Noname manuscript No.
(will be inserted by the editor)

Detecting Anomalies within Smart Buildings using Do-It-Yourself Internet of Things

Yasar Majib · Mahmoud Barhamgi · Behzad Momahed Heravi · Sharadha Kariyawasam · Charith Perera

the date of receipt and acceptance should be inserted later

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 minutes 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 of unavailability of network

Y. Majib
Cardiff University, UK. E-mail: MajibY@Cardiff.ac.uk

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 compromiseable device in a cyber network can allow attackers to gain control over the complete building management systems Alex 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.

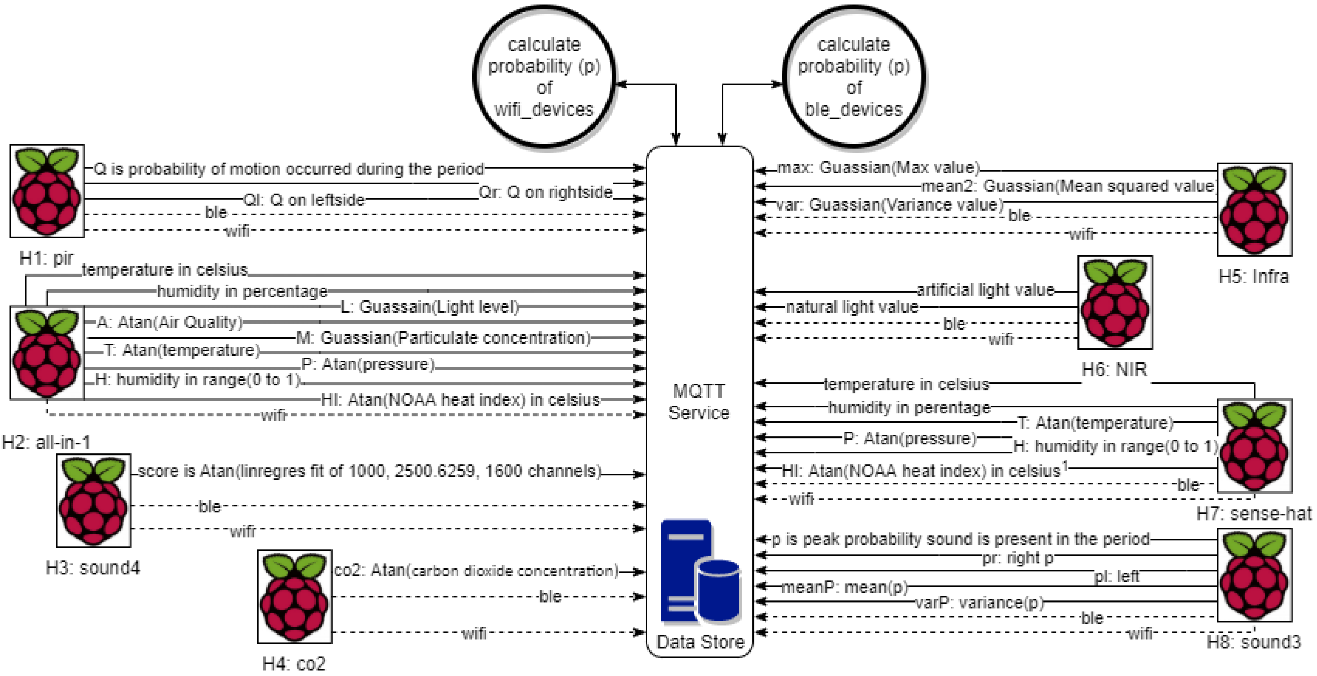
calculate probability (p) of wifi_devices
calculate probability (p) of ble_devices
Q is probability of motion occurred during the period
Ql: Q on leftside
Qr: Q on rightside
ble
wifi
H1: pir
temperature in celsius
humidity in percentage
L: Guassain(Light level)
A: Atan(Air Quality)
M: Guassian(Particulate concentration)
T: Atan(temperature)
P: Atan(pressure)
H: humidity in range(0 to 1)
HI: Atan(NOAA heat index) in celsius
wifi
H2: all-in-1
score is Atan(linregres fit of 1000, 2500.6259, 1600 channels)
ble
wifi
H3: sound4
co2: Atan(carbon dioxide concentration)
ble
wifi
H4: co2
MQTT Service
Data Store
max: Guassian(Max value)
mean2: Guassian(Mean squared value)
var: Guassian(Variance value)
ble
wifi
H5: Infra
artificial light value
natural light value
ble
wifi
H6: NIR
temperature in celsius
humidity in perentage
T: Atan(temperature)
P: Atan(pressure)
H: humidity in range(0 to 1)
HI: Atan(NOAA heat index) in celsius1
ble
wifi
H7: sense-hat
p is peak probability sound is present in the period
pr: right p
pl: left
meanP: mean(p)
varP: variance(p)
ble
wifi
H8: sound3

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

- 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.
- 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

Figures 2hold visualization of some of the unique data streams. We structured Sub-figures as a 1x2 matrix where the left side (x1) graph shows all data and the right side (x2) graph shows one-day activity. The left side graph of figures 2(A1) and 2(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 Figures 2(E2) and 2(F2), we observed that the 24 hours 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, figures 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, CO_2, infra, BLE devices and particulate concentration are stable and low-valued at 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.

Table 1 Data Stream Details

Data Stream Properties					Normal Data					Anomaly Data				
Name	Description	Sensor Device	Topic	Process	Min	Max	Average	SD	MAD	Min	Max	Average	SD	MAD
temp	**temperature sensor**	**Sense-HAT**	**sense-hat**	None	**3.1**	**26.7**	**22.5823**	**2.16122**	**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**	**1.88704**	**2.07564**	**30.5**	**43.8**	**37.35104**	**2.1766**	**2.37216**
T	Atan transformed temperature	Sense-HAT	sense-hat	Atan	0.01965	0.94034	0.72561	0.25949	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**	**0.13184**	**0.03382**	**0.5869**	**0.93243**	**0.878565**	**0.05539**	**0.02667**
H	humidity scaled 0-1	Sense-HAT	sense-hat	Scale	0.324	0.583	0.38445	0.01887	0.02076	0.305	0.438	0.37351	0.02177	0.02372
HI	Atan transformed NOAA heat index	Sense-HAT	sense-hat	Atan	0.01757	0.93836	0.65749	0.28259	0.16325	0.08645	0.96135	0.511415	0.27952	0.37915
max	**Guassian transformed maximum value**	**Infiltec Model INFRA20**	**infra**	Guassain	**4E-10**	**0.9966**	**0.00338**	**0.03547**	**2.8E-06**	**4E-10**	**0.88429**	**0.002099**	**0.02009**	**8.7E-07**
mean2	Guassian transformed mean squared value	Infiltec Model INFRA20	infra	Atan	7.8E-12	0.99999	0.00433	0.03985	5.4E-06	5.8E-12	0.9163	0.002814	0.0249	2.1E-06
var	Guassian transformed variance value	Infiltec Model INFRA20	infra	None	1.8E-05	0.99859	0.03748	0.10021	0.00334	1.9E-05	0.99997	0.037592	0.09251	0.00238
C	**Atan transformed CO2 concentration**	**DFRobot SEN0219**	**co2**	None	**0.15119**	**0.85903**	**0.2504**	**0.09349**	**0.0339**	**0.18182**	**0.82685**	**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**	**0.068757**	**0.18547**	**0**
artificial	**Probability of artificial light**	**SparkFun AS7263**	**nir**	None	**0**	**0.98599**	**0.33197**	**0.4551**	**0**	**0**	**0.99736**	**0.324322**	**0.45155**	**0**
p	**Peak probability of presense of sound**	**NA**	**sound3**	None	**0**	**1**	**0.24323**	**0.39295**	**0.02058**	**0**	**1**	**0.262058**	**0.40104**	**0.02377**
pr	Peak probability of presense of sound on right side	NA	sound3	None	0	1	0.25626	0.3752	0.03195	0	1	0.278179	0.37982	0.03816
pl	Peak probability of presense of sound on left side	NA	sound3	None	0	1	0.24323	0.39295	0.02058	0	1	0.262058	0.40104	0.02377
meanP	Mean probability of presense of sound	NA	sound3	None	0	1	0.15991	0.33798	0.00017	0	1	0.158854	0.33858	0.00032
varP	Variance of probability of presense of sound	NA	sound3	None	0	1	0.1694	0.36446	1.2E-14	0	1	0.179248	0.37252	7.8E-14
score	**Atan transformed of linregres fit of 1.6K channels**	**NA**	**sound4**	None	**0.24367**	**0.60479**	**0.51678**	**0.02923**	**0.00835**	**0.24875**	**0.57387**	**0.51733**	**0.02633**	**0.00509**
Q	**Probability of motion**	**Passive InfraRed**	**pir**	None	**0.029**	**0.545**	**0.06509**	**0.07245**	**0.00148**	**0.03**	**0.541**	**0.063051**	**0.06494**	**0.00148**
Qr	Probability of motion on right side	Passive InfraRed	pir	None	0.017	0.27	0.03582	0.03701	0.00148	0.018	0.292	0.034289	0.03283	0.00148
Ql	Probability of motion on left side	Passive InfraRed	pir	None	0.012	0.3	0.02927	0.03617	0.00148	0.012	0.291	0.028704	0.03283	0.00148
temperature	temperature sensor	Sense-HAT	all-in-1	None	7.7	28.3	24.7235	1.56751	1.4826	7.6	27.4	24.65208	1.26133	1.4826
humidity	humiditiy sensor	Sense-HAT	all-in-1	None	7.3	21	13.5195	2.62001	2.52042	5.3	18.8	12.10845	3.01125	3.11346
L	**Guassian transformed light level**	**Sense-HAT**	**all-in-1**	None	**0**	**1**	**0.23047**	**0.37456**	**0.00032**	**0**	**0.99995**	**0.229474**	**0.34466**	**0.0032**
A	**Atan transformed of air quality**	**Sense-HAT**	**all-in-1**	None	**0.64147**	**0.78394**	**0.74921**	**0.01241**	**0.01239**	**0.65031**	**0.80128**	**0.74926**	**0.01134**	**0.01189**
M	**Guassian transformed of particulate concentration**	**Sense-HAT**	**all-in-1**	None	**0**	**0.80237**	**0.02732**	**0.07666**	**0.00415**	**0**	**0.55781**	**0.016666**	**0.0342**	**0.00567**
T	Atan transformed temperature	Sense-HAT	all-in-1	None	0.02646	0.95303	0.89234	0.06747	0.0296	0.02627	0.94664	0.900589	0.04086	0.03671
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	0.03428
HI	Atan transformed heat index	Sense-HAT	all-in-1	None	0.02183	0.94771	0.82037	0.15561	0.059	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	0.0252	0.053	0.188	0.121084	0.03011	0.03113
p	**Probability of BLE devices**	**All**	**ble_devices**	None	**0.13118**	**0.98806**	**0.38963**	**0.17093**	**0.12864**	**0.13118**	**0.99996**	**0.494271**	**0.26898**	**0.25697**
p	**Probability of WiFi devices**	**All**	**wifi_devices**	None	**0**	**0.99999**	**0.37118**	**0.20366**	**0.15017**	**0**	**1**	**0.655721**	**0.23514**	**0.27778**

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 figure 3. For example, in figure 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 Figures 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 Figures 3(C1) and 3(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 hours is considered an anomaly.

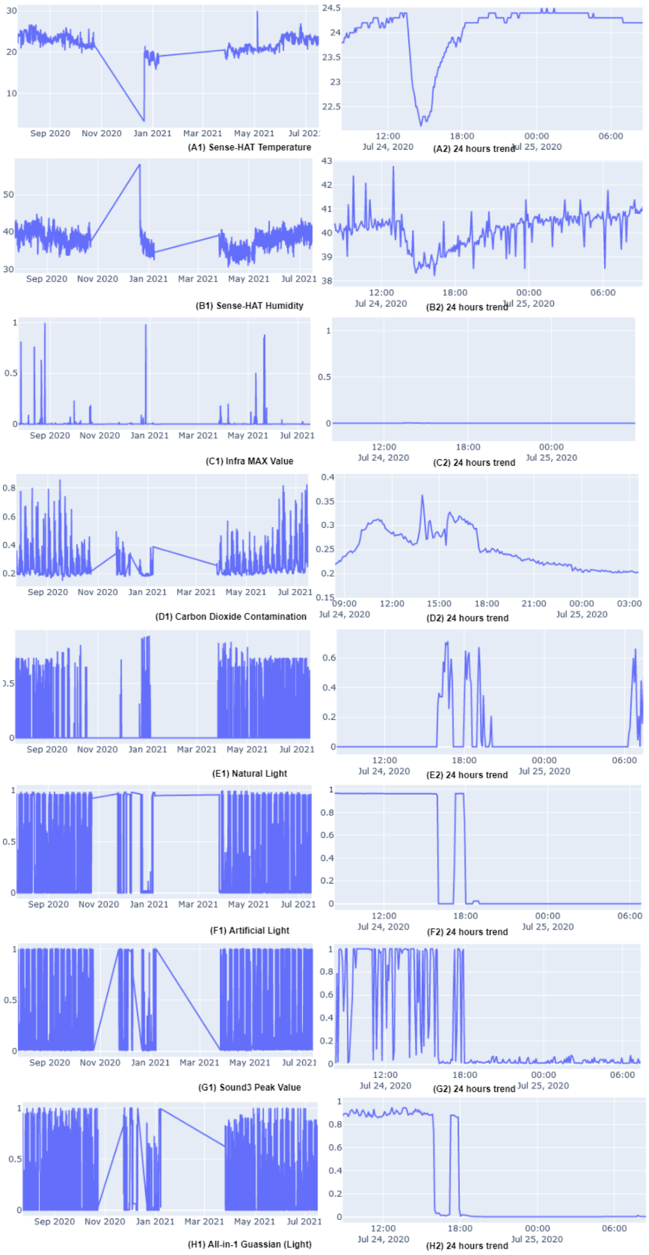

Fig. 2 Single Data Streams

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 techniques 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 uni-variate. Reduction techniques help in reducing ML model generation time to a minimum. The resulting data sample from reduction techniques is computationally expensive 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 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 to 2127 microseconds to execute the process.

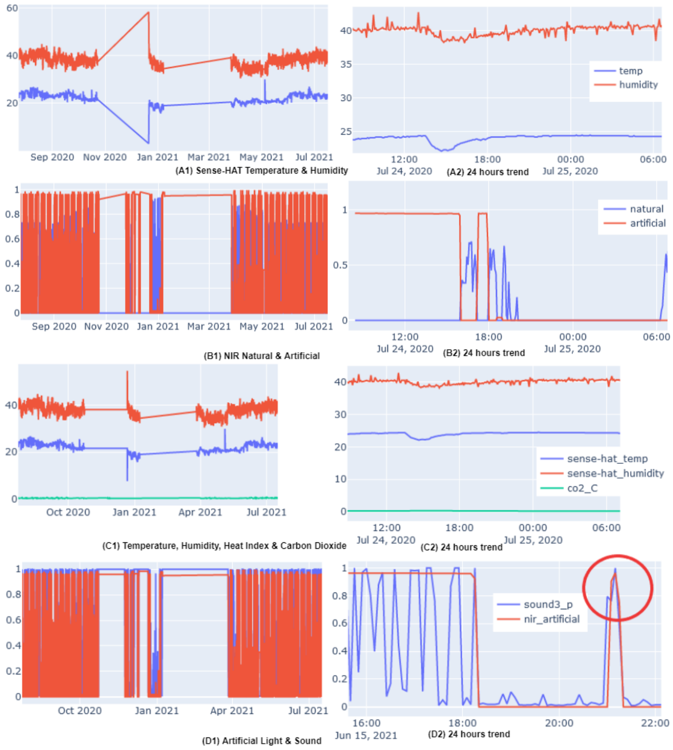

Fig. 3 Multi Data Streams

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 equation 1. In equation 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 to 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 equation 2. We used the MinMaxScaler function from the sklearn library to perform this scaling operation.

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

In equation 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.

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

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.

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

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.

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

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°C. In contrast, at night time, it remains below 18°C. 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. [36484, 14], dimensions of the resulting sample become [Rows, Time Steps, data points], e.g. [36484, 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.

$$\begin{aligned} X \in \{R0, R1, R2, \ldots, R-T\} \\ Y \in \{X+1, X+2, X+3 \ldots, X+T\} \end{aligned} \tag{8}$$

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 submethod, 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 equation 9 and threshold by using equation 10.

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

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.

$$\begin{aligned} Threshold &= (8 \cdot \sigma(MAE)) + MAE \\ \sigma- &> StandardDeviation \end{aligned} \tag{10}$$

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 univariate, 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 CO_2 Liu 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.

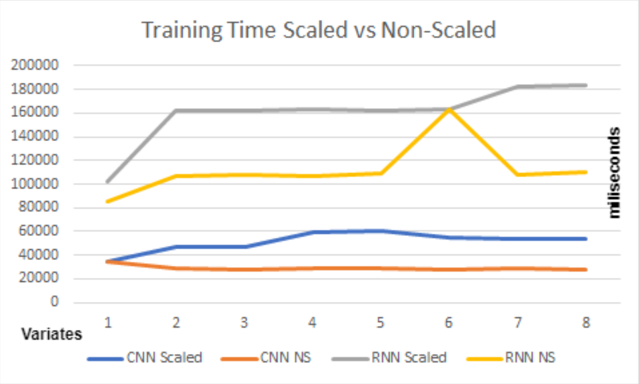

Fig. 4 Scaled vs Non-Scaled and RNN vs CNN Model Training Times

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 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 figure 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 figure 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.

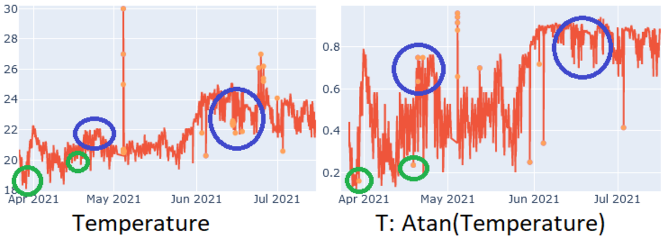

Fig. 5 Temperature vs Atan (Temperature) Comparison

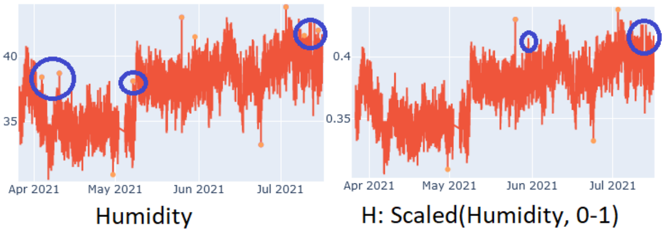

Fig. 6 Humidity vs Percentage (Humidity) Comparison

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 (CO_2) 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 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 figure 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.

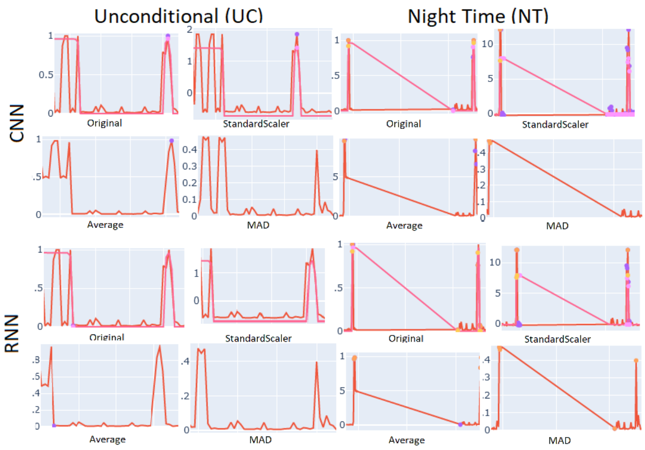

Fig. 7 Sound & Light Known "Anomalous Activity" Analysis

3.3.5 Point, Contextual, Combined Anomalies

Looking closely at figure 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°C. Another event marked anomalous in highlighted point 2 shows a sudden dip in temperature from 22.6°C to 22.9°C 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 hours on Monday, Tuesday, and Thursday and was detected as anomalous True Positive (TP). The office starts at 0700 hours on Friday and Wednesday, as shown in the black circle. The Friday 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 hours 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 methodsLiu et al. (2015) Laptev et al. (2015). The results are promising, but labelled data is rare in the real 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 and Yun (2018), financial systems Gran and Veiga

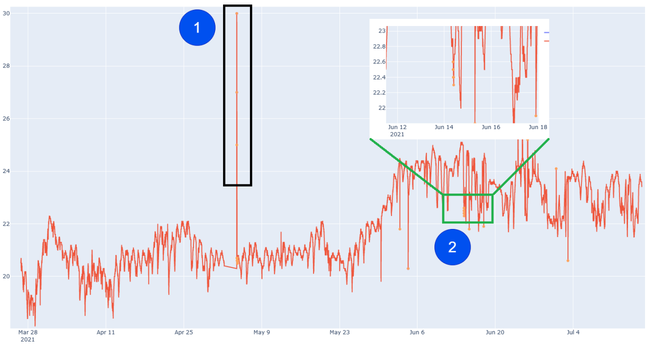

Fig. 8 1-Point Anomaly vs 2-Contextual Anomaly in Temperature Data Stream

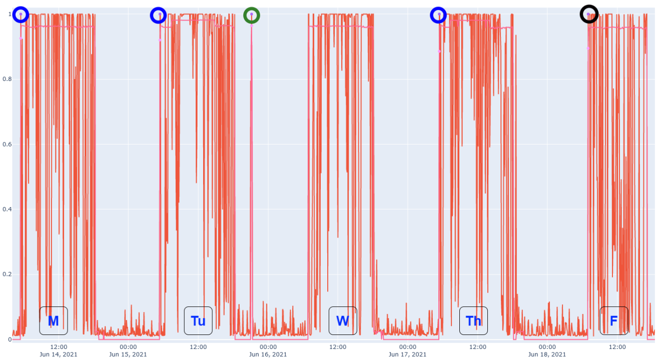

Fig. 9 Combined Contextual Anomalies in Sound and Artificial Light Data Streams

(2010), healthcare 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 stationery 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 and Yun (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. The 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 OneClassSVM (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 (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 CO_2 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 comport monitoring system for buildings which also provides the virtual visualization of thermal conditions in buildings. Authors of Ngah 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 Xu 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 ingester-end to keep data synchronised at capturing stage.

Importance of "normal" dataset: For the above-learnt 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.

Acknowledgement

This work is partially supported by EPSRC PETRAS (EP/S035362/1) and GCHQ National Resilience Fellowship.

References

Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. *Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017*, 2018-Janua:1–6, 2018.

Alex Schiffer. How a fish tank helped hack a casino, dec 2017.

Mennatallah Amer, Markus Goldstein, and Slim Abdennadher. Enhancing one-class Support Vector Machines for unsupervised anomaly detection. *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description, ODD 2013*, pages 8–15, 2013.

Mattia Antonini, Massimo Vecchio, Fabio Antonelli, Pietro Ducange, and Charith Perera. Smart audio sensors in the internet of things edge for anomaly detection. *IEEE Access*, 6:67594–67610, 2018.

Daniel B. Araya, K. Grolinger, Hany F. Elyamany, Miriam A.M. Capretz, and G. Bitsuamlak. Collective contextual anomaly detection framework for smart buildings. *Proceedings of the International Joint Conference on Neural Networks*, 2016-Octob:511–518, 2016.

Daniel B. Araya, Katarina Grolinger, Hany F. ElYamany, Miriam A.M. Capretz, and Girma Bitsuamlak. An ensemble learning framework for anomaly detection in building energy consumption. *Energy and Buildings*, 144:191–206, 2017.

Ayesha Asif and Muhammad Zeeshan. Indoor temperature, relative humidity and CO2 monitoring and air exchange rates simulation utilizing system dynamics tools for naturally ventilated classrooms. *Building and Environment*, 180(January):106980, 2020.

Francesco Aurino, Mariano Folla, Francesco Gargiulo, Vincenzo Moscato, Antonio Picariello, and Carlo Sansone. One-class SVM based approach for detecting anomalous audio events. *Proceedings - 2014 International Conference on Intelligent Networking and Collaborative Systems, IEEE INCoS 2014*, pages 145–151, 2014.

Catriona Brady. RATING TOOLS FOR RESILIENCE-UNDRR and the World Green Building Council. 2021.

Sebastian Buschjager, Philipp Jan Honysz, and Katharina Morik. Generalized isolation forest: Some theory and more applications extended abstract. *Proceedings - 2020 IEEE 7th International Conference on Data Science and Advanced Analytics, DSAA 2020*, 2(4):793–794, 2020.

Alfonso Capozzoli, Fiorella Lauro, and Imran Khan. Fault detection analysis using data mining techniques for a cluster of smart office buildings. *Expert Systems with Applications*, 42(9):4324–4338, 2015.

Nicholas Clements, Rongpeng Zhang, Anja Jamrozik, Carolina Campanella, and Brent Bauer. The spatial and temporal variability of the indoor environmental quality during three simulated office studies at a living lab. *Buildings*, 9(3), 2019.

Djamel Djenouri, Roufaida Laidi, Youcef Djenouri, and Ilangko Balasingham. Machine learning for smart building applications: Review and taxonomy. *ACM Computing Surveys*, 52(2), 2019.

Varick L. Erickson, Miguel Á Carreira-Perpiñán, and Alberto E. Cerpa. Occupancy modeling and prediction for building energy management. *ACM Transactions on Sensor Networks*, 10(3), 2014.

Amir Farzad and T. Aaron Gulliver. Unsupervised log message anomaly detection. *ICT Express*, 6(3):229–237, 2020.

Qingsong Feng, Yabin Zhang, Chao Li, Zheng Dou, and Jin Wang. Anomaly detection of spectrum in wireless communication via deep auto-encoders. *Journal of Supercomputing*, 73(7):3161–3178, 2017.

Afreen Ferdoash, Shubham Saini, Jitesh Khurana, and Amarjeet Singhz. Poster abstract: Analytics driven operational efficiency in HVAC systems. *BuildSys 2015 - Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built*, pages 107–108, 2015.

Romain Fontugne, Jorge Ortiz, Nicolas Tremblay, Pierre Borgnat, Patrick Flandrin, Kensuke Fukuda, David Culler, and Hiroshi Esaki. Strip, bind, and search: A method for identifying abnormal energy consumption in buildings. *IPSN 2013 - Proceedings of the 12th International Conference on Information Processing in Sensor Networks, Part of CPSWeek 2013*, pages 129–140, 2013.

Aurea Gran and Helena Veiga. Wavelet-based detection of outliers in financial time series. *Computational Statistics and Data Analysis*, 54(11):2580–2593, 2010.

Manish Gupta, Jing Gao, Charu C. Aggarwal, and Jiawei Han. Outlier Detection for Temporal Data: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 26(9):2250–2267, 2014.

Hasan Hayat, Thomas Griffiths, Desmond Brennan, Richard P. Lewis, Michael Barclay, Chris Weirman, Bruce Philip, and Justin R. Searle. The state-of-the-art of sensors and environmental monitoring technologies in buildings. *Sensors (Switzerland)*, 19(17), 2019.

Yassine Himeur, Abdullah Alsalemi, Faycal Bensaali, and Abbes Amira. Smart power consumption abnormality detection in buildings using micromoments and improved K-nearest neighbors. *International Journal of Intelligent Systems*, (August 2020):2865–2894, 2021a.

Yassine Himeur, Khalida Ghanem, Abdullah Alsalemi, Faycal Bensaali, and Abbes Amira. Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives. *Applied Energy*, 287(April):1–41, 2021b.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, nov 1997.

Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 387–395, 2018.

Vikramaditya Jakkula and Diane J. Cook. Detecting anomalous sensor events in smart home data for enhancing the living experience. *AAAI Workshop - Technical Report*, WS-11-07(December 2014):33–37, 2011.

Nikolay Laptev, Saeed Amizadeh, and Ian Flint. Generic and scalable framework for automated time-series anomaly detection. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015-Augus:1939–1947, 2015.

Jia Li, Shimin Di, Yanyan Shen, and Lei Chen. FluxEV: A Fast and Effective Unsupervised Framework for Time-Series Anomaly Detection. *WSDM 2021 - Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 824–832, 2021a.

Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2021b.

Dapeng Liu, Youjian Zhao, Haowen Xu, Yongqian Sun, Dan Pei, Jiao Luo, Xiaowei Jing, and Mei Feng. Opprentice : Towards Practical and Automatic Anomaly Detection Through Machine Learning Categories and Subject Descriptors. In *ACM Internet Measurement Conference*, 2015.

Fei Tony Liu, Kai Ming Ting, and Zhi Hua Zhou. Isolation forest. *Proceedings - IEEE International Conference on Data Mining, ICDM*, pages 413–422, 2008.

Xinyu Liu, Enhan Mai, Xiangxiang Xu, Hae Young Noh, Lin Zhang, Xinlei Chen, and Pei Zhang. Poster abstract: Individualized calibration of industrial-grade gas sensors in air quality sensing system. *SenSys 2017 - Proceedings of the 15th ACM Conference on Embedded Networked Sensor Systems*, 2017-Janua:5–6, 2017.

Yuan Luo, Ya Xiao, Long Cheng, Guojun Peng, and Danfeng Daphne Yao. Deep Learning-based Anomaly Detection in Cyber-physical Systems: Progress and Opportunities. *ACM Computing Surveys*, 54(5), 2021.

Luis Martí, Nayat Sanchez-Pi, José Manuel Molina, and Ana Cristina Bicharra Garcia. Anomaly detection based on sensor data in petroleum industry applications. *Sensors (Switzerland)*, 15(2):2774–2797, 2015.

Sean Meyn, Amit Surana, Yiqing Lin, and Satish Narayanan. Anomaly detection using projective Markov models in a distributed sensor network. *Proceedings of the IEEE Conference on Decision and Control*, pages 4662–4669, 2009.

Afiqah Ngah Nasaruddin, Boon Tuan Tee, Mohd Tahir Musthafah, and Md Eirfan Safwan Md Jasman. Ambient data analytic on indoor environment monitoring for office buildings in hot and humid climates. *Data in Brief*, 27, 2019.

Dong Yul Oh and Il Dong Yun. Residual error based anomaly detection using auto-encoder in SMD machine sound. *Sensors (Switzerland)*, 18(5):1–14, 2018.

Joern Ploennigs, Bei Chen, Anika Schumann, and Niall Brady. Exploiting generalized additive models for diagnosing abnormal energy use in buildings. *BuildSys 2013 - Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, 2013.

Ju Peng Poh, Jun Yu Charles Lee, Kah Xuan Tan, and Eric Tan. Physical access log analysis: An unsupervised clustering approach for anomaly detection. *ACM International Conference Proceeding Series*, 2020.

Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. Time-series anomaly detection service at Microsoft. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 3330680(c):3009–3017, 2019.

Stefano Rovetta, Zied Mnasri, and Francesco Masulli. Detection of Hazardous Road Events from Audio Streams: An Ensemble Outlier Detection Approach. *IEEE Conference on Evolving and Adaptive Intelligent Systems*, 2020-May, 2020.

Jagriti Saini, Maitreyee Dutta, and Gonçalo Marques. Indoor air quality prediction systems for smart environments: A systematic review. *Journal of Ambient Intelligence and Smart Environments*, 12(5):433–453, 2020.

Bernhard Schölkopf, Robert Williamson, Alex Smola, John Shawe-Taylor, and John Piatt. Support vector method for novelty detection. *Advances in Neural Information Processing Systems*, pages 582–588, 2000.

Bharath Sudharsan, Simone Salerno, Duc Duy Nguyen, Muhammad Yahya, Abdul Wahid, Piyush Yadav, John G. Breslin, and Muhammad Intizar Ali. TinyML Benchmark: Executing Fully Connected Neural Networks on Commodity Microcontrollers. *7th IEEE World Forum on Internet of Things, WF-IoT 2021*, 0:883–884, 2021.

Mojtaba Valinejadshoubi, Osama Moselhi, Ashutosh Bagchi, and Ashraf Salem. Development of an IoT and BIM-based automated alert system for thermal comfort monitoring in buildings. *Sustainable Cities and Society*, 66(November 2020):102602, 2021.

WHO. Air pollution is one of the biggest environmental threats to human health, alongside climate change., 2021.

Wentai Wu, Ligang He, Weiwei Lin, Yi Su, Yuhua Cui, Carsten Maple, and Stephen A. Jarvis. Developing an Unsupervised Real-time Anomaly Detection Scheme for Time Series with Multi-seasonality. *IEEE Transactions on Knowledge and Data Engineering*, 4347(c):1–1, 2020.

Haolong Xiang, Zoran Salcic, Wanchun Dou, Xiaolong Xu, Lianyong Qi, and Xuyun Zhang. OPHiForest: Order Preserving Hashing Based Isolation Forest for Robust and Scalable Anomaly Detection. *International Conference on Information and Knowledge Management, Proceedings*, pages 1655–1664, 2020.

Yizhe Xu, Chengchu Yan, Jingfeng Shi, Zefeng Lu, Xiaofeng Niu, Yanlong Jiang, and Faxing Zhu. An anomaly detection and dynamic energy performance evaluation method for HVAC systems based on data mining. *Sustainable Energy Technologies and Assessments*, 44(February): 101092, 2021.

Weizhong Yan and Lijie Yu. On accurate and reliable anomaly detection for gas turbine combustors: A deep learning approach. *Proceedings of the Annual Conference of the Prognostics and Health Management Society, PHM*, pages 440–447, 2015.

Weishan Zhang, Wuwu Guo, Xin Liu, Yan Liu, Jiehan Zhou, Bo Li, Qinghua Lu, and Su Yang. LSTM-Based Analysis of Industrial IoT Equipment. *IEEE Access*, 6:23551–23560, 2018.

Bendong Zhao, Huanzhang Lu, Shangfeng Chen, Junliang Liu, and Dongya Wu. Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics*, 28(1):162–169, 2017.