Building and Environment 183 (2020) 107212

Contents lists available at ScienceDirect

Building and Environment

journal homepage: www.elsevier.com/locate/buildenv

Anomaly detection based on machine learning in IoT-based vertical plant wall for indoor climate control

Yu Liu [a,*], Zhibo Pang [b], Magnus Karlsson [a], Shaofang Gong [a]

[a] Department of Science and Technology, Linköping University, Campus Norrköping, 60221 Norrköping, Sweden
[b] ABB AB, Corporate Research, Forskargränd 7, 72178, Västerås, Västmanland, Sweden

ARTICLE INFO

Keywords:
Vertical plant wall
Indoor climate control
Anomaly detection
Internet of Things
Machine learning
Neural networks

ABSTRACT

Indoor climate is closely related to human health, comfort and productivity. Vertical plant wall systems, embedded with sensors and actuators, have become a promising application for indoor climate control. In this study, we explore the possibility of applying machine learning based anomaly detection methods to vertical plant wall systems so as to enhance the automation and improve the intelligence to realize predictive maintenance of the indoor climate. Two categories of anomalies, namely point anomalies and contextual anomalies are researched. Prediction-based and pattern recognition-based methods are investigated and applied to indoor climate anomaly detection. The results show that neural network-based models, specifically the autoencoder (AE) and the long short-term memory encoder decoder (LSTM-ED) model surpass the others in terms of detecting point anomalies and contextual anomalies, respectively, therefore can be deployed into vertical plant walls systems in industrial practice. Based on the results, a new data cleaning method is proposed and a prediction-based method is deployed to the cloud in practice as a proof-of-concept. This study showcases the advancements in machine learning and Internet of things can be fully utilized by researches on building environment to accelerate the solution development.

1. Introduction

The indoor climate of a modern building, i.e., the air temperature, relative humidity, poisonous gas concentrations, volatile organic compounds level, is closely related to human health, comfort, and work productivity. Consequently, the purification of indoor air quality has raised more and more concerns in the public society and academic community [1,2]. Many commercial products and solutions are available in the market to improve indoor air quality, among which the vertical plant wall system (VPS) has proved to be a promising solution [3]. A vertical plant wall system, also named as living wall [4] or vertical greenery system [5] in the literature, is consisted of diverse green vegetation planted on the surface of a vertical growing medium. A statically placed vertical plant wall system can contribute to the indoor climate by passively purifying the air while the procedure can be accelerated and enhanced by integrating lighting, ventilation and irrigation systems to the plant wall [3]. The advancement of Internet of Things (IoT) and cloud technologies has sparked revolution in many traditional applications and paved the way for the approaching smart home and smart city era. In light of this, our previous study has proposed an IoT and public cloud-based remote monitoring and management system for the vertical plant wall industry, aiming at tackling the maintenance challenges encountered by plant wall suppliers [6]. As shown in Fig. 1, in such a system, a series of environment sensors are locally deployed on the plant wall while the sensory data are continuously transmitted to the Azure cloud server for storage and visualization purpose. Through a web-based human–machine interface (HMI), administrators are capable of performing remote control over the actuators, i.e., the lighting, ventilation and watering systems, in order to maintain a purified indoor climate.

As a continuation of the previous research, a promising task is to take full advantage of the collected environmental data to realize indoor climate control with an active vertical plant wall system. The indoor climate data stream is continuously processed and analyzed by the cloud server in a near real-time manner to detect potential anomalies of environmental parameters. When anomalies are detected, alarms can be automatically triggered and sent to the administrator and reactive measures can be taken to adjust the indoor climate to prevent drastic changes in an indoor environment.

The core to realize indoor climate control is early anomaly detection of time series environmental data. In the context of indoor climate, an anomaly can be categorized into point anomaly and contextual anomaly [7]. A point anomaly refers to a single outlier value that

* Corresponding author.
E-mail addresses: yu.a.liu@liu.se (Y. Liu), pang.zhibo@se.abb.com (Z. Pang), magnus.b.karlsson@liu.se (M. Karlsson), shaofang.gong@liu.se (S. Gong).

https://doi.org/10.1016/j.buildenv.2020.107212
Received 6 May 2020; Received in revised form 24 July 2020; Accepted 15 August 2020
Available online 24 August 2020

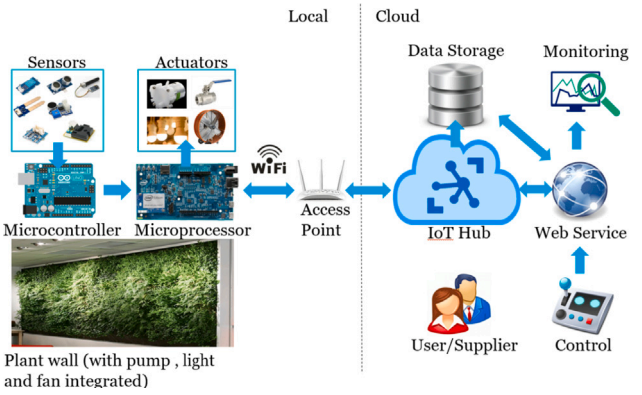


Fig. 1. System overview of Azure cloud and IoT-based remote monitoring and management system for vertical plant walls [6]. Environmental parameters fetched from sensors are transmitted to the cloud and monitored by administrators while control commands can be invoked to manipulate the actuators.

is largely different from the rest of the environmental data. This can be caused by software errors in device firmware, switch noise in the circuit or temporary interference of sensor readings. A point anomaly may not be harmful to the indoor climate but can interfere further data mining and lead to biased results. Besides, frequently appeared point anomalies also imply potential risk of failure of the sensing unit in a vertical plant wall system. A contextual anomaly refers to a series of data points that are inappropriate in a given period of time, although individually, the data values are within the normal range of an environmental parameter. It can be caused by external interference to an indoor climate, therefore, early detection of contextual anomalies enables reactive operations to be taken in time, i.e., to manipulate the lighting, ventilation and irrigation actuators in the vertical plant wall system so as to stabilize the indoor climate.

In this study, we intend to integrate novel anomaly detection methods based on machine learning into the specific application, a cloud-based plant wall system, to actively detect the anomalous indoor climate so as to further enhance the automation, ease the administrator's responsibility in maintenance, improve the reliability and scalability of the plant wall system, and finally reach indoor climate control in an intelligent manner. The main contributions of this study are as follows.

- Comprehensively investigated the performance of both prediction based and pattern recognition based anomaly detection methods on both point and contextual anomalies in a real indoor CO_2 concentration dataset and validate the results with a real temperature dataset.
- Proposed the most efficient anomaly detection method for vertical plant wall systems to detect point and contextual anomalies.
- Proposed a prediction-based method for data preprocessing to eliminate point anomalies and implemented real-time anomaly detection functions in the Azure cloud platform as a proof of concept.
- Showcased an exploitation of novel machine learning and Internet of Things technologies into building environment science to speed up the solution development.

The remainder of this paper is organized as follows. Section 2 states our motivation for conducting this research. Section 3 describes related work and Section 4 introduces background information about different anomaly detection techniques. Section 5 outlines the experiment design and details while Section 6 discusses the results and the potential adoption of the results. The last section concludes the paper.

2. Motivation

In this section, we detail the motivation of conducting such a research in four aspects to highlight the necessities.

2.1. Why vertical plant wall system?

The invention of the concept of vertical plant walls can date back to 1930s while it started to gain worldwide popularity and gradually turned out to be a modern decoration trend in urban architecture design [8]. A plethora of studies in the literature have investigated the influences of vertical plant walls to the environment that are deployed to either exterior [9–12] or indoor space [13–16] of buildings. Although to what extent can a vertical plant wall benefit environment is dependent on various factors, e.g., removal of the volatile organic compound (VOC) pollutant, is related to plant species, light intensity, indoor temperature and VOC concentration [17], a common conclusion can still be drawn that a vertical plant wall do have positive impact on environment by improving the air quality, increasing biodiversity, mitigating heat island effects, and reducing energy consumption through heating and cooling effect [18]. In recent years, active vertical plant walls have gained increased popularity, which feature a ventilation system to actively force air through the substrate of a plant wall. Besides, lighting and irrigation systems can also be integrated and remotely manipulated [6]. Researches have been conducted to investigate an active plant wall's effect on indoor air quality control [3,19–23], its airflow features [24] and pollution resistance [25]. The results suggest active plant walls achieve significant efficiency in maintaining VOC [3,19–21] and PM [3,21,22] to a low concentration, and greatly accelerate CO_2 reduction with moderate increase of illumination intensity and time [23,26]. Our experiments conducted in realistic places such as a workshop and an elderly house also reveal that an active vertical plant wall with 1.5 m^2 plants can balance the polluted air quality during inactive period such as night and weekend time.

The results accumulated from the literature and our research suggest that vertical plant walls, especially integrated with active actuators, apart from their original aesthetic functionality, can be an promising approach to contribute the building indoor climate.

2.2. Why anomaly detection for indoor climate control?

Many researches have exhibited that human physiological and psychological health is highly dependent on the building indoor climate [2,27,28]. For instance, Kim et al. [28] found out that positive indoor air quality and thermal comfort would cause blood pressure to lower. Jaber et al. [29] identified that decreasing the CO_2 concentration from high level to a healthier level would significantly improve occupants' memory and attention tasks. The indoor climate control in modern buildings mostly relies on heating, ventilation and air conditioning (HVAC) systems or other building automation systems that are mainly operated according to preset time schedules or thresholds, which brings considerable building energy consumption. Recent studies have shed light on strategies to optimize setpoints for HVAC systems that enable energy consumption savings [30] and real-time control [31]. However, these setpoint-based control methods are always triggered after the thresholds are exceeded, which leads to an inevitable latency before countermeasures are taken and potential sensation of discomfort.

In light of this, to realize indoor climate control in a predictive manner becomes significantly valuable. The concept of predictive maintenance (PDM) has been pervasively adopted in various industries [32] which can be introduced to indoor climate control field. PDM of indoor climate requires a comprehensive monitoring of historical and current environment conditions and provides prediction of upcoming environmental fluctuations, so as to enable intervention activities to react in advance. The core task to achieve PDM of indoor climate is to recognize the anomalous indoor climate changes as early as possible. In this way can the control methods, i.e., manipulations of lighting, ventilation and irrigation of a vertical plant system, be undertaken ahead of real sensation of discomfort by occupants. Therefore, in order to realize PDM of indoor climate, in this study, we focus on anomaly detection methods of indoor climate parameters that are collected by a vertical plant wall system.

2.3. Why internet of things?

Internet of Things reflects the strong connectivity of assets between physical and digital worlds through the Internet infrastructure [33]. IoT plays an essential role to the landing of information technology in a variety of research fields including building and environment, in which applications involve indoor climate monitoring and control, facility and energy management, occupants and resource tracking, and enhancement of indoor comfort, etc. [34–36].

In the context of vertical plant wall systems, IoT has greatly accelerated the widespread employment of this unique application [6]. Due to the nature that vertical plant walls are installed in a broad range of locations with different customers, the regular maintenance and plantcare can be costly and time consuming. With IoT technology, the environment sensors deployed to the plant wall is able to provide real-time tracking of indoor climate and growing conditions of plants while lighting, ventilation and irrigation systems can be remotely manipulated by building or plant wall maintainers. The environmental data are centrally stored in the cloud database which enables further data analytics on historical and real time data. In short, IoT is the key enabler to the transform from automation towards digitalization in the vertical plant wall industry.

2.4. Why machine learning?

The ultimate goal of IoT utilized in building environment is the indoor climate data collected from massive deployed sensors and the intelligence generated and mined from sensory data. As the last piece of the digital transform map, machine learning helps to process and extract hidden information from vast data accumulated from indoor climate that can hardly be analyzed by human efforts. Benefiting from the evolving cloud computing technology, mature machine learning techniques such as neural networks that rely heavily on powerful computing resource are able to be exploited by a variety of applications in the building environment to optimize operations [37], energy consumption [38] and human comfort [39], etc. In this study, we utilize novel machine learning methods, specifically neural networks, to dedicate to anomaly detection tasks for indoor climate control so as to take full advantage of IoT data collected by vertical plant walls and boost the intelligence in indoor climate control.

As a transdisciplinary study, we also showcase how evolving researches in isolated disciplines such as IoT and machine learning can be employed to counteract the obstacles and accelerate the problem-solving and solution development in the research field of building environment.

3. Related work

Anomaly detection techniques in time series data have been long researched in a multitude of disciplines but have not been witnessed in researches on vertical plant wall related indoor climate monitoring and maintenance. In this section, existing anomaly detection methods in the literature are reviewed.

One classic method that was popular in the past is the Autoregressive Integrated Moving Average (ARIMA) model. In [40], the authors developed an early warning system for network intrusion detection with an ARIMA model. The model is used to predict future trend of a network traffic based on previous data, and the normality of the system is estimated in real-time. In [41], the authors improved the detection accuracy of traditional ARIMA model by using a sliding window to determine historical data for modeling, rapidly updating the model after each sliding window, and making traffic prediction by short step exponential weighted average method. ARIMA model is effective at modeling data with regular daily or weekly patterns [42], however, in a frequently sampled indoor climate stream, the same pattern usually repeats after thousands of data points, which implies an extremely long moving window in the model and demands heavy computation.

Recent researches have attempted to use machine learning based approaches for anomaly detection. The authors in [43] developed a prediction-based online anomaly detection system for smart meter data. Wang et al. [44] adopted k-Nearest Neighbor (KNN) method to predict and detect anomalies in subjective thermal comfort votes from building occupants. In [45], Xin et al. presented an artificial neural network (ANN) based method to predict indoor air temperature and relative humidity values using an overlapped moving window. Similarly, Hamrani et al. [46] compared a series regression models in predicting greenhouse gas emissions. Buyuksahin et al. [47] enhanced the forecasting accuracy of time series data with a hybrid ARIMA-ANN model. However, with these prediction-based approaches, even though indoor environment parameters can be predicted several steps ahead, they are merely effective at detecting point anomalies but none of them can address the contextual anomalies.

A plethora of studies concentrate on context-aware approaches in anomaly detection. In [48], Daniel et al. proposed a collective contextual anomaly detection framework for smart buildings' energy consumption. They employed an autoencoder as a pattern learner to train the model to reconstruct input data patterns. By comparing the reconstruction error with a pattern recognizer engine, the normality of the data is determined. This framework was enhanced in [49] by introducing an overlapped sliding window to replace the non-overlapped sliding window and an ensemble of several detection methods. In [50], Cheng et al. had an analytical investigation of autoencoder-based methods for detecting anomalies in building energy data. A feedforward neuron network and an one-dimension convectional neuron network (CNN) are evaluated and benchmarked. Boquet et al. [51] also applied an autoencoder solution for anomaly detection in road traffic dataset. In terms of recurrent neural networks (RNN), Kim et al. [52] designed an RNN network to detect the anomalies in semiconductor manufacturing process while Xu et al. [53] proposed an improved long short-term memory (LSTM) RNN model for anomaly detection in IoT communication. Similarly, in [54], the authors exploited LSTM network to detect anomalies in sequence data in an automobile control network. Many studies found out that an ensemble of different machine learning methods has superior performance. For example, the authors in [55] and [56] utilized LSTM-based encoder–decoder structure to capture the behavior and health of a machine while in [57] a similar ensemble model was adopted to detect anomalous executions of assistant robots. Ding et al. [58] proposed an anomaly detection method using an ensemble of LSTM and the Gaussian mixture model. Similarly, Xu et al. [59] utilized an ensemble of LSTM and statistical modeling method, namely Grubbs's test, for anomaly detection in energy consumption dataset. These aforementioned studies have shown the promising potentials of machine learning, especially deep neural network based anomaly detection algorithms used in isolated areas. Therefore, a comprehensive study on applying machine learning based anomaly detection approaches to vertical plant wall systems for indoor climate control is significant to the building and environment research community.

Several classic machine learning methods such as one-class support vector machine(SVM) and principle component analysis (PCA) are served as out-of-box modules by the Azure cloud platform to accelerate the development of general purpose anomaly detection solutions. However, the effects of applying those aforementioned methods into real-time anomaly detection for indoor climate are still lack of study. In this paper, we investigate a series of most representative prediction and pattern recognition based anomaly detection methods, aiming at proposing a model that is most suitable for realizing indoor climate control with a vertical plant wall system.

4. Experiment methodology

In this section, the fundamentals of the models and measurement metrics used in the anomaly detection experiment are reviewed. The first set of models examined in the experiment is linear regression models which are commonly adopted as an efficient machine learning method to solve supervised learning problems [60]. Apart from this classic method, neuron network models emerge as promising candidates in machine learning.

4.1. Neural networks

Neural networks is a novel concept that includes a large class of models and learning methods. The proposal of neural networks has paved the way for traditional machine learning to enter the deep learning era.

4.1.1. Artificial Neural Network (ANN)

A feedforward artificial neural network is one of the most widely used neural networks and the start point of other network architectures. A typical structure of an ANN with multi-layers is shown in Fig. 2(a). The simplest case of an ANN is a single hidden layer network with M hidden cells and K output cells. In such a network, given a input X, the output Y_k is modeled as a linear combination of Z_m,

$$\begin{aligned} Z_m &= \sigma\left(\alpha_{0m} + \alpha_m^T X\right), m = 1, \ldots, M \\ T_k &= \beta_{0k} + \beta_k^T Z, k = 1, \ldots, K \\ f_k(X) &= g_k(T), k = 1, \ldots, K \end{aligned} \tag{1}$$

where σ is a nonlinear activation function and $g_k(T)$ is the output function [60]. For regression use case, the number of output cells is set to one ($K = 1$) and the identity function is used as the output function so that the ANN can generate a predictive target value.

4.1.2. Autoencoder

Autoencoder is a neural network model that is traditionally used for dimension reduction and feature learning while recent research has brought it into the landscape of generative modeling. The architecture of an autoencoder is similar to a regular ANN with strictly reduced widths in the hidden layers, as shown in Fig. 2(b), which can be seen as a combination of an encoder function f and a decoder function g. The encoder part learns the features of the input data and finally produces a reduced representation $\boldsymbol{h} = f(\boldsymbol{x})$, in the middle layer while the decoder part tries to reconstruct the input in the output layer $r = g(\boldsymbol{h})$. Typically, a loss function $L(\boldsymbol{x}, g(f(\boldsymbol{x})))$ such as the mean squared error is utilized to train the network by minimizing the loss so as to ensure the reconstructed output to be as close as possible to the input [61,62].

4.1.3. Recurrent neural network

A recurrent neural network is an essential deep learning model specialized for processing sequential data. As illustrated in Fig. 2(c), a RNN model takes a series of discrete time values $\boldsymbol{x}_t$ as input and has recurrent connections between consecutive hidden states $\boldsymbol{h}_t$, which enables a RNN to capture the relationship between neighboring input values. By taking advantage of sharing parameters, i.e., $\boldsymbol{w}_x, \boldsymbol{w}_h$, and $\boldsymbol{w}_y$, across all the parts of the model, a RNN can deal with variable-length input data. A typical variation of RNNs is bidirectional RNN, which introduces two hidden state layers in which hidden cells are connected in two reversed directions, in this way is the model capable to detect information not only from the past but also from the future in a sequence.

Long short-term memory (LSTM) is a special kind of RNN model designated to solving the long term dependencies problem in RNNs by imposing gate-controlled structure in the recurrent module. The module can be described using the following equations [61].

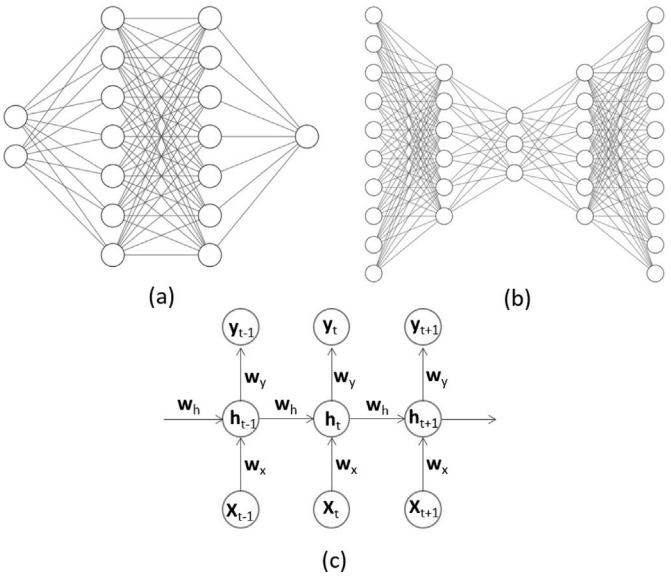


Fig. 2. Neural networks architectures: (a) Artificial neural networks. (b) Autoencoder. (c) Recurrent neural networks.

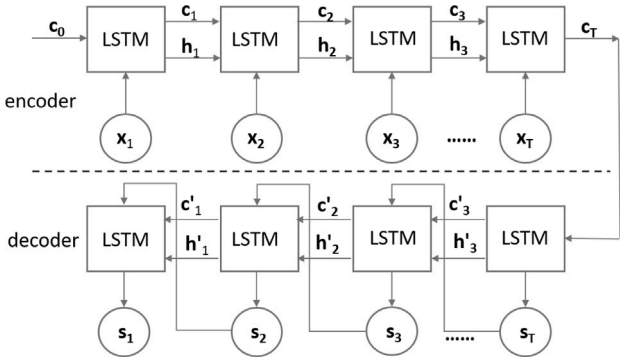


Fig. 3. The architecture of an LSTM-encoder–decoder [63].

$$\begin{aligned} \boldsymbol{f}_t &= \sigma\left(U_f \boldsymbol{x}_t + W_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f\right) \\ \boldsymbol{i}_t &= \sigma\left(U_i \boldsymbol{x}_t + W_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i\right) \\ \boldsymbol{o}_t &= \sigma\left(U_o \boldsymbol{x}_t + W_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o\right) \\ \boldsymbol{c}_t &= \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \tanh\left(U_c \boldsymbol{x}_t + W_c \boldsymbol{h}_{t-1} + \boldsymbol{b}_c\right) \\ \boldsymbol{h}_t &= \boldsymbol{o}_t \odot \tanh\left(\boldsymbol{c}_t\right) \end{aligned} \tag{2}$$

An LSTM module takes a vector $\boldsymbol{x}_t$ and the hidden state of the previous LSTM module $\boldsymbol{h}_{t-1}$ as input, with which three gates' status $\boldsymbol{f}_t$, $\boldsymbol{i}_t$, and $\boldsymbol{o}_t$ are determined. These three gates are used to control the computation of current internal state $\boldsymbol{c}_t$ and the hidden state $\boldsymbol{h}_t$, which are delivered to the next LSTM module later.

4.1.4. LSTM-encoder–decoder

An LSTM-encoder–decoder is a model constructed using a combination of LSTM models and autoencoders, as shown in Fig. 3. Similar to an autoencoder, the model is split into an encoder and a decoder part, both of them are LSTM networks. The encoder LSTM is able to learn the features from variable-length sequence data ($\boldsymbol{x}_1$ to $\boldsymbol{x}_T$ in the figure) and produce a fixed length representation ($\boldsymbol{c}_T$). The produced representation is then fed into the decoder LSTM to reconstruct the variable-length sequence in a reversed order ($\boldsymbol{s}_T$ to $\boldsymbol{s}_1$). By stacking more LSTM layers, one can make the LSTM-encoder–decoder model

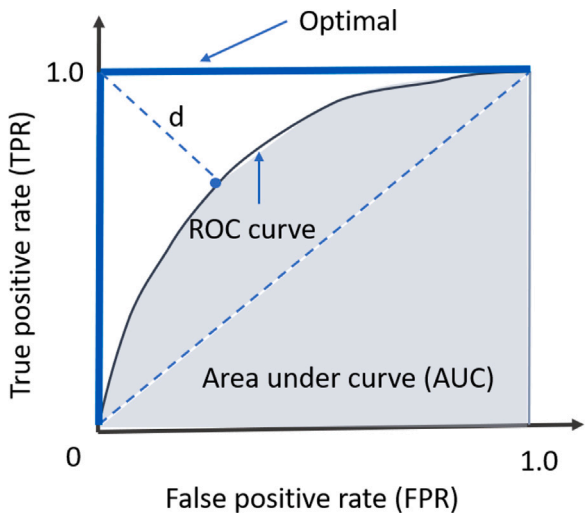


Fig. 4. Receiver operating characteristic (ROC) curve and threshold determinator [64]. The area under curve (AUC) reflects the performance of the classifier, i.e., a classifier with the optimal performance has an area under curve close to 1. The threshold is determined by locating a point in the ROC curve which has the nearest distance d to point (0, 1.0).

become deeper so as to achieve superior performance in sequence to sequence tasks [65].

4.2. Measurement

4.2.1. Prediction performance

Mean absolute error (MAE) and mean squared error (MSE) are utilized to estimate the prediction accuracy of models.

$$MAE = \frac{1}{N}\sum|\hat{y} - y| \tag{3}$$

$$MSE = \frac{1}{N}\sum(\hat{y} - y)^2 \tag{4}$$

The definitions of MAE and MSE are expressed in Eq. (5) and (6) where $\hat{y}$ is the predicted value, y the target value and N the number of samples. MAE tends to reflect the overall errors without considering the direction of errors while MSE is an additional metric used when models' MAE metrics are too close, as MSE penalizes large errors in a prediction model.

4.2.2. Classification performance

The nature of anomaly detection is a classification problem, therefore, receiver operating characteristic (ROC) curve is adopted as a graphical method to interpret the performance of classification models [64]. As shown in Fig. 4, a ROC curve is drawn using a series of true positive rate (TPR) and true negative rate (TNR) pairs which are generated by applying different thresholds to the loss function of a model. TPR and TNR are calculated using Eqs. (5) and (6)

$$TPR = \frac{TP}{P} \tag{5}$$

$$FPR = \frac{FP}{N} \tag{6}$$

where true positive (TP) and false positive (FP) are the numbers of correctly and falsely identified anomalies, respectively. P and N represent the total number of anomalies and normal points, respectively. To compare the performance of classification models, the area under the ROC curve (AUC) metric is employed. A higher AUC value indicates superior performance of a classifier model, therefore, the optimal model has a ROC curve with an AUC value equal to 1.0, as shown in Fig. 4, suggesting that the model can correctly detect all anomalies without making any false identification. In a realistic model, an AUC above 0.8 will indicate a model with excellent performance [66], and the optimal threshold is determined by finding the point that is nearest to the point $(0, 1.0)$ in the ROC curve, i.e., to iterate different thresholds in the curve till the distance d is minimal where d is defined in Eq. (9).

$$d = \sqrt{(1 - TPR)^2 + FPR^2} \tag{7}$$

5. Experimental design

In this study, two different categories of anomaly detection methods, i.e., prediction-based and pattern recognition-based detection methods, are applied to the sensory data collected from an indoor environment, to examine the effectiveness of detecting both point anomalies and contextual anomalies, and verify the possibility to integrate real-time anomaly detection into the cloud-based plant wall system. In this section, we demonstrate our experiment details.

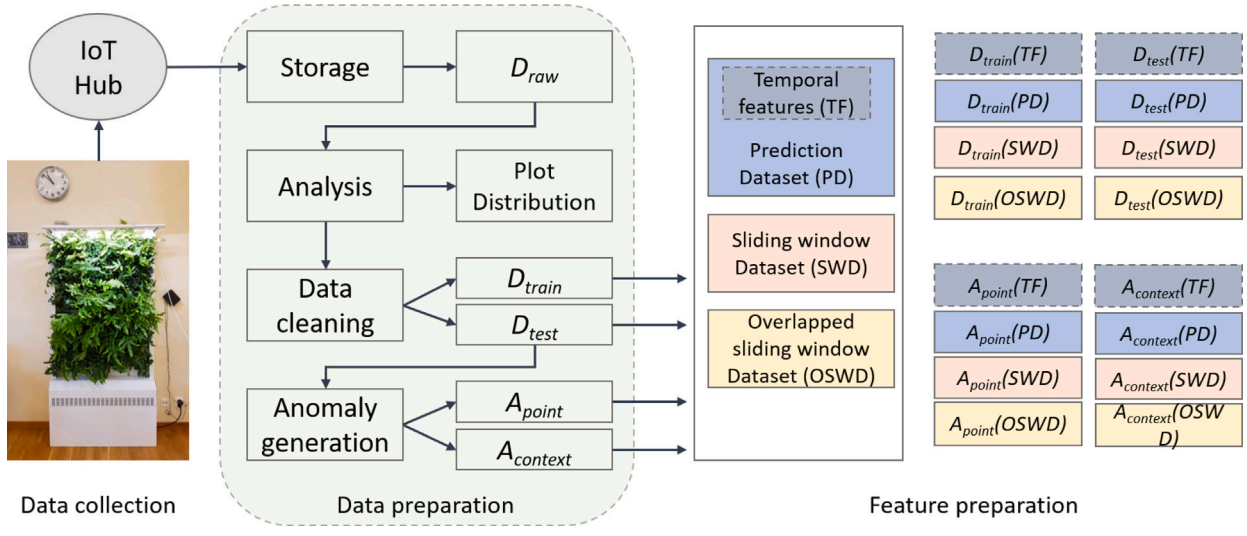


Fig. 5. Dataset preparation procedure. (1) Data collection: Environmental data read from sensors are transmitted to the cloud IoT Hub infrastructure and stored in the database. (2) Dataset preparation: The raw dataset is analyzed and performed data cleaning. It is then split into training set and test set. Anomaly generation is applied to the test set to create point and context anomaly datasets. (3) Feature preparation: Temporal features and prediction, sliding window or overlapped sliding window dependent features are extracted from training, test and anomaly datasets to create the final training and test datasets for different detection methods.

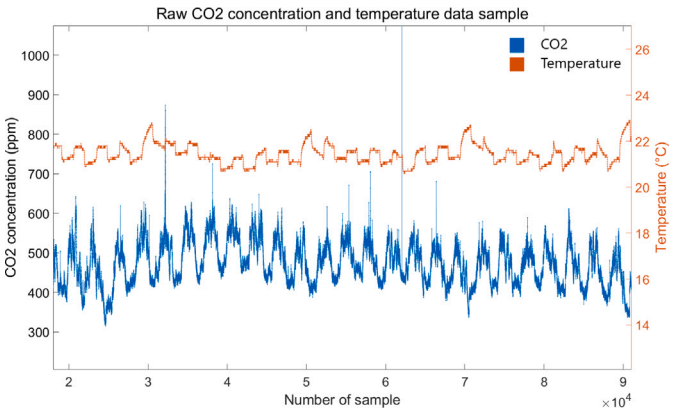


Fig. 6. Raw data samples of indoor CO_2 concentration level collected from an elderly house and temperature variation in a green room lab.

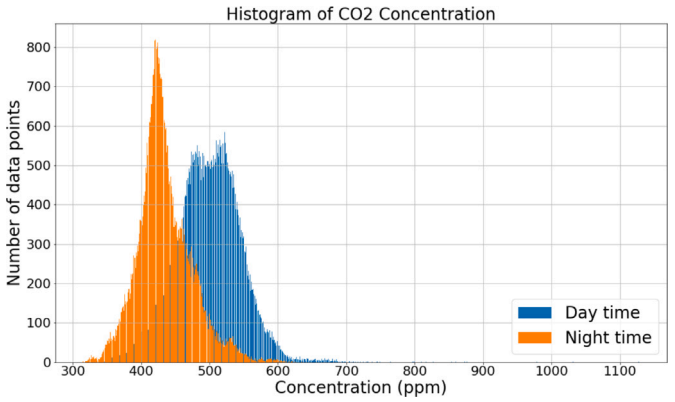


Fig. 7. CO_2 concentration level distributions. The day time and night time CO_2 concentrations roughly follow two normal distributions.

Table 1
CO_2 distribution parameters.

Name	μ (ppm)	σ (ppm)
Day time CO_2 distribution	498.80	46.56
Night time CO_2 distribution	434.41	36.35

5.1. Data collection

Fig. 5 illustrates how the training and test datasets are constructed. In the experiment, the raw data is based on the collected CO_2 data by a vertical plant wall that has been placed in an elderly home located in Norrköping, Sweden. The CO_2 concentration level in the elderly house was sensed, continuously transmitted to the cloud and stored in the database every 30 s. Hence, in the experiment we have 115 431 data points in total as D_{raw}. Besides, we also use another dataset of 137 592 records of temperature variations in our green room lab to validate the study. The raw data samples are shown in Fig. 6.

5.2. Data preparation

5.2.1. Data analysis

From the raw data graph, we can clearly observe that the CO_2 concentration level fluctuates following a day and night pattern with some point anomalies showing up as noise. The distributions of the data points in the day time and in the night time are plotted in Fig. 7, which suggests that the data points in the day time and in the night time roughly conform to two different normal distributions, respectively. The distribution parameters are listed in Table 1.

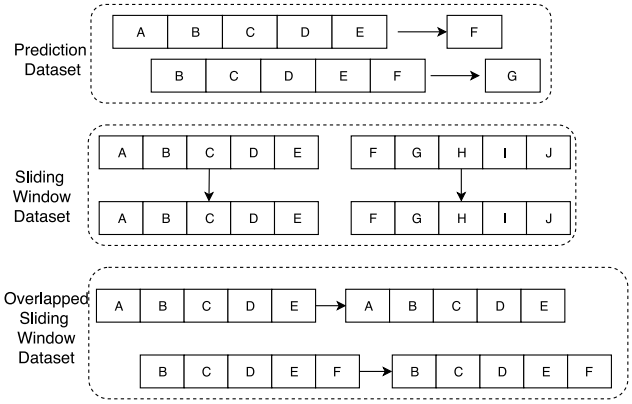


Fig. 8. Feature preparation for prediction and pattern recognition methods. In prediction dataset, the observation window moves one step ahead and uses the next data point as target. In sliding window dataset and overlapped sliding window dataset, the observation window moves LAG steps and one step ahead, respectively. In both cases, the data points in the observation windows are used as target.

5.2.2. Data cleaning

In order to get a normal dataset D_{normal} and ensure the quality of training, data cleaning is performed on D_{raw} by filtering the abnormal data points. Empirically, point anomalies rarely appear in the dataset. We manually identified 453 anomalies in D_{raw} and found out that more than 99% of the data points are within three standard deviations of the mean values for both day time and night time. Therefore, we use $\mu - 3\sigma$ and $\mu + 3\sigma$ as the lower and upper boundaries to filter the raw dataset. If the data point is outside of this normal range, we replace this outlier with the average of its neighboring values, and this procedure is repeated to get the point anomalous data vanished.

After that, this cleaned dataset D_{normal} was split into normal training dataset D_{train} and normal test dataset D_{test}, which account for 80% and 20% of D_{normal}, respectively. The training set D_{train} is used to train the anomaly detection model to learn normal features while D_{test} is used for performance evaluation.

5.2.3. Anomalous data generation

Two anomalous test sets A_{point} and $A_{context}$, and corresponding gold standards are manually generated for point anomaly and contextual anomaly scenarios, respectively, to get TPR and FPR values.

To generate an anomalous test dataset, a contaminating probability P is decided at first. In our experiment, this probability is set to 0.05, indicating 5% data points of the normal test set D_{test} will be contaminated (i.e., replaced) by anomalous values.

For point anomalies, a uniform distribution function is applied to the index in D_{test} to choose which data point is to be replaced. Once selected, the value is replaced by an anomalous value that is generated by uniform distribution within $[minimal, \mu - 3\sigma]$ and $[\mu + 3\sigma, maximal]$ according to whether the time stamp belongs to day or night period. For contextual anomalies, several consecutive data points with variant lengths that are randomly chosen from $[5, 20]$ are selected from D_{test}. These consecutive data points are replaced with values that are 12 to 18 h far from the originals so as to guarantee an obvious contextual difference.

5.3. Feature preparation

After D_{train}, D_{test}, A_{point}, and $A_{context}$ are generated, the real training and test datasets are constructed by extracting or appending features. The feature preparation procedure is depicted in Fig. 8 where three types of features and targets are prepared.

An observation window size LAG shall be decided first. For a linear regression and artificial neural network model, the LAG value can be as

Table 2
Added temporal features.

Feature	Details
Hour	Hour value extracted from the timestamp of the target data point
Minute	Minute value extracted from the timestamp of the target data point
Second	Second value extracted from the timestamp of the target data point
Stamp	Timestamp converted to second value in a single day
Mean	The mean value within one observation lag window
Std	The standard deviation value within one observation lag window
Difference	The difference between the first data point and the last data point within an observation window
Pre-difference	The difference between the mean values of current observation window and the previous window
Post-difference	The difference between the mean values of the later observation window and current window
Q1	The 25th percentile
Q2	The 50th percentile
Q3	The 75th percentile
IRQ	Interquartile range (difference between Q1 and Q3)

Table 3
Performance comparison of different regressors. Huber regressor tops the others in MAE and MSE metrics.

Regressor name	Description	MAE	MSE
Linear	Ordinary least squares linear regression	4.7442	42.1025
Lasso	Linear model trained with L1 prior as regularizer	4.7447	42.1083
Ridge	Linear least squares with L2 regularization	4.7443	42.1032
ElasticNet	Linear regression with combined L1 and L2 priors as regularizer	4.7527	42.2010
Huber	Linear regression model that is robust to outliers	4.7377	42.0868
LassoLars	Lasso model fit with Least Angle Regression	45.4761	3073
Passive Aggressive	Passive Aggressive Regression	5.7975	58.77

large as 1440 which represents that a 12 h observation window is used for predicting the next value. However, for an LSTM or bi-LSTM model, as LAG value increases, the training time increases dramatically, e.g., a LAG set to 120 takes 15 min for an LSTM model with 128 units to finish one epoch of training. In the end, to make a balance between performance and training difficulty, the LAG value is constantly set to 10 in all experiments.

For prediction-based methods, as the observation window moves along the time increase direction, all the data points within the observation window become the features while the first data point after the window becomes the target of this sample, resulting in a one-step ahead prediction. Besides, as a common practice [48], a series of temporal features (TF), as listed in Table 2, are extracted and added to the time series data to construct a separate training and test dataset for multi-variant experiment.

For pattern recognition-based methods, the features are prepared with both a sliding window and an overlapped sliding window. In the former case, all datasets are divided into windows with fixed length LAG while in the latter case, the consecutive windows are overlapped with $LAG - 1$ data points. Due to the nature of pattern recognition, the targets are the same as the features. The models are trained to recognize the normal data patterns and the model's output shall be as close to the input as possible. When applied to the anomalous dataset, the models can hardly recover the original data which causes large deviations between anomalous input and predicted output. In this way are we able to detect the anomalous windows.

In the end, a series of training and test datasets based on D_{train}, D_{test}, A_{point}, and $A_{context}$ are generated, as shown in Fig. 5.

5.4. Experiment 1: Point anomaly detection

In this experiment, both prediction and pattern recognition based methods are evaluated for point anomaly detection.

5.4.1. Prediction-based methods

Regression. Regression models are the simplest one to implement but usually with surprisingly good performance. Taking advantage of the Scikit-learn linear model library, 7 different regression methods, i.e., Linear Regressor, Lasso, Ridge, ElasticNet, Huber Regressor, LassoLars, and Passive Aggressive Regressor, are evaluated. The regression models are trained with $D_{train}(PD)$ and measured with $D_{test}(PD)$ to decide which regression model shall be used in the later comparison with neural network models, which can be denoted in this

$$D_{train}(PD) \rightarrow D_{test}(PD)$$

The detailed MAE and MSE metrics are listed in Table 3. The results show that the first five regressors performance are quite close in this dataset, which outperform the others to a considerable level, among which Huber regressor has topped all in the two metrics.

Neural networks. Three neural network models, ANN, LSTM and bi-LSTM, are benchmarked for prediction-based detection. During the model training procedure, the hyperparameters are properly tuned to find models with an optimal performance. Thus, mean squared error is used as the metric for loss function and Adam optimizer is involved to adjust the learning rate in an adaptive manner. We set the training epoch to 10 to make a balance between performance and time efficiency. During each epoch, the training set is shuffled to ensure the model remain general and reduce overfitting. Besides, 10% of data points in the training set are selected as a validation set in each epoch to prevent the models from overfitting training data.

$D_{train}(PD)$ is the training set, $D_{test}(PD)$ is used to get the MAE and MSE metrics, and $A_{point}(PD)$ is used for drawing the ROC curve as mentioned in Section 3. For each model, training and test sets that are appended with temporal features are also applied to get the performance as a comparison to pure time series data. The whole procedure can be denoted as follows.

$$D_{train}(PD) \rightarrow D_{test}(PD), A_{point}(PD)$$
$$D_{train}(TF) \rightarrow D_{test}(TF), A_{point}(TF)$$

Table 4
A comparison of point and contextual anomaly detection for CO_2 using prediction (PD), sliding window (SWD) and overlapped sliding window dataset (OSWD). Autoencoder and LSTM-encoder–decoder top the others in point and contextual anomaly detection respectively, by comprehensively considering the network complexity, training time, area under curve (AUC), and optimal true positive rate (TPR) and false positive rate (FPR) metrics. (White cells: point anomaly detection results. Gray cells: contextual anomaly detection results.).

Model	Hidden layers	Neurons	Total parameters	Training time (s)	MAE (D_test)	MSE (D_test)	AUC	Opt. TPR (%)	Opt. FPR (%)
Regression (PD)	/	/	/	1.8	4.7	42.2	0.9895	98.1	4.9
					4.7	42.1	0.6606	61.9	40.2
ANN (PD)	3	[200, 200, 50, 1]	52501	60	5.6	52.9	0.9932	97.6	4.0
					4.7	42.0	0.6902	58.8	31.7
LSTM (PD)	1	[128, 1]	66689	230	4.8	42.5	0.9891	98.6	5.1
					4.8	42.6	0.6591	53.0	28.8
bi-LSTM (PD)	2	[256, 1]	133377	380	4.7	41.9	0.9907	98.1	4.8
					8.1	99.9	0.6066	52.9	37.8
ANN (TF)	4	[200, 200, 50, 1]	54701	60	10.1	142.1	0.9947	97.6	3.5
					8.0	99.1	0.6898	58.2	29.9
AE (SWD)	4	[128, 32, 128, 10]	11050	**10**	2.9	15.8	**0.9969**	**98.6**	**0.9**
					3.3	30.3	0.8770	77.2	13.9
LSTM (SWD)	2	[128, 10]	66689	30	2.2	12.6	0.9981	98.5	1.8
					3.0	48.1	0.8587	76.7	19.8
bi-LSTM (SWD)	2	[256, 10]	133377	40	2.9	16.5	0.9976	98.5	1.3
					4.3	78.7	0.8648	80.2	16.1
LSTM multi-layer (SWD)	6	[128, 64, 32, 64, 128, 10]	252161	90	11.2	411.4	0.9947	97.9	3.1
					6.0	187.5	0.8692	78.7	19.1
LSTM-ED (SWD)	3	[128, 128, 10]	198273	50	4.4	33.3	0.9969	99.0	2.1
					6.0	189.6	**0.8757**	**80.7**	**14.8**
LSTM-ED multi-layer (SWD)	5	[128, 64, 64, 128, 10]	247937	80	7.3	170.8	0.9873	98.2	5.6
					10.5	558.9	0.8711	76.2	10.3
AE (OSWD)	4	[128, 32, 128, 10]	11050	40	0.2	0.08	0.9967	98.9	2.6
					0.14	0.04	0.7607	84.2	34.4
LSTM (OSWD)	2	[128, 10]	66689	270	0.06	0.006	0.9691	92.9	4.4
					0.45	0.27	0.7792	81.6	30.6
LSTM-ED (OSWD)	3	[128, 128, 10]	198273	500	5.9	66.4	0.9911	98.1	5.9
					4.9	45.1	0.8635	83.9	32.2
SVM	/	/	/	/	/	/	0.96	86.6	9.6
							0.48	/	/
Isolate forest	/	/	/	/	/	/	0.9525	90.5	7.3
							0.47	/	/
PCA	/	/	/	/	/	/	0.6845	78.6	48.5
							0.69	61.2	24.9

5.4.2. Pattern recognition-based methods

As a comparison to prediction-based methods, six different neural network models (AE, LSTM, bi-LSTM, LSTM with multi-layer, LSTM-ED, and LSTM-ED with multi-layer) are evaluated for pattern recognition-based detection. The sliding window and overlapped sliding window datasets are employed to benchmark each model, as denoted below.

$$D_{train}(SWD) \rightarrow D_{test}(SWD), A_{point}(SWD)$$
$$D_{train}(OSWD) \rightarrow D_{test}(OSWD), A_{point}(OSWD)$$

During the training, the parameter settings such as loss function, optimizer, shuffle switch, validation set portion and epoch number, remain the same as the ones used before.

In addition to the aforementioned models, several baseline models, i.e, the SVM, PCA and isolation forest models, are also introduced into the benchmarking as a performance reference. All of the models are implemented using Tensorflow, Keras, Numpy and Scikit-learn libraries to speed up the benchmarking.

5.5. Experiment 2: Contextual anomaly detection

In this experiment, the models are benchmarked to measure the capabilities of detecting contextual anomalies in environmental data. The experiment is also categorized into two groups corresponding to prediction-based and pattern recognition-based methods. Similarly, for pattern recognition models, both sliding window and overlapped sliding window datasets are employed. The model training settings are kept the same as in experiment 1. The used training and test datasets are denoted as follows.

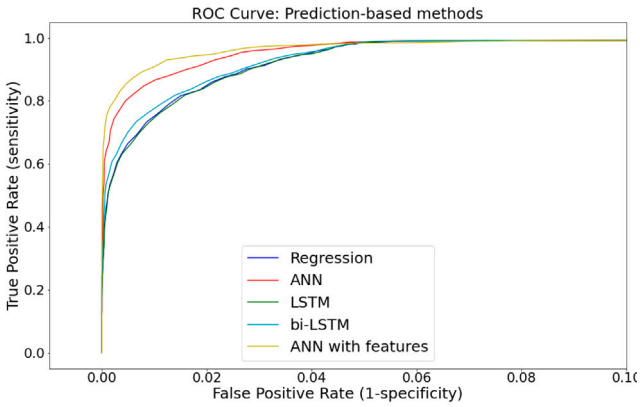
$$D_{train}(PD) \rightarrow D_{test}(PD), A_{context}(PD)$$
$$D_{train}(TF) \rightarrow D_{test}(TF), A_{context}(TF)$$
$$D_{train}(SWD) \rightarrow D_{test}(SWD), A_{context}(SWD)$$
$$D_{train}(OSWD) \rightarrow D_{test}(OSWD), A_{context}(OSWD)$$

6. Results and discussion

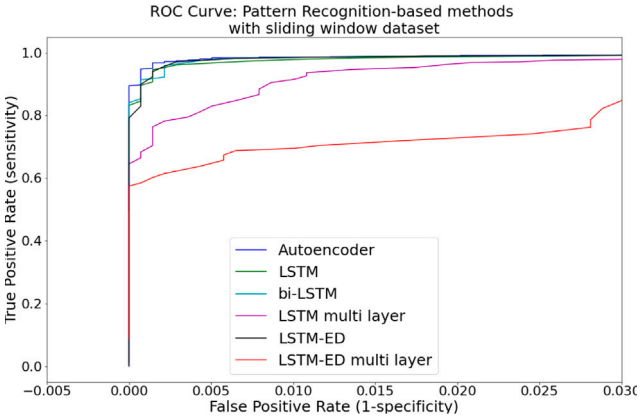
6.1. Experiment 1 on CO_2 dataset

The results of point anomaly detection benchmarking are shown in Table 4 (white cells) which exhibits the models parameters, training time, performances in terms of MAE, MSE and AUC. The optimal TPR and FPR values are also presented. The ROC graphs are depicted in Fig. 9.

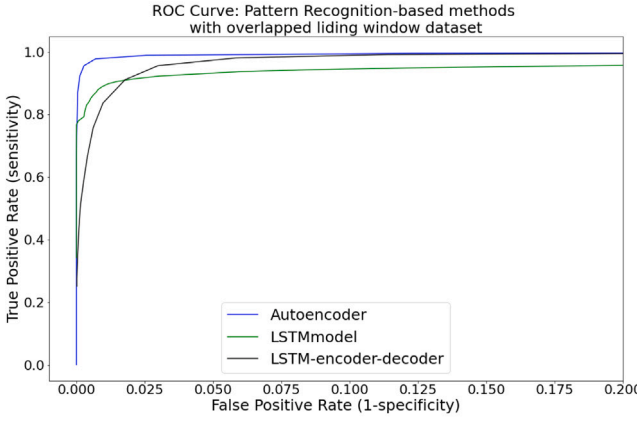
Among prediction-based methods, the AUC performance of ANN model tops the rest models, which can be observed in the ROC curve in Fig. 9(a). Benefiting from high complexity of network structure and



(a) ROC of prediction-based models

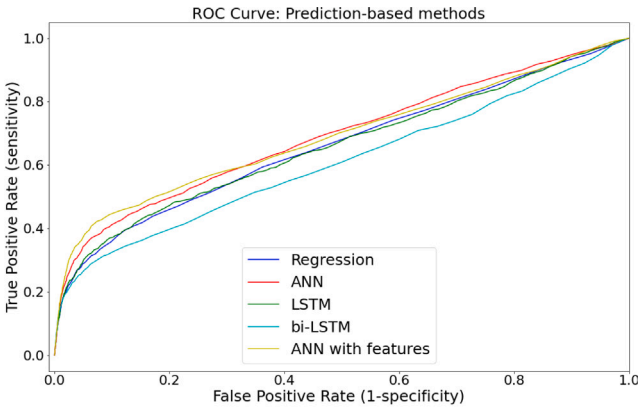


(b) ROC of pattern recognition-based models using sliding window dataset

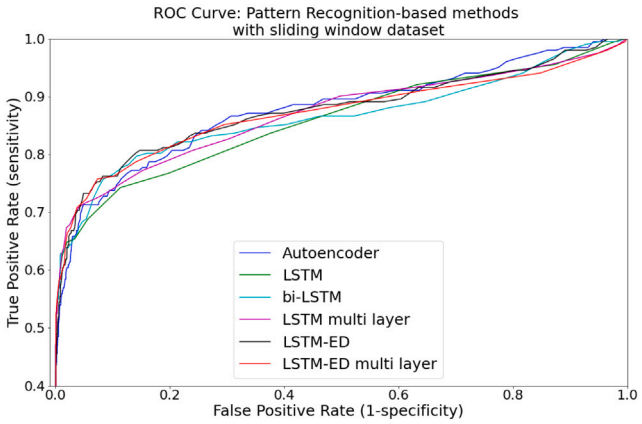


(c) ROC of pattern recognition-based models using overlapped sliding window dataset

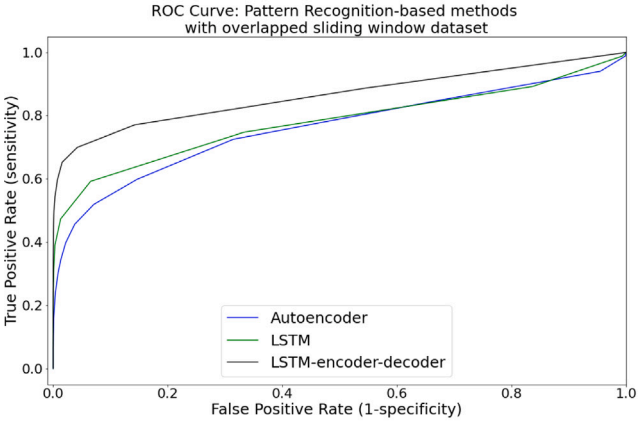
Fig. 9. Receiver operating characteristic (ROC) of point anomaly detection. (a) ANN with temporal features tops prediction-based methods. (b, c) Autoencoder tops pattern recognition-based models in both sliding window and overlapped sliding window datasets.



(a) ROC of prediction-based models



(b) ROC of pattern recognition-based models using sliding window dataset



(c) ROC of pattern recognition-based models using overlapped sliding window dataset

Fig. 10. Receiver operating characteristic (ROC) of contextual anomaly detection. (a) ANN models still top prediction-based methods. (b, c) LSTM-ED shows optimal in sliding window and overlapped sliding window datasets.

long training time, LSTM and bi-LSTM models gain superior TPR with a slight sacrifice in FPR metrics. With temporal features appended to the sensory data, the performance of the ANN model improves to some degree, as expected. However, the same training dataset with temporal features cannot achieve any benefits when applied to LSTM related models but only cause explosions of the loss during training procedure. This is because LSTM structure is designed to handle time series data points while the dataset with added temporal features is not a sequential data anymore, thus, the internal correlations between consecutive values have lost.

Regarding the pattern recognition-based methods, the performance of neural network models (e.g., AE, LSTM and LSTM-ED) with the sliding window dataset are higher, which suggests an overlapped sliding window is not necessary for point anomaly detection. Among all, the autoencoder model shows a superior AUC value (0.9967) with surprisingly limited parameters (11 050) and short training time (10 s),

i.e, Compared to LSTM, bi-LSTM and LSTM-ED models, autoencoder is 5, 12 and 18 times less in parameters, and 3, 4 and 5 times shorter in training time, respectively. Besides, the TPR is quite high (98.6%) and the FPR is the lowest (0.9%) compared to other models, when its optimal threshold is applied. Fig. 9(b) suggests that the performance of AE, LSTM, bi-LSTM and LSTM-ED models are quite close. However, considering the network complexity and training difficulty, the AE model is still preferred. In general, due to the nature that a point anomaly is not difficult to recognize, all the models benchmarked with the prediction dataset and the sliding window dataset have relatively high AUC values.

The performance of the three baseline models can hardly catch up with the aforementioned models, which hints the fact that we cannot rely on these out-of-box tools provided by the cloud platform to detect point anomalies, but have to implement more complicated solutions, e.g., neural network models.

6.2. Experiment 2 on CO_2 dataset

Table 4 (gray cells) shows the performance comparison of different models tested in contextual anomaly detection. The ROC curves of prediction and pattern recognition-based models are shown in Fig. 10. Comparing the ROC curves in 10(a), 10(b) and 10(c), it is obvious that pattern recognition-based methods are more efficient in detecting contextual anomalies than prediction-based methods. This conclusion is also verified by the AUC values in the table that most pattern recognition-based methods have a AUC value above 0.8 while prediction-based methods can hardly reach 0.7. This complies to the intuitive expectation that contextual anomalies are prone to damage of patterns in the data stream so that they are more likely to be detected by pattern recognition-based methods.

In general, the performances among all of the pattern recognition-based methods are close to each other regardless of using sliding window or overlapped sliding window, i.e., the models do not benefit from a fine-grained dataset with shorter time step (30 s) between two consecutive samples. The performance of the LSTM-ED model with sliding window dataset tops the experiment with a AUC value of 0.8757 and the TPR reaches 80.7% and FPR 14.8%, showing that an ensemble of multiple neural network structures does improve the performance.

Compared to our self-implemented models, the three baseline models perform much worse. The AUC values of the SVM and Isolate forest models are below 0.5, suggesting their performance in the benchmarking datasets are worse than a random classifier.

Due to the high complexity of recognizing anomalies in a environmental context, AUC metrics of all models are not as high as the AUC in experiment 1. Moreover, the optimal TPR metric also can hardly be higher than 80%, except the LSTM model and the LSTM-ED model with overlapped sliding window dataset, which is realized by sacrificing the FPR metrics to a high degree. The results imply that there are still space to improve the accuracy of detecting contextual anomalies.

6.3. Validation on temperature dataset

A temperature dataset collected in a university lab is used to validate the results. The dataset exhibits similar day and night variations as in the CO_2 dataset. In addition to that, the temperature goes up during weekend due to the shutdown of ventilation system. The same experiment configurations, i.e., training and test data preparation and neural network parameters, are applied to the temperature dataset, and the results of point and contextual anomaly detection are recorded in Table 5.

In both point and contextual anomaly detection, ANN still maintains the advantage among prediction-based methods, similar to the results in the CO_2 dataset. However, added temporal features only worsen ANN model's performance. This is due to the added temporal features cannot reflect the extra weekly pattern in the temperature dataset, but only introduce interference in this case. Looking at pattern recognition-based methods, the same conclusion in the CO_2 dataset holds here. On one hand, the performance of autoencoder still shows dominance in the point anomaly detection in both sliding window and overlapped sliding window datasets. With the simplest network structure and lowest training time (10 s), an autoencoder achieves an AUC value of 0.9938 with considerably high TPR (97.1%) and the lowest FPR (0.9%). On the other hand, the LSTM-ED model with a sliding window dataset keeps on top of the other models with the highest AUC value 0.8870, near highest TPR (77.6%) and lowest FPR (9.0%).

In conclusion, even though an extra weekly pattern is introduced to the temperature dataset, the benchmarking results are generally in accordance with that of the CO_2 dataset, suggesting that the autoencoder model gets the best performance in point anomaly detection while the ensemble model LSTM-ED is optimal for contextual anomaly detection for indoor climate data. It is worth noting that the CO_2 and temperature datasets used in this study show stable fluctuation patterns. Considering the high complexity of indoor environments, the anomaly detection performance of proposed models applied to environmental parameters that are lack of visible variation patterns can vary and still needs further investigation.

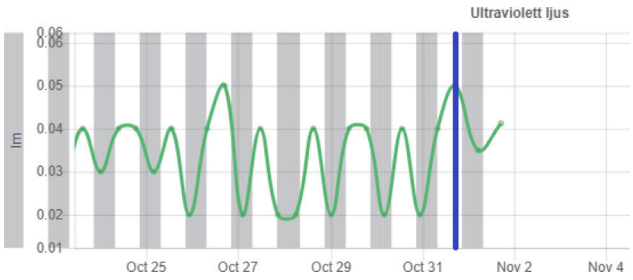


Fig. 11. Prediction-based anomaly detection example for ultraviolet light data deployed in the Azure cloud. Day and night time are marked in white and gray colors. The data curve to the right side of the blue bar is shown with predicted values. Once the difference between real values and the predicted values exceeds the threshold, an alarm will be generated.

6.4. Adoption of the results

With a comprehensive consideration of the two experiments, we conclude that the autoencoder and LSTM-ED models surpass the others in terms of detecting point anomaly and contextual anomaly respectively in an indoor environment, and thus can be deployed into the plant wall system to realize PDM tasks.

In practice, point anomalies are rarely shown up and usually are not harmful to the environment. However, they may lead to biased data analytic results and thus need to be eliminated before further data mining are applied to the sensory data. Base on the results of experiment 1, we propose a new approach for data cleaning. First, the autoencoder model is deployed to detect point anomalies. When a point anomaly window is detected, prediction-based methods can be utilized to make reasonable prediction of the anomalous data. Among the prediction-based methods, regression has the lowest loss in terms of MAE and MSE, indicating it is good at reproducing the original value of a data point. Besides, the cost of implementing and training a regression model is limited. Therefore, regression can be adopted to reproduce the data values in the anomalous window and use the reproduced values to replace the anomalous data points. In this way can we realize data cleaning in a flexible manner.

As a proof-of-concept, a prediction-based model is deployed into the cloud-based plant wall system. The model is deployed in a container hosted in the Azure cloud platform, working together with a web-based interface. Fig. 11 shows an example of applying the method into ultraviolet light sensor data. In the picture, the curve to the right of

Table 5
A comparison of point and contextual anomaly detection for temperature data. The neural network parameters remain the same as used in CO_2 dataset. Autoencoder and LSTM-encoder–decoder still show optimal performances in point and contextual anomaly detection, respectively. (White cells: point anomaly detection results. Gray cells: contextual anomaly detection results.).

Model	Training time (s)	MAE (D_test)	MSE (D_test)	AUC	Opt. TPR (%)	Opt. FPR (%)
Regression (PD)	2.2	0.01	0.0006	0.9828	93.7	6.7
		0.01	0.0006	0.6944	70.2	32.6
ANN (PD)	90	0.009	0.01	0.9850	95.1	4.7
		0.02	0.001	0.7898	72.0	23.9
LSTM (PD)	330	0.03	0.001	0.9789	95.2	5.8
		0.05	0.003	0.7694	69.7	35.2
bi-LSTM (PD)	580	0.07	0.006	0.9847	94.4	5.5
		0.04	0.002	0.7378	78.1	58.0
ANN (TF)	90	0.39	0.2	0.9793	92.3	5.4
		0.13	0.03	0.6939	62.0	31.4
AE (SWD)	**10**	0.014	0.0007	**0.9938**	**97.1**	**0.6**
		0.04	0.002	0.8830	76.8	8.5
LSTM (SWD)	30	0.03	0.005	0.9943	98.0	6.7
		0.05	0.009	0.6609	56.4	34.5
bi-LSTM (SWD)	60	0.03	0.002	0.9946	95.4	2.2
		0.05	0.008	0.8821	80.0	13.3
LSTM multi-layer (SWD)	120	0.08	0.02	0.9935	98.2	5.7
		0.07	0.02	0.8537	72.9	9.0
LSTM-ED (SWD)	60	0.02	0.0007	0.9950	98.1	9.3
		0.04	0.008	**0.8870**	**77.6**	**8.0**
LSTM-ED multi-layer (SWD)	100	0.07	0.008	0.9977	97.2	1.1
		0.08	0.014	0.8465	77.2	19.6
AE (OSWD)	90	0.03	0.001	0.9928	96.6	1.3
		0.05	0.004	0.8683	82.3	21.2
LSTM (OSWD)	310	0.006	6.35	0.9860	95.6	3.9
		0.002	1.12	0.8648	77.4	15.6
LSTM-ED (OSWD)	570	0.03	0.002	0.9940	97.0	3.8
		0.01	0.0005	0.8847	77.6	12.5
SVM	/	/	/	0.9539	90.9	14.8
				0.465	/	/
Isolate forest	/	/	/	0.7777	52.2	13.6
				0.474	/	/
PCA	/	/	/	0.8508	87.2	33.2
				0.4483	/	/

the blue line is predicted by the model. Once the real value vary from the prediction to some extent, anomaly alert will be generated to the administrators.

7. Conclusion

Vertical plant wall systems, embedded with sensors and actuators, have become a promising application for indoor climate control. In this study, we explore the possibility of applying machine learning based anomaly detection methods to vertical plant wall systems so as to enhance the automation and improve the intelligence to realize predictive maintenance of the indoor climate. It is also an example showcasing the advancements in machine learning and Internet of things can be fully utilized by researches on building environment to accelerate the solution development. The study benchmarked the performances of two category of methods, i.e., prediction-based and pattern recognition-based detection methods, by applying the models to the CO_2 and temperature dataset collected from real indoor environment. The results suggest that the autoencoder neural network model surpasses the others in point anomaly detection while the LSTM-ED ensemble model has a superior performance in contextual anomaly detection. Both of them surpass the performance of baseline models provided by the Azure cloud platform as out-of-box tools.

This study can be seen as a start point of integrating machine learning based anomaly detection methods into the vertical plant wall system to realize indoor climate control. There are several future work worthy of further exploration. The performance of machine learning models highly relies on the training data, which can be affected by many factors such as the quality of sensors, the sampling rate, and the size of dataset. Thus, how to improve the anomaly detection performance on less frequently sampled data with cheap sensors and small dataset shall be investigated. Besides, this study is limited to anomaly detection on univariate indoor environment data in which a stable pattern can be observed. It is worth noting that the complexity of indoor climate can vary from building to building and from parameter

to parameter. In complicated environment where a clear variation pattern cannot be easily observed for some indoor climate parameters, environmental anomalies can hardly be captured by univariate sensor data analysis or recognized by a single machine learning models. In this case, multivariate sensor data analysis, namely sensor fusion can be explored, which comprehensively estimates multiple sensors to find the hidden patterns, reduce uncertainty and improve anomaly detection accuracy so as to expand indoor climate control to more situations.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The Swedish Environmental Protection Agency and the Norrköping Fund for Research and Development in Sweden are acknowledged for financial support of the study. Vertical Plants System Sweden AB is acknowledged for providing the vertical plant wall as experiment platforms.

References

[1] B. Yu, Z. Hu, M. Liu, H. Yang, Q. Kong, Y. Liu, Review of research on air-conditioning systems and indoor air quality control for human health, Int. J. Refrig. 32 (1) (2009) 3–20, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140700708000984.

[2] J. Kim, M. Kong, T. Hong, K. Jeong, M. Lee, Physiological response of building occupants based on their activity and the indoor environmental quality condition changes, Build. Environ. 145 (2018) 96–103, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360132318305663.

[3] A. Bondarevs, P. Huss, S. Gong, O. Weister, R. Liljedahl, Green walls utilizing internet of things, Sens. Transducers 192 (2015) 16–21.

[4] E. Shafiee, M. Faizi, S.-A. Yazdanfar, M.-A. Khanmohammadi, Assessment of the effect of living wall systems on the improvement of the urban heat island phenomenon, Build. Environ. (2020) 106923, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360132320302821.

[5] A.B. Daemei, M. Azmoodeh, Z. Zamani, E.M. Khotbehsara, Experimental and simulation studies on the thermal behavior of vertical greenery system for temperature mitigation in urban spaces, J. Build. Eng. 20 (2018) 277–284, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2352710218302869.

[6] Y. Liu, K.A. Hassan, M. Karlsson, O. Weister, S. Gong, Active plant wall for green indoor climate based on cloud and internet of things, IEEE Access 6 (2018) 33631–33644.

[7] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Comput. Surv. 41 (3) (2009) 15:1–15:58, [Online]. Available: http://doi.acm.org/10.1145/1541880.1541882.

[8] P. Blanc, V. Lalot, J. Nouvel, G. Bruhn, The Vertical Garden: From Nature to the City, W.W. Norton, 2008, [Online]. Available: https://books.google.se/books?id=GRiV6E_xn8gC.

[9] U. Mazzali, F. Peron, P. Romagnoni, R.M. Pulselli, S. Bastianoni, Experimental investigation on the energy performance of living walls in a temperate climate, Build. Environ. 64 (2013) 57–66, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S036013231300084X.

[10] Q. Chen, B. Li, X. Liu, An experimental evaluation of the living wall system in hot and humid climate, Energy Build. 61 (2013) 298–307, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378778813001060.

[11] T. Koyama, M. Yoshinaga, H. Hayashi, K. ichiro Maeda, A. Yamauchi, Identification of key plant traits contributing to the cooling effects ofgreen façades using freestanding walls, Build. Environ. 66 (2013) 96–103, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360132313001315.

[12] E. Shafiee, M. Faizi, S.-A. Yazdanfar, M.-A. Khanmohammadi, Assessment of the effect of living wall systems on the improvement of the urban heat island phenomenon, Build. Environ. (2020) 106923, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360132320302821.

[13] J. Sánchez-Reséndiz, L.R.-G. a, F. Olivieri, E. Ventura-Ramos, Experimental assessment of the thermal behavior of a living wall system in semi-arid environments of central Mexico, Energy Build. 174 (2018) 31–43, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378778818304407.

[14] K. Gunawardena, K. Steemers, Living walls in indoor environments, Build. Environ. 148 (2019) 478–487, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360132318307091.

[15] D. Tudiwer, A. Korjenic, The effect of an indoor living wall system on humidity, mould spores and CO_2-concentration, Energy Build. 146 (2017) 73–86, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378778817313889.

[16] L. Pérez-Urrestarazu, R. Fernández-Cañero, A. Franco, G. Egea, Influence of an active living wall on indoor temperature and humidity conditions, Ecol. Eng. 90 (2016) 120–124, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925857416300507.

[17] M. Dela Cruz, J.H. Christensen, J.D. Thomsen, R. Müller, Can ornamental potted plants remove volatile organic compounds from indoor air? — a review, Environ. Sci. Pollut. Res. 21 (24) (2014) 13909–13928, http://dx.doi.org/10.1007/s11356-014-3240-x, [Online]. Available.

[18] V.O.-D. Cosola, F. Olivieri, L. Ruiz-García, J. Bacenetti, An environmental life cycle assessment of living wall systems, J. Environ. Manag. 254 (2020) 109743, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0301479719314616.

[19] T. Pettit, M. Bettes, A. Chapman, L. Hoch, N. James, P. Irga, F. Torpy, The botanical biofiltration of VOCs with active airflow: is removal efficiency related to chemical properties? Atmos. Environ. 214 (2019) 116839, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1352231019304686.

[20] F. Torpy, N. Clements, M. Pollinger, A. Dengel, I. Mulvihill, C. He, P. Irga, Testing the single-pass VOC removal efficiency of an active green wall using methyl ethyl ketone (MEK), Air Qual. Atmos. Health 11 (2) (2017) 163–170, http://dx.doi.org/10.1007/s11869-017-0518-4, [Online]. Available.

[21] T. Pettit, P.J. Irga, F.R. Torpy, The in situ pilot-scale phytoremediation of airborne VOCs and particulate matter with an active green wall, Air Quality, Atmosphere & Health 12 (1) (2018) 33–44, http://dx.doi.org/10.1007/s11869-018-0628-7, [Online]. Available.

[22] T. Pettit, P. Irga, P. Abdo, F. Torpy, Do the plants in functional green walls contribute to their ability to filter particulate matter? Build. Environ. 125 (2017) 299–307, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360132317304109.

[23] F. Torpy, M. Zavattaro, P. Irga, Green wall technology for the phytoremediation of indoor air: a system for the reduction of high CO_2 concentrations, Air Quality, Atmosphere & Health 10 (5) (2016) 575–585, [Online]. Available.

[24] P. Abdo, B.P. Huynh, P.J. Irga, F.R. Torpy, Evaluation of air flow through an active green wall biofilter, Urban For. Urban Green. 41 (2019) 75–84, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1618866718308276.

[25] N.J. Paull, P.J. Irga, F.R. Torpy, Active green wall plant health tolerance to diesel smoke exposure, Environ. Pollut. 240 (2018) 448–456, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0269749118304068.

[26] F. Torpy, P. Irga, M. Burchett, Profiling indoor plants for the amelioration of high CO_2 concentrations, Urban For. Urban Green. 13 (2) (2014) 227–233, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1618866713001325.

[27] J. Kim, T. Hong, J. Jeong, C. Koo, M. Kong, An integrated psychological response score of the occupants based on their activities and the indoor environmental quality condition changes, Build. Environ. 123 (2017) 66–77, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360132317302810.

[28] J. Kim, T. Hong, M. Kong, K. Jeong, Building occupants' psycho-physiological response to indoor climate and CO_2 concentration changes in office buildings, Build. Environ. 169 (2020) 106596, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S036013231930808X.

[29] A.R. Jaber, M. Dejan, U. Marcella, The effect of indoor temperature and CO_2 levels on cognitive performance of adult females in a university building in Saudi Arabia, in: CISBAT 2017 International Conference Future Buildings & Districts – Energy Efficiency from Nano to Urban Scale, Energy Procedia 122 (2017) 451–456, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S187661021732982X.

[30] S. Papadopoulos, C.E. Kontokosta, A. Vlachokostas, E. Azar, Rethinking HVAC temperature setpoints in commercial buildings: The potential for zero-cost energy savings and comfort improvement in different climates, Build. Environ. 155 (2019) 350–359, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S036013231930232X.

[31] W. Li, C. Koo, S.H. Cha, T. Hong, J. Oh, A novel real-time method for HVAC system operation to improve indoor environmental quality in meeting rooms, Build. Environ. 144 (2018) 365–385, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360132318305201.

[32] I. Analytics, Predictive maintenance report 2019-2024, 2019, [Online]. Available: https://iot-analytics.com/product/predictive-maintenance-report-2019-2024/.

[33] P. Ray, A survey on Internet of Things architectures, J. King Saud Univ.-Comput. Inf. Sci. 30 (3) (2018) 291–319, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1319157816300799.

[34] A. Daissaoui, A. Boulmakoul, L. Karim, A. Lbath, IoT and big data analytics for smart buildings: A survey, in: The 11th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 3rd International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops, Procedia Comput. Sci. 170 (2020) 161–168, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050920304506.

[35] S. Dhanalakshmi, M. Poongothai, K. Sharma, IoT based indoor air quality and smart energy management for HVAC system, in: Third International Conference on Computing and Network Communications (CoCoNet'19), Procedia Comput. Sci. 171 (2020) 1800–1809, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050920311741.

[36] Y. Jeon, C. Cho, J. Seo, K. Kwon, H. Park, S. Oh, I.-J. Chung, IoT-based occupancy detection system in indoor residential environments, Build. Environ. 132 (2018) 181–204, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360132318300611.

[37] X. Dai, J. Liu, X. Zhang, A review of studies applying machine learning models to predict occupancy and window-opening behaviours in smart buildings, Energy Build. 223 (2020) 110159, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378778820303017.

[38] K. Mason, S. Grijalva, A review of reinforcement learning for autonomous building energy management, Comput. Electr. Eng. 78 (2019) 300–312, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045790618333421.

[39] M. Ayoub, A review on machine learning algorithms to predict daylighting inside buildings, Sol. Energy 202 (2020) 249–275, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0038092X20303509.

[40] A.H. Yaacob, I.K.T. Tan, S.F. Chien, H.K. Tan, ARIMA based network anomaly detection, in: 2010 Second International Conference on Communication Software and Networks, 2010, pp. 205–209.

[41] Q. Yu, L. Jibin, L. Jiang, An improved ARIMA-based traffic anomaly detection algorithm for wireless sensor networks, Int. J. Distrib. Sens. Netw. 12 (1) (2016) 9653230, http://dx.doi.org/10.1155/2016/9653230, [Online]. Available.

[42] S. Ahmad, A. Lavin, S. Purdy, Z. Agha, Unsupervised real-time anomaly detection for streaming data, Neurocomputing 262 (2017) 134–147, Online Real-Time Learning Strategies for Data Streams, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231217309864.

[43] X. Liu, P.S. Nielsen, Scalable prediction-based online anomaly detection for smart meter data, Inf. Syst. 77 (2018) 34–47, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306437917303216.

[44] Z. Wang, T. Parkinson, P. Li, B. Lin, T. Hong, The Squeaky wheel: Machine learning for anomaly detection in subjective thermal comfort votes, Build. Environ. 151 (2019) 219–227, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360132319300861.

[45] X. Shi, W. Lu, Y. Zhao, P. Qin, Prediction of indoor temperature and relative humidity based on cloud database by using an improved BP neural network in chongqing, IEEE Access 6 (2018) 30559–30566.

[46] A. Hamrani, A. Akbarzadeh, C.A. Madramootoo, Machine learning for predicting greenhouse gas emissions from agricultural soils, Sci. Total Environ. (2020) 140338, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0048969720338602.

[47] Umit. Çavuş, Büyükşahin, Şeyda. Ertekin, Improving forecasting accuracy of time series data using a new ARIMA-ANN hybrid method and empirical mode decomposition, Neurocomputing 361 (2019) 151–163, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231219309178.

[48] D.B. Araya, K. Grolinger, H.F. ElYamany, M.A.M. Capretz, G. Bitsuamlak, Collective contextual anomaly detection framework for smart buildings, in: 2016 International Joint Conference on Neural Networks, IJCNN, 2016, pp. 511–518.

[49] D.B. Araya, K. Grolinger, H.F. ElYamany, M.A. Capretz, G. Bitsuamlak, An ensemble learning framework for anomaly detection in building energy consumption, Energy Build. 144 (2017) 191–206, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378778817306904.

[50] C. Fan, F. Xiao, Y. Zhao, J. Wang, Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data, Appl. Energy 211 (2018) 1123–1135, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306261917317166.

[51] G. Boquet, A. Morell, J. Serrano, J.L. Vicario, A variational autoencoder solution for road traffic forecasting systems: Missing data imputation, dimension reduction, model selection and anomaly detection, Transp. Res. C 115 (2020) 102622, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968090X19309611.

[52] C. Kim, J. Lee, R. Kim, Y. Park, J. Kang, DeepNAP: Deep neural anomaly pre-detection in a semiconductor fab, Inform. Sci. 457–458 (2018) 1–11, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002002551830375X.

[53] R. Xu, Y. Cheng, Z. Liu, Y. Xie, Y. Yang, Improved Long Short-Term Memory based anomaly detection with concept drift adaptive method for supporting IoT services, Future Gener. Comput. Syst. 112 (2020) 228–242, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X20302235.

[54] A. Taylor, S. Leblanc, N. Japkowicz, anomaly detection in automobile control network data with long short-term memory networks, in: 2016 IEEE International Conference on Data Science and Advanced Analytics, DSAA, 2016, pp. 130–139.

[55] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, G. Shroff, LSTM-based encoder-decoder for multi-sensor anomaly detection, in: ICML 2016 Anomaly Detection Workshop, New York, USA, 2016.

[56] P. Malhotra, V. Tv, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, G. Shroff, Multi-sensor prognostics using an unsupervised health index based on LSTM encoder-decoder, in: 1st SIGKDD Workshop on Machine Learning for Prognostics and Health Management, 2016.

[57] D. Park, Y. Hoshi, C.C. Kemp, A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder, IEEE Robot. Autom. Lett. 3 (3) (2018) 1544–1551.

[58] N. Ding, H. Ma, H. Gao, Y. Ma, G. Tan, Real-time anomaly detection based on long short-Term memory and Gaussian Mixture Model, Comput. Electr. Eng. 79 (2019) 106458, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045790618334372.

[59] C. Xu, H. Chen, Abnormal energy consumption detection for GSHP system based on ensemble deep learning and statistical modeling method, Int. J. Refrig. 114 (2020) 106–117, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140700720300992.

[60] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference and Prediction, second ed., Springer, 2009, [Online]. Available: http://www-stat.stanford.edu/~tibs/ElemStatLearn/.

[61] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, http://www.deeplearningbook.org.

[62] C.C. Aggarwal, Neural Networks and Deep Learning, Springer, Cham, 2018, p. 497.

[63] S. Park, B. Kim, C. Kang, C. Chung, J. Choi, Sequence-to-Sequence Prediction of Vehicle Trajectory via LSTM Encoder-Decoder Architecture, 2018, pp. 1672–1678.

[64] A. Tharwat, Classification assessment methods, Appl. Comput. Inform. (2018) [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2210832718301546.

[65] K. Cho, B. van Merriënboer, C. Gulcehre, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, 2014.

[66] J.N. Mandrekar, Receiver operating characteristic curve in diagnostic test assessment, J. Thorac. Oncol. 5 (9) (2010) 1315–1316, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1556086415306043.