An Information-theoretical Framework for Data-driven Building Automatic Fault Detection and Diagnosis Support

by

Jiajing Huang

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved April 2024 by the
Graduate Supervisory Committee:

Teresa Wu, Chair
Kasim Selcuk Candan
Jin Wen
Zheng O'Neill
Giulia Pedrielli

ARIZONA STATE UNIVERSITY

August 2024

ABSTRACT

Buildings are complex and integrated systems consisting of multiple sensors, subsystems, and automatically controlled components, among which the heating, ventilation, and air conditioning (HVAC) systems are critical for building energy consumption and indoor environment quality. HVAC systems usually suffer from faults, such as malfunctioning sensors, equipment, and control systems, which significantly affect a building's performance. Automatic fault detection and diagnosis (AFDD) tools have shown great potential in improving building energy efficiency and indoor environment quality. With the rapid advancement of internet-of-things (IoT) and sensor techniques, data-driven AFDD approach has drawn increased attentions due to its ease of implementation and low costs while achieving high fault detection and diagnosis performances. While promising, there are certain challenges, including reliable baseline constructions, issues of the use of building simulation for fault detection, and efficient fault-to-symptom root cause analyses, that hinder the performances of data-driven AFDD in a real, whole building system.

The goal of this dissertation is to develop an information-theoretical framework for data-driven building automatic fault detection and diagnosis support. The first phase is to propose a decision-making metric, termed Eigen-Entropy (EE), to support the baseline construction for building AFDD by sampling from historical normal datasets. The second phase is to extend the use of EE to extract features from graph-structured data obtained from simulation building data to realize cross-datasets (simulation-to-real) fault detection improvements. The third phase is to utilize EE with causal inference to construct Bayesian Networks by determining the fault-to-symptoms relationships in the building systems. Experiments have shown the proposed information-theoretical framework has satisfactory capabilities, efficiency and efficacy for building AFDD.

DEDICATION

To my dear family, who always support me unconditionally

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my advisor, Dr. Teresa Wu, for her continuous guidance throughout my doctoral journey. Dr. Wu stands out as an exceptional mentor and leader in research, characterized by her empathy, patience, motivation and enthusiasm. The way she approaches her work, her research mindset, and her life lessons will have a lasting and significant impact on me, shaping me into a better individual and scholar. It is my fortune to have her as both an advisor and life coach for my Ph.D. journey.

Next, I would like to express my sincere gratitude to my committee members who are also my research collaborators: Dr. K. Selcuk Candan, Dr. Jin Wen, Dr. Zheng O'Neill, and Dr. Giulia Pedrielli. Their valuable insights, and thoughtful guidance throughout my dissertation have been truly invaluable. Their encouragement and the opportunities they provided me to participate in significant, interdisciplinary, and stimulating projects have been remarkable. It is my great honor to have such esteemed researchers on my supervisory committee.

My sincere thanks also go to my collaborators who have helped me conduct this research. They are Dr. Ojas Pradhan and Naghmeh Ghalamsiah from Drexel University; Dr. Zhiyao Yang, Guowen Li and Mengyuan Chu from Texas A&M University; Ahmet Kapkic from Arizona State University.

Additionally, I extend my heartfelt thanks to Dr. Rong Pan, my Data Science, Analytics and Engineering (DSAE) program chair, . As a first Ph.D. student in this newly-launched program, I have received a lot of attentions and consistent supports from him.

Thank you to all past and present members of AMCII, friends and colleagues for your accompany, help and guidance: Dr. Hyunsoo Yoon, Dr. Hope Lancaster, Dr. Yinlin Fu, Dr. Fei Gao, Dr. Nathan Gaw, Dr. Xiaonan Liu, Dr. Yanzhe (Josh)

Xu, Zhiyang Zheng, Suryadipto Sarkar, Firas AI-Hindawi, Md Mahfuzur Rahman Siddiquee, Jay Shah, Fulin Cai, Teng Li, You Zhou, Abhidnya Patharkar, Matt Liljenstolpe, Dr. Yiming Che, Amogh Joshi, Asra Aghaei, Fazle Rafsani, Devam Sheth; Dr. Zhe Gao, Dr. Yunyi Kang, Dr. Tingnan Lin, Dr. Jian Zhou, Dr. Xinyu Zhao, Dr. Menghan Liu, Dr. Ting Yan (Betrica) Fok, Dr. Tiankai Xie, Yaoxin Zhuo, Yichuan Li, Weiying Wang, Shu Wan, Yuze Liao, Hao Mei. I would also like to thank my best friend and high school mate, Wenlong Qiu, and my undergraduate advisor, Dr. Haixia Qian, for their continuous encouragement.

Lastly, I would like to extremely acknowledge my parents, Kening Huang and Huibing Peng, my sisters, Yisi Huang and Yidie Huang, my brothers-in-law, Yang Zhou and Junyi Liu, my newphews, Yifan Zhou and Xuanyu Zhou, my grandmother, Xuehua Cai, my aunt's family, Xiaofei Peng, Guozhen Ye and Dongping Ye, and other family members; my good friend, Linda Csellak, my elementary school teacher, Xiaoqin Wang, and her family members, Richard King and Zhige Song for supporting me unconditionally and spiritually throughout my life.

This dissertation is based on work supported by the U.S. Department of Energy (#DE-EE0009150: Securing Grid-interactive Efficient Buildings (GEB) through Cyber Defense and Resilient System (CYDRES)), the U. S. National Science Foundation (PFI-RP #2050509: Data-Driven Services for High Performance and Sustainable Buildings) and the U. S. National Science Foundation (PIRE #2309030: Building Decarbonization via AI-empowered District Heat Pump Systems).

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Chapter 1

INTRODUCTION

1.1 Background

Buildings are complex and integrated systems consisting of multiple sensors, subsystems, and automatically controlled components, among which the heating, ventilation, and air conditioning (HVAC) systems are critical for building energy consumption and indoor environment quality. The HVAC systems have been reported to be responsible for 20% of building energy consumption (Pérez-Lombard et al., 2008), and such consumption accounts for 36% of global final energy use and 39% of energy-related carbon dioxide emission, according to the United Nations Environment Programme (Global Status Report 2018). However, among this primary energy use, 30% is wasted because of operation faults and malfunctions in the HVAC systems (Katipamula and Brambley, 2005a), and in the United States, key faults in building systems are estimated to result in additional 0.37 to 17.96 exajoule energy consumption annually (Roth et al., 2004). Studies have shown that automatic fault detection and diagnosis (AFDD) in HVAC systems provides great potential for energy savings (Roth et al., 2004). AFDD is the process including fault detection, identification, and isolation and it is able to achieve annual 10% median energy savings with two-year simple payback periods in the United States. Here faults are typically defined as deviations from normal operating conditions in a building system (Katipamula and Brambley, 2005a).

In general, AFDD methods can be grouped into two categories: (1) qualitative and quantitative model-based, and (2) process history-based (Katipamula and Bram-

bley, 2005a,b; Kim and Katipamula, 2018). Traditional qualitative and quantitative model-based approaches including rule-based and physics-model based, are familiar to building engineers and researchers mainly because these methods are interpretable from building physical perspective, but their drawbacks are low generalizability and high deployment costs because there is a need of customizations on the physics-based modeling, rules and thresholds for individual building systems; consequently, the market adoption rate for these methods are relatively low (Frank et al., 2018). Process history-based approaches (e.g., data-driven and machine learning based), on the other hand, can overcome the issues of low scalability and high implementation costs. With the rapid developments of data science and machine learning techniques, and the wide adoption of building automation systems (BAS) or other sensor technologies, data-driven AFDD has gained increased attention. Different from traditional qualitative and quantitative approaches commonly found in commercial-off-the-self AFDD products, data-driven AFDD does not require expert knowledge and heuristics as a priori, and thus has the potential for low costs while achieving high fault detection and diagnosis accuracy (Hu et al., 2021).

Many articles have concentrated on data-driven AFDD methods developed for component level and whole building level in recent years. Han et al. (2023) develop a data-informed framework for building energy managements showing great potential for whole building fault identifications; Malkawi et al. (2023) propose an IoT, data-driven-based architecture to monitor building normal operations. Zhang et al. (2023) study a sensor cost analysis workflow to quantify the economic implications of installing new sensors for AFDD using the concept of sensor threshold marginal cost. Chen et al. (2023) conduct an extensive review of data-driven AFDD methods covering from the process to the evaluation metrics; Li et al. (2023) provide a systematical review of data-driven AFDD focusing on the vulnerabilities in BAS and strategies for

cyber resilient control; Zhang et al. (2021) offer a comprehensive review of machine learning models for AFDD support in terms of building load prediction, and further (Zhang et al., 2023) conduct an insightful analyses on challenges and opportunities of machine learning control to support building AFDD.

Although data-driven AFDD is prevailing, there are challenges that hinder its effectiveness and efficiency in the field of building engineering:

- Challenge I: Reliable fault detection baseline construction

  No matter what AFDD methods (including physics-model based, rule-based or data-driven) are, a baseline is needed to enable fault detection. A baseline, generated from rules or collected building data to represent normal building operation conditions, depicts the normal operational conditions because an exact fault-free status may rarely exist in a real building (Chen, 2019). Note that the most AFDD building research uses the term "normal" rather than "fault-free" to reflect a satisfied building operation, since a normal status typically exists after a building is freshly commissioned. Here "historical normal data" are denoted as those collected for baseline construction purposes, such as those data collected right after a building is thoroughly commissioned.

  There are existing studies focusing on baseline generation for the whole building AFDD. Lin and Claridge (2015) propose a temperature-based approach that constructed the baseline by considering cooling and heating energy consumptions during the post-commissioning period. Miller et al. (2015) introduce a day-typing process using Symbolic Aggregate ApproXimation (SAX) to extract and cluster the most common daily profiles to create a baseline. Fan et al. (2015) present a time series-based data mining to respond to dynamic building data by integrating SAX, motif discovery and temporal association rule. Chen and Wen

(2017) further extend the SAX model to SAX based weather pattern matching model (SAX-WPM) to construct a baseline for the whole building systems by clustering time-series with the same weather motif. Li et al. (2010) develop an energy prediction model for baseline construction by removing data outliers and infrequent energy abnormalities that existed in historical data. However, it is noted the research to-date heavily relies on physics-based models as well as building domain knowledge, when constructing a baseline model. In addition, existing work has all taken trial-and-error approaches. In other words, given a dataset, evaluation is required to assess the effectiveness of the dataset. Often, multiple iterations are needed to identify the appropriate dataset as a baseline. When being applied in a real building, such approaches are not able to answer the following question: *how many and what samples should be included in the baseline.* **It is concluded there lacks a metric that can support the sampling decision-making using historical normal datasets to construct reliable baselines.**

- Challenge II: Application of building simulation for real-world fault detection

  Most data-driven AFDD strategies reported in the literature are developed and evaluated using simulated system data, primarily because obtaining real building data for analysis is a real challenge (Li and O'Neill, 2018; Shi and O'Brien, 2019). Implementing faults in real buildings and collecting data that reflect the impacts of these faults is practically difficult. It is even more arduous to establish reliable "ground truth" by cleaning and analyzing real building data since unexpected, naturally occurring system faults can lead to anomalies, complicating or sometimes even obscuring the effects of artificially injected faults. Simulations have thus been widely adopted in building and other energy-related

engineering domains due to the ease and lower performing costs (Ye et al., 2023). For example, Kang et al. (2022) apply simulation modeling for cooling load prediction and optimal control for ice-based thermal energy storage systems in commercial buildings; Jiang et al. (2023) utilize simulation to validate the use of various machine learning models with a customized loss function for building energy consumption prediction. Granderson et al. (2020) apply simulation tools, experimental test facilities and in-situ field operation to generate building AFDD data, and further expand the dataset (Granderson et al., 2023) covering a wider range of fault severity in different scenarios to support for diagnostic algorithm development and performance testing.

While the use of simulation in building research is prevailing, AFDD strategies trained by simulated system data may not achieve satisfactory effectiveness when applied to real building systems. Huang et al. (2022) show that AFDD models using simulated building data fail to capture the fault symptoms occurred in the real building systems. It is concluded that significant differences of measurement data quality and characteristics exist between simulation and real building systems from the data-driven perspective. This also suggests that measurements from simulation system, even after careful calibration, may contain information that differs from real-world scenarios. As simulations rely on real-world data and are thus bound by physical laws, there must exist some patterns among various building components that may remain consistent between real-world and simulated datasets. Therefore, **it is needed to identify such similar patterns to support the use of simulation for real building fault detection.**

- Challenge III: Bayesian networks construction for cross-level fault diagnosis

  Modern building HVAC systems typically include multiple, highly coupled subsystems such as cooling/heating plant, primary air distribution, and terminal air distribution subsystems. Due to the coupling effect among building components, a fault occurring in one equipment or subsystem may propagate and influence other equipment or subsystems (Yan et al., 2017; Cauchi et al., 2018). Hence, component-level AFDD methods may not be efficient and suitable solutions to the root cause analysis for cross-level faults, i.e., faults causing adverse effects across multiple components and subsystems (Chen et al., 2022). Chen et al. (2022) provide an example of a chiller supply water temperature sensor bias fault (e.g., sensor reading higher than actual temperature) in the chiller plant which may cause the cooling valve open position in a downstream air handling unit (AHU) to be lower than normal. In this case, a component-level AFDD tool that only monitors the AHU might result in false alarms such as a cooling coil valve fault or a supply air temperature sensor fault. Hence, a root cause analysis is necessary to ensure the correct diagnosis of cross-level faults.

  Compared with fault detection studies, much fewer fault diagnosis/root cause analysis studies exist (Chen et al., 2023). A root cause analysis has shown its importance to improve quality assurance, reliability and performance (Wang et al., 2018). Bayesian networks (BNs) have been extensively studied as a root cause analysis technique. For example, Wang et al. (2018) conduct an analysis on root causes of occurring alarms in thermal power plants based on posterior probability from BN. In their research, the BN is constructed by one child node and multiple parent nodes that describe the relationship between an alarm variable and root-cause variables using the process knowledge. Lokrantz et al. (2018)

propose a BN-based graphic probabilistic models using the expert knowledge to identify the causality of failure and quality deviation among multiple manufacturing stages, where network parameters are trained by historical data, and root cause is inferred according to defect types and measurements. Liu et al. (2022) develop a strong relevant mechanism BN combining process mechanism analysis with historical data mining for un-monitored root cause variables in chemical plants fault diagnosis, which showed great practicability and satisfactory performances in fault propagation recognition. Amin et al. (2021) present a hybrid data-driven method integrating principal component analysis with the Bayesian networks for fault detection and diagnosis in process plants, which demonstrated a strong efficacy of diagnosis performance while maintaining lesser false diagnosis. The same authors (Amin et al., 2019) develop a dynamic Bayesian network-based fault detection and root cause diagnosis, which has an ability to convert the continuous process data into meaningful evidence instead of a probabilistic domain. There are also some BN studies emphasizing cross-level fault diagnosis in an HVAC system. For example, Chen et al. (2022) propose a whole building fault diagnosis method based on Discrete BN to isolate faults causing significant abnormalities in multiple subsystems/equipment during system operation, and further designed a weather and schedule-based pattern matching Discrete BN to diagnose cross-level faults in building HVAC systems for real-time fault diagnosis and isolation. Pradhan et al. (2021) develop a dynamic BN-based approach that incorporated the temporal dependencies of fault nodes between time steps using temporal conditional probabilities to improve accuracy for a whole building level fault diagnosis.

The aforementioned BN-based diagnosis methods highly rely on heuristics processes to learn causal relationships among fault status and symptoms, and their

causal mechanism is primarily determined by the expert knowledge. While promising, heuristics processes by the domain knowledge may not be adequate and effective for the fault diagnosis in more complex buildings, especially those with multiple coupled subsystems. Other than being labor intensive, these approaches may not discover underlying coupling effects among the subsystems comprehensively. Thus, **there is a need of a data-driven approach to interrogate the fault-symptom causalities to construct the BN structure.**

As is observed from the above-mentioned challenges, the key deficiency of current data-driven AFDD methods in the building systems still involve the expert domain knowledge as a priori. Consequently, data-driven AFDD methods based on expert knowledge is labor intensive. When facing with those commercial buildings with medium or large sizes, these approaches may become unrealistic to implement. In fact, data collected from BAS sensor readings contain important characteristics reflecting building conditions, and data collected from buildings under faulty conditions contain important information different from those under normal conditions, and such information can be quantified by the information entropy. Moreover, since building BAS data consist of various time-series collected from multiple sensors and components, an information quantification method on multi-variate time-series may be one viable solution. The goal of this dissertation is to develop the information-theoretical framework to address aforementioned challenges to support building AFDD from data-driven perspective.

1.2 Research Objective and Contributions

The overall objective of this dissertation is the development of a novel framework based on an information-theoretical metric to tackle these challenges. Specifically, the proposed information-theoretical metric, termed Eigen-Entropy, is to realize the information quantification on multivariate time-series. Consequently, the contributions in this dissertation include:

- **An Eigen-Entropy-based sampling framework for AFDD baseline construction is proposed.** The proposed method uses Eigen-Entropy (EE), defined through eigenvalues from a correlation magnitude matrix on multiple building sensor measurements as a decision-making criterion. Contributions come out of this method: (1) It is able to measure the data heterogeneity and thus automatically determine how many samples and what samples from the historical dataset are to be included in the baseline that provides sufficient information for building AFDD; (2) The use of stopping criterion make it have the potential to construct a baseline in real-time for online AFDD implementation.

- **An Eigen-Entropy-graph-based feature extraction approach for cross-dataset (simulation-to-real) fault detection in building HVAC systems is proposed.** The proposed method includes acquiring graph structures from simulated building data, extracting their Eigen-Entropies as features to train AFDD models, and 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. Contributions come out of this method: (1) This is first time to use the graph techniques with information theory to support the building fault detection; (2) The extracted features based on the graph and information entropy are transferable from simulation to the real-scenario datasets, enhancing

the cross-dataset detection performances.

- **An Eigen-Entropy-based causality framework for cross-level faults diagnosis and isolation in building HVAC systems is proposed.** The proposed method utilizes causal inference to determine the causal mechanisms between faults status and symptoms to construct the BN model, where Eigen-Entropy is particularly used for characterizing synchronicity, which describes the trends of movements among the symptoms over the time. Contributions come out of this method: (1) a term synchronicity, measured by Eigen-Entropy, is defined, and able to capture the interactions, i.e., the trends of aligned movements over time, among multiple symptoms under fault status; (2) it utilizes cause effect estimations (counterfactual inferences) to induce the causal structure between faults and synchronicity among symptoms.

1.3 Dissertation Organization

The proposed dissertation research will be presented in the following chapters. Chapter 2 presents literature review. Chapter 3 presents the proposed Eigen-Entropy and its fundamental concepts. Chapter 4 presents the development of topic (I): Eigen-Entropy for building fault detection baseline construction. Chapter 5 presents the development of topic (II): Eigen-Entropy for cross-dataset fault detection. Chapter 6 presents the development of topic (III): Eigen-Entropy for fault diagnosis Bayesian network construction. Chapter 7 draws the conclusions and future work.

Chapter 2

LITERATURE REVIEW

As discussed in the introduction, there are three main challenges in building data-driven AFDD, namely (I) reliable fault detection baseline construction; (II) applications of building simulation for real-world fault detection; and (III) Bayesian networks construction for cross-level fault diagnosis. In fact, from the data-driven perspective, the first challenge can be viewed as a sampling problem, the second one can be viewed as featrue extraction, and the third one can be viewed as causal structure learning. Consequently, this chapter presents some literature related to sampling methods, graph feature extraction, and causal structure learning.

2.1 Sampling Methods

AFDD baseline construction is indeed a sampling problem. Sampling is in general a statistical procedure concerning the selection of a subset of individual observations to capture characteristics of the whole population for different applications (Fuller, 2009). If done properly, sampling can save time and cost while supporting statistical inferences (Fuller, 2009). There are in general two categories of sampling approaches: probability and non-probability sampling (Fuller, 2009). (a) In *probability sampling*, each observation from the population is assigned with a certain probability of selection and is chosen by incorporating a random mechanism. Some common probability sampling methods are simple random sampling, stratified sampling, cluster sampling, and systematic sampling (Berndt, 2020; Fuller, 2009). *Simple random sampling* assigns each observation with an equal probability of being sampled. *Stratified sam-*

pling divides the whole population into several subgroups termed strata; then, each observation within a stratum will be selected randomly, and selected observations across the strata become samples. *Cluster sampling* aggregates observations from the population into larger units, called clusters. Samples are then randomly selected from the clusters. *Systematic sampling* selects members from a list of population members according to a random starting point and at fixed periodic intervals. (b) In *non-probability sampling*, samples are selected subjectively and purposedly. This includes availability sampling, purposive sampling, quota sampling, and respondent-assisted sampling (Berndt, 2020). *Availability sampling* is a procedure in which samples are selected from a target population based on availability, self-selection, and/or at the discretion of the researchers, while *purposive sampling* selects samples that fit and meet the purpose of the study and specific criteria for inclusion/exclusion. *Quota sampling* realizes sample collection by combining availability sampling and purposive sampling to target specific numbers of observations with characteristics of interest, while *respondent-assisted sampling*, or *snowball sampling*, selects samples regarding previously selected observations in the population. It is noted that, unlike probability sampling, this set of sampling methods does not have an explicit stochastic process involved, and mostly relies on subjective judgment.

Sampling methods provide a guideline on which and how many samples must be selected as representatives to guarantee the generalization of study conclusions. Multiple sampling strategy factors include research objective, methodology, definition, and nature of the population, as well as the availability of resources and degree of confidence in generalized conclusions (Fuller, 2009). When comparing probability vs. non-probability sampling (Berndt, 2020), it is noted that probability sampling is generally preferred in studies requiring confirmatory purposes, quantitative design with a heterogeneous population, representative and unbiased samples capturing es-

sential characteristics of the population, and statistical inferences from the samples. Non-probability sampling is favored when studies are exploratory or descriptive, a qualitative research design is required without the need for statistical inferences or representative samples, or a sampling frame is not available. The focus of this research is on probability sampling, which has been widely used in different fields of study, including machinery safety (Hajar et al., 2019), geology (Brus and van den Akker, 2018), railroads (Chen and Liu, 2019), and building engineering (Chen, 2019).

Existing probability sampling research is mostly model-based. That is, they either assume a known distribution as a prior or rely on a specific model to extract probability parameters. One example is active learning, a machine learning methodology that selects samples to be annotated for training to reduce the labor-intensive efforts of manual annotations (Settles, 2009). Hajar et al. (2019) present two discrete random sampling strategies, additive random sampling (ARS) and jittered random sampling (JRS), for machine monitoring. Both ARS and JRS sampling show potential for simplified implementation in a remote application having a low frequency rate while maintaining easy real-time operation management (Hajar et al., 2019).

On the other hand, model-free probability sampling takes a different approach and divides the dataset into subgroups to guide the data sampling decisions. For example, Brus and van den Akker (2018) utilize a stratified sampling survey to analyze the seriousness of subsoil compaction problems in the Netherlands. In their study, stratification is accomplished by a map showing five levels of subsoil compaction risks, and stratified sample data are used to estimate areal fraction, an indicator of over-compactness in the subsoil. Chen and Liu (2019) propose a high-dimensional clustering-based stratified sampling (HDCSS) method for roadway asset condition inspection, which yields a relatively smaller number of samples, potentially leading to inspection cost savings. While promising, it is noted that cluster-based sampling

may suffer from increased sampling errors when the base cluster selected already has biases (Berndt, 2020). Stratified sampling may also be challenging when there exist no stratifiable structures in the dataset (Berndt, 2020). In addition, both stratified and cluster sampling require subgroups to be identified first before the sampling.

As reviewed above, both model-based and model-free sampling approaches require pre-processing to either derive the distribution, estimate the probability parameters, or identify the subgroups. There is a need of a sampling decision-making metric without extensive pre-processing. One direction is the use of entropy. Entropy is an information-theoretic measurement to quantify information richness (Shannon, 1948) and has been used as a decision criterion for different applications. For example, Wang and Yao (2016) propose nonlinear correlation information entropy (NCIE) based on Pearson's correlation coefficient to remove redundant objectives in many-objective optimization problems (MaOP). Xia and Liu (2020) extend NCIE to a supervised learning algorithm to determine features for synthetic aperture radar (SAR) image recognition. Wang et al. (2020) propose differential correlation information entropy (DCIE) for feature selection in classification problems.

To the best of my knowledge, there is few research on using entropy for sampling decisions. Although researchers (Settles, 2009) explore the use of entropy in active learning, as mentioned earlier, active learning requires distribution information to be drawn a priori. Some examples include studies by Rossini et al. (2020) who used entropy to smoothen time series data, Li et al. (2020) who used entropy-based oversampling (EOS) methods in imbalance learning, Salehi et al. (2021) who used relative entropy for semi-supervised section measurement, and Xu et al. (2021) who used cross-entropy based noise correction for data and model quality improvement in crowdsourcing; Yet, these entropy-based approaches under specific distributions may not be suitable for high-dimensional problems (Geyer et al., 2019).

2.2 Graph Feature Extraction

Graph-structured data are common types in real-world scenarios. Graph-structured data refers to a type of data representation where entities are represented as nodes, and the connections between these nodes are represented as edges. Compared to tabular data, graphs are able to provide insightful information about how different features are interconnected and influence each other. Due to their capabilities for characterizations on complex feature dependencies, graphs have been widely used in various fields such as telecommunication network design (Liu et al., 2022), computer networks efficiency optimization (Khan and Javaid, 2022), and social media analysis (Mao et al., 2023). As a result, feature extraction is needed and important to support the use of graphs for various purposes (classification, regression, etc).

Graph feature extraction approaches can be either machine learning (ML) based or deep learning (DL) based. Commonly used ML-based feature extraction approaches on graph include independent component analysis (ICA) and clustering. For example, Huang et al. (2022) utilize ICA to investigate the specific features related to ischemic vascular disease by means of brain graphs; Han et al. (2022) develop a graph combined clustering independent component analysis to support machine-to-machine communications; Li et al. (2022) present a scalable and parameter-free graph fusion framework to seek for a joint graph compatible across multipleview clusters. Besides, other recent works have emphasized on the feature engineering on graphs. For example, Xiong et al. (2024) propose an adaptive graph-based feature normalization for facial expression recognition; Zare and Nouri (2022) study the flow noise visibility descriptor via graph-based feature extractions; Yang et al. (2022) develop graph-based feature extraction framework to support rotating machinery diagnosis.

There are many recent works focusing on the DL-based graph feature extrac-

tion approaches, including convolutional neural networks (CNN) and graph neural networks (GNN). For example, Yu et al. (2023) utilize dynamic graph embedding method to transforming features extracted by CNN into graph-structured forms to support pattern recognition for rolling bearing fault diagnosis. Abrate and Bonchi (2021) apply three different DL-based feature extractions on brain graph-structured data to study Autism spectrum disorder and Attention-deficit/hyperactivity disorder; Kang et al. (2022) design a GNN based on link representation to obtain node embedding for potential molecular associations identification.

Lately, there have been studies focusing on utilizing information entropy for extracting features from graphs. For instance, Yang et al. (2023) introduce a method based on maximum mutual information to extract features from graph-structured data for analyzing Alzheimer's disease; Xu et al. (2021) devise a novel deep second-order Rényi entropy graph kernel to handle larger graphs effectively, thereby overcoming the size constraints commonly encountered in graph kernels; Additionally, Aziz et al. (2021) propose a technique for estimating the entropy of intricate networks by assuming they are undirected and unlabelled graphs, and they investigate its applications using an information functional that is applicable to graphlets of varying sizes. However, the entropy extraction discussed in these approaches still relies on specific assumptions (e.g., Gaussian distribution) or models (e.g., Gaussian mixture model), which could introduce bias in the features extracted or not fully capture the characteristics of the graph structure.

2.3 BN and Causal Structure Learning

Building fault diagnosis in nature is a problem of root cause analysis. One notable emerging field for RCA support is causal learning (CL). CL uses the observational data to learn causality and thus provides new opportunities to address domain specific challenges for cause-to-effect identifications (Cheng et al., 2022). In general, CL research focuses on two categories (Schölkopf, 2022): (1) causal effects estimation; and (2) causal structure learning. Causal effects estimation is to investigate how much changing one variable will influence another given a causal structure assumption between these two variables. This can be done by the counterfactual inference (Pearl, 1988, 2009) which assesses the strength of causality between two events by inferring the likelihood of one event not occurring when another is absent. Certain works have focused on counterfactual causal effects estimation. For example, Sjölander (2021) addresses marginal counterfactual means estimation by linear and log-linear fixed effects models in the presence of clustered confounders; Porcher et al. (2019) develop a multinomial counterfactual modeling to estimate monotone treatment effects for patients receiving radiotherapy; Samoilenko et al. (2020) apply closed-form regression-based and marginal structural model-based approaches to estimate the causal mediation effects of maternal inhaled corticosteroids doses on birthweight; Miles et al. (2019) present an approach for direct and overall effects estimation under certain interventions on health clinic studies.

Causal structure learning , on the other hand, is to induce the structure describing the causal relationships from variables to others, and BN is one of the prevailing causal structure learning tools as it has shown the ability to represent the probabilistically conditional independence in a graph model, providing an efficient and expressive way for knowledge representations and acquirements (Jiang et al., 2019). Hunte

et al. (2022) develop BN approach for systematic product safety and risks assessment; Lee et al. (2018) employ a BN-based model adopting domain knowledge to study readmissions reduction for chronic obstructive pulmonary disease patients; Giovanis (2019) apply BN to examine the causal effect of income and air pollution on life satisfaction; Yoo et al. (2022) propose a causal BN algorithm that help to provide a better mechanistic understanding of big data through an order search.

As previously discussed, there is a clear connection between causal effects and BN structures. While BN structures and observational data can be used to estimate causal effects, these estimations can also serve to validate the reliability of BN structures. Limited research has been conducted on constructing causal structures based on estimating causal effects from observational data, particularly in the context of BN construction for the cross-level fault diagnosis in the whole building systems. Previous studies in this area have heavily relied on the expertise of domain specialists. Moreover, existing research on BN-driven fault diagnosis has predominantly assumed that components and systems in buildings operate independently (Chen et al., 2022,a,b); however, this assumption may not hold in real-world situations where building components are interdependent. These interdependencies, or coupling effects, among components are important characteristics and may ensure the robustness of the BN structure for supporting cross-level whole building fault diagnosis. However, very few studies have explored the causal relationships leading from building faults to coupling effects. While some recent algorithms like the MI-Kruskal-K2 algorithm (Li et al., 2022) offer data-driven approaches for learning BN structures, they also hold similar assumptions on independence among non-descendant nodes given a parent node within the BN structure. Consequently, these approach again may not capture the component-wise coupling effects in the whole building systems when they are applied to BN-based fault diagnosis strategy.

2.4 Summary

As reviewed above, it is noticed that the common deficiencies existing in sampling methods, graph feature extraction and BN structure learning include relying on specific data distribution assumptions, specific modelings or domain knowledge. When facing with high-dimensional data with more complex structure, these methods may be labor-intensive and not meet the need of efficient data-driven building AFDD. Consequently, here is to introduce a metric Eigen-Entropy (EE), which is based on eigenvalues derived from a correlation coefficient matrix, to address these deficiencies. In the following chapters, I will provide details about this metric, and show how this metric works to overcome each challenge in the data-driven building AFDD.

Chapter 3

EIGEN-ENTROPY: AN INFORMATION ENTROPY FOR MULTIVARIATE

3.1 Background

Entropy is an information-theoretic measurement to quantify information richness (Shannon, 1948) and has been used as a decision criterion for different applications. For example, Wang and Yao (2016) propose the concept of nonlinear correlation information entropy (NCIE) based on Pearson's correlation coefficient to remove redundant objectives in many-objective optimization problems (MaOP). Xia and Liu (2020) extend NCIE to a supervised learning algorithm to determine features for synthetic aperture radar (SAR) image recognition. Wang et al. (2020) propose differential correlation information entropy (DCIE) for feature selection in classification problems. For example, to the best of my knowledge, research on using entropy for sampling decisions is limited. Although researchers in (Settles, 2009) explore the use of entropy in active learning, as mentioned earlier, active learning requires distribution information to be drawn a priori. Some examples include studies by Rossini et al. (2020) who use entropy for locally robust decision on time-series smoothing, Li et al. (2020) who use entropy-based oversampling (EOS) methods in imbalance learning, Salehi et al. (2021) who use relative entropy for semi-supervised section measurement, and Xu et al. (2021) who use cross-entropy based noise correction for data and model quality improvement in crowdsourcing. However, these entropy-based approaches under specific distributions may not be suitable for high-dimensional problems (Geyer et al., 2019).

To address the use of entropy for high-dimensional sampling issues, this chapter

presents Eigen-Entropy (EE), an information entropy based on eigenvalues derived from a correlation coefficient matrix. The proposed EE framework has these contributions: (1) EE is a model-free decision metric since it relies on data to extract information regarding sample sufficiency without any assumptions on data distributions; (2) the theoretical analysis demonstrates that EE is able to well characterize the heterogeneity of a dataset. The following sections present the design of EE and corresponding mathematical proofs.

3.2 Methodology

3.2.1 Eigenvalues and Positive Semi-define Matrices

Let $\mathbf{A} \in \mathbb{R}^{m \times m}$ be a matrix with non-negative entries:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mm} \end{pmatrix} \tag{3.1}$$

where $a_{jk} \geq 0$, $j, k = 1, \dots, m$.

The eigenvalue λ is defined as a scalar, such that:

$$\mathbf{A}\mathbf{v} = \lambda \mathbf{v} \tag{3.2}$$

where $\mathbf{v}$ is the corresponding eigenvector satisfying this equation and $\mathbf{v} \neq \mathbf{0}$.

There exist eigenvalues for $\mathbf{A}$ (Strang, 2016):

$$\lambda_1 + \lambda_2 + \ldots + \lambda_m = \mathbf{tr}(\mathbf{A}) = a_{11} + a_{22} + \ldots + a_{mm} \tag{3.3}$$

where tr(A) is the trace of $\mathbf{A}$, and $\lambda_i, i = 1, \ldots, m$ are the corresponding eigenvalues of $\mathbf{A}$. For a symmetric matrix, the eigenvalues are real, and the eigenvectors are orthogonal (Strang, 2016). A symmetric real matrix $\mathbf{A}$ s positive semi-definite (PSD), denoted by $\mathbf{A} \succeq \mathbf{0}$, if $\mathbf{u}^T\mathbf{A}\mathbf{u} \geq 0$ for every non-zero vector $\mathbf{u} \in R^m$. For a symmetric matrix $\mathbf{A}$, it is PSD if and only if all its eigenvalues are non-negative (Gantmacher, 1977).

3.2.2 Information Entropy

Entropy is a term from physics that measures the degree of chaotic states in a (heat) system (Clausius, 1877). Shannon (1948) extended this concept in information theory to describe the expected volume of information a message contains. Shannon's entropy (H) is defined as

$$H = -\sum_{i=1}^{N} p_i \log p_i \tag{3.4}$$

where N is the number of values a random variable can have, and p_i is the probability of the random variable having the value of i ($\sum_{i=1}^{N} p_i = 1$). It is worth noting that entropy reaches a maximum when $p_i = \frac{1}{N}$ for all i's (uniformly distributed) (Shannon, 1948).

3.2.3 *Eigen-Entropy*

Let $\mathbf{X} \in \mathbb{R}^{n \times m}$ denote a dataset with n samples, where each sample has m features. Then the dataset can be present $\mathbf{X}$ as a matrix below:

$$\mathbf{X} = [\mathbf{x_1}, \ldots, \mathbf{x_n}]^T = \begin{pmatrix} x_{11} & x_{12} & \ldots & x_{1m} \\ x_{21} & x_{22} & \ldots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \ldots & x_{nm} \end{pmatrix} \tag{3.5}$$

where $\mathbf{x_i} = [x_{i1}, \ldots, x_{im}], i = 1, \ldots, n$. Given this, the correlation coefficient matrix on the feature space of $\mathbf{X}$ is defined as

$$\mathbf{C} = \frac{1}{n}\mathbf{X_S}^T\mathbf{X_S} = \begin{pmatrix} 1 & c_{12} & \ldots & c_{1m} \\ c_{21} & 1 & \ldots & c_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \ldots & 1 \end{pmatrix} \tag{3.6}$$

where

$$\mathbf{X_S} = \begin{pmatrix} \frac{x_{11}-\mu_1}{\sigma_1} & \frac{x_{12}-\mu_2}{\sigma_2} & \ldots & \frac{x_{1m}-\mu_m}{\sigma_m} \\ \frac{x_{21}-\mu_1}{\sigma_1} & \frac{x_{22}-\mu_2}{\sigma_2} & \ldots & \frac{x_{2m}-\mu_m}{\sigma_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{x_{n1}-\mu_1}{\sigma_1} & \frac{x_{n2}-\mu_2}{\sigma_2} & \ldots & \frac{x_{nm}-\mu_m}{\sigma_m} \end{pmatrix} \tag{3.7}$$

In Eq. (3.7), μ_j denotes the mean and σ_j denotes the standard deviation of feature j. c_{jk} denotes the correlation between features j and k. That is, $\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}, \sigma_j =$

$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij}-\mu_j)^2}, c_{jk} = \frac{\sum_{i=1}^{n}(x_{ij}-\mu_j)(x_{ik}-\mu_k)}{\sigma_j\sigma_k}(j,k=1,\ldots,m, j\neq k)$, and $c_{jj}=1$.

Let

$$\mathbf{X_S^*} = \begin{pmatrix} \frac{|x_{11}-\mu_1|}{\sigma_1} & \frac{|x_{12}-\mu_2|}{\sigma_2} & \cdots & \frac{|x_{1m}-\mu_m|}{\sigma_m} \\ \frac{|x_{21}-\mu_1|}{\sigma_1} & \frac{|x_{22}-\mu_2|}{\sigma_2} & \cdots & \frac{|x_{2m}-\mu_m|}{\sigma_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{|x_{n1}-\mu_1|}{\sigma_1} & \frac{|x_{n2}-\mu_2|}{\sigma_2} & \cdots & \frac{|x_{nm}-\mu_m|}{\sigma_m} \end{pmatrix} \tag{3.8}$$

Then the correlation magnitude matrix $\mathbf{C}^*$ is derived as

$$\mathbf{C}^* = \frac{1}{n}\mathbf{X_S^*}^T\mathbf{X_S^*} = \begin{pmatrix} 1 & c_{12}^* & \cdots & c_{1m}^* \\ c_{21}^* & 1 & \cdots & c_{2m}^* \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1}^* & c_{m2}^* & \cdots & 1 \end{pmatrix} \tag{3.9}$$

where $c_{jk}^* \geq 0, j,k=1,\ldots,m$.

Next part shows that $\mathbf{C}^*$ is positive semi-definite (PSD).

Proof. For any $\mathbf{a}\in R^m$ and $\mathbf{a}\neq\mathbf{0}$,

$$E((\frac{1}{\sqrt{n}}\mathbf{X_S^*})\mathbf{a})^2 \geq \mathbf{0} \tag{3.10}$$

Then,

$$
\begin{aligned}
E((\frac{1}{\sqrt{n}}\mathbf{X}_\mathbf{S}^*)\mathbf{a})^2 &= E(\mathbf{a}^T(\frac{1}{\sqrt{n}}\mathbf{X}_\mathbf{S}^*)^T(\frac{1}{\sqrt{n}}\mathbf{X}_\mathbf{S}^*)\mathbf{a}) \\
&= E(\mathbf{a}^T\frac{1}{n}\mathbf{X}_\mathbf{S}^{*T}\mathbf{X}_\mathbf{S}^*)\mathbf{a}) \\
&= E(\mathbf{a}^T\mathbf{C}^*\mathbf{a}) \\
&= \mathbf{a}^T\mathbf{C}^*\mathbf{a} \geq \mathbf{0}
\end{aligned}
\tag{3.11}
$$

Since $\mathbf{a}^T\mathbf{C}^*\mathbf{a} \geq \mathbf{0}$, $\mathbf{C}^*$ is PSD.

□

According to Gantmacher (1977), for the symmetric matrix $\mathbf{C}^*$, which is PSD, its eigenvalues are real and non-negative; that is, $\lambda_i \geq 0, i = 1, \ldots, m$. This non-negative property of an eigenvalue is important to support the definition of eigen-entropy (EE).

Definition 1. Following the form of Shannon's entropy, Eigen-Entropy(EE) is defined as

$$
EE = -\sum_{i=1}^{m} \frac{\lambda_i}{m} \log \frac{\lambda_i}{m} \tag{3.12}
$$

Next part is about the relationship between the degree of correlation captured by the correlation magnitude matrix and its eigenvalues.

Proposition 1. Without loss of generality, consider $\mathbf{C}^*$, where all the non-diagnoal entries $c_{ij}^* = c$. λ is one of the corresponding eigenvalues. As the value of c increases, eigenvalue λ increases when $\lambda \in [1, \infty)$ and decreases when $\lambda \in [0, 1)$.

Proof. Construct a new correlation coefficient matrix $\mathbf{C}^{'}$ from $\mathbf{C}^{*}$ by replacing c with αc, $\alpha > 1$:

$$\mathbf{C}^{'} = \begin{pmatrix} 1 & \alpha c & \dots & \alpha c \\ \alpha c & 1 & \dots & \alpha c \\ \vdots & \vdots & \ddots & \vdots \\ \alpha c & \alpha c & \dots & 1 \end{pmatrix} \tag{3.13}$$

According to Chiang and Lin (2013), it is shown that if $\mathbf{C}^{*}$ has an eigenvalue λ, then $\frac{\mathbf{C}^{*}}{\alpha}$ has eigenvalue $\frac{\lambda}{\alpha}$. Thus, when $\lambda^{'}$ is an eigenvalue of $\mathbf{C}^{'}$, then $\frac{\lambda^{'}}{\alpha}$ is an eigenvalue of $\frac{\mathbf{C}^{'}}{\alpha}$.

Note that $\mathbf{C}^{*}$ can be reconstructed using $\frac{\mathbf{C}^{'}}{\alpha}$ and identify matrix as follows:

$$\frac{\mathbf{C}^{'}}{\alpha} + (1 - \frac{1}{\alpha})\mathbf{I} = \begin{pmatrix} 1 & c & \dots & c \\ c & 1 & \dots & c \\ \vdots & \vdots & \ddots & \vdots \\ c & c & \dots & 1 \end{pmatrix} \tag{3.14}$$

Chiang and Lin (2013) have shown that if $\mathbf{C}^{*}$ has the eigenvalue λ, then $\mathbf{C}^{*} + \alpha\mathbf{I}$ has the eigenvalue $\lambda + \alpha$. Given this, the left-hand side of Eq. (3.14) has the eigenvalue $\frac{\lambda^{'}}{\alpha} + 1 - \frac{1}{\alpha}$, while right-hand side is $\mathbf{C}^{*}$ with the eigenvalue λ. Thus,

$$\lambda = \frac{\lambda^{'}}{\alpha} + 1 - \frac{1}{\alpha} \tag{3.15}$$

or equivalently,

$$\lambda^{'} - \lambda = (\alpha - 1)(\lambda - 1) \tag{3.16}$$

From Eq.(3.16), it is concluded that as α increases, the eigenvalue $\lambda^{'}$ increases when $\lambda \in [1, \infty)$, or decreases when $\lambda \in [0, 1)$.

□

Next is about establishment of the relationship between correlation magnitude matrix $\mathbf{C}^*$ and EE.

Proposition 2. Let correlation magnitude matrix $\mathbf{C}^*$ be such that all the non-diagonal entries $c_{ij} = c$. As c increases, EE decreases.

Proof. $\mathbf{C}^*$ is PSD and its eigenvalues $\lambda_i \geq 0, i = 1, \ldots, m$

$$tr(\mathbf{C}^*) = \lambda_1 + \lambda_2 + \ldots + \lambda_m = m \tag{3.17}$$

Thus $\sum_{i=1}^{m} \frac{\lambda_i}{m} = 1$.

Let $p_i = \frac{\lambda_i}{m}$, and replace the $\frac{\lambda_i}{m}$ term in the EE definition with p_i. Set

$$EE = \sum_{i=1}^{m} p_i \log p_i \tag{3.18}$$

As with Shannon's entropy, EE reaches its maximum when $p_i = \frac{1}{m}$, or equivalently, when $\lambda_i = 1$. Now connect c with EE. There are two scenarios:

- $\lambda_i \in [1, \infty)$, that is, $p_i = \frac{\lambda_i}{m} \geq \frac{1}{m}$.

  As c increases, λ_i increases. Thus, p_i will move further away from the maximum entropy point ($\frac{1}{m}$), and EE will decrease.

- $\lambda_i \in [0, 1)$, that is, $p_i = \frac{\lambda_i}{m} < \frac{1}{m}$.

  As c increases, λ_i decreases. Thus, p_i will move further away from the maximum entropy point ($\frac{1}{m}$), and EE will decrease.

It is concluded that as c increases, EE decreases.

□

3.3 Conclusion

In summary, a detailed description of the proposed Eigen-Entropy (EE) method is presented. The key idea of EE is to obtain entropy derived from eigenvalues of a correlation magnitude matrix from multivariate data. The correlation magnitude matrix, $\mathbf{C}^*$, is based on the correlation coefficient matrix and takes the absolute values from correlation coefficients which measure the strength of the correlations (either positive or negative). Since $\mathbf{C}^*$ records the absolute magnitude of the correlations between the variables that define the feature space, and the relationship between $\mathbf{C}^*$ and EE renders EE a potential metric to guide decision-making for sampling. Taking a dataset $\mathbf{X} \in \mathbb{R}^{n \times m}$ (that is, n samples with m features) as an example. EE can be calculated for the first k samples from $\mathbf{X}$, denoted as EE(k). When the $(k+1)$ sample is to be added to the subset, if the new sample increases the variance (σ^2) of the dataset with $(k+1)$ samples (i.e., the dataset is more diversified (heterogeneous), the magnitude of the correlation (c) decreases, and EE($k+1$) increases, and vice versa. Thus it is concluded that the proposed EE can be used as a single metric for decision-making on high dimensional data. More details are described in Huang et al. (2023).

Depending on the application purposes, it is expected to use the proposed EE to support sample selection, feature extraction, etc. The following chapters provide details about construction of EE (Chapter 3), and illustrations on how EE is used to construct baselines for building fault detection (Chapter 4), how EE is used to extract features from graph-structed data to support cross-dataset building fault detection (Chapter 5), and how EE is used to construct Bayes Networks (Chapter 6).

Chapter 4

EIGEN-ENTROPY FOR BUILDING FAULT DETECTION BASELINE CONSTRUCTION

4.1 Background

Buildings are complex and integrated systems consisting of multiple sensors, subsystems, and automatically controlled components, among which the HVAC systems are critical for building energy consumption and indoor environment quality. The HVAC systems have been reported to be responsible for 20% of building energy consumption (Pérez-Lombard et al., 2008) , and such consumption accounts for 36% of global final energy use and 39% of energy-related carbon dioxide emission, according to the United Nations Environment Programme (International Energy Agency & United Nations Environment Programme, 2018). However, among this primary energy use, 30% is wasted because of operation faults and malfunctions in the HVAC systems (Katipamula and Brambley, 2005a). Studies have shown that automatic fault detection and diagnosis (AFDD) on HVAC systems provide great potential for energy savings (Roth et al., 2004). AFDD is the process including fault detection, identification, and isolation. Here faults are typically defined as deviations from normal operating conditions in a building system (Katipamula and Brambley, 2005a). Numerous AFDD methods have been developed for component level and whole building level for building systems over the past decades. These AFDD methods can be generally categorized as qualitative and quantitative model-based methods (such as rule-based and physics-based approaches), and process history-based methods (mostly various data-driven and machine learning based approaches) (Katipamula and Bram-

bley, 2005a,b). No matter what AFDD methods are, a baseline, that can be generated from rules or collected building data to represent normal building operation conditions, is needed to enable fault detection. A baseline depicts the normal operational conditions because an exact fault-free status may rarely exist in a real building (Chen, 2019). This study follows most AFDD building research to use the term "normal" rather than "fault-free" to reflect a satisfied building operation, since a normal status typically exists after a building is freshly commissioned. Here "historical normal data" are denoted as those collected for baseline construction purposes, such as those data collected right after a building is thoroughly commissioned.

There are existing studies focusing on baseline generation for the whole building AFDD. Lin and Claridge (2015) propose a temperature-based approach that constructed the baseline by considering cooling and heating energy consumptions during the post-commissioning period. Miller et al. (2015) introduce a day-typing process using Symbolic Aggregate ApproXimation (SAX) to extract and cluster the most common daily profiles to create a baseline. Fan et al. (2015) present a time series-based data mining to respond to dynamic building data by integrating SAX, motif discovery and temporal association rule. Chen and Wen (2017) further extend the SAX model to SAX based weather pattern matching model (SAX-WPM) to construct a baseline for the whole building systems by clustering time-series with the same weather motif. Li et al. (2010) develop an energy prediction model for baseline construction by removing data outliers and infrequent energy abnormalities that existed in historical data. However, a closer examination reveals that existing research heavily relies on physics-based models as well as building domain knowledge, when constructing a baseline model. In addition, existing work has all taken trial-and-error approaches. In other words, given a dataset, evaluation is required to assess the effectiveness of the dataset. Often, multiple iterations are needed to identify the appropriate dataset

as a baseline. When being applied in a real building, such approaches are not able to answer the following questions: how many and what samples should be included in the baseline. As a result, it is questionable whether the existing research can lead to the construction of a baseline in real time when the sensor data stream-flowing in to support AFDD.

Consequently, this research presents the use of information entropy to seek the answers of how many samples and what samples are in constructing the baseline. Information entropy is to quantify the amount of information presented in random variables, distributions, and events (Shannon, 1948). The proposed method is based on Eigen-Entropy (EE) extracted from the correlation matrix among multiple building sensor readings and outdoor air weather information. Please note an important fact of building HVAC systems is that the systems reconfigure and perform differently under different weather and/or internal conditions. For example, the number of components that a HVAC system functions under the cooling mode, as well as their corresponding parameter settings, is significantly different from that under the heating mode. In the meantime, a transition between two modes could happen within a short period (such as minutes) and for many times during a day. Consequently, it is needed that corresponding baseline that reflects similar weather and/or internal conditions (hence similar operation) when determining whether an incoming sample contains faults or not to differentiate abnormality caused by faults from abnormality caused by operation modes. Outdoor air enthalpy, in short, enthalpy calculated from outdoor air temperature and humidity measurements, is adopted as the feature to reflect outdoor weather. Enthalpy is a property of moist air reflecting the heat needed to condition such air. It can be used to reflect weather conditions and further indicate HVAC operation mode.

As demonstrated in Figure 4.1, EE baseline construction takes a two-step ap-

proach: (1) given a library of historical datasets collected from real building systems under normal conditions, first construct a baseline candidate set by identifying samples from this library that match the incoming data (a.k.a. snapshot) in terms of enthalpy. Here a snapshot is defined as an equal-sized segment of a given time period. For example, if the 24-hour period is divided into 30-min time segments, there will be 48 segments, namely 48 snapshots. (2) Since EE is proved to measure the similarities of the data samples, the higher EE is, the more heterogenous the dataset is, which is assumed to lead to more discriminatory power for fault detection. Under this assumption, EE calculated from these candidate samples is used as a decision criterion to finalize the baseline.

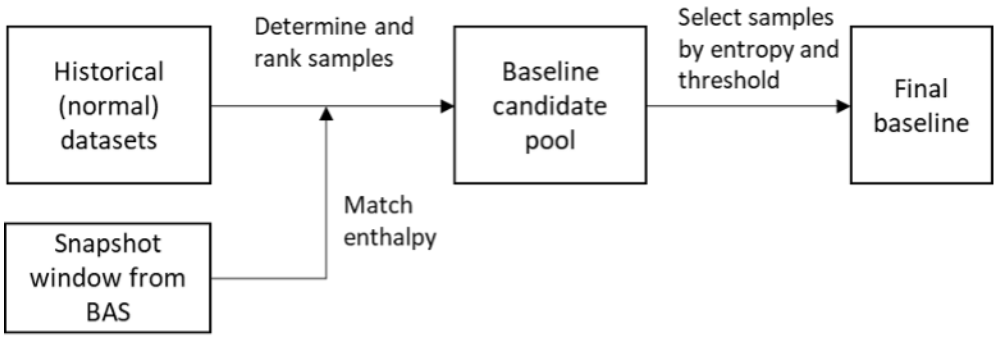

Figure 4.1: Workflow of baseline construction for incoming snapshot window

4.2 Methodology

According to my preliminary study (Huang et al., 2021), weather conditions and BAS (building automation system) sensor measurement are separated for baseline construction. That is, all the collected samples are ranked by enthalpy (weather conditions) firstly, and their entropy is calculated by the covariance of features (sensor readings) within the samples using Pearson's correlation. Since it is found that the

online samples collected over time sometimes have the same values, which leads to an issue of the nonexistence of Pearson's correlation coefficient, cosine similarity is used to derive the correlation coefficient matrix, resulting in EE. Cosine similarity measures similarity between two random variables by angles on the base of cosine (Connor, 2016). It has been commonly used, particularly in nonlinearly high-dimensional positive spaces, to perform tasks such as imagery classifications (Li et al., 2022) and power prediction (Ghasvarian Jahromi et al., 2020).

Cosine similarity between two random variable vectors $\mathbf{f_j}$ and $\mathbf{f_k}$, denoted by $\mathbf{cos} < \mathbf{f_j}, \mathbf{f_k} >$, is defined by:

$$\mathbf{cos} < \mathbf{f_j}, \mathbf{f_k} > = \frac{\mathbf{f_j}^\mathbf{T}\mathbf{f_k}}{||\mathbf{f_j}||||\mathbf{f_k}||} \tag{4.1}$$

where $\mathbf{f_j} = [f_{1j}, ..., f_{nj}]^T$ and $\mathbf{f_k} = [f_{1k}, ..., f_{nk}]^T$. $\mathbf{f_j}^\mathbf{T}\mathbf{f_k}$ is the inner product between $\mathbf{f_j}$ and $\mathbf{f_k}$. $||\mathbf{f_j}||$ and $||\mathbf{f_k}||$ are L2-norm of $\mathbf{f_j}$ and $\mathbf{f_k}$ respectively. The range of $\mathbf{cos} < \mathbf{f_j}, \mathbf{f_k} >$ is [-1,1], where 0 indicates two variables are uncorrelated, 1 indicates (positively linear) correlated between two variables, and -1 indicates (negatively linear) correlated.

During a fault detection process for a building, there are two types of data: incoming snapshot data typically collected or streamlined from a building automation system (BAS) that needs to be judged if it contains faults; And historical data that represent baseline (normal conditions). Note that both incoming snapshot data and historical data consist of multiple features (sensor readings) and EE is extracted on the feature space of historical data according to equations (3.6), (3.9), (3.12) & (4.1). Only typical building automation system measurements (illustrated in Chen (2019)) are used for this study. The goal of the proposed baseline method is to dynamically (in real time) construct a customized baseline based on the incoming snapshot data's weather condition, time of the day, and operation mode by selecting appropriate (i. e.,

similar weather, time of the day, and operation mode) samples from historical normal datasets. As discussed in (Chen, 2019), the main challenge for building system fault detection is to differentiate abnormality caused by faults from other factors, such as weather, operation mode (occupied, setback, etc.), and internal load.

Historical data often contains multiple operation modes. There could be different operation modes even within the same season. Chen (2019) suggests identifying operation mode by classifying historical dataset based on their corresponding operation mode. Internal load for a commercial building is typically correlated to occupancy. For buildings that do not have measurements on occupancy, which is the situation considered in this study, time of the day is used as the best surrogate of occupancy. Hence Chen et al. (2022a) suggest that if more detailed occupancy information is not available, then only historical data that had a similar time frame as the incoming data would be used to construct the baseline. Figure 4.2 illustrates the baseline construction workflow using EE.

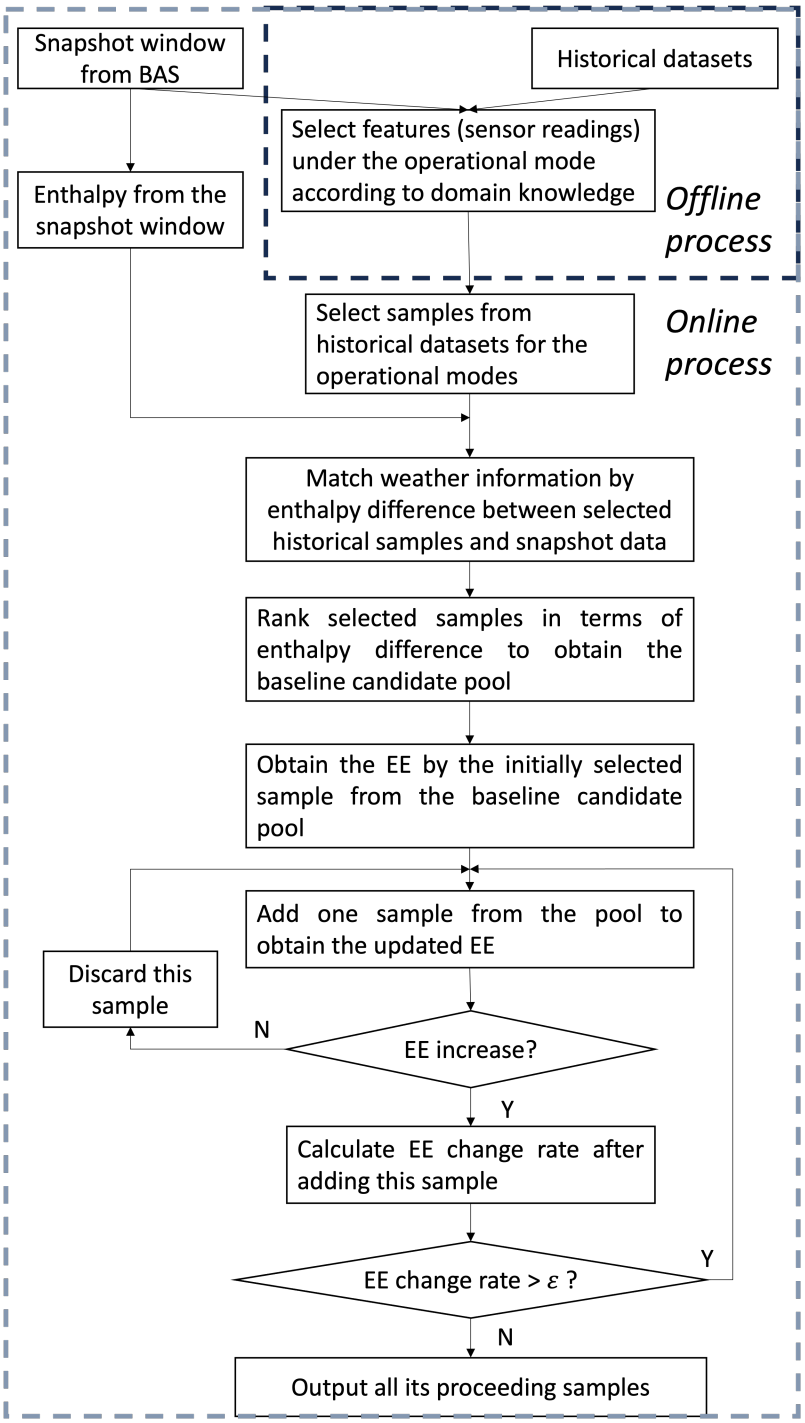

Figure 4.2: Procedures of determination of the number of samples for each snapshot window by EE

Algorithm 1 EE baseline construction

Input: Given a test case with respect to historical normal datasets, snapshot window $SW_j, j = 1, \ldots, w$, each SW_j with v samples. For the test case, it has enthalpy (ENL) and p sensor measurements. For the normal samples, denoted as FF, only p sensor measurements are considered.

Output: Baseline **B**

1: Determine the operation mode by domain knowledge for incoming snapshot data, for the operation mode, all normal samples from m-day historical datasets are selected, $FF_i, i = 1, \ldots, n$.

2: Initialization: $\mathbf{B} = \emptyset$

3: **For** $j = 1, \ldots, w$ **do**:

4: For the snapshot window, calculate average enthalpy, $\overline{ENL}_{SW_j}$, from the v samples within the SW_j

5: Select all samples from historical database that are within the time windows $(SW_{j-u}, \ldots, SW_j, \ldots, SW_{j+u})$. Let the total samples be l, and then calculate enthalpy difference for each sample, $\Delta ENL_{FF_k} = |ENL_{FF_k} - \overline{ENL}_{SW_j}|$

6: Rank all FF_k in ascending order according to $\Delta ENL_{FF_k}, k = 1, ..., l$, and get a set of ranked samples RH_j, $RH_j = \{FF_{(1)}, ..., FF_{(l)}\}$ (Note each $FF_{(k)}$ has p sensor measures).

7: Standardize RH_j on each measure, and get $RH_j' = \{FF_{(1)}', ..., FF_{(l)}'\}$.

8: Initially, take the first s samples from RH_j' as $C_j = \{FF_{(1)}', ..., FF_{(s)}'\}$. Given C_j, and calculate Eigen-Entropy using Eqs (3.6), (3.9), (3.12) & (4.1), denoted as EE.

9: **for** $q = s + 1, \ldots, l$ **do**:

10: Append sample q to C_j and calculate the Eigen-Entropy using Eqs (3.6), (3.9), (3.12) & (4.1), denoted as EE*.

11: If EE* > EE and EE* / $(|C_j| - 1) \geq \epsilon$, then keep sample q in C_j, and update EE with EE*.

12: Otherwise, stop and append **B** with C_j.

13: **Return B**

The EE baseline construction process includes an offline portion, which is accomplished after historical datasets are collected, and an online portion, which is being executed for every incoming snapshot window in real time. During the offline process, a feature selection process is conducted under each operation mode to identify the features that are highly relevant to building operation. Details about this feature selection process, which uses a Partial Least Square Regression and Genetic Algorithm (PLSR-GA) method, are described in (Chen, 2019; Chen et al., 2022a). The PLSR-GA method is used to identify the subset of candidate features that represent the system performance under each operational mode. The GA searching process is used to facilitate the process of searching candidate feature subsets. In this study, every five consecutive days are grouped into one selection scenario under each operation mode. If a feature is selected in three scenarios, then the feature is used in the development of the PCA model. The details of the online EE baseline construction are summarized in Algorithm 1. The first step is to determine the operation mode and the corresponding m-day historical dataset based on which the incoming snapshot (SW) data are by the domain knowledge (Algorithm 1, lines 1). Notice that the number of days of historical data used for this study is summarized in Table 4.2. Internal load for a commercial building is typically correlated to occupancy. For buildings that do not have measurements on occupancy, which is the situation considered in this study, time of the day is used as the best surrogate of occupancy. Next, for an incoming SW, corresponding and neighboring SWs in historical datasets should be explored (so that only historical data that have a similar time of the day would be initially considered).

Taking one incoming snapshot window SW#7 (9:00 – 9:30) as an example (Algorithm 1, line 3), both corresponding SW #7 and neighboring SWs - SW#4 - SW#6 (7:30 – 9:00) and SW#8 - SW#10 (9:30 – 11:00) in historical datasets should be

considered (Algorithm 1, lines 5). Among these corresponding and neighboring SWs in historical datasets, a small number of samples (e.g., 6 samples) with the similar enthalpy values as the incoming SW are selected as an initialization (Algorithm 1, line 4). Then remove those samples resulting in a decrease in cumulative entropy (Algorithm 1, lines 6-12) to obtain the candidate pool. Finally, calculate the entropy change rate starting from the second sample (Algorithm 1, line 12). If the entropy change rate is below threshold ϵ (a small number), this indicates that adding one more sample will not help increase EE; thus, the heterogeneity of the baseline, the procedure stops (Algorithm 1, line 12). Finally, the baseline is determined (Algorithm 1, line 13).

4.3 Experiments

Two sets of experiments are conducted for evaluating the proposed method. The first experiment is to assess the accuracy of the baseline in helping fault detection method to detect faults. For this experiment, the incoming snapshot data are from real building systems that contain artificially injected faults. To verify the robustness of the baseline, the second experiment is to assess the false alarm rate of the fault detection method using the constructed baseline.

4.3.1 Datasets

Datasets used for evaluation are collected from a medium sized commercial building in Philadelphia, PA (Chen, 2019) as part of an AFDD algorithm development project (see Figure 4.3). Typical HVAC and BAS systems are equipped for this building. Three libraries of historical normal datasets under three different operation

modes (two in the summer season and one in transitional season) have been collected for baseline development (see Table 4.1). The incoming snapshot data are from two groups of test cases: those that contain artificially injected faults (for Experiment I) and those that do not (for Experiment II). Both experiments contain 14 cases.

Figure 4.3: The medium sized commercial building in Philadelphia, PA

Note samples in each dataset are collected for one day period with a 5-min sample rate. According to sensitivity analysis of snapshot window size in (Chen, 2019), 30-min is an adequate value for generating a more accurate weather baseline with a relatively low computation burden. Hence in this study, each test case and each historical dataset have 48 SWs ($w = 48$ in Algorithm 1), and each SW with a size of

30 minutes and 6 samples ($v = 6$ in Algorithm 1). Each sample consists of multiple features besides enthalpy. Since building systems operate under different modes in different seasons, there are different numbers of features in each sample for different operation modes ($p = 100$ for Mode 1, $p = 182$ for Mode 2, and $p = 167$ for Mode 3, respectively, in Algorithm 1) and there are different numbers of historical datasets for different operation mode.

Table 4.1: Description of the three operation modes (Adapted from Chen (2019))

Operation mode	Season	Description
Mode 1	Summer	The chiller is on continuously. The air handling unit (AHU)'s cooling coil is operated to maintain the supply air temperature during the unoccupied time.
Mode 2	Summer	The chiller is on continuously during an occupied time and is operated under an on/off operation mode during the unoccupied time. The AHU supply air temperature may not be strictly maintained at its setpoint during an unoccupied time when the chiller is turned off.
Mode 3	Transitional season and winter	The chiller plant is not in use. AHU's outdoor dampers are adjusted following a pre-defined economizer control strategy. On a day when the outdoor air temperature is extremely low, the AHU preheating coil is used to preheat the incoming outdoor air to prevent freezing the coils.

Table 4.2: Summary of libraries of historical datasets

Library	Operation mode	# of historical datasets (days)	# of samples in each dataset	# of features in each sample
Library 1	Mode 1	45	288	100
Library 2	Mode 2	55	288	182
Library 3	Mode 3	101	288	167

Table 4.3: Summary of 14 fault test cases

Fault test case name	Fault description	# of candidate samples	# of features
20160706	System stopped at 4:00 PM to 11:30 PM	60480	100
20160907	AHU-2 supply air temperature sensor negative bias 4F	60480	100
20160911	Operator fault, chiller off	60480	182
20161201	AHU-2 outdoor air damper stuck at 90% open	60480	167
20170103	AHU-2 outdoor air damper stuck at 80% open	60480	167
20170114	Occupied from 1:30 AM to 7:00 AM	60480	167
20170811	AHU-2 cooling coil valve position software override at 100% open	56448	182
20170915	Chiller chilled water differential pressure sensor positive bias 0.1 psi	56448	182
20180709	AHU-2 supply air temperature sensor bias fault negative 3.5F	60480	182
20180710	AHU-2 OA damper stuck at 30% open	60480	182
20180711	AHU-2 cooling coil valve stuck at 80%	60480	182
20180718	AHU-2 OA damper stuck at 60% open	60480	182
20180722	Change weekend occupied schedule to end at 8:20 PM	60480	100
20180723	CHWS temperature sensor negative bias 3.0 F	60480	182

Table 4.4: Summary of 14 false alarm evaluation cases

Case name	Operational Period	# of candidate samples	# of features
20160726	7:00 AM - 9:00 PM	60480	100
20160826	7:00 AM - 9:00 PM	60480	100
20160827	7:00 AM - 9:00 PM	60480	100
20160828	7:00 AM - 9:00 PM	60480	100
20160829	7:00 AM - 9:00 PM	60480	100
20161202	7:00 AM - 9:00 PM	60480	167
20161208	7:00 AM - 9:00 PM	60480	167
20170120	7:00 AM - 9:00 PM	60480	167
20170124	7:00 AM - 9:00 PM	60480	167
20170611	7:00 AM - 9:00 PM	60480	182
20170619	7:00 AM - 9:00 PM	60480	182
20170702	7:00 AM - 9:00 PM	60480	182
20170705	7:00 AM - 9:00 PM	60480	182
20170816	7:00 AM - 9:00 PM	60480	182

4.3.2 Benchmark Methods

Since the building researchers more care about the capacity of fault detection performances in cases with artificial injected faults, two benchmark algorithms, SAX-WPM model and MPMI, are included for fault detection comparisons in Experiment I.

MPMI is a general method to assess the information richness of a multivariate dataset. For a sample $x_i = [f_{i1}, f_{i2}, \ldots, f_{im}]$, MPMI is defined as

$$MPMI(x_i) = log \frac{p(f_{i1}, f_{i2}, \ldots, f_{im})}{p(f_{i1}), p(f_{i2}), \ldots, p(f_{im})} \tag{4.2}$$

where $p(f_{i1}, f_{i2}, \ldots, f_{im})$ and $p(f_{ij})$ are the joint probability of m features and the probability of feature j in x_i, respectively.

For each $x_i \in \mathbf{X}$, both joint probability and individual feature probability are estimated by histograms under a specific bin number. The MPMI of each sample is then normalized to obtain normalized MPMI (NMPMI). A rejection sampling algorithm (Dutta et al., 2019) is applied for sampling decisions. That is, let s_i be picked from a uniform distribution $\mathcal{U}(0, 1)$: (1) if $NMPMI(x_i) > s_i$, x_i is selected; (2) otherwise, x_i is discarded. There is one parameter in implementing MPMI: the bin number. Here, set the bin number to 128 as in (Dutta et al., 2019).

In SAX-WPM, the symbolic aggregate approximation (SAX) method is first employed to find similar patterns within a time series dataset, and these patterns are used to dynamically select qualified samples to generate a baseline. The SAX-WPM method has shown satisfactory performance over conventional data-driven baseline

construction methods for high-dimensional building data (Chen, 2019). There are two parameters in implementing SAX-WPM (Chen, 2019): (1) the snapshot window size, which, as the name implies, divides a day (24 hours) into snapshots (e.g., 1 hour) for use in building control and building fault detection; (2) the number of symbolic letters, which is used to categorize weather conditions. As in (Chen, 2019), the snapshot window size is set to 30 min and the number of symbolic letters is set to 10.

4.3.3 Evaluations

Following the literature on anomaly detection (Albert, 2009; Chen, 2019; Jolliffe, 2002; Kano et al., 2001), Hotelling's T-Square (T^2) is adopted, incorporating the principal component analysis (PCA) to identify whether a systematic abnormality exists in the building operations with respect to the baseline constructed. Specifically, T_i^2, Hotelling's T-square for a sample i, is defined as

$$T_i^2 = \mathbf{x_i^T P} \sum a \, \mathbf{P^T x_i} \tag{4.3}$$

where x_i is the sample i, $\mathbf{P}$ is a loading matrix obtained from PCA, and $\sum a$ is a set of the nonnegative eigenvalues corresponding to a principal components. Because T_i^2 follows the F distribution, its upper bound can be obtained as

$$T_{threshold}^2 = \frac{a(n-1)}{n-a} F_{a,n-1,\alpha} \tag{4.4}$$

where n is the number of samples and α is the level of significance. Here α is set to 0.05, and an abnormal samples i is flagged when $T_i^2 \geq T_{threshold}^2$.

According to (Chen, 2019), fault detection ratio (FDR) is defined to evaluate the performances of the constructed baselines for fault detection. For each test case, FDR is defined as

$$FDR = \frac{D}{N} \tag{4.5}$$

where D is the number of samples flagged as faulty and N is the number of total samples respectively within a given time period. A test case is considered faulty if its FDR ≥ 0.5, following building domain knowledge (Chen, 2019).

4.3.4 Experimental Results

Experiment I focuses on 14 artificial fault injection cases, and it is needed to determine the threshold ϵ to generate the baselines. Consequently, 9 different thresholds ϵ are tested, ranging from 0.01 to 0.09 with 0.01 increments. After determining the threshold from Experiment I, it is needed to validate the robustness of this value for baseline constructions by 14 false alarm evaluation cases in Experiment II. In other words, constructed baselines under the chosen threshold should be able to detect as many faulty cases as possible in Experiment I (i.e., a faulty case is successfully detected if its FDR ≥ 0.5), and will not mis-detect normal cases as faulty ones in Experiment II (i.e., a normal case is not mis-detected as faulty if its FDR < 0.5).

The results of the fault detection test for 14 fault test cases in Experiment I using EE-constructed baselines are shown in Figure 4.4. Here the detection rate (fraction of detected fault test cases) is reported. It can be observed that baselines constructed by the proposed method are able to detect all fault test cases under $\epsilon = 0.07$ (see Figure 4.4 (A)). It is worth noting that an average number of baseline samples per case decreases as expected when ϵ increases (see Figure 4.4 (B)), mainly because a

higher value of ϵ indicates that additional samples lead to a larger entropy change rate, and thus a smaller number of only qualified samples can be included.

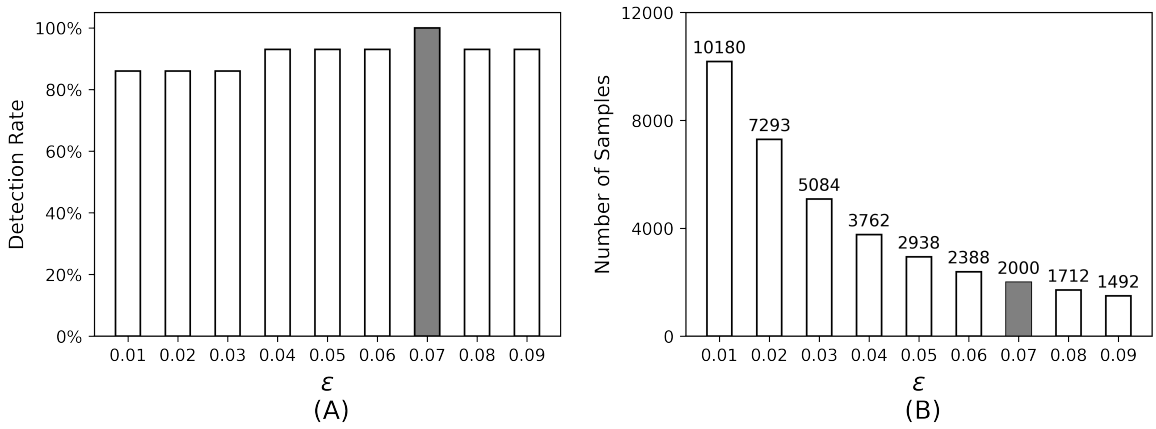

Figure 4.4: Detection results for 14 artificial fault injection cases using baselines constructed by EE under different ϵ 's: (A) fraction of detected fault test cases; (B) average number of baseline samples per case. $\epsilon = 0.07$ is highlighted in grey.

Next is to validate this $\epsilon = 0.07$ in Experiment II. Note that in Experiment II, all cases are under normal operation. Hence a smaller value of FDR is desired since it indicates a lower chance of misdetection. Observing from Figure 4.5, none of these cases are flagged as faulty ones, or in other words, there is no misdetection. Consequently, $\epsilon = 0.07$ is considered as the stopping criterion in the proposed method to generate the final baseline, given that the number of baseline samples is relatively small while all fault test cases are detected under this value and none of false alarm evaluation cases are mis-detected.

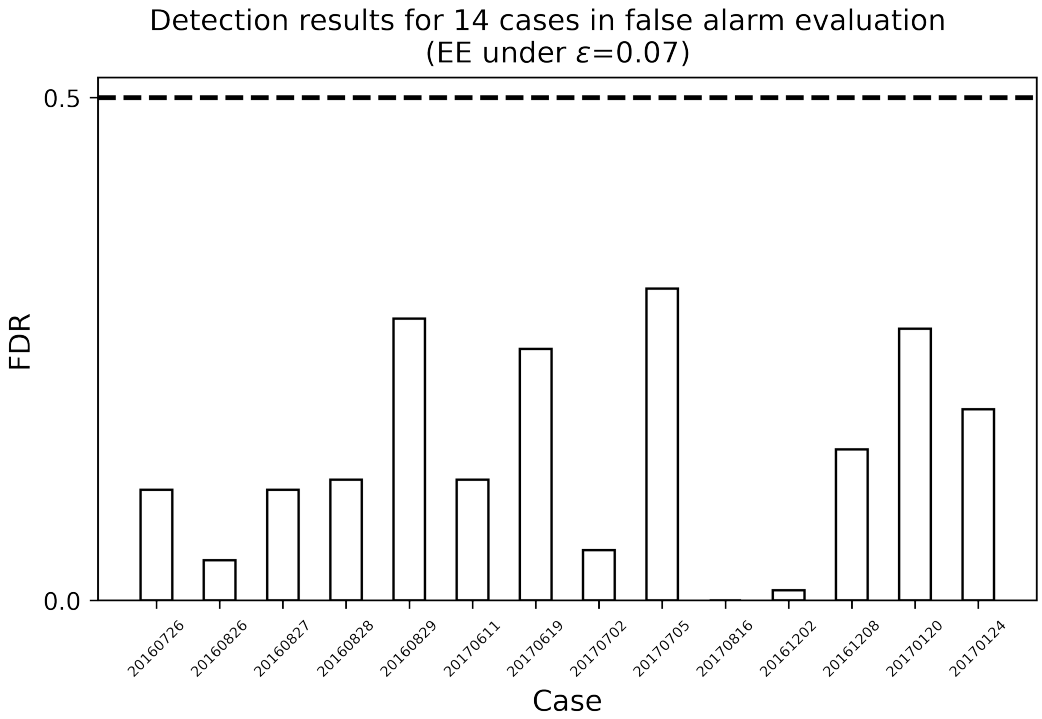

Figure 4.5: Detection results for 14 false alarm evaluation cases using baselines constructed by EE under $\epsilon = 0.07$. None of cases are misdetected.

Finally, Table 4.5 presents the comparison of the fault detection results for the MPMI baselines, SAX-WPM baselines, and those by the proposed EE method under $\epsilon = 0.07$. As previously mentioned, a fault test case is detected when its FDR is greater than or equal to 0.5; therefore, it is observed that among the 14 fault test cases, baselines by MPMI failed to detect 2 cases, cases 20160911 and 20170114, (see grey-colored rows in Table 4.5), while those by SAX-WPM and EE are able to detect all cases. For case 20160911, the number of samples in the baseline constructed by EE is 1788, 89.3% less than that of MPMI, which is 22896, and 69.7% less than that of SAX-WPM, which is 5904; for case 20170144, the number of samples in the baseline constructed using EE is 2273, 86.4% less than that of MPMI, which is 16656, and 76.9% less than that of SAX-WPM, which is 9840. For all 14 cases, the average

number of baseline samples per case using MPMI and SAX-WPM are 15574 and 8099 respectively, while that using EE is 2000.(87.2% and 75.3% less than those of MPMI and SAX-WPM, respectively, see Figure 4.6). Therefore, baselines constructed using the EE method require a significantly smaller number of samples than those by the MPMI method, which also indicates that the proposed method is promising.

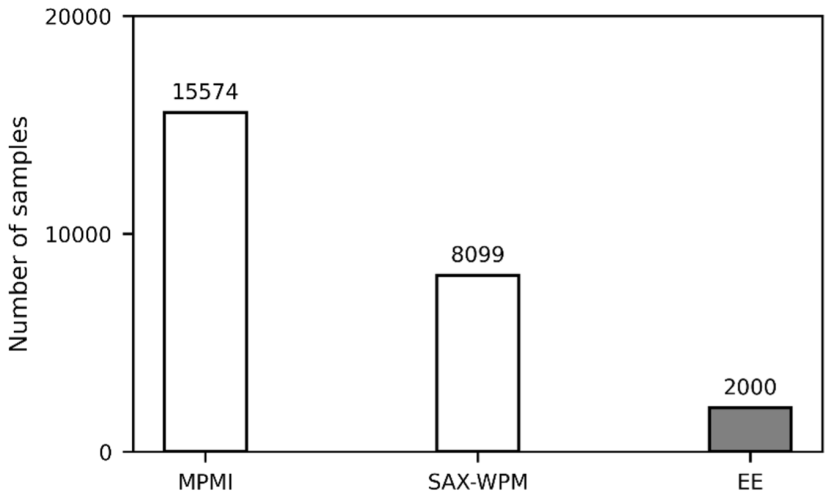

Figure 4.6: Comparison of the results using the MPMI, SAX-WPM, and EE ($\epsilon = 0.07$) methods in terms of the average number of constructed baseline samples across 14 fault test cases.

In summary, this case study demonstrates that the sampling outputs (baselines) by the proposed EE method (EE-based heterogeneous sampling) outperform the literature-reported method (MPMI) for fault detections. Additionally, baselines constructed by the EE method require significantly fewer samples than those by the MPMI method. It is concluded that the EE-based sampling method is able to make sampling decisions such as *which* samples and *how many* samples should be included in the heterogeneous sampling scenario.

Table 4.5: Comparison of detection results for 14 fault test cases using MPMI, SAX-WPM, and EE baselines in terms of FDR and the average number of constructed baseline samples. The two cases that are marked are not detected by the MPMI baselines but are detected by the EE baselines. (MPMI: bin = 128; SAX-WPM: snapshot window size = 30-min, number of symbolic letters = 10; EE: $\epsilon = 0.07$)

Fault test case name	FDR			Number of samples		
	MPMI	SAX-WPM	EE	MPMI	SAX-WPM	EE
20160706	1.00	1.00	1.00	17232	10416	2083
20160907	0.90	1.00	1.00	15216	7344	1905
20160911	0.12	0.92	0.52	22896	5904	1788
20161201	0.88	1.00	0.88	18480	8880	2051
20170103	0.99	0.99	0.99	18672	6432	2142
20170114	0.16	0.78	0.52	16656	9840	2273
20170811	0.89	0.90	0.94	17856	7200	1990
20170915	1.00	1.00	1.00	17952	8016	2024
20180709	1.00	1.00	1.00	22800	7296	1846
20180710	1.00	1.00	0.98	22800	10608	2054
20180711	1.00	1.00	1.00	22896	10320	2095
20180718	1.00	1.00	1.00	22752	8304	2012
20180722	1.00	1.00	1.00	17376	8352	2053
20180723	0.98	1.00	0.99	22848	4224	1679

4.4 Eigen-Entropy for Other Applications

Previous sections present that the use of Eigen-Entropy for building fault detection baseline construction. Depending on application purposes, EE can also be used to select homogeneous samples. In methodology section, a more generalized EE-based sampling method is described and how it is used to support homogeneous sampling.

Given a dataset $\mathbf{X} \in \mathbb{R}^{n \times m}$, Algorithm 2 presents the EE(Eigen-Entropy)-based sampling algorithm to select the subset $\mathbf{S}$ from $\mathbf{X}$. The first step is to normalize the dataset $\mathbf{X}$ on each feature (Algorithm 2, line 2) and fill $\mathbf{S}$ with some initial samples to calculate the EE. Next, for each of the remaining data samples, calculate new EE on the updated $\mathbf{S}$ (with the added sample) to observe the change of EE (increasing or decreasing) and evaluate the rate of change from the EE (Algorithm 2, line 8). The addition is finalized if the rate of change is below a pre-defined threshold, ϵ, a small number. Depending on the applications, in the case where data leading to the more homogeneous dataset is desired, EE is expected to decrease; in the case where data leading to the more heterogeneous dataset is desired, EE is expected to increase.

Moreover, EE-based sampling in Algorithm 2 adopts greedy search strategy to identify the samples to be included or excluded in the sampling process. It is noted greedy search may be trapped at local optimum if the EE-curve is not monotonous increasing or decreasing. Fortunately, the objective here is to continuously update the EE-curve with added samples to keep the monotonous property of the EE-curve. For illustration, two example samplings, one on homogeneous sampling case study and one on heterogenous sampling case (baseline construction) study are shown in Figure 4.7.

Algorithm 2 EE-based Sampling - Generalized

Input: $\mathbf{X} \in \mathbb{R}^{n \times m}$, n samples, m features

Output: The subset $\mathbf{S}$ from $\mathbf{X}$ determined by EE

Initialization:

    Normalize $\mathbf{X}$ on each feature to obtain $\mathbf{X}^{'}$

    Initialize $\mathbf{S}$ with a few samples from $\mathbf{X}^{'}$

    Calculate EE using $\mathbf{S}$ (see **Definition 1**)

Sampling Decision:

    For each of the remaining samples

        Temporarily add the data element into $\mathbf{S}$

        Calculate the rate of EE change as $\frac{EE}{\#\, of\ samples\ in\ \mathbf{S}}$

    For applications where homogeneous samples are desired

        If EE decreases, keep the element in $\mathbf{S}$

        Otherwise, remove the data element from $\mathbf{S}$

        If the rate of EE change is greater than ϵ, sampling process continues

        Otherwise, stop

    For applications where heterogeneous samples are desired

        If EE increases, keep the element in $\mathbf{S}$

        Otherwise, remove the data element from $\mathbf{S}$

        If the rate of EE change is greater than ϵ, sampling process continues

        Otherwise, stop

Return S

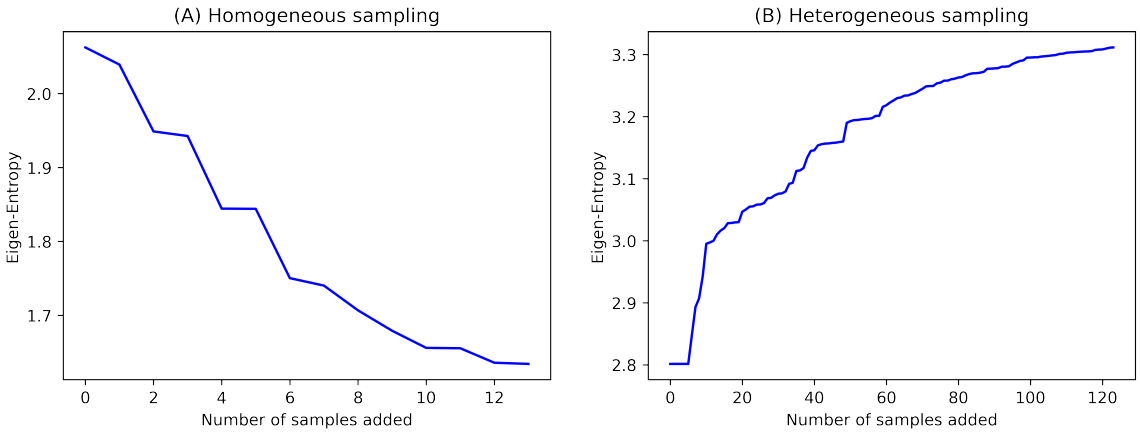

Figure 4.7: Monotonous property of EE-curve is kept while updated with added samples for (A) a homogeneous sampling case (B) a heterogeneous sampling case.

Finally, it is worth noticing that the direction of EE change in this sampling method is not order-dependent. For example, if a sample will result in the increase in the EE, it will eventually increase the EE no matter which order it is added. In other words, a sample will have the same impact on the overall dataset regardless of its order.

Proposition 3. The direction of EE change is order-independent. In other words, a sample resulting in a more heterogeneous dataset will eventually lead to the increase of the dataset regardless of the order it is added.

Proof. Without loss of generality, suppose there is a dataset $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_p}]$ such that its correlations are c. There is one sample $\mathbf{x_{p+1}}$ making $\mathbf{X}$ more heterogeneous, and another sample $\mathbf{x_{p+2}}$ making $\mathbf{X}$ more homogeneous, respectively.

- Scenario 1. Add $\mathbf{x_{p+1}}$ first and $\mathbf{x_{p+2}}$ second.

  c will first decrease by adding $\mathbf{x_{p+1}}$ to $\mathbf{X}$ since the dataset becomes more heterogeneous, and then increase by adding $\mathbf{x_{p+2}}$ to $\mathbf{X}$ since the dataset becomes more

homogeneous. As a result. EE will first increase after adding $\mathbf{x_{p+1}}$, and then decrease after adding $\mathbf{x_{p+2}}$. Suppose $\mathbf{X}^{'} = [\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_p}, \mathbf{x_{p+1}}, \mathbf{x_{p+2}}]$ whose mean and covariance becomes $\mu_{\mathbb{X}'}$ and $\mathbf{\Sigma}_{X'}$ respectively.

- Scenario 2. Add $\mathbf{x_{p+2}}$ first and $\mathbf{x_{p+1}}$ second.

  c will first increase by adding $\mathbf{x_{p+2}}$ to $\mathbf{X}$ since the dataset becomes more homogeneous, and then decrease by adding $\mathbf{x_{p+1}}$ to $\mathbf{X}$ since the dataset becomes more heterogeneous. As a result. EE will first decrease after adding $\mathbf{x_{p+2}}$, and then increase after adding $\mathbf{x_{p+1}}$. Suppose $\mathbf{X}^{''} = [\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_p}, \mathbf{x_{p+2}}, \mathbf{x_{p+1}}]$ whose mean and covariance becomes $\mu_{\mathbb{X}''}$ and $\mathbf{\Sigma}_{X''}$ respectively.

Obviously, it is obvious that $\mu_{\mathbb{X}'} = \mu_{\mathbb{X}''}$ and $\mathbf{\Sigma}_{X'} = \mathbf{\Sigma}_{X''}$. Consequently, the EE of $\mathbf{X}^{'}$ is equal to that of $\mathbf{X}^{''}$. Thus it is concluded that the direction of EE change is order-independent. □

Consequently, the above proof has shown that the direction of EE change in this sampling method is not order-dependent.

4.4.1 Eigen-Entropy for Homogeneous Sampling in Imbalanced Learning

Here let's first focus on the imbalanced learning problem to demonstrate the use of Eigen-Entropy to assist sampling decisions where homogeneous data are to be sampled. Data imbalance is a common phenomenon for machine learning problems in many fields of study, e.g., cardiovascular disease studies (Fan et al., 2022) or credit card fraudulent transactions (Carcillo et al., 2021). As a result, prediction regarding the minorities are of great importance. Extensive research has proposed solutions to address the challenges of an imbalanced dataset by balancing distributions of majorities and minorities classes (Wan et al., 2018). Oversampling techniques are

frequently used (He and Ma, 2013). Common oversampling techniques include the synthetic minority over-sampling technique (SMOTE) (Chawla et al., 2002), the majority weighted minority over-sampling technique (MWMOTE) (Barua et al., 2014), SMOTE + Tomek links (SMTL) (Batista et al., 2004), SMOTE + Edited nearest neighbors (SMENN) (Batista et al., 2003), EasyEnsemble (EASY) (Liu et al., 2009), and Balance-Cascade (BC) (Liu et al., 2009). The basic idea behind most of these oversampling techniques is to generate synthetic samples based on the distribution information from the minority class to balance the datasets. However, these approaches may generate noisy or wrong minority samples leading to misclassification (Li et al., 2020).

Instead of using distribution information, the argument is that if the minority data samples are carefully selected (e.g., homogeneous samples being selected) as the basis for generating synthetic samples, the synthetic dataset may be less noisy, resulting in better classification performance. Thus the proposed EE is used as a metric to guide oversampling decisions. Specifically, given a multi-class imbalanced dataset, for each of the minority classes, Algorithm 2 is applied to identify the homogeneous samples to generate the synthetic samples to obtain a balanced dataset. In this case, the smaller the EE is, the less diverse (more homogeneous) the dataset is. In the next section, where EE is compared against oversampling techniques , cosine similarity is used as a correlation coefficient measurement because Pearson's correlations may not exist, which is observed (Huang et al., 2021), and cosine similarity indeed overcomes issues of the nonexistence of correlations(Connor, 2016).

4.4.2 Imbalance Datasets

Two sets of experiments are conducted to validate the efficacy of EE-based homogeneous sampling. In experiment I, ten publicly available real-world datasets from KEEL (Alcala-Fdez et al., 2011) and UCI repositories (Fernández et al., 2008) are used in the comparison experiments (see Table 4.6). The first 7 datasets, vehicle1, segment0, page_block0, penbased , thyroid, shuttle, and ecoli-0-1-4-7_vs_2-3-5-6 (ecoli) are from KEEL, while the remaining, letter, waveform database generator version 1 (wavefm3), and landsat are from UCI. Among these 10 datasets, 4 (vehicle1, segment0, page_block0, and ecoli) are with binary classes, and the other 6 are multiclass datasets. Features in these datasets are numerical.

Experiment II focuses on the FANCY dataset. FANCY (Harer et al., 2017) is a clinical study to evaluate how acute kidney injury will impact the human's renal functionality in a long run, whose dataset is provided by the University of Virginia with approval from the Institutional Review Board. Since glomerular filtration rate (GFR) (NIDDK, nd) is an important index measuring the renal functionality, the value of GFR less than 90 indicates the chronic kidney disease. Therefore, in this FANCY dataset, there are 9 diseases and 24 healthy cases respectively. Each instance has 14 urine biomarkers, or equivalently, 14 features. Table 4.7 summarizes the statistics of data information and more details can be referred to (Huang et al., 2022).

Table 4.6: Statistics of 10 experimental datasets (IR: imbalanced ratio)

Datasets	#Instance	#Features	#Classes	IR
vehicle1	846	18	2	2.9
segment0	2308	19	2	6.02
page_block0	5472	10	2	8.79
penbased	1100	16	10	1.95
thyroid	720	21	3	36.94
shuttle	2175	9	7	853
ecoli	336	7	2	10.59
letter	5000	16	26	0.96
wavefm3	5000	21	3	2.04
landsat	2000	36	6	1.98

Table 4.7: Statistics of FANCY datasets

Class	#Instance	#Features	Label
Disease (GFR < 90)	9	14	1
Healthy (GFR ≥ 90)	24	14	0

4.4.3 Benchmark Methods for Imbalance Learning

For comparison purposes, in experiment I, four SMOTE-based oversampling techniques, SMOTE (Chawla et al., 2002), MWMOTE (Barua et al., 2014), SMTL (Batista et al., 2004), SMENN (Batista et al., 2003), are included because SMOTE has been widely adopted as an oversampling technique (He and Ma, 2013). In addition, an entropy-based imbalance degree sampling method, called the entropy-based oversampling approach (EOS), is included owing to its superior performance in imbalance learning (Li et al., 2020). As a result, there are a total of five methods that are compared against the proposed method.

While in experiment II, only SMOTE method is included. It is worth noting that all these techniques are to generate more samples from the non-majority classes to obtain the same number of samples as that from the majority class. With the same goal, instead of randomly selecting samples as the seed to generate new samples such as in SMOTE, the proposed method is to identify homogeneous samples from the non-majority classes using EE as the seed for new sample generations via SMOTE. Thus, the proposed method is termed EE-SMOTE.

4.4.4 Evaluations

To investigate the information richness of the sampled data to support imbalanced learning, two commonly used base classifiers, multilayer perceptrons (MLP) (Bishop, 2006) and AdaBoost (Hastie et al., 2009) are implemented (Table 4.8) in experiment I. For each dataset, 5-fold cross-validation is performed. Each classifier is trained 10 times and the output is the average performance over the 10 runs. The One-vs-Rest strategy (Rifkin and Klautau, 2004) is applied to multi-class datasets.

Table 4.8: Summary of two base classifiers used in experiment I

Base Classifier	Parameters
AdaBoost	100 boosting iterations
MLP	100 epochs, 0.1 learning rate, 10 hidden layer neuros

Table 4.9: Summary of five base classifiers used in experiment II

Base Classifier	Parameters
LR	'L2' penalty
SVM-L	'L2' penalty, C = 1.0
SVM-RBF	C = 1.5
RF	100 estimators, 'gini' criterion
XGBoost	100 boosting iterations

While in expriment II, five commonly used classifiers are selected, including Logistic Regression (LR) (Cox, 1958), Support Vector Machine with Linear kernel (SVM-L) (Cortes and Vapnik, 1995), Support Vector Machine with Radial basis function kernel (SVM-RBF) (Cristianini and Taylor, 2000), Random Forest (RF) (Breiman, 2001), and XGBoost (Chen and Guestrin, 2016), and corresponding parameters are shown in Table 4.9. To validate the performance of classifiers, Leave-one-out cross-validation (Hastie et al., 2009) is used for validation: that is, one instance is selected for testing, while the rest instances are used for training. Therefore, each classifier is trained 48 times and the performance results are obtained after all iterations.

Recall (Powers, 2011), precision (Powers, 2011), F-measure (Hripcsak and Rothschild, 2005) and G-mean (Guo et al., 2016) is used to evaluate performances in

experiment I. Note that only F-measure is used in experiment II. These evaluation metrics are defined as below:

$$Recall = \frac{TP}{TP + FN} \tag{4.6}$$

$$Precision = \frac{TP}{TP + FP} \tag{4.7}$$

$$F - measure = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{4.8}$$

$$G - mean = \sqrt{Recall \times \frac{TP}{TN + FP}} \tag{4.9}$$

where TP denotes the number of minority samples correctly identified; FP denotes the number of non-minority samples incorrectly identified as the minority class; TN denotes the number of non-minority samples correctly identified; FN denotes the number of minority samples incorrectly identified as the non-minority classes.

4.4.5 Experimental Results on Imbalance Datasets

In experiment I, experiments are conducted varying ϵ from 0.01 to 0.09 with 0.01 increments and it is observed that ϵ as 0.08 offers the most satisfactory classification results in terms of performance average and standard deviation (see Figure 4.8). Thus, the results reported below are with ϵ set to 0.08. With the synthetic samples generated from the five benchmark methods and the proposed EE-SMOTE, both the (MLP) and AdaBoost are implemented. Figure 4.9 illustrates the average performance of the two classifiers in terms of precision, recall, F1-score, and G-mean. The

reason for taking the average performance is to achieve a fair comparison as in (Li et al., 2020). It is observed that EE-SMOTE outperforms the comparison methods on precision (Figure 4.9(A)), recall (Figure 4.9(B)), F-measure (Figure 4.9(C)), and G-mean (Figure 4.9(D)). It is also observed that EE-SMOTE has the smallest standard deviations on all four metrics, indicating the robustness of the algorithm.

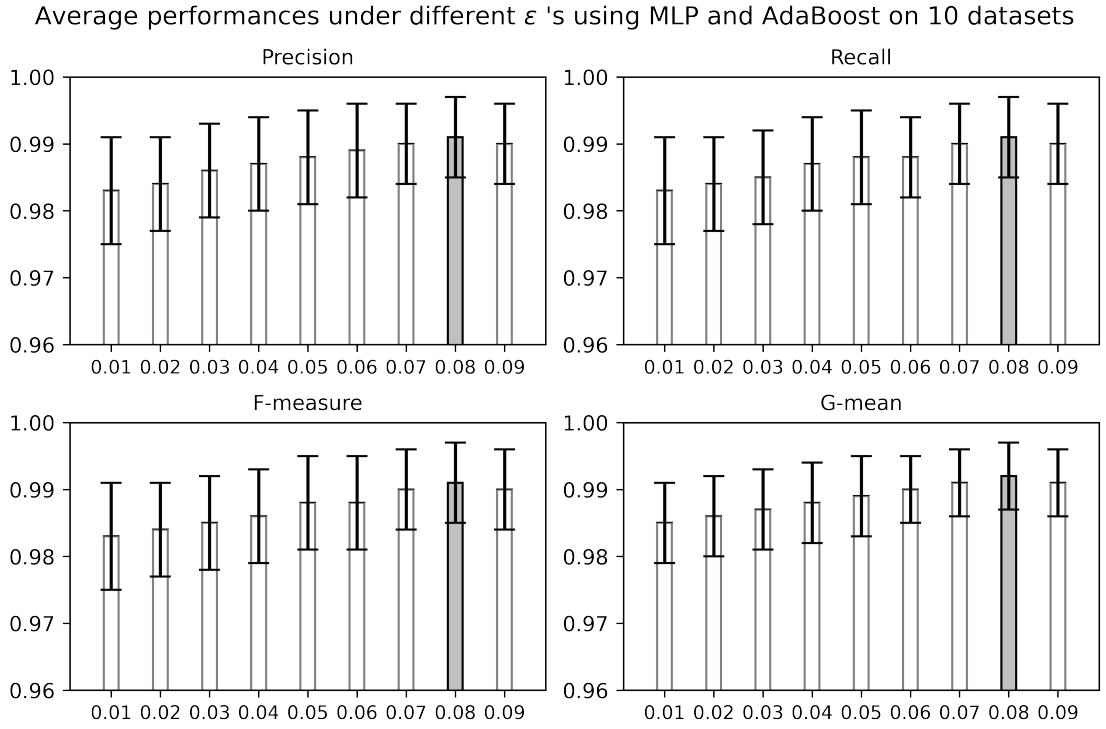

Figure 4.8: Average performances under different ϵ's from 0.01 to 0.09 using MLP and AdaBoost on 10 public datasets in terms of precision, recall, F-measure, and G-mean. The most satisfactory results occur at $\epsilon = 0.08$ (in grey)

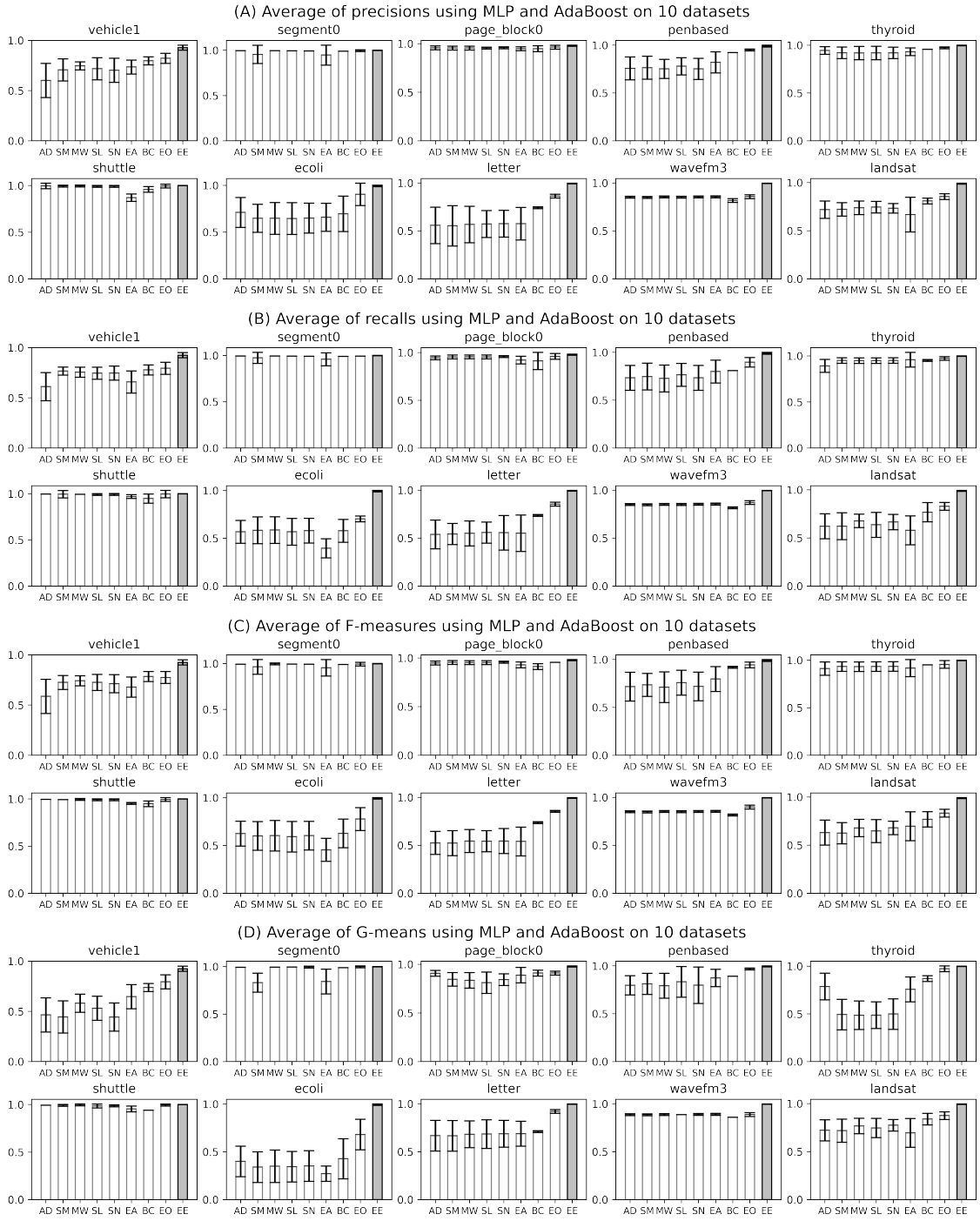

Figure 4.9: Average performances using MLP and AdaBoost on 10 public datasets. In each subplot, the x-axis indicates the methods (SMOTE (SM), MWMOTE (MW), SMTL (SL), SMENN (SN), EOS (EO), and EE-SMOTE (EE) under $\epsilon = 0.08$ (highlighted in grey)).

In experiment II, the performances of 5 classifiers using SMOTE and EE-SMOTE can be observed in Table 4.10. Similarly, EE-SMOTE has shown great superiority over SMOTE for FANCY dataset imbalanced learning. Hence, it is concluded that the EE-SMOTE sampling method is able to improve imbalanced classification problems for clinical study.

Table 4.10: Performances of 5 classifiers on FANCY dataset (SMOTE vsĖE-SMOTE)

Classifier	SMOTE	EE-SMOTE
LR	0.82	0.96
SVM-L	0.89	0.94
SVM-RBF	0.92	1.00
RF	0.88	1.00
XGBoost	0.78	0.96

4.5 Conclusion

This chapter proposes and validates an Eigen-Entropy sampling method. Eigen-Entropy (EE), defined through eigenvalues from a correlation magnitude matrix using multivariate datasets, is used as a decision-making criterion. It is able to automatically determine which samples and how many samples should be collected to construct a subset to support specific applications. Experimental results show that baselines construct by this method can reach comparable AFDD performances to those by an existing nonlinear mathematical model (SAX-WPM), and outperform those by another information-entropy-based method (MPMI) for the fault detection cases. This demonstrates the accuracy and robustness of the baseline. The imbalance learning case studies show that the proposed EE-SMOTE method outperforms five other methods reported in the literature.

Chapter 5

EIGEN-ENTROPY FOR CROSS-DATASET BUILDING FAULT DETECTION

5.1 Background

Buildings are complex and integrated systems consisting of multiple sensors, subsystems, and automatically controlled components. According to the United Nations Environment Programme, 36% of global energy use and 39% of energy-related carbon dioxide emission is attributed to building systems (International Energy Agency & United Nations Environment Programme, 2018). And, 30% of building energy usage is wasted (Pérez-Lombard et al., 2008) due to malfunctioning control, operation, and building equipment (Energy Conservation in Buildings & Communities programme, 1996). It is estimated additional energy consumption caused by some key building faults is anywhere between 0.37 to 17.96 EJ each year in the U.S. (Roth et al., 2004). One viable solution for an energy-efficient building system is automatic fault detection and diagnosis (AFDD) (Roth et al., 2004). From building design to the retrofit and commissioning process, understanding the reliability of a building and its energy faults is critical. Faults that degrade the performance of the entire building should be detected, diagnosed, and rectified, while in practice, significant follow-up and technical assistance to correct faults are required once detected and diagnosed. Over the past decades, many AFDD methods have been developed for component level and whole building level. Katipamula, Brambly, and Kim provide a comprehensive review and classification for methods used for Heating, Ventilation, and Air Conditioning (HVAC) system AFDD (Katipamula and Brambley, 2005a,b; Kim and Katipamula, 2018). Generally speaking, there are two groups of AFDD methods:

qualitative and quantitative model-based methods (such as rule-based and physics-model based); and process history-based methods (mostly various data-driven and machine learning-based methods). Qualitative and quantitative models are easy to understand and are popular among building engineers and researchers. However, the issues are the high development cost, low scalability due to their needs to be customized for each specific building/project (such as the associated physics-based models, rules, and thresholds). As a result, the market adoption rate has been low (Frank et al., 2018). Process history-based methods have therefore received great attention in recent years for their good scalability and low implementation cost. However, the performance of a process history-based method heavily relies on the data that the method is trained from, and it is recognized that the quality of the training data strongly affects the performance of process history-based AFDD tools (Omri et al., 2021).

Literature-reported AFDD methods are mostly developed and evaluated using simulated system data (Li and O'Neill, 2018; Shi and O'Brien, 2019). This is due to the difficulties of obtaining and analyzing real building data. Implementing faults and obtaining data that contain fault impacts in real buildings are already challenging. Cleaning and analyzing real building data to obtain "ground truth" is even more arduous since unexpected naturally occurred faults could exist in the system and cause abnormalities or complicate (sometimes even eliminate) the fault impacts expected from the artificially implemented faults.

Strategies that are only tested using simulated data might experience difficulties when applied to real buildings due to the two following issues: 1) data quality issues with typical building automation system data (missing data, noise, sensor faults, sensor accuracy, etc.), and 2) inherited differences between simulated data and real data, in terms of fault symptoms and data characteristics. After all, no model is perfect to

represent reality completely. These challenges exist for all AFDD strategies but are more significant for process history-based (data-driven) methods since their performance depends on the data quality. Consequently, the preliminary research (Huang et al., 2022) studies two sets of experiments to seek answers to this question. The first experiment is to evaluate data-driven fault detection strategies on real and simulated building data separately. The results show that the fault detection performances are not affected by fault detection strategies, sizes of training data, and the number of cross-validation folds when training and blind test data come from the same data source, namely, simulated or real building data. The second experiment is to conduct a cross-dataset study, that is, develop the model using simulated data and tested on real building data. The results indicate the model trained on simulated data is not generalized to be applied for real building data for fault detection. Kolmogorov-Smirnov Test is conducted to confirm that there exist statistical differences between the simulated and real building data and identify a subset of features with similarities between the two datasets. Further, using the subset of the feature, cross-dataset experiments show fault detection improvements on most fault cases. The conclusion is that AFDD strategies trained by simulations may not achieve satisfactory effectiveness when directly applied to real building systems, mainly because measurements from simulation system data may still contain different information from real-world scenarios (More details can be referred to the Appendix A). As simulations are validated by real-world domain knowledge, it is assumed that the relationships among various building components remain consistent between real-world and simulated datasets. This assumption prompts to explore leveraging these relationships to enhance the AFDD system.

Consequently, this chapter presents an innovative method for feature extraction from graphs to facilitate the implementation of cross-dataset AFDD strategies for

HVAC systems. Specifically, this approach involves extracting these relationships by training the model on simulated data and applying these learned relationships to real data, utilizing graph techniques.Graph techniques involve a set of interconnected components that are used to depict connections and tackle complexities in systems. They are effective for capturing relationships from simulated data and applying them to real data analysis in this scenario.The graph comprises nodes representing features and edges symbolizing the connections between them, reflecting relationships or interdependencies. In this scenario, graph structures are generated 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 be utilized for 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.

5.2 Methodology

In this section, some concepts from graph theory are introduced. Then the proposed algorithm presents how to create graphs and extracted corresponding EEs as features for both simulation and real building data respectively.

5.2.1 Graph

In graph theory, a graph is a mathematical structure used to present pairwise relationship between objects (Diestel, 2017). 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 (in the context of building systems, m features), it can be represented by an $m \times m$ adjacency matrix, where elements of the matrix indicate edges, as illustrated in Figure 5.1.

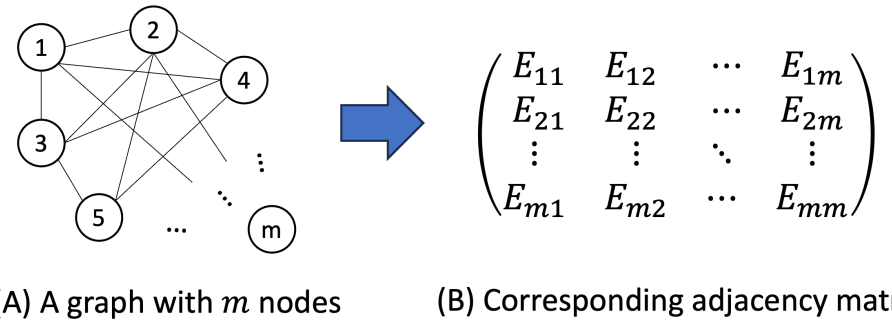

Figure 5.1: An example of graph and corresponding adjacency matrix

As previously mentioned, if the elements in a building system are viewed as nodes and the connections between elements as edges, then the relationships between pairs of building components can be represented by an adjacency matrix. This matrix helps visualize how the components are interconnected and how they influence each other. These connections reflect the correlation values among the features.

In the building domain, the interconnections between components are often assessed through correlations (Liang et al., 2021). Subsequently, the graph's adjacency matrix resembles the correlation magnitude matrix when using 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 (Huang et al., 2022). Hence, EE

is preferred for demonstrating correlation patterns rather than using direct correlation values derived 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.

5.2.2 EE-graph-based Feature Extraction for Cross-dataset AFDD

Algorithm 3 presents the EE-Graph-based feature extraction approach on both real and simulation building data. When presented with a fault test case containing fault and fault-free datasets obtained from an actual building system, the first step involves selecting a fault test case from a simulation that matches this real-world scenario in terms of fault type and severity level. Subsequently, it is essential to verify the common features shared by these two cases, denoted as m, as outlined (Algorithm 3, line 1). This is because designs of simulation building systems may differ from real ones in terms of sensors or zone areas even though they are the same type of buildings (commercial or residential). After identifying the matched simulation fault test case, it is needed to select those fault and fault-free samples from simulation and real data that match the operation mode between these two building systems (Algorithm 3, line 2), since any building systems will operate differently under normal conditions in response to the environment (i.e., the position of outdoor air damper will be open at 40 % of position during the summer period). Then each feature is normalized in fault and fault-free data for simulation and real building, respectively (Algorithm 3, lines 3-4).

In this research, data from simulating building performance is utilized to train a data-driven AFDD strategy. This involves acquiring knowledge of graph structures from simulations during the training phase and subsequently applying it to an actual

building system. Given n_{simf} fault and n_{simfr} fault-free samples in simulation, the first step is to obtain N_{simf} and N_{simfr} fault and fault-free snapshot windows by dividing these numbers using the window size of W(Algorithm 3, line 5). Subsequently, derive M feature correlations by computing correlations for every pair from m features using each snapshot window of N_{simf} and N_{simfr}, respectively (Algorithm 3, line 6)., and then conduct two-sample t test on each of M feature correlations. This helps to identify significant feature correlations that can distinguish between fault and fault-free snapshot windows to form a set of $M^{'}$ significant edges (Algorithm 3, line 7). Based on the t-statistics of these connections, these connections are divided into $(q-1)$ groups based on the q quantiles (Algorithm 3, line 8) and generate P graphs from all groups (Algorithm 3, line 9). For each of P graphs, EE is calculated by Eq (3.12) and thus for each of fault and fault-free snapshot windows, there are P EEs for each matched simulation fault test case. Afterwards, P graph structures learned from the matched simulation fault test case can be replicated for the real fault test case, and P EEs can be extracted (Algorithm 3, lines 11-13). Ultimately, the obtained EEs are generated for both the actual and the corresponding simulated fault test scenarios to substantiate the investigation (Algorithm 3, line 14). Figure 5.2 also provides a corresponding flowchart of the whole process.

Algorithm 3 EE-graph-based feature extraction

Input: A fault test case from a real building system containing a pair of fault and fault-free real datasets, each real dataset consisting of m_r features, and multiple fault test cases from a simulation building system, each case containing a pair of fault and fault-free simulated datasets, each simulated dataset consisting of m_s features

Output: Extracted EEs for the real and the matched simulation fault test cases

Initialization

1: Select the simulated fault test case matching the real one according to the fault type, fault severity level and operation mode, and identify m common features.

2: Select n_{simf} and n_{simfr} samples from the simulation fault and fault-free datasets, and n_{realf} and n_{realfr} samples for the real fault and fault-free datasets by matching the operation mode between simulation and real building systems, respectively.

3: Normalize each feature for simulation building samples (both fault and fault-free).

4: Repeat the Step 3 for real building samples (both fault and fault-free).

Training phase: feature extraction on the matched simulation fault test case

5: Obtain N_{simf} and N_{simfr} fault and fault-free snapshot windows in simulation fault test case for by dividing the total number of samples using the window size of W.

6: For each pair from the m features, calculate feature correlations using samples present in each of fault and fault-free snapshot windows, respectively. Therefore, M feature correlations can be calculated for both N_{simf} and N_{simfr} snapshot windows corresponding to faults and fault-free scenarios.

7: Conduct two-sample t-test on each of M feature correlations, and obtain corresponding t-statistics and p-values. Select those feature correlations that can distinguish between fault and fault-free samples from each of the snapshot windows (i.e. whose p-value $< \alpha = 0.05$). These selected $M^{'}$ correlations form a set of significant edges, $\mathbf{E} = \{\mathbf{e_1}, \mathbf{e_2}, \ldots, \mathbf{e_{M'}}\}$, and the set of corresponding t-statistics for $\mathbf{E}$ is $\mathbf{T} = \{\mathbf{t_1}, \mathbf{t_2}, \ldots, \mathbf{t_{M'}}\}$.

Algorithm 3 EE-graph-based feature extraction (continued)

8: Group significant edges from $\mathbf{E}$ into q clusters by the q quantiles of absolute value of t-statistics from $\mathbf{T}$.

9: For each cluster, construct graph(s) using edges. Then obtain P graphs in total for both simulation fault and fault-free snapshot windows.

10: Compute the EEs for each of the graphs represented by P using Equation 3.12 for both simulation fault and fault-free snapshot windows., $\mathbf{EE^{simf}}$ and $\mathbf{EE^{simfr}}$, where $\mathbf{EE^{simf}} = \{\mathbf{EE^{simf}}(i)\}$, $\mathbf{EE^{simf}}(i) = \{EE_{ip}^{simf}\}$, $i = 1, \ldots, N_{simf}$; $\mathbf{EE^{simfr}} = \{\mathbf{EE^{simfr}}(j)\}$, $\mathbf{EE^{simfr}}(j) = \{EE_{jp}^{simfr}\}$, $j = 1, \ldots, N_{simfr}$; $p = 1, ..., P$.

Test phase: feature extraction on real building data

11: Repeat Step 4 to obtain N_{realf} and N_{realfr} fault and fault-free snapshot windows in real fault test case respectively.

12: Follow procedures of Step 9 to construct P graphs for both real fault and fault-free snapshot windows.

13: Repeat Step 10 to obtain EEs for real fault and fault-free snapshot windows, $\mathbf{EE^{realf}}$ and $\mathbf{EE^{realfr}}$,where $\mathbf{EE^{realf}} = \{\mathbf{EE^{realf}}(i)\}$, $\mathbf{EE^{realf}}(i) = \{EE_{ip}^{realf}\}$, $i = 1, \ldots, N_{realf}$; $\mathbf{EE^{realfr}} = \{\mathbf{EE^{realfr}}(j)\}$, $\mathbf{EE^{realfr}}(j) = \{EE_{jp}^{realfr}\}$, $j = 1, \ldots, N_{realfr}$; $p = 1, ..., P$.

14: Return $\mathbf{EE^{simf}}$ and $\mathbf{EE^{simfr}}$ for the simulated fault test case, $\mathbf{EE^{realf}}$ and $\mathbf{EE^{realfr}}$ for the real fault test case, respectively.

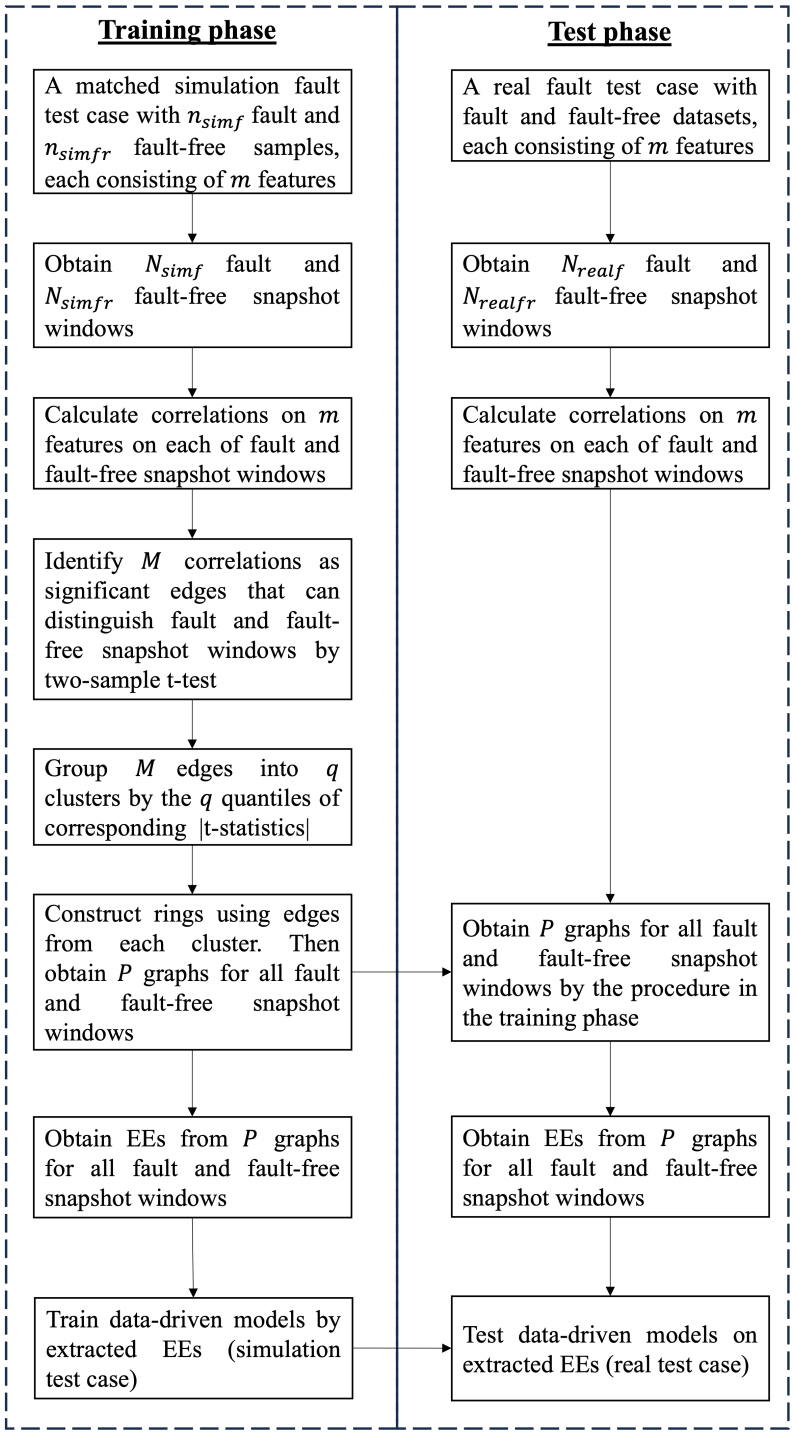

Figure 5.2: Flowchart of cross-dataset AFDD

5.3 Experiments

5.3.1 Datasets

Simulation building datasets used for training in this study are generated by Lawrence Berkeley National Laboratory (LBNL) (Granderson et al., 2022) 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. Figure 5.3 illustrates this building structure and floor layout. It can be noticed thatthe conditioned floor space consists of a single interior zone and four perimeter zones, where AHU distributes conditioned air by five VAV boxes. Figure 5.4 shows the schematic diagram of the corresponding simulated AHU. Main AHU components include the supply and return fans (both with a variable frequency driver), cooling and heating coils and control valves, and outdoor air and return air dampers, while main VAV components include reheating coil and control valves, and terminal damper.

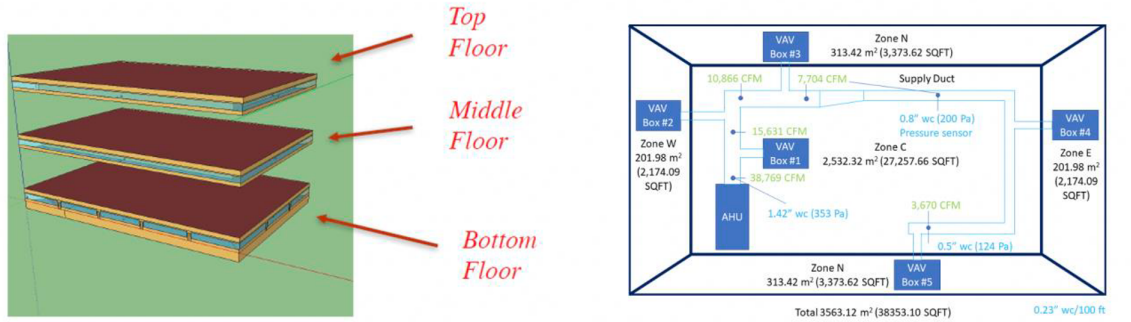

Figure 5.3: DOE large office reference building and the one floor layout (adapted from (Granderson et al., 2022))

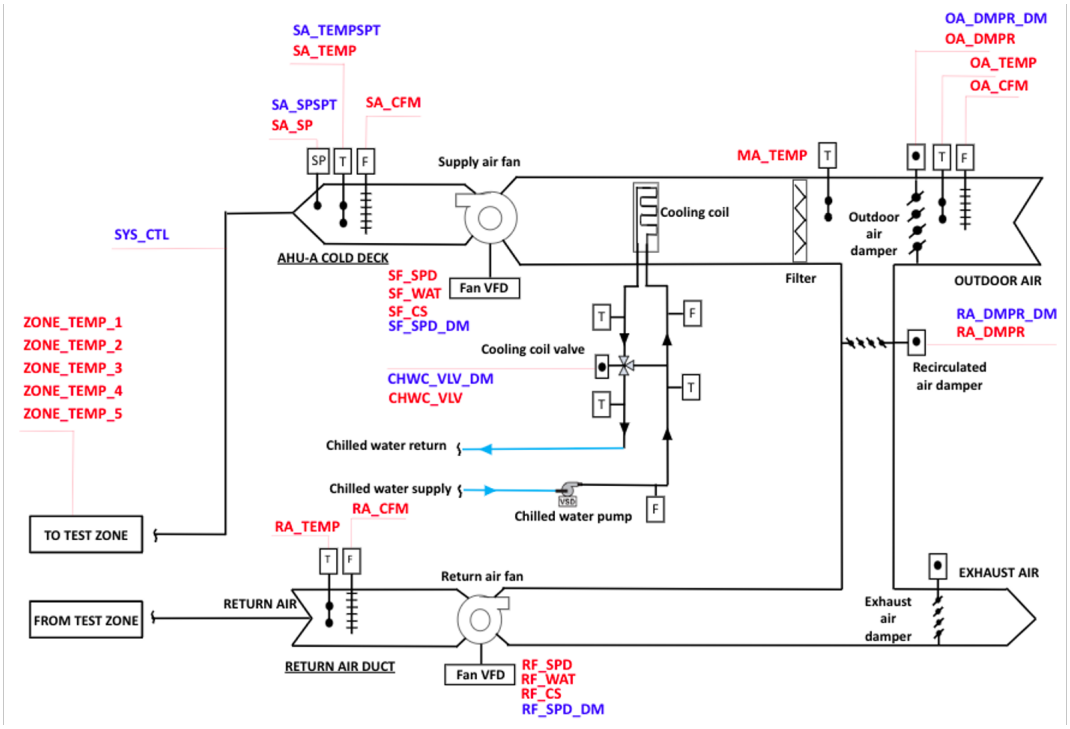

Figure 5.4: Schematic diagram of a single-duct AHU (adapted from (Granderson et al., 2022))

Real building datasets used for test are generated from the ASHRAE 1312 research project (Wen and Li, 2011; Li and Wen, 2007). These datasets are collected from a laboratory building that is set up like a small office building whose layout is shown in Figure 5.5. As can be 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. Figure 5.6 demonstrates the main components of the AHU system, which consists of supply air and return air fans; preheat, cooling, and

heating coils; heating and cooling control valves; recirculated air, exhaust air, and outdoor air dampers.

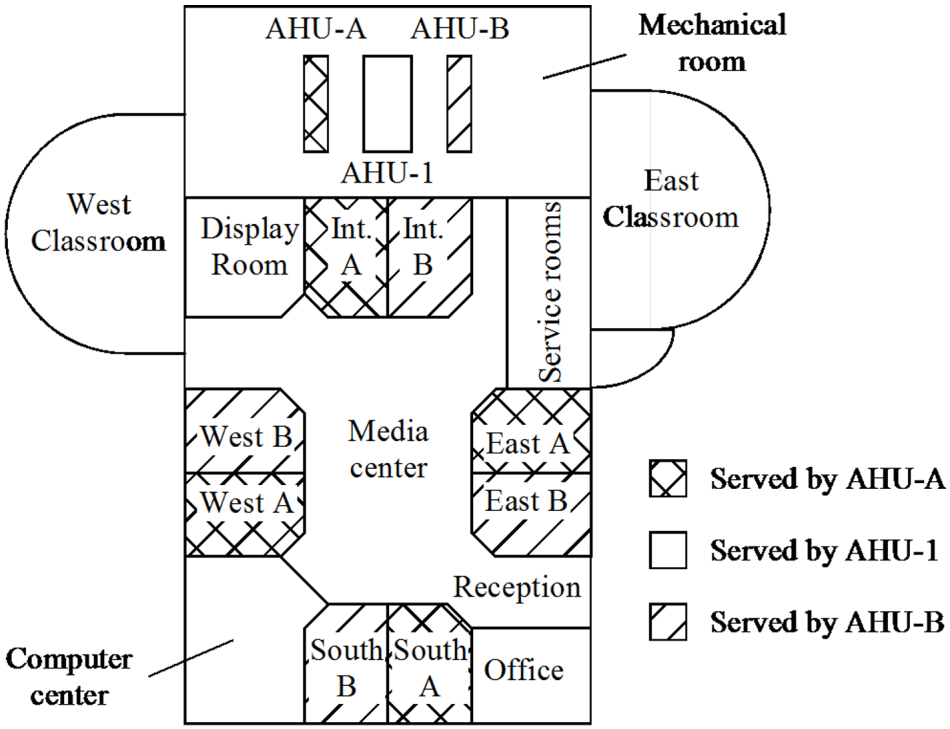

Figure 5.5: Energy resource station (ERS) experimental setup (adapted from (Wen and Li, 2011))

Notice that in reality, the operating conditions of an AHU are subject to the outdoor air temperature and humidity due to seasonal variations. Therefore, as shown in Figure 5.7, there are four operational modes for an AHU, namely, mechanical heating mode (Mode 1), economizer cooling mode (Mode 2), mechanical and economizer cooling mode (Mode 3), and mechanical cooling mode (Mode 4). More detailed descriptions about these modes can be found in (Granderson et al., 2022).

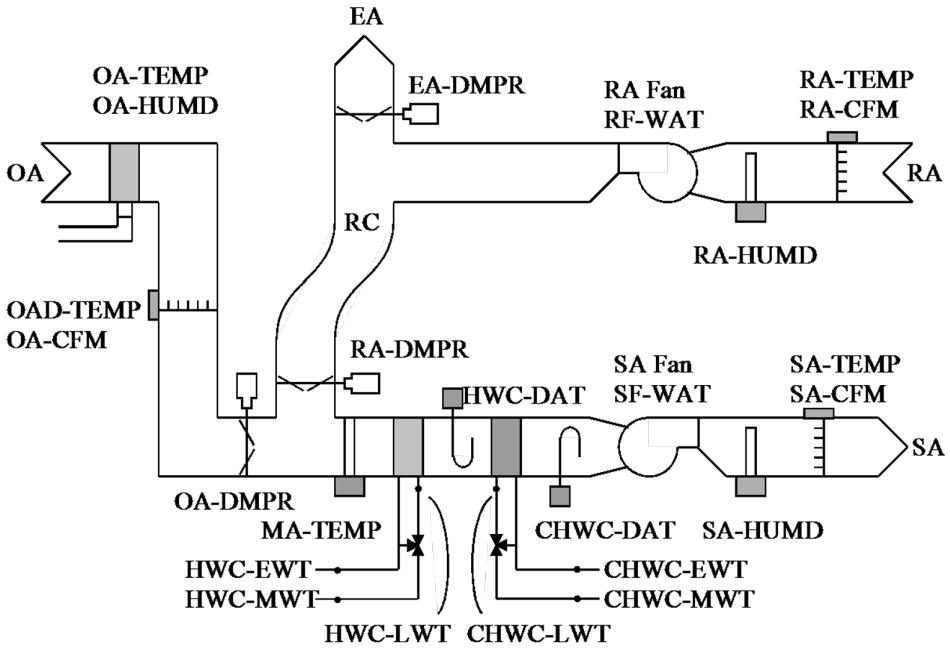

Figure 5.6: Schematic diagram of an AHU in ERS (adapted from (Wen and Li, 2011))

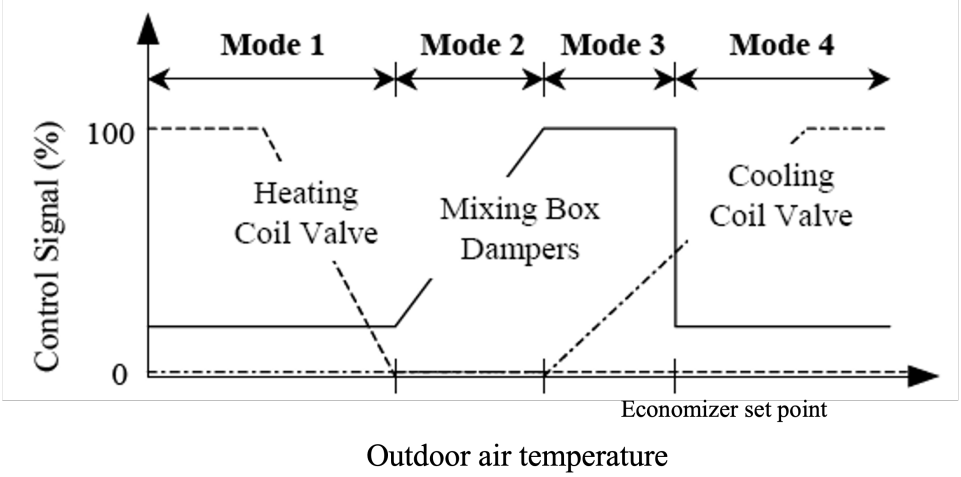

Figure 5.7: Operating modes of an AHU (adapted from (Wen and Li, 2011))

As illustrated earlier, there is a need to confirm simulation fault test cases that match studied real fault test cases according to the similar fault types, fault severity levels and operation modes. It is feasible to link a simulated fault test case to multiple

real cases. To illustrate this point, five pairs of cases have been recognized, with each pair comprising a real fault test case and its corresponding simulated counterpart. (see Table 5.1), all operating under the mechanical cooling mode (Mode 4). Among these cases, three faults are regarding cooling coil valve, and the rest are regarding OA damper. Please notice that all simulation fault test cases share the same fault-free datasets because the data is generated with the same environment (e.g., weather) from the same normally operating virtual testbed; however, fault-free datasets may slightly differ among all real fault test cases because they are collected under the different environments. Finally, since the size of the studied simulation office building is different from that of real one, there is a need to identify common features between these two fault test cases, and the results are shown in Table 5.2.

Table 5.1: Summary of paired fault test cases for cross-dataset AFDD

Case No.	Training datasets (LBNL simulation)			Test datasets (ASHRAE real building data)		
	Fault test case	# samples		Fault test case	# samples	
1	Cooling Coil Valve	n_{simf}	172k	Cooling Coil Valve	n_{realf}	600
	Stuck 25% Open	n_{simfr}	171k	Stuck 15% Open	n_{realfr}	600
2	Cooling Coil Valve	n_{simf}	168k	Cooling Coil Valve	n_{realf}	600
	Stuck 50% Open	n_{simfr}	171k	Stuck 65% Open	n_{realfr}	600
3	OA Damper Stuck	n_{simf}	170k	OA Damper Stuck	n_{realf}	600
	10% Open	n_{simfr}	171k	Fully Closed	n_{realfr}	600
4	OA Damper Stuck	n_{simf}	118k	OA Damper Stuck	n_{realf}	600
	75% Open	n_{simfr}	171k	45% Open	n_{realfr}	600
5	OA Damper Stuck	n_{simf}	118k	OA Damper Stuck	n_{realf}	600
	75% Open	n_{simfr}	171k	55% Open	n_{realfr}	600

*n_{simf}: simulation fault samples; n_{simfr}: simulation fault-free samples.

n_{realf}: real fault samples; n_{realfr}: real fault-free samples.

Table 5.2: Summary of features shared between simulation and real fault test case

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

5.3.2 Benchmark Methods

For comparison purposes, the graph neural network (GNN) is employed as the benchmark algorithm. GNN (Wu et al., 2023) is a type of neural network that processes data structured as graphs, enabling the capturing of complex relationships and interdependencies between nodes. In this study, the GNN model extracts features from the created graphs for both simulated and actual data, which are then compared with the detection performance obtained using the proposed approach. Besides, detection performances achieved with raw features and significant edges are included for comparisons as well (Note raw features mean the original features without any pre-processing applied, and significant edges mean those identified correlations in Alogrithm 3 Line 6).

5.3.3 Evaluations

Machine learning models, such as decision tree (DT) (Yan et al., 2016), random forest (RF) (Wang et al., 2021), and support vector machine (Li et al., 2016), have been widely applied to support building AFDD. In this research, two classifiers, DT and RF, are used because they are non-parametric and are capable of modeling complex relationships. Even though SVM can provide good classification performance, it may require higher computational expenses for training when handling with the large volume of data (Cervantes et al., 2008).

According to (Lin et al., 2020), four metrics, AUC, recall, precision, and F-measure (F) are commonly used as data-driven AFDD performance evaluations. The latter three metrics are calculated as below:

$$Recall = \frac{TP}{TP + FN} \tag{5.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{5.2}$$

$$F = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{5.3}$$

Here, true positive (TP) means the number of fault samples/snapshot windows correctly identified; true negative (TN) means the number of fault-free samples/snapshot windows correctly identified; false positive (FP) means the number of fault-free samples/snapshot windows incorrectly identified as fault ones; false negative (FN) means the number of fault samples/snapshot windows incorrectly identified as fault-free ones.

Recall is defined as the number of correctly identified fault samples over total number of fault samples (see Eq (5.1)). Precision is defined as the number of true

fault samples over total number of predicted fault samples (see Eq (5.2)). F-measure is a combined metric derived from recall and precision (see Eq (5.3)). Three metrics emphasize more the detection of true fault samples. All these metrics range from 0 to 1. The greater these metrics are, the more effective the performance of the AFDD model.

AUC, or area under the curve, is to measure the performance of a classifier at various thresholds (McClish, 1989), representing the degree of separability archived by the classifier. The range of AUC is 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 (McClish, 1989); if AUC ≥ 0.6, then it is said to be acceptable discrimination (Yang and Berdine, 2017).

5.3.4 Experimental Results

Snapshot windows for every paired fault test case, simulation, and real building data, comprising both faulty and fault-free samples, are acquired using a window size of 30, as recommended in. (Chen et al., 2022a) ($W = 30$ in Algorithm 3, line 5). During the training phase, significant edges are identified to distinguish fault and fault-free snapshot windows. Tables 5.3 and 5.4 summarize the information about the number of snapshot windows and significant edges identified, respectively.

Note that in Algorithm 3, the number of quantiles, q, is needed to group significant edges into clusters so as to construct graphs in each cluster. Since there is no set rule about this value of the quantiles, three commonly used ones in statistics (Witte and Witte, 2017), namely tertiles (3-quantiles), quartiles (4-quantiles) and quintiles (5-quantiles), are tested. Table 5.5 presents the test outcomes achieved with DT and RF models trained using features extracted from simulated data and applied to

real building snapshot windows. It has been noted that the test outcomes for quintiles surpass those for tertiles and quartiles, as all instances exhibit an AUC ≥ 0.6 when employing DT or RF. This suggests that both classifiers can effectively differentiate between fault snapshot windows and those without faults. In this instance, other measures such as recall, precision, and F-measure surpass at least two scenarios. Moreover, on an average, extracted features under quintiles are fewer than those under tertiles and quartiles (see Table 5.6). Consequently, results under quintiles is used to compare against another approaches.

Table 5.3: Summary of the number of snapshot windows for paired fault test cases

Case No.	Training datasets (LBNL simulation)			Test datasets (ASHRAE real building data)		
	Fault test case	N_{simf}	N_{simfr}	Fault test case	N_{realf}	N_{realfr}
1	Cooling Coil Valve Stuck 25% Open	5.7k	5.7k	Cooling Coil Valve Stuck 15% Open	20	20
2	Cooling Coil Valve Stuck 50% Open	5.6k	5.7k	Cooling Coil Valve Stuck 65% Open	20	20
3	OA Damper Stuck 10% Open	5.7k	5.7k	OA Damper Stuck Fully Closed	20	20
4	OA Damper Stuck 75% Open	5.7k	5.7k	OA Damper Stuck 45% Open	20	20
5	OA Damper Stuck 75% Open	5.7k	5.7k	OA Damper Stuck 55% Open	20	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 5.4: Number of significant edges identified for training fault test cases

Training fault test case	# Significant edges
Cooling Coil Valve Stuck 25% Open	26
Cooling Coil Valve Stuck 50% Open	49
OA Damper Stuck 10% Open	39
OA Damper Stuck 75% Open	54

Table 5.5: Results on real snapshot windows under three quantiles using DT and RF (AUC $\geq$ 0.60 underlined and bold)

DT - Decision Tree												
Case No.	Tertiles				Quartiles				Quintiles			
	AUC	Recall	Precision	F	AUC	Recall	Precision	F	AUC	Recall	Precision	F
1	0.45	0.00	0.00	0.00	0.43	0.10	0.29	0.15	**0.68**	0.90	0.62	0.73
2	**0.95**	0.90	1.00	0.95	0.18	0.25	0.22	0.23	**0.63**	0.25	1.00	0.40
3	**0.83**	0.65	1.00	0.79	0.50	0.65	0.50	0.57	**0.90**	0.80	1.00	0.89
4	**0.95**	0.90	1.00	0.95	**1.00**	1.00	1.00	1.00	**0.93**	0.85	1.00	0.92
5	**0.90**	0.80	1.00	0.89	**0.98**	0.95	1.00	0.97	**0.88**	0.75	1.00	0.86

RF - Random Forest												
Case No.	Tertiles				Quartiles				Quintiles			
	AUC	Recall	Precision	F	AUC	Recall	Precision	F	AUC	Recall	Precision	F
1	0.45	0.00	0.00	0.00	**0.93**	0.90	0.95	0.92	**0.70**	0.90	0.64	0.75
2	**0.98**	0.95	1.00	0.97	**0.63**	0.25	1.00	0.40	**0.60**	0.20	1.00	0.33
3	**0.85**	0.70	1.00	0.82	0.55	0.65	0.54	0.59	**0.85**	0.70	1.00	0.82
4	**0.98**	0.95	1.00	0.97	**1.00**	1.00	1.00	1.00	**0.88**	0.75	1.00	0.86
5	**0.95**	0.90	1.00	0.95	**0.98**	0.95	1.00	0.97	**0.80**	0.60	1.00	0.75

Table 5.6: Summary of number of extracted features under three quantiles

Case No.	Tertiles	Quartiles	Quintiles
1	4	4	2
2	22	14	11
3	9	3	4
4	29	14	15
5	29	14	15
Average	19	10	9

Figure 5.8 displays the detection results obtained by DT when utilizing raw features, significant edges, GNN features, and EEs across quintiles. Using DT as the detection model, it is evident that employing EE features (the proposed method) results in AUC values exceeding 0.60 for all four test cases. Conversely, the utilization of other features (e.g., raw features, significant edges and GNN features),do not yield satisfactory AUC values. Specifically, in Case 1, EEs achieve AUC of 0.68, recall of 0.90, precision of 0.62, and F-measure of 0.73, whereas raw features attain AUC of 0.50, recall of 0.00, precision of 0.00 and F-measure of 0.00; significant edges attain AUC of 0.30, recall of 0.00, precision of 0.00 and F-measure of 0.00; GNN features attain AUC of 0.50, recall of 1.00, precision of 0.50, and F-measure of 0.67. In Case 2, EEs achieve AUC of 0.63, recall of 0.25, precision of 1.00, and F-measure of 0.40, whereas raw features attain AUC of 0.50, recall of 0.00, precision of 0.00 and F-measure of 0.00; significant edges attain AUC of 0.55, recall of 0.10, precision of 1.00 and F-measure of 0.18; GNN features attain AUC of 0.50, recall of 0.00, precision of 0.00, and F-measure of 0.00. In Case 3, EEs achieve AUC of 0.90, recall of 0.80, precision of 1.00, and F-measure of 0.89, whereas raw features attain AUC of 0.50, recall of 0.11, precision of 0.50 and F-measure of 0.18; significant edges attain AUC

of 0.90, recall of 0.80, precision of 1.00 and F-measure of 0.89; GNN features attain AUC of 0.50, recall of 1.00, precision of 0.50, and F-measure of 0.67; In Case 4, EEs achieve AUC of 0.93, recall of 0.85, precision of 1.00, and F-measure of 0.92, whereas raw features (Raw) attain AUC of 0.50, recall of 0.00, precision of 0.00 and F-measure of 0.00; significant edges (Edge) attain AUC of 0.93, recall of 0.85, precision of 1.00 and F-measure of 0.92; GNN features attain AUC of 0.50, recall of 0.00, precision of 0.00, and F-measure of 0.00; In Case 5, EEs achieve AUC of 0.88, recall of 0.75, precision of 1.00, and F-measure of 0.86, whereas raw features (Raw) attain AUC of 0.50, recall of 0.00, precision of 0.00 and F-measure of 0.00; significant edges (Edge) attain AUC of 0.93, recall of 0.85, precision of 1.00 and F-measure of 0.92; GNN features attain AUC of 0.50, recall of 0.00, precision of 0.00, and F-measure of 0.00.

Figure 5.9 displays the detection results obtained by RF when utilizing raw features, significant edges, GNN features, and EEs across quintiles. Similarly, using RF as the detection model, it is evident that employing EE features (the proposed method) results in AUC values ≥ 0.60 for all four test cases. Conversely, the utilization of other features (e.g., raw features, significant edges and GNN features),do not yield satisfactory AUC values. Specifically, in Case 1, EEs achieve AUC of 0.70, recall of 0.90, precision of 0.64, and F-measure of 0.75, whereas raw features attain AUC of 0.50, recall of 0.00, precision of 0.00 and F-measure of 0.00; significant edges attain AUC of 0.48, recall of 0.00, precision of 0.00 and F-measure of 0.00; GNN features attain AUC of 0.50, recall of 0.00, precision of 0.00, and F-measure of 0.00. In Case 2, EEs achieve AUC of 0.60, recall of 0.20, precision of 1.00, and F-measure of 0.33, whereas raw features attain AUC of 0.50, recall of 0.00, precision of 0.00 and F-measure of 0.00; significant edges attain AUC of 0.63, recall of 0.25, precision of 1.00 and F-measure of 0.40; GNN features attain AUC of 0.98, recall of 1.00, precision of 0.95, and F-measure of 0.98. In Case 3, EEs achieve AUC of 0.85, recall of 0.70,

precision of 1.00, and F-measure of 0.82, whereas raw features attain AUC of 0.49, recall of 0.00, precision of 0.07 and F-measure of 0.00; significant edges attain AUC of 0.88, recall of 0.75, precision of 1.00 and F-measure of 0.86; GNN features attain AUC of 0.55, recall of 1.00, Precision of 0.53, and F-measure of 0.69; In Case 4, EEs achieve AUC of 0.88, recall of 0.75, precision of 1.00, and F-measure of 0.86, whereas raw features attain AUC of 0.50, recall of 0.00, precision of 0.00 and F-measure of 0.00; significant edges attain AUC of 0.93, recall of 0.85, precision of 1.00 and F-measure of 0.92; GNN features attain AUC of 0.50, recall of 0.00, precision of 0.00, and F-measure of 0.00; In Case 5, EEs achieve AUC of 0.80, recall of 0.60, precision of 1.00, and F-measure of 0.75, whereas raw features attain AUC of 0.50, recall of 0.00, precision of 0.00 and F-measure of 0.00; significant edges attain AUC of 0.88, recall of 0.75, precision of 1.00 and F-measure of 0.86; GNN features attain AUC of 0.50, recall of 0.00, precision of 0.00, and F-measure of 0.00.

In conclusion, both DT or RF models trained with EE features are capable of identifying faults in all fault test cases with superior AUC values (AUC $\geq$ 0.60) compared to models trained with other features. This indicates that the suggested method is successful in enhancing fault detection across datasets.

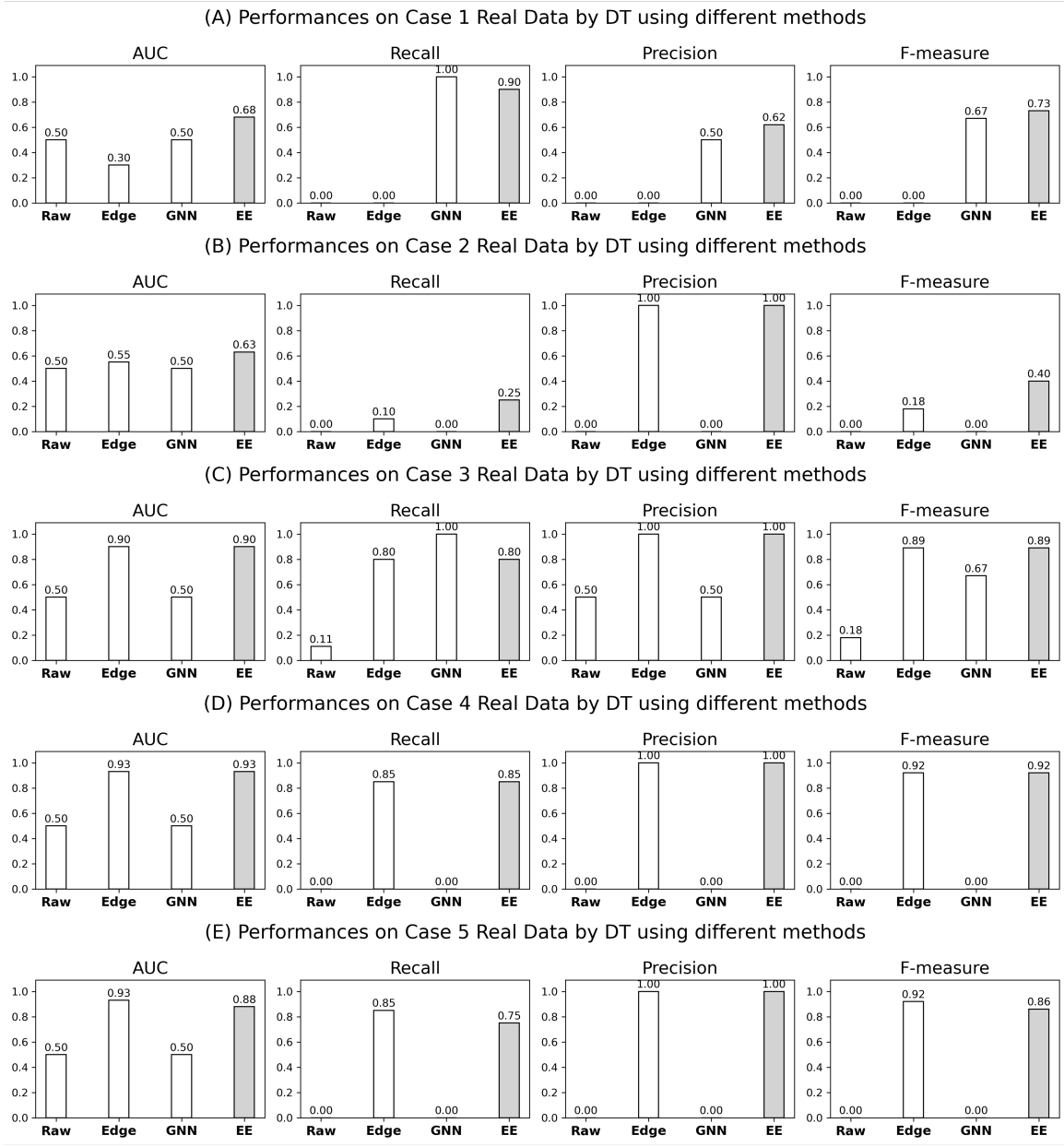

Figure 5.8: Comparisons of DT performances for 5 fault test cases using different features (Raw: Original features; Edges: Significant edges; GNN: extracted GNN features; EE: features by the proposed method (highlighted in grey))

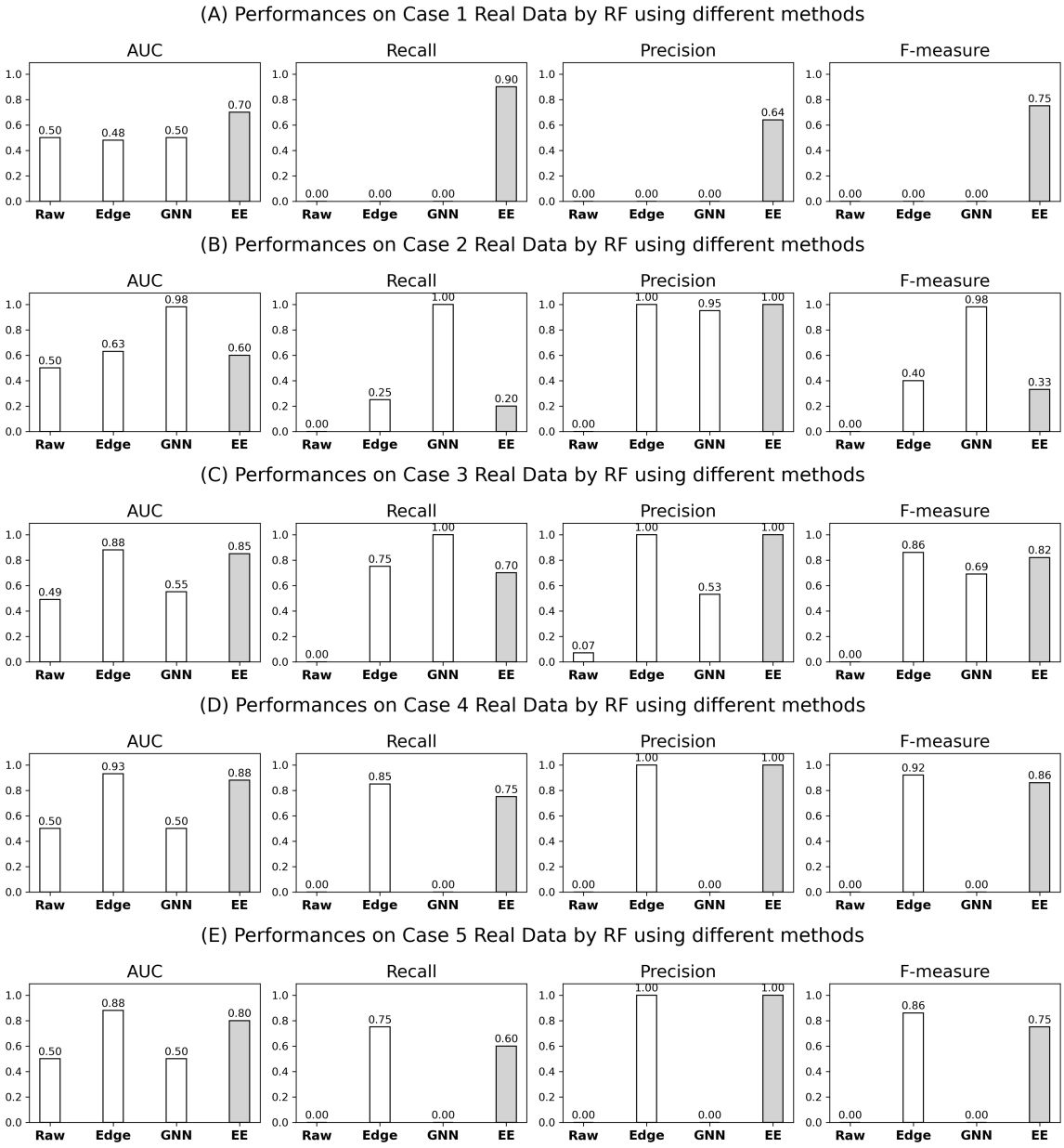

Figure 5.9: Comparisons of RF performances for 5 fault test cases using different features (Raw: Original features; Edges: Significant edges; GNN: extracted GNN features; EE: features by the proposed method (highlighted in grey))

5.4 Conclusion

This chapter presents 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, five 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 the proposed method from simulation data can notably enhance fault detection performance in real-world building fault scenarios. Additionally, the proposed method outperforms the benchmark and other approaches for the most fault cases, showcasing the effectiveness and generalizability of the proposed approach for analyzing building HVAC systems across different datasets. This delivers two important messages: (1) graph-structured data can be applied to support for AFDD by characterizing the coupling effects among components in the building systems; (2) EE, as a type of information entropy, can be used as features to support for machine learning tools.

Chapter 6

EIGEN-ENTROPY FOR FAULT DIAGNOSIS BAYESIAN NETWORK CONSTRUCTION

6.1 Background

Building Heating, Ventilation and Air conditioning (HVAC) systems are complex with a variety of sensors, subsystems and automatically controlled components. According to the United Nations Environment Programme, approximately 135 EJ operational energy demand and 10 Gt energy-related carbon dioxide emission are attributed to the building systems in 2021 (United Nations Environment Programme, 2022). It is also reported that 30% of this energy usage (Pérez-Lombard et al., 2008) is wasted due to malfunctioning sensors and components in the HVAC systems (Energy Conservation in Buildings & Communities programme, 1996; R.Brambley and Katipamula, 2009). Automatic fault detection and diagnosis (AFDD) technologies thus are vital to ensure satisfactory building performances, especially from the aspect of energy efficiency (Roth et al., 2004). Field studies and practices indicate that AFDD technologies can not only achieve up to 20% building energy savings (Piette et al., 2001) but also improve equipment lifecycles and indoor comforts (Granderson et al., 2020; CIBSE, 2007; Ginestet et al., 2008) .

Modern building HVAC systems typically include a set of multiple, highly coupled subsystems such as cooling/heating plant, primary air distribution, and terminal air distribution subsystems. Due to the coupling effect among building components, a fault occurring in one equipment or subsystem may propagate and influence other equipment or subsystems (Yan et al., 2017; Cauchi et al., 2018). Hence, component-

level AFDD methods may not be efficient and suitable solutions to the root cause analysis for cross-level faults, i.e., faults causing adverse effects across multiple components and subsystems (Chen et al., 2022). Chen et al. (2022) has provided an example of a chiller supply water temperature sensor bias fault (e.g., sensor reading higher than actual temperature) in the chiller plant which would cause the cooling valve open position in a downstream air handling unit (AHU) to be lower than normal. In this case, a component-level AFDD tool that only monitors the AHU might result in false alarms such as a cooling coil valve fault or a supply air temperature sensor fault. Hence, a root cause analysis is necessary to ensure the correct diagnosis of cross-level faults.

Compared with fault detection studies, much fewer fault diagnosis/root cause analysis studies exist (Chen et al., 2023). A root cause analysis has shown its importance to improve quality assurance, reliability and performance in many fields, such as power systems (Wang et al., 2018), manufacturing (Lokrantz et al., 2018). Bayesian networks (BNs) have been extensively studied as a root cause analysis technique. For example, Wang et al. (2018) conduct an analysis on root causes of occurring alarms in thermal power plants based on posterior probability from BN. In their research, the BN is constructed by one child node and multiple parent nodes that describe the relationship between an alarm variable and root-cause variables using the process knowledge. Lokrantz et al. (2018) propose a BN-based graphic probabilistic models using the expert knowledge to identify the causality of failure and quality deviation among multiple manufacturing stages, where network parameters are trained by historical data, and root cause is inferred according to defect types and measurements. Liu et al. (2022) develop a strong relevant mechanism BN combining process mechanism analysis with historical data mining for unmonitored root cause variables in chemical plants fault diagnosis, which showed great practicability and satisfactory

performances in fault propagation recognitions. Amin et al. (2021) present a hybrid data-driven method integrating principal component analysis with the Bayesian networks for fault detection and diagnosis in process plants, which demonstrated a strong efficacy of diagnosis performance while maintaining lesser false diagnosis. In (Amin et al., 2019), the same authors develop a dynamic Bayesian network-based fault detection and root cause diagnosis, which had an ability to convert the continuous process data into meaningful evidence instead of a probabilistic domain. There are also some BN studies emphasizing cross-level fault diagnosis in an HVAC system. For example, Chen et al. (2018) propose a whole building fault diagnosis method based on Discrete BN to isolate faults causing significant abnormalities in multiple subsystems/equipment during system operation, and further design a weather and schedule-based pattern matching Discrete BN to diagnose cross-level faults in building HVAC systems for real-time fault diagnosis and isolations (Chen et al., 2022). Pradhan et al. (2021) develop a dynamic BN-based approach that incorporated the temporal dependencies of fault nodes between time steps using temporal conditional probabilities to improve accuracy for a whole building level fault diagnosis.

However, existing BN-based diagnosis methods highly rely on heuristics processes to learn causal relationships among fault status and symptoms, and their causal mechanism is primarily determined by the expert knowledge. While promising, heuristics processes by the domain knowledge may not be adequate and effective for the fault diagnosis in more complex buildings, especially those with multiple coupled subsystems. Other than being labor intensive, these approaches may not discover underlying coupling effects among the subsystems comprehensively. This leverages a data-driven approach to interrogate the fault-symptom causalities to construct the BN structure. A notable emerging field is causal learning (CL) which uses the observational data to learn causality and it is believed that CL presents new opportunities to address do-

main specific challenges (Cheng et al., 2022). In general, CL research focuses on two categories (Schölkopf, 2022): (1) causal effects estimation; and (2) causal structure learning. Causal effects estimation is to investigate how much changing one variable will influence another given a causal structure assumption between these two variables. This can be done by the counterfactual inference (Pearl, 1988, 2009) which assesses the strength of causality between two events by inferring the likelihood of one event not occurring when another is absent. Causal structure learning, on the other hand, is to induce the structure describing the causal relationships from variables to others, and BN is one of the prevailing causal structure learning tools as it has shown the ability to represent the probabilistically conditional independence in a graph model, providing an efficient and expressive way for knowledge representations and acquirements (Jiang et al., 2019).

Despite the extensive research conducted on BN, there have been limited investigations into the causal structure construction from the causal effect estimation perspective using observation data, especially the BN construction for building fault diagnosis. As reviewed earlier, most building research using BN has heavily relied on the domain experts' knowledge. Additionally, the existing research on BN-based fault diagnosis primarily relied on the assumption of the symptom independence (Chen et al., 2022,a,b), which may not always be valid.

Consequently, this chapter presents an entropy-based causality framework for cross-level faults diagnosis and isolation in building HVAC systems. The hypothesis behind this methodology is that the building is an interconnected system comprising multiple subsystems, and when a fault occurs, co-evolving of multiple symptoms may present some unique patterns. Motivated by this idea, a new concept termed "synchronicity" is introduced to describe the co-evolving patterns, in conjunction with a CL-based framework to induce the BN structures. Specifically, Eigen-Entropy is

employed to characterize the "synchronicity". Next, causal inference is used for causation measurements between the fault status and symptoms to decide what symptoms should be included in the BN structure model. Finally, the performance of the proposed framework is validated by three fault test cases. More details can be found in (Huang et al., 2024).

6.2 Methodology

This section begins with introductions on three concepts, namely BN model, Pearl Causality, and Synchronicity. Next the proposed entropy-based causality learning framework, termed Eigen-Entropy Causal Learning (EECL), is presented for BN structures learning.

6.2.1 BN Model for Building System Fault Diagnosis

Bayesian Network (BN) is a probabilistic graphical model representing a set of variables and their conditional dependencies via a directed acyclic graph, which can be used to reveal causal relationships between faults and symptoms. Figure 6.1 below illustrates an example of a BN model for fault diagnosis in the building systems.

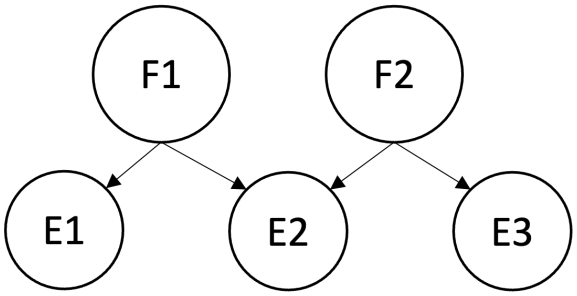

Figure 6.1: A BN model for fault diagnosis

In this BN model, F1 and F2 are fault nodes that represent two distinct faults, while E1, E2 and E3 are evidence nodes that are indicatives (symptoms) of the presence of a fault. Arcs from the fault nodes to the evidence nodes demonstrate the direct causation that a fault has on the occurrence of the evidence. By Bayes theorem (Barber, 2012), posterior probabilities of faults (F1 and F2) given these evidence nodes (E1, E2, and E3) can be calculated to infer which fault is more likely to affect the system. For instance, if the posterior probability of F1 is greater than that of F2, the fault is said to be F1. Clearly, the link between faults and evidence nodes plays a crucial role in facilitating fault diagnosis in building systems based on BN.

6.2.2 Pearl Causality

Pearl Causality (Pearl, 2009), a method for estimating causal effects in fault-symptom relationships, is often employed to support the inference of BN structure. It Also known as counterfactual inference, it assesses the likelihood that one event is the cause of another, which is usually evaluated by the probability of necessity. Given two binary-valued events, X and Y, let x and y stand for the propositions $X = 1$ and $Y = 1$, respectively, $x^{'}$ and $y^{'}$ stand for their complements ($X = 0$ and $Y = 0$). The probability of necessity (PN) is then defined as:

$$PN = P(Y_{X=0} = 0|X = 1, Y = 1) = P(y^{'}_{x^{'}}|x, y) \tag{6.1}$$

Consequently, PN stands for the probability that event y would not have occurred in the absence of event x, denoted as $(y^{'}_{x^{'}})$, given that x and y did actually occur.

Supposing the frequencies of X and Y are as shown in Table 6.1, the PN can be calculated as (Pearl, 1999):

Table 6.1: Frequency data of events X and Y

	$X = 1$	$X = 0$
$Y = 1$	n_{11}	n_{01}
$Y = 0$	n_{10}	n_{00}

$$PN = P(y'_{x'}|x, y) = \frac{P(y) - P(y|x')}{P(x, y)} = \frac{\frac{n_{11}+n_{01}}{n} - \frac{n_{01}}{n_{01}+n_{00}}}{\frac{n_{11}}{n}} \tag{6.2}$$

where $n = n_{11} + n_{10} + n_{01} + n_{00}$. When $PN \geq 0.5$, the causal relationship from x to y is confirmed (Tian and Pearl, 2000).

In this study, event X serves as an indication of the fault status in the building systems. Specifically, when $X = 1$, it signifies the occurrence of a fault, indicating fault conditions. On the other hand, when $X = 0$, it represents normal operations, indicating fault-free conditions. Additionally, event Y is another indicator that describes specific properties related to multiple symptoms, referred to as synchronicity.

To illustrate this idea, take an example of one fault, AHU Cooling Coil Valve Stuck Fully Open (CoolCoiValStuck_0) and two evidence nodes, AHU Cooling Coil Discharge Air Temperature (CC-DA-TEMP) and AHU Cooling Coil Valve Control Signal (CC-VLV). Therefore, $X = 1$ indicates the occurrence of CoolCoiValStuck_0, $X = 0$ indicates the nonoccurrence of CoolCoiValStuck_0 (fault-free condition); $Y = 1$ indicates the synchronicity exist between CC-DA-TEMP and CC-VLV, and $Y = 0$ is asynchronicity. Suppose $n_{11} = 16$ (the frequency of synchronicity under this fault) and $n_{01} = 14$ (the frequency of synchronicity under fault-free condition); $n_{10} = 484$ (the frequency of synchronicity under this fault) and $n_{00} = 986$ (the frequency of asynchronicity under fault-free condition). Given these values, PN can be obtained by Eq (2), which is 0.54, greater than 0.5. Thus, it is concluded that the synchronicity

between CC-DA-TEMP and CC-VLV is attributed to CoolCoiValStuck_0.

Consequently, the following parts will provide a comprehensive explanation of the term synchronicity is explained and elucidate how Pearl Causality is employed to ascertain the BN structure. This BN structure will then be utilized to facilitate fault diagnosis in the building systems.

6.2.3 Eigen-Entropy and Synchronicity

As discussed earlier, the EE method has been used to quantify the homogeneity (or heterogeneity) of a tabular dataset. The interest of this research is on multiple time-series, and here "synchronicity" is defined to describe the multi-time series dataset property which can be measured by EE.

Definition 2. Multiple time series collected from the system may present some co-evolving patterns. If the trend of movements aligns over time, that is, the time series may increase, decrease or remain constant together with respect to time, this pattern is defined as synchronicity.

Without loss of generality, consider two time series, X_1 and X_2, where $X_1 = [x_{11}, x_{12}, \ldots, x_{1n}]$, $X_2 = [x_{21}, x_{22}, \ldots, x_{2n}]$, and x_{ij} refers the time point j of time series i. At time t, cosine similarity between X_{1t} and X_{2t}, is defined as:

$$cos < X_{1t}, X_{2t} > = \frac{\sqrt{\sum_{i=1}^{t} x_{1i} x_{2i}}}{\sqrt{\sum_{i=1}^{t} x_{1i}^2} \sqrt{\sum_{i=1}^{t} x_{2i}^2}} \tag{6.3}$$

The corresponding cosine similarity magnitude matrix on X_{1t} and X_{2t} is

$$\mathbf{C_t^*} = \begin{pmatrix} 1 & c_{12}^* \\ c_{21}^* & 1 \end{pmatrix} \tag{6.4}$$

where c_{12}^t is the magnitude of cosine similarity between X_{1t} and X_{2t}, $c_{12}^t = |cos < X_{1t}, X_{2t} > |$ and $c_{12}^t = c_{21}^t$. Thus eigenvalues λ_1^t and λ_2^t can be derived from $\mathbf{C_t^*}$ to obtain EE (see Eq 3.12). Please note for multiple time-series, the dimension of the cosine similarity magnitude matrix increases accordingly.

Given the magnitude of cosine similarity between X_{1t} and X_{2t}, corresponding Eigen-Entropy can be obtained by Eq 3.12. EE_{t1} be the Eigen-Entropy calculated for X_1 and X_2 at time point t_1 (X_{1t_1} and X_{2t_1}) and EE_{t2} be the Eigen-Entropy calculated for X_1 and X_2 at time point t_2 (X_{1t_2} and X_{2t_2}), where $t_1 < t_2$. If $EE_{t2} < EE_{t1}$, the trends of the time series are assumed to have become more synchronous or aligning from time points t_1 and t_2, and vice versa. Therefore, the value of EE indicates the degree of alignment between the movements of these time-series, or their synchronicity. If the time-series movements are well aligned or highly positively correlated, the value of EE would be zero or close to zero. As the movements become less aligned, the value of EE increases, reaching its maximum when the movements are completely misaligned or highly negatively correlated.

To illustrate this concept, Figure 6.2 presents a simple example which shows the relationships among the movement trends, cosine similarities, and EEs for these two time-series under different scenarios: (1) both X_1 and X_2 increase over time; (2) X_1 remains constant while X_2 increases over time; and (3) X_1 decreases while X_2 increases over time. In Figure 6.2 (A), where X_1 and X_2 exhibit well aligned movements over time (scenario (1)), indicating a strong positive correlation, the cosine similarity

between X_1 and X_2 increases, and the corresponding EE decreases. In Figure 6.2 (B), as X_1 and X_2 show divergent movements (scenario (2)), the cosine similarity between X_1 and X_2 decreases, and the corresponding EE increases. In Figure 6.2 (C), X_1 and X_2 exhibit movements in completely opposite directions (scenario (3)), depicting a strong negative correlation, the cosine similarity decreases significantly, and the EE increases markedly over time.

Note that the patterns for cases where both X_1 and X_2 decrease, X_1 remains constant while X_2 decreases, and X_1 increases while X_2 decreases, are similar to Figure 6.2 (A), (B), and (C) respectively. Therefore, it is concluded that EE can be used as a metric to measure the synchronicity, describing the phenomenon of multiple time series showing trends of aligned movements over time.

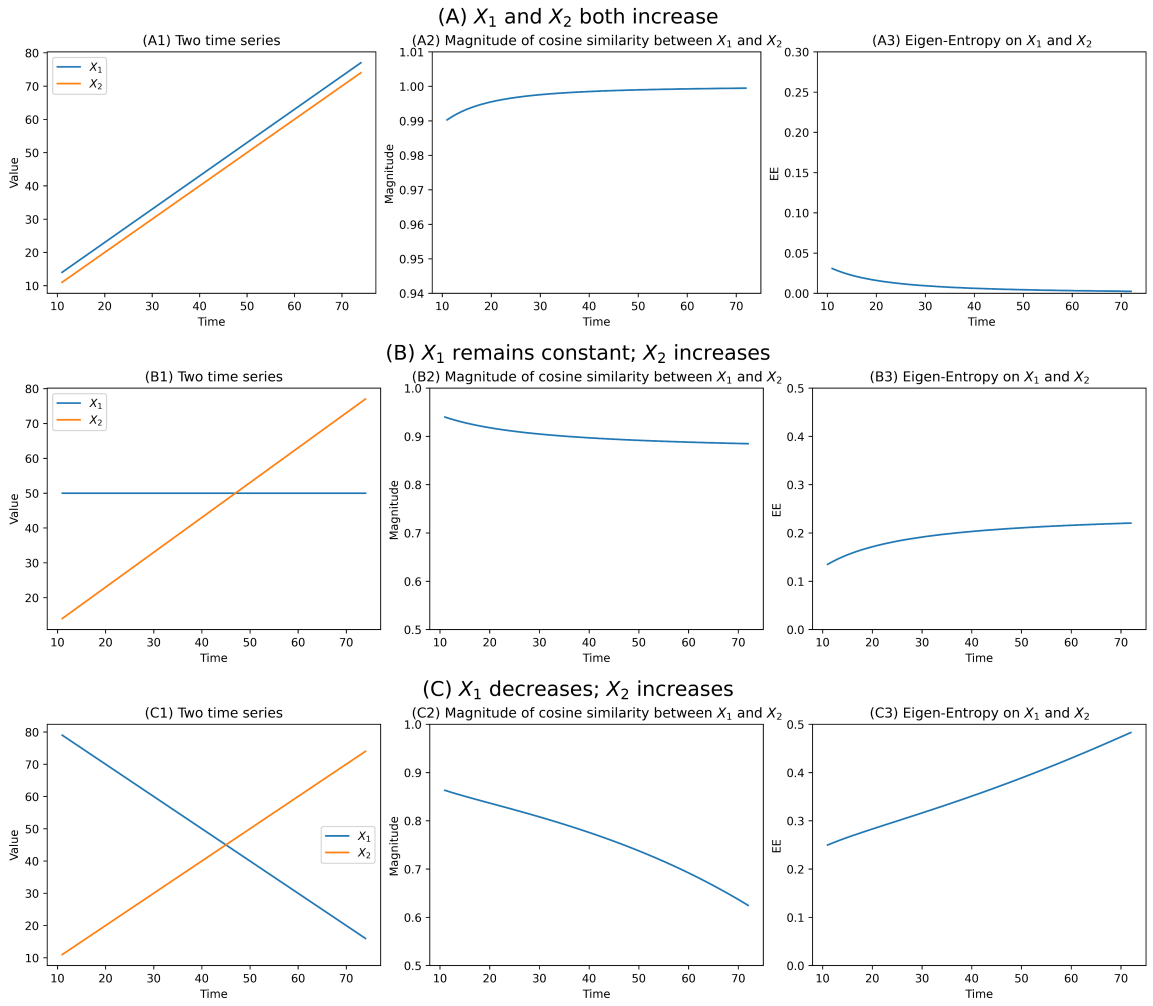

Figure 6.2: Trend of movement, cosine similarity and EE between two time series X_1 and X_2 over time when (A) X_1 and X_2 both increase; (B) X_1 decreases and X_2 increases; (C) X_1 remains constant and X_2 increases.

6.2.4 Eigen-Entropy-based Causality Learning (EECL)

This research focuses on the building HVAC fault diagnosis/root cause isolation. This involves the use of evidence nodes, which are comprised of sensor readings obtained from the building system over a specific time frame. These evidence nodes can be regarded as multiple time series. Upon analyzing the data, it is observed that when a system contains fault(s), the evidence nodes exhibit synchronicity. This has led to an assumption that the synchronicity among evidence nodes is attributed to the system fault(s). As a result, this causal assumption leads to identify evidence nodes for constructing BNs. In other words, it is needed to decide which evidence nodes are associated with which fault.

Algorithm 4 presents the detailed procedure. Specifically, given k fault nodes (a.k.a., fault test cases), each fault node including one fault dataset (i.e., data from the system that contain fault(s)) and corresponding baseline dataset, each dataset consisting of d days' data, each day's data with t time points (t samples), and m evidence nodes (m symptoms), Algorithm 4 presents the EECL method to determine the evidence nodes for each fault node so as to construct a BN for fault diagnosis. The initialization stage is to identify a set of critical evidence nodes for each fault node as candidates to support the BN construction. This involves four main steps. The first step (Algorithm 4, line 1) is to obtain feature importance score of each evidence node by training a machine learning model (e.g., random forest classifier) using all k fault datasets, and then select those evidence nodes whose importance scores are greater than a set value through sensitivity analysis (e.g., in this study, 0.05). This forms a set of critical evidence nodes that can differentiate all k fault datasets, saying $\mathbf{E_{all}}$. Then follow a similar procedure to obtain multiple sets of evidence nodes for any individual test case i (Algorithm 4, line 2), $\mathbf{E_{i,j}}$'s ($i \neq j$), each set containing

Algorithm 4 BN construction by EECL for fault diagnosis

Input: k fault nodes, each fault node with one fault and one baseline datasets, each dataset consisting of d days' data, each data with m evidence nodes and t time points

Output: Bayesian networks for fault diagnosis, BN

Initialization

1: Obtain a set of critical evidence nodes that can differentiate all k fault datasets, $\mathbf{E_{all}}$, according feature importance scores from the machine learning model.

2: **For** fault node i, $i = 1, ..., k$ **do**:

3: Repeat Step 1 to obtain multiple sets, each set containing critical evidence nodes that can differentiate its fault dataset from another fault dataset of fault node j, $\mathbf{E_{i,j}}$, $j = 1, ..., k, i \neq j$.

4: Repeat Step 1 to obtain a set of critical evidence nodes that can differentiate its fault dataset baseline, E_i.

5: Obtain a set of critical evidence nodes by taking the union of $\mathbf{E_{all}}$, $\mathbf{E_{i,j}}$'s and $\mathbf{E_i}$, $\mathbf{E_i^{'}}$.

6: Assign importance score to each evidence node in $\mathbf{E_i^{'}}$ by taking its maximum value of importance scores among $\mathbf{E_{all}}$, $\mathbf{E_{i,j}}$'s and $\mathbf{E_i}$.

7: Rank each evidence node in $\mathbf{E_i^{'}}$ by its importance score in a descending order to obtain a ranked evidence node set, $\hat{\mathbf{E}}_\mathbf{i} = \{e_{(1)}, e_{(2)}, ...\}$.

Evidence nodes selection

8: **For** fault node i, $i = 1, ..., k$ **do**:

9: **For** day p, $p = 1, ..., d$ **do**:

10: Normalize each evidence node from $\hat{\mathbf{E}}_\mathbf{i}$ of fault data with respect to its baseline data in day p.

11: Calculate Eigen-Entropy (EE) by Eq 3.12 on $\hat{\mathbf{E}}_\mathbf{i}$ for each q time point on fault data in day p using the first qth samples, $EE_q, q = 2, ..., t$.

Algorithm 4 BN construction by EECL for fault diagnosis (continued)

12: Derive normalized EE for each q time point, NEE_q, where $NEE_q = \frac{EE_q}{q}$.

13: Assign 1 for each q time point, NEE_q if $NEE_q < \epsilon$; 0 otherwise.

14: Repeat Steps 11-13 for baseline data in day p.

15: Obtain frequency table given results from Steps 11-14, and calculate PN by Eq 6.2.

16: If $PN < 0.5$, update $\hat{\mathbf{E}}_\mathbf{i}$ by removing the last ranked evidence node, and repeat Steps 11-14.

17: Otherwise, stop and output $\hat{\mathbf{E}}_\mathbf{i}$.

18: Return BN by linking k fault nodes with $\{\hat{\mathbf{E}}_\mathbf{1}, ..., \hat{\mathbf{E}}_\mathbf{k}\}$.

critical evidence nodes that can distinguish its fault dataset from that of other test case (Algorithm 4, line 3), and also to identify a set of critical evidence nodes that can differentiate between its fault dataset and baseline, saying $\mathbf{E_i}$ (Algorithm 4, line 4). Hence, a set of critical evidence nodes is obtained for the test case by taking the union of $\mathbf{E_{all}}$, multiple $\mathbf{E_{i,j}}$'s and $\mathbf{E_i}$, saying $\mathbf{E_i^{'}}$, which contains critical evidence nodes from previous steps. Next, assign an importance score to each evidence node in $\mathbf{E_i^{'}}$ by taking the maximum value of its importance scores among $\mathbf{E_{all}}$, $\mathbf{E_{i,j}}$'s and $\mathbf{E_i}$ (Algorithm 4, lines 5-6), and finally, rank all evidence nodes in $\mathbf{E_i^{'}}$ in a descending order according to their importance score to obtain a set of ranked, critical evidence nodes, $\hat{\mathbf{E}}_\mathbf{i}$ (Algorithm 4, line 7).

Figure 6.3 illustrates an example given the scenario of three fault node, F_1, F_2 and F_3, one baseline node B, and four evidence nodes, E_1, E_2, E_3, and E_4. Using a machine learning model (e.g., random forest), the important scores for E_1, E_2, E_3, and E_4 that can distinguish three fault nodes (F_1 vs. F_2 vs. F_3) are all 0.2; Next focus on the F_1 only, and assess these evidence nodes by distinguishing pairwise fault nodes (F_1 vs. F_2; F_1 vs. F_3): the scores for E_1, E_2, E_3, and E_4 are 0.15, 0.25, 0.20, and 0.05 for F_1 vs. F_2, and 0.23, 0.24, 0.36, 0.17 for F_1 vs. F_3 respectively; then

obtain the scores for F_1 vs. B are 0.05, 0.62, 0.00 and 0.33. Therefore, from these scoring results, rank evidence nodes according to their max score in a descending order (scores: $E_2 > E_3 > E_4 > E_1$).

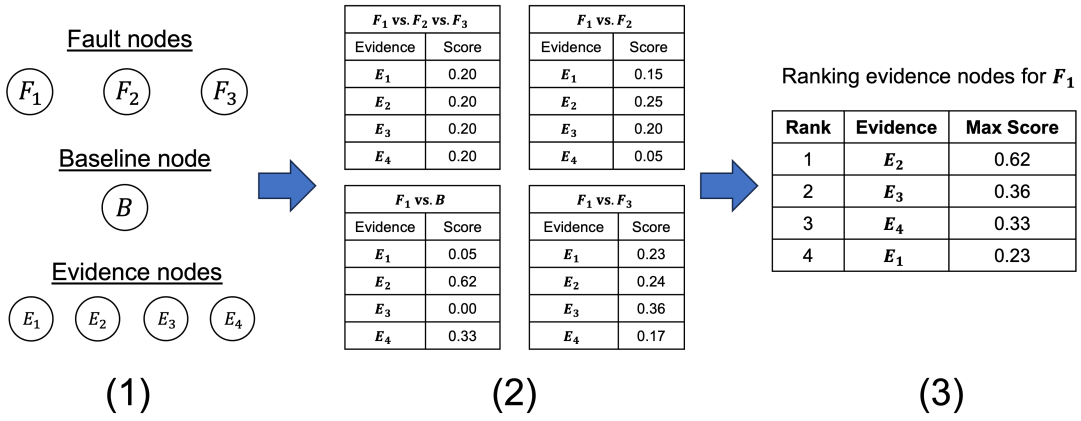

Figure 6.3: An illustrative example of procedures of obtaining the set of ranked critical evidence nodes for one fault node. (1) 3 fault nodes and one baseline with 4 evidence nodes; (2) For one fault node, F_1, obtain (importance) scores for each evidence node under different comparison scenarios: (3) Rank evidence nodes by max score for F_1.

Next stage is to start selecting evidence nodes for each fault node. For each day in a fault node, the first step is normalize each evidence node in $\hat{\mathbf{E}}_\mathbf{i}$ of the fault data with respect to its corresponding baseline (Algorithm 4, line 10). Then for both fault data and baseline, calculate EE on $\hat{\mathbf{E}}_\mathbf{i}$ for each time point q and obtain corresponding normalized EE, NEE_q; assign 1 if NEE_q is smaller than a certain threshold ϵ, indicating the existence of synchronicity among evidence nodes at time point q (that is, evidence nodes exhibit the trend of aligned movements over time q); 0 otherwise (Algorithm 4, lines 11-14).

After going through all d days' data, a frequency table can be obtained for this fault node, where $X = 1$ indicating fault conditions, $X = 0$ indicating fault-free con-

ditions, $Y = 1$ indicating synchronicity, otherwise, $Y = 0$. The frequency information is used for probability of necessity (PN) calculations to assess the causal relationship from fault status to synchronicity among $\hat{\mathbf{E}}_\mathbf{i}$ (Algorithm 4, line 15). If the $PN < 0.5$, it indicates the causal relationship does not hold between that fault condition and the synchronicity of the evidence nodes; thus, remove the last ranked evidence node from $\hat{\mathbf{E}}_\mathbf{i}$ (note that in this approach no evidence node is added) and continue the process; otherwise stop and output the final evidence nodes for this fault node (Algorithm 4, lines 16-17). Finally, go through the procedures for all fault nodes, and construct BN by linking all fault nodes to corresponding evidence nodes selected (Algorithm 4, line 18).

6.3 Experiments

6.3.1 Datasets

A virtual HVAC system testbed developed using Modelica in Dymola environment (Fritzson, 2014) is used to generate experimental data in the experiment, and Figure 6.4 shows the schematics of the HVAC system. The developed HVAC system model is for a one-floor, five-zone medium-sized office building, which has one Air Handler Unit (AHU) connected with five Variable Air Volume (VAV) terminal boxes serving five zones (four exterior zones, and one interior zone, respectively). Heating and cooling are delivered by a single-duct VAV system and the reheat in the VAV terminals is supplied by electric resistance coils. The chilled water is supplied by a central chiller plant which consists of a chiller, a waterside economizer, a cooling tower, and one chilled water pump and one condenser water pump. A boiler, fed by natural gas, supplies the hot water to the AHU heating coil.

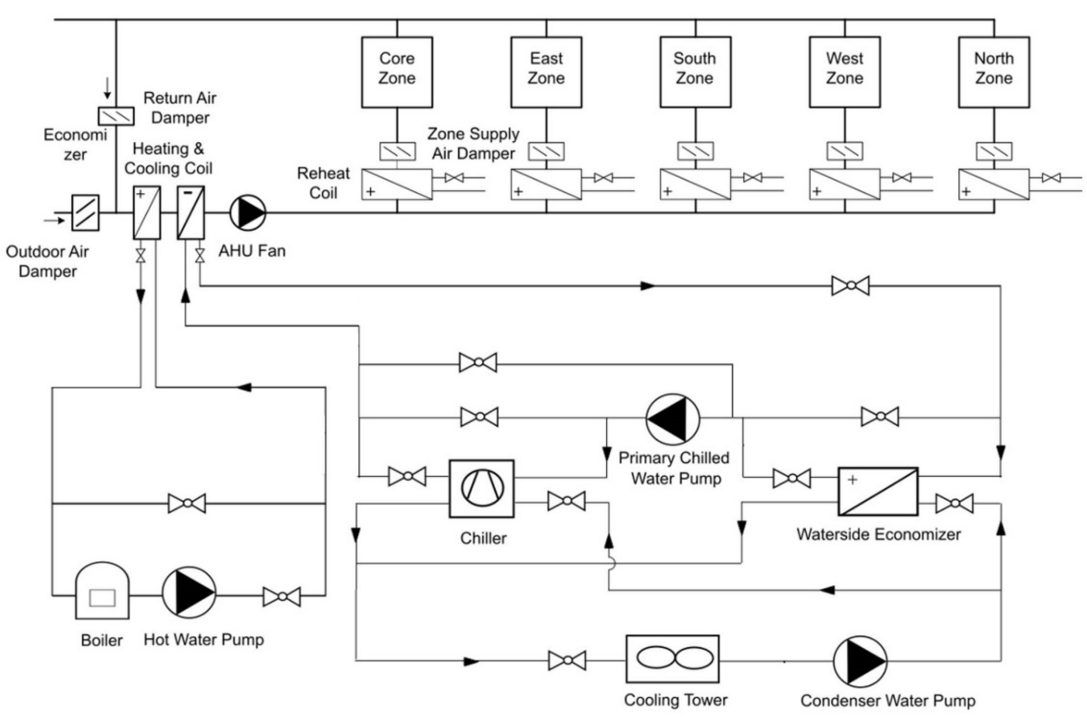

Figure 6.4: Schematic diagram of the simulated HVAC system

Figure 6.5 presents the Modelica model for the studied HVAC system, which is developed based on the open-source Modelica Buildings Library (MBL) (Wetter et al., 2014) and verified against a medium-sized office DOE prototype model (Goel et al., 2014) developed by Pacific Northwest National Laboratory in EnergyPlus (Crawley et al., 2001). The system model consists of three components, namely an HVAC system, a building envelope model, and a model for air flow through building leakage and through open doors based on wind pressure and flow imbalance of the HVAC system. The HVAC system is sized for Chicago, IL, USA in climate zone 5A. The HVAC system control complies with American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) standards and guidelines or literature-reported practices. For example, the air-side control sequences follow ASHRAE Guideline 36 (ASHRAE, 2018) and the water-side control sequences follow ASHRAE project RP-

1711 (Taylor et al., 2019). More details of this HVAC system model can be found in (Fu et al., 2021; Lu et al., 2021; Li et al., 2024).

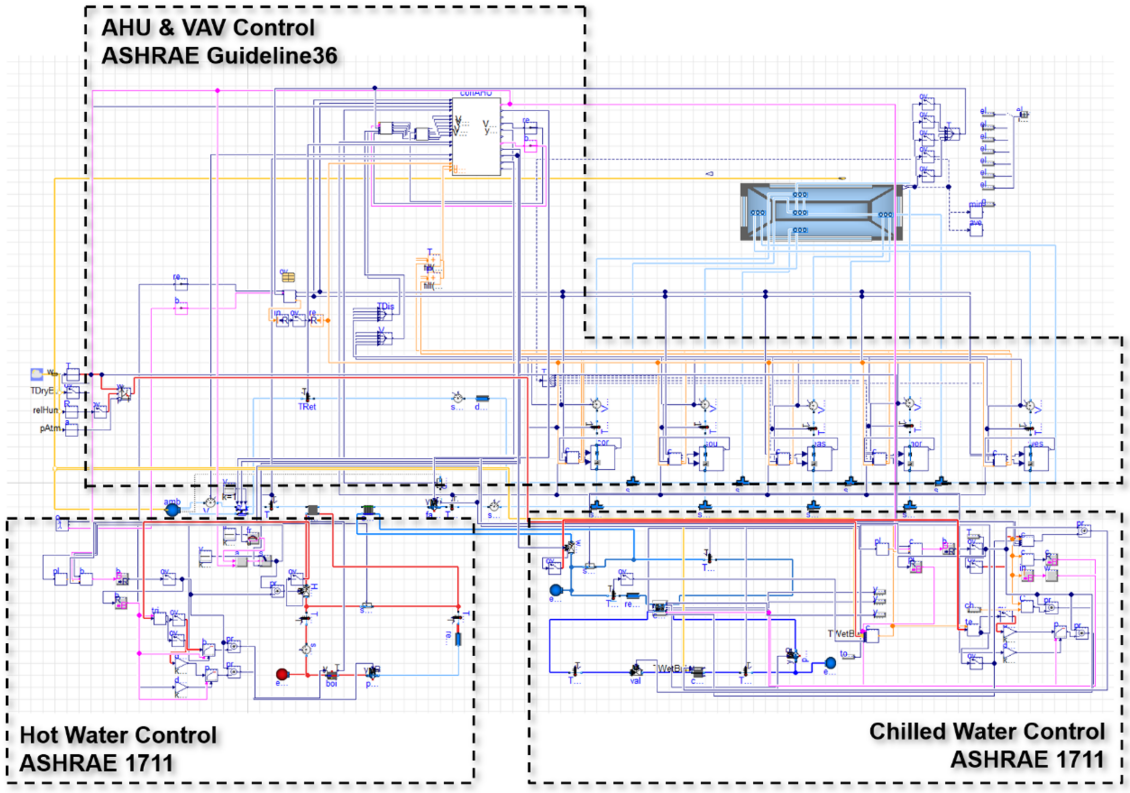

Figure 6.5: Modelica implementation of the studied HVAC system for a commercial building

In this study, three fault datasets and one fault-free dataset collected from this virtual testbed are used. Specifically, these three fault datasets are those collected when the virtual testbed is operated under one of the three different commonly-occurring physical fault conditions, namely, AHU Cooling Coil Valve Stuck Fully Open (CooCoiValStuck_0), AHU Outdoor Air Damper Stuck Fully Closed (OADamStuck_100) and Supply Duct Leakage at a degradation rate of 20% (SupDucLea_20), respectively (see Table 6.2), while the fault-free dataset is collected when the virtual

testbed is operated under normal conditions. Specifically, the fault-free dataset serves as the baseline for each fault node; in other words, the three fault nodes share the same baseline dataset. Since the HVAC system is sized for Chicago, IL, USA in climate zone 5A and the fault injection period starts at the beginning of the day on July 9 and continues for four weeks until August 5, the ranges of temperature and relative humidity are 24-29 °C and 50% to 70% respectively. Each dataset (both fault and fault-free) consists of 28-days' data, each day's data collected under the sampling rate of 5 minutes, thus containing 120 time points (samples) for the entire occupied hours (unoccupied hours excluded); consequently, there are 3360 samples in total for each fault node. Here 15 days' data (1800 samples) from each fault node are used as the training dataset while 5 days' data (600 samples) are used as the test datasets for validation, and the remaining 8 days are excluded because these days correspond to the periods when the building is unoccupied. Detailed information about the training and testing days can be found in Table 6.3.

Table 6.2: Description of three fault nodes

Fault #	Fault node name	Fault node description
1	CoolCoiValStuck_0	AHU Cooling Coil Valve Stuck Fully Open
2	OADamStuck_100	AHU Outdoor Air Damper Stuck Fully Closed
3	SupDucLea_20	Supply Duct Leakage at a degradation rate of 20%

It is worth noticing that both fault and fault-free datasets have 132 evidence nodes. Out of these 132 evidence nodes, 50 are related to the AHU or chiller. Since building faults usually occur in AHUs or chillers, these evidence nodes are important and thus considered as the candidates for the BN construction. The descriptions of these evidence nodes can be found in Table 6.4.

Table 6.3: Days considered for BN-based model training and test

Datasets	Occupied days
Training	Days 1-3; Days 6-10; Days 13-17; Days 20-21
Test	Days 22-24; Day 27-28

Table 6.4: Description of evidence nodes

Evidence #	Evidence node name	Evidence node description
E1	SA-TEMP	AHU Supply Air Temperature
E2	SA-TEMP-STP	AHU Supply Air Temperature Set Point
E3	OA-DB-TEMP	AHU Outdoor Air Dry Bulb Temperature
E4	OA-WB-TEMP	AHU Outdoor Air Wet Bulb Temperature
E5	MA-TEMP	AHU Mixed Air Temperature
E6	RA-TEMP	AHU Return Air Temperature
E7	CC-DA-TEMP	AHU Cooling Coil Discharge Air Temperature
E8	SF-SPD	AHU Supply Air Fan Speed
E9	OA-DMPR	AHU Outdoor Air Damper Control Signal
E10	RA-DMPR	AHU Return Air Damper Control Signal
E11	EA-DMPR	AHU Exhaust Air Damper Control Signal
E12	SA-CFM	AHU Supply Air Mass Flow Rate
E13	OA-CFM	AHU Outdoor Air Mass Flow Rate
E14	RA-CFM	AHU Return Air Mass Flow Rate
E15	EA-CFM	AHU Exhaust Air Mass Flow Rate
E16	CC-VLV	AHU Cooling Coil Valve Control Signal
E17	HC-VLV	AHU Heating Coil Valve Control Signal
E18	SAD-FLOW	AHU Supply Air Duct Static Pressure

Table 6.4 Description of evidence nodes (Continued)

Evidence Index	Evidence node name	Evidence node description
E19	SAD-FLOW-STP	AHU Supply Air Duct Static Pressure Set Point
E20	CC-HTR	AHU Cooling Coil Heat Transfer Rate
E21	HC-HTR	AHU Heating Coil Heat Transfer Rate
E22	SF-PWR-CONS	AHU Supply Air Fan Power Consumption
E23	CHWS-TEMP	Chilled Water Loop: Chilled Water Supply Temperature
E24	CHWR-TEMP	Chilled Water Loop: Chilled Water Return Temperature
E25	CWS-TEMP	Condenser Water Loop: Cooling Water Supply Temperature
E26	CWR-TEMP	Condenser Water Loop: Cooling Water Return Temperature
E27	HWS-TEMP	Hot Water Loop: Hot Water Supply Temperature
E28	HWR-TEMP	Hot Water Loop: Hot Water Return Temperature
E29	CHWS-TEMP-STP	Chilled Water Loop: Supply Chilled Water Temperature Set Point
E30	CHW-DIFF-FLOW	Chilled Water Loop: Measured Differential Pressure
E31	CHW-FLOW-STP	Chilled Water Loop: Differential Pressure Setpoint
E32	HWS-TEMP-STP	Hot Water Loop: Supply Hot Water Temperature Set Point
E33	HW-DIFF-FLOW	Hot Water Loop: Measured Differential Pressure
E34	HW-FLOW-STP	Hot Water Loop: Differential Pressure Setpoint
E35	CHW-FLOW-CC	Chilled Water Loop: Chilled Water Flow Rate into the Cooling Coil
E36	CHL-CHW-FLOW	Chiller: Chilled Water Flow Rate of the Chiller
E37	WSE-CHW-FLOW	Chilled Water Flow Rate of the Water Side Economizer (WSE)
E38	CW-FLOW	Condenser Water Loop: Cooling Water Flow Rate
E39	CHL-CW-FLOW	Chiller: Cooling Water Flow Rate of the Chiller
E40	WSE-CW-FLOW	Cooling Water Flow Rate of the Water Side Economizer (WSE)
E41	HW-FLOW-HC	Hot Water Loop: Hot Water Loop Flow Rate into the Heating Coil
E42	BLR-HW-FLOW	Boiler: Boiler Hot Water Flow Rate
E43	HW-FLOW-BYPS	Hot Water Loop: Bypass Hot Water Flow Rate
E44	CHL-PWR	Chiller Power consumption
E45	DIFF-SAT-STP	AHU Supply Air Temperature and Supply Air Temperature Setpoint Difference
E46	DIFF-OAT-MAT	Difference between AHU Outdoor Air Temperature and Mixed Air Temperature
E47	CHW-COOLING	Chilled Water Cooling Capacity
E48	VAV-FLOW-SUM	Summation of VAV Flowrate
E49	MA-TEMP-1	AHU Mixed Air Temperature Curve Fit (MAT = f (OAT, RAT, SAflow, RAflow))
E50	MA-TEMP-2	AHU Mixed Air Temperature Curve Fit (MAT = f (OAT, RAT, SAflow, OAdmpr))

6.3.2 Benchmark Methods

For comparison purpose, MI-Kruskal-K2 (MIKK2) algorithm (Li and Yu, 2022) is included. MI-Kruskal-K2 algorithm contains three steps. The first step is to obtain the mutual information (MI) between variables; Next is to use the Kruskal algorithm in graph theory to construct the maximum spanning tree to obtain the maximum node in-degree, μ and search the maximum spanning tree through Depth First Search to get the node order ρ. In the final step, K2 algorithm is applied for construct optimal Bayesian Network structure by calling node in-degree μ and node order ρ.

Besides, the structure of BN developed based on expert knowledge and physical analysis is included for comparison. The values of an evidence node from a fault dataset are compared with those from a baseline dataset to observe if this evidence node demonstrates abnormality under a fault condition. Figure 6.6 illustrates an example showing the effects of the fault: Cooling Coil Valve Stuck Fully Open on two evidence nodes, DIFF-SAT-STP (i.e., AHU Supply Air Temperature and Supply Air Temperature Setpoint Difference) and MA-TEMP-1 (i.e., Mixed Air Temperature). There are larger differences between the values of DIFF-SAT-STP under the fault scenario (see Figure 6.6(A) in purple) and those under baseline (see Figure 6.6(B) in green), while there are many overlaps between the values of MA-TEMP-1. Consequently, DIFF-SAT-STP rather than MA-TEMP-1 will be selected for the BN since this evidence node has shown significant abnormality under the fault condition according to the criteria described in (Pradhan, 2023). Following the same procedures, nine evidence nodes are selected for the fault 'CooCoiValStuck_0', nine for the fault 'OADamStuck_100, and eight for the fault 'SupDucLea_20'.

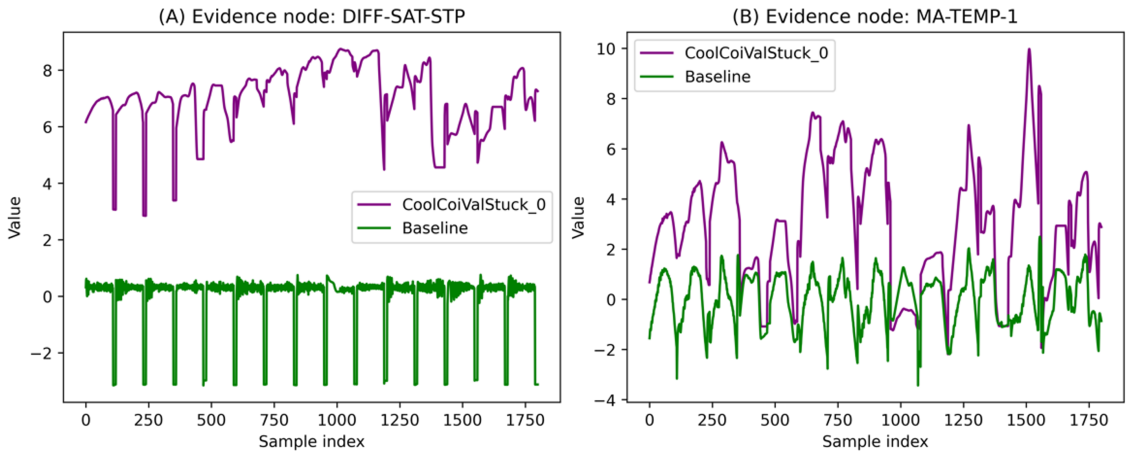

Figure 6.6: Effects of the fault Cooling Coil Valve Stuck Fully Open on two evidence nodes: (A) DIFF-SAT-STP and (B) MA-TEMP-1

6.3.3 Evaluations

In the BN model, every fault sample for each fault node is assessed by utilizing posterior probabilities derived from prior and conditional probabilities. Define s_{ij} as the ith fault sample from fault node j whose true label is $Y(s_{ij})$. Using BN, k posterior probabilities, $P_1(s_{ij})$, $P_2(s_{ij})$,...,$P_k(s_{ij})$, can be obtained, indicating likelihoods of s_{ij} belonging to fault nodes $1, 2, \ldots, k$. Thus, the predicted label for s_{ij}, $\hat{Y}(s_{ij})$, will be based on the maximum of these posterior probabilities, saying $\hat{Y}(s_{ij}) = argmax_r\{P_r(s_{ij})\}$, where $r \in \{1, \ldots, k\}$. Therefore, for any s_{ij}, an indicator, $I(s_{ij})$, is defined such that:

$$I(s_{ij}) = \begin{cases} 1, & \text{if } \hat{Y}(s_{ij}) = Y(s_{ij}) \\ 0, & \text{otherwise} \end{cases} \tag{6.5}$$

where $I(s_{ij}) = 1$ indicating that the ith fault sample from fault node j is correctly

identified by the BN; $I(s_{ij}) = 0$ indicating this sample is incorrectly identified. If there are n fault samples in fault node j, isolation accuracy (IA) for fault node j is defined as:

$$IA_j = \frac{\sum_{i=1}^{n} I(s_{ij})}{n} \tag{6.6}$$

Once the IA is calculated for each fault node, the average isolation accuracy (AIA) over all (say k) fault nodes is defined as:

$$AIA = \frac{\sum_{j=1}^{k} I(IA_j)}{k} \tag{6.7}$$

If there are m evidence nodes in the BN, the sufficient isolation accuracy (SIA) for the BN is defined as:

$$SIA = \frac{AIA}{m} \tag{6.8}$$

It is expected that a robust BN contains as fewer evidence nodes as possible while maintaining satisfactory isolation accuracy, which can be measured by SIA. In other words, a robust BN has a higher SIA.

6.3.4 Experimental Results

As is shown in Algorithm 4 initialization stage, it is needed to deterimine $\mathbf{E_i'}$ for each fault node i, and Table 6.5 provides the ranking of each evidence nodes under each fault conditions. Moreover, a threshold ϵ is needed to determine if there is synchronicity among evidence nodes. As there is no set rule to determine ϵ to identify significant synchronicity, experiments are conducted by varying ϵ from 0.001 to 0.005 with increments of 0.001. This is because when ϵ is greater than 0.006, the PN is less than 0.5 indicating that there is no causal relationship according to (Pearl,

1999). Through observations, it is found that $\epsilon = 0.005$ yields satisfactory results, as the SIA of the constructed BN under this threshold is 11.38%, surpassing the results obtained from other values (refer to Table 6.5). Moreover, observing from Figure 6.7 , normalized EE values based on selected evidence nodes under fault conditions are below $\epsilon = 0.005$, which agrees to the assumption that fault conditions will lead to the synchronicity among evidence nodes. Therefore, results with $\epsilon = 0.005$ are reported and the corresponding BN is shown in Figure 6.8.

Figures 6.9 and 6.10 present BNs constructed by expert knowledge and MIKK2 algorithm respectively. Posterior probabilities for each fault node obtained using BNs by three methods are compared in the analysis. Two important parameters, prior and conditional probabilities predefined as in (Pradhan, 2023) are used for both faults and evidence nodes for posterior probability calculations, and the posterior probabilities are generated by BayesFusion software (BayesFusion, 2018, 2021).

Table 6.5: Ranking of critical evidence nodes for three test cases

Rank	CooCoiValStuck_0		OADamStuck_100		SupDucLea_20	
	Evidence	Score	Evidence	Score	Evidence	Score
1	CC-DA-TEMP	0.1773	RA-CFM	0.1889	RA-CFM	0.1889
2	DIFF-SAT-STP	0.1492	EA-CFM	0.1754	CC-DA-TEMP	0.1773
3	CC-VLV	0.1418	CC-DA-TEMP	0.1741	EA-CFM	0.1754
4	SA-TEMP	0.1024	OA-DMPR	0.1628	DIFF-SAT-STP	0.1492
5	CWR-TEMP	0.0998	DIFF-SAT-STP	0.1448	CHW-COOLING	0.1281
6	RA-CFM	0.0912	OA-CFM	0.1182	OA-CFM	0.1182
7	EA-CFM	0.0880	SA-TEMP	0.1007	OA-DMPR	0.1100
8	RA-TEMP	0.0812	MA-TEMP-2	0.0992	CWR-TEMP	0.0998
9	CHWS-TEMP-STP	0.0758	MA-TEMP	0.0971	SA-TEMP	0.0858
10	CHW-FLOW-CC	0.0601	CC-VLV	0.0786	CC-HTR	0.0759
11	OA-CFM	0.0546	RA-TEMP	0.0766	CHWS-TEMP-STP	0.0758
12			CWR-TEMP	0.0739	CHL-PWR	0.0664
13			SA-CFM	0.0635	SA-CFM	0.0644
14			SF-PWR-CONS	0.0594	CHW-FLOW-CC	0.0601
15			CHW-FLOW-CC	0.0512	SF-PWR-CONS	0.0594
16					MA-TEMP-2	0.0568
17					RA-TEMP	0.0509

Table 6.6: Results under different ϵ's ($\epsilon = 0.005$ highlighted in grey)

ϵ	# evidence node in BN	AIA	SIA
0.001	19	78%	4.11%
0.002	19	78%	4.11%
0.003	19	78%	4.11%
0.004	19	78%	4.11%
0.005	8	91%	11.38%

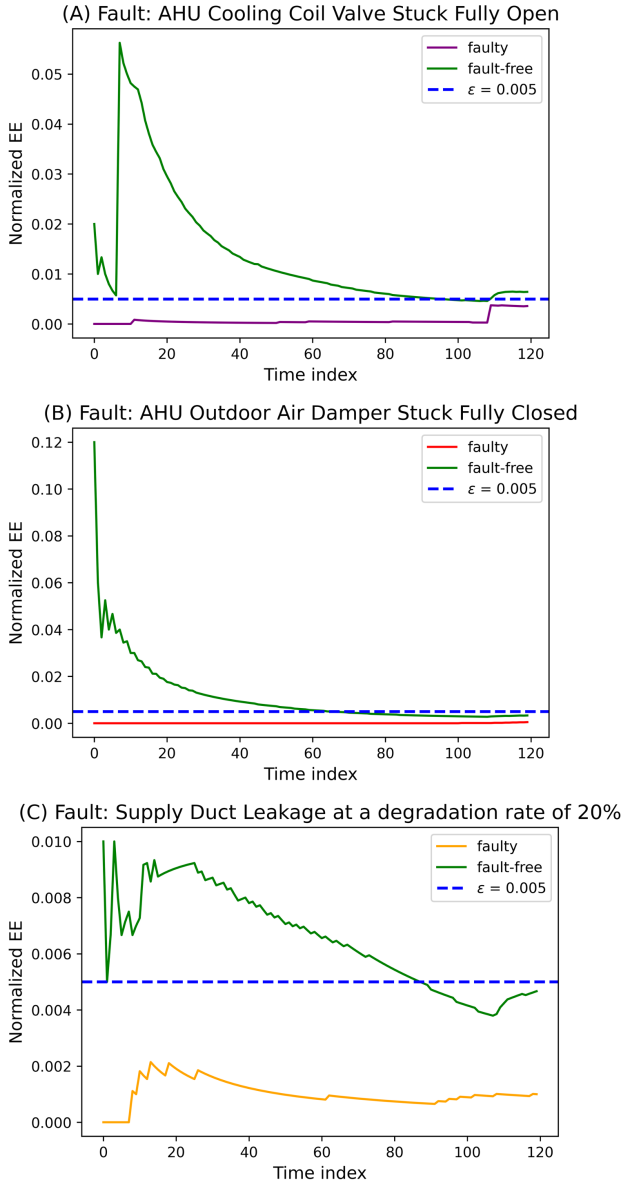

Figure 6.7: Normalized EEs on selected evidence nodes over time for (A) AHU Cooling Coil Valve Stuck Fully Open; (B) AHU Outdoor Air Damper Stuck Fully Closed; and (C) Supply Duct Leakage at a degradation rate of 20%. Each case shows normalized EEs under fault condition below the threshold ($\epsilon = 0.005$), which agrees to the assumptions that fault conditions will lead to evidence synchronicity.

The isolation accuracies for each individual fault using different BNs are shown in Table 6.7. For CooCoiValStuck_0, the isolation accuracy using BN by EECL is 98%, significantly higher than that by MIKK2 (52%), but slightly lower than that by expert knowledge (100%); for OADamStuck_100, the isolation accuracy by EECL is 84%, slightly higher than that by MIKK2 (81%) but lower than that by expert knowledge (100%); for SupDucLea_20, the isolation accuracy by EECL is 90%, lower than that by MIKK2 (100%) and that by expert knowledge (100%). Observing from Figure 6.11 (A) and (B), BN constructed by EE is with 8 evidence nodes and can achieve AIA of 91%, which includes fewer evidence nodes and maintain higher AIA than that by MIKK2 (19 evidence nodes and AIA of 78%); Although BN by expert knowledge can reach AIA of 100%, it includes 50% more evidence nodes than that by EECL. Moreover, as is observed from Figure 6.11 (C), BN constructed by EECL reaches SIA of 11.38%, higher than those by Expert (8.33%) and by MIKK2 (4.11%) respectively. This indicates the efficiency of EECL for BN construction, as EECL requires 33.3% fewer evidence nodes and yields a 36.6% higher SIA compared to expert knowledge, and 57.9% fewer and 1.77 times higher compared to MIKK2, respectively.

As shown in Figure 6.12, the BN using EECL includes eight evidence nodes for CooCoiValStuck_0, three for OADamStuck_100, and four for SupDucLea_20, while the number of evidence nodes by expert knowledge are nine, nine and eight, and the number of evidence node by MIKK2 are eleven, fifteen and seventeen respectively; moreover, three evidence nodes are shared for CooCoiValStuck_0, one for OADamStuck_100, and two for SupDucLea_20 among three BNs, respectively. Therefore, the proposed EECL method for BN construction is able to reach a satisfactory isolation accuracy for the cross-level fault diagnosis in the building systems for this given case study.

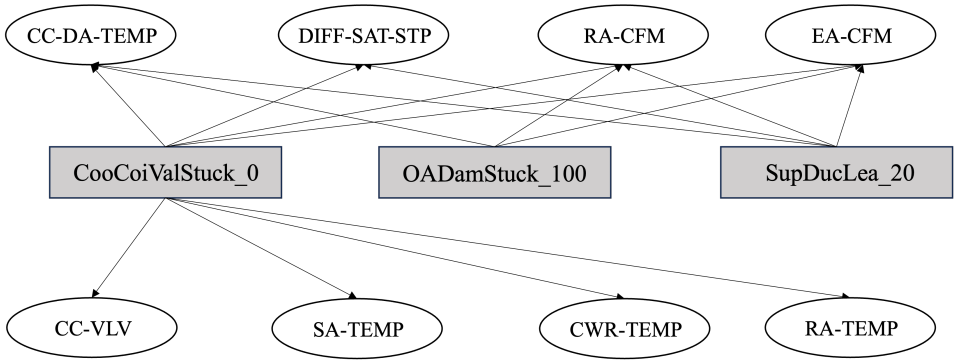

Figure 6.8: BN constructed by EECL under $\epsilon = 0.005$

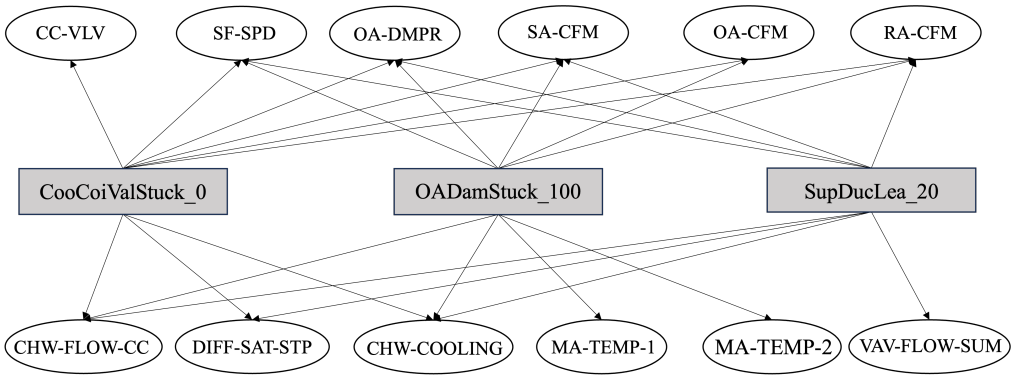

Figure 6.9: BN constructed by expert knowledge

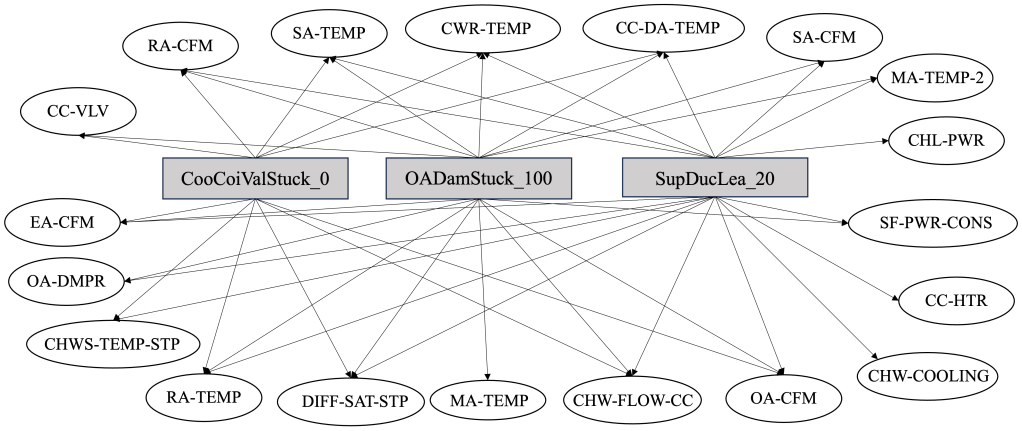

Figure 6.10: BN constructed by MI-Kruskal-K2 algorithm

Table 6.7: Ranking of critical evidence nodes for three test cases

Fault node	Isolation accuracy		
	Expert	MIKK2	EECL
CooCoiValStuck_0	100%	52 %	98%
OADamStuck_100	100%	81%	84%
SupDucLea_20	100%	100%	90%

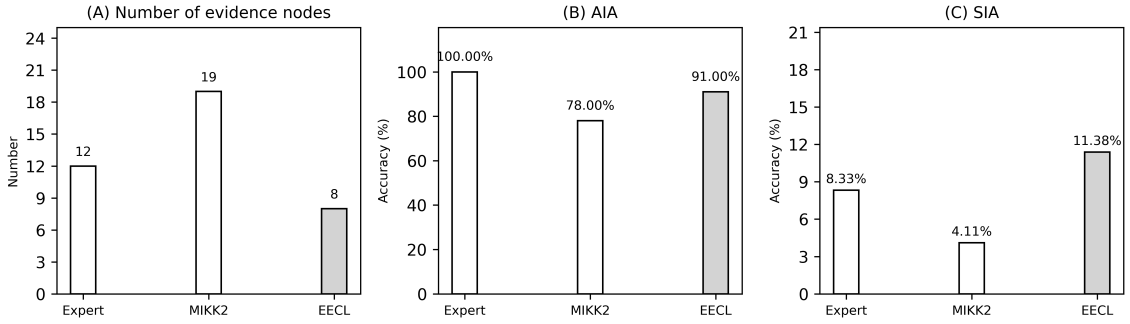

Figure 6.11: Comparisons among BNs by Expert Knowledge, MIKK2 and by EECL in terms of (A) number of evidence nodes; (B) AIA and (C) SIA.

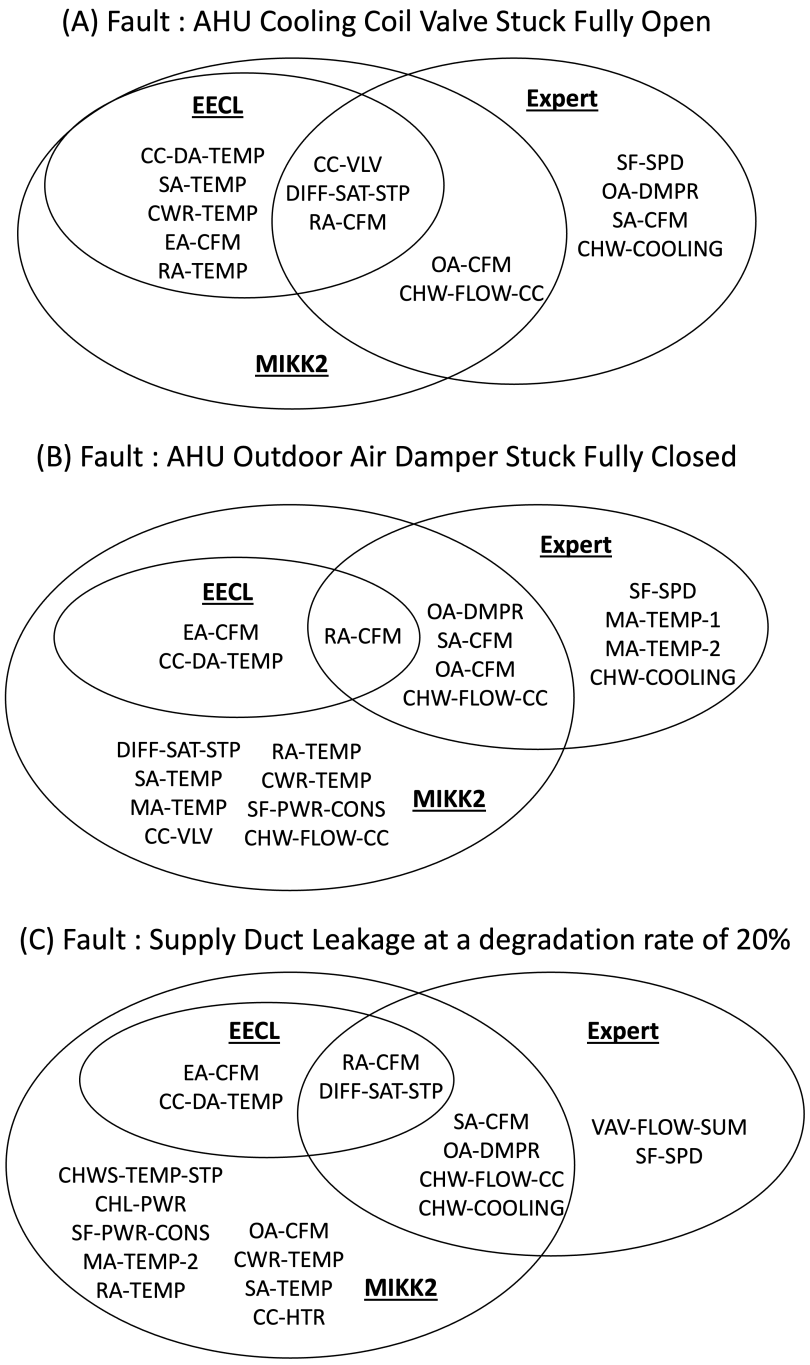

Figure 6.12: Comparisons of BN evidence nodes among by EECL, by expert knowledge and by MIKK2 for three fault nodes (common evidence nodes among three BNs are in the overlap of three circles).

6.4 Conclusion

In this research, the BN structure is constructed using a data-driven causal learning approach. To facilitate causal learning, the concept of "synchronicity" is introduced to describe the interactions among evidence nodes. The direction of causality from the fault status to the synchronicity is characterized by Pearl Causality. This process ultimately results in the construction of the BN structure that can be used to diagnose cross-level faults in a building HVAC system. As discussed in (Chen et al., 2022), automatic process of the BN structure construction is demanding due to the time-consuming and labor-intensive natures of determining BN structures by expert knowledge, and the developed EECL method has the potential for overcoming this deficiency since it is purely data-driven, and does not require any prior knowledge. Additionally, while expert knowledge method can determine the presence of the causal relationships between faults and evidence nodes, it does not provide any measures on the strength of these causations. In contrast, the proposed EECL method is able to quantify the causal relationships in terms of an evaluation metric (i.e., PN), which helps to reduce uncertainties of causality determined explicitly by expert knowledge.

In addition to causation characterizations, the proposed EECL method takes full considerations on interactions among the evidence nodes, which is measured by EE, an information entropy used for multivariate time-series. These interactions may reveal important and interesting patterns specific to a fault, but they may be overlooked by expert knowledge because experts often treat evidence nodes individually. This deficiency also applies in BN constructed by MIKK2 algorithm since evidence nodes are also treated as independent. BN structure by EECL utilizes the parameter model that includes prior and conditional probabilities for the fault and evidence nodes determined by expert knowledge and is able to achieve satisfactory fault diag-

nosis/isolation accuracy (see Experiments).

In fact, several mainstream methods, such as randomized controlled trials, regression analysis, propensity score matching, has been widely used for causal effect estimations. However, the purpose of the above-mentioned is to estimate how strong the causal effect is from one variable to another given a causal assumption is established. That is to say, these approaches have causal assumption as priori to characterize the strength of causality. In contrast, Pearl causality utilizes PN to assess whether the causal relationship is valid from one variable to another by using frequency table. Importantly, decision criterion on causality (PN ≥ 0.5) is uniform, and if PN is greater or equal to 0.5, then such causal assumption holds true between two variables. In this study, the goal is to learn causal structure by determining which evidence nodes should be connected to the fault node, and Pearl causality here aims to answer whether the causal relationships are valid from the fault to evidence nodes.

While promising, the proposed EECL method like most data-driven approach is still influenced by several factors such as data volume, data quality, and data information. Despite the rapid development of sensor technology, collections of a large amount of high-quality, information-rich data are still challenging with the current BAS. For example, the proposed EECL may discard a certain evidence node containing many missing values even though it is important and interpretable from a physical knowledge perspective. Another is that the BN by EECL still relies heavily on the expert knowledge to diagnose cross-level faults due to the same parameter model. There is a need for data-driven parameterizations (i.e., determination on prior and conditional probabilities) to support the diagnosability of BN. Finally, Pearl causality in EECL acquires PN by using frequency table from binary outcomes (e.g., synchronicity or asynchronicity), which may not apply to a more complex scenario when outcomes are multi-class (e.g., weak, medium or strong synchronicity).

Chapter 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

The overall objective of this dissertation is the development of a novel framework based on an information-theoretical metric to tackle three challenges in data-driven building automatic fault detection and diagnosis (AFDD), namely reliable fault detection baseline construction, application of building simulation for real-world fault detection, and Bayesian networks (BN) construction for cross-level fault diagnosis. Specifically, a novel information-theoretical metric, termed Eigen-Entropy (EE), is proposed to quantify the information richness on multivariate time-series. The key idea of EE is to obtain entropy derived from eigenvalues of a correlation magnitude matrix from multivariate data. Theoretical proofs show the relationships between EE and correlations on feature space, and hence enable EE as a valid metric for decision-making support.

In the context of baseline construction, EE is used as the sampling decision-making metric to automatically select qualified data samples from historical normal datasets to construct a robust baseline that effectively supports fault detection in a whole building systems. Through experiments, it has been demonstrated that baselines created using EE achieve AFDD outcomes comparable to those obtained with the existing algorithms. Consequently, this underscores capability of the proposed EE method for baseline constructions. Additionally, the proposed EE has shown the potential for online AFDD baseline constructions.

In the context of building simulation for real-world fault detection, EE is used

to extract features from graph-structured data. Specifically, the proposed method utilizes graph structures learned from simulation and transferred to the real building data, and then extract corresponding entropies (EEs) as features to train and test data-driven AFDD models (e.g., decision tree and random forest). To validate the effectiveness of the proposed methods, five paired fault test cases, each with a matched simulation and real datasets under fault and fault-free conditions, are studied. Experimental results show that features extracted by the proposed method from simulation can significantly improve fault detection performances on real building fault test cases, compared to raw features, signficant edges and GNN features. This demonstrates the efficacy and transferability of the extracted graph features.

In the context of BN construction for cross-level fault diagnosis, EE-based causality framework, EECL, is proposed to support BN structure constructions for fault diagnosis/isolation from the data-driven perspective. The proposed method includes two phases. In the first phase, EE is used for characterizing synchronicity, which describes the trends of movements among the evidence nodes over the time. In the second phase, counterfactual inference is applied to determine what and how evidence nodes should be connected to each fault node so as to build up the BNs to support fault diagnosis, including cross-level faults, in the building system. The BN constructed by the developed EECL method is evaluated against that by expert knowledge based and benchmark methods using three cross-level fault test cases simulated using a virtual testbed. Experimental results show that the EECL based BN can achieve satisfactory isolation accuracy with fewer evidence nodes, indicating the efficacy of EECL approach for fault diagnosis.

7.2 Future Work

Although the Eigen-Entropy framework shows potential, it relies on eigenvalues derived from a correlation coefficient matrix that is real, symmetric, and positive semi-definite. This approach may not be as effective when applied to matrices of different types. Therefore, a potential future research direction is to enhance the Eigen-Entropy framework to accommodate a broader range of high-dimensional data. Further improvements are necessary to optimize the application of Eigen-Entropy in the specific contexts mentioned earlier. These enhancements should focus on:

(I) Baseline construction: there is one parameter used in EE method, ϵ (a small number), as the stopping criteria. To determine the value for this parameter, empirical experiments are conducted. In the future, the applicability of optimization needs to be explored to determine the parameter.

(II) Feature extraction on building simulaton: clusters of significant edges for graph construction is determined by the quantiles, which is confirmed by empirical experiments. In the future, one task is to explore optimization on edges clustering. Another task is to conduct theoretical investigations on the use of EE on graphs.

(III) BN-based fault diagnosis: The current research focuses on a smaller set of fault test cases, each from one fault category. In the future, it is of great interest in investigating the capability of the proposed method for the fault diagnosis on multiple fault test cases, as well as the test cases from the same fault categories. Another interesting topic is to explore data-driven parameterizations for a robust diagnosability of BN by EECL.

REFERENCES

Abrate, C. and F. Bonchi (2021). Counterfactual graphs for explainable classification of brain networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, New York, NY, USA, pp. 2495–2504. Association for Computing Machinery.

Albert, J. (2009). *Bayesian Computation with R* (2nd ed.). New York, NY: Springer.

Alcala-Fdez, J., A. Fernandez, J. Luengo, J. Derrac, and S. Garcia (2011). Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing 17*, 255–287.

Amin, M. T., F. Khan, S. Ahmed, and S. Imtiaz (2021). A data-driven bayesian network learning method for process fault diagnosis. *Process Safety and Environmental Protection 150*, 110–122.

Amin, M. T., F. Khan, and S. Imtiaz (2019). Fault detection and pathway analysis using a dynamic bayesian network. *Chemical Engineering Science 195*, 777–790.

ASHRAE (2018). Ashrae guideline 36-2018, high-performance sequences of operation for hvac systems.

Aziz, F., M. S. Akbar, M. Jawad, A. H. Malik, M. I. Uddin, and G. V. Gkoutos (2021). Graph characterisation using graphlet-based entropies. *Pattern Recognition Letters 147*, 100–107.

Baratloo, A., M. Hosseini, A. Negida, and G. E. Ashal (2015). Part 1: simple definition and calculation of accuracy, sensitivity and specificity. *Emergency 3*(2), 48–49.

Barber, D. (2012). *Bayesian reasoning and machine learning* (1st ed.). Cambridge: Cambridge University Press.

Barua, S., M. M. Islam, X. Yao, and K. Murase (2014). Mwmote–majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Transactions on Knowledge and Data Engineering 26*(2), 405–425.

Batista, G. E., A. Bazzan, and M. Monard (2003). Balancing training data for automated annotation of keywords: a case study. In *Proceedings of the Second Brazilian Workshop on Bioinformatics*, pp. 35–43.

Batista, G. E., R. C. Prati, and M. C. Monard (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter 6*(1), 20–29.

BayesFusion (2018). Genie modeler user manual version 2.2.4.

BayesFusion (2021). Smile wrappers programmer's manual.

Berndt, A. E. (2020). Sampling methods. *Journal of Human Lactation 36*(2), 224–226.

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

Breiman, L. (2001). Random forest. *Machine Learning 45*, 5–32.

Brus, D. J. and J. J. H. van den Akker (2018). How serious a problem is subsoil compaction in the netherlands? a survey based on probability sampling. *SOIL 4*(1), 37–45.

Carcillo, F., Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi (2021). Combining unsupervised and supervised learning in credit card fraud detection. *Information Sciences 557*, 317–331.

Cauchi, N., K. A. Hoque, M. Stoelinga, and A. Abate (2018). Maintenance of smart buildings using fault trees. *ACM Trans. Sen. Netw. 14*(3–4), 1–25.

Cervantes, J., X. Li, W. Yu, and K. Li (2008). Support vector machine classification for large data sets via minimum enclosing ball clustering. *Neurocomputing 71*(4), 611–619.

Chawla, N. V., K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer (2002). Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research 16*(1), 321–357.

Chen, T. and C. Guestrin (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, New York, NY, USA, pp. 785–794. Association for Computing Machinery.

Chen, Y. (2019). *Data-Driven Whole Building Fault Detection and Diagnosis*. Ph. D. dissertation, Department of Civil, Architectural and Environmental Engineering, Drexel University, Philadelphia, PA, USA.

Chen, Y. and J. Wen (2017). A whole building fault detection using weather based pattern matching and feature based pca method. In *2017 IEEE International Conference on Big Data (Big Data)*, pp. 4050–4057.

Chen, Y., J. Wen, T. Chen, and O. Pradhan (2018). Bayesian networks for whole building level fault diagnosis and isolation. In *Proceedings of the 2018 International High Performance Buildings Conference*, pp. 266.

Chen, Y., J. Wen, and J. Lo (2022a). Using Weather and Schedule-Based Pattern Matching and Feature-Based Principal Component Analysis for Whole Building Fault Detection—Part I Development of the Method. *ASME Journal of Engineering for Sustainable Buildings and Cities 3*(1), 011001.

Chen, Y., J. Wen, and J. Lo (2022b). Using Weather and Schedule Based Pattern Matching and Feature Based Principal Component Analysis for Whole Building Fault Detection—Part II Field Evaluation. *ASME Journal of Engineering for Sustainable Buildings and Cities 3*(1), 011002.

Chen, Y., J. Wen, O. Pradhan, L. J. Lo, and T. Wu (2022). Using discrete bayesian networks for diagnosing and isolating cross-level faults in hvac systems. *Applied Energy 327*, 120050.

Chen, Z. and X. C. Liu (2019). Roadway asset inspection sampling using high-dimensional clustering and locality-sensitivity hashing. *Computer-Aided Civil and Infrastructure Engineering 34*(2), 116–129.

Chen, Z., Z. O'Neill, J. Wen, O. Pradhan, T. Yang, X. Lu, G. Lin, S. Miyata, S. Lee, C. Shen, R. Chiosa, M. S. Piscitelli, A. Capozzoli, F. Hengel, A. Kührer, M. Pritoni, W. Liu, J. Clauß, Y. Chen, and T. Herr (2023). A review of data-driven fault detection and diagnostics for building hvac systems. *Applied Energy 339*, 121030.

Cheng, L., R. Guo, R. Moraffah, P. Sheth, K. S. Candan, and H. Liu (2022). Evaluation methods and measures for causal learning algorithms. *IEEE Transactions on Artificial Intelligence 3*, 924–943.

Chiang, C. Y. and M. M. Lin (2013). The eigenvalue shift technique and its eigenstructure analysis of a matrix. *Journal of Computational and Applied Mathematics 253*, 235–248.

CIBSE (2007). *CIBSE Guide H: Building Control Systems*. CRC Press.

Clausius, R. (1877). *Macmillan*. London, UK: Chelsea Publishing Company.

Connor, R. (2016). A tale of four metrics. In L. Amsaleg, M. E. Houle, and E. Schubert (Eds.), *Similarity Search and Applications*, Cham, pp. 210–217. Springer International Publishing.

Cortes, C. and V. Vapnik (1995). Support-vector networks. *Machine Learning 20*, 273–297.

Cox, D. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological) 20*(2), 215–232.

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

Cristianini, N. and J. Taylor (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods* (1st ed.). Cambridge, UK: Cambridge University Press.

Diestel, R. (2017). *Graph Theory* (5th ed.). Springer Publishing Company, Incorporated.

Dutta, S., A. Biswas, and J. Ahrens (2019). Multivariate pointwise information-driven data sampling and visualization. *Entropy 21*(7), 699.

Energy Conservation in Buildings & Communities programme, I. E. A. (1996). Real time simulation of hvac systems for building optimisation, fault detection and diagnostics. international energy agency. Technical Report IEA ECBCS Annex25, International Energy Agency.

Fan, C., F. Xiao, H. Madsen, and D. Wang (2015). Temporal knowledge discovery in big bas data for building energy management. *Energy and Buildings 109*, 75–89.

Fan, W., Y. Si, W. Yang, and M. Sun (2022). Class-specific weighted broad learning system for imbalanced heartbeat classification. *Information Sciences 610*, 525–548.

Fernández, A., S. García, M. J. del Jesus, and F. Herrera (2008). A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. *Fuzzy Sets and Systems 159*(18), 2378–2398.

Frank, S., X. Jin, D. Studer, and A. Farthing (2018). Assessing barriers and research challenges for automated fault detection and diagnosis technology for small commercial buildings in the united states. *Renewable and Sustainable Energy Reviews 98*, 489–499.

Fritzson, P. (2014). *Principles of object oriented modeling and simulation with Modelica 3.3: a cyber-physical approach* (2nd ed.). Piscataway, NJ: Wiley-IEEE Press.

Fu, Y., Z. O'Neill, and V. Adetola (2021). A flexible and generic functional mock-up unit based threat injection framework for grid-interactive efficient buildings: A case study in modelica. *Energy and Buildings 250*, 111263.

Fuller, W. A. (2009). *Sampling Statistics* (1st ed.). Hoboken, NJ: John Wiley & Sons Inc.

Galler, M. (2020). Users guide to the hvacsim+ configuration tool. Technical Report NIST Technical Note 2110, National Institute of Standards and Technology.

Gantmacher, F. R. (1977). *The Theory of Matrices*, Volume 1. New York, NY: Chelsea Publishing Company.

Gao, J. and M. Bergés (2018). A large-scale evaluation of automated metadata inference approaches on sensors from air handling units. *Advanced Engineering Informatics 37*, 14–30.

Geyer, S., I. Papaioannou, and D. Straub (2019). Cross entropy-based importance sampling using gaussian densities revisited. *Structural Safety 76*, 15–27.

Ghasvarian Jahromi, K., D. Gharavian, and H. Mahdiani (2020). A novel method for day-ahead solar power prediction based on hidden markov model and cosine similarity. *Soft Computing 24*, 4991–5004.

Ginestet, S., D. Marchio, and O. Morisot (2008). Evaluation of faults impacts on energy consumption and indoor air quality on an air handling unit. *Energy and Buildings 40*(1), 51–57.

Giovanis, E. (2019). Worthy to lose some money for better air quality: applications of bayesian networks on the causal effect of income and air pollution on life satisfaction in switzerland. *Empirical Economics 57*, 1579–1611.

Goel, S., R. A. Athalye, W. Wang, J. Zhang, M. I. Rosenberg, Y. Xie, P. R. Hart, and V. V. Mendon (2014). Enhancements to ashrae standard 90.1 prototype building models.

Granderson, J., 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 (2022). Lbnl fault detection and diagnostics datasets.

Granderson, J., G. Lin, Y. Chen, A. Casillas, J. Wen, Z. Chen, P. Im, S. Huang, and J. Ling (2023). A labeled dataset for building hvac systems operating in faulted and fault-free states. *Scientific Data 10*, 342.

Granderson, J., G. Lin, A. Harding, P. Im, and Y. Chen (2020). Building fault detection data to aid diagnostic algorithm creation and performance testing. *Scientific Data 7*, 65.

Guo, H., H. Liu, C. Wu, W. Zhi, Y. Xiao, and W. She (2016). Logistic discrimination based on g-mean and f-measure for imbalanced problem. *Journal of Intelligent and Fuzzy Systems 31*(3), 1155–1166.

Hajar, M., M. El Badaoui, A. Raad, and F. Bonnardot (2019). Discrete random sampling: Theory and practice in machine monitoring. *Mechanical Systems and Signal Processing 123*, 386–402.

Han, H., L. Fang, W. Lu, W. Zhai, Y. Li, and J. Zhao (2022). A gcica grant-free random access scheme for m2m communications in crowded massive mimo systems. *IEEE Internet of Things Journal 9*(8), 6032–6046.

Han, J. M., S. Lim, A. Malkawi, X. Han, E. Chen, S. Salimi, T. Dokka, T. Hegli, and K. Edwards (2023). Data-informed building energy management (dibem) towards ultra-low energy buildings. *Energy and Buildings 281*, 112761.

Harer, M., C. Pope, M. Conaway, and J. Charlton (2017). Follow-up of acute kidney injury in neonates during childhood years (fancy): a prospective cohort study. *Pediatric Nephrology 32*(6), 1067–1076.

Hastie, T., J. Friedman, and R. Tibshirani (2009). *The Elements of Statistical Learning*. New York, NY: Springer.

He, H. and Y. Ma (2013). *Imbalanced Learning: Foundations, Algorithms and Applications*. Hoboken, NJ: John Wiely & Sons, Inc.

Hripcsak, G. and A. S. Rothschild (2005). Agreement, the f-measure, and reliability in information retrieval. *Journal of the American Medical Informatics Association 12*(3), 296–298.

Hu, G., T. Zhou, and Q. Liu (2021, 05). Data-driven machine learning for fault detection and diagnosis in nuclear power plants: A review. *Frontiers in Energy Research 9*, 663296.

Huang, J., R. Cheng, X. Liu, L. Chen, and T. Luo (2022). Abnormal static and dynamic functional connectivity of networks related to cognition in patients with subcortical ischemic vascular disease. *Neuroradiology 64*, 1–11.

Huang, J., N. Ghalamsiah, A. Patharkar, O. Pradhan, M. Chu, T. Wu, J. Wen, Z. O'Neill, and K. S. Candan (2024). An entropy-based causality framework for cross-level faults diagnosis and isolation in building hvac systems. *Energy and Buildings 317*, 114378.

Huang, J., T. Li, Y. Xu, T. Wu, H. Yoon, J. Charlton, and K. Bennett (2022). Ee-smote: An oversampling method in conjunction with information entropy for imbalanced learning. In K. Ellis, W. Ferrell, and J. Knapp (Eds.), *IISE Annual Conference and Expo 2022*, IISE Annual Conference and Expo 2022. Institute of Industrial and Systems Engineers, IISE.

Huang, J., J. Wen, H. Yoon, O. Pradhan, T. Wu, Z. O'Neill, and K. Selcuk Candan (2022). Real vs. simulated: Questions on the capability of simulated datasets on building fault detection for energy efficiency from a data-driven perspective. *Energy and Buildings 259*, 111872.

Huang, J., T. Wu, H. Yoon, O. Pradhan, J. Win, and Z. O'Neill (2021). Automatic fault detection baseline construction for building hvac systems using joint entropy and enthalpy. In A. Ghate, K. Krishnaiyer, and K. Paynabar (Eds.), *IISE Annual Conference and Expo 2021*, IISE Annual Conference and Expo 2021, pp. 536–541. Institute of Industrial and Systems Engineers, IISE.

Huang, J., H. Yoon, O. Pradhan, T. Wu, J. Wen, Z. O'Neill, and K. S. Candan (2022). A cosine-based correlation information entropy approach for building automatic fault detection baseline construction. *Science and Technology for the Built Environment 28*(9), 1138–1149.

Huang, J., H. Yoon, T. Wu, K. S. Candan, O. Pradhan, J. Wen, and Z. O'Neill (2023). Eigen-entropy: A metric for multivariate sampling decisions. *Information Sciences 619*, 84–97.

Hunte, J. L., M. Neil, and N. E. Fenton (2022). A causal bayesian network approach for consumer product safety and risk assessment. *Journal of Safety Research 80*, 198–214.

International Energy Agency & United Nations Environment Programme, U. N. (2018). 2018 global status report: Towards a zero-emission, efficient and resilient buildings and construction sector.

Jiang, L., L. Zhang, C. Li, and J. Wu (2019). A correlation-based feature weighting filter for naive bayes. *IEEE Transactions on Knowledge and Data Engineering 31*(2), 201–213.

Jiang, Z., Z. Deng, X. Wang, and B. Dong (2023). Pandemic: Occupancy driven predictive ventilation control to minimize energy consumption and infection risk. *Applied Energy 334*, 120676.

Jolliffe, I. (2002). *Principal component analysis* (2nd ed.). New York, NY: Springer.

Kang, C., H. Zhang, Z. Liu, S. Huang, and Y. Yin (2022). LR-GNN: a graph neural network based on link representation for predicting molecular associations. *Briefings in Bioinformatics 23*(1), bbab513.

Kang, X., X. Wang, J. An, and D. Yan (2022). A novel approach of day-ahead cooling load prediction and optimal control for ice-based thermal energy storage (tes) system in commercial buildings. *Energy and Buildings 275*, 112478.

Kano, M., S. Hasebe, I. Hashimoto, and H. Ohno (2001). A new multivariate statistical process monitoring method using principal component analysis. *Computers & Chemical Engineering 25*(7), 1103–1113.

Katipamula, S. and M. R. Brambley (2005a). Review article: Methods for fault detection, diagnostics, and prognostics for building systems—a review, part i. *HVAC & R Research 11*(1), 3–25.

Katipamula, S. and M. R. Brambley (2005b). Review article: Methods for fault detection, diagnostics, and prognostics for building systems—a review, part ii. *HVAC & R Research 11*(2), 169–187.

Khan, M. A. and N. Javaid (2022). Computationally efficient topology optimization of scale-free iot networks. *Computer Communications 185*, 1–12.

Kim, W. and S. Katipamula (2018). A review of fault detection and diagnostics methods for building systems. *Science and Technology for the Built Environment 24*, 3–21.

Lee, S., S. Wang, P. A. Bain, C. Baker, T. Kundinger, C. Sommers, and J. Li (2018). Reducing copd readmissions: A causal bayesian network model. *IEEE Robotics and Automation Letters 3*(4), 4046–4053.

Li, G., Y. Hu, H. Chen, H. Li, M. Hu, Y. Guo, S. Shi, and W. Hu (2016). A sensor fault detection and diagnosis strategy for screw chiller system using support vector data description-based d-statistic and dv-contribution plots. *Energy and Buildings 133*, 230–245.

Li, G., L. Ren, Y. Fu, Z. Yang, V. Adetola, J. Wen, Q. Zhu, T. Wu, K. Candan, and Z. O'Neill (2023). A critical review of cyber-physical security for building automation systems. *Annual Reviews in Control 55*, 237–254.

Li, G., Z. Yang, Y. Fu, Z. O'Neill, L. Ren, O. Pradhan, and J. Wen (2024). A hardware-in-the-loop (hil) testbed for cyber-physical energy systems in smart commercial buildings. *Science and Technology for the Built Environment 30*, 415–432.

Li, L., H. He, and J. Li (2020). Entropy-based sampling approaches for multiclass imbalanced problems. *IEEE Transactions on Knowledge and Data Engineering 32*(11), 2159–2170.

Li, S. and J. Wen (2007). Description of fault test in summer of 2007. Technical Report ASHRAE RP-1312, ASHRAE Research Project 1312.

Li, X., C. P. Bowers, and T. Schnier (2010). Classification of energy consumption in buildings with outlier detection. *IEEE Transactions on Industrial Electronics 57*(11), 3639–3644.

Li, X. and H. Yu (2022). Bayesian network structure learning algorithm based on node order constraint. In *2022 3rd International Conference on Electronic Communication and Artificial Intelligence (IWECAI)*, pp. 214–217.

Li, X., H. Zhang, R. Wang, and F. Nie (2022). Multiview clustering: A scalable and parameter-free bipartite graph fusion method. *IEEE Transactions on Pattern Analysis and Machine Intelligence 44*(1), 330–344.

Li, Y. and Z. O'Neill (2018). A critical review of fault modeling of hvac systems in buildings. *Building Simulation 11*, 1–23.

Li, Y., N. Zhong, D. Taniar, and H. Zhang (2022). Mcgnet+: An improved motor imagery classication based on cosine similarity. *Brain Informatics 9*, 3.

Liang, W., M. Lv, and X. Yang (2021). Development of a physics-based model for analyzing formaldehyde emissions from building material under coupling effects of temperature and humidity. *Building and Environment 203*, 108078.

Lin, G. and D. Claridge (2015). A temperature-based approach to detect abnormal building energy consumption. *Energy and Buildings 93*, 110–118.

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

Liu, M., X. Qi, and H. Pan (2022). Construction of network topology and geographical vulnerability for telecommunication network. *Computer Networks 205*, 108764.

Liu, N., M. Hu, J. Wang, Y. Ren, and W. Tian (2022). Fault detection and diagnosis using bayesian network model combining mechanism correlation analysis and process data: Application to unmonitored root cause variables type faults. *Process Safety and Environmental Protection 164*, 15–29.

Liu, X.-Y., J. Wu, and Z.-H. Zhou (2009). Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 39*(2), 539–550.

Lokrantz, A., E. Gustavsson, and M. Jirstrand (2018). Root cause analysis of failures and quality deviations in manufacturing using machine learning. *Procedia CIRP 72*, 1057–1062. 51st CIRP Conference on Manufacturing Systems.

Lu, X., Y. Fu, Z. O'Neill, and J. Wen (2021). A holistic fault impact analysis of the high-performance sequences of operation for hvac systems: Modelica-based case study in a medium-office building. *Energy and Buildings 252*, 111448.

Malkawi, A., S. Ervin, X. Han, E. Chen, S. Lim, S. Ampanavos, and P. Howard (2023). Design and applications of an iot architecture for data-driven smart building operations and experimentation. *Energy and Buildings 295*, 113291.

Mao, L., E. Chu, J. Gu, T. Hu, B. J. Weiner, and Y. Su (2023, Jun). A 4d theoretical framework for measuring topic-specific influence on twitter: Development and usability study on dietary sodium tweets. *J Med Internet Res 25*, e45897.

Massey Jr, F. (1951). The kolmogorov-smirnov test for goodness of fit. *Journal of the American Statistical Association 46*, 68–78.

McClish, D. K. (1989). Analyzing a portion of the roc curve. *Medical Decision Making 9*(3), 190–195.

Miles, C. H., M. Petersen, and M. J. van der Laan (2019). Causal inference when counterfactuals depend on the proportion of all subjects exposed. *Biometrics 75*(3), 768–777.

Miller, C., Z. Nagy, and A. Schlueter (2015). Automated daily pattern filtering of measured building performance data. *Automation in Construction 49*, 1–17.

NIDDK (n.d.). Explaining your kidney test results: A tool for clinical use.

Omri, N., Z. Al Masry, N. Mairot, S. Giampiccolo, and N. Zerhouni (2021). Towards an adapted phm approach: Data quality requirements methodology for fault detection applications. *Computers in Industry 127*, 103414.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA: Morgan Kaufmann.

Pearl, J. (1999). Probabilities of causation: Three counterfactual interpretations and their identification. *Synthese 121*, 93–149.

Pearl, J. (2009). *Causality: Models, Reasoning and Inference* (2nd ed.). Cambridge, UK: Cambridge University Press.

Piette, M. A., S. K. Kinney, and P. Haves (2001). Analysis of an information monitoring and diagnostic system to improve building operations. *Energy and Buildings 33*(8), 783–791.

Porcher, R., J. Jacot, J. S. Wunder, and D. J. Biau (2019). Identifying treatment responders using counterfactual modeling and potential outcomes. *Statistical Methods in Medical Research 28*(10-11), 3346–3362.

Powers, D. M. W. (2011). Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation. *Journal of Machine Learning Technologies 2*(1), 37–63.

Pradhan, O. (2023). *A dynamic Bayesian network framework for data-driven fault diagnosis and prognosis of smart building systems*. Ph. D. dissertation, Department of Civil, Architectural and Environmental Engineering, Drexel University, Philadelphia, PA, USA.

Pradhan, O., J. Wen, Y. Chen, X. Lu, M. Chu, Y. Fu, Z. O'Neill, T. Wu, and K. S. Candan (2021). Dynamic bayesian network-based fault diagnosis for ashrae guideline 36: high performance sequence of operation for hvac systems. In *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, BuildSys '21, New York, NY, USA, pp. 365–368. Association for Computing Machinery.

Pérez-Lombard, L., J. Ortiz, and C. Pout (2008). A review on buildings energy consumption information. *Energy and Buildings 40*(3), 394–398.

R.Brambley, M. and S. Katipamula (2009). Commercial building retuning. *ASHRAE Journal 51*, 12–23.

Rifkin, R. and A. Klautau (2004). In defense of one-vs-all classification. *Journal of Machine Learning Research 5*, 101–141.

Rossini, R., S. Poccia, K. S. Candan, and M. L. Sapino (2020). Ca-smooth: Content adaptive smoothing of time series leveraging locally salient temporal features. In *Proceedings of the 11th International Conference on Management of Digital EcoSystems*, MEDES '19, New York, NY, USA, pp. 36–43. Association for Computing Machinery.

Roth, K., D. Westphalen, P. Llana, and M. Feng (2004). The energy impact of commercial building controls and performance diagnostics: market characterization, energy impact of building faults and energy savings potential. In *the International Refrigeration and Air Conditioning Conference*, pp. 665.

Salehi, F., M. R. Keyvanpour, and A. Sharifi (2021). Smkfc-er: Semi-supervised multiple kernel fuzzy clustering based on entropy and relative entropy. *Information Sciences 547*, 667–688.

Samoilenko, M., N. Arrouf, L. Blais, and G. Lefebvre (2020). Comparing two counterfactual-outcome approaches in causal mediation analysis of a multicategorical exposure: An application for the estimation of the effect of maternal intake of inhaled corticosteroids doses on birthweight. *Statistical Methods in Medical Research 29*(10), 2767–2782.

Schölkopf, B. (2022). *Causality for Machine Learning* (1st ed.)., pp. 765–804. New York, NY, USA: Association for Computing Machinery.

Settles, B. (2009). Active learning literature survey. Technical Report 1648, University of Wisconsin–Madison.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell Labs Technical Journal 27*, 379–423.

Shi, Z. and W. O'Brien (2019). Development and implementation of automated fault detection and diagnostics for building systems: A review. *Automation in Construction 104*, 215–229.

Sjölander, A. (2021). Estimation of marginal causal effects in the presence of confounding by cluster. *Biostatistics 22*(3), 598–612.

Steinskog, D., D. Tjøtheim, and N. Kvamstø (2007). A cautionary note on the use of the kolmogorov-smirnov test for normality. *Journal of the American Statistical Association 135*(3), 1151–1157.

Strang, G. (2016). *Introduction to Linear Algebra* (5th ed.). Wellesley, MA: Wellesley-Cambridge Press.

Taylor, S., B. Gill, and R. Kiriu (2019). Advanced sequences of operation for hvac systems – phase ii central plants and hydronic systems. Technical Report ASHRAE RP-1711, American Society of Heating, Refrigerating and Air Conditioning Engineers.

Tian, J. and J. Pearl (2000). Probabilities of causation: Bounds and identification. *Annals of Mathematics and Artificial Intelligence 28*, 287–313.

United Nations Environment Programme, U. N. (2022). 2022 global status report for buildings and construction: towards a zero-emission, efficient and resilient buildings and construction sector.

Wan, Z., H. He, and B. Tang (2018). A generative model for sparse hyperparameter determination. *IEEE Transactions on Big Data 4*(1), 2–10.

Wang, H., D. Feng, and K. Liu (2021). Fault detection and diagnosis for multiple faults of vav terminals using self-adaptive model and layered random forest. *Building and Environment 193*, 107667.

Wang, H. and X. Yao (2016). Objective reduction based on nonlinear correlation information entropy. *Soft Computing 20*(6), 2393–2407.

Wang, J., Z. Yang, J. Su, Y. Zhao, S. Gao, X. Pang, and D. Zhou (2018). Root-cause analysis of occurring alarms in thermal power plants based on bayesian networks. *International Journal of Electrical Power & Energy Systems 103*, 67–74.

Wang, X., Y. Yan, and X. Ma (2020). Feature selection method based on differential correlation information entropy. *Neural Processing Letters 52*(2), 1339–1358.

Wen, J. and S. Li (2011). Tools for evaluating fault detection and diagnostic methods for air-handling units. Technical Report ASHRAE RP-1312, American Society of Heating, Refrigerating and Air Conditioning Engineers.

Wetter, M., W. Zuo, T. Nouidui, and X. Pang (2014). Modelica buildings library. *Journal of Building Performance Simulation 7*(4), 253–270.

Witte, R. S. and J. S. Witte (2017). *Statistics* (11th ed.). Hoboken, NJ: John Wiley & Sons Inc.

Wu, L., P. Cui, J. Pei, L. Zhao, and X. Guo (2023). Graph neural networks: Foundation, frontiers and applications. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, New York, NY, USA, pp. 5831–5832. Association for Computing Machinery.

Xia, H. and Z. Liu (2020). Target classification of SAR images using nonlinear correlation information entropy. *Journal of Applied Remote Sensing 14*(3), 036520.

Xiong, Y.-J., Q. Wang, Y. Du, and Y. Lu (2024). Adaptive graph-based feature normalization for facial expression recognition. *Engineering Applications of Artificial Intelligence 129*, 107623.

Xu, L., L. Bai, X. Jiang, M. Tan, D. Zhang, and B. Luo (2021). Deep rényi entropy graph kernel. *Pattern Recognition 111*, 107668.

Xu, W., L. Jiang, and C. Li (2021). Improving data and model quality in crowdsourcing using cross-entropy-based noise correction. *Information Sciences 546*, 803–814.

Yan, R., Z. Ma, Y. Zhao, and G. Kokogiannakis (2016). A decision tree based data-driven diagnostic strategy for air handling units. *Energy and Buildings 133*, 37–45.

Yan, Y., P. B. Luh, and K. R. Pattipati (2017). Fault diagnosis of hvac air-handling systems considering fault propagation impacts among components. *IEEE Transactions on Automation Science and Engineering 14*(2), 705–717.

Yang, C., K. Zhou, and J. Liu (2022). Supergraph: Spatial-temporal graph-based feature extraction for rotating machinery diagnosis. *IEEE Transactions on Industrial Electronics 69*(4), 4167–4176.

Yang, J., S.-P. Wang, and T. Wu (2023). Maximum mutual information for feature extraction from graph-structured data: Application to alzheimer's disease classification. *Applied Intelligence 53*, 1870–1886.

Yang, S. and G. Berdine (2017). The receiver operating characteristic (roc) curve. *The Southwest Respiratory and Critical Care Chronicles 5*(19), 34–36.

Ye, Y., C. A. Faulkner, R. Xu, S. Huang, Y. Liu, D. L. Vrabie, J. Zhang, and W. Zuo (2023). System modeling for grid-interactive efficient building applications. *Journal of Building Engineering 69*, 106148.

Yoo, C., E. Gonzalez, Z. Gong, and D. Roy (2022). A better mechanistic understanding of big data through an order search using causal bayesian networks. *Big Data and Cognitive Computing 6*(2), 56.

Yu, Z., C. Zhang, and C. Deng (2023). An improved gnn using dynamic graph embedding mechanism: A novel end-to-end framework for rolling bearing fault diagnosis under variable working conditions. *Mechanical Systems and Signal Processing 200*, 110534.

Zare, M. and N. M. Nouri (2022). A new analysis of flow noise outside the time-frequency representation using graph-based feature extraction. *Ocean Engineering 266*, 112700.

Zhang, L., Z. Chen, X. Zhang, A. Pertzborn, and X. Jin (2023). Challenges and opportunities of machine learning control in building operations. *Building Simulation 16*, 831–852.

Zhang, L., M. Leach, J. Chen, and Y. Hu (2023). Sensor cost-effectiveness analysis for data-driven fault detection and diagnostics in commercial buildings. *Energy 263*(Part B), 125577.

Zhang, L., J. Wen, Y. Li, J. Chen, Y. Ye, Y. Fu, and W. Livingood (2021). A review of machine learning in building load prediction. *Applied Energy 285*, 116452.

APPENDIX A

REAL VS. SIMULATED: PRELIMINARY RESEARCH ON CROSS-DATASET BUILDING FAULT DETECTION

The preliminary study (Huang et al., 2022) is designed to examine how training data obtained from simulation might affect a data-driven AFDD strategy's performance, in terms of accuracy, false alarm rate, etc., when the developed strategy is used to analyze real building data. Two datasets generated from an American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) project (Wen and Li, 2011) are used. The two datasets include those from a real building and those from the real building's digital twin, i.e., simulation models representing the real building. The performance of the fault detection strategy on cross-dataset is then examined.

The two datasets generated from the ASHRAE 1312 research project (Li and Wen, 2007; Wen and Li, 2011) are collected from a laboratory building that was set up like a small office building. The office building layout is shown in Figure A.1. The building consisted of two variable air volume (VAV) air handling unit (AHU) HVAC systems, each of which served 4 different rooms. The design of the test facility was intended to have each AHU serving room with nearly identical loads. As can be observed, each HVAC system served rooms facing east, west, south, and one interior room.

While the two systems (A and B) may not generate the same data, the performance is found to be very similar under all operating conditions. During the study, System B (AHU-B and all B rooms) is continuously operated in a fault-free state, while System A (AHU-A and all A rooms) is artificially injected with various commonly occurring faults.

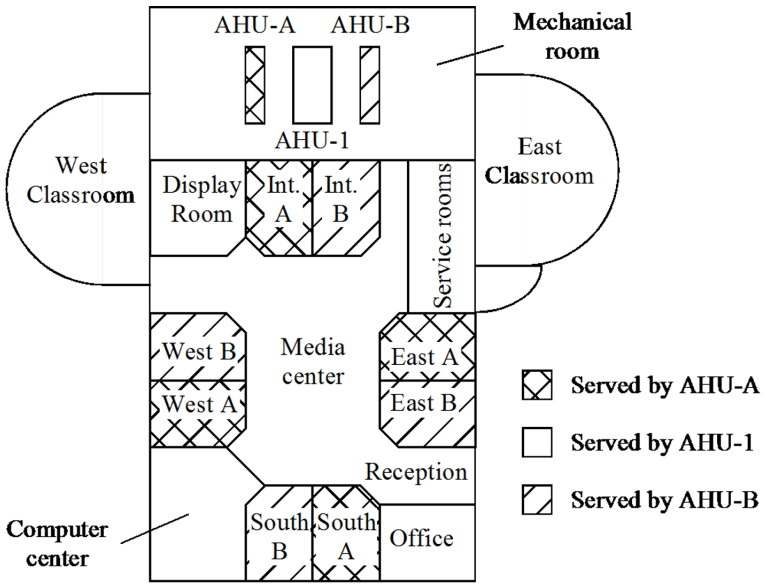

Figure A.1: Energy Resource Station experimental setup

In the same project, dynamic behaviors of the HVAC systems and the four building zones that are served by the AHU, and four VAV boxes, are modeled using HVACSIM+ software (Galler, 2020). The model (called the 1312 model hereafter) was systematically validated using the real building data collected from carefully designed tests to ensure that the 1312 model simulated the dynamic behavior of the test facility for both fault free and faulty operation under three seasons (winter, summer, and spring). It was concluded in the ASHRAE 1312 project that the fault models were able to replicate all major fault symptoms although detailed dynamics between simulated data and real building measured data did not always overlap. This is because the simulation models were physics-based leading to the data generated may exhibit some variations from the real measured data. For example, these simulations most time don't consider the behaviors associated with the latency associated with sensor and control systems.

In this study, two datasets, i.e., real building dataset and their corresponding

simulated dataset, are selected from the ASHRAE 1312 project's 2007 summer tests. Each dataset includes fault test data generated from the A system and fault free data from the B system. During a fault test, a fault was artificially implemented into the system for 12 hours from 6:00 am to 6:00 pm. There are 16 types of fault tests used in this study as described in Table A.1. The data sampling rate for both real and simulated datasets is one minute. ASHRAE 1312 project provides 24 measurements (also referred to as features in later sections) as summarized in Table A.2. For this study, Outdoor Air Temperature (OA-TEMP) and (OA-HUMD) are not included since they represent weather conditions, not building/system conditions. Moreover, the simulation testbed does not simulate humidity variations. Therefore, two humidity-related features, i.e., Supply Air Humidity (SA-HUMD) and Return Air Humidity (RA-HUMD) are not included. As a result, there are 20 features considered in this study.

Each of the two datasets includes the same test days. Since the measurement sampling rate in the test facility is one minute, for each test day, each feature has 1440 samples. Considering that a fault is implemented from 6:00 am to 6:00 pm on a test day, each test day contains 720 data points representing faulty operation and 720 data points representing fault-free operation. More detailed descriptions about fault testing facilities and faulty operational conditions can be referred to ASHRAE 1312-RP project reports (Wen and Li, 2011).

Table A.1: Implemented 16 AHU faults during a summer period

Category	Fault Name
Equipment	AHU Duct Leaking Fault - After Supply Fan
	AHU Duct Leaking Fault - Before Supply Fan
	Return Fan Complete Failure
Controlled Device	Heating Coil Valve Leaking - Stage 1 (0.4GPM)
	Heating Coil Valve Leaking - Stage 2 (1.0GPM)
	Heating Coil Valve Leaking - Stage 3 (2.0GPM)
	Cooling Coil Valve Stuck Fully Closed
	Cooling Coil Valve Stuck Fully Open
	Cooling Coil Valve Stuck 15% Open
	Cooling Coil Valve Stuck 65% Open
	OA Damper Stuck - Fully Closed
	OA Damper Leaking - 45% Open
	OA Damper Leaking - 55% Open
Controller	Cooling Coil Valve Control Unstable
	Cooling Coil Valve Reverse Action
	Return Fan at 30% SPD

Table A.2: Description of the 20 Features used in this study

Category	Features	Abbreviation
Temperature	Supply Air Temperature	SA-TEMP
	Mixed Air Temperature	MA-TEMP
	Return Air Temperature	RA-TEMP
	Heating Coil Discharge Air Temperature	HWC-DAT
	Cooling Coil Discharge Air Temperature	CHWC-DAT
Position	Exhaust Air Damper Position	EA-DMPR
	Return Air Damper Position	RA-DMPR
	Outdoor Air Damper Position	OA-DMPR
	Heating Coil Valve Position	HWC-VLV
	Cooling Coil Valve Position	CHWC-VLV
Pressure	Supply Fan Differential Pressure	SF-DP
	Return Fan Differential Pressure	RF-DP
	Supply Air Static Pressure	SA-SP
Airflow Rate	Supply Airflow Rate	SA-CFM
	Return Airflow Rate	RA-CFM
	Outdoor Airflow Rate	OA-CFM
Fan Speed	Supply Fan Speed	SF-SPD
	Return Fan Speed	RF-SPD
Fan Power	Supply Fan Power	SF-WAT
	Return Fan Power	RF-WAT

Extensive efforts have been dedicated to investigating machine learning models for building AFDD. Here, RF is of interest because RF shows a strong ability to deal with flexible and overlapping decision boundaries, tolerate noisy data (Gao and Bergés, 2018), and improve classification performance by reducing overfitting on decision trees (Bishop, 2006). For the cross-dataset studies, RF is applied with an 80% training size, and a 10-fold CV as the AFDD strategy.

The performance metrics used for fault detection strategy evaluation are accuracy, sensitivity, specificity (Baratloo et al., 2015), and F1-score (Hripcsak and Rothschild, 2005). Accuracy measures a strategy's ability to identify both abnormal and normal samples correctly. Sensitivity, known as the true positive rate (TPR), measures a strategy's ability to identify abnormal samples. Specificity, known as the true negative rate (TNR), measures a strategy's ability to identify normal samples. F1-score, alternative to accuracy, measures a strategy's ability to identify both abnormal and normal samples correctly. Accuracy, TPR, TNR and F1-score are calculated by equations as:

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{A.1}$$

$$Sensitivity(TPR) = \frac{TP}{TP + FN} \tag{A.2}$$

$$Specificity(TNR) = \frac{TN}{TN + FP} \tag{A.3}$$

$$F1 - score = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{A.4}$$

In these equations, true positive (TP) denotes the number of abnormal samples correctly identified, the false negative (FN) denotes the number of abnormal samples incorrectly identified as normal; true negative (TN) denotes the number of normal samples correctly identified as normal, while false positive (FP) denotes the number of normal samples incorrectly identified as anomalous.

In the cross-dataset study, the fault detection strategy is trained by simulated data and tested on the real building data using all 20 features. Given the interest of the study is fault detection, the focus is the discussions on overall accuracy and sensitivity. As seen in Table 4, out of 16 faults, only 2 fault cases (Return Fan Complete Failure, OA Damper Stuck Fully Closed) achieve over 0.90 accuracy and 0.85 sensitivity, and 3 fault cases (Cooling Coil Valve Stuck Fully Closed, Heating Coil Leaking - Stage 2 (1.0 GPM), Return Fan at 30% SPD) achieve over 0.85 accuracy and 0.75 sensitivity. The remaining 11 cases were performed with accuracy or sensitivity below 0.5. Thus, in most fault cases, fault detection strategy trained by simulated building data and tested on real building data cannot reach comparable fault detection performances to those of models trained and tested by simulated building data, as shown in (Huang et al., 2022).

The conclusion is that the RF strategy shows a degradation of performance when compared to the performance from the model where both training and test data are from the same source. It is hence of interest to investigate further as to what the cause is for this performance degradation. A hypothesis is that, although the simulation model is validated from a physics perspective, i.e., the absolute values of key measurements and the fault symptoms are similar to those from a real building, the features generated from a simulation model, from a data science perspective (e.g., distribution), differ from those in the real building dataset. Therefore, a statistical test is conducted to evaluate the differences.

Table A.3: Random Forest trained by simulated data and tested by real data (*: faults being detected with > 0.90 accuracy and > 0.85 sensitivity. **: faults being detected with > 0.85 accuracy and > 0.75 sensitivity)

Fault Type	Accuracy	Sensitivity	Specificity	F1-Score
AHU Duct Leaking Fault - After Supply Fan	0.51	0.79	0.22	0.62
AHU Duct Leaking Fault - Before Supply Fan	0.50	0.01	0.99	0.03
Return Fan Complete Failure*	0.94	0.88	1.00	0.93
Heating Coil Valve Leaking - Stage 1 (0.4GPM)	0.50	0.00	1.00	0.01
Heating Coil Valve Leaking - Stage 2 (1.0GPM)**	0.89	0.79	1.00	0.88
Heating Coil Valve Leaking - Stage 3 (2.0GPM)	0.50	0.00	1.00	0.00
Cooling Coil Valve Stuck Fully Closed**	0.92	0.83	1.00	0.91
Cooling Coil Valve Stuck Fully Open	0.50	0.00	1.00	0.00
Cooling Coil Valve Stuck 15% Open	0.54	0.33	0.76	0.35
Cooling Coil Valve Stuck 65% Open	0.50	0.00	1.00	0.00
OA Damper Stuck - Fully Closed*	0.96	1.00	0.93	0.97
OA Damper Leaking - 45% Open	0.50	0.00	1.00	0.00
OA Damper Leaking - 55% Open	0.50	0.00	1.00	0.00
Cooling Coil Valve Control Unstable	0.49	0.77	0.21	0.60
Cooling Coil Valve Reverse Action	0.62	0.24	1.00	0.36
Return Fan at 30% SPD**	0.90	0.81	1.00	0.89

Kolmogorov-Smirnov Test (KS test) (Massey Jr, 1951) is a nonparametric statistical test to determine how significantly different real dataset vs. simulated datasets is for fault detection. KS test is widely used since it does not need assumptions on the data distribution. The KS two-sample test hypothesis is defined as:

H_0 : Two samples collected from different sources come from the same distribution

H_a : Two samples collected from different sources do not come from the same distribution

The test statistic is defined as $D = |E_1(i) - E_2(i)|$, where E_1 and E_2 are the empirical functions for the two samples.

Considering fault detection, in either simulated or the real building dataset, there exist both faulty and fault-free conditions. Two KS tests are conducted on the 20 features (see Table A.2): the first KS test is on a simulated dataset to identify a subset of the features that mostly differ in the faulty vs. fault-free condition; (2) given the subset features, the second KS test is to identify the similar features in comparing simulated data (fault and fault-free combined) vs. real data (fault and fault-free combined). For both KS tests, 0.4 is used as a cut-off threshold here as in (Steinskog et al., 2007).

Table A.4: Number of feature subsets by two KS tests for each fault test

Fault Type	# selected features
AHU Duct Leaking Fault - After Supply Fan	11
AHU Duct Leaking Fault - Before Supply Fan	11
Return Fan Complete Failure	7
Heating Coil Valve Leaking - Stage 1 (0.4GPM)	13
Heating Coil Valve Leaking - Stage 2 (1.0GPM)	3
Heating Coil Valve Leaking - Stage 3 (2.0GPM)	12
Cooling Coil Valve Stuck Fully Closed**	6
Cooling Coil Valve Stuck Fully Open	12
Cooling Coil Valve Stuck 15% Open	10
Cooling Coil Valve Stuck 65% Open	9
OA Damper Stuck - Fully Closed	14
OA Damper Leaking - 45% Open	14
OA Damper Leaking - 55% Open	11
Cooling Coil Valve Control Unstable	0
Cooling Coil Valve Reverse Action	12
Return Fan at 30% SPD	5

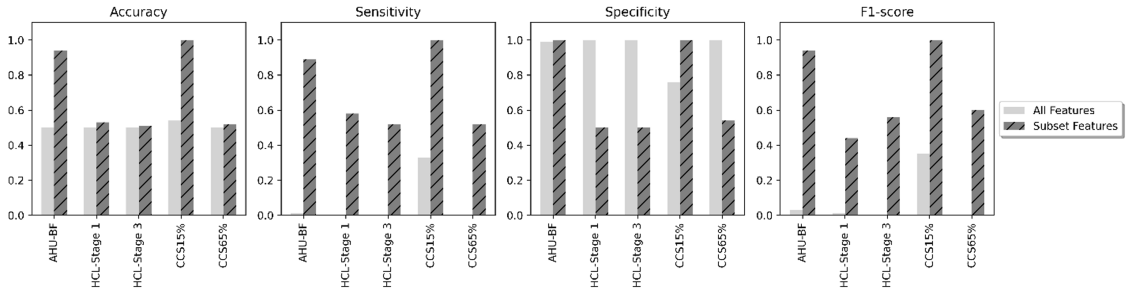

Figure A.2: Performances comparisons between using all features and sub features. (1) AHU-BF: AHU Duct Leaking Fault - Before Supply Fan, (2) HCL-Stage 1: Heating Coil Valve Leaking - Stage 1 (0.4GPM), (3) HCL-Stage 3: Heating Coil Valve Leaking - Stage 3 (2.0GPM), (4) CCS15%: Cooling Coil Valve Stuck 15% Open, (5) CCS65%: Cooling Coil Valve Stuck 65% Open.

The final number of subset features selected by the two KS tests for each fault test is summarized in Table A.4. It is interesting to observe there are no common features selected for Cooling Coil Valve Control Unstable. This is because, for an unstable control, the fault indicator is typically the frequency of a control device position change. That is, the control device oscillates with a much higher frequency than that from the baseline. Such feature may need wavelet-based approaches instead statistical KS test to be selected.

The next step is to use the selected subset features for the cross-datasets studies again to explore whether these features can help to improve fault detection strategy performances. Please note this is not to develop a better fault detection strategy, instead, this is to understand how training data affects a fault detection method's performance (accuracy and scalability). By examining more closely the subset features for the cross-dataset studies, it is interesting to see whether the accuracy of certain subset of a training dataset would affect the developed fault detection strategy.

The subset features identified by the two KS tests for faulty data in Table 5 are

used by the random forest strategy. Since there are no subset features identified by the KS test for the case Cooling Coil Valve Control Unstable, meaning that features from simulated datasets cannot capture any characteristics of those from real datasets, this case is excluded for the comparison study. As a result, it is of great interest in investigating the 10 cases (see Table 4) with accuracy or sensitivity below 0.75 when the full feature set was used. The RF strategy is trained using simulated building data on the subset features with respect to each fault test. The model is then tested on the real building data. The results show that the fault detection performance improved for 5 out of the 10 fault tests. These include (1) AHU Duct Leaking Fault - Before Supply Fan, (2) Heating Coil Valve Leaking - Stage 1 (0.4GPM), (3) Heating Coil Valve Leaking - Stage 3 (2.0GPM), (4) Cooling Coil Valve Stuck 15% Open, (5) Cooling Coil Valve Stuck 65% Open (see Figure A.2). Therefore, it is concluded the RF strategy using the selected features identified by KS tests may improve the accuracy and sensitivity for some fault cases.

In summary, cross-dataset study between real and simulated building data indicates that simulated building data differ from real ones in terms of statistical learning, although they are validated to be similar by real building data from a physical perspective. KS test assists in identifying similar features between real and simulated building data, which indirectly indicates that simulated building data are not always similar to real ones because a fraction of similar features is less than 30%. However, AFDD strategy incorporating these identified features show promises to improve the sensitivity for some fault cases, meaning that these features are an important component in true fault instances. Specifically, for the 3 fault cases from the Equipment category, cross-dataset experiment on the full feature set was able to detect 1 fault, and cross-dataset experiment on the selected feature set was about to detect the additional 1 fault. For the 10 fault cases from Controlled Device category, cross-

dataset experiment on the full feature set was able to detect 3 faults, and using the selected feature sets, additional 4 fault cases were able to be detected. For the 3 fault cases from Controller category, KS selected features have no improvements for detection. This cross-dataset study raises a warning for data-driven AFDD strategy development using simulated fault data. Clearly, under the statistical learning lens, simulated data often contain different information (e.g. distribution difference of Heating Coil Discharge Air Temperature in Return Fan at 30% SPD) from real building data. Different learning strategies, such as transfer learning, may need to be explored for this purpose.