2022 8th International Conference on Control, Decision and Information Technologies
CoDIT'22 | Istanbul, Turkey / May 17-20, 2022

Fault Detection and Diagnosis in AHU System with Data Driven Approaches

Yanis Masdoua, Moussa Boukhnifer and Kondo H. Adjallah

2022 8th International Conference on Control, Decision and Information Technologies (CoDIT) | 978-1-6654-9607-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/CODIT55151.2022.9803907

***Abstract*— Energy consumption in buildings has become a real concern for scientists and seeking to reduce this consumption is essential. Heating, ventilation, and air conditioning (HVAC) systems account for more than 50% of this consumption. One of the solutions to reduce this excessive consumption is to detect and diagnose faults that can appear instantaneously and quickly with fault diagnostic detection systems (FDD) based on artificial intelligence. The paper presents a strategy based on a data-driven approach for the detection and diagnosis of sensor faults that may appear in the Air Handling Unit (AHU) systems. A Decision Tree, Random Forest and SVM algorithm were used to detect and diagnose temperature sensor faults occurring in the AHU. The comparison between these methods shows that the Random Forest gives the best result with 96% accuracy.**

I. Introduction

Today's world faces many challenges. One of the most worrying is energy consumption. Energy consumption concerns all the fields, whether it is the industry, the transport, and especially buildings. Buildings are responsible for more than 40% of the world's energy consumption [1]. It is more than necessary to do everything possible to reduce excessive energy consumption while keeping a certain comfort for the world population. One of the solutions is to provide optimal control of Heating, Ventilation, and Air Conditioning (HVAC) systems which are responsible for more than 60% of the energy consumption in buildings [2]. The concern with heating and cooling systems is that they are subject to numerous failures during their operation. For this purpose, FDD algorithms are used, which are however underutilized in the real world. Building operators who use FDD estimate that a consumption reduction of more than 7% can be achieved only with good and timely fault detection [3].

There are several approaches to designing an FDD system, we can mention the so-called Rule-based methods [4], which is a method that usually requires human intervention, especially of an expert to establish rules to be followed in order to ensure diagnosis. We also have what we call Model-based methods [5], which require a perfect knowledge of the environment and complex mathematical modeling of the latter and the systems involved without forgetting the various interactions between them. Face to this complexity, this approach is somewhat neglected by the community and developers of FDD systems. To overcome these problems related to these methods mentioned above, we have interested to the Data-driven methods [6], which do not require any prior knowledge and which only consider operational data.

Data-driven methods have been widely used in recent years, they can be found in several fields, including transport [7] [8], energy [9], healthcare [10] and many others. Data-driven methods can be divided into two categories, the supervised, and the unsupervised approachs. The first one stipulates that the fault is known a priori and a mapping between the features and the fault is made. For the unsupervised approach, the classification algorithm is fed with input data, and the clustering is done based on the similarity between the different samples according to the features.

The AHU is the most used component in HVAC systems. A lot of work, especially based on data-driven methods has been done to detect and diagnose faults that may occur during the operation of the system [11]. Among the techniques used, many are based on Neural Networks [12]. However, we can also find work that relies on a machine learning methods that are used in many problems related to classification, the Support Vector Machines SVM [13].

In this paper, we propose to compare three data-driven methods used for sensor-related fault classification. The paper is structured as follows: Section II presents the system and the data used in this work. Section III provides a theoretical overview of the data-driven methods used in this work, the Decision Tree, SVM, and the Random Forest method. Section IV presents the obtained results with the different methods for the classification of temperature sensor-related faults. Finally, a conclusion, in which we give the advantages and limitations of the approaches as well as the improvements that can be made in future works.

II. Materials and Data

The used data in this work was generated by Pacific Northwest National Laboratory. The building model is developed using two software, EnergyPlus for modeling the building envelope [14] and Dymola for modeling the HVAC system [15].

Yanis Masdoua, Université de Lorraine, LCOMS, 57000, Metz, France yanis.masdoua@univ-lorraine.fr
Moussa Boukhnifer , Université de Lorraine, LCOMS, 57000, Metz, France, moussa.boukhnifer@univ-lorraine.fr
Kondo H. Adjallah , Université de Lorraine, LCOMS, 57000, Metz, France, kondo.adjallah@univ-lorraine.fr

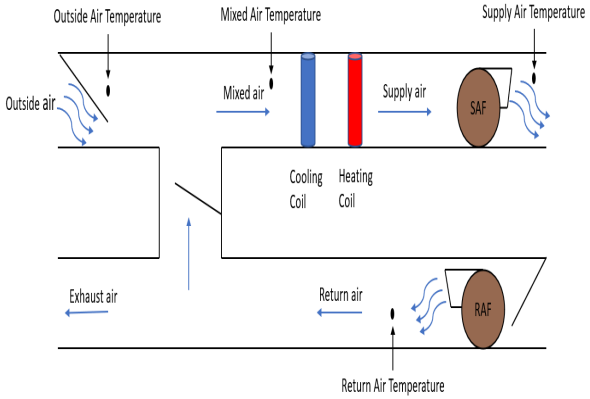

Fig. 1: AHU Structure

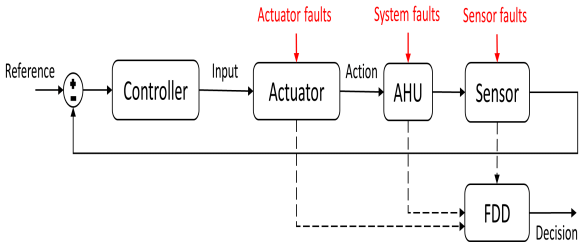

Fig. 2: General control scheme

A. System Description

The building consists of three floors, with heating and cooling provided by AHU. Five zones are provided by five variable air volume VAV boxes.

The main components of the studied mid-level AHU are supply air fan with variable frequency drive (VFD), return fan, cooling and heating control valves, cooling and heating coils, outdoor air (OA) and return air (RA) dampers (see Fig.1).

The control scheme is given in Fig 2. In this study we will focus on the FDD block, thus the classification of the faults injected to the sensor block.

B. Database

The data were recorded at a 1min interval. Faulted and unfaulted scenarios on the AHU using Typical Meteorological Year version 3 (TMY3) files for Chicago, IL. The Outside Air temperature deviation are directly imposed in the simulation model by overriding the control signals and each fault lasts for one day.

The data are collected according to two processes, the first during normal operation and the second in the presence of faults. We have six faults. The fault is an additive one for the external sensor that measures the incoming temperature to the AHU. Each fault occurs in a well-defined period, so we have six periods that we can call periods with faults and a period without faults that will serve as a reference (see table I).

Scenarios		Fault occured time
Fault	Fault intensity	
Outdoor air temperature	x' = x + 1 (°C)	2/6/17 - 2/12/17 5/9/17-5/14/17 8/7/17-8/13/17 11/6/17-11/12/1
	x' = x + 2 (°C)	2/13/17 - 2/19/17 5/16/17-5/21/17 8/14/17-8/20/17 11/13/17-11/19/17
	x' = x + 4 (°C)	2/20/17 - 2/26/17 5/23/17-5/28/17 8/21/17-8/27/17 11/20/17-11/26/17
	x' = x - 1 (°C)	2/27/17-3/5/17 5/30/17-6/4/17 8/28/17-9/3/17 11/27/17-12/3/17
	x' = x - 2 (°C)	3/6/17-3/12/17 6/6/17-6/11/17 9/4/17-9/10/17 12/4/17-12/10/17
	x' = x - 4 (°C)	3/13/17-3/19/17 6/13/17-6/18/17 9/11/17-9/17/17 12/11/17-12/17/17
Unfaulted		1/30/17-2/5/17 5/2/17-5/7/17 7/31/17-8/6/17 10/30/17-11/5/17

TABLE I: Input scenarios included in dataset [16]

C. Data Processing

1) Collection and transformation: In this first part, we present the task that has been done in order to improve the database. The data that was retrieved required some formatting since the outdoor temperature recorded was sometimes in Celsius while the data table stated that all values were recorded in Fahrenheit. We had to verify all the recorded days and compare them with the real values.

2) Data Cleaning: This step is essential before any use of machine learning algorithms, it ensures the accuracy of the data provided and verifies that the database does not contain any missing values.

Sometimes a data set may contain extreme values that fall outside the expected range and do not resemble the other data. These values are called outliers and, data-driven models should be improved by removing these outliers. Fig. 3 shows the temperature distribution in boxplot form, and gives the main temperature features, Outside Air temperature, Return Air temperature, Supply Air temperature, Mixed Air temperature contain outliers, that have been removed. These outliers are often due to problems of sensors acquisitions or information quantification.

3) Features Selection: To maximize the performance and reduce the training time of the classification algorithms, an attribute (feature) selection step is necessary. Unlike models based on Deep Neural Networks, where sometimes the selection is done automatically by the network. In this work the selection is done based on a correlation between the different attributes and the objective of this classification.

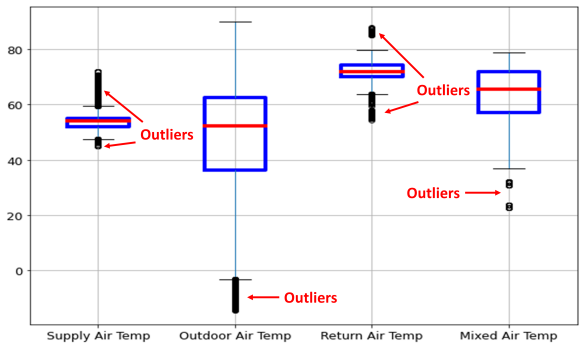

Fig. 3: Removing outliers from Dataset

Finally, we divide the database into two parts, one part of 75% will be used for training the classification models and the rest part (25%) for testing.

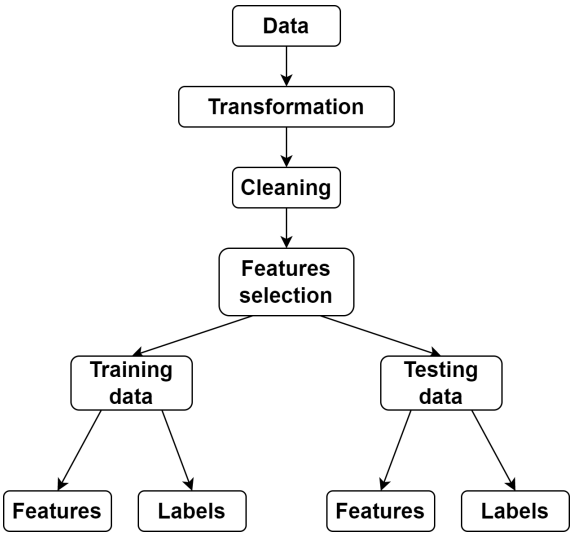

Fig. 4: Data preparation process

III. Data-Driven Methods

In this section, some theoretical notions are presented of the methods used in this work. Classification is a supervised learning technique that maps input data to output labels based on many samples input-output pairs identified during the training phase [17]. In this section, we present the theory behind the three algorithms used in this work, the Decision Tree, the Random Forest and the SVM.

A. Decision Tree

A Decision Tree is a predictive algorithm that can be used to solve both classification and regression tasks. For classification tasks, the model will be referred to as a classification tree and for regression tasks as a regression tree.

The abbreviation CART stands for Classification And Regression Trees. It designates a statistical approach which constructs tree-based predictors for both regression and classification. Building a CART tree is done in two steps. The building of a maximum tree is the initial process, followed by pruning, which creates a series of ideal subtrees pruned from the maximal tree. At each step of the partitioning, a part of the space is split into two sub-parts. The nodes of the tree are associated with the elements of the partition. The root of the tree is associated with the input space, its two child nodes are associated with the two sub-parts obtained by the first partitioning step and so on. To optimize the cut, in classification we often talk about impurity of the nodes. This is most often represented by the Gini index [18] which is defined by the following equation:

$$Gini(n) = \sum_{k=1}^{K} p(k|t)(1 - p(k|t)) \quad (1)$$

where p is the proportion of observations of class k in the node n.

We want to reduce the Gini purity function in classification to maximize the homogeneity of the nodes that arise.The CART algorithm's second phase is pruning, which involves finding the best pruned subtree of the maximal tree. Pruning is a pattern selection process in which the pruned subtrees serve as the patterns. This approach minimizes a penalized criterion whose penalty is proportional to the tree's number of leaves.

The structure of the Decision Tree is given in Fig. 5. It contains the principal node (root node), branches, and leaf nodes. Each node in the tree represents a test on an attribute and the result (decision) is given on the descending branch. The leaves represent a result that can be a class or category for a classification Decision Tree or a numerical quantity for a regression problem. In our work, we will use the Decision Tree for the classification of faults related to the outdoor temperature sensor, so we will have the data (attributes) that will be as input data for the model and at each node, a test is performed and will give a decision on each leaf, which in our case will be a numerical value that will represent a class. A leaf with a value of 0 for normal operation, -4, -2, -1, 1, 2, 4 for faults -4°C, -2°C, -1°C, 1°C, 2°C, 4°C respectively. One of the advantages of Decision Trees is that their results are easy to read and interpret. The data can also generate important information about probabilities. Compared to other decision techniques, Decision Trees require less effort for data preparation. It is rarely necessary to perform input normalization or scaling before training the algorithm, unlike Neural Networks. A slight change in the training data can lead to a major change in the algorithm output. This small issue can be quickly overcome with the Random Forest algorithm.

B. Random Forest Algorithm

The Random Forest term was introduced by [19], which is composed of several Decision Trees generated using a random subset of data. In classification tasks, the principle is the following, each Decision Tree will converge to a result (a class) independently of the other trees, then a vote

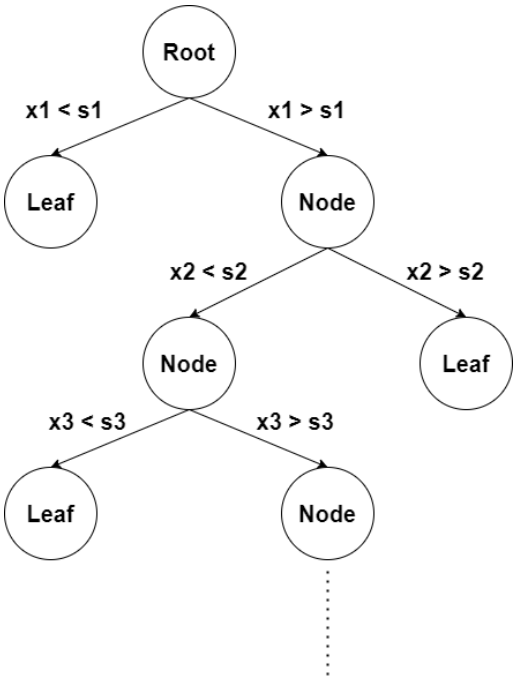

Fig. 5: Decision Tree structure

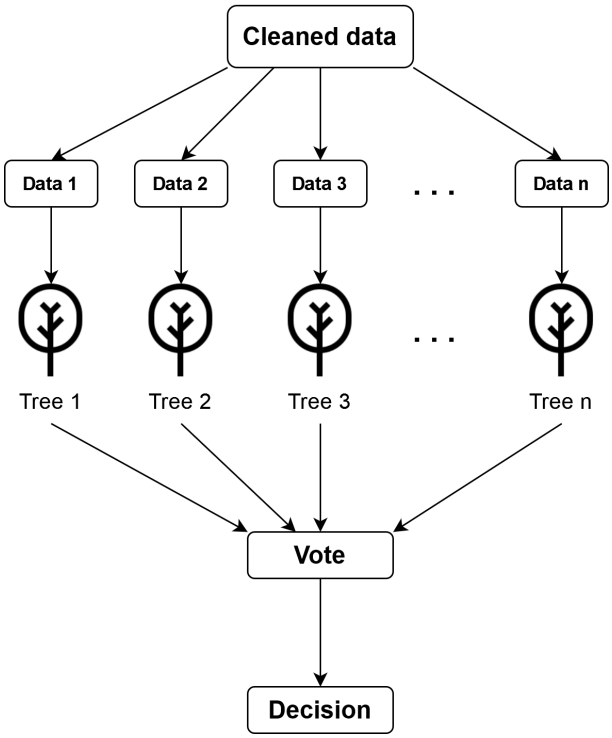

Fig. 6: Random Forest structure

is performed to choose the result that the Random Forest algorithm will give as output. During the training phase, we will use a certain number of features to train the model, it is essential to use the same features during its use, test, or validation. For example, we train the algorithm with a set of temperature data, when using the algorithm, the same types of temperatures must be used. The Random Forest is capable of handling high dimensional databases with many features, which is very useful in the real world.

There are two different methods of inserting randomness into a Random Forest algorithm. One is based on the data selection for each tree, and the other is how the split criteria are chosen. As said before, the Decision Trees in a Random Forest use a little different data set. The final result is based on the votes of all the Decision Trees. As a result, abnormalities tend to be smoothed because the data causing the abnormalities is found in some, but not all, of the Decision Trees, while the more general data is found in most or all of the trees. When each tree is created, it has a unique set of data and this data is a random subset of all available data. This technique is known as bootstrapping. The second way a Random Forest adds randomness to a Decision Tree is by deciding which condition to place at each node to split the tree. For a given feature, the split will be at the point that maximizes the information gained on the tree.

One of the biggest drawbacks of the Random Forest and Decision Tree algorithms we have discussed is that they do not extrapolate. They have a very low generalization of problems and they are very specialized in their tasks. The structure of the Random Forest is given in Fig. 6.

C. Support Vector Machine

SVM (Support Vector Machine) is a machine learning algorithm mainly used for classification, but there are many uses for regression tasks. It can solve both linear and non-linear problems by finding an optimal separation hyperplane that is specified by a number of support vectors. The central idea of SVM is to find a maximal marginal hyperplane (MMH) that best divides the data set into classes, in our case, dividing the data into groups of faulty and unfaulted data.

In a binary case where we have only two classes, the separation is simple to perform. The separating hyperplane is represented by the equation :

$$H(x) = w^T x + b \tag{2}$$

where w is an m-dimensional vector and b is a term. The decision function, for an example x, can be expressed as follows:

$$\begin{Bmatrix} Class = 1 & if & H(x) > 1 \\ Class = -1 & if & H(x) < -1 \end{Bmatrix} \tag{3}$$

The hyperplane $w^T x + b = 0$ represents a separating hyperplane of the two classes, and the distance between this hyperplane and the nearest example is called the margin 7.

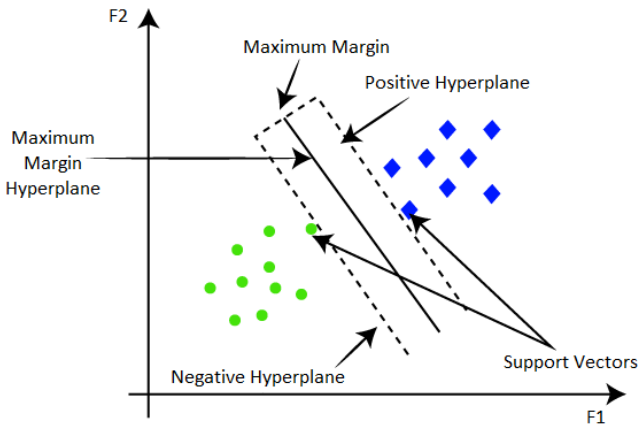

Fig. 7: Representation of the hyperplane that separates the Support Vectors Machine

The region between the two hyperplanes $w^T x + b = -1$ and $w^T x + b = +1$ is called the generalization region of the learning machine. The larger this region is, a better generalization capacity is obtained. Support Vector Machines are in their origin binary. However, real world problems are in most cases multiclass. In this cases, one does not try to assign a new instance to one of two classes but to one of many, i.e. the decision is no longer binary and a single hyperplane is not sufficient.
The multiclass SVM algorithms simplify the multiclass issue to a composite of numerous biclass hyperplanes, allowing the decision borders between the different classes to be drawn. The best known and oldest method is One-vs-rest (OvR) [20]. It consists in determining for each class k a hyperplane $H_k(w_k, b_k)$ separating it from all other classes.

For a k class problem, a hyperplane H_k is defined for each class k by the following decision function:

$$\begin{aligned} H_k(x) &= sign(\langle w_k, x\rangle + b_k) \\ &= \begin{Bmatrix} +1 & if & f_k(x) > 0 \\ 0 & otherwise & \end{Bmatrix} \end{aligned} \tag{4}$$

The returned value of the hyperplane allows to know if x belongs to the class k or not. To know its belonging to a class, we present x to all the hyperplanes, which gives the decision function of the equation:

$$k^* = Max(H_k(x)) \tag{5}$$

An SVM can be linear or nonlinear. The implementation of linear SVM models differs in complexity based on the number of features used. In the case where the classes are linearly separable, the hyperplane will be presented by a line in the case of two features and a two-dimensional plane when it is three features (see Fig. 7). In the case where the classes are non-linearly separable, classification is more difficult. This is called a non-linear classifier, and a Kernel transformation is necessary to increase the size of the support vectors, making them linearly separable.

IV. Experimental results

For the experimental part, all the data processing and development of the classifier for the FDD was done on Python. The manipulation and processing of the data was done with the Pandas and Numpy libraries. The development of machine learning models and statistics were done with the Sci-kit learn library.
For more reliability in the results, the k-Fold cross-validation is used. k-Fold Cross-validation is a resampling technique used to validate and estimate the skill of machine learning models against a limited sample of data. It takes a single parameter k, which is the number of sub-groups obtained from dataset, in our work, we choose $k = 10$.
By collecting the processed data (Fig. 4), the training of the model is performed under sci-kit learn environment. Several tests were performed to optimize the algorithm.
Table II represents how the depth of the trees affects the performance of the classifier. We can observe that by increasing the size of the trees we have a better performance. The same goes for the number of trees in the forest. A better performances are obtained when increasing the number of trees (Table III). Remember that the Random Forest algorithm consists of a vote of several independent Decision Trees with each time different inputs. So the fact of adding trees, if they have been well optimized, we will have a better estimation.

Size of Trees	classification accuracy (%)
10	78.53
15	90.13
20	95.58
25	96.27
30	**96.46**

TABLE II: Results of classification according to the depth of trees in the forest

Trees in forest	classification accuracy (%)
20	92.64
50	93.91
100	94.12
250	96.37
500	**96.46**

TABLE III: Results of classification according to the number of trees in the forest.

The completed optimal model (Random Forest) achieves an average accuracy of 96.46% of classification. The confusion matrix is given in Fig. 8. We can see that in the most extreme cases (-4 °C and +4 °C) the accuracy is higher than 98%. On the other hand, the classifier has a little difficulty to diagnose and separate the unfaulty and +1°C faulty cases, but still achieve a satisfactory scores of 93.6% and 93.1% respectively.

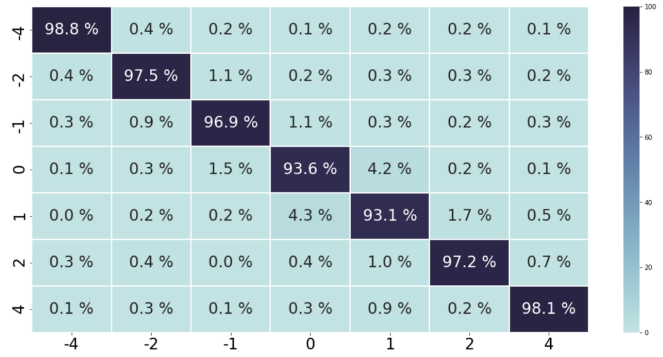

Fig. 8: Confusion matrix

Table IV shows the comparison between the different algorithms used SVM, Decision Tree, and Random Forest. From this comparison, we can say two things, the first one is that the two algorithms Decision Tree and Random Forest are better than the SVM. Second, we note that the Random Forest is slightly better than the Decision Tree, we must also take into account that the Random Forest has a better generalization than the Decision Tree.

Models	classification accuracy (%)
SVM	74.81
Decision Tree	93.39
Random Forest	**96.46**

TABLE IV: Comparison between the obtained results with the different models SVM, Decision Tree, and Random Forest.

V. Conclusion

Fault detection and diagnosis system based on the Random Forest algorithm for an AHU system has been developed in this paper. The obtained results were satisfactory with a performance of more than 96%. These results validate the data-driven approaches applied to complex systems where modeling is often very challenging. The algorithm was able to detect the fault and above all to diagnose it reliably.
Future research is planned to generalize fault detection and diagnosis in the AHU, the FDD system will consider not only sensor-related faults but also actuator faults and intrinsic faults directly related to the AHU system.

References

[1] V. Harish and A. Kumar, "A review on modeling and simulation of building energy systems," Renewable and Sustainable Energy Reviews, vol. 56, pp. 1272–1292, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032115014239

[2] R. Kalbasi, A. Shahsavar, and M. Afrand, "Incorporating novel heat recovery units into an AHU for energy demand reduction-exergy analysis," Journal of Thermal Analysis and Calorimetry, vol. 139, no. 4, pp. 2821–2830, Feb. 2020. [Online]. Available: https://doi.org/10.1007/s10973-019-09060-4

[3] H. Kramer, G. Lin, C. Curtin, E. Crowe, and J. Granderson, "Building analytics and monitoring-based commissioning: industry practice, costs, and savings," Energy Efficiency, vol. 13, no. 3, pp. 537–549, Mar. 2020. [Online]. Available: https://doi.org/10.1007/s12053-019-09790-2

[4] Y. Guo, J. Wang, H. Chen, G. Li, R. Huang, Y. Yuan, T. Ahmad, and S. Sun, "An expert rule-based fault diagnosis strategy for variable refrigerant flow air conditioning systems," Applied Thermal Engineering, vol. 149, pp. 1223–1235, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1359431118323639

[5] H. Habibi, I. Howard, and S. Simani, "Reliability improvement of wind turbine power generation using model-based fault detection and fault tolerant control: A review," Renewable Energy, vol. 135, pp. 877–896, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960148118315106

[6] H. Chen and B. Jiang, "A review of fault detection and diagnosis for the traction system in high-speed trains," IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 2, pp. 450–465, 2019.

[7] A. Benterki, M. Boukhnifer, V. Judalet, and C. Maaoui, "Artificial intelligence for vehicle behavior anticipation: Hybrid approach based on maneuver classification and trajectory prediction," IEEE Access, vol. PP, pp. 1–1, 03 2020.

[8] A. Benterki, M. Boukhnifer, V. Judalet, and M. Choubeila, "Prediction of surrounding vehicles lane change intention using machine learning," in 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), vol. 2, 2019, pp. 839–843.

[9] K. Amasyali and N. M. El-Gohary, "A review of data-driven building energy consumption prediction studies," Renewable and Sustainable Energy Reviews, vol. 81, pp. 1192–1205, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032117306093

[10] P. K. Yeng, L. O. Nweke, A. Z. Woldaregay, B. Yang, and E. A. Snekkenes, "Data-driven and artificial intelligence (ai) approach for modelling and analyzing healthcare security practice: A systematic review," in Intelligent Systems and Applications, K. Arai, S. Kapoor, and R. Bhatia, Eds. Cham: Springer International Publishing, 2021, pp. 1–18.

[11] C. Fan, Y. Liu, X. Liu, Y. Sun, and J. Wang, "A study on semi-supervised learning in enhancing performance of AHU unseen fault detection with limited labeled data," Sustainable Cities and Society, vol. 70, p. 102874, Jul. 2021, publisher: Elsevier BV. [Online]. Available: https://scholars.cityu.edu.hk/en/publications/a-study-on-semisupervised-learning-in-enhancing-performance-of-ahu-unseen-fault-detection-with-limited-labeled-data(b8e1383c-0958-4090-95be-6ef1de341abe).html

[12] M. Karami and L. Wang, "Fault detection and diagnosis for nonlinear systems: A new adaptive Gaussian mixture modeling approach," Energy and Buildings, vol. 166, pp. 477–488, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378778817326452

[13] K. Yan, Z. Ji, H. Lu, J. Huang, W. Shen, and Y. Xue, "Fast and accurate classification of time series data using extended elm: Application in fault diagnosis of air handling units," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49, no. 7, pp. 1349–1356, 2019.

[14] D. B. Crawley, C. O. Pedersen, L. K. Lawrie, and F. C. Winkelmann, "Energyplus: Energy simulation program," ASHRAE Journal, vol. 42, pp. 49–56, 2000.

[15] M. Dempsey, "Dymola for multi-engineering modelling and simulation," in 2006 IEEE Vehicle Power and Propulsion Conference, 2006, pp. 1–6.

[16] G. Jessica, L. Guanjing, H. Ari, I. Piljae, and C. Yan, "Dataset for building fault detection and diagnostics algorithm creation and performance testing," 2020.

[17] D. A. Pisner and D. M. Schnyer, "Chapter 6 - Support vector machine," in Machine Learning, A. Mechelli and S. Vieira, Eds. Academic Press, 2020, pp. 101–121. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128157398000067

[18] B. H. Menze, B. M. Kelm, R. Masuch, U. Himmelreich, P. Bachert, W. Petrich, and F. A. Hamprecht, "A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data," BMC Bioinformatics, vol. 10, no. 1, p. 213, Jul. 2009. [Online]. Available: https://doi.org/10.1186/1471-2105-10-213

[19] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, Oct. 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

[20] Support Vector Machines. John Wiley, 2009, ch. 7, pp. 89–132. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470503065.ch7