Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Data driven fault detection and diagnostics for hydronic and monitoring systems in a residential building

To cite this article: M A F Abdollah *et al* 2022 *J. Phys.: Conf. Ser.* **2385** 012012

Data driven fault detection and diagnostics for hydronic and monitoring systems in a residential building

M A F Abdollah[1,] **R Scoccia**[1] **and M Aprille**[1]

[1] Department of Energy, Politecnico di Milano, Via R. Lambruschini 4, 20156, Milan (Italy)

Abstract. Buildings are responsible for 40% of the global energy use and associated with up to 30% of the total CO2 emissions. The drive to reduce the environmental impact of the built environment was the catalyst to the increasing installation of meters and sensors to monitor the energy use and environmental monitoring. This is key to cost effective Fault Detection and Diagnostics (FDD) which guarantees enhanced thermal comfort for the occupants and reduction in energy use. Most of FDD research work in buildings was focused on the commercial buildings due to the higher consumption and the higher saving potential, while limited work was directed towards residential buildings. This paper investigates the usage of two supervised machine learning algorithms, namely Random Forest, K nearest neighbour, to detect and diagnose twelve faults in both the monitoring system of the indoor/outdoor conditions, and the hydronic circuit inside an apartment located in Milan using minimal features that are easy to access and inexpensive to monitor to cut down in both computational and financial costs. The thermal zones are being conditioned using an electric air to water heat pump connected to fan coils for cooling and radiant floor for heating. The faults include valve leakage, faulty temperature sensors and recirculating pump's inadequate flow rate. The faults were modelled in a Modelica based detailed model of the apartment. After tuning the hyper-parameters of all three algorithms, the Receiver Operator Characteristics curve for each fault were compared for each algorithm to compare the optimal one to be used. The Random Forest algorithms showed the highest accuracy with almost 89% across the twelve faults. Generalization of the trained algorithm across different weathers were tested but the results were not promising.

1 Introduction

Buildings are complex integrated systems consists of multiple components and subsystems and sensors. According to the United Nations Environment Programme, 39% of the carbon dioxide emissions is attributed to the building systems [1]. Due to malfunctioning in the control, monitoring systems and equipment, 30% of the energy use in buildings is wasted [2], [3]. It is estimated that 0.011 to 0.57 TWyr of additional energy consumption is due to faulty operations in buildings systems in the US [4]. Automated Fault Detection and Diagnosis (AFDD) provide a solution to tackle this problem [4]-[5]. Many AFDD methods have been developed in both component level and whole building level. Katipamula, Brambly, and Kim provided a comprehensive classification and review of system AFDD for Heating, Ventilation and Air conditioning (HVAC) [6]–[8]. According to this classification, AFDD methods can be divided into two main categories: qualitative and quantitative model-based methods (rule-based and physics-model based), and process history-based methods (mostly data driven and machine learning-based methods). Qualitative and quantitative methods are very popular among engineers and researcher as they are clear and easy to understand, however, due to their low scalability, their specificity and high development cost, the market adaptation has been low [9]. Therefore, process history based AFDD methods received greater attention in the past decade due to its scalability and low implementation cost. However, Process history-based performance depends heavily on the quality of the data used for training [10]. The majority of the AFDD methods are developed and evaluated using simulated system data [11], [12]. This is because implementing faults, cleaning and handling real buildings data are challenging. Naturally occurred faults in the building might reduce or eliminate the impact of the faults itself [13]. There are also the issues with the data quality coming from the Building Automation systems (noisy sensors, missing data, sensor faults and sensors accuracy). Many studies attempted to detect and diagnose anomalies in buildings by analysing the daily energy patterns[14], [15]. Few studies have been dedicated to study the applicability of FDD strategies in the residential sector. In [16] the authors reviewed FDD methods for residential air conditioning systems. It showed that 65% of the installed residential systems require repairs [17]. It was reported that at least 17% of residential air conditioners are operating below their rated efficiency [18]. Consequentially, the peak demand level and the overall energy consumption increase [19].

Extensive efforts have been dedicated to investigating ML models for AFDD including Random Forest (RF) [13]–[14], [15], [16], Support Vector Machine (SVM) [13]–[15], [16], Decision Tree [15], K-Nearest Neighbor (KNN) [16], neural networks [24], [25]

Very few papers investigated any FDD algorithms on hydronic systems for heating or cooling in the residential context, mainly due to the huge cost of the infrastructure required for those to be applicable. This paper is testing the applicability of an FDD algorithm for a hydronic system for heating and cooling in a residential context, based on supervised machine learning algorithms using minimal measurements to cut down on both financial and computational costs.

In section 2 the case study will be presented. Section 3 will present the dynamic simulations performed and the faults implanted in the model. In section 4 the data driven model used and their evaluation matrices will be introduced. Section 5 will be the classification results and the attempt to generalize the trained model to be used in different weathers. Finally in the last chapter, the conclusions of the study.

2 Case study

2.1. Envelope properties, scheduling and setpoints

The apartment presented consists of five rooms as shown in Figure1. A living room with an area of 32.5 m^2 two bedrooms with an area of 18.84 and 20.07 m^2 and two bathrooms. It is adjacent to two other apartments in the North-West and North-East, while the living area and the second bedroom border on the North-West and North-East respectively with a common area that includes a stairwell and elevators. The remaining part of the perimeter walls is in contact with the external environment.

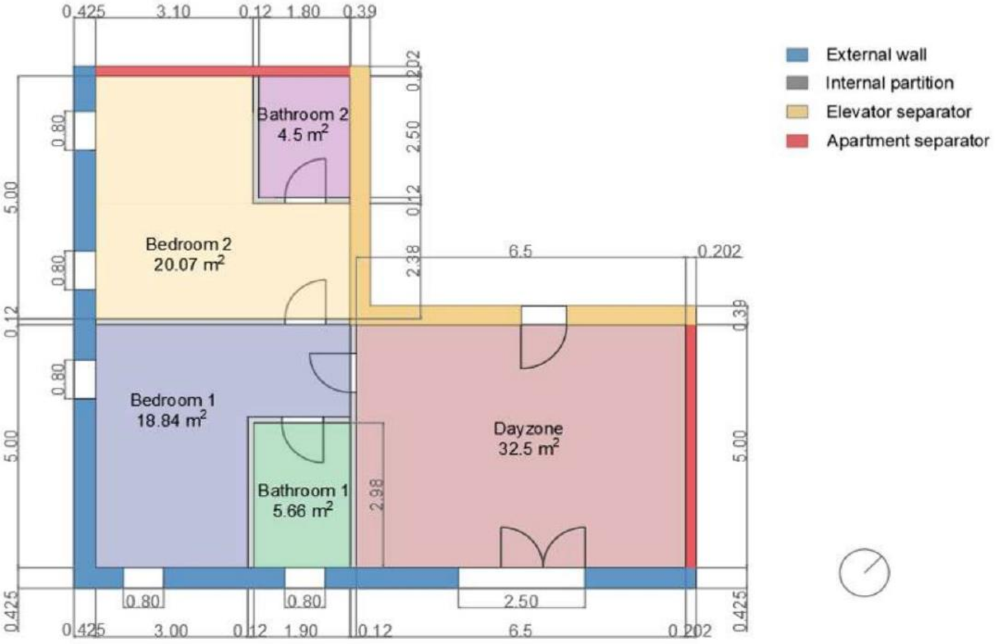

Figure 1. Apartment plan, areas of the rooms and adjacency information.

Table 1 summarizes some of the features of the apartment including the U values of envelope components, net walkable area, number of occupants.

Table 1. Case study characteristics

Net walkable surface	81.6 m^2
Local average height	2.7 m
Number of occupants	2
U values ($W/m^2/K$)	
External walls	0.225
Internal walls towards stairwell	0.685
Partition between apartments	0.267
Windows	1.31

The model considered infiltration, natural ventilation, and mechanical ventilation for both summer and winter.

For the shading system, it was assigned a different degree of opening depending on the season in question. In the cooling season they were closed 85% in the living area and 70% in the sleeping area, while in the heating season they were closed at 45% in both thermal zones. The internal loads were taken as 60 W/person for the sensible loads and 20 W/person for the latent following the occupancy schedule shown in Figure 2. The set points temperatures are set to be 20 °C with a setback of 18 °C in winter and 26 °C with a setback of 18 °C in summer corresponding to the occupancy schedule.

Journal of Physics: Conference Series **2385** (2022) 012012 doi:10.1088/1742-6596/2385/1/012012

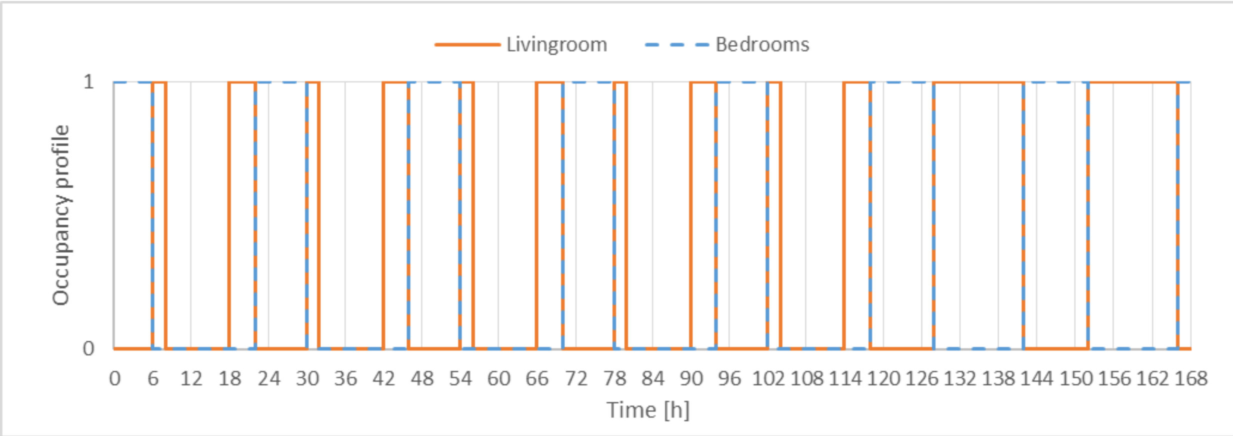

Figure 2. Occupancy schedule

2.2. Mechanical system scheme

An entirely electric plant system was considered as shown in Figure 3. The plant is characterized by an air-water heat pump connected to a radiant floor for heating and fan coils for cooling and 180-liter storage tank for the production of domestic hot water. The heat pump and the storage tank were built into the perimeter structure of the house. Finally, there is also a photovoltaic system of 23.3 m^2 area and 3.2 kWp is considered.

The hydronic system consists of a pump regulated by a flow signal, two three-way valves that regulate the distribution of water in the system, pipes and joints that distribute it where required. This system can manage n thermal zones that are served either by radiant floors or by fan coils, whose circuits can be operated using two-way valves.

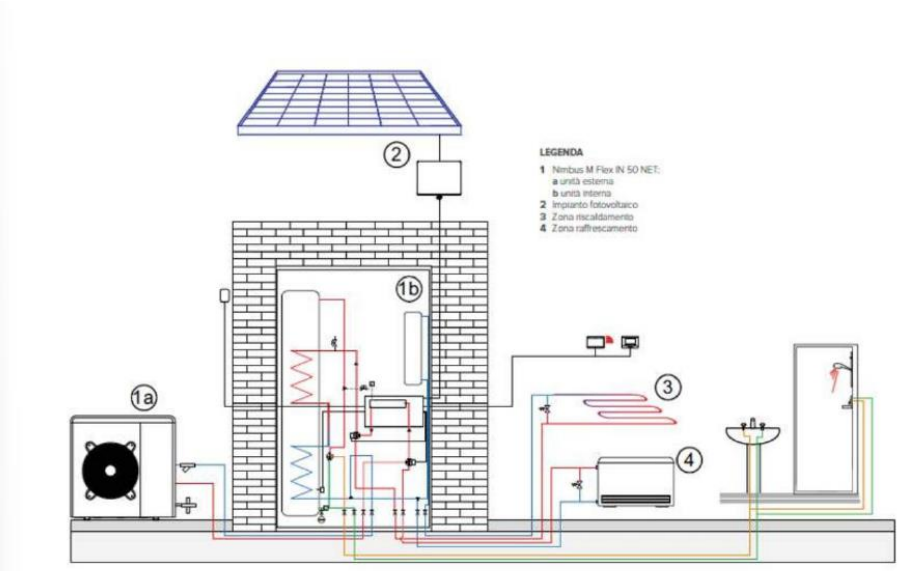

Figure 3. Schematic of the plant design [Source: Ariston].

3 Dynamic simulations and faults implemented

A digital twin was created for this project using Modelica language [14]. Buildings and IBPSA's libraries were used to model all the components [26], [27]. The simulations were run with a time step of 10 minutes. Simulations were run from 15th of December to the 1st of April in the winter case and from

1st of June till 1st of August for the Summer case. 12 faults were considered both in the hydronic system and the monitoring system. Figure 4 shows a simplified scheme of the system with the faults implemented highlighted. While Table 2 contains all the faults implemented and their description. The faults are simulated for the whole period in both summer and winter. The sensors faults are modelled by adding constant noise that ranges between -0.5°C to +1.5°C in the first case and a random noise with a mathematical function that adds a deviation to the measurements that adds up to +5°C to the readings. While the valve faults are added by increasing the leakage parameters provided from by the library in the component features. Finally for the circulating pump fault, a mathematical function was provided to the input signal of the pump to decrease the required flow rate by 10%.

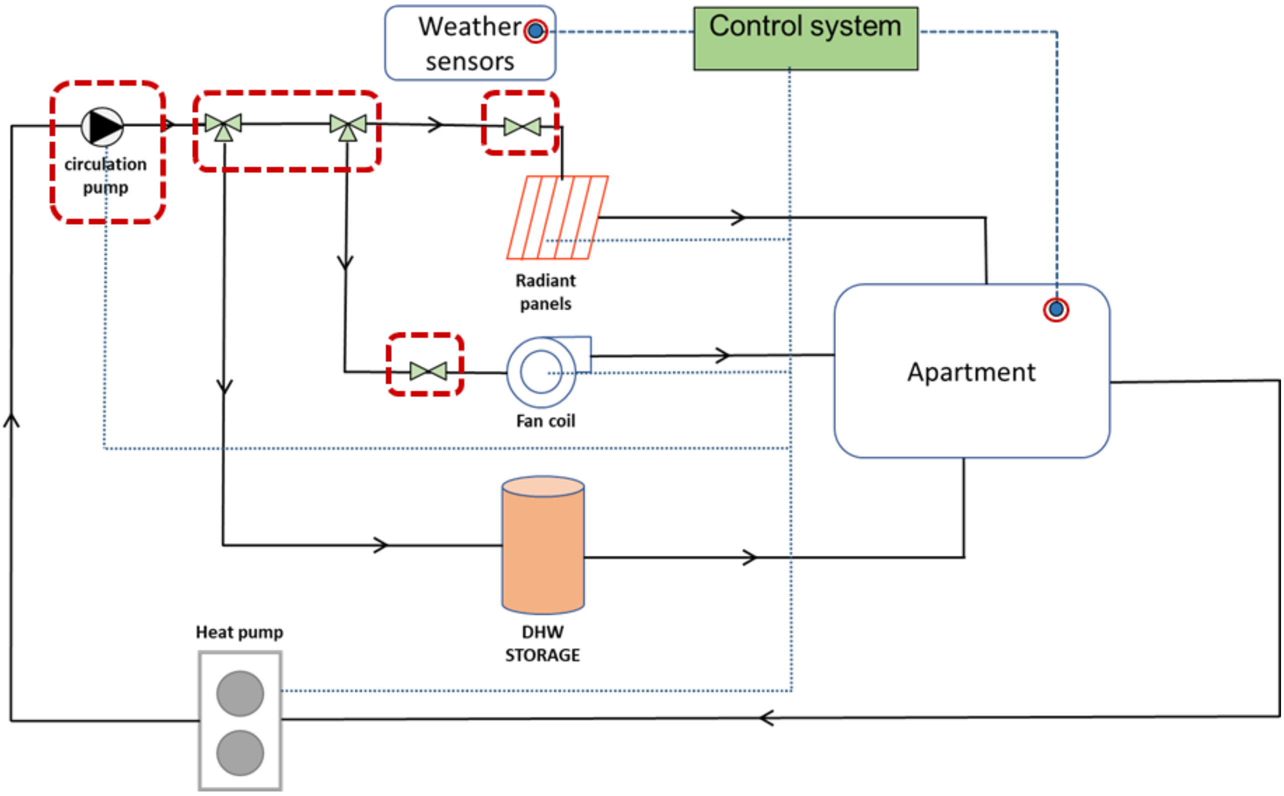

Figure 4. Simplified system scheme with the faults implemented highlighted.

Table 2. Faults implemented in the model

Fault no.	Fault	Description
1-4	Three-way valve no.1&2 leakage 1 & 5%	The flow going into the three-way valve is leaking 1 & 5% of its volume to the wrong direction
5	Two-way valve no.1 blockage 1%	1% of the flow going into the two-way valve is blocked
6-8-10	Weather, Bedroom 1 and living room dry-bulb temperature sensor with added constant noise	Random noise is added to the sensors reading. The noise is ranging from -0.5°C/+1°C

7-9-11	Weather, Bedroom 1 and living room dry-bulb temperature sensors with increasing deviations and added noise	Deviation and random noise are added to the sensors reading. The noise is ranging from -0.5°C/+1°C, while the deviation is adding up to +5°C
12	Circulating pump inadequate flow	The circulating pump is only providing 90% of the supposed flow rate.

All the faults were simulated for the entire simulation time mentioned above.

4 Automated fault detection algorithms

In this Section the algorithms used to detect faulty operation of the system will be displayed and the assessment methods used to evaluate their performance.

4.1 Data driven models

Data driven approach means that the decisions are made based on data analytics instead of intuition of an expert. In the fault detection and diagnostic context, statistical models are trained on the data from the system at hand from both healthy and faulty operations. The source of the data can be a simulation of the system or from experimental setup. The trained statistical model should be able to detect faulty operation of the system without the need of an experts or intuition about the system.

In this section the Machine Learning (ML) algorithms used are listed. In this paper RF and KNN were used. RF shows a strong ability to deal with flexible and overlapping decisions boundaries, tolerate noisy data [15], and due to the bagging concept, it reduces the probability of overfitting over decision trees [28], while KNN is computationally inexpensive and works well with low dimension data [15].

The objective of this study is to evaluate the ability of these algorithms in detecting every faulty operation and distinguish it from healthy operations, also all the other faulty operations, using minimal measurements. One vs Rest (OVR) approach is used to achieve this where every fault will be labeled as a class and compared against all other classes.

The models are trained using 70% of the data set and tested on the remaining 30%. That means that the faults are assumed to run for 31.5 days in winter case and 18 days in the summer case.

Before evaluating each model, a Hyperparameter tunning will be done to tune the algorithm to the problem at hand.

4.1.1 Random Forest

Random forests or random decision forests is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time [15]. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned [14], [15], [16]. Fundamentally, random forest decreases the probability of the overfitting problem that the decision trees suffer from. Random forests outperform decision trees, but their accuracy is lower than gradient boosted trees. In particular, trees that are grown very deep tend to learn highly irregular patterns, they overfit their training sets, i.e., have low bias, but very high variance. Random forests are a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance [28]. This comes at the expense of a small increase in the bias and some loss of interpretability, but generally greatly boosts the performance in the final model. Forests are like the pulling together of decision tree algorithm efforts. Taking the teamwork of many trees thus improving the performance of a single random tree. Though not quite similar, forests give the effects of a k-fold cross validation [16].

In this paper, random forest was used for classification purposes. The hyperparameters used for the task is specified in the Appendix.

4.1.2 K nearest neighbor

The k-nearest neighbours (KNN) algorithm is a data classification method for estimating the likelihood that a data point will become a member of one group, or another based on what group the data points nearest to it belong to [17]. The k-nearest neighbour algorithm is a type of supervised machine learning algorithm used to solve classification and regression problems. However, it is used for classification problems. No training is needed for the algorithm, simply the new points needed to be classified will be clustered depending on the smaller Euclidean distance with the nearest point of a cluster [16].

4.2 Evaluation metrics

In this study, 4 evaluation metrices are used to evaluate the performance of the model, namely accuracy, sensitivity, specificity [29] and Area Under the Curve (AUC) [30].

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. The matrices can be calculated for each class performance or for the total one. For this study, the Accuracy, sensitivity, and specificity will be evaluating the total performance, while the individual performance will be calculated using the Receiver Operating characteristic (ROC) curve.

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

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

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

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. For visualization purposes, the Confusion Matrix will be presented which is a matrix that include all the predicted labels and the true labels.

AUC – ROC curve is a performance measurement for classification problem at various thresholds settings. ROC is a probability curve and AUC represents degree or measure of separability. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. By analogy, Higher the AUC, better the model is at distinguishing between patients with disease and no disease [30].

After the Hyperparameters tuning, 3 folds cross validation will be done. Eighty percent of the data are used for training and 20% for testing.

As mentioned in the introduction chapter, the features used to train the model are inexpensive to monitor in the real building and easily accessible. The features used are as follows:

1. Thermal zones dry bulb temperatures (Bedroom 1&2, bathroom 1&2 and the living room)
2. Inlet and outlet temperatures of the heat pump.

3. Heat pump electrical consumption.

5 Results and discussion

For this study, the accuracy, sensitivity, and specificity will be evaluating the total performance, while the individual performance will be calculated using the Receiver Operating characteristic (ROC) curve.

5.1 Milano

The RF is performing better than the KNN across most of the faults. That's because the RF can grow vertically and adapt to the training data more accurately. But this comes with the cost of risking overfitting which is slightly by the bagging concept. While the KNN is simply calculating the Euclidean distance from the points.

As shown in Figure 5 which assesses the performance of the RF, All the faults have Area Under the Curve (AUC) of 0.9 or higher. In the Figures the AUC is simply labelled as "area." The lowest AUC were for the classes 0,4 and 5 which are for Healthy and the Three-way valve number 2 leakages. The effect that those two faults have on the used features are minimal. This causes the lowering the AUC for the healthy operation. All the other faults have AUC of 0.97 up to 1.

As for the KNN algorithm assessment, it can be seen in Figure 6, the AUC ranges from 0.71 to 1. The same 3 classes have the lowest performance. In general, the performance of KNN is lower than the one of the RF but on the other hand the time to train the algorithm is much lower and far less likely to overfit the data.

Across both algorithms, three-way valve 1 leakage, two-way valve leakage and the sensors noise with deviation have the highest AUC due to their high effect on the temperature of the rooms. On the other hand, three-way valve 1 leakage has the lowest performance due to its unnoticeable effect on the features used in comparison with the healthy operation. This also leads to decrease in the AUC of the healthy operation.

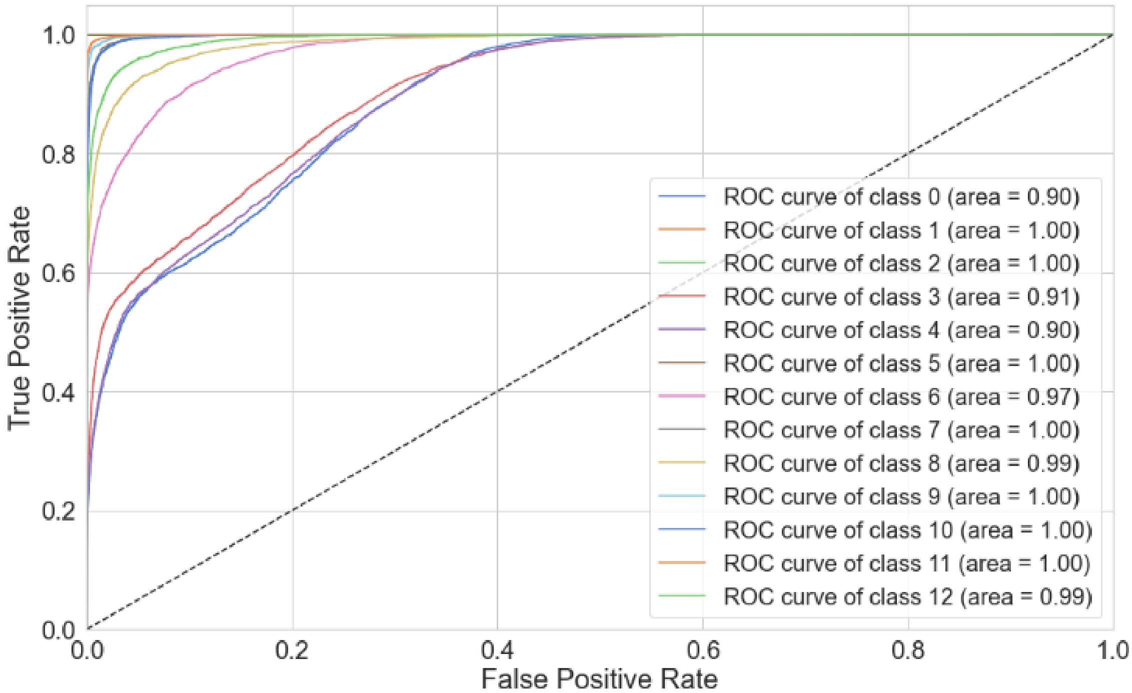

Figure 5. ROC curve for Random Forest

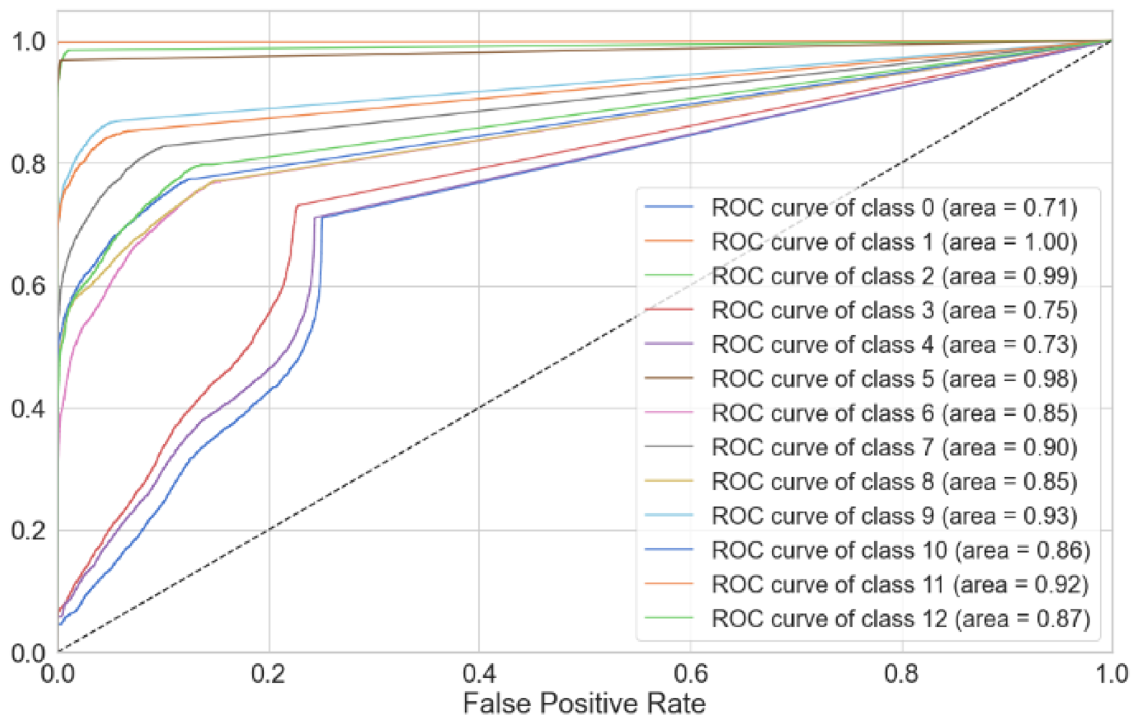

Figure 6. ROC for KNN

Table 3. AUC score for each fault

Class no.	Fault	RF-AUC	KNN_AUC
0	Healthy operation	0.9	0.71
1	Three-way valve 1 (1% leakage)	1.00	1.00
2	Three-way valve 1 (5% leakage)	1.00	0.99
3	Three-way valve 2 (1% leakage)	0.91	0.75
4	Three-way valve 2 (5% leakage)	0.90	0.73
5	Two-way valve leakage	1.00	0.98
6	Weather sensor constant noise	0.97	0.90
7	Weather sensor noise + deviation	1.00	0.85
8	Livingroom sensor constant noise	0.99	0.93
9	Livingroom sensor noise + deviation	1.00	0.85
10	Bedroom1 sensor constant noise	1.00	0.86
11	Bedroom1 sensor noise + deviation	1.00	0.92

12	Inadequate pump performance	0.99	0.87

Table 4. Evaluation methods comparison

Evaluation method	RF	KNN
Accuracy	0.97	0.94
Sensitivity	0.83	0.62
Specificity	0.98	0.96

5.2 Generalization over Palermo's weather

An attempt to use the model trained on Milano's weather to test the same faults for the weather of Palermo as mentioned above to test the generalization capabilities. The model performance decreased noticeably. However, five out of twelve faults retained acceptable performance (AUC higher than 75%) in the case of RF and two faults in case of KNN. Faults three and four (Three-way valve 2 leakage 1&5 %) still have the worst performance and they have the highest deterioration performance rate. The main conclusion is that both algorithms have to be re-trained when changing the weather files.

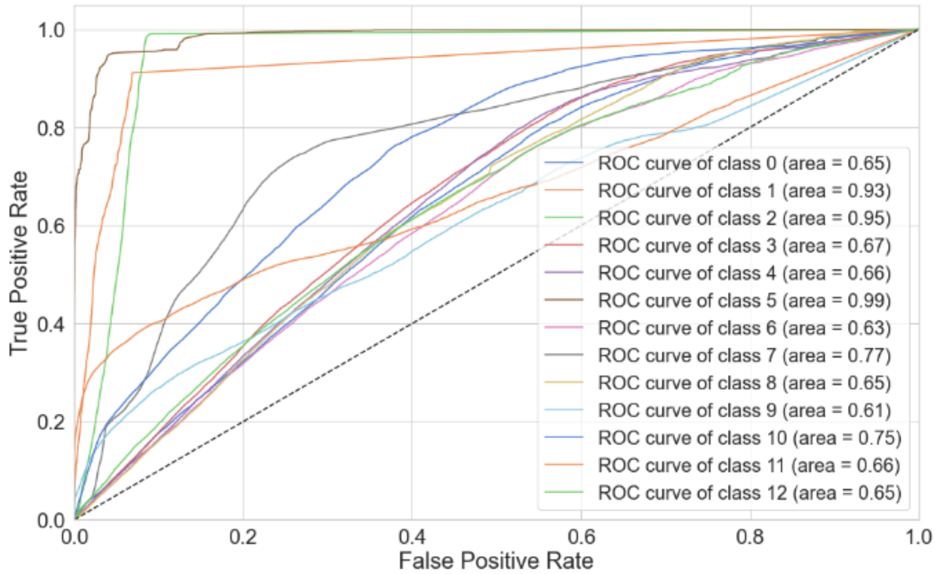

Figure 7. ROC curve for the tested algorithm on Palermo's weather for Random Forest

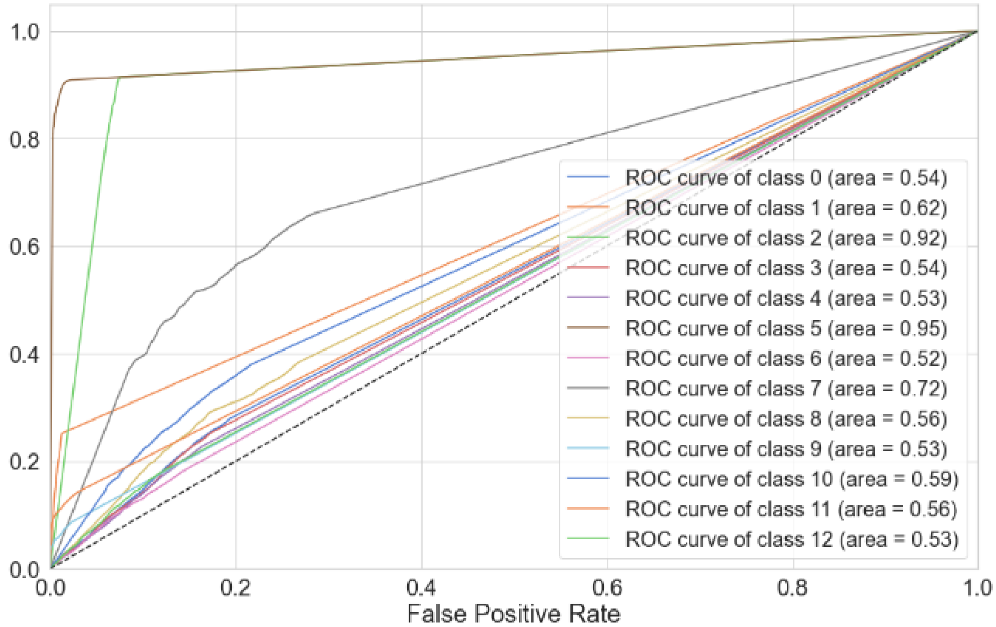

Figure 8. ROC curve for the tested algorithm on Palermo's weather for KNN

Despite the drop in performance, it seems that in the case of RF, the faults that have AUC of 1 can be generalized with AUC of 0.75 or higher in the new case. This might be promising in case of using more extensive list of features in the training phase as this will most likely lead to a higher AUC across all faults which will lead to a higher generalization capability. But this is not the case if minimal features are used

Table 5. Evaluation method for the generalization attempt.

Evaluation method	RF	KNN
Accuracy	0.886	0.886
Sensitivity	0.260	0.265
Specificity	0.938	0.938

6 Conclusions

This work was done to assess the performance of two supervised machine learning algorithms to detect and diagnose faults in the monitoring and hydronic system in a residential apartment using minimal features that are easily accessible and inexpensive to monitor. Random Forest and K-nearest neighbor were chosen according to the literature due to their low computational expense and their capabilities of capturing pattern in the time series introduced.

The case study was a two-bedroom apartment with an area of 81.6 m^2. The generation system used was an electric air-water heat pump connected to fan coils for cooling and radiant floor for heating as an emission system. A digital twin was used to simulate this system using Modelica language.

Both algorithms seemed to perform well across all faults, although Random Forest performed better with an accuracy = 0.97, sensitivity =0.83 and specificity = 0.98 while for the same matrices the scores for K- nearest neighbor were 0.94, 0.62 and 0.96, respectively.

For the individual faults, the RF had an average AUC of 0.95 while the KNN had 0.86. The second three-way valve leakage seems to have the lowest detection rate due to its low impact on the features used in the training.

To evaluate the generalization capabilities of the trained algorithms, they were used to detect the same faults in the same system but in weather of Palermo. The sensitivity dropped from 0.83 and 0.62 to 0.26 and 0.3265 for the RF and KNN, respectively.

The main downsides of the supervised methods are that they are highly dependent on the quality of the data provided. Both healthy and faulty data must exist to perform such process which rarely exist in real buildings. Highly accurate digital twin must be used to generate such data which takes time to be developed and tuned.

Future works will involve the usage of a combination of unsupervised algorithms to cluster the data and supervised algorithms to name those clusters with the hope of achieving continuous learning and achieve less dependency on the data from simulations.

7 Appendices

In this Appendix, the hyperparameters used for the two algorithms used are specified. First of all, Scikit-Learn library was used to apply RF and KNN [31].

For the RF, the following are the hyperparameters optimized as specified in the documentation [32]:

- The number of trees in the forest was 900, while the default was 100.
- Criterion used was “Gini”. The criterion is the function to measure the quality of a split. Supported criteria are “gini” for the Gini impurity and “log loss” and “entropy” both for the Shannon information gain. Note: this parameter is tree specific.
- Max_features: the number of features to consider when looking for the best split. “Sqrt” was used which means that the maximum number of features equal the square root of the number of features.
- Min_sample_split: the minimum number of samples required to split an internal node. This value was set to five. The default value is two.
- Bootstrap which determines whether bootstrap samples are used when building trees. If False, the whole dataset is used to build each tree. It was set to False.

For the KNN the parameters optimized were as follows [33]:

- Number of neighbours to use by default for neighbours’ queries. The value used was seven.
- Weight function used in prediction. “distance” was used which indicates that the weight points by the inverse of their distance. In this case, closer neighbours of a query point have a greater influence than neighbours which are further away.
- Leaf size: leaf size passed to BallTree or KDTree. This can affect the speed of the construction and query, as well as the memory required to store the tree. The optimal value depends on the nature of the problem. The used value is 50, while the default is 30.

8 Bibliography

[1] I. Hamilton and O. Rapf, "E2020 global status report for buildings and construction: towards a zero-emissions, efficient and resilient buildings and contruction sector," *Glob. Alliance Build. Constr.*, pp. 1–7, 2020.

[2] L. Pérez-Lombard, J. Ortiz, and C. Pout, "A review on buildings energy consumption information," *Energy Build.*, vol. 40, no. 3, pp. 394–398, 2008.

[3] M. R. Brambley and S. Katipamula, "Commercial Building Retuning A Low-Cost Way to Improve Energy Performance," *ASHRAE J.*, vol. 51, no. October, 2009.

[4] K. W. Roth, D. Westphalen, P. Llana, and M. Feng, "The Energy Impact of Faults in US Commercial Buildings," *Int. Refrig. Air Cond. Conf.*, pp. 600–609, 2004.

[5] G. Lin, H. Kramer, and J. Granderson, "Building fault detection and diagnostics: Achieved savings, and methods to evaluate algorithm performance," *Build. Environ.*, vol. 168, no. October 2019, p. 106505, 2020.

[6] W. Kim and S. Katipamula, "A review of fault detection and diagnostics methods for building systems," *Sci. Technol. Built Environ.*, vol. 24, no. 1, pp. 3–21, 2018.

[7] S. Katipamula and M. R. Brambley, "Review article: Methods for fault detection, diagnostics, and prognostics for building systems—a review, part II," *HVAC R Res.*, vol. 11, no. 2, pp. 169–187, 2005.

[8] S. Katipamula, M. R. Brambley, and M. R. Brambley, "Methods for Fault Detection , Diagnostics , and Prognostics for Building Systems — A Review , Part I," *HVAC R Res.*, vol. 11, no. 1, pp. 3–25, 2005.

[9] S. Frank, X. Jin, D. Studer, and A. Farthing, "Assessing barriers and research challenges for automated fault detection and diagnosis technology for small commercial buildings in the United States," *Renew. Sustain. Energy Rev.*, vol. 98, no. January, pp. 489–499, 2018.

[10] N. Omri, Z. Al Masry, N. Mairot, S. Giampiccolo, and N. Zerhouni, "Towards an adapted PHM approach: Data quality requirements methodology for fault detection applications," *Comput. Ind.*, vol. 127, p. 103414, 2021.

[11] Y. Li and Z. O'Neill, "A critical review of fault modeling of HVAC systems in buildings," *Build. Simul.*, vol. 11, no. 5, pp. 953–975, 2018.

[12] Z. Shi and W. O'Brien, "Development and implementation of automated fault detection and diagnostics for building systems: A review," *Autom. Constr.*, vol. 104, no. March, pp. 215–229, 2019.

[13] S. Pourarian, J. Wen, D. Veronica, A. Pertzborn, X. Zhou, and R. Liu, "A tool for evaluating fault detection and diagnostic methods for fan coil units," *Energy Build.*, vol. 136, pp. 151–160, 2017.

[14] R. Chiosa, M. S. Piscitelli, and A. Capozzoli, "A data analytics-based energy information system (EIS) tool to perform meter-level anomaly detection and diagnosis in buildings," *Energies*, vol. 14, no. 1, 2021.

[15] M. S. Piscitelli, S. Brandi, A. Capozzoli, and F. Xiao, "A data analytics-based tool for the detection and diagnosis of anomalous daily energy patterns in buildings," *Build. Simul.*, vol. 14, no. 1, pp. 131–147, 2021.

[16] A. P. Rogers, F. Guo, and B. P. Rasmussen, "A review of fault detection and diagnosis methods for residential air conditioning systems," *Build. Environ.*, vol. 161, no. July, p. 106236, 2019.

[17] P. E. Group, T. Downey, and S. P. Manager, "What Can 13 , 000 Air Conditioners Tell Us ? Draft Final Report," pp. 0–15.

[18] J. Proctor and T. Downey, "Transforming Routine Air Conditioner Maintenance Practices to Improve Equipment Efficiency and Performance," *Int. Energy Progr. Eval. Conf.*, 1999.

[19] A. P. Rogers and B. P. Rasmussen, "Opportunities for consumer-driven load shifting in commercial and industrial buildings," *Sustain. Energy, Grids Networks*, vol. 16, pp. 243–258, 2018.

[20] J. Gao and M. Bergés, "A large-scale evaluation of automated metadata inference approaches on sensors from air handling units," *Adv. Eng. Informatics*, vol. 37, no. February, pp. 14–30, 2018.

[21] T. Mulumba, A. Afshari, K. Yan, W. Shen, and L. K. Norford, "Robust model-based fault diagnosis for air handling units," *Energy Build.*, vol. 86, pp. 698–707, 2015.

[22] B. Wells, "Zodiac," *Notes Queries*, vol. s9-XI, no. 271, p. 188, 1903.

[23] A. Bhattacharya, J. Ploennigs, and D. Culler, "Short Paper: Analyzing Metadata Schemas for Buildings," pp. 33–34, 2015.

[24] B. Fan, Z. Du, X. Jin, X. Yang, and Y. Guo, "A hybrid FDD strategy for local system of AHU based on artificial neural network and wavelet analysis," *Build. Environ.*, vol. 45, no. 12, pp. 2698–2708, 2010.

[25] Y. Zhu, X. Jin, and Z. Du, "Fault diagnosis for sensors in air handling unit based on neural network pre-processed by wavelet and fractal," *Energy Build.*, vol. 44, no. 1, pp. 7–16, 2012.

[26] "Modelica Buildings library." .

[27] "GitHub - ibpsa/modelica-ibpsa: Modelica library for building and district energy systems developed within IBPSA Project 1." .

[28] D. B. Araya, K. Grolinger, H. F. ElYamany, M. A. M. Capretz, and G. Bitsuamlak, "An ensemble learning framework for anomaly detection in building energy consumption," *Energy Build.*, vol. 144, pp. 191–206, 2017.

[29] A. Baratloo, M. Hosseini, A. Negida, and G. El Ashal, "r h c iv o e f."

[30] Sarang Narkhede, "Understanding AUC - ROC Curve," *Towar. Data Sci.*, pp. 6–11, 2019.

[31] "scikit-learn: machine learning in Python — scikit-learn 1.1.1 documentation." .

[32] "sklearn.ensemble.RandomForestClassifier — scikit-learn 1.1.1 documentation." .

[33] "sklearn.neighbors.KNeighborsClassifier — scikit-learn 1.1.1 documentation." .