiver operating characteristic (ROC) curve", *The Chronicles*, vol. 5, no. 19, pp. 34-36, 2017.