



Detecting anomalies within smart buildings using do-it-yourself internet of things

Yasar Majib¹ · Mahmoud Barhamgi² · Behzad Momahed Heravi³ · Sharadha Kariyawasam³ · Charith Perera¹

Received: 31 December 2021 / Accepted: 30 July 2022 / Published online: 24 September 2022
© The Author(s) 2022

Abstract

Detecting anomalies at the time of happening is vital in environments like buildings and homes to identify potential cyber-attacks. This paper discussed the various mechanisms to detect anomalies as soon as they occur. We shed light on crucial considerations when building machine learning models. We constructed and gathered data from multiple self-build (DIY) IoT devices with different in-situ sensors and found effective ways to find the point, contextual and combine anomalies. We also discussed several challenges and potential solutions when dealing with sensing devices that produce data at different sampling rates and how we need to pre-process them in machine learning models. This paper also looks at the pros and cons of extracting sub-datasets based on environmental conditions.

Keywords Anomaly detection · Machine learning · Internet of things · Smart buildings

1 Introduction

An anomaly is something unexpected, abnormal or distanced from the ordinary. From a technology perspective, an anomaly results from equipment malfunction, cyber or physical intrusion, financial fraud (e.g. credit card usage by hackers), terrorist activity, and an abrupt change detected by sensors in the physical environment due to an accident. Following are the types of anomalies:

1. **Point Anomalies:** A single sample, different from normal samples. For example, a credit card (CC) transaction with an amount much larger than the CC holder's routine transactions.
2. **Collective anomalies:** A sample is a collection of several data points considered anomalous if it differs from other samples. For example, an electrocardiogram (ECG) is a collection of readings of the heart's activity over a specific period as one data sample.
3. **Contextual anomalies:** If a sample is contextually different from normal samples. Time is the context in time-

series data considering a situation where data is streaming from sensors. An anomalous sample depends on a set of time-series values, e.g. a temperature trend of the last 30 min showing 20 °C increases 50% abruptly. In some other time (context) 30°C is considered normal temperature.

Our work looked into all the above types of anomalies in our dataset. We proposed multiple solutions to look for abnormalities in various contexts, e.g. time-series, multivariate, and inter-device sensor combinations. The high-level idea behind anomaly detection is to (i) save resources by finding faults in systems in advance, (ii) respond to events as early as possible (iii) deal with security breaches. Equipment with the least latency from sensors is microcontrollers, and these devices are resource-constrained. With the rapidly growing IoT domain, there are a few off-the-shelf microcontrollers available now (Sudharsan et al. 2021) which support machine learning (ML) on edge using libraries, e.g. TensorFlow. Detecting anomalies as soon as they occur can help save a building from various challenges. Gas leakage by equipment malfunction or pipeline cracks, discomfort due to a sudden change in environment (temperature, humidity, noise, air quality, and others), infrastructure damage, physical access at a non-working time, or unauthorised personnel cyber-physical attacks related. Detecting anomalies at the edge ensures early response and reduces the risk of it getting ignored by the central system in case

✉ Yasar Majib
MajibY@Cardiff.ac.uk

¹ Cardiff University, Cardiff, UK

² Qatar University, Doha, Qatar

³ VortexIoT, Neath, UK

of unavailability of network connectivity due to technical problems or cyber-attacks, e.g. Daniel of Service (DoS). We collected data from self-built physical devices with 32 data streams from 14 unique sensors. We have combined intra-device data streams and inter-devices unique sensors' streams. Other than the original "unconditional" dataset, we applied two (02) environmental conditions to the data set, then applied data preprocessing (scaling and reduction) techniques to each resulting data set and then used different ML algorithms. We tested all models using both normal and anomaly data sets and presented the results in HTML format at [GitLab/CyPhyRadar](#). We evaluated the models based on computational time vs the number of detected anomalies.

1.1 Contributions

- Impact of environmental conditions' based data set in anomaly detection
- Pros and cons of conventional (scaling/reduction) and unconventional (atan) data preprocessing methods
- Comparison of different ML techniques
- Relations between various sensors in the context of discovering anomalies in building
- Best practices to transform univariate data into time-series format
- Handling missing data and synchronizing data streams from different devices

2 Anomaly detection within smart buildings

It is not energy-saving anymore; it is about the overall resilience of smart buildings, which is the next big challenge. Smart buildings require mechanisms to mitigate or prevent fire, gas leakages, attacks, disasters, accidents, safety and security-related issues, and other unforeseen challenges. Secondary sensor networks can help mitigate such events by observing physical channels such as external eyes and ears. Any compromise-able device in a cyber network can allow attackers to gain control over the complete building management systems (Schiffer 2017).

2.1 Data collection setup

We have implemented a sensing network consisting of various 14 different environmental sensors, Arduino based microcontrollers and RaspberryPi (RPi) microprocessors, as shown in Table 1. The sensor reads the environmental changes and transfers readings to the attached RPi, directly or through a microcontroller, which then transforms and/or transfers these values to the ingestor using unique Message Queuing Telemetry Transport (MQTT) channels. The data set consists of 32 different data streams from eight (8) device

sets, i.e. sensor-Arduino-RPi (DSet). Temperature, humidity sensors and some other associated data streams were duplicated in two device-sets; although both device-sets were at the same place, one of the DSet's sensors was influenced by a nearby heat source. Thus, the readings are different in these data streams. Timestamp and other properties were added to every new entry by the ingestor before inserting it into the data set. The probability of BLE and WiFi devices in the area was also calculator by the ingestor after receiving collective BLE and WiFi devices' information from all other physical devices; these data streams in channels `ble_devices` and `wifi_devices` were considered as virtual devices. Figure 1 shows the overall architecture of data collection setup, processing points, devices' and channels' names. We divided the data sets from July 24, 2020, to January 7, 2021, and from March 26, 2021, to July 16, 2021, into two subsets, normal and abnormal, respectively. Both data sets were captured during normal routine operations, and some naturally occurring unusual activities were recorded in the time-frames of both data sets. We used the normal subset for training and testing machine learning models, whereas the anomaly subset was for testing purposes only.

- Physical devices = 8
- Virtual devices = 2
- Environmental conditions = 3
- Pre-processing techniques = 8
- Data streams (total) = 32
- Intra-device combinations = 626
- Data streams (unique sensors) = 14
- Inter-device combinations of unique sensors = 16383
- Machine learning techniques = 4

2.2 Data collection challenges

Some of the main challenges in data collection are:

- Time synchronisation, microcontrollers do not come with an internal time clock, making it tricky to keep data synchronised from different host devices, assuming the reporting time between each device is different.
- Handling heterogeneous data types, contexts and formats
- Low-resolution sensors, e.g., some generate integer values for reading instead of floating-point values, e.g., temperature value 22 instead of 22.0–22.9.
- Some sensors generate arbitrary data, which is very difficult to detect and troubleshoot on edge.
- Dual channel sensors like temperature-humidity have sensing errors in either of the channels creating difficulty to troubleshoot on edge.
- Different communication mediums have different latency, which is also a challenge in time synchronisation.

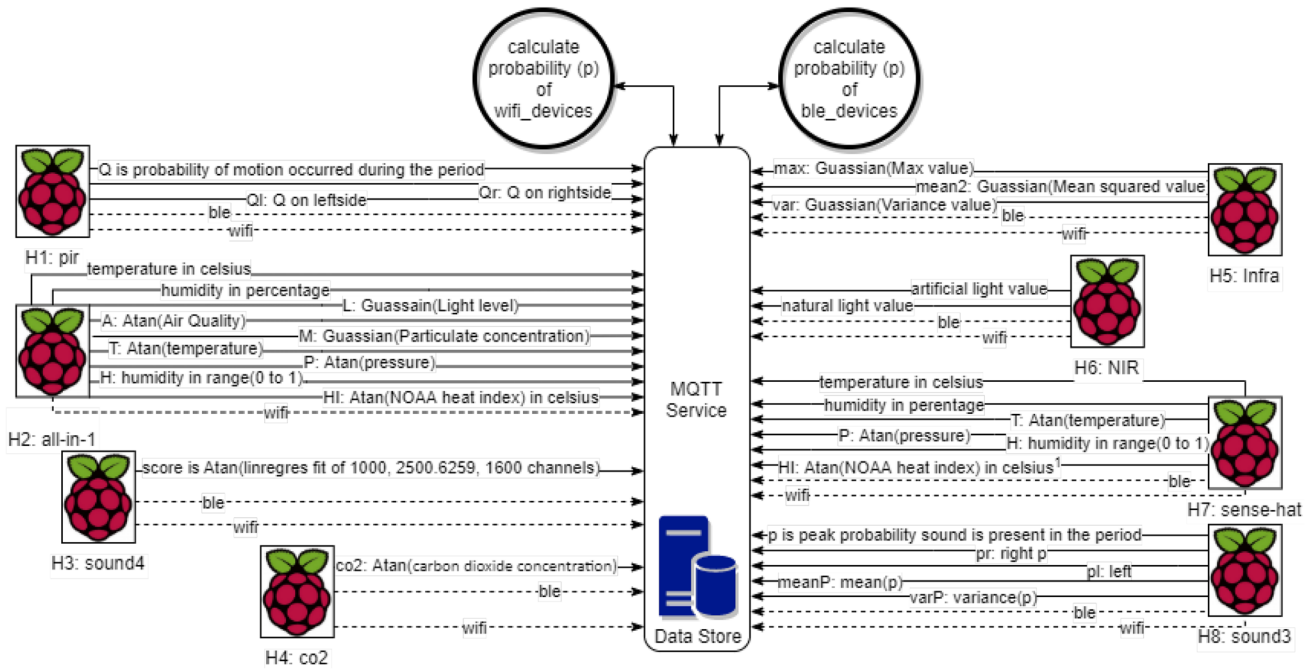


Fig. 1 H1: passive InfraRed, H2: all-in-1 multi-sensor, H3: Sound4, H4: carbon dioxide, H5: infra-sound, H6: light, H7: sense-hat multi-sensor, H8: Sound3

- Communication modules provide limited access to the chip via AT Commands.
- Skipped or missed part of data at random times due to equipment malfunction, network connectivity, electric power or other issues.

2.3 Data cleaning and normalisation

We pre-processed the data sets before performing ML-associated operations to save time and computational resources. There were various possible combinations of errors in data sets like null, non-numeric, or irrelevant values when capturing data due to sensor malfunctions or ingestion processing. We removed all rows with null values, converted the date and time into a DataFrame supported format, changed the type data type of all other values to integer or float, and normalised data sets.

2.4 Data streams overview and analysis

Analysing all data streams, individually and jointly, is very important before applying operations. Analysis helps in getting a better understanding of data streams and helps in estimating which pre-processing technique with which type of model should be used to do further processing. The best way to visualise data streams is by graphs; we used interact-able graphs using Plotly-library to better understand the data streams from all sensors. We joined data streams

from all devices to better understand the relations between each combination. Moreover, the Table 1 hosts details of all individual data streams with description, host device, MQTT topic, edge-processing technique (Process), minimum value, maximum value, average, standard deviation (SD), and median absolute deviation (MAD).

2.4.1 Single data streams

Figure 2 hold visualization of some of the unique data streams. We structured sub-figures as a 1 × 2 matrix where the left side (1×) graph shows all data and the right side (2×) graph shows one-day activity. The left side graph of Fig. 2(A1) and (B1) that there is a sudden dip in temperature and increased humidity near the end of October 2020 till the end of December 2020. We also observe that Air Quality is dropping abruptly at the same time. Though these events resulted from disconnection and/or power failure on the device, both were considered anomalous and kept in the data sets; we will discuss other aspects later in the paper. In Fig. 2(E2) and (F2), we observed that the 24 h trend of artificial light, and natural is identical except a few activities of artificial light can be found in the nighttime. The light sensor in the all-in-1 device, Fig. 2(H1) and (H2), share similar trends. It is noticeable that natural light trends are gradual compared to artificial light. We also noticed that activities related to Sound, Light, CO₂, infra, BLE devices and particulate concentration are stable and low-valued at

Table 1 Data stream details

Name	Description	Data Stream Properties				Normal Data				Anomaly Data			
		Sensor Device	Topic	Process		Min	Max	SD	MAD	Min	Max	Average	SD
temp	temperature sensor	Sense-HAT	sense-hat	None	3.1	26.7	22.523	1.92738	18.1	30	21.88942	1.49556	1.63086
humidity	humidity sensor	Sense-HAT	sense-hat	None	32.4	58.3	38.4451	2.07564	30.5	43.8	37.35104	2.17766	2.37216
T	Atan transformed temperature	Sense-HAT	sense-hat	Atan	0.01965	0.94034	0.72561	0.09736	0.11915	0.96171	0.636083	0.22574	0.29127
P	Atan transformed pressure	Sense-HAT	sense-hat	Atan	0.13972	0.93009	0.83703	0.03882	0.5869	0.93243	0.878565	0.05589	0.02667
H	humidity scaled 0-1	Sense-HAT	sense-hat	Scale	0.324	0.583	0.38445	0.01887	0.305	0.438	0.37351	0.02177	0.02872
HI	Atan transformed NOAA heat index	Sense-HAT	sense-hat	Atan	0.01757	0.93836	0.65749	0.28259	0.08645	0.96135	0.511415	0.07952	0.37915
max	Gaussian transformed maximum value	Infiltec Model INFRA20	infra	Gaussian	4E-10	0.99966	0.00338	0.03547	2.8E-06	4E-10	0.842429	0.020099	0.02009
mean2	Gaussian transformed mean squared value	Infiltec Model INFRA20	infra	None	7.8E-12	0.99999	0.00443	0.02985	5.4E-06	5.8E-12	0.9163	0.002814	0.0249
var	Gaussian transformed variance value	Infiltec Model INFRA20	infra	None	1.8E-05	0.99859	0.03748	0.00334	1.9E-05	0.99997	0.037592	0.00251	0.00238
C	Atan transformed CO2 concentration	DfRobot SEN0219	co2	None	0.15119	0.85903	0.2504	0.09549	0.18182	0.8339	0.27269	0.09757	0.04537
natural	Probability of natural light	SparkFun AS7263	nir	None	0	0.9382	0.03925	0.14193	0	0	0.88205	0.18547	0
artificial	Probability of artificial light	SparkFun AS7263	nir	None	0	0.9382	0.03925	0.14193	0	0	0.88205	0.18547	0
D	Peak probability of presence of sound	NA	sound3	None	0	1	0.24323	0.4551	0	1	0.99736	0.45155	0
Pr	Peak probability of presence of sound on right side	NA	sound3	None	0	1	0.25626	0.3752	0	1	0.262058	0.40104	0.02377
pl	Peak probability of presence of sound on left side	NA	sound3	None	0	1	0.24323	0.39295	0	1	0.278179	0.37982	0.03816
meanP	Mean probability of presence of sound	NA	sound3	None	0	1	0.15991	0.33738	0.00017	0	1	0.158854	0.33858
varP	Variance of probability of presence of sound	NA	sound3	None	0	1	0.1694	0.36446	1.2E-14	0	1	0.179248	0.37252
score	Atan transformed of lmregres fit of L6K channels	NA	sound4	None	0.24387	0.60479	0.51678	0.02923	0.00835	0.24875	0.57387	0.02663	0.00509
Qr	Probability of motion on right side	Passive InfraRed	pir	None	0.029	0.545	0.06509	0.07245	0.00148	0.03	0.541	0.063051	0.06494
Ql	Probability of motion on left side	Passive InfraRed	pir	None	0.017	0.27	0.03582	0.03701	0.00148	0.018	0.292	0.034289	0.03283
temperature	temperature sensor	Sense-HAT	all-in-1	None	0.012	0.3	0.02927	0.03617	0.00148	0.012	0.291	0.028704	0.03283
humidity	humidity sensor	Sense-HAT	all-in-1	None	7.7	28.3	24.7295	1.56751	1.4826	7.6	27.4	24.65208	1.26133
L	Gaussian transformed light level	Sense-HAT	all-in-1	None	7.3	21	13.5195	2.62001	2.52042	5.3	18.8	12.10845	3.01125
A	Atan transformed of air quality	Sense-HAT	all-in-1	None	0	0.64147	0.23047	0.37456	0.00032	0	0.99995	0.229474	0.34466
M	Gaussian transformed of particulate concentration	Sense-HAT	all-in-1	None	0	0.80237	0.02732	0.07666	0.00415	0	0.80128	0.74926	0.01184
T	Atan transformed temperature	Sense-HAT	all-in-1	None	0.02646	0.95303	0.89234	0.06747	0.0296	0.02627	0.94664	0.016666	0.0842
P	Atan transformed pressure	Sense-HAT	all-in-1	None	0.01005	0.92438	0.80709	0.15813	0.04238	0.01005	0.92627	0.856712	0.07473
HI	Atan transformed heat index	Sense-HAT	all-in-1	None	0.02183	0.94771	0.82037	0.15561	0.02164	0.9383	0.84311	0.08332	0.07736
H	humidity scaled 0-1	Sense-HAT	all-in-1	None	0.073	0.21	0.1352	0.0262	0.052	0.53	0.188	0.121084	0.03011
P	Probability of BLE devices	All	ble_devices	None	0	0.98806	0.38963	0.17093	0.12864	0	0.99996	0.494271	0.26898
P	Probability of WiFi devices	All	wifi_devices	None	0	0.99989	0.37118	0.20366	0.15017	0	0.655721	0.23514	0.27778

night time. Thus we decided to filter data sets based on daylight conditions as well. We also observe a regular (not everyday) activity before the start of daylight time; this issue has consequences which will be discussed later in the paper.

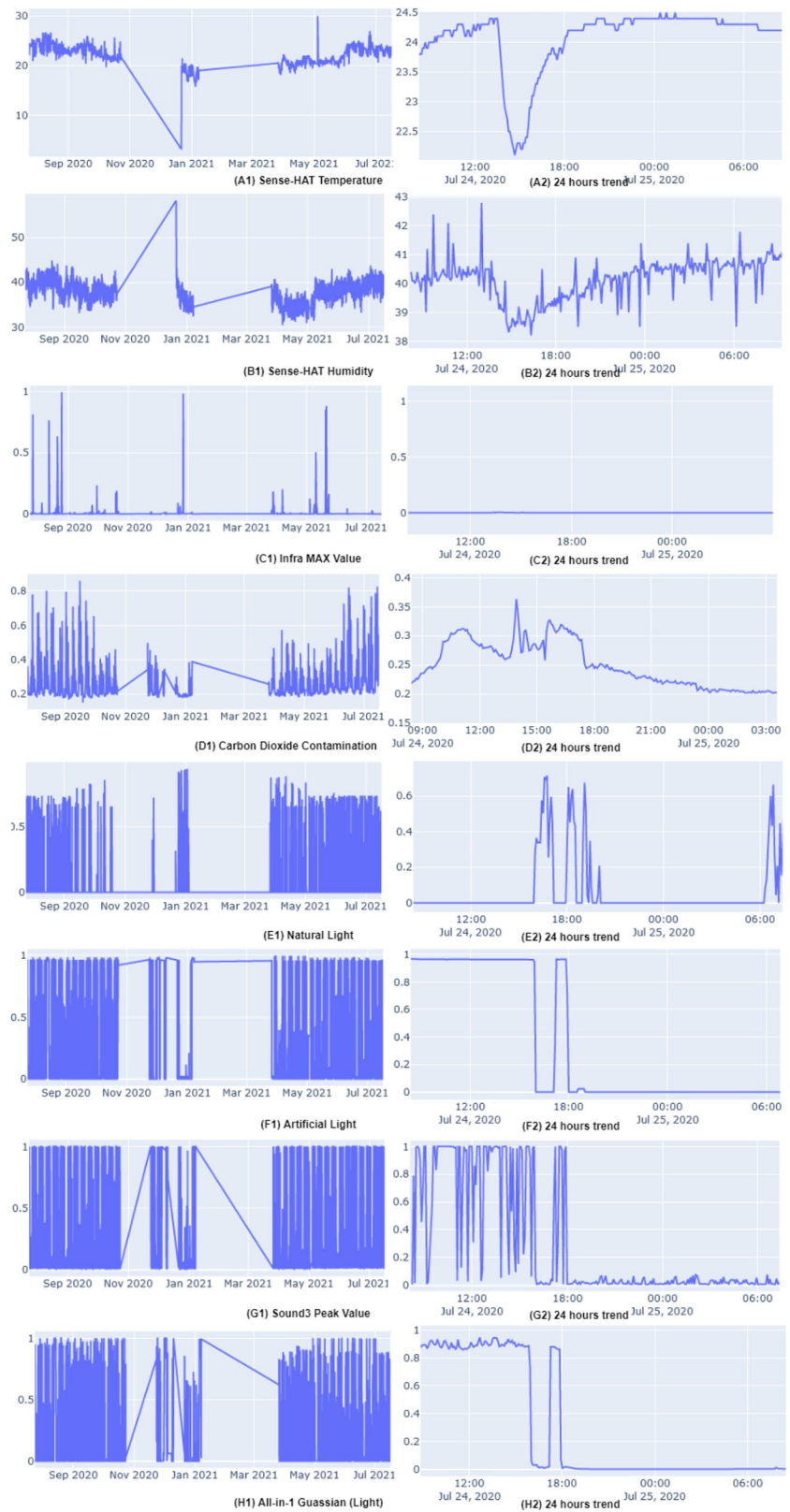
2.4.2 Multi data streams

Analysing relations between different data streams is difficult, ineffective and time-consuming when done separately. So we visualised multiple data streams to analyse the relations demonstrated in Fig. 3. For example, in Fig. 3(A1), it can straightforwardly be noticed that the values of temperature and humidity go opposite directions around the end of October 2020 till the end of December 2020. We can also notice the relation between natural and artificial light in Fig. 3(B1) and (B2). There are two possible types of multi-data streams in the given setup, intra-device and inter-device. Visualising multiple data streams from one device is comparatively easy as there are a limited number of combinations. On the other hand, inter-device data stream combinations can be enormous, so we chose only (14) unique sensors' data streams, see **bold** items in Table 1. We choose a couple of inter-device combination graphs for demonstration which can be seen in Fig. 3(C1) and (D1). Figure 3(D2) has a different situation plotted in which a fire alarm went off at night time and visited by a staff member to evaluate the situation, which triggered the light in the room as seen in the red circle. This activity is perfect to be considered a contextual anomaly. From the left side graph, we can see a regular activity of sound and light in the daytime. Later in this paper, we will evaluate ML models by considering two things (i) the regular activity detected as an anomaly, and (ii) the sound and light activity around 2100 h is considered an anomaly.

2.5 Data scaling and reduction techniques

The machine learns from the provided data instead of legacy statistical or mathematical algorithms in the ML context. It makes pre-processing of data sets an essential part of the process. Data standardisation is being largely practised for pre-processing data sets before performing ML. It drastically decreases the size of the input sample (in some cases) and time for a model generation compared to non-scaled data. We adopted two techniques for standardisation, StandardScaler and MinMaxScaler. Standardisation can only convert data into a certain range and can be reversed but can not reduce the dimensions of the input sample in the case of multivariate data. So, we used reduction techniques to convert multivariate data into univariate. Reduction techniques help in reducing ML model generation time to a minimum. The resulting data sample from reduction techniques is computationally expensive

Fig. 2 Single data streams



to reverse. Which makes it hide properties of individual data streams or sensor values, e.g. value of temperature and humidity can only be known by the edge device but

will be kept unknown by the fog or cloud device. Scaling techniques are feasible on cloud/fog where a complete data set is available to evaluate a given ML model. We

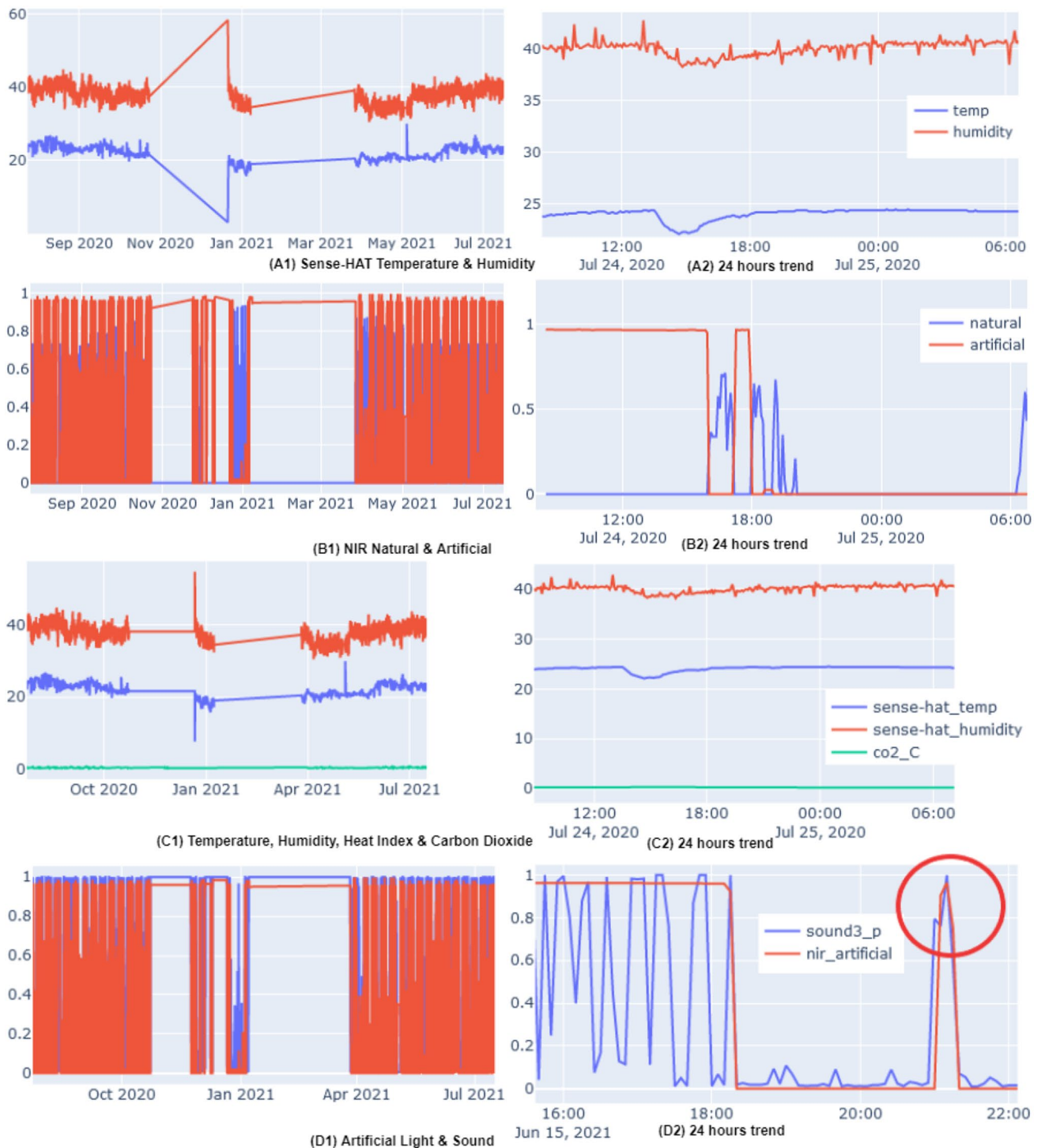


Fig. 3 Multi data streams

did not consider data scaling for ML models destined to run on edge devices (microcontrollers). We added another dimension to data sets after applying pre-processing techniques to convert the data into time series, and the resulting sample was three-dimensional. We used two scaling

techniques and five reduction techniques on the available data to evaluate the time difference for model generation. We experienced that scaling techniques take less time (a few microseconds) versus reduction techniques which takes 1500–2127 μ s to execute the process.

2.5.1 Scaling techniques

We used the following data scaling techniques for this work. *Standard Scaler* calculate the mean and standard deviation of the input sample before applying Eq. 1. In Eq. 1 *SSd* is the standard scaler output sample of input sample *d*, *u* is equal to the mean of sample *d* and *s* is equal to the standard deviation of input sample *d*.

$$SSd = \frac{(d - u)}{s} \tag{1}$$

The resulting output sample has a mean = 0 and standard deviation = 1. We used the *StandardScaler* function from the *sklearn* library to perform this scaling operation.

MinMax Scaler is simpler than *StandardScaler*, there is no pre-calculation required as compared to *StandardScaler*, and most frequently used for input sample standardisation. The output sample is in the range of 0–1. The corresponding output value of the minimum value in the sample will be 0, and the corresponding output value of the maximum value in the sample will be 1. These values are calculated using the Eq. 2. We used the *MinMaxScaler* function from the *sklearn* library to perform this scaling operation.

$$MMd = \frac{(d - d_{min})}{(d_{max} - d_{min})} \tag{2}$$

In Eq. 2 *MMd* is the *MinMax* scaler output sample of input *d*, *d*(min) is the minimum value in input sample *d* and *d*(max) is the maximum value in input sample *d*.

2.5.2 Reduction techniques

We used the following data reduction techniques for this paper.

Average is the sum of all values divided by the number of values resulting in a single value for each sample. Average can reflect the central tendency of multiple data streams while converting the input sample into univariate. Average requires the least processing resources as compared to other pre-processing techniques. We used the *average* function from the *NumPy* library to execute this operation on the multi-variate input samples.

$$\bar{m} = \left(\frac{1}{n}\right) \sum_{i=1}^n x_i \tag{3}$$

Standard deviation (SD) results in a univariate data stream that can reflect the spread of a multivariate input sample. It takes slightly more processing resources than average as the average input sample is a prerequisite for the SD equation

to be executed. We used the *std* function from the *NumPy* library to execute this operation on multi-variate input samples.

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}} \tag{4}$$

Median absolute deviation (MAD) calculates variability in the input sample, it is more computationally complex than SD because it is dependent on the median value of the input sample. MAD is more resilient in terms of outlier detection as compared to SD. We used the *median_abs_deviation* function from *scipy.stats* library for this operation.

$$MAD = median(x_i - \bar{x}) \tag{5}$$

Kurtosis (Ku) calculates the relative peakedness of an input sample, it requires both average and SD of the input sample thus the computational power requires is more than the previous techniques. We noticed that *Ku* is effective on larger data points in terms of influencing anomaly detection. We used *stats.kurtosis* function from *scipy* library for this operation.

$$K = \frac{1}{n} \sum_{i=1}^n \frac{(x_i - \bar{x})^4}{\sigma^4} \tag{6}$$

Skewness (Skew) calculates the trends of the input sample, it can be a normal, negative or positive skewness value. *Skew* is the most computationally complex in our discussed techniques, it requires precomputed average and SD of the input sample. It is also effective on larger data points where a curve can be formed. We used *stats.skew* function from *scipy* library for this operation.

$$Sk = \frac{1}{n} \sum_{i=1}^n \frac{(x_i - \bar{x})^3}{\sigma^3} \tag{7}$$

2.6 Data conversion to time series

We tried and compared different algorithms to convert series data in a time-series format, i.e. each row contains the number of future rows. In streaming data scenarios, anomalies are categorised based on data trends instead of points, e.g. the temperature in daytime hits 30 °. In contrast, at night time, it remains below 18 °. Considering a microcontroller without an internal clock can only be aware of the context be current values rather than time. The ML model shall be trained using a time-series-based input sample to achieve this functionally. Let us say the dimensions of the input sample are [Rows, data points], e.g. [36,484, 14], dimensions of the resulting sample become [Rows, Time Steps, data points], e.g. [36,484, 74,

14]. Let us say R represents data rows in the data set, T represents the number of required time-steps for each sample, X represents the use-able rows, and Y is the resulting time-series sample.

$$\begin{aligned} X &\in \{R0, R1, R2, \dots, R - T\} \\ Y &\in \{X + 1, X + 2, X + 3 \dots, X + T\} \end{aligned} \quad (8)$$

2.7 Anomaly detection techniques selection

We used the following anomaly detection techniques in this paper.

2.7.1 OneClassSVM (OSSVM)

Support vector machine (SVM) is one of the most common ML methods (Djenouri et al. 2019). SVM is primarily used for classification (supervised ML) but can also be adopted for clustering (unsupervised ML). SVM is memory efficient, flexible, and suitable in high dimensional spaces and even works with a smaller number of samples compared to dimensions. It has a sub-method, OneClass for outlier-detection, that tries to discover decision boundaries to achieve maximum distance between data points and source by using a clustering mechanism. The main idea behind OneClass was stalled because of its incompetence in finding outliers and determining non-linear decision boundaries. However, with the introduction of soft margins and kernels, these issues were resolved (Amer et al. 2013). OneClass SVM splits all given data points from the source and amplifies the distance from this subspace to the source in the training phase. The function returns a binary output for each input row where +1 means smaller distance and -1 means larger distance where larger distance considers an anomaly (Schölkopf et al. 2000). It is widely used in various applications for both supervised and unsupervised learning methods. It is also heavily adopted in academia. An anomaly classifier using SVM was proposed (Araya et al. 2017) for detecting abnormal consumption behaviour. A method proposed by Ferdoash et al. (2015) to calculate excessive airflow in Heating Ventilation and Air Conditioning (HVAC) units in a large-scale Building Management System (BMS). They also calculated the pre-cooling start time for reaching the required temperature using temperature sensors. Jakkula and Cook (2011) the proposes OneClass SVM for anomaly detection in smart home environments using publicly available smart environment data sets. Himeur et al. (2021a) proposed a method to detect anomalous power consumption in buildings. OCSVM is highly effective on

point anomalies and can be inferred on fog devices to be used in real-time environments.

2.7.2 Isolation forest (IF)

IF is one of the top-most used algorithms in the outlier detection domain because of its speed and simplicity. IF is based on ensemble learning. The idea behind IF is that randomly developed decision trees can quickly isolate an outlier in the data set instead of detecting outliers using density or distance from other samples. Outliers are isolated because of the shorter path in the tree as they have fewer relations with other data points (Liu et al. 2008). In terms of functional performance in outlier detection, IF is the most popular algorithm (Buschjager et al. 2020). We use the IsolationForest function from the SKLearn library to perform model generation. The function requires all samples as input and return a list of anomaly score for each sample. IF is also effective for point anomalies only. It is not suitable for fog devices in real-time scenarios as it requires a complete dataset.

2.7.3 CNN

In Deep Neural Networks (DNN), Convolutional Neural Network (CNN) is on the most wanted neural networks list. The name ‘‘Convolutional’’ comes from the matrixes-based linear operation. CNN models consist of multiple layers, e.g. max-pooling, fully-connected, and others (Albawi et al. 2018). It brings significant improvement in computer vision (CV), Time series prediction and Natural Language Processing (NLP). It covers a wide range of application scenarios by providing single and multidimensional layers, i.e. 1-D CNN, supporting Time Series Prediction and Signal Identification. 2-D CNN enables Image Classification, Object Detection, Image Segmentation and Face Recognition and 3-D CNN, which helps in Human Action Recognition and Object Recognition/Detection (Li et al. 2021b). In contrast with other classification approaches, e.g. feature-based, CNN can find and learn relations and generate in-depth features from time-series data streams automatically, e.g. speech recognition, ECG, price stocks, pattern recognition, rule discovery, and many more (Zhao et al. 2017). All platforms support CNN, i.e. Edge (microcontrollers), Fog (RaspberryPi, Mobile Platforms) and Cloud (High-performance Linux, Windows or Other OSes). We implemented CNN by using TensorFlow API.

2.7.4 RNN

A recurrent Neural Network (RNN) is also a type of DNN, and it is designed with built-in memory, making it more suitable for time-series-based data streams. Another feature of RNN is that it can process information in bi-directional

instead of forwarding direction only. Typical RNN has a known issue of vanishing or exploding gradient, which affects its accuracy and overall performance. With the help of Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997), which is designed with a memory cell to hold information over a period of time, this problem can be resolved. LSTM is complex but sophisticated, and has three gates input, output and forget. RNN models can predict the future value from time-based input compared with the data sample to calculate the loss. If the loss is greater than the threshold (pre-computed using the training sample), the data sample can be categorised as an anomaly. LSTM is widely used in various applications commonly based on time-series data. LSTM is available only on Fog and Cloud devices using the TensorFlow library. Anomaly detection in a time-series context is a significant application of LSTM.

3 Experimentation results

This section will discuss the results of different combinations of data pre-processing and ML models. We tested selective TF models on all platforms (Cloud-Fog-Edge) and SKLearn models on Cloud and Fog only. SKLearn models predictions are binary (Anomaly = - 1, Normal = 1) whereas TF models were based on future prediction, so the output was non-binary. Results for TF models were calculated using a two steps process. First, we calculated the Mean Absolute Error (MAE) for the predicted loss method using Eq. 9 and threshold by using Eq. 10.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y - x| \tag{9}$$

The equatrefeq:mae calculates the mean absolute error (average loss) of all input samples by calculating absolute loss for each sample, where n represents a number of samples, y represents predicted and x represents expected values of each sample.

$$Threshold = (8 \times \sigma(MAE)) + MAE \tag{10}$$

$\sigma - > StandardDeviation$

Equation 10 dynamically calculates the threshold by calculating the standard deviation of MAE, multiplying it eight times and adding it with MAE. If the resulting loss of an input sample is greater than the threshold, the sample is considered anomalous.

3.1 Architectural configurations

As discussed previously, we are using four types of ML Models to train and test available data sets. These models are from two different APIs, Sci-kit Learn (SKLearn)

and TensorFlow (TF). SKLearn and RNN based models are available on Cloud and Fog platforms, whereas CNN is also deployable on edge devices. In this section, we will discuss the configurations of each algorithm. We configure the OCSVM model with 0.5 nu, “auto” gamma and “RBF” kernel parameters. We configure IF model for “auto” contamination parameter. Early-Stopping to monitor loss with min_delta = 1e-2 and patience = 3 was configured for both CNN and RNN models. We converted the dataset for both NN models into 74-time steps. We also fixed 100 epochs (max), adam optimizer, and batch size to be 10 for both NN models. Our CNN model requires TensorFlow version 2.1.1 and RNN on the 2.4.1 version. We configured CNN models with Conv1D layer, kernel size of 32, 5 filters and mean-squared error for loss calculation. We used LSTM layers for RNN models with 32 neurons and mean-absolute-error for loss calculation.

3.2 Data streams’ configurations

We divided our data sets into two sub-datasets depending on daylight conditions, e.g., day time sub-dataset (DT) and night time sub-dataset (NT). We used unconditional data set (UC) for ML models as well. We implemented these scenarios on these two types of streams. Converting datasets into sub-datasets reduces the ML model generation time as well as inference time. It also supports (in some cases) the implementation of point-based anomaly detection, e.g. illumination. Events at nighttime can be detected with high accuracy and low computational resources if the ML model is trained using the NT sub-dataset. On the other hand, sub-datasets are limited to specific circumstances only, e.g. if the buildings are designed to be illuminated 24x7.

i. Univariate (Single Data Streams): each data stream from all devices was used to train, test, and analyse models. Because these Data Streams were already uni-variate, reduction techniques were not applicable. ii. Multivariate (Multiple Data Stream): There can be enormous possible combinations between intra-device and inter-device data streams. Research has already been conducted about relations between physical channels like temperature-humidity with CO₂Liu et al. (2017). Showing all possible combinations of multi-data streams is overwhelming; thus, we have presented results of a few of these combinations and preserved all models and results stored for detailed analysis.

3.3 Results

3.3.1 Univariate vs multivariate

Reduction techniques returns univariate data so the model training time is identical for all number of data stream combinations. Total training time also depends on the number

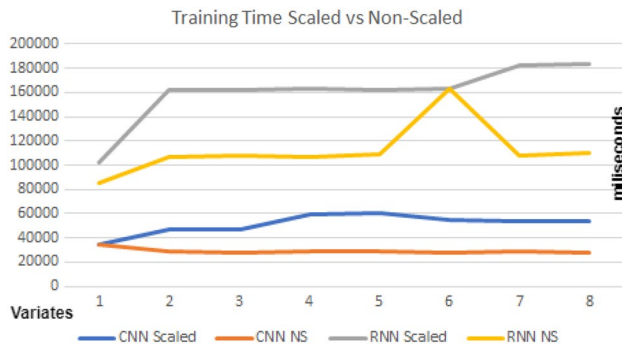


Fig. 4 Scaled vs non-scaled and RNN vs CNN model training times

of epochs executed before early stopping condition becomes true. Figure 4 shows model training times of scaled vs non-scaled dataset, it can be observed that scaled dataset took more time for training in both CNN and RNN methods. It is also obvious to see that RNN CNN is efficient when compared to RNN. Due to limited knowledge of known anomalies in the dataset, it is difficult to determine overall efficiency of ML models.

3.3.2 Detecting anomalies using individual sensor data streams (univariate)

A comparison of temperature with edge-processed T data streams, which is atan (temperature), from the sense-hat device. We had 32 data streams, out of which 14 were from unique sensors, and 18 were associated streams. While comparing different sensor and associated data streams, we found that atan converted data streams required a lesser threshold value to find anomalies in novel data. The transformed data streams were ineffective at certain stages where change suddenly fluctuated. As seen in circled in blue colour where anomalies are shown in orange dots in Fig. 5, a

few anomalies found in T, all at a lower temperature, was not detected in the temperature model can be seen in green circles. When it comes to humidity, the edge-processed scaled data stream H was less sensitive as compared to the unprocessed data stream, as demonstrated in Fig. 6, the blue circles highlight the difference. Since we generated models for three environmental conditions, we found that the sum of anomalies found in two daylight condition-based data sets (dark=0, light=1) was equal to the number of anomalies found in the unconditional data set.

We also noticed that there is no difference in non-scaled streams vs scaled streams in temperature and its associated data streams, e.g. T. Whereas other sensors and associated data streams show different results, e.g. a number of anomalies found original data stream of humidity sensor were noticeably different from StandardScaler but comparatively similar with MinMax. We observed that StandardScaler decreases sensitivity resulting in lesser anomalies as compared to the non-scaled data stream. It was also observed that MinMaxScaler increased sensitivity resulting in more anomalies. We found an obvious difference when comparing a number of anomalies in pressure (P) and particulate concentration (M) data streams where StandardScaler results in drastically increased sensitivity, the number of anomalies are greater using a smaller threshold level. On the other hand, anomalies found in carbon dioxide (CO₂) in scaled versions of data streams were fewer as compared to non-scaled data stream based models, which point toward a decrease in sensitivity. Another noticeable trend in the number of anomalies is that the sum of both conditional anomalies was marginally greater than the unconditional data set except for standard scaler based models. We found a unique trend in artificial sensor condition-based models. No anomalies were found in non-scaled and MinMax scaler models in conditional data sets, but standard scaled models found

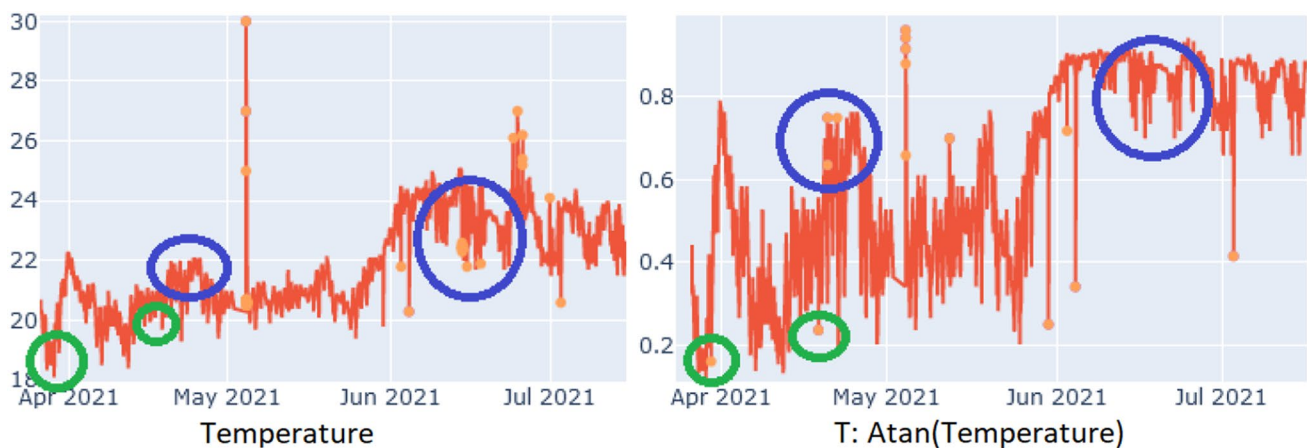


Fig. 5 Temperature vs atan (temperature) comparison

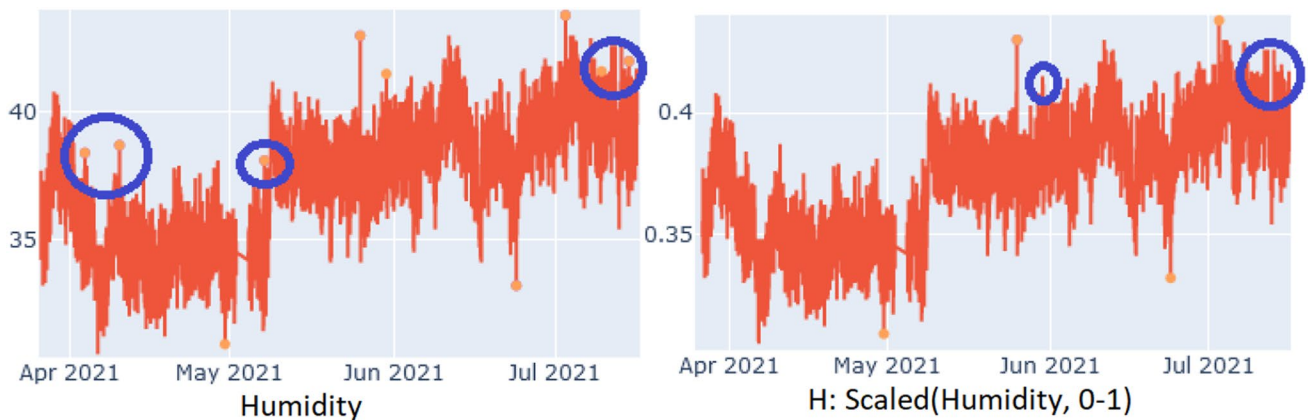


Fig. 6 Humidity vs percentage (humidity) comparison

anomalies. Anomalies found in unconditional data set based models were similar to non-scaled and scaled models. Sound sensor-based models show an opposite reaction when it comes to anomalies; we found zero anomalies in UC and DT. Whereas NT based models found anomalies, non-scaled and MinMaxScaler models were pretty much similar. However, the StandardScaler model found more anomalies that represent increased sensitivity similar to previously discussed pressure and particulate concentration models.

3.3.3 Detecting anomalies using intra-device (multivariate)

The total number of unique intra-device combinations of data streams was 626. We choose a few of them for analysis in this paper. We noticed that most of the data preprocessing techniques could find almost similar anomalies in the sense-hat device (all data streams), except MinMaxScaler, which was extremely sensitive, and MAD was too insensitive. Kurtosis and Skewness were not effective. Zero anomalies were found when implemented on the temperature and humidity (Temp-Humidity) set. The behaviour of MinMaxScaler was the same in Temp-Humidity but turns regular when used on all other associated streams, i.e., T, P, H and HI (T-P-H-HI) MAD were also able to find the same contextual anomalies on this set. When looking at the results of all data streams in All-in-1, we found that MAD was most sensitive on UC and most insensitive on DT (zero anomalies). The average was not effective (a few anomalies detected) on NT and UC, whereas it could find the same contextual anomalies as other techniques. We noticed that temperature sensor readings were regularly dipping randomly and abruptly, which was one of the reasons for its influence over other data streams and thus on statistical outcomes. Looking at other models

in all-in-1 devices, excluding temperature-related values, we found few anomalous activities.

3.3.4 Detecting anomalies using inter-device multiple data streams (multivariate)

As discussed in an earlier section about the one known anomalous activity based on sound and light sensors' data, we analyzed the particular activity to learn the effectiveness of different algorithms and pre-processing techniques. We found that the CNN model with scaled, non-scaled and average sound and artificial values can spot the anomalous activity without spotting false positives (usual everyday activity). In contrast, RNN models were not successful in detecting the particular activity, as shown in Fig. 7. We also noticed that false positives were found in all models, along with detecting anomalous activity in the NT dataset. We also found that SKLearn based models overwhelmed false positives in all datasets.

3.3.5 Point, contextual, combined anomalies

Looking closely at Fig. 8, the two highlighted portions of the timeline of the temperature data stream from the sense-hat device. We observed at the end of April 2021 temperature sensor malfunctioned, resulting in an extreme increase to 30 °. Another event marked anomalous in highlighted point 2 shows a sudden dip in temperature from 22.6 to 22.9 ° detected. While looking at historical data, both points are in the normal range, but this activity is considered anomalous in context. Figure 9 shows the combined activity of artificial light and sound for the week commencing on June 14, 2021. In the context, office activity started early, i.e. at 0530 h on Monday, Tuesday, and Thursday and was detected as anomalous True Positive (TP). The office starts at 0700 h on Friday and Wednesday, as shown in the black circle. The Friday

Fig. 7 Sound & light known “anomalous activity” analysis

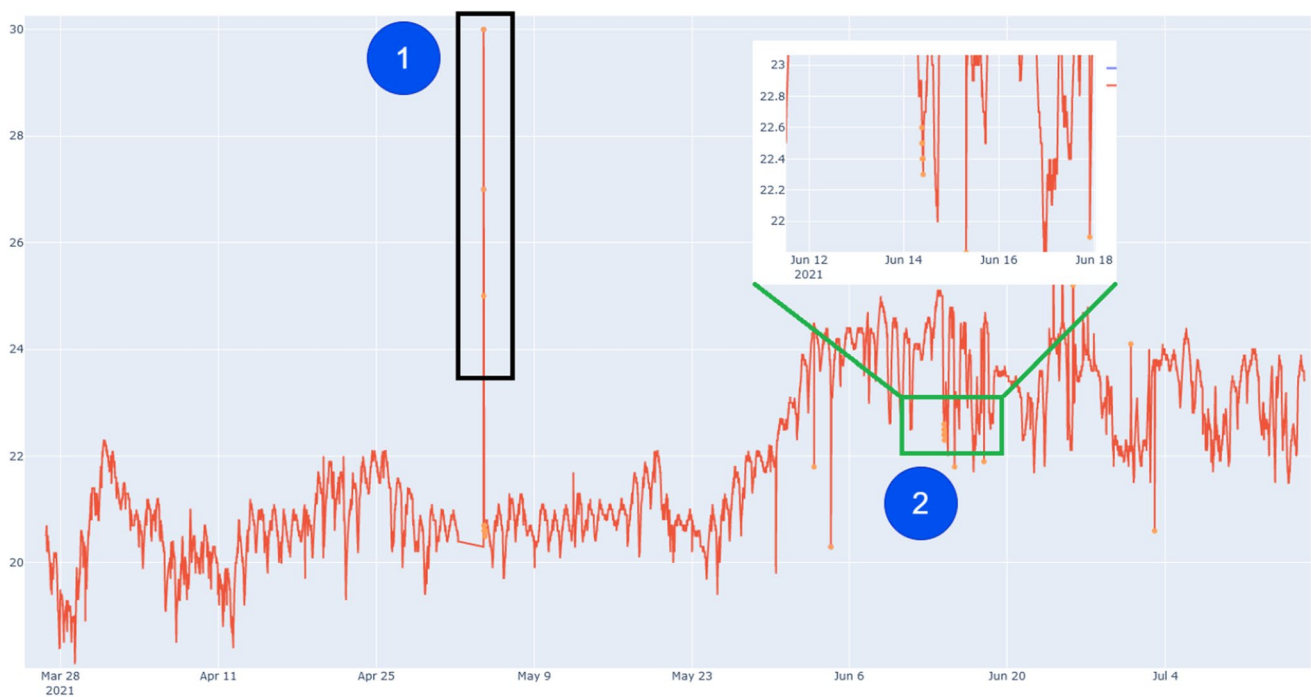
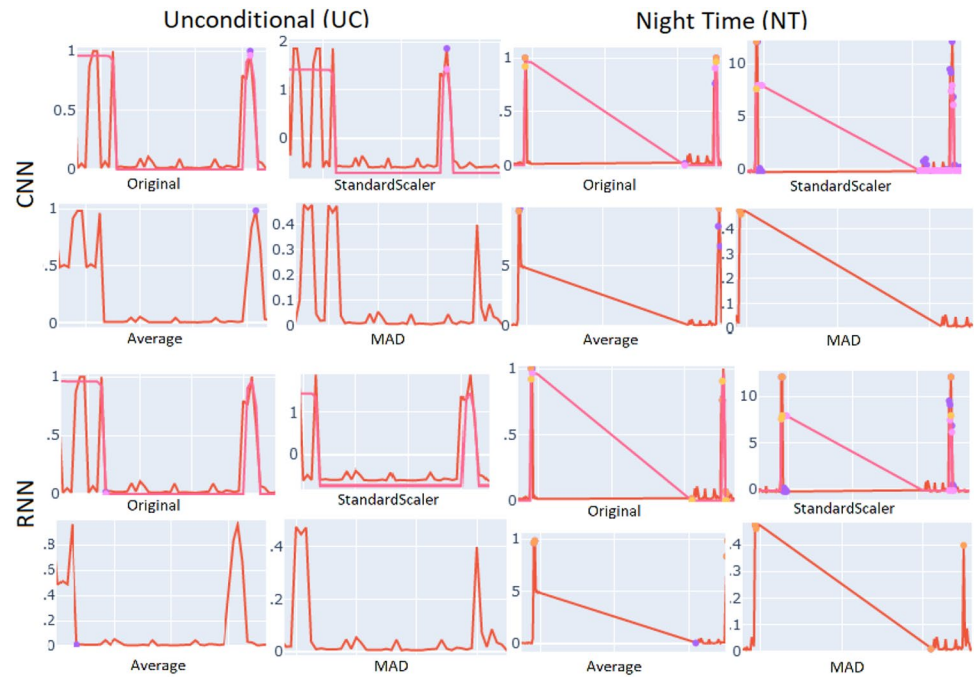


Fig. 8 1-Point anomaly vs 2-contextual anomaly in temperature data stream

morning activity was detected as False Positive (FP). On the other hand, the Wednesday activity was accurately detected as True Negative (TN). In addition to day start activities, a TP anomaly was detected around 2100 h due to a response initiated as a result of a (separately operated) fire alarm.

4 Related work

There are some suggestions for supervised anomaly detection methods (Liu et al. 2015; Laptev et al. 2015). The results are promising, but labelled data is rare in the real

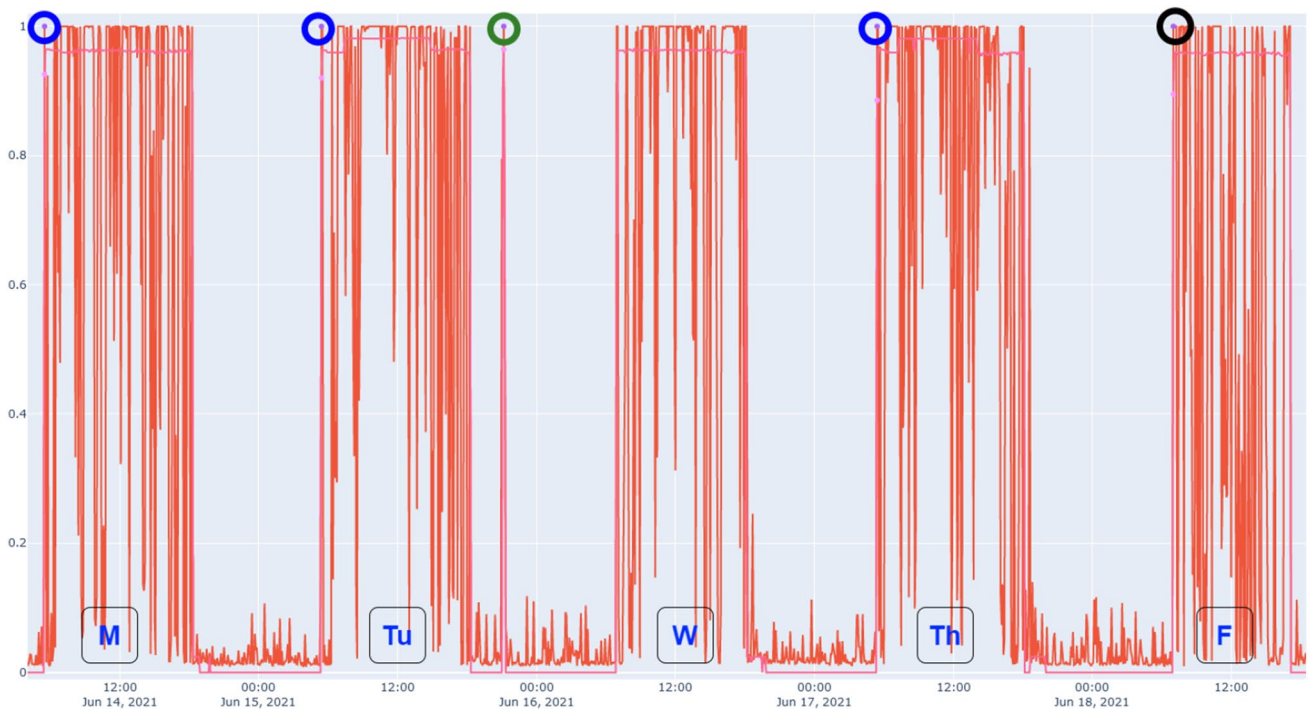


Fig. 9 Combined contextual anomalies in sound and artificial light data streams

world. Perhaps unsupervised ML methods have become the focus of attention because of the excellent performance and the flexibility provided (Li et al. 2021a). The scope of anomaly detection is not limited to specific areas. However, everywhere e.g., industry (Oh Dong and Yun Il 2018), financial systems (Gran and Veiga 2010), health-care and maintenance of spacecraft by detecting anomalies (Gupta et al. 2014), cyber-physical system (Luo et al. 2021), and smart buildings (Araya et al. 2016).

4.1 Anomaly detection techniques for IoT data

Research conducted by Microsoft (Ren et al. 2019) led to the development of an algorithm for detecting anomalies in time-series data using residual spectrum processing and convolutional neural networks (SR-CNN). However, they were mainly concerned about stationary and seasonal data, resulting in ineffective results on non-stationary data. Data from Surface-mounted audio sensors used with semi-supervised CNN auto-encoders (Oh Dong and Yun Il 2018) to detect faults in industrial machinery. A deep autoencoders based model has been proposed for detecting spectrum anomalies in wireless communications (Feng et al. 2017). The model developed in this work is to detect anomalies that may occur due to an abrupt change in the signal-to-noise ratio (SNR) of the monitored communications channel. In a critical infrastructure environment, if phasor data is manipulated, the control centres may take the wrong actions, negatively

impacting power transmission reliability. To mitigate this threat (Yan and Yu 2015) proposed a deep autoencoder technique. Zhang et al. (2018) study uses data from a number of heterogeneous IIoT sensors, including temperature, pressure, vibration, and others, to develop an RNN-LSTM based regression model to predict failures in pumps at a power station. A new RNN-LSTM based method was developed (Hundman et al. 2018) to detect anomalies in a massive amount of telemetry data from spacecraft. They also offered a method for evaluating that was non-parametric, dynamic, and unsupervised. Another solution proposed (Wu et al. 2020) to detect anomalies in multi-seasonality time-series data using RNN-GRU also proposed a Local Trend Inconsistency metric on top of their proposed anomaly detection algorithm. The authors of (Martí et al. 2015) proposed a combination of Yet Another Segmentation Algorithm (YASA) and One-ClassSVM (OCSVM) in order to detect anomalous activities in turbomachines in the petroleum industry. The authors of (Aurino et al. 2014) used OCSVM to detect gunshots from audio signals. OCSVM grouped with DNN used to detect road traffic activities by Rovetta et al. (2020). Isolation Forest (IF) was used to detect anomalies in smart audio sensors (Antonini et al. 2018). IF is also used, in combination with order-preserving hashing techniques, to detect anomalies by Xiang et al. (2020). Another novel approach proposed by Farzad and Gulliver T (2020) uses autoencoder based IF for log-based anomaly detection.

4.2 Environmental monitoring within buildings

In today's world, human beings spend 90% of their time in built environments which includes residential, commercial, education, as well as transport, i.e. vehicles, Brady (2021). Monitoring an indoor environment is different from industrial or mission-critical infrastructure, where normal activities are largely known because of the heterogeneous nature of activities. There are several environmental monitoring applications other than anomaly detection, e.g. Energy Monitoring, Comfort Level Monitoring. Environment monitoring is well researched. The heterogeneous nature of environments requires the selection of the suitable parameters, sensors technologies, communication mediums, placement and power arrangements. Major parameters in this domain are temperature, humidity, carbon emissions, illumination, airflow, and occupancy (Hayat et al. 2019). Air Quality (AQ) is becoming a critical matter. WHO reported that there are almost 7 million premature deaths are being caused by air pollution annually (WHO 2021). Authors of Saini et al. (2020) presented a survey of system architectures used for Indoor Air Quality (IAQ) data collection as well as methods and applications for prediction. Indoor environment quality plays an essential role in the health and well-being of human beings, Clements et al. (2019) presented a living lab to simulate real office spaces to support further research on environmental monitoring in the built environment. Occupancy monitoring is essential to determine air-conditioning and illumination requirements in buildings, Erickson et al. (2014) proposed a wireless sensor network based occupancy model to be integrated with buildings conditioning systems. Based on two seasons of monitoring IAQ and thermal comforts in school building (Asif and Zeeshan 2020) recorded more than 50% increase in CO₂ levels during class times. Thermal comfort has critical importance for the well-being and productivity of occupants in indoor environments, Valinejadshoubi et al. (2021) proposed an integrated sensor-based thermal comfort monitoring system for buildings which also provides the virtual visualization of thermal conditions in buildings. Authors of Nasaruddin et al. (2019) presented temperature and relative humidity monitoring solutions in high temperature and humid climate environments using well-calibrated thermal micro-climate devices and a single-board microcontroller.

4.3 Anomaly detection within buildings

Researchers propose a wide variety of methods for anomaly detection in buildings. The diversity of techniques reflects extensive work being done in this domain. Unsupervised learning has been used for fault detection and diagnostics in smart buildings. Authors of Capozzoli et al. (2015) proposed a simple technique based on unsupervised learning that can

automatically detect anomalies in energy consumption based on the historically recorded data of active lighting power and total active power. They adopt statistical pattern recognition and ANN along-with other anomaly detection methods. A novel method, Strip, Bind, and Search (SBS), based on unsupervised learning proposed by Fontugne et al. (2013) to help identify devices with anomalous behaviour by looking at inter-device relationships. The authors of Yizhe et al. (2021) also proposed a data mining based unsupervised learning technique to detect anomalies in HVAC systems; the proposed work also performs dynamic energy performance evaluation. In the models proposed by Araya et al. (2017), overlapping sliding windows and ensemble anomaly detection were used to identify anomalies. The same authors also proposed a Collective Contextual Anomaly detection using similar techniques in their previous work (Araya et al. 2016). A Generalized Additive Model was proposed by Ploennigs et al. (2013) for diagnosing building problems based on the hierarchy of sub-meters. A Two-Step clustering algorithm based on unsupervised machine learning was proposed by Poh et al. (2020) to detect anomalous behaviour from physical access data of employees about their job profiles. In a distributed sensor network, an anomaly detection technique was proposed by Meyn et al. (2009) using semi-empirical Markov Models for time-series data. In a recent survey conducted by Himeur et al. (2021b), the authors concluded that anomaly detection techniques could help in the reduction of energy consumption to benefit all stakeholders.

5 Lessons learnt and discussion

DIY based (single-board computers, microcontrollers, sensors) IoT devices are widely available and becoming easy to deploy. These devices are micro-manageable and cost-effective, but it is a laborious job which leads to various challenges; while doing this research, we learnt the following lessons: (i) missing data due to run-time errors, (ii) threshold calculation, (iii) inter-device synchronisation, (iv) importance of "normal" dataset, (v) an overwhelming number of ML models, (vi) converting time-series data for unsupervised ML processing and (vii) handling interactive graphs.

Missing data DIY devices are prone to configuration, deployment, and handling problems when used for capturing data on a long-term basis. There is no built-in notification system that can alert in case of any error; thus, the errors persist silently for an extended period, ultimately affecting the dataset. During our data-capturing stage, we faced various scenarios where data collection stopped, e.g. device power outage, sensor malfunctions, communication errors, etc. thus; the data is missing during those time slots.

Threshold calculation Anomaly decision in time-series data using an unsupervised approach is based on loss and

threshold. The threshold is critical in the decision process and calculating the threshold for each configuration (data stream combinations with sub-datasets). A maximum loss value from a normal dataset (training dataset) can be used as a threshold; to achieve that, an utterly normal dataset (without any capture-time errors) is required.

Inter-device synchronisation Due to multiple device setups, there were synchronisation errors due to missed data in devices at different time slots. Data lost from any single device or frequency differences can result in synchronisation issues. This creates a unique challenge when combining data streams from inter-device. It is recommended to use a single host device for all sensors or create a master table with a single timestamp at the ingestor-end to keep data synchronised at capturing stage.

Importance of “normal” dataset For the above-learned lessons, we observe the critical importance of a completely normal dataset, e.g. without run-time errors (communication, power, hardware).

An overwhelming number of ML models Due to the number of data streams, the number of combinations was in the thousands. The resulting ML models and associated results were overwhelming and difficult to observe and manage. A systematic approach needed to be adopted to handle the heterogeneous configuration of datasets, models, and results.

Converting time-series data for unsupervised ML processing Time-series conversion of data sets using pandas data-frames is far more computationally expensive than using the NumPy library. It is wise to test and compare all available methods for each sub-task before starting mass processing. The result is the same for both methods.

Handling interactive graphs For unsupervised learning approaches for time series, analysing data using interactive graphs is vital but requires extensive computational resources to load and interact graphs with multiple data streams.

6 Conclusion and future work

In this paper, we captured data streams from various in-situ sensors using different devices with a variety of configurations. We were able to detect point, contextual and combined anomalies. We compared different ML methods combined with several data pre-processing techniques to better understand how to efficiently detect anomalous activities in a smart building environment. We also evaluated the performance of the conditional dataset (based on environmental conditions, e.g. daylight). We found that it can work better for detecting point anomalies as the activities are filtered for certain situations. A clean, anomaly-free dataset is required for model training for better results. Unconventional scaling techniques, e.g., atan, can lower sensitivity for detection

and an overhead during the data-capturing process; atan and other conversions can be performed in bulk at any later stage with reasonable computational resources. We explored relations between various sensors in finding anomalies in buildings. We also explored effective techniques to pre-process datasets to optimise ML models. We also introduced an inter-device data synchronisation technique to fill up missing time slots and trim time-series datasets when comparing different datasets. Threshold plays a vital role in reducing false positives and increasing true positives. A dynamic threshold calculation is essential to deal with the overwhelming configuration of data streams. The day of the week can also be used as a context for anomaly detection in time-series datasets, but a large dataset is required for modelling. Availability of a dataset with known anomalies will be an important step towards determining overall efficiency of ML models.

Acknowledgements This work is partially supported by EPSRC PET-RAS (EP/S035362/1) and GCHQ National Resilience Fellowship.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Alex S (2017) How a fish tank helped hack a casino
- Amer M, Goldstein M, Abdennadher S (2013) Enhancing one-class Support Vector Machines for unsupervised anomaly detection. In: Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description, ODD 2013, p 8–15
- Antonini M, Vecchio M, Antonelli F, Ducange P, Perera C (2018) Smart audio sensors in the internet of things edge for anomaly detection. *IEEE Access* 6:67594–67610
- Araya Daniel B, Grolinger K, Elyamany Hany F, Capretz Miriam AM, Bitsuamlak G (2016) Collective contextual anomaly detection framework for smart buildings. In: Proceedings of the international joint conference on neural networks, 2016-October, p 511–518
- 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
- Asif A, Zeeshan M (2020) Indoor temperature, relative humidity and CO₂ monitoring and air exchange rates simulation utilizing system dynamics tools for naturally ventilated classrooms. *Build Environ* 180(January):106980
- Aurino F, Folla M, Gargiulo F, Moscato V, Picariello A, Sansone C (2014) One-class SVM based approach for detecting anomalous audio events. In: Proceedings—2014 international conference on

- intelligent networking and collaborative systems, IEEE INCoS 2014, p 145–151
- Brady C (2021) RATING TOOLS FOR RESILIENCE-UNDRR and the World Green Building Council
- Buschjäger S, Honysz PJ, Morik K (2020) Generalized isolation forest: some theory and more applications extended abstract. In: Proceedings—2020 IEEE 7th international conference on data science and advanced analytics, DSAA 2020, vol 2, no 4, p 793–794
- Capozzoli A, Lauro F, Khan I (2015) Fault detection analysis using data mining techniques for a cluster of smart office buildings. *Expert Syst Appl* 42(9):4324–4338
- Clements N, Zhang R, Jamrozik A, Campanella C, Bauer B (2019) The spatial and temporal variability of the indoor environmental quality during three simulated office studies at a living lab. *Buildings* 9(3)
- Djenouri D, Laidi R, Djenouri Y, Balasingham I (2019) Machine learning for smart building applications: review and taxonomy. *ACM Comput Surv* 52(2)
- Erickson VL, Carreira-Perpiñán MÁ, Cerpa AE (2014) Occupancy modeling and prediction for building energy management. *ACM Trans Sens Netw* 10(3)
- Farzad A, Aaron Gulliver T (2020) Unsupervised log message anomaly detection. *ICT Express* 6(3):229–237
- Feng Q, Zhang Y, Li C, Dou Z, Wang J (2017) Anomaly detection of spectrum in wireless communication via deep auto-encoders. *J Supercomput* 73(7):3161–3178
- Ferdoash A, Saini S, Khurana J, Singh A (2015) Poster abstract: analytics driven operational efficiency in HVAC systems. In: *BuildSys 2015—proceedings of the 2nd ACM international conference on embedded systems for energy-efficient built*, pp 107–108
- 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: *IPSN 2013—proceedings of the 12th international conference on information processing in sensor networks*, part of CPSWeek 2013, p 129–140
- Gran A, Veiga H (2010) Wavelet-based detection of outliers in financial time series. *Comput Stat Data Anal* 54(11):2580–2593
- Gupta M, Gao J, Aggarwal CC, Han J (2014) Outlier detection for temporal data: a survey. *IEEE Trans Knowl Data Eng* 26(9):2250–2267
- Hayat H, Griffiths T, Brennan D, Lewis RP, Barclay M, Weirman C, Philip B, Searle JR (2019) The state-of-the-art of sensors and environmental monitoring technologies in buildings. *Sensors (Switzerland)* 19(17)
- Himeur Y, Alsalemi A, Bensaali F, Amira A (2021a) Smart power consumption abnormality detection in buildings using micromoments and improved K-nearest neighbors. *Int J Intell Syst* August 2020:2865–2894
- Himeur Y, Ghanem K, Alsalemi A, Bensaali F, Amira A (2021b) Artificial intelligence based anomaly detection of energy consumption in buildings: a review, current trends and new perspectives. *Appl Energy* 287(April):1–41
- Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
- Hundman K, Constantinou V, Laporte C, Colwell I, Soderstrom T (2018) Detecting spacecraft anomalies using LSTMs and non-parametric dynamic thresholding. In: *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining*, p 387–395
- Jakkula V, Cook DJ (2011) Detecting anomalous sensor events in smart home data for enhancing the living experience. In: *AAAI Workshop—technical report, WS-11-07 (December 2014)*: 33–37
- Laptev N, Amizadeh S, Flint I (2015) Generic and scalable framework for automated time-series anomaly detection. In: *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining*, 2015-August, p 1939–1947
- Li J, Di S, Shen Y, Chen L (2021a) FluxEV: a fast and effective unsupervised framework for time-series anomaly detection. In: *WSDM 2021—proceedings of the 14th ACM international conference on web search and data mining*, p 824–832
- Li Z, Liu F, Yang W, Peng S, Zhou J (2021b) A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE Trans Neural Netw Learn Syst*, p 1–21
- Liu D, Zhao Y, Xu H, Sun Y, Pei D, Luo J, Jing X, Feng M (2015) Opprentice : towards practical and automatic anomaly detection through machine learning categories and subject descriptors. In: *ACM internet measurement conference*
- Liu FT, Ting KM, Zhou ZH (2008) Isolation forest. In: *Proceedings—IEEE international conference on data mining, ICDM*, p 413–422
- Liu X, Mai E, Xu X, Noh HY, Zhang L, Chen X, Zhang P (2017) Poster abstract: individualized calibration of industrial-grade gas sensors in air quality sensing system. In: *SenSys 2017—proceedings of the 15th ACM conference on embedded networked sensor systems*, 5–6 January 2017
- Luo Y, Xiao Y, Cheng L, Peng G, Yao DD (2021) Deep learning-based anomaly detection in cyber-physical systems: progress and opportunities. In: *ACM computing surveys*, 54(5)
- Martí L, Sanchez-Pi N, Molina JM, Garcia Ana CB (2015) Anomaly detection based on sensor data in petroleum industry applications. *Sensors (Switzerland)* 15(2):2774–2797
- Meyn S, Surana A, Lin Y, Narayanan S (2009) Anomaly detection using projective Markov models in a distributed sensor network. In: *Proceedings of the IEEE conference on decision and control*, p 4662–4669
- Nasaruddin AN, Tee BT, Musthafah MT, Jasman MESM (2019) Ambient data analytic on indoor environment monitoring for office buildings in hot and humid climates. *Data in Brief* 27
- Oh Dong Y, Yun Il D (2018) Residual error based anomaly detection using auto-encoder in SMD machine sound. *Sensors (Switzerland)* 18(5):1–14
- Ploennigs J, Chen B, Schumann A, Brady N (2013) Exploiting generalized additive models for diagnosing abnormal energy use in buildings. In: *BuildSys 2013—Proceedings of the 5th ACM workshop on embedded systems for energy-efficient buildings*
- Poh JP, Lee Jun YC, Tan KX, Tan E (2020) Physical access log analysis: an unsupervised clustering approach for anomaly detection. In: *ACM international conference proceeding series*
- Ren H, Xu B, Wang Y, Yi C, Huang C, Kou X, Xing T, Yang M, Tong J, Zhang Q (2019) Time-series anomaly detection service at Microsoft. In: *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining*, 3330680(c), p 3009–3017
- Rovetta S, Mnasri Z, Masulli F (2020) Detection of hazardous road events from audio streams: an ensemble outlier detection approach. In: *IEEE conference on evolving and adaptive intelligent systems*, 2020-May
- Saad A, Abed MT, Saad A (2018) Understanding of a convolutional neural network. In: *Proceedings of 2017 international conference on engineering and technology, ICET 2017*, 1–6 January 2018
- Saini J, Dutta M, Marques G (2020) Indoor air quality prediction systems for smart environments: a systematic review. *J Ambient Intell Smart Environ* 12(5):433–453
- Schölkopf B, Williamson R, Smola A, Shawe-Taylor J, Platt J (2000) Support vector method for novelty detection. *Adv Neural Inf Process Syst*, p 582–588
- Sudharsan B, Salerno S, Nguyen DD, Yahya M, Wahid A, Yadav P, Breslin JG, Ali MI (2021) TinyML benchmark: executing fully connected neural networks on commodity microcontrollers. In:

- 7th IEEE world forum on internet of things, WF-IoT 2021, p 883–884
- Valinejadshoubi M, Moselhi O, Bagchi A, Salem A (2021) Development of an IoT and BIM-based automated alert system for thermal comfort monitoring in buildings. *Sustain Cities Soc* 66 (November 2020): 102602
- WHO (2021) Air pollution is one of the biggest environmental threats to human health, alongside climate change
- Wu W, He L, Lin W, Su Y, Cui Y, Maple C, Jarvis SA (2020) Developing an unsupervised real-time anomaly detection scheme for time series with multi-seasonality. *IEEE Trans Knowl Data Eng* 4347(c):1
- Xiang H, Salcic Z, Dou W, Xu X, Qi L, Zhang X (2020) OPHiForest: order preserving hashing based isolation forest for robust and scalable anomaly detection. In: International conference on information and knowledge management, proceedings, p 1655–1664
- Yizhe X, Yan C, Shi J, Zefeng L, Niu X, Jiang Y, Zhu F (2021) An anomaly detection and dynamic energy performance evaluation method for HVAC systems based on data mining. *Sustain Energy Technol Assess* 44(February), p 101092
- Yan W, Yu L (2015) On accurate and reliable anomaly detection for gas turbine combustors: a deep learning approach. In: Proceedings of the annual conference of the prognostics and health management society, PHM, p 440–447
- Zhang W, Guo W, Liu X, Liu Y, Zhou J, Li B, Qinghua L, Yang S (2018) LSTM-based analysis of industrial IoT equipment. *IEEE Access* 6:23551–23560
- Zhao B, Huanzhang L, Chen S, Liu J, Dongya W (2017) Convolutional neural networks for time series classification. *J Syst Eng Electron* 28(1):162–169

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.