Energy Informatics

RESEARCH

Open Access

Detecting faults in the cooling systems by monitoring temperature and energy

Keshav Kaushik[1]* and Vinayak Naik[1,2]

*Correspondence:
p20180414@goa.bits-pilani.ac.in

[1] CSIS, BITS Pilani, Goa, Zuarinagar, India
[2] APPCAIR, BITS Pilani, Goa, Zuarinagar, India

Abstract

The cooling systems contribute to 40% of overall building energy consumption. Out of which, 40% is wasted because of faulty parts that cause anomalies in the cooling systems. We propose a three-stage, non-invasive part-level anomaly detection technique to identify anomalies in both cooling systems, a ducted-centralized and a ductless-split. We use COTS sensors to monitor temperature and energy without invading the cooling system. After identifying the anomalies, we find the cause of the anomaly. Based on the anomaly, the solution recommends a fix. If there is a technical fault, our proposed technique informs the technician regarding the faulty part, reducing the cost and time needed to repair it. In the first stage, we propose a domain-inspired time-series statistical technique to identify anomalies in cooling systems. We observe an *AUC-ROC* score of more than 0.93 in simulation and experimentation. In the second stage, we propose using a rule-based technique to identify the cause of the anomaly. We classify causes of anomalies into three classes. We observe an *AUC-ROC* score of 1. Based on the anomaly classification, we identify the faulty part of the cooling system in the third stage. We use the Nearest-Neighbour Density-Based Spatial Clustering of Applications with Noise (NN-DBSCAN) algorithm with transfer learning capabilities to train the model only once, where it learns the domain knowledge using the simulated data. The trained model is used in different environmental scenarios with both types of cooling systems. The proposed algorithm shows an *accuracy* score of 0.82 in simulation deployment and 0.88 in experimentation. In the simulation we used both ducted-centralized and ductless-split cooling systems and in the experimentation we evaluated the solution with ductless-split cooling systems. The overall *accuracy* of the three-stage technique is 0.82 and 0.86 in simulation and experimentation, respectively. We observe energy savings of up to 68% in simulation and 42% during experimentation, with a reduction of ten days in the cooling system's downtime and up to 75% in repair cost.

Keywords: Anomaly detection, Fault identification, Cooling systems, DBSCAN

Introduction

The cooling systems contribute to 40% of buildings' energy consumption (Vishwanath et al. 2017). They consume more energy when there are anomalous instances, and these instances occur due to faults in the cooling system. Faults in cooling systems lead to energy wastage of up to 40% of its overall lifetime energy consumption (Narayanaswamy et al. 2014). Anomalies and the cause behind these anomalies must be identified in real-time. Delay in detecting anomalies increases downtime and energy wastage (Rashid et al. 2019). Real-time identification of anomalies guides users to quickly take necessary action when a fault occurs and reduces the wastage of energy by cooling systems.

The cooling systems are categorized into two categories—ducted-centralized cooling systems and ductless-split cooling systems. In a ducted-centralized cooling system, a compressor is connected to many Air Handling Units (AHUs) using ducts. Here, the ducts transfer cold air from the compressor to the rooms. However, in the ductless split cooling systems, one compressor is connected to one AHU using copper pipes which transfer cold air from the compressor to the room. Both of the cooling systems come with an Energy Rating (ER) which represents the maximum cooling capacity of the cooling system. The user decides on a particular cooling system based on the requirements. Both types of cooling systems are prone to anomalies.

Real-time detection of anomalies in cooling systems is an important task, especially in critical systems where it is required to maintain the room's temperature throughout with minimum downtime. Detection of these anomalies comes under the scope of time-series anomaly detection as sensors produce time-series data. The two primary techniques to detect such anomalies are the classical approach and the Deep Learning (DL) approach. In the classical approach, we perform a time or frequency domain analysis of the time series signal to find anomalous instances and use distance metrics to identify anomalies. In the DL approach, we train a model to learn normal behavior. If the variables are not highly correlated, we consider them anomalies (Malki et al. 2022).

The techniques mentioned above are capable of identifying anomalies in time-series data. However, these techniques cannot explain the reason behind the anomalies. These techniques are data-sensitive and require a large amount of data. However, anomalies in real-world systems depend on various environmental and deployment factors. We do not have a large dataset at the initial stages of real-world deployment. Hence, these approaches are not suitable for such applications. For example, identifying an unusually high change in energy consumption by the cooling systems. There are various reasons behind this unusual change in energy consumption for example—technical, incorrect set temperature, or the AC has degraded over time and is not capable of cooling the room now. In this paper, we propose a novel NN-DBSCAN technique to identify the cause of anomalies. The proposed technique uses Transfer Learning, by virtue of which, it facilitates its deployment in new setups.

The current state of the art requires professional assistance for Fault Detection and Diagnosis (FDD) in cooling systems (Li and Braun 2009). They deployed a large set of sensors on each physical part of the cooling system to observe the changes. For example, Li and Braun (2009), proposed to detect faults using temperature sensors. These temperature sensors are deployed on the condensing unit, liquid line, suction line, and evaporating lines. Based on these data points, they proposed to identify faults. These techniques are accurate but require many sensors, which increases the cooling system's cost. The need for professional assistance makes these solutions not scalable to residential cooling systems.

Janetzko et al. proposed a novel unsupervised visual anomaly detection technique (Janetzko et al. 2014). They first understood the energy consumption pattern and then predicted the energy consumption during normal execution. If there was any deviation, they assigned it as an anomalous instance. The instance is represented as a coloring matrix where different intensities of colors are assigned based on the significance of deviation. Araya et al. proposed the use of ensemble learning techniques to identify the anomalies in energy systems (Araya et al. 2017). They proposed a collective contextual anomaly detection using sliding window (CCAD-SW). They used multiple anomaly classifiers and ensembled the results of those classifiers for the identification of anomalies. The solution is evaluated using HVAC data collected from a real-world building. These solutions are only capable of identifying anomalies and are not able to identify the cause of anomaly and the faulty part.

In this paper, we propose an IoT and Machine Learning-based solution to identify anomalies and the cause of those anomaly in a real-world system. We collect temporal data using energy and temperature sensors in real-time. Using the collected data, we identify anomalous instances observed during the execution of the cooling system. Figure 1 shows the deployment of an IoT sensor connected with the AHU of the cooling system. It records the overall energy consumption of the cooling system.

We propose three staged non-invasive part-level anomaly detection techniques to identify the fault and its cause in real-time. Here, we do not connect sensors to each part of the cooling system separately, making the proposed approach non-invasive. At first, we use statistical inference to find the anomalous instances of energy consumption during the execution of the cooling system. Here, we use domain-inspired statistical inference to identify a significant change in energy consumption concerning the past data. Then, we define a set of rules based on domain knowledge to

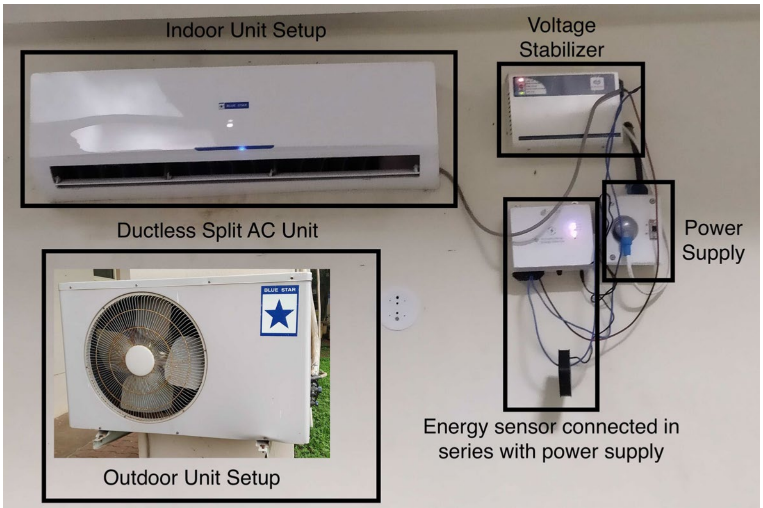


Fig. 1 An IoT-based sensor connected with AHU in series. It records the overall energy consumption of the cooling system. The sensor records the current consumption by the cooling system using the current transformer. The sensor is 4 G enabled, which provides us data anywhere, anytime around the globe, when required. We present the outdoor unit setup which forwards the cold air to Indoor Unit Setup which circulates the cold air in the room

identify the cause of anomalous energy consumption instances. Once the anomaly is identified from stage one, it is forwarded to the second stage. Finally, we identify the faulty part. We propose a non-invasive solution that only considers the overall energy consumption and environmental conditions of the cooling system and identifies the faulty part without explicitly connecting a sensor to each part of the cooling system. Our proposed solution is out-of-the-box, i.e., it does not require repeated training with every new real-world deployment. It learns the percentage impact of each cooling system part and identifies the faulty part. From our knowledge, these two features are a big step forward from state-of-the-art. The proposed solution is a plug-and-play solution where we just need to connect the energy sensor with the cooling system.

To the best of our knowledge, no one has considered finding faults and the faulty cooling system part in real-time. The proposed technique uses a set of rules that are constructed with the inference of domain knowledge to identify the faults. The following are the significant contributions of this paper. This paper is an extended version of the paper published in Energy Informatics. Academy Conference 2023 (Kaushik and Naik 2023).

1. We propose the use of domain-inspired statistical inference for the real-time identification of anomalies in real-world systems. Here, we use $O(n)$ space to identify faults, reducing memory overhead. Other state-of-the-art solutions require $O(n^2)$, they need long-term historical data for prediction (Sathe and Aggarwal 2016). Solutions with high space complexity need to perform more statistical operations, leading to an increase in computing time.
2. We propose a rule-based method to identify the cause of anomalies in cooling systems. These rules are deduced from the domain knowledge.
3. If the cause of the anomaly is classified as a technical anomaly, we help the technician by identifying the faulty part of the cooling system. We propose NN-DBSCAN with a transfer learning framework to classify the faulty part of the cooling system. The proposed technique requires only a single training. Once trained, it is used in any deployment, irrespective of cooling system deployment.
4. We evaluate our proposed solution in a simulation environment using EnergyPlus, where we deploy more than forty faults. We also evaluate it in an experimentation setup with six cooling systems. We observe an $AUC - ROC$ score of 0.95 in the simulation and 0.93 in the experimentation setup to identify anomalies. We observe an $AUC - ROC$ score of 1 in simulation and experimentation deployment for identifying the cause of anomaly using domain-inspired rules. Finally, depending on the anomaly's cause, we identify the faulty part of the cooling system. Here, we observe an *accuracy* score of 0.82 in the simulation and 0.88 in the experimentation. We observe energy savings of up to 68% and 42% in simulation and experimentation, respectively, with a reduction in downtime of the cooling system by ten days and a reduction in repair cost up to 75% reasoning early identification of faulty parts.

Background and problem statement

Cooling systems cool the enclosed area by removing heat and humidity. Using a chemical refrigerant, the cooling system transfers unwanted heat and moisture to the external environment. It consists of five major components—compressor, condenser, evaporator, expansion valve, and AHU. Figure 2 shows a basic architecture of the cooling system.

Compressor: The compressor of a cooling system changes the pressure on the refrigerant by increasing the temperature so that the refrigerant reaches a gaseous state. After reaching the gas state, the compressor stops working, and the gas starts cooling down.

Condenser: The condenser receives high-pressure gas from the compressor. It works on the principle of heat transfer, where heat is transferred from a hot substance to a cold substance. Here, the gaseous refrigerant is converted back to liquid refrigerant.

Evaporator: The refrigerant flowing in the evaporator tubes gets converted into vapors due to reduced pressure in evaporator tubes. This process makes the tube cooler and exchanges hot air from the enclosed environment.

AHU: It exchanges the cold air from the cooling system with the hot air from the room.The primary component of an AHU unit is the fan, which helps in the circulation of air in the room.

Problem statement To investigate whether faults have a discernible impact on temperature and energy consumption, we inject commonly observed faults (Li and O'Neill 2019) in each of the five components. Some of the commonly observed faults are refrigerant leakage, component failure, blockage of the expansion valve, compressor motor failure, etc. We plot the measured temperature and energy values in Fig. 3. We see an opportunity to differentiate the faults from the measured values. However, an appropriate clustering technique is needed to accurately identify the faults, which we address in this paper.

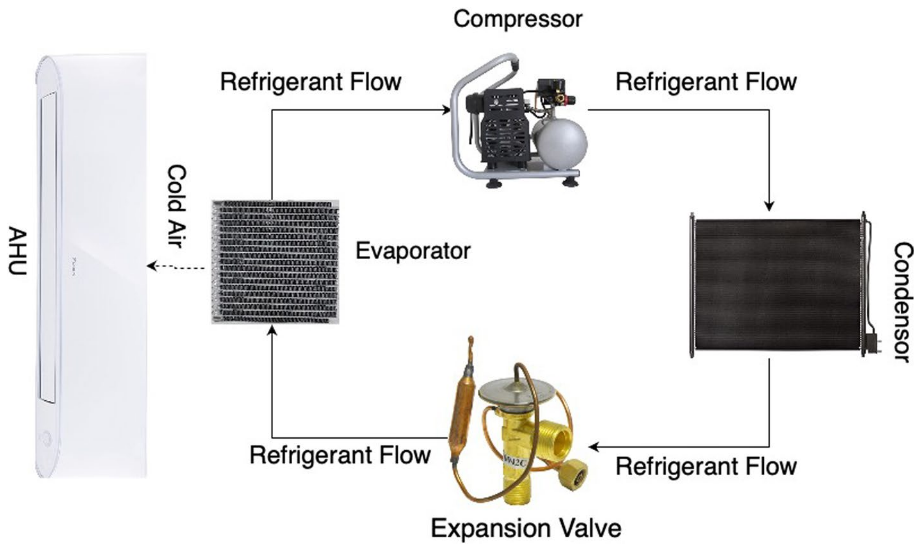


Fig. 2 The architecture of the cooling system consists of five major components—compressor, condenser, evaporator, expansion valve, and AHU. The compressor contributes the highest energy consumption, and the AHU the lowest

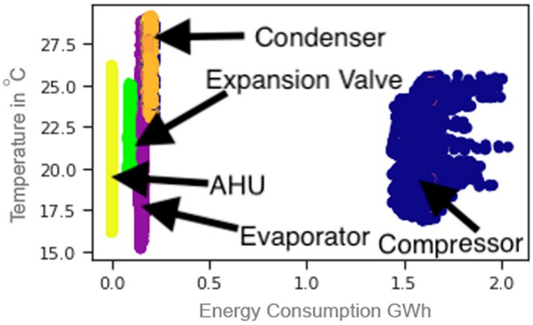


Fig. 3 (1) represents the energy consumption by the cooling system with different technical anomalies for each part. Here, (**a**) represents the expansion valve, (**b**) condenser, (**c**) compressor, (**d**) AHU, and (**e**) evaporator.Each faulty component has an impact on energy consumption and temperature. With an appropriate clustering technique, we can identify the faulty component

The proposed technique to identify an anomalous instance, its cause, and the faulty component responsible for the cause consists of three stages. In the first stage, we detect the anomaly in real-time using time-series patterns. In the second stage, we identify the cause behind the anomaly using domain-inspired rules. In the third stage, we identify the faulty component if a technical fault exists. Here are the definitions of the terms we use in our paper.

T_{set}	Set temperature of ductless-split cooling system ($°C$)
T_{room}	Present temperature of the room ($°C$)
T_{goal}	Desired final temperature of the room ($°C$)
$T_{external}$	External environmental temperature ($°C$)
τ	Change in temperature per unit time by ductless-split cooling systems ($°C$)
P	Energy consumption per hour by the ductless-split cooling systems ($W \cdot h^{-1}$)
ΔT	Change in the room temperature ($°C$)
Δt	Time interval between measurements (min)
PA	Past anomaly instance
ER	Energy Rating of the cooling system
P_{AP}	Energy consumption by the faulty part
AN	Anomaly cause
$Anomaly\ Cause_1$	Anomaly occurred due to wrong T_{set}
$Anomaly\ Cause_2$	Anomaly occurred due to technical fault
$Anomaly\ Cause_3$	Anomaly occurred due to cooling requirements not met

We measure $AUC - ROC$, $accuracy$, and F_1 scores for evaluating the solution.

$AUC - ROC$: It is used to measure the performance of the classification algorithms at various thresholds. It presents the capability of a classification algorithm to distinguish between classes. The AUC can be ranged from 0 to 1. Here, 1 represents the perfect classifier. The higher the value of AUC, the better the performance of the classification algorithm. The ROC curve is plotted between *True Positive Rate* (TPR) and *False Positive Rate* (FPR), and the area under the ROC curve is called AUC.

$$TPR = \frac{TP}{TP + FN} \tag{1}$$

Here, TP represents True Positives, and FN represents False Negatives. The equation represents the probability of the correct classification of positive instances.

$$FPR = \frac{FP}{FP + TN} \tag{2}$$

Here, FP represents false positives, and TN represents true negatives. The equation represents the probability of correctly classifying the instance as false.

We only use the $AUC - ROC$ score for stage 1 and 2 evaluations. The reason behind this is the data set. In stage 1, we have a data set from the sensors. Here, the anomalies are rare. In stage 2, we have three causes of anomalies, and the occurrence of each cause of the anomaly is different.

Accuracy: It measures the success of the prediction of classes. It tells how often our proposed technique correctly predicts all predictions made by the model. We use this metric to evaluate stage 3 because, in stage 3, we collect data from the simulation with the same number of occurrences of each fault.

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3}$$

The accuracy is calculated using the correct predictions and total predictions. Using this, we can check the degree of predictions made by the proposed solution conforms to the correct value.

F_1: It identifies the distribution of prediction. It is used to measure the performance of the classifications made. This is calculated by taking the harmonic mean of precision and recall. Evaluation using F_1 is important because it helps in indicating the performance of the classifier when there are uneven instances. For example, the occurrence of anomaly is rare; hence, using only accuracy is not sufficient in these cases. F_1 score is a more vital evaluation in these cases. Here, precision is calculated by $\frac{TP}{TP+FP}$ and recall using $\frac{TP}{TP+FN}$.

$$F_1\ score = 2 \times \frac{precision \times recall}{precision + recall} \tag{4}$$

Related work

The problem of anomaly detection is a focus of research. We classify the existing work into (a) ones focused on general techniques for any domain and (b) ones focused on a specific domain.

General techniques

Malki et al. (2022) proposed the use of ARIMA to predict future values of IoT data, and based on this data, they used the LightGBM model and Prophet to identify the faulty instances. The approach to identifying this type of anomaly is point anomaly detection. There, they did not identify the cause or faulty component of the IoT system.

Arjunan et al. (2015) proposed to identify anomalies using data from multiple users. Each user was a neighborhood because they had similar environments, thus having similar responses to the environmental factors affecting energy consumption. The neighborhood is decided based on prior knowledge. The proposed method did not detect anomalies while reducing the fault positive rate. Rashid et al. (2016) proposed a Collect Compare and Score framework to identify the anomalies. They collected data from smart meters and compared it with past data. If there was a significant difference between the two, they used local outlier factors to score the anomaly within the range of 0–1.

Narayanaswamy et al. proposed the Model Cluster and Compare framework in Narayanaswamy et al. (2014). They used unsupervised clustering to detect the anomalies automatically. The first step was to identify the abnormal instances, the second was to compare and perform clustering, and finally, they used intelligent rules for grouping the anomalies. The proposed technique could identify faults. However, it did not identify the faulty part of the system.

Rashid et al. (2019) proposed the UNUM rule-based technique on appliance-level energy consumption data to identify the behavior of the duty cycle of an appliance. They used k-means to identify the ON-OFF state of the system. Lastly, using a rule-based technique, they identified whether there existed a fault or not. If we use their anomaly detection technique, it identifies anomalies, but it does not identify the cause of anomalies and the faulty part of the cooling system.

Chiosa et al. an IoT-based method for Anomaly Detection and Diagnosis (ADD) (Chiosa et al. 2022). The proposed solution works on the energy meter level data to perform ADD. The proposed solution uses the advantages of both supervised and unsupervised learning algorithms. The proposed solution consists of six stages— (1) Data pre-processing, (2) Subsequence and context definition, (3) Group definition, (4) Contextual Matrix Profile (CMP) calculation, (5) Anomaly detection and the last step (6) is Anomaly Diagnosis. Step (1) includes the replacement of missing values from the dataset using linear interpolation, then in step (2) authors proposed to use Classification and Regression Tree (CART) to identify sub-sequences. Then in step (3) Hierarchical clustering is performed to divide the energy data into the groups, the data in these clusters are then used to generate the CMP matrix in step (4). Then, in step (5) the authors proposed to identify anomalous instances using the Euclidean distance. Finally, a severity score is calculated based on the computed CMP. The proposed solution only identifies the existence of anomalies. They do not focus on identifying the cause and faulty components.

Wang et al. proposed a secure federated learning-based framework and explored the forecasting capabilities of these for building data analysis (Wang et al. 2023). They proposed the use of Deep Learning models for data analysis. The architecture of the proposed solution contains seven components, namely (1) Federated server, (2) Federated client, (3) Data prepossessing, (4) Deep learning model, (5) Load forecasting, (6) Unsupervised learning model, and (7) Anomaly prediction. The working of proposed solution is categorized into two (1) Energy data platform—here the deep learning models are trained with the energy data and then the weights from deep learning models is forwarded to federated learning framework. In (2) the data is used for data analysis. Here they use Gaussian Mixture Model Clustering for predicting anomalies. The proposed solution focuses on identifying the instances of anomalies.

Domain-specific techniques

Seem (2007) proposed a data analysis method to identify anomalies in home energy consumption. Li and O'Neill (2019) proposed a probabilistic framework to rank the faults in the Heating Ventilation and Cooling (HVAC) system. They used occupant comfort and energy consumption data for this purpose. The proposed technique only focused on identifying the impact of the fault and not the faulty part. However, our proposed techniques identify the faulty part of the cooling system.

Zhao et al. (2017) proposed a Bayesian-based probabilistic technique to identify the faulty part inside AHU. Here, they used a set of twelve sensors to identify the exact faulty part. They also assigned a prior probability of occurrence of a particular fault. Our proposed technique uses only energy consumption data to identify the fault.

Rashid and Singh (2018) proposed to identify the patterns of past energy consumption data. Based on these patterns, they proposed to identify anomalies. There, they used only energy consumption data collected using smart sensors. They achieved an AUC score of 0.89 for chillers. However, our proposed architecture not only identifies anomalies but also finds the cause of the anomaly.

Fontugne et al. (2013) proposed the Strip Bind and Search (SBS) framework to identify the anomalies in buildings. They proposed to identify the pattern of the relationship between the devices and their usage patterns. If they observed a deviation in the relationship, they classified it as an anomaly. However, it did not identify the class of anomaly. It is important to identify the class of anomaly as it allows us to suggest the user take necessary action.

Methodology

We present the method's three-staged approach to identify anomalies and their causes. The proposed methodology first focuses on identifying anomalies in cooling systems. If it identifies the existence of an anomaly, the rule-based technique is used to classify the type of anomaly into three categories ranging from human error to technical fault. Based on these rules, we suggest an action to the user to make the execution cooling system more energy-efficient. Finally, the proposed methodology identifies the faulty parts of the cooling system, making them easier and faster to repair.

Identifying the anomalies

The cooling system's energy consumption follows a pattern. When the cooling system's compressor is ON, it consumes more energy. When it is OFF, it consumes less energy. This ON and OFF cycle continues for the entire duration of execution. The systems have an *ER* that specifies the maximum energy a cooling system can consume. This *ER* also represents the capacity of the cooling system.

Due to an anomaly generated by a fault or a misconfiguration, the energy consumption by the cooling system fluctuates. This leads to a change in the cooling cycle of the cooling system. Generally, a cooling system's average cooling cycle is 30 min, where the compressor is in the ON state for 20 min and in the OFF state for 10 min.

We propose to identify such changing patterns of the energy consumption of cooling systems due to the anomalies. In the first iteration, when we do not have any past data for the new deployment, we use the ER of the cooling system as a benchmark and compare it with current energy consumption data. For the subsequent cycles, we take the moving average of energy consumption by the cooling system in three cycles and compare the average value with data from the past non-anomalous cycle. We declare an anomaly if the difference between the two values is more than 5%.

The proposed anomaly detection Algorithm 1 does not require substantial storage space for identifying faults and anomalies compared to earlier proposed techniques discussed in Malki et al. (2022). The space complexity of the proposed algorithm is $O(n)$. n is the number of sensory values input to the anomaly detection algorithm. We want to minimize the memory requirement so that the solution works on a low-cost IoT device. In Sathe and Aggarwal (2016), the authors use an $n \times n$ matrix to identify the anomalous instances, which takes $O(n^2)$ space. The time complexity of our algorithm is also $O(n)$.

The time-series anomaly identification techniques in this subsection identify the existence of anomalies. However, only awareness about existence will not enable us to overcome it. In the next sub-section, we take a step forward to identify the cause of anomalies using a rule-based approach.

Algorithm 1 Real-time anomaly detection

```
Input: $Avg(P)$

Result: Instances of Anomaly
Initialization: $PD = 0$, An array-based queue $Q$ containing 45 slots to store
  energy consumption data of AC for the past 90 minutes of continuous
  execution.
  while $True$ do
      $M_A$ = Average of data in $Q$
      $PD$ = Calculate percentage difference of $M_A$ and $Avg(P)$
      if $PD \geq 5$ then
          Anomaly detected
      else
          $Avg(P) = M_A$
      end
      /*This loop will continue until the system is ON*/
end
```

Identifying a cause of the anomaly

We classify the reason behind any change in the cooling system's energy consumption into the following three categories.

- $Anomaly\ Cause_1$ – Wrong T_{set}
- $Anomaly\ Cause_2$ – Technical fault in the cooling system
- $Anomaly\ Cause_3$ – Cooling requirements are not satisfied

In Frank et al. (2018), the authors discussed important faults in cooling systems. Those are cooling system ON/OFF modes with setpoint schedules, oversized equipment design, air duct leakage, AHU motor degradation, compressor flow, condenser fan, inefficient evaporator airflow, and malfunctioning sensor. We consider all these and categorize them into three types based on the actions needed to fix them. In anomaly cause 1, we consider faults due to incorrect cooling system ON/OFF modes with setpoint schedules. $Anomaly\ Cause_2$ category consists of air duct leakage, AHU motor degradation, compressor flow, condenser fan, inefficient evaporator airflow, and sensor faults. $Anomaly\ Cause_3$ identifies the faults due to oversized and undersized equipment design.

In Table 1, we use data collected from the environment and the cooling system. When an anomaly exists, we identify its cause using these rules. These rules are based on domain knowledge. Using these rules, we conclude and suggest a fix. We measure these parameters every 2 min and use those values in the algorithm. The parameters are for both types of cooling systems.

Anomaly Cause$_1$

The T_{set} of the cooling system is not according to the cooling requirements, and the cooling system cannot reach the T_{goal}. The T_{room} is below or above the desired levels. In both scenarios, the energy consumption pattern will change. When the T_{set} is less than T_{goal}, the system will consume more energy to cool the room. When the T_{set} is more than T_{goal}, the system will go into an issue called short cycling. Both issues are identified by observing a change in energy consumption patterns.

To detect a wrong T_{set}, we first identify whether it can cool the room with the given cooling system. The cooling systems deployed in a room come with a cooling limit. We calculate

Table 1 Identifying a cause of the anomaly

Anomaly cause	Rule No.	Rule	Actions
$Anomaly\ Cause_1$	Rule 1	$T_{room} < T_{goal}$	T_{set} of AC is high—Change T_{set}
	Rule 2	$T_{goal} < T_{room}$	T_{set} of AC is low—Change T_{set}
$Anomaly\ Cause_2$	Rule 3	$\Delta T_{external} < \Delta T_{room}$	Cooling in T_{room} is less than $T_{external}$—Identify faulty part
	Rule 4	$\Delta T_{room} < 0$	Unable to cool—Identify faulty part
	Rule 5	$PA > 4$	More than 4 continuous anomalies—Identify faulty part
$Anomaly\ Cause_3$	Rule 6	$T_{room} < T_{set}$	No need to cool—Turn OFF cooling system
	Rule 7	$\Delta T_{room} \leq 0$ && $T_{set} = minimum$ && $T_{room} \leq T_{goal}$	Unable to reach the T_{goal}—Cooling system not sufficient

the cooling requirements using the room's heat load and decide on the maximum cooling capacity of the system to be deployed in the room. To identify whether it is possible to cool or not, we use τ calculated as follows:

$$\Delta T = T_{goal} - T_{room} \tag{5}$$

ΔT represents the change in the room temperature in a given duration of time. The ΔT considers all dynamic heat loads inside the room, which change with time.

$$\tau = \frac{\Delta T}{\Delta t(2\ minutes)} \quad (^{\circ}C \cdot min^{-1}) \tag{6}$$

τ represents the capacity of the cooling system. The cooling system's ability depends upon whether the system can bring the temperature down to the desired level and is checked by multiplying the τ by the time for which the system is executed. We consider the ideal continuous execution time of the cooling system ON cycle to be 20 min. Here, 20 represents the ideal cooling cycle time (Kaushik and Naik 2023).

$$\tau * 20 \tag{7}$$

Rule 1 and Rule 2 in Table 1 are to identify whether there exists an anomaly of the "$Anomaly\ Cause_1$" type. These rules compare the T_{room} with T_{goal} to identify whether the cooling system is cooling or not. These rules suggest the user take the necessary action to increase or decrease the T_{set}. We mention the rules in Table 1.

Anomaly Cause₂

Detecting technical faults in the cooling system requires an expert's opinion. A trained engineer who is an expert in the domain uses a set of multi-meters and sensors to identify faults. Detecting these faults automatically is a non-trivial task. To detect these faults automatically, we propose to use the domain knowledge from the literature to construct a set of rules that help us identify the existence of technical faults in an AC. If the faults are not repaired, the cooling system will continue to waste energy.

The Rule numbers 3, 4, and 5 are deduced from the domain knowledge. We consider the impact of the cooling system in the environment where it is deployed. Using Rule 3, we identify an impact of the cooling system's execution on the T_{room} and compare it with the change in $T_{external}$. If the cooling (reduction in temperature) is less than the change in $T_{external}$, the cooling system consumes abnormal energy.

In Rule 4, if we observe that no change in the T_{room} while the cooling system is being executed and consuming energy. We conclude that the cooling system cannot cool the room. This leads to disruption in the execution cycle of the cooling system. Rule 5 observes the anomalous behaviour of the cooling systems. If the system has continuously observed more than four anomalies with $Anomaly\ Cause_1$ classification, then it is a technical fault. If the user does not take action shown in Table 1, the cooling system will continue its execution with faults, thereby wasting energy.

Anomaly Cause$_3$

The deployment of the cooling system is based on the static factors affecting the room's heat load. However, the real-world environment is not static. There are dynamic factors that change the heat load of the room over a duration of time. For example, if a large number of people enter the room at any instance of time, the heat load of the room will be increased, and the AC may not be capable enough to maintain the T_{goal}. For example, if the user assumes that an AC with a particular tonnage will be sufficient based on room size.

Even if we deploy a cooling system with sufficient cooling capacity, its cooling capacity reduces over time. Hence, checking the cooling capacity of the cooling systems is required, especially in critical environments where maintaining cooling levels is needed. To check the cooling systems' capabilities to satisfy the cooling requirements, we use Rules 6 and 7.

Rule 6 checks whether the T_{room} is already less than the T_{set}. If this is the case, the cooling system will consume significantly less energy executing the fans of AHU. However, the compressor will be turned OFF. In this scenario, the cooling does not need to execute its cooling cycle. Due to the default functionality, the cooling system will try to turn ON the compressor and immediately turn it OFF. It causes the problem of short cycling.

Rule 7 identifies anomalous instances as $Anomaly\, cause_3$ when the change in T_{room} is less than equal to 0, which represents the cooling system, can cool, and the T_{set} of the cooling system is set to a minimum. However, the room's T_{goal} is not reached, leading to continuous execution of the cooling system compressor at total capacity. When the cooling system is executing at full capacity and cannot reach the T_{goal}, we conclude that the system cannot satisfy the cooling requirements.

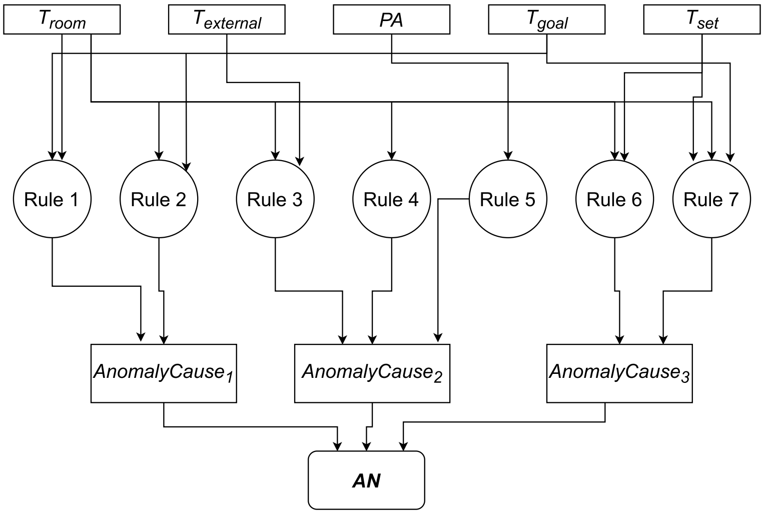


Fig. 4 We show Workflow for identifying the cause of the anomaly. We present the parameters used by each rule and the flow of data and decisions. Here, the rectangular box represents the data variable, the circular represents decision capabilities, and the rectangle with curve edges represents both decision and data variable

Using the proposed rule-based technique, the anomalous instance of the cooling system could be classified into two anomaly causes simultaneously.For example, if the cooling system cannot cool the room, the anomaly could be classified in two categories— $Anomaly\,Cause_1$ and $Anomaly\,Cause_2$. To identify the exact cause, we use the following.

Figure 4 presents the workflow of the proposed solution. This workflow shows the dependencies of each anomaly on the rules and input variables. From this rule-based causal tree, we get active high (1) or active low (0) values for each anomaly cause. When there is more than one active high anomaly, the particular type of anomaly is selected based on its priority. Here, the priority of $Anomaly\,Cause_1$ is the highest, and for $Anomaly\,Cause_3$ is the lowest. The priority is based on the chances of occurrence and the intensity of the anomaly.

With our proposed solution, we can identify anomalies and classify them into types. We suggest an action when the $Anomaly\,Cause_1$ and $Anomaly\,Cause_3$ are identified. However, when $Anomaly\,Cause_2$ is identified, in the next subsection, we propose to identify the faulty part of the cooling system.

Identifying the faulty part

We classify anomalies into $Anomaly\,Cause_1$, $Anomaly\,Cause_2$, and $Anomaly\,Cause_3$ classes using the discussed domain-inspired rules. When the anomaly is classified as $Anomaly\,Cause_1$ and $Anomaly\,Cause_3$, our technique suggests an action to the user. For example, if $Anomaly\,Cause_1$ is observed, it suggests to change the cooling system's T_{set}. If it is $Anomaly\,Cause_3$, it indicates that the cooling system is incapable of cooling or is not required. However, when the anomaly is classified as $Anomaly\,Cause_2$, we identify the part of the faulty AC to reduce the mean time to repair.

To identify the part of the cooling system with fault, we propose a Machine Learning technique, NN-DBSCAN. It is based on the principle of DBSCAN. We have k independent and identically distributed samples $J = j_1, j_2, \ldots, j_k$ drawn from distribution F over $\mathbb{R}^D$. Using two hyper-parameters of DBSCAN, we find a set of n clusters with high empirical density for samples in J (Schubert et al. 2017). These hyper-parameters are *eps* and *minPts*. The *eps* is the maximum possible distance between the two samples to be considered a neighborhood. The *minPts* is the minimum number of samples to be considered as a core point for a cluster. The DBSCAN is a clustering algorithm, not a classification algorithm, because it is an unsupervised approach. We use DBSCAN to classify the anomalies by taking the average of each cluster and comparing it with each cause's average from the data set. The cluster with the closest average is assigned to the particular cause.

To calculate the *eps* hyper-parameter, we use the Nearest Neighbour technique. We plot an elbow curve and select the value where the elbow occurs based on the distances obtained using the Nearest Neighbor. To calculate the second hyper-parameter *minPts*, the standard approach suggests selecting it to be twice the number of features. However, this approach does not always lead to an optimal value (Sefidian 2022). We use the gradient descent technique to get the optimal-*minPts* by executing the DBSCAN algorithm multiple times (Ramadan et al. 2022). The detailed formal algorithm is shown in 2.

Algorithm 2 Optimal-minPts

```
Input: Dataset
Result: Optimal value of minPts for DBSCAN
Initialization: df = dataframe with 20% of randomly picked points from the
 input dataset, A = [], F = 0, minpoints = 0, minpts˙low = 0, minpts˙high =
 0, and minPts = 1.
 F = Extract number of features from df
 minpoints = 2*F
 while minPts != 0 do
     minpts˙low = minpts-1
      minpts˙high = minpts+1
      Acc˙minpts˙low = Accuracy(DBSCAN(minpts˙low))
      Acc˙minpts˙high = Accuracy(DBSCAN(minpts˙high))
      Acc˙minpoints = Accuracy(DBSCAN(minpoints))
      /*Accuracy() is a function that calculates the accuracy, and DBSCAN()
      is a function to run DBSACN with given inputs*/i = len(A)
      if A[i] == A[i-1] then
          minPts = A[i]
           /*If the last two entries in array A are the same then it's the
           optimum number of minPts for DBSCAN*/
           return minPts
     else
          if Accuracy(DBSCAN(minpoints)) is maximum then
               minPts = minpoints
                break
          else
               if minpts˙high¿minpts˙low then
                    minpoints = $2 * minpoints$
                     /*Increase minpoints in multiple of two*/
                     break
               else
                    minpoints = $\frac{minpoints}{2}$
                     /*Decrease minpoints in fraction of two*/
                     break
               end
          end
          A.append(minPts)
           break
     end
end
```

The challenge with DBSCAN is that we need a large amount of data to obtain density-based clusters accurately. However, in the real world, the problem is that we do not have enough data to classify the data points into clusters using DBSCAN. To deal with this challenge, we propose NN-DBSCAN with transfer learning capabilities. NN-DBSCAN requires training, which is performed using data from the simulation, and it is done once. In NN-DBSCAN, we train the model with transfer learning capabilities to be used in other deployments with the same application. During training, we calculate the initial cluster value, the centroid of the clusters formed during training. Then, we use these initial cluster values as domain knowledge in different deployments, enabling transfer learning. We allocate data to the cluster in other deployments based on the Euclidean distance between the observed point from the data set and the initial cluster value. Once we have the *minPts* in each cluster, the algorithm continues as DBSCAN. The Algorithm 3 represents the proposed NN-DBSCAN.

To obtain cluster centers from the past data, we take the average energy consumption for each cluster, where each cluster represents a different anomalous part. We then calculate a fingerprint of a particular anomalous part concerning normal/usual energy consumption.

$$Anomalous\_Part = \frac{(ER + P_{AP})}{|ER|} \times 100 \tag{8}$$

The Eq. 8 gives us a percentage contribution when a part of the cooling system is faulty. This percentage is the same for all the cooling systems, as the principle working of each is the same. So, we use the same percentage of fingerprints obtained during training for the initial cluster center assignment.

$$incenter_i = \frac{Anomalous\_Part_i \times ER}{100} \tag{9}$$

Equation 9 computes the initial center for the cluster of a particular anomaly cause. Here, $i = 1, 2, 3, 4,$ and 5 represent different parts with the possible anomaly. The initial centers calculated using Eq. 9 are used as initial cluster points to enable transfer learning capabilities in the proposed solution. Each cooling system part is used for different functionalities with varying energy consumption. Further, we observe that the number of clusters formed with NN-DBSCAN equals the number of parts present in the cooling system.

To identify the faulty part, our proposed NN-DBSCAN algorithm assigns each faulty part to one of the incenters of clusters, representing each part of the cooling system, based on the Euclidean distance between the observed point and the incenter. Once the number of instances in each cluster exceeds the *minPts*, the algorithm assigns the cluster based on the density. The calculation of the incenter is based on the percentage of energy consumed by each part of the cooling system when we simulate the faults for one cooling system model.

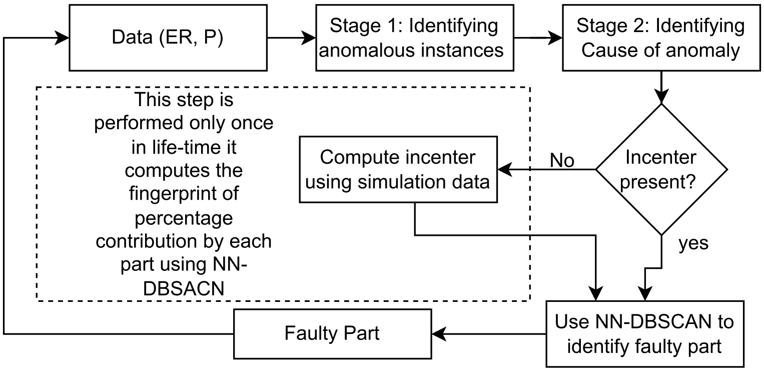

Data (ER, P)
Stage 1: Identifying anomalous instances
Stage 2: Identifying Cause of anomaly
Incenter present?
No
yes
Compute incenter using simulation data
This step is performed only once in life-time it computes the fingerprint of percentage contribution by each part using NN-DBSACN
Use NN-DBSCAN to identify faulty part
Faulty Part

Fig. 5 The proposed techniques takes *P* and *ER* of the cooling system as input. Using those, it identifies the existence of anomalies in stage 1. Stage 2 identifies the causes of these anomalous instances. If it is a technical anomaly, it identifies the faulty part using our proposed NN-DBSCAN. We train the model only once using simulated data. We transfer the model to any new cooling system using the latter's *ER*

Through transfer learning, we transform the learned model for other cooling systems. The transfer learning technique takes *ER* data from the datasheet as input and computes values for the incenters for the new model. Our use of transfer learning for NN-DBSCAN enables zero training requirements when deployed to the new cooling systems. Figure 5 shows a flowchart of the proposed three-stage solution. The three-staged proposed solution takes data points as input, which are used to identify anomalies in stage 1. Once anomalies are confirmed, stage 2 uses a rule-based approach to identify the anomaly's cause. If the anomaly is classified as a technical anomaly, then the proposed solution identifies the faulty part using NN-DBSCAN.

Algorithm 3 NN-DBSCAN

```
Input: $Dataset$, $pretraining$, $P$, $ER$, $Ncluster$, $incenter[]$, $minPts$, $eps$

Result: FaultyPart
Initialization: $Anomalous\_Part[]$, FaultyPart, Dist[].
 if $pretraining == 0$ then
    /*When the $incenter[]$ is empty*/
     $Accuracy = Accuracy$(DBSCAN(dataset with 50% test and train))
     $Nclusters$ = number of clusters formed with DBSCAN() for $i \leftarrow 0$ to
     $Ncluster$ do
        $Anomalous\_Part$.append(average value $P$ for each cluster)
         /*estimating fingerprint of each part*/
         increment i
    end
    $pretraining = 1$
     for $i \leftarrow 0$ to $Ncluster$ do
        incenter.append(compute using Equation 9)
         /*Computing energy fingerprint of each part for respective cooling
         system*/
         increment i
    end
else
    if $Points\ in\ each\ cluster\ is\ \leq\ minPts$ then
        Dist = EuclideanDistance(P to incenter)
         /*Calculate the Euclidean distance from each point to the incenter*/
         FaultyPart = min(EuclideanDistance(Pto incenter))
         /*Assign the point to the cluster with minimum Euclidean distance*/
    else
        Run DBSCAN($minPts$,$eps$,P)
         FaultyPart = P˙C($incenter[]$, P)
         /*Function P˙C finds the observed anomalous instance P is assigned
         to which cluster*/
    end
    return FaultyPart
end
```

Complexity analysis Stage 1 of the proposed solution uses a moving average, which takes $O(n)$ time. In stage 2, we use the rule-based technique that requires at most eight comparisons for every data point, which leads to a time complexity of $O(1)$. In stage 3, our proposed NN-DBSCAN has $O(5 * minPts)$ complexity, where 5 is the number of clusters. The number of $minPts$ is always less than n. Hence, the overall complexity of the proposed technique is $O(n)$. Here, n is the number of sensory values input to the anomaly detection algorithm.

Evaluation setup

This section describes the experimentation setup, simulation setup, and metrics we use to evaluate our proposed technique. We evaluate the proposed technique in an experimentation environment by deploying a set of energy sensors and environmental sensors in the environment. These sensors provide the energy consumption by the cooling system and room temperature to a data server at intervals of every two minutes.

The temperature data collected using environmental sensors helps us obtain a thermal model of the enclosed room. We deploy multiple environmental sensors in the room for each cooling system. A complete thermal modeling of an environment is impossible if we only collect the room's temperature data. The external temperature also affects the thermal model. Hence, we deploy environmental sensors to collect temperature data from the external environment.

We evaluate our proposed solution in two different experiment scenarios. In the first deployment, the heat load consists of one server rack, one switching rack, forty PCs with Intel i7 processors, three high-performance computers, and two tower servers. In the second deployment, we have three server racks and two network switching racks. Both the setups consist of six ductless-split cooling systems.

In the experimental environment, deploying many faults that can occur in a cooling system is challenging. To overcome this challenge, we perform simulations using the EnergyPlus simulator. Here, we consider more than forty faults and use the proposed technique to identify faults and their causes. These faults are consequences of faults in one of the five components of the cooling system. We encode these faults in the simulator.

We simulate a building environment with an area of $123m^2$. The building is an office environment. It comprises fifteen people, thirty personal computers, and a server rack. We assume each person can access two computers, one desktop and a laptop. The total energy consumption by all the heat sources is 40kWh. We use weather data to simulate an external environment, with the average external temperature of $26, 32, and\ 38^{\circ}C$. With T_{goal} set at $19^{\circ}C$.

To generate anomalous cases in simulation and experiments, we collect data using the EnergyPlus simulator. We inject faults in the cooling systems by changing the configuration and thresholds. The paper published by OSTI (Li and O'Neill 2019) details how to inject faults in the cooling systems using a simulator. For example, a reduction in refrigerant flow rate in the compressor leads to a faulty compressor. We use actual ACs with some faults we can instrument in the experimentation setup. The simulations consider all the possible faults.

We evaluate the proposed three-step solution in simulation and experimentation environments. In both simulation and experimentation environments, we have considered buildings with similar dimensions. The external temperature in experimentation is uncontrolled. However, in simulations, we evaluated the solution in six different environments where the external temperature ranges from $24\ to\ 42^{\circ}C$. We train the proposed NN-DBSCAN using simulation data to identify the faulty part. During this training, it learns the domain knowledge. Due to its capability of transfer learning, we do not need to train the model again while testing on different deployments.

Learning of the model

We use the data from the simulation with a ductless-split cooling system at an average $T_{external}$ of $38^{\circ}C$. We opt for the ductless-split cooling system because the principle working of both ductless-split and ducted-centralized cooling systems are the same, and the energy consumption ratio by each component is also the same. We split the data set equally into training and testing. We use a 50:50 split of the dataset for training with all types of anomalies so that it can learn the fingerprint of each part of the cooling system. We have an equal number of instances in the dataset for each anomalous part. Any other split configuration leads to missing cases of a particular type of anomaly. Hence, the technique is unable to learn complete domain knowledge. To test the generalizability of the proposed approach, we perform testing with different datasets where no training is performed.

Results

Simulation

Data collection and preparation

We deploy a ducted centralized cooling system and a ductless-split cooling system in a simulated building environment. In these cooling systems, we inject faults by changing the configuration and capacity of parts. More than forty faults occur based on the changes in the configuration of the five major AC parts. We collect simulation data in three environments, where $T_{external}$ ranges from $24^{\circ}C$ to $42^{\circ}C$. We manually labeled the data set's fault cause and the faulty component. The data is collected at intervals of two minutes for one year using EnergyPlus. The simulation dataset consist of P, T_{room}, $T_{external}$, AN, and FaultyPart. The training data for our proposed technique consists of 21,600 data points, out of which there are 12,600 data points representing more than forty types of anomalies. To test our proposed technique, we use a simulated data set with 1,576,800 data points for anomalies in six different scenarios with a ductless-split cooling system and ducted-centralized cooling system in three different external environments where the $T_{external}$ ranges from $24^{\circ}C$ to $42^{\circ}C$.

Identifying anomalies

For this stage, we do not need to train the proposed approach before use. It is a statistical method that only needs energy consumption by the cooling system without anomaly as apriori. We get this apriori energy consumption from past data or the data sheet of the cooling system. Here, we observe an $AUC - ROC$ score of 0.95.

Identifying cause of the anomaly

At this stage, no training is required. The proposed approach classifies the anomalies based on rules. In stage 2, we observe the $AUC - ROC$ score of 1.

Identifying the faulty component

This stage requires training. Here, we first train the proposed NN-DBSCAN and then evaluate it with the help of test data. We compare the results obtained from the proposed NN-DBSCAN with other state-of-the-art techniques. These techniques are Neural Network (NN), XGBoost, CatBoost, DBSCAN, and LightGBM.

We construct a NN with one node at the input layer and two hidden layers, each with ten nodes. We opt for the softmax activation function and auto loss function(Zhao et al. 2021). We use XGBoost with multi:softprob loss function, CatBoost with MultiClass loss function, DBSCAN with same *eps* and *minPts* as NN-DBSCAN here, we assign causes to each cluster manually, and LightGBM with multi_logloss as loss function (Chen and Guestrin 2016; Dorogush et al. 2018; Schubert et al. 2017; Ke et al. 2017). We compare these state-of-the-art techniques' *accuracy* score and F_1 score with our proposed technique that uses NN-DBSCAN.

Figure 6 compares the proposed NN-DBSCAN with other state-of-the-art classification approaches. Here, we observe that our proposed NN-DBSCAN outperforms compared to CatBoost, NN, XGBoost, and LightGBM. It performs as well as the standard DBSCAN. However, NN-DBSCAN stores relevant domain information that can be used in the future for transfer learning when deployed with another cooling system. We observe an *accuracy* score of 0.82 and F_1 score of 0.79 with NN-DBSCAN.

We compare these classification techniques in a new simulation deployment with a ducted-centralized cooling system instead of a ductless-split cooling system. We do not perform any training. Our proposed NN-DBSCAN results in an *accuracy* score of 0.8 and F_1 score of 0.76. The closest state-of-the-art solution is DBSCAN, with an *accuracy* score of 0.67 and F_1 score of 0.64, and NN performed worst with an *accuracy* score of 0.28 and F_1 score of 0.42.

The NN applies mathematical operations and combinations to the dataset's features. In our case, we use a single feature, P, for anomalous part identification, due to which the NN cannot learn the pattern of the associated anomalous part. XGBoost, CatBoost, and LightGBM are tree-based classification techniques. These tree-based techniques are highly prone to slight data variations. CatBoost performed better than the others. In the density-based approach, DBSCAN and NN-DBSCAN perform better because of their ability to differentiate between clusters with high and low density.

From the above-discussed results, we observe that the *accuracy* of the proposed model and other state-of-the-art reaches up to 82%. This is because the energy fingerprint of the three cooling system components is similar. These three components are the condenser, evaporator, and expansion valve. The condenser and evaporator are coils; they both have identical energy footprints. The working of the expansion valve is to control the flow to the evaporator, the energy footprint of the expansion valve is low, and due to a fault in the expansion valve, the evaporator does not perform its expected function. Hence, the footprint of the evaporator is added to the energy footprint of the expansion valve. These reasons make it difficult for the ML model to distinguish between the

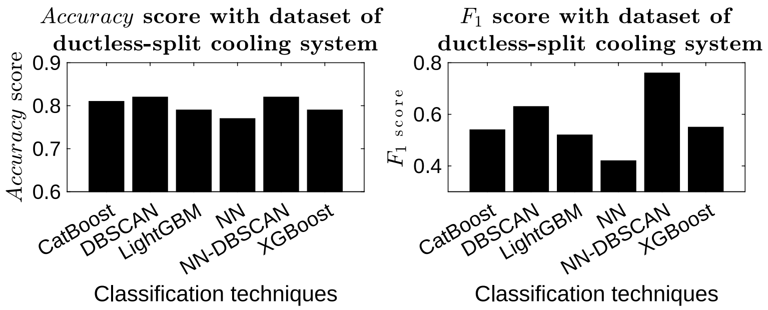


Fig. 6 A comparison of proposed NN-DBSCAN with the standard techniques used for classification in time-series application. Here, NN-DBSCAN performs similarly to DBSCAN and better than all other standard state-of-the-art techniques in terms of *accuracy* and F_1 score. Our proposed solutions show a higher F_1 score, meaning it can correctly classify the distributions. This is possible due to the initial cluster assignment in our approach. The other state-of-the-art approaches are not capable of doing so, leading to misclassification when there is less data available

three with high *accuracy*. Combining these three classes, we obtain an *accuracy* score of 1 with the proposed NN-DBSCAN.

Figure 7 depicts the performance of the proposed three staged techniques with an average *accuracy* score of 0.82 and an average F_1 score of 0.73 in six unique simulated deployments. In the six deployment scenarios, we consider both ductless-split and ducted-centralized cooling systems with external temperatures of $26°C$, $32°C$, and $38°C$. Figure 7 shows that our proposed approach works with different cooling systems in different environments after a single training. With our proposed solution, we observe a confidence interval of 0.82 ± 0.02 for *accuracy* with a confidence level of 95%. This represents that the proposed technique will classify with an accuracy of 0.80 to 0.82 for 95% of times.

Efficacy of the proposed interventions

We further evaluate our proposed technique concerning energy savings when the user takes the suggested action. Here, we observe the maximum energy savings of up to 68% with a mean energy savings of 34%. Early identification of faulty parts leads to a reduction in repair and downtime of the cooling systems. It also saves the repair cost because the early identification prevents the complete damage of the faulty part.

Experimentation

Data collection and preparation

We collect energy and temperature data using sensors in a data server deployed at a remote location. We use industry-grade sensors; we tested the accuracy of their measurements using a multimeter. The sensor self-calibrates itself every two hours. The data is collected using Representational State Transfer (REST) APIs at an interval of two minutes. The data set features are T_{room}, $T_{external}$, P, and timestamp. The user provides the T_{set} and T_{goal}. We prepossess this time-series data. The data has been collected for more than two and a half years. We use data from the complaint management system (CMS) as the ground truth for an anomaly shown in Fig. 8. The CMS contains information related to observed anomalies by the occupants. This CMS data includes the date

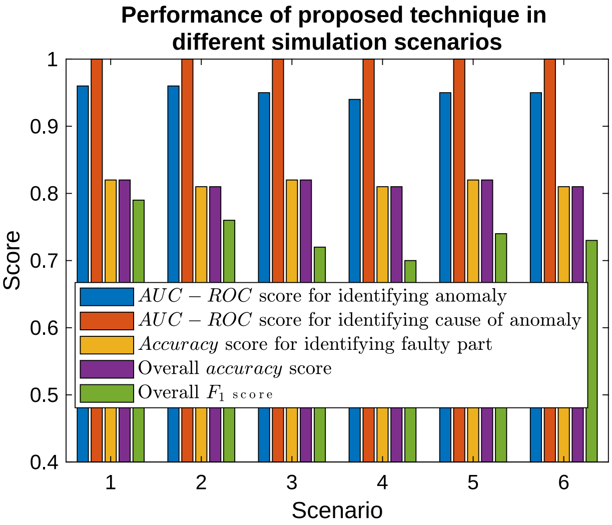
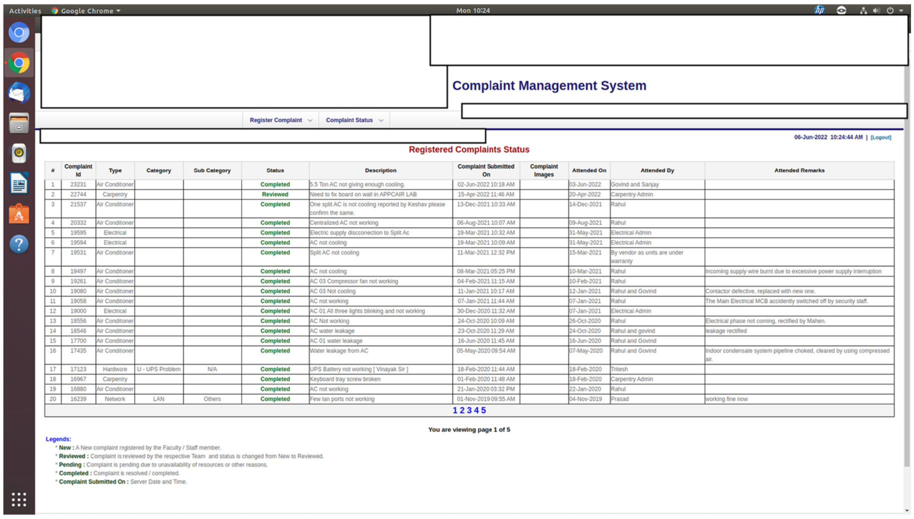


Fig. 7 We present the performance of the proposed NN-DBSCAN in three different environments. With combinations of the two types of cooling systems. There are six scenarios on the x-axis: 1 represents the scenario with a ductless-split cooling system with an average external temperature of 26°*C* here, we observe that the *AUC* — *ROC* score for stage 1 is 0.97, for stage 2 is 1 and accuracy score for stage 3 is 0.83. The overall *accuracy* and F_1 scores are 0.83 and 0.78, respectively. Scenario 2 represents the scenario with a ducted-centralized cooling system with an average external temperature of 26°*C* here, *AUC* — *ROC* score for stage 1 is 0.95, for stage 2 is 1 and accuracy score for stage 3 is 0.81. The overall *accuracy* and F_1 scores are 0.81 and 0.76, respectively. Scenario 3 represents the scenario with a ductless-split cooling system with an average external temperature of 32°*C* here we observe *AUC* — *ROC* score for stage 1 is 0.93, for stage 2 is 1 and the accuracy score for stage 3 is 0.82. The overall *accuracy* and F_1 scores are 0.82 and 0.72, respectively. Scenario 4 represents the scenario with a ducted-centralized cooling system with an average external temperature of 32°*C* here, *AUC* — *ROC* score for stage 1 is 0.95, for stage 2 is 1, and the accuracy score for stage 3 is 0.81. The overall *accuracy* and F_1 scores are 0.81 and 0.71, respectively. Scenario 5 represents the scenario with a ductless-split cooling system with an average external temperature of 38°*C* here, *AUC* — *ROC* score for stage 1 is 0.97, for stage 2 is 1, and the accuracy score for stage 3 is 0.83. The overall *accuracy* and F_1 scores are 0.83 and 0.75, respectively. Scenario 6 represents the scenario with a ducted-centralized cooling system with an average external temperature of 38°*C* here, *AUC* — *ROC* score for stage 1 is 0.96, for stage 2 is 1, and the accuracy score for stage 3 is 0.82. The overall *accuracy* and F_1 scores are 0.82 and 0.73, respectively



Complaint Management System

06-Jun-2022 10:24:44 AM |

Registered Complaints Status

#	Complaint Id	Type	Category	Sub Category	Status	Description	Complaint Submitted On	Complaint Images	Attended On	Attended By	Attended Remarks
1	23231	Air Conditioner			Completed	5.5 Ton AC not giving enough cooling.	02-Jun-2022 10:18 AM		03-Jun-2022	Govind and Sanjay	
2	22744	Carpentry			Reviewed	Need to fix board on wall in APPCAIR LAB	15-Apr-2022 11:46 AM		20-Apr-2022	Carpentry Admin	
3	21537	Air Conditioner			Completed	One split AC is not cooling reported by Keshav please confirm the same.	13-Dec-2021 10:33 AM		14-Dec-2021	Rahul	
4	20332	Air Conditioner			Completed	Centralized AC not working	06-Aug-2021 10:07 AM		09-Aug-2021	Rahul	
5	19595	Electrical			Completed	Electric supply discconection to Split Ac	19-Mar-2021 10:32 AM		31-May-2021	Electrical Admin	
6	19594	Electrical			Completed	AC not cooling	19-Mar-2021 10:09 AM		31-May-2021	Electrical Admin	
7	19531	Air Conditioner			Completed	Split AC not cooling	11-Mar-2021 12:32 PM		15-Mar-2021	By vendor as units are under warranty	
8	19497	Air Conditioner			Completed	AC not cooling	08-Mar-2021 05:25 PM		10-Mar-2021	Rahul	Incoming supply wire burnt due to excessive power supply interruption
9	19261	Air Conditioner			Completed	AC 03 Compressor fan not working	04-Feb-2021 11:15 AM		10-Feb-2021	Rahul	
10	19080	Air Conditioner			Completed	AC 03 Not cooling	11-Jan-2021 10:17 AM		12-Jan-2021	Rahul and Govind	Contactor defective, replaced with new one.
11	19058	Air Conditioner			Completed	AC not working	07-Jan-2021 11:44 AM		07-Jan-2021	Rahul	The Main Electrical MCB accidently switched off by security staff.
12	19000	Electrical			Completed	AC 01 All three lights blinking and not working	30-Dec-2020 11:32 AM		07-Jan-2021	Electrical Admin	
13	18556	Air Conditioner			Completed	AC Not working	24-Oct-2020 10:09 AM		26-Oct-2020	Rahul	Electrical phase not coming, rectified by Mahen.
14	18546	Air Conditioner			Completed	AC water leakage	23-Oct-2020 11:29 AM		24-Oct-2020	Rahul and govind	leakage rectified
15	17700	Air Conditioner			Completed	AC 01 water leakage	16-Jun-2020 11:45 AM		16-Jun-2020	Rahul and Govind	
16	17435	Air Conditioner			Completed	Water leakage from AC	05-May-2020 09:54 AM		07-May-2020	Rahul and Govind	Indoor condensate system pipeline choked, cleared by using compressed air.
17	17123	Hardware	U - UPS Problem	N/A	Completed	UPS Battery not working [Vinayak Sir]	18-Feb-2020 11:44 AM		18-Feb-2020	Tritesh	
18	16967	Carpentry			Completed	Keyboard tray screw broken	01-Feb-2020 11:48 AM		18-Feb-2020	Carpentry Admin	
19	16880	Air Conditioner			Completed	AC not working	21-Jan-2020 03:32 PM		22-Jan-2020	Rahul	
20	16239	Network	LAN	Others	Completed	Few lan ports not working	01-Nov-2019 09:55 AM		04-Nov-2019	Prasad	working fine now

1 2 3 4 5

You are viewing page 1 of 5

Legends:

- **New :** A New complaint registered by the Faculty / Staff member.
- **Reviewed :** Complaint is reviewed by the respective Team and status is changed from New to Reviewed.
- **Pending :** Complaint is pending due to unavailability of resources or other reasons.
- **Completed :** Complaint is resolved / completed.
- **Complaint Submitted On :** Server Date and Time.

Fig. 8 The complaint management system with information related to the faults, cause of fault, and the downtime of the cooling system. We use the data from the CMS as ground truth, representing the faults reported by the technicians in the cooling systems. We compare the ground truth with the predictions made by our solution to evaluate its performance

and time when the anomaly was first identified, the timestamp of the complaint's resolution, and the technician's reason. These complaints are logged by an occupant or the room administrator based on changes in room temperature measured using a temperature sensor or high-temperature warning on servers. We collect this data from six different ductless-split cooling systems. The experimentation dataset contains 221,450 data points. Note that the model is pre-trained and is not connected to this data.

Identifying anomalies

We use this stage's proposed domain-inspired statistical technique to identify the anomaly. Here, no pre-training is required. We observe an $AUC - ROC$ score of 0.93 in an experimentation deployment.

Identifying cause of the anomaly

We observed a AUC score of 1 in the simulation with the rule-based technique. Similarly, in experimentation deployment, the AUC score obtained at this stage is 1.

Identifying the faulty component

We do not perform any model pre-training for experimentation deployment. We take the model which is already pre-trained with simulated data.

Here, our proposed NN-DBSCAN classifies the faulty components with an $AUC - ROC$ score of 0.88 and an F_1 score of 0.81, shown in Figure 9. The significant increase in *accuracy* compared to simulation is because the anomalies from the evaporator, condenser, and expansion valve are rare in real-world experimentation.

Figure 10 depicts the performance of the proposed technique and observations in five scenarios— 1. $AUC - ROC$ score of 0.93 for identifying anomalies, 2. $AUC - ROC$ score of 1 for identifying the anomaly's cause, 3. *accuracy* score of 0.88 with NN-DBSCAN for identifying faulty part, 4. an overall *accuracy* score of 0.86, and 5. overall F_1 score of 0.84. Here, the overall scores represent combining all three stages. The overall *accuracy* score of the proposed technique in experimentation deployment is 0.86. With our proposed solution, we observe a confidence interval of 0.86 ± 0.04 for *accuracy* with a confidence level of 95%. This represents that the proposed technique will classify with an accuracy of 0.82 to 0.86 for 95% of times.

Efficacy of the proposed interventions

During experimentation, our proposed solution identifies anomalous instances of the cooling systems. Here, we observe maximum energy savings of up to 42% and mean energy savings of 30% with a downtime reduction of more than ten days when the action is taken immediately after identifying a faulty component by our proposed technique. In this scenario, a fault exists in the cooling system's compressor. The faulty AC stopped working ten days after our proposed solution flagged the anomaly, and the technician repaired it in an hour. Reducing downtime is crucial in critical environments where it is required to maintain the T_{goal} throughout the day. Early identification of the faulty part using our proposed techniques saves up to 75% of repair costs as it prevents the complete damage of the faulty part. We observe a mean reduction of 60% repair cost of cooling systems.

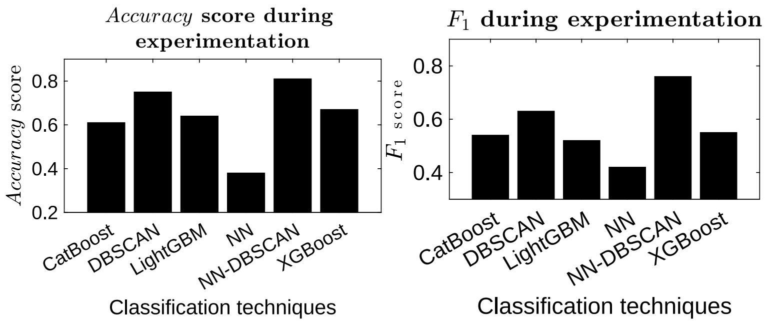


Fig. 9 *Accuracy* and F_1 score with the proposed approach during experimentation. We get an *accuracy* score of 0.88 and an F_1 score of 0.81. With NN-DBSCAN, we observe that F_1 and *accuracy* scores are the highest compared to other state-of-the-art techniques. A higher F_1 score presents the reliability of our solution. Our proposed technique outperforms the state-of-the-art

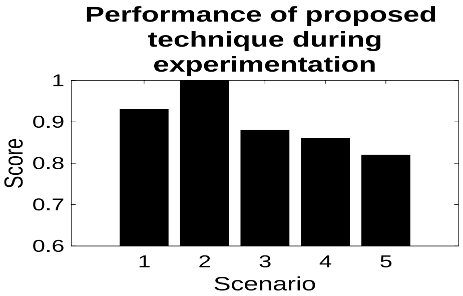


Fig. 10 On the x-axis, we represent the five scenarios—*AUC* − *ROC* for identifying anomalies, *AUC* − *ROC* for identifying the cause of anomalies, *accuracy* score for identifying faulty parts, overall *accuracy* score, and overall F_1 score. We observe *AUC* − *ROC* scores of 0.93 and 1 for identifying anomalies and identifying the cause of the anomaly, respectively. The *accuracy* score is 0.88 for identifying the faulty part of the cooling system. The overall *accuracy* of the proposed approach is 0.86, and the overall F_1 score is 0.82. Our solution's overall performance is considered excellent as the AUC-ROC score is always above 0.8 (Mandrekar 2010)

We observe that NN-DBSCAN takes 0.031 s. The overall inference latency for identifying anomalies in the cooling system using our proposed three-staged anomaly detection technique is 0.05 seconds.

Conclusion and discussion

This paper proposes a novel, non-invasive part-level anomaly detection technique. It is a three-stage approach. In the first stage, we identify whether an anomaly exists or not. In the second stage, we classify the type of anomaly, and in the third stage, we identify the faulty part of the cooling system. To identify anomalies, we propose a statistical method with domain inference to identify whether there exists an anomaly in the cooling system or not. We use the energy rating specified in the datasheet of the cooling system. At this stage, we observe an *AUC* score of up to 0.95 in simulation and 0.93 during experimentation.

Once the anomalous instance is identified, we identify the cause. Where we have a set of rules based on domain knowledge to identify the type of anomaly, this identification also represents the cause. If the anomaly type is "*Anomaly Cause*1" then the set temperature of the cooling system is wrong. We suggest the user increase or decrease the set temperature to overcome the anomaly. If it is "*Anomaly Cause*2", we inform the user that the anomaly cause is technical and identify the faulty part. Finally, if it is "*Anomaly Cause*3", we tell the user that either the cooling system is not capable enough to cool the room or you need a cooling system with lower capacity. We observe an $AUC - ROC$ score of 1 using the set of rules.

When an anomaly is classified as "*Anomaly Cause*2", we assist the technician by identifying the faulty part of the cooling system. We proposed NN-DBSCAN for this purpose. The proposed NN-DBSCAN is required to be trained only once in its lifetime. During pre-training, it obtains domain knowledge. In the case of cooling systems, it learns about the energy fingerprint of each part in the cooling system. Using this domain knowledge, our proposed techniques work with all cooling systems in any enclosed environment.

In the simulation, we observe a maximum *accuracy* score of 0.82. This is because the energy fingerprints of the condenser, evaporator, and expansion valve are similar because of their working principle. However, in the experimentation, we observe a maximum *accuracy* score of 0.88. The reason behind this significant increase is that during experimentation, the occurrence of a fault in the three discussed components is rare. The *accuracy* score for the proposed three-staged technique is 0.86 in experimentation and up to 0.82 in simulation, which is comparable to the *accuracy* of the state-of-the-art FDD methods that use a large number of sensors for the identification of faulty parts in cooling systems. The state-of-the-art which is an invasive method, where expert opinion is also required for the identification of the anomalies. These techniques claim to work with an *accuracy* of 80%, which is less than what our solution achieved.

We conduct case studies. We observe significant energy savings using the proposed techniques. In the simulation, we observe energy savings of up to 68% when the cooling system's compressor is faulty. During the experimentation with our proposed technique saves up to 42% of energy. We observe a reduction of downtime by ten days and a reduction in repair cost by up to 75% due to the early identification of the faulty part using our proposed technique.

Our proposed solution requires pre-training before deployment for detecting anomalies; hence, the classification of faulty parts depends on the fidelity of the pre-training data. The rule-based system proposed in this paper focuses on environmental factors not dependent on the hardware setup.

Funding
No funding source is available for this work.

Availability of data and materials
Part of the data used in this research is available at https://data.openei.org/submissions/5971 (Kaushik and Naik 2023).

Declaration

Ethics approval and consent to participate
This declaration is not applicable.

Consent for publication
This declaration is not applicable.

Competing interests
The authors declare that they have no Conflict of interest

Received: 19 March 2024 Accepted: 12 June 2024
Published online: 17 June 2024

References

Araya DB, Grolinger K, ElYamany HF, Capretz MAM, Bitsuamlak G (2017) An ensemble learning framework for anomaly detection in building energy consumption. Energy Build 144:191–206. https://doi.org/10.1016/j.enbuild.2017.02.058

Arjunan P, Khadilkar HD, Ganu T, Charbiwala ZM, Singh A, Singh P (2015) Multi-user energy consumption monitoring and anomaly detection with partial context information. In: Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments. BuildSys '15, pp. 35–44. Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2821650.2821662

Chen T, Guestrin C (2016) Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '16, pp. 785–794. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/2939672.2939785

Chiosa R, Piscitelli MS, Fan C, Capozzoli A (2022) Towards a self-tuned data analytics-based process for an automatic context-aware detection and diagnosis of anomalies in building energy consumption timeseries. Energy Build 270:112302. https://doi.org/10.1016/j.enbuild.2022.112302

Dorogush AV, Ershov V, Gulin A (2018) CatBoost: gradient boosting with categorical features support. arXiv. https://doi.org/10.48550/ARXIV.1810.11363. https://arxiv.org/abs/1810.11363

Fontugne R, Ortiz J, Tremblay N, Borgnat P, Flandrin P, Fukuda K, Culler D, Esaki H (2013) Strip, bind, and search: a method for identifying abnormal energy consumption in buildings. In: 2013 ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), pp. 129–140. https://doi.org/10.1145/2461381.2461399

Frank SM, Kim J, Cai J, Braun JE (2018) Common faults and their prioritization in small commercial buildings: February 2017–December 2017 https://doi.org/10.2172/1457127

Janetzko H, Stoffel F, Mittelstädt S, Keim DA (2014) Anomaly detection for visual analytics of power consumption data. Comput Graphics 38:27–37. https://doi.org/10.1016/j.cag.2013.10.006

Kaushik K, Naik V (2023a) A real-time non-invasive anomaly detection technique for cooling systems. In: Energy Informatics: Third Energy Informatics Academy Conference, EI.A 2023, Campinas, Brazil, December 6-8, 2023, Proceedings, Part I, pp. 135–151. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-031-48649-4_8

Kaushik K, Naik V (2023b) An energy consumption dataset for ductless-split cooling systems to train large models. In: Proceedings of the 10th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation. BuildSys '23, pp. 216–219. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3600100.3623721

Kaushik K, Naik V (2023c) Ocsrl: an model-based reinforcement learning approach to optimize energy consumption of cooling systems. In: 2023 IEEE 19th International Conference on e-Science (e-Science), pp. 1–10. https://doi.org/10.1109/e-Science58273.2023.10254663

Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu T-Y (2017) Lightgbm: a highly efficient gradient boosting decision tree. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS'17, pp. 3149–3157. Curran Associates Inc., Red Hook, NY, USA

Li H, Braun JE (2009) Development, evaluation, and demonstration of a virtual refrigerant charge sensor. HVAC &R Res 15(1):117–136. https://doi.org/10.1080/10789669.2009.10390828

Li Y, O'Neill Z (2019) An innovative fault impact analysis framework for enhancing building operations. Energy Buildings 199:311–331. https://doi.org/10.1016/j.enbuild.2019.07.011

Malki A, Atlam E-S, Gad I (2022) Machine learning approach of detecting anomalies and forecasting time-series of iot devices. Alex Eng J 61(11):8973–8986. https://doi.org/10.1016/j.aej.2022.02.038

Mandrekar JN (2010) Receiver operating characteristic curve in diagnostic test assessment. J Thorac Oncol 5(9):1315–1316. https://doi.org/10.1097/JTO.0b013e3181ec173d

Narayanaswamy B, Balaji B, Gupta R, Agarwal Y (2014) Data driven investigation of faults in hvac systems with model, cluster and compare (mcc). In: Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings. BuildSys '14, pp. 50–59. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/2674061.2674067

Ramadan HS, Maghawry HA, El-Eleamy M, El-Bahnasy K (2022) A heuristic novel approach for determination of optimal epsilon for dbscan clustering algorithm. J Theor Appl Inf Technol 100(1):2243–2253

Rashid H, Arjunan P, Singh P, Singh A (2016) Collect, compare, and score: a generic data-driven anomaly detection method for buildings. In: Proceedings of the Seventh International Conference on Future Energy Systems Poster Sessions. e-Energy '16. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/2939912.2942354

Rashid H, Singh P (2018) Monitor: an abnormality detection approach in buildings energy consumption. In: 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC), pp. 16–25. https://doi.org/10.1109/CIC.2018.00-44

Rashid H, Singh P, Stankovic V, Stankovic L (2019) Can non-intrusive load monitoring be used for identifying an appliance's anomalous behaviour? Appl Energy 238:796–805. https://doi.org/10.1016/j.apenergy.2019.01.061

Sathe S, Aggarwal C (2016) LODES: Local Density Meets Spectral Outlier Detection, pp. 171–179. https://doi.org/10.1137/1.9781611974348.20

Schubert E, Sander J, Ester M, Kriegel HP, Xu X (2017) Dbscan revisited, revisited: why and how you should (still) use dbscan. ACM Trans Database Syst. https://doi.org/10.1145/3068335

Seem JE (2007) Using intelligent data analysis to detect abnormal energy consumption in buildings. Energy Build 39(1):52–58. https://doi.org/10.1016/j.enbuild.2006.03.033

Sefidian AM (2022) How to Determine Epsilon and MinPts Parameters of DBSCAN Clustering. http://www.sefidian.com/2020/12/18/how-to-determine-epsilon-and-minpts-parameters-of-dbscan-clustering/

Vishwanath A, Chandan V, Mendoza C, Blake C (2017) A data driven pre-cooling framework for energy cost optimization in commercial buildings. In: Proceedings of the Eighth International Conference on Future Energy Systems. e-Energy '17, pp. 157–167. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3077839.3077847

Wang R, Yun H, Rayhana R, Bin J, Zhang C, Herrera OE, Liu Z, Mérida W (2023) An adaptive federated learning system for community building energy load forecasting and anomaly prediction. Energy Build 295:113215. https://doi.org/10.1016/j.enbuild.2023.113215

Zhao Y, Wen J, Xiao F, Yang X, Wang S (2017) Diagnostic Bayesian networks for diagnosing air handling units faults—part i: faults in dampers, fans, filters and sensors. Appl Therm Eng 111:1272–1286. https://doi.org/10.1016/j.applthermaleng.2015.09.121

Zhao X, Liu H, Fan W, Liu H, Tang J, Wang C (2021) Autoloss: automated loss function search in recommendations. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. KDD '21, pp. 3959–3967. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3447548.3467208

Publisher's Note