Anomaly Detection in an IoT-Acquired Environmental Sensor Data

Ashwath Kumar Channabasaya Salimath

Bachelor of Engineering, Computer Engineering

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

M.Sc. Computer Science (Data Science)

Supervisor: Prof. George Iosifidis

April 2019

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Ashwath Kumar Channabasaya Salimath

September 12, 2018

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Ashwath Kumar Channabasaya Salimath

September 12, 2018

Acknowledgments

Nothing is impossible, unless you think it is.

I would first like to thank my thesis advisor Prof. George Iosifidis of the School of Computer Science and Statistics at Trinity College Dublin. Prof. Iosifidis always guided me whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this research to be my work but steered me in the right the direction whenever he thought I needed it. I would like to thank Ambisense Ltd for giving me an opportunity to work on a real-world problem. I would also like to thank Dr Fiachra Collins and Conor Forde, Ambisense Ltd, for constant feedback on the research work and helping me to understand the problem in detail.

I would also like to thank following researchers who guided me in the research project: Rob Romijnders (Frosha.io), Rami Krispin (Apple iCloud), Pankaj Malhotra (TCS Innovation Labs), David Mack (Octavian.AI), Andrew Jefferson (Octavian.AI), Prof. Arthur White (Trinity College Dublin), Bharathi Selvan (Allied Irish Banks).

I would also like to acknowledge Prof. Ivana Dusparic of the School of Computer Science and Statistics at Trinity College Dublin as the second reader of this thesis, and I am gratefully indebted to her for her valuable comments on this thesis.

Finally, I must express my very profound gratitude to my dearest parents and my lovable siblings and true friends for providing me with unfailing support and continuous encouragement throughout my year of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. I would also like to owe my gratitude towards my Guru, Paramahansa Yogananda, for blessing me with ever-increasing spiritual strength in the journey of life. At last, I would like to thank my spiritual family, Self Realization Fellowship Dublin, for their constant help and motivation.

Ashwath Kumar Channabasaya Salimath

University of Dublin, Trinity College
April 2019

Anomaly Detection in an IoT-Acquired Environmental Sensor Data

Ashwath Kumar Channabasaya Salimath

, M.Sc. Computer Science

University of Dublin, Trinity College, 2019

Supervisor: Prof. George Iosifidis

A demand in monitoring a landfill site is increasing. A landfill site produces methane which is a poisonous gas and harmful for the existence of nature. The monitoring of the landfill sites are made possible using sensors, but the data quality issues arising from sensors persist. This research focusses on solving the problem of anomaly detection, in turn, solving the issues related to Data Quality and giving an indication for the presence of a subtle anomaly. An exhaustive search to solve the problem and to apply the existing techniques is the core idea. This also involved speaking to other researchers from the industry. Basic techniques ranging from Gaussian Mixture Model to an Autoencoder is implemented. Finally, Various problems in finding a perfect solution are reported, and an ensemble approach to solving the problem of anomaly detection in ecological monitoring is proposed.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Sensor systems have progressed environmental monitoring and research by giving tremendous amounts of information at temporal and spatial details, i.e. very close to real-time [1][2][3]. The development of wireless technologies has empowered associations with sensors in remote areas, making them conceivable to transmit information promptly through phones, radios, or neighbourhood. Likewise, the headways in cyberinfrastructure have enhanced information stockpiling limit, handling rate, and correspondence transmission capacity. It is conceivable to convey the most current data from sensors to end clients (e.g., within minutes after their accumulation). Although sensor systems can give numerous disadvantages, they are defenceless to breakdowns which can result in lost or low-quality information. For some instances, it is inevitable. Attempts to limit the failure of sensors must be taken at any rate, in turn, enhancing the general nature of the information/data. It has turned into a regular practice to push gushing sensor information online with constrained or no quality control. This data is typically conveyed to end clients directly, with no checks or assessments having been performed.

Observing data points over a period of time with appropriate transformation may uncover significant data about frameworks practices and patterns [4]. As of now, Business ventures are over-burdened with information and are searching for a quick examination, But have not yet wholly believed in the powerful algorithms, for example, Deep Learning and AI. Scholastic analysts incline toward essential tools like Matlab or Mathematica to run Time Series Analysis. Be that as it may, Statistics and

probabilistic instruments have increased widespread acknowledgement for quite a long time. Time Series Analysis had been frequently fit in with back and estimating. TSA can be connected to recognise irregularities in the Internet of Things (IoT) systems. In the Literature Review section, Specific consideration is paid to abnormalities that happen in savvy urban areas IoT utilisation cases for instance. A definitive point of this exploration work is to mount plug n play Anomaly Detection Engine (ADE).

In the most recent decade, the use of sensors in all areas has fundamentally expanded. This will keep growing. Predictive maintenance [5] is done when the maintained asset/resource is costly or imperative for crucial business functions. Predictive maintenance is additionally required when there are necessary financial or security/safety implications concerning the legitimate use of the devices/machinery. This isn't the same case if there is an occurrence of sensor parts which are generally shoddy and assume a minor role in the machinery processes. For such resources, support ordinarily implies straightforward substitution and Reactive Maintenance procedure would be the most well-known decision. Reactive Maintenance (otherwise called breakdown maintenance) alludes to the repairs that are done when hardware has just stopped working, so as to reestablish the machinery to its ordinary working condition [6]. Reactive maintenance can be a part of a balanced maintenance procedure yet it shouldn't be your go-to for all repairs. Appropriate planning of substitution of sensors has a coordinate effect on upkeep costs and its results. Notably, In situations where procedures, for example, environmental monitoring rely upon the sensor readings and the sensor disappointment or glitch may stop the task or cause misfortunes concerning the derived data.

We can apply machine learning methods to screen the present condition or anticipate the failure of sensors given their estimations and propose an ideal time for their substitution with a specific end goal to maintain a strategic distance from sensor failures. [7].

1.1 Aim

The thesis aims to provide a solution to a real-world challenge, i.e. by detecting anomalies in an IoT-acquired environmental data using machine learning techniques. The data has been provided by a company called AmbiSense Ltd. based in Dublin, Ireland. Detecting anomalies will boost the integrity of analysis of the data leading to the following:

- Finding of gas movement dangers and pathways
- Better educated field adjusting to support the effectiveness of gas yield
- Disposing of air spills and aerobic conditions
- Eventually giving better quality and amount of fuel for gas utilisation plants
- Increased revenues. Smart, field deployable, monitoring instruments and networks.

1.2 Scope and Objectives

The scope is to research and implement machine learning algorithms to detect anomalies in an environmental sensor data acquired through IoT devices.

The Objectives are defined as below:

- Identify machine learning theory/algorithms applicable for use with the data.
- Evaluate the effectiveness of selected ML algorithms.
- Refine the implementation of evaluated ML algorithms to real-time updating environmental data.

1.3 Landfill Gas Explosions

The time when the first landfill gas blast happened, the waste administration industry came to understand the dangers involved with the age of landfill gas in the landfills and the punishments concerning damage to the people and the environment.

It took a few landfill explosions before operators and regulators made a move, Although the landfill administrators would have gained from the parallel encounters of landfill gas blast which were there from the beginning. It is astonishing that a push to keep these landfill gas blasts was not taken notwithstanding when the appalling loss of life from the UK water industry's Abbeystead Explosion in the 1970s which was at that point well understood.

Some of the landfill gas explosion accidents are below:

1. Loscoe Explosion, Derbyshire
   A blast happened in a bungalow which was adjoining the landfill site at Loscoe amid March 1986. This was because of a massive fall in barometric pressure (29 millibars in 7 hours). The subsequent examination demonstrated that two more houses had been unfit for living for the former nine months, and others for brief periods.

Figure 1.1: **Loscoe Landfill Gas Explosion UK.** The explosion happened due to a massive fall in baromateric pressure.

2. Winston-Salem, North Carolina, USA
   A gas blast occurred in weapons depot constructed near a landfill site in 1969. The building was raised seven years sooner when the site was operational, however about seven days before, some additional blast material was deposited over the place, and it might be the explanation behind gas migration into the building. The blast caused three individuals death, and twenty-five individuals were harmed [8].

   After the closure of the site and it was found that there were some alterations to the pathways for gas migration. These modifications or blockages caused the landfill gas to aggregate in an encased region and achieved an explosive limit.

Figure 1.2: **Winston-Salem Explosion USA.** The explosion took place due to gas migration from a landfill site.

These notable and influential gas explosion incidents which to a substantial degree have moulded the UK Environment Agency's landfill improvement necessities and gas migration prevention guidelines. At that point, the EU Landfill Directive in the late 1990s was presented which is currently exemplified in the EU ATEX Directive (and UK's DSEAR controls). Gratefully, with current altogether enhanced landfill practices, landfill gas explosionaccidents have turned into a relic of times gone by.

1.4 Quality Assurance vs Quality Check

The ideas of Quality Assurance (QA) and Quality Control (QC) have a particular importance, yet they are frequently utilized together and are firmly related. With regards to sensor systems, QA and QC are closely aligned. As QA is process oriented and QC is product oriented, It is hard to decide when the information/data has turned into something useful or tangible. QA could be characterized as an arrangement of procedures to guarantee that the sensor system and conventions are followed to limit errors in the information/data produced. The motivation behind QA is to extract "satisfactory/acceptable" data information/data while reducing/diminishing the need for restorative measures to enhance information/data quality, though QC happens after the information/data is created. QC tests whether the information/data meets the quality necessities as delineated by the end clients. QA can be thought of a proactive or preventive procedure to keep away from data quality issues, and QC is a procedure to identify suspect information/data after they have been produced. The focal point of this reasearch is to utilize Machine Learning procedures to characterize QC and provide appropriate reasoning for the anomalous data.

1.5 Quality Assurance

Environmental sensors more often than not, deliver low-quality data and fail to transmit the data [9]. Sensors often fail due to hardware problems. The crucial part is to identify unpretentious impedances amid which information/data is still produced however with traded off quality. Since when sensors stop working, by and large, the subsequent loss of information might be recognizable (with the remarkable exemption of event detectors).The subtle impedances may result from natural conditions, for example, unreasonable dampness or extraordinary temperatures that surpass the working scope of the sensor. Steps ought to be taken to maintain a strategic distance from or possibly limit sensor failures when planning sensor networks and setting up protocols; in any case, the advantages or disadavantages of this method should be adjusted against the shrinkage in information loss.

Beside limiting information loss because of failing sensors, sensor replication is help-

ful for identifying subtle anomalies, for example, calibration drift, which is often hard to recognize with unreplicated sensors. Drifts tend to occur when sensor parts weaken after some time due to age-related procedures including consumption, weariness, and photodegradation. At least three replicate sensors are required to identify drift since it is hard to figure out which sensor is drifting with two sensors. Sensors require routine support and planned sensor calibration that, now and again which should be possible just by the manufacturer only. In situations where unscheduled support is required, loading new parts nearby guarantee that any piece of the system can be supplanted promptly. The role of field experts is very essential as they are regularly mindful of sensor-related errors coming about because of routine upkeep, repairs, or different service interruptions. Tracking these support occasions is urgent for distinguishing and understanding the origin of incorrect information/data.

1.6 QC and Machine Learning

Recently there has been a prevalence of machine learning techniques being applied to ecological sensor. Machine learning strategies portray a data-driven way to deal with QC where machine learning models are prepared to utilize exact information gathered from sensors in a push-button fashion.

Machine learning provides excellent results without the need of expansive information about the senors or phenomena being estimated, But it needs a chronicle of labelled data that contains cases of anomalous and non-anomalous data to demonstrate model training and validation effectiveness. Generative models, for example, Bayesian systems take in a joint probability distribution over the procedure, creating both the data input and the output. Discriminative algorithms, for example, logistic regression encodes a useful mapping from an arrangement of inputs(sensor readings) to an arrangement of output labels (anomalous or non-anomalous). Artificial neural networks, support vector machines, decision trees, and probabilistic models have all turned out to be prominent machine-learning approaches [10][11][12]. Software bundles, for example, WEKA [13][14] and MATLAB [15] contain libraries for applying machine-learning algorithms on the sensor data. In any case, It can be as yet difficult to utilize these above software packages accurately. The data must be carefully separated out into

training, validation, and test sets to ensure that it generalises well to unseen future observations [16].

1.7 QA/QC Best Practices

The best practices for the QA/QC [17] of ecological sensor data are as follows:

1. Automation of QA/QC procedures.
2. A required level of human investigation should be maintained.
3. Limiting information/data loss by means of customary planned maintenance and repairs.
4. Having prepared access to new parts of the devices.
5. Recording the date and time of known system events that may influence the measurements of the sensors.
6. An automated alert framework ought to be actualized to caution about potential sensor issues.
7. Performing range checks on quantitative data.
8. Retain the original unmanipulated data of the system.
9. Comparing the data with data from related sensors.
10. Utilizing flags to pass on information about the generated data from the system. Evaluating the uncertainty in the data, if that is practical. Complete metadata should be provided.
11. Report all QA/QC strategies that were used for the system. Archive information/data processing steps.
12. Archive all versions of the generated data, work processes, QC methods, and models utilized (called as data provenance).

1.8 Challenges

- Unavailability of a labelled dataset. Due to which, the performance of the machine learning models cannot be verified.
- Landfill site is in operation. So, Gas behaviour is unpredictable.
- Dataset size is small. There is a need to have bigger dataset, So that it is easy for algorithms to generalise.
- Presence of Missing Data. As, the data is a time series dataset, So there is a loss of temporal dependency between sensor samples.

1.9 Contribution

- Applied Gaussian Mixture Model to help the company in labelling the dataset.
- Performed a Benchmark Analysis with existing anomaly detection methods on this dataset.
- An Autoencoder using H2O was built and various threshold functions have been devised to see if a sensor sample is an outlier or not.
- Ensemble Approach has been proposed which will have a balance between an Autoencoder and an Outlier Ensemble.

Chapter 2

Literature Review

The Internet has advanced from its unique point of giving access to web assets all inclusive to what is ordinarily considered these days the Internet of Things. It is normal that the Internet of Things will connect with each other and have an existence on the Internet just with an IPv6 address for example. The expected market for the Internet of Things is evaluated in trillions, exceptionally a long way from the worldwide human count. This has prompted the making of new plans of action with the improvement of devoted IoT systems, for example, SigFox, LoRa, Symphony Link, and NB-IoT [18], and creation of IoT consistent devices from microcontrollers makers, for example, Microchip, Intel, and Raspberry Pi [19]. Programming organizations have concocted virtual machines and tools for analytics on big data. Analytics will help smart cities to be able to provide better services to its people. Network device constructors like Cisco and Juniper, for example, have thought of gateways and routers to suit device association, directing, and IoT information travel. The bunch of advances required inside the Internet of Things ecosystems ought to engage savvy situations as it happens in like manner in smart urban areas.

2.1 IoT Value Chain

The IoT value chain demonstrates the value-added features to IoT [4]. Data Analytics for IoT is a key benefit for both consumers and service providers. Data Analytics can

become one of the high paying technologies than key technology empowering agents like SDN, IPv6, and 5G, much more than machine robotization. It can be called Analytics as a Service (AaaS). Cisco's yearly Visual Networking Index reports that machine-to-machine (M2M) associations will represent the greater part of the world's 27.1 billion gadgets and associations and these associations bolster IoT applications by 2021. [20].

Figure 2.1: **IoT Value Chain.**

2.2 Types of Anomalies

Identification of rare events or observations in a system which raise concern by deviating from the majority of the data is called as Anomaly Detection [21]. It is also known as Outlier Detection. Basically, the meaning of anomalies will vary as per the industry context i.e. It might refer to a Bank fraud, a defect in the medical process or some errors in the text. Anomalies are also known as novelties, noise, exceptions, deviations and outliers [22].

The identification of anomalous behaviour is pertinent in a variety of areas, for example, system health monitoring, event detection in sensor networks, others. One of the other applications is to use Anomaly Detection to remove anomalous data from the dataset. When we apply supervised learning, It's better to train the algorithm on non-anomalous data. It is proved that removing anomalous data from the dataset results in a significant increase in accuracy from a statistical point of view [23][24].

The types of anomalies [4] can be descibed as below:

1. Static vs Dynamic
   Abnormalities are characterised as data points not following current patterns present in the data. Static refers to the anomalies in the same direction but with different characteristics. Moreover, Dynamic refers to the anomalies in the opposite direction.

2. Outlier
   Relatively impossible values in the sensor data stream can be thought as anomalous. When there is an excessive deviation of the value from the mean, suppose by $\pm 4\sigma$, at that point, we can consider this data point as anomalous [25]. (The limit can likewise be computed utilising the percentile.)

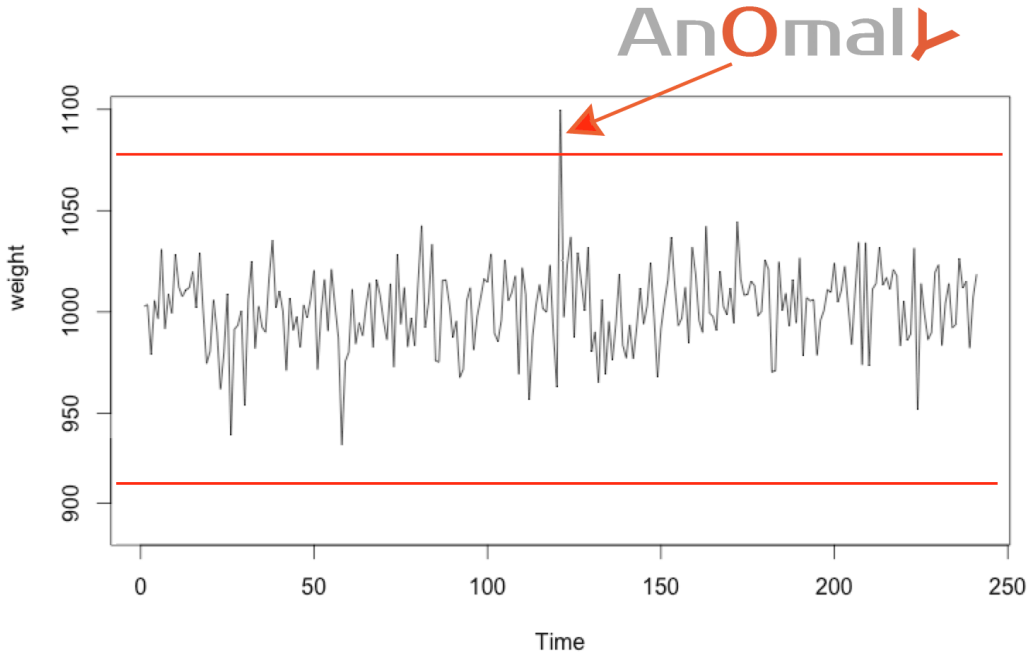


Figure 2.2: **An example of an outlier.**

3. Contextual
   The variation from the norm is context specific. This sort of anomalous behaviour is prevalent in time-series information. Business use case, i.e. Spending $100 on food consistently amid the Christmas season is typical, however perhaps odd in

some other seasons. Another use case, A data point could be an abnormality in one setting yet not in another. For instance, a temperature of 35C in January is an abnormality in a northern European nation however typical in a southern hemisphere island for that month.

4. Collective
   A set of data points collectively helps us to identify anomalous behaviour. Business use case, for example, Someone is endeavouring to duplicate information from a remote machine suddenly to a local host, an inconsistency that would be hailed as a potential attack. Another example, Contextual anomalies happen when there is stretching in time of a specific anomaly as it occurs in telecom transmission; there is an accumulation of delays that result in jitters.

2.3 Anomaly Detection Methods

We are dealing with time series sensor data. The anomaly detection can be applied using different time series models including machine learning. It is difficult to find a one size fit all solution for the problem that we are addressing and as well as we have no de facto time series model which will suit the anomaly detection application.

2.3.1 Autoregressive Models

A sequence of measurements of the same sensor/variable(s) generated over a period of time is called as Time Series. Time Series data is usually sampled at hourly, daily, monthly, quaterly or on yearly basis. Lets take an example of a variable y being measured over a period of time i.e. the sensor readings sampled hourly. To describe a variable that it is being measured as a time series, We denote that variable as y_t. Here, y_t means that y is measured in a time period of t.

$$y_t = \beta_0 + \beta_1 y_{t-1} + \epsilon_t \tag{2.1}$$

When we regress a value from a time series on previous values of that same time series, i.e. y_t on y_{t-1}, We formally call this as an Autoregressive Model. In this type of model, the response variable of the previous time period i.e. y_{t-1} becomes a predictor

for y_t and we have an error component Beta also. The assumptions for Beta is same as the assumptions of errors in a simple linear regression model. The number of preceding values in the time series which were used to predict the value at the present time forms an order. This refers to the order of an autoregressive model. So, Equation 2.1 model is called as a first-order autoregression, which is represented as AR(1).

If we want to predict y_t based on the y_{t-1} and y_{t-2}, then the autoregressive model would be the following:

$$y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \epsilon_t \tag{2.2}$$

The model is called as AR(2) i.e. second-order autoregression. The value y at time t is predicted using the values at time (t-1) and (t-2). Generally, the above equation can be written as k^{th} order autoregression as follows:

$$y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + ... + \beta_k y_{t-k} + \epsilon_t \tag{2.3}$$

The above equation is called as AR(k) which is a multiple linear regression. The value y of the time series at any time t is a function of the values at times (t-1), (t-2), ..., (t-k).

Autocorrelation Function and Partial Autocorrelation Function

Autocorrelation (ACF) for a time series is given by the following:

$$correlation(y_t, y_{t-1}) \tag{2.4}$$

i.e. the coefficient of correlation between the values y_t and y_{t-1} is called as Autocorrelation function. Here, the value k is called as lag which refers to the time gap being considered for autocorrelation. We can say that a lag k autocorrelation will refer to the correlation between values that are k time periods apart.

The reason to use ACF is that it helps us to measure the linear relationship between a sensor event at time t and a sensor event at previous times. A transformation of the time series is required when we want to only measure the association between y_t and

y_{t-k}, if we assume AR(k) model. The reason for transformation is that we want to filter out the linear influence of the variables that lie in between y_t and y_{t-k}. These variables are y_{t-1}, y_{t-2}, y_{t-3}, ..., $y_{t-(k-1)}$. Then, By calculating the autorrelation of the transformed time series, we obtain PACF (Partial Auto Correlation Function).

PACF is useful for identifying the order k of an autoregressive model. We can think of differences between ACF and PACF as analogues to the differences between R_2 and partial R_2 [26]. It is important to understand that the choice of order k makes sense. There is a graphical approach for identifying the lag of an autoregressive model i.e. by plotting ACF and PACF values vs. lag k. In case of ACF vs. lag, If we see large values of ACF and a non-random pattern, then it means that the values y in the time series are serially correlated. In case of PACF vs. lag, the pattern usually appears in a random fashion, but large values of PACF at a given lag k indicates that this value of k as a possible choice for the order of the model.

2.3.2 Seasonal and Trend decomposition using Loess

In STL Decomposition, Data instances together with the noise or from multiple data sets over a period of time are decomposed and are analyzed to detect anomalous behaviour. STL method was developed in 1990 [27] and is a robust method for decomposing time series data. In STL, Loess is a method which is used to estimate non-linear relationships. STL has several advantages over classical decomposition methods i.e. STL can handle any type of seasonality i.e. It doesn't limits to only monthly and quarterly data, which is also applicable in the case of methods like SEATS and X11.

The rate of change in the seasonal component can be controlled by the user and the seasonal component can change over time. The user can also control the smoothness of the trend-cycle. Basically, In STL, we have three components i.e. seasonal, trend-cycle and remainder component. The user can specify a robust decomposition due to which STL can be robust to outliers. When the user specifies robust decomposition, the estimates of the trend-cycle and the seasonal components are not affected whereas the remainder component gets affected.

Apart from the above, STL has some disadvantages also. STL is not able to handle the calendar variation or trading day automatically, and it is able to provide facilities for additive components only. There are two main parameters that needs to be chosen while using STL decomposition. These parameters are (t.window) i.e. trend-cycle window and (s.window) i.e. seasonal window. These two parameters control how rapidly the seasonal component and the trend-cycle component will change in the STL decomposition. For example, Following figure describes the electrical equipment orders with its three additive components i.e. trend, seasonal and remainder are shown. These additive components using flexible trend-cycle and fixed seasonality [28].

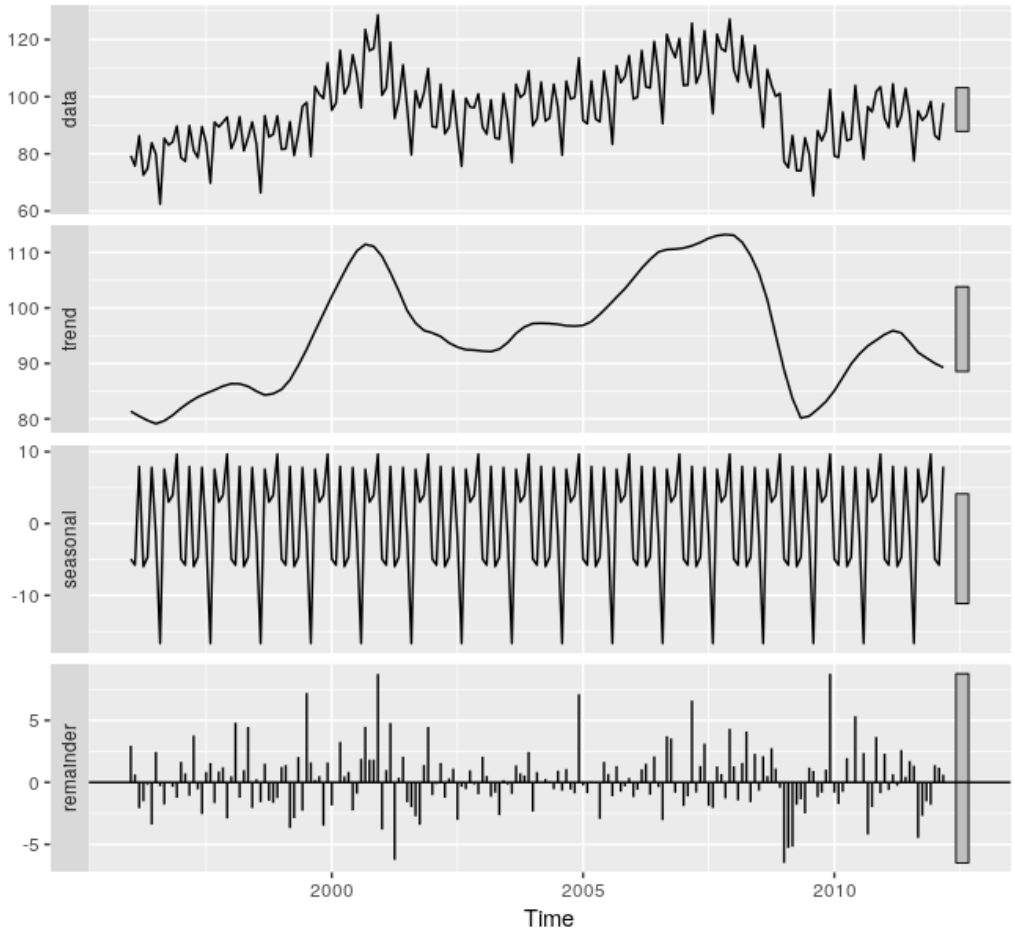


Figure 2.3: **STL Decomposition: Electrical Equipment Orders**

2.3.3 Symbolic Time Series Analysis

In Symbolic TSA (Time Series Analysis), We consider time series as a sequence of pairs. Each pair is represented as (time index, value). If there is a constant difference between the values then the time index may be implied. The time series is segmented into "Windows" which represent the time series between two-time indexes. Symbols can represent Windows, as symbols have probabilities in a finite space, And we can think of this as a probability of a time series. These symbols can be represented as an integer and they are easy to store and manipulate.

Generic data mining methods have some constraints in terms of processing. Suppose, We have only one gigabyte of main memory and we want to apply K-means clustering. This might take a few hours due to the limited main memory. This is one of the reasons to convert a time series into a symbolic representation. In generic Data Mining, An approximation of the data is created which will fit into the main memory, yet retains the crucial features. Then the problem is solved approximately in the main memory. Some of the common approximation methods [29][30] are provided in the below figure 2.4.

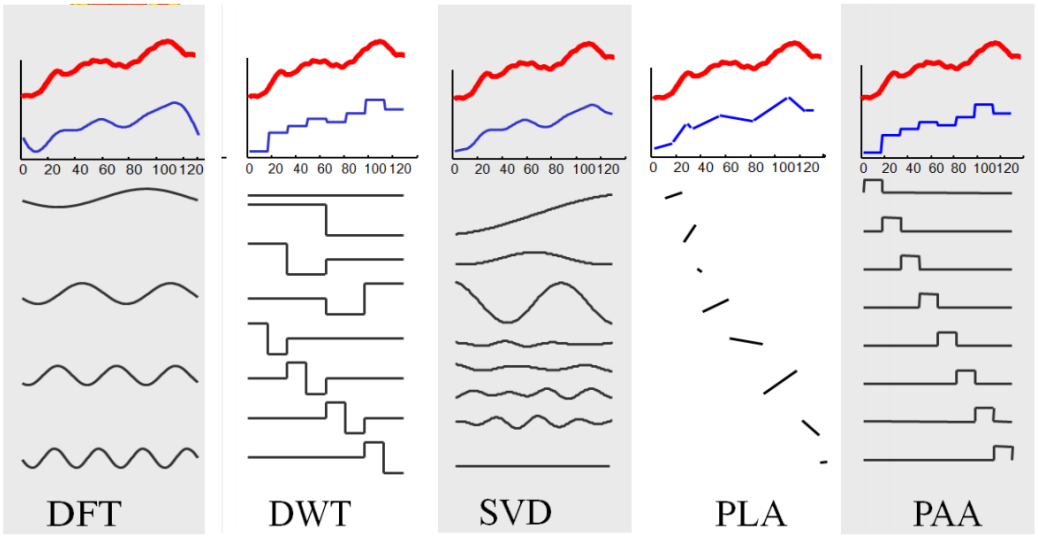


Figure 2.4: **Common Approximation Methods.**

Symbolic Representation Of Time Series

Various algorithms exist to represent time series in a Finite Symbol Space, and these algorithms are thought of as "Feature Reducers". Self Organizing Maps are known as a traditional form of Feature Reducer, whereas SAX (Symbolic Aggregate approXimation) is also a Feature Reducer but designed specifically for time series data. Although, there are many ways to reduce the time series.

Normalization of Time Series

It refers to the normalisation of the time series to zero mean and a unit of energy. This normalisation procedure ensures that all the elements of the input vector are transformed into an output vector whose mean is approximately zero and the stand deviation is in the range close to 1. The formula for the transformation is described below:

$$x_{\imath}\prime = \frac{x_{\imath} - \mu}{\sigma}, i \epsilon N \tag{2.5}$$

To have meaningful comparisons between the two time series, then both the time series must be normalised. This is an essential step which allows an algorithm to focus on the structural similarities/dissimilarities rather than taking amplitude in its analysis.

SAX

SAX is a method to reduce time series into a series of a symbol. The technique was developed in the early 2000s by Dr Eamonn Keogh et al.[31]. "SAX is the first symbolic representation of time series that allows for dimensionality reduction and indexing with a lower-bounding distance measure" [31]. This method allows the reduction of a time series of arbitrary length n to a string of arbitrary length, where $w << n$.

The SAX representation of the time series is obtained with two steps as given below:

1. Reduce dimension by PAA (Piecewise Aggregate Approximation)
   The time series C of length n can be represented in a w-dimensional space by a

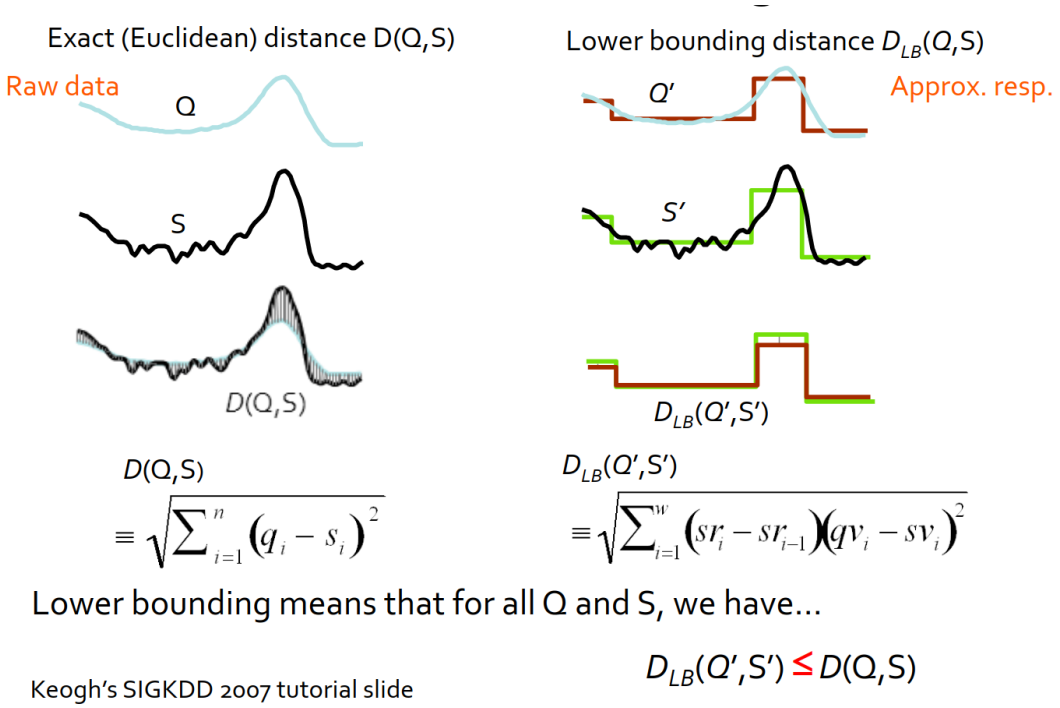


Figure 2.5: **Euclidean and Lower Bounding Distance.**

vector $C' = C'_1 + C'_2 + ... + C'_w$. The i^{th} element is calculated by

$$\overline{C_i} = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} C_j \tag{2.6}$$

2. Discretization

   Normalize C' to have a Gaussian Distribution as described in the above subsection. Then, Determine breakpoints which produce a equal-sized area under the Gaussian curve.

   Various anomaly detection applications are being studied based on the symbolic representation of the time series.

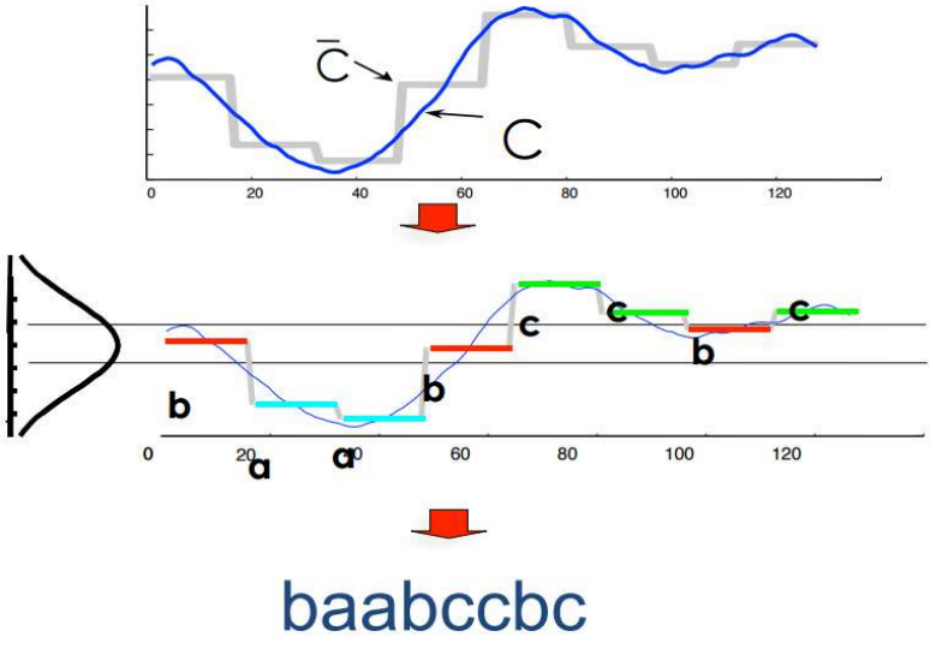


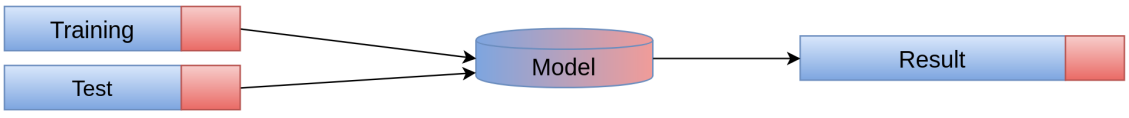
Figure 2.6: **SAX Method.**

2.3.4 Machine Learning Methods

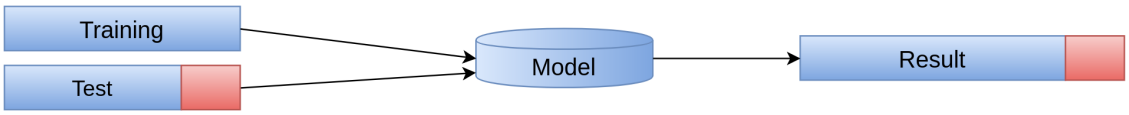
In Machine Learning, there are mainly three branches, i.e. supervised learning, semi-supervised learning and unsupervised learning. In supervised mode, the detection is done using featuring or inference. However, In unsupervised learning, the detection is difficult as we dont have any labels and the unusual pattern is unknown. In unsupervised, the algorithm that learns from the data points is required to be analysed. Some of the supervised methods are Naive Bayes, Decision Trees, Random Forest, K-nearest Neighbour, Support Vector Machine (SVM), Deep Learning. Some of the famous unsupervised algorithms are N-SVM, K-means clustering, DBSCAN, Latent Dirichlet Allocation (LDA) and Stream Clustering. In the following section, the notion of machine learning is described in detail.

2.4 Availability of Labelled Dataset

Depending on the availability of the labels for the dataset, accordingly, the decision is made between supervised, semi-supervised and unsupervised learning. In supervised, there is a need that the labels should be present in both the training set and the test set, whereas there is some relaxation concerning semi-supervised and unsupervised learning. Following figure 2.7 describes the three fundamental approaches to machine learning, i.e. Supervised learning required fully labelled dataset, semi-supervised learning requires a dataset with normal data points/samples, i.e. anomaly-free samples and unsupervised learning requires an unlabeled dataset.



(a). Supervised learning uses the labelled dataset i.e. normal data points and as well as anomalous data points. The algorithm will classify the instances in two labels, i.e. Anomalous OR Non-Anomalous.



(b). Semi-supervised learning uses normal data points to learn about the patterns in the dataset. Any unusual patterns should be detected by this type of model.

(b). Unsupervised learning uses an unlabelled dataset. The anomalous behaviour in the data is identified using the internal properties of the dataset. The anomalous data points exhibit different nature.

Figure 2.7: **Types of Machine Learning Algorithms.** Algorithms defined as per the availability of labels.

2.5 Python Outlier Detection (PyOD)

PyOD [32] is a very powerful software package which covers the most useful algorithms used for outlier/anomaly detection. It is helpful in identifying outlying objects in a

multivariate data. It provides three functionalities, i.e.

1. Outlier detection algorithms (see figure 2.8)
2. Outlier ensembles (see figure 2.9)
3. Outlier detection utility functions

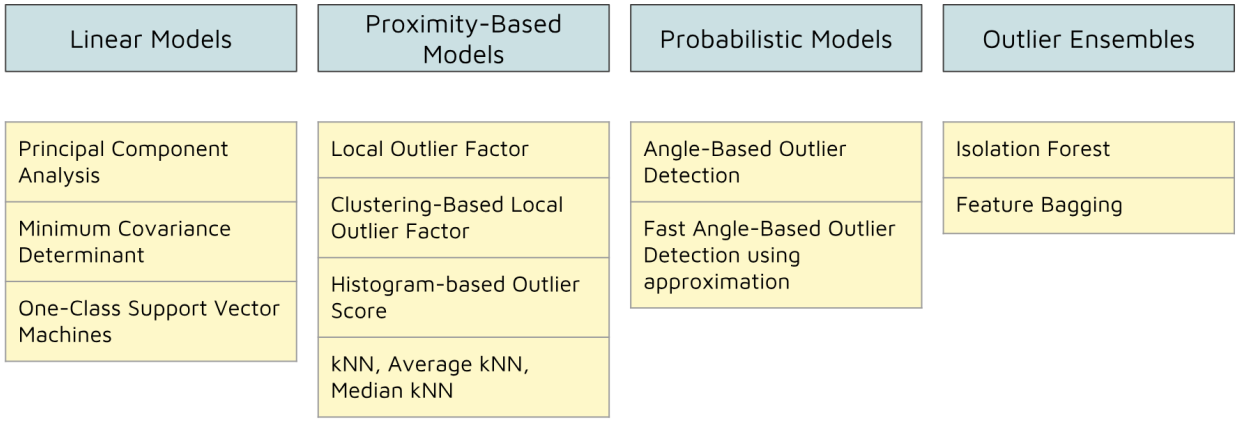


Figure 2.8: **Python Outlier Detection (PyOD).** Python toolkit to identify outlying objects in data with both unsupervised and supervised approaches.

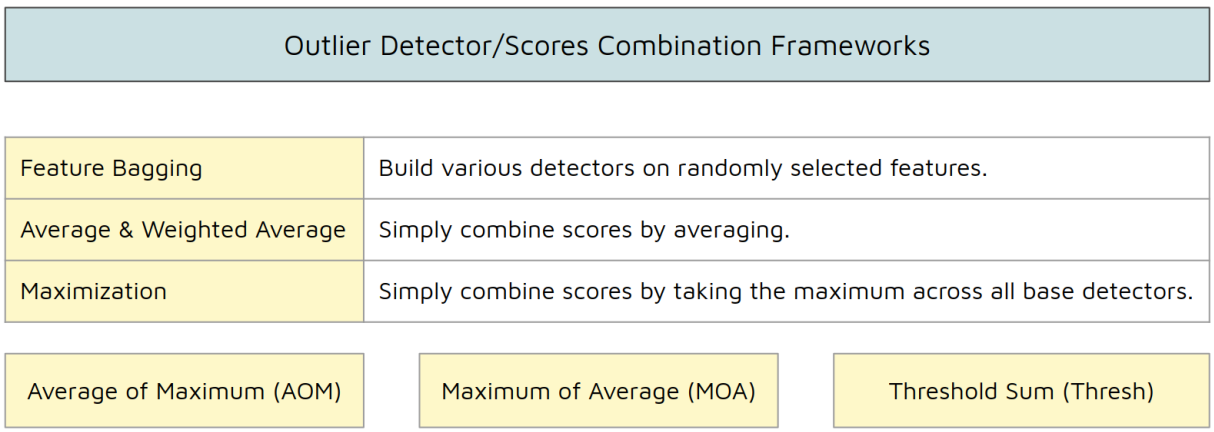


Figure 2.9: **Python Outlier Detection (PyOD).** Outlier Detector/Scores Combination Frameworks.

2.6 Python Outlier Detection (PyOD) Methods

2.6.1 Principal Component Analysis

Principal component analysis (PCA) [33][34] refers to a mathematical procedure which transforms some (possibly) correlated variables into a (smaller) number of uncorrelated variables. These uncorrelated variables are called as principal components. The objectives of carrying out PCA is to reduce or discover the dimensionality of the dataset and to identify if new meaningful underlying variables exist. The visual explanation of PCA is provided in the figure 2.10 [35] .

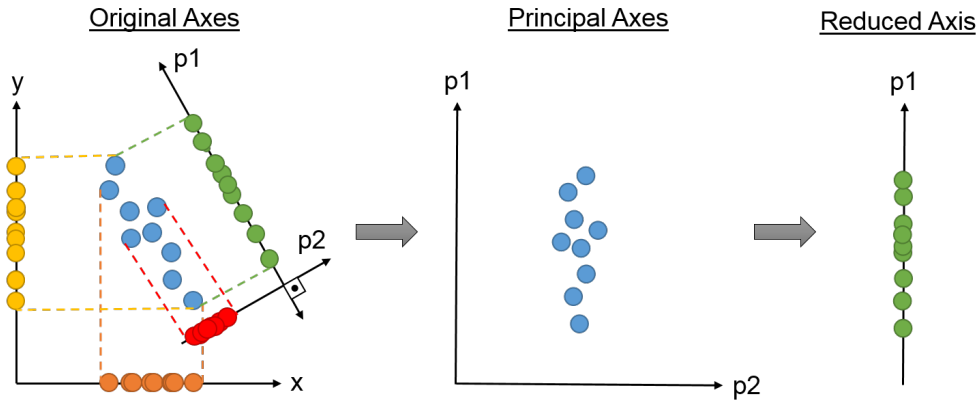


Figure 2.10: **PCA Visual Explanation.** In two-dimension PCA is an orthogonal coordinate system transformation that prioritizes maximum variance.

The original variables Xs are transformed into Principal Components Ys using eigenanalysis. The eigenvalues are the corresponding variances of the principal components in decreasing order of importance. The eigenvectors are the corresponding weight sets for the principal components which are linear combinations of the original variables.

Each eigenvector has a corresponding eigenvalue. The eigenvalue is a scalar that indicates how much variance is there in the data along that eigenvector. A principal component with larger eigenvalue does an excellent job of explaining the variance in the data, otherwise not. When performing PCA, it is always good to normalise the dataset.

Selecting Principal Components

Enough principal components are retained which explain the cumulative variance of $> 50-70\%$. Kaiser criterion and scree plot can be used to consider principal components. In the Kaiser criterion, the principal components with eigenvalues > 1 are retained. Scree plot represents the ability of principal components to explain the variation in data.

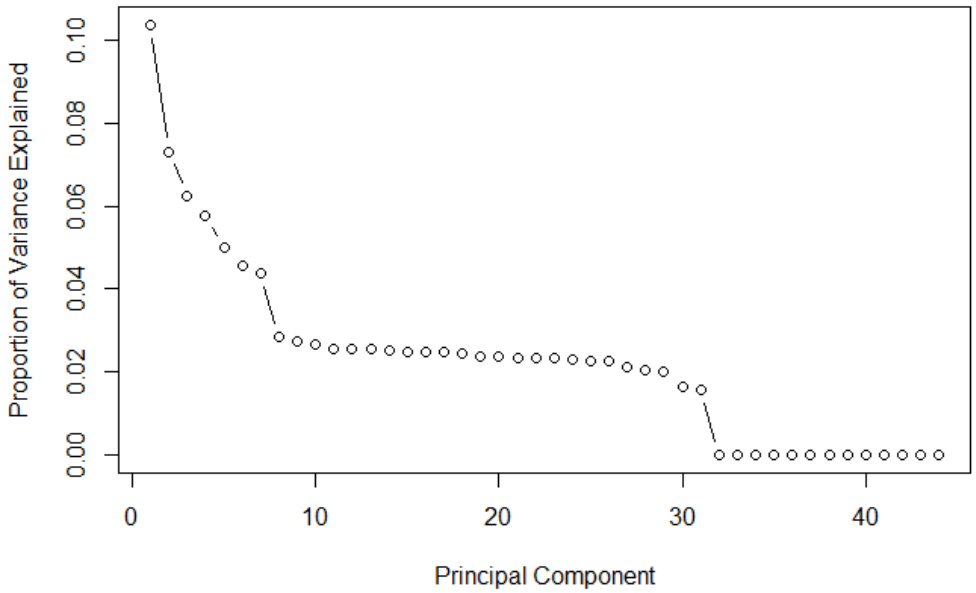


Figure 2.11: **Scree Plot.** It used to identify the number of principal components which explain variation in the data.

Outlier Detection

PCA uses the sum of weighted projected distances to the eigenvector hyperplane as the outlier scores [36].

2.6.2 Minimum Covariance Determinant

An outlier is typical in a noisy dataset. Geometrically, the covariance matric specifies an ellipsoid the circumscribes the primary dimensions of the data in N-space and N refers to the number of features in the dataset. A paper by Hubert and Verboven [37]

illustrates (see figure 2.12) that the presence of outlier data stretches the ellipsoid along the axis of the outliers relative to the mean. For many applications, there is an interest for robust covariance described by the smaller ellipse. Rousseeuw developed an algorithm called FAST-MCD [38] which uses the Mahalanobis distance. The Mahalanobis distances are used as the outlier scores to determine outliers [39].

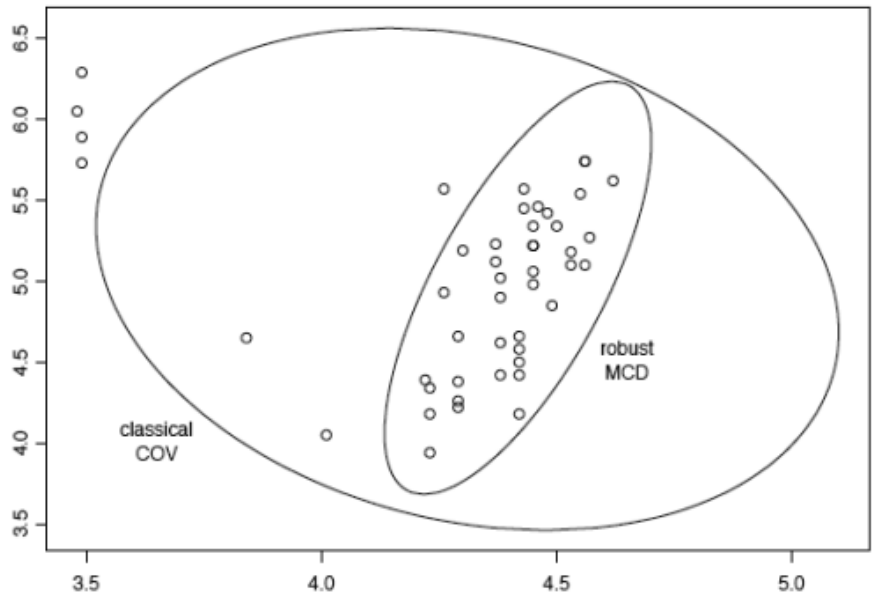


Figure 2.12: **Classical and Robust Tolerance Ellipse (97.5%).** The data points outside the robust MCD ellipsoid are possible outliers.

2.6.3 One-Class Support Vector Machine (OCSVM)

The idea behind one-class SVM is to train a classification model that can distinguish normal data from outliers. The normal dataset, i.e. one class dataset is assume known apriori. The model training involves separation of all data points from the origin with a maximised margin, i.e. it learns the decision boundary of the normal class. If we consider a training set which contains samples labelled as both normal and outlier, then the classifier cannot detect an unseen anomaly. Because the training set is typically heavily biased, as the percentage of outliers in the dataset is usually less, i.e. less than 0.5 %. One-class SVM solves the above problem of class imbalance. Once the one-class SVM model is trained, then it is used to predict if a sample is an outlier or not. One of

the best thing with one-class SVM is that it uses kernel function, e.g., Gaussian RBF kernel, to detect non-linear patterns. The idea of one-class SVM is also helpful in the automatic identification of abnormal events embedded in normal time-series data [40].

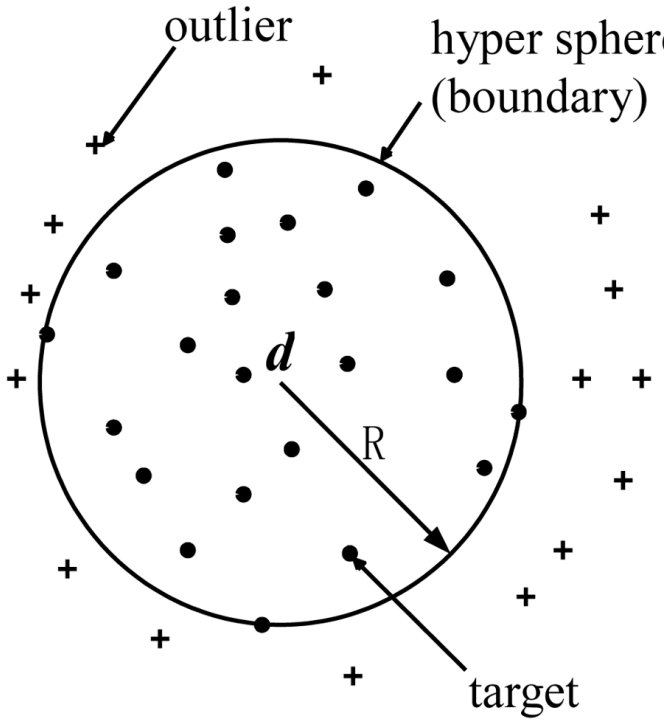


Figure 2.13: **One-class SVM.** A one-class SVM approach to outlier detection.

2.6.4 Local Outlier Factor

The local outlier factor (LOF) strategy scores points in a multivariate dataset whose cases (rows) are thought to be produced independently from the same probability distribution [41]. Local outlier factor is a density-based technique that depends on the nearest neighbours search. The LOF technique scores every datum point by figuring out the proportion of the average densities of the point's neighbours toward the density of the point itself. As indicated by the technique, the assessed density of a point p is the number of p's neighbours divided by the aggregate of distances to the point's neighbours.

Let N(p) be the set of neighbours of point p, k be the number of points in this set, and d(p,x) be the distance between points p and x. Then, the estimated density is given as below:

$$\hat{f} = \frac{k}{\sum_{x \in N(p)} d(p, x)} \tag{2.7}$$

moreover, the LOF score is given as below:

$$LOF(p) = \frac{\frac{1}{k} \sum_{x \in N(p)} \hat{f}(x)}{\hat{f}(p)} \tag{2.8}$$

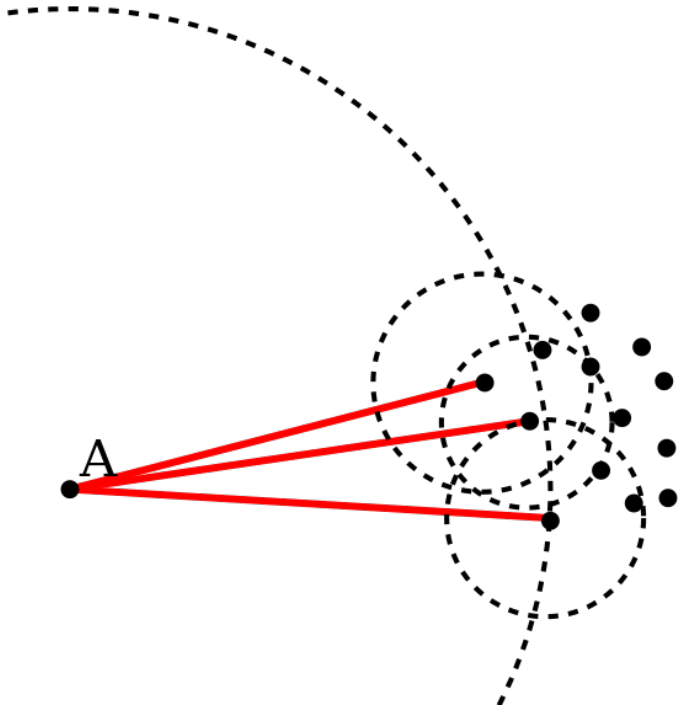


Figure 2.14: **Local Outlier Factor Intuition.** LOF score of point A is high because its density is low as compared to its neighbours densities. Dotted circles here represent the distance to each points third nearest neighbour.

2.6.5 Cluster-based Local Outlier Factor

The idea for Cluster-based local outlier factor (CBLOF) is to first cluster the dataset using the k-means algorithm and then use the distance from the data instance to the centroid as the anomaly score [42]. The data is separated into a large cluster and small cluster. The CBLOF score is calculated as below:

$$CBLOF(p) = \begin{cases} \\ \mid C_i \mid \cdot d(p, C_j), \text{ if } \mathrm{C}_i \in \mathrm{LC} \quad \text{where} \quad \mathrm{p} \in \mathrm{C}_i \end{cases} \tag{2.9}$$

The CBLOF method is not local, i.e. different densities in the data are not taken into account. Small clusters far from other clusters are discarded. There is a need to define thresholds for small and far clusters. An unweighted cluster-based local outlier factor works better on the real-world dataset.

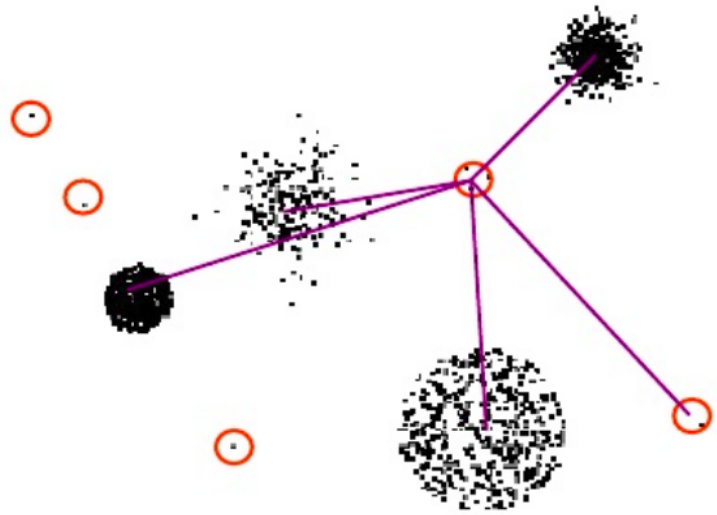


Figure 2.15: **Cluster-based local outlier factor.** Red circles represent an outlier.

2.6.6 Histogram-based Outlier Score

Histogram-based outlier score (HBOS) creates a histogram with a fixed or a dynamic binwidth with which it calculates an outlier score [43]. A separate univariate histogram

for every feature (column) in the dataset is calculated. In the static mode, the binwidth will be same, i.e. every bin will be equally distributed over the value range. Whereas, In the dynamic mode, the binwidth can vary. In case of dynamic mode, It is possible to specify a minimum number of cases (rows) to be contained in a bin.

To calculate the outlier score, firstly the histograms are normalised to one concerning their heights. Then the score is inverted, due to which the anomalies will have a high score and normal cases will have a low score. The HBOS score calculation is given below:

$$HBOS(p) = \sum_{i=0}^{d} \log\left(\frac{1}{hist_i(p)}\right) \tag{2.10}$$

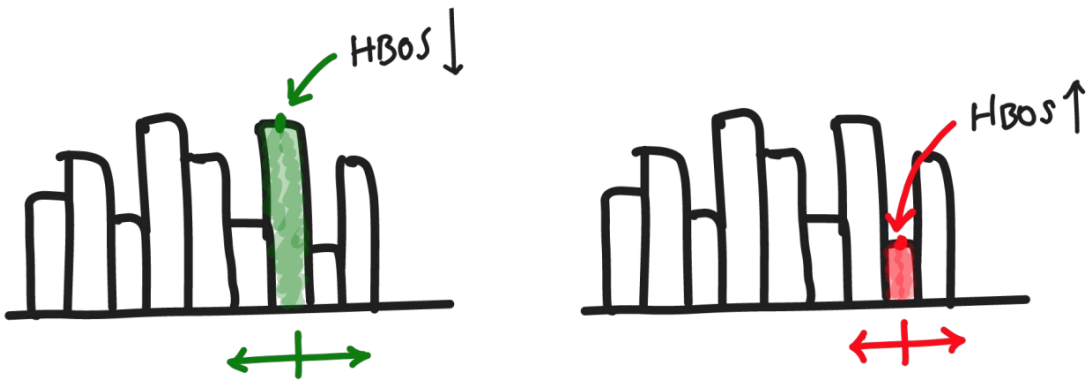


Figure 2.16: **Histogram-based Outlier Score.** High HBOS score represents an outlier and vice-versa.

2.6.7 k-Nearest Neighbours

The fundamental assumption in a k-nearest neighbour based technique is that the normal points have close neighbours while anomalies are situated far from other points. It generally consists of a two-step approach, i.e. firstly, compute the neighbourhood for each data record, and lastly analyse the neighbourhood to determine whether the data point is anomalous or not. In a Distance-based method, Anomalous data points which

are most distant from the other data points. In a Density-based method, Anomalous data points are in low-density regions [44].

Distance-based kNN approach

1. For each data point d, Calculate the distance to the k^{th} nearest neighbour d_k.
2. All data points are sorted as per the distance d_k.
3. Usually, Outliers are those data points that have the largest distance d_k and are located in the more sparse neighbourhoods. So, the data points which have top n% d_k are characterised as outliers, where n is a user-dependent parameter.
4. This method is not suited for the datasets which have modes with varying densities.

Average kNN or kNN Sum

This method uses the average distance to k nearest neighbours as the outlier score or sum of all k distances [45].

Median kNN

This method uses the median distance to k nearest neighbours as the outlier score.

2.6.8 Angle-based Outlier Detection (ABOD) and FastABOD

The rationale behind Angle-based Outlier Detection (ABOD) [46] is that the angles are more stable than distances in a high dimensional space. This can be compared to the popularity of cosine-based similarity measures for text data. In the below figure 2.17, the Object o is an outlier if the most of the other objects are located in a similar direction. Moreover, Object o is not an outlier if many other objects are located in different directions.

The underlying assumptions for ABOD method are given below:

1. Generally, Outliers are at the border of a data distribution

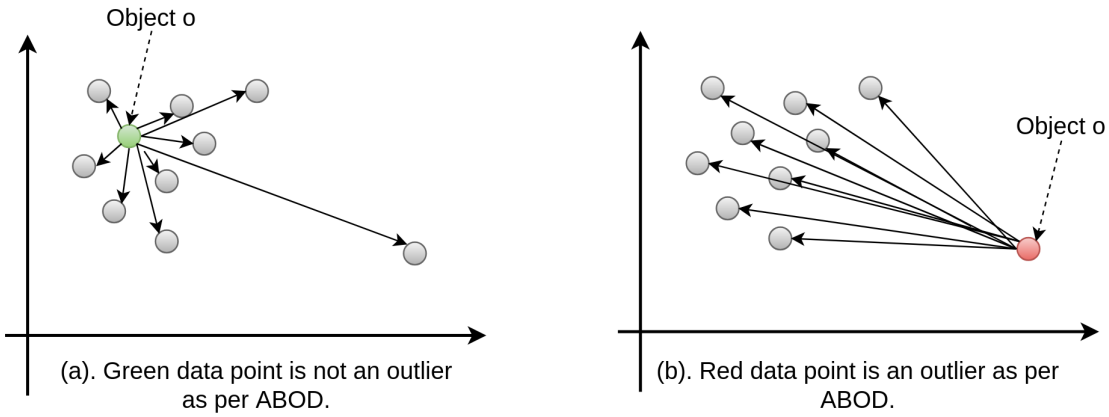


Figure 2.17: **Angle-based Outlier Detection Intuition.** It represents an example of Angle-based outlier.

2. Moreover, the Non-anomalous data points are in the centre of the data distribution.

ABOD Model

Consider the angle between px and py for two data points x and y from the dataset for the point p (see 2.18). Also, Consider the spectrum of these angles as given in figure 2.19. The broadness of such spectrum for a given point p is a score for the outliers, i.e. the variance of the angle spectrum is measured weighted by the corresponding distances. Weighing by corresponding distances is vital as angles are less reliable for lower dimensional data sets.

$$ABOD(p) = VAR_{x,y \in D} \left[\frac{\langle \overrightarrow{xp}, \overrightarrow{yp} \rangle}{\| \overrightarrow{xp} \| \cdot \| \overrightarrow{yp} \|} \right] \tag{2.11}$$

Then, a small ABOD score represents an outlier and vice versa.

FastABOD

The naive ABOD algorithm is in O(n^3). Fast Angle-Based Outlier Detection using an approximation, i.e. FastABOD is an improvement over the ABOD algorithm. FastABOD is based on random sampling for mining top-n outliers.

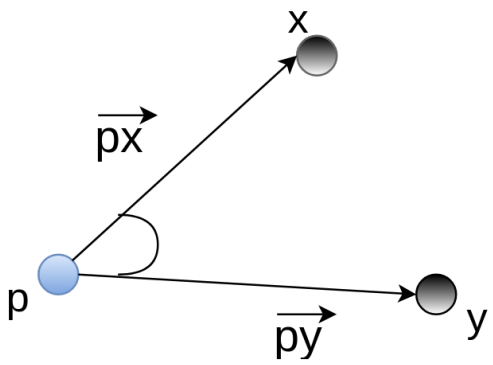


Figure 2.18: **ABOD Model Consideration.** It represents angle between px and py for a given point p, where x and y are also data points.

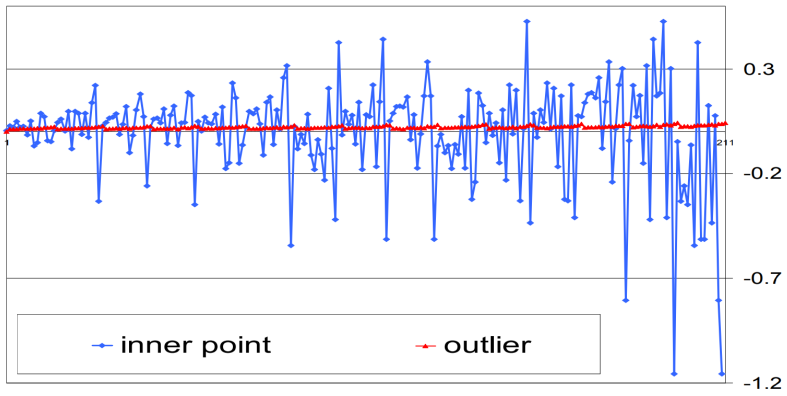


Figure 2.19: **ABOD Spectrum.** It represents variance of the angle spectrum for a given example point p.

2.6.9 Isolation Forest

Instead of profiling normal data points, the Isolation Forest [47] explicitly determines the anomalous data points. Isolation forest is an ensemble based on the decision trees. This method can be scaled up to work with large and high-dimensional datasets. Firstly, the partitions are created by randomly selecting a feature. Then, the random split value is selected based on the selected features minimum and maximum value.

In principle, the outliers are generally less frequent than the normal data points, i.e. the outliers lie very far from regular data points in a feature space. The outliers

are identified closer to the root of a tree with fewer splits using random sampling.

The below figure 2.20 represents the intuition of identifying normal vs outlier data points [47]. As shown, the normal data point (on the left) would require fewer splits than an outlier data point (to the right).

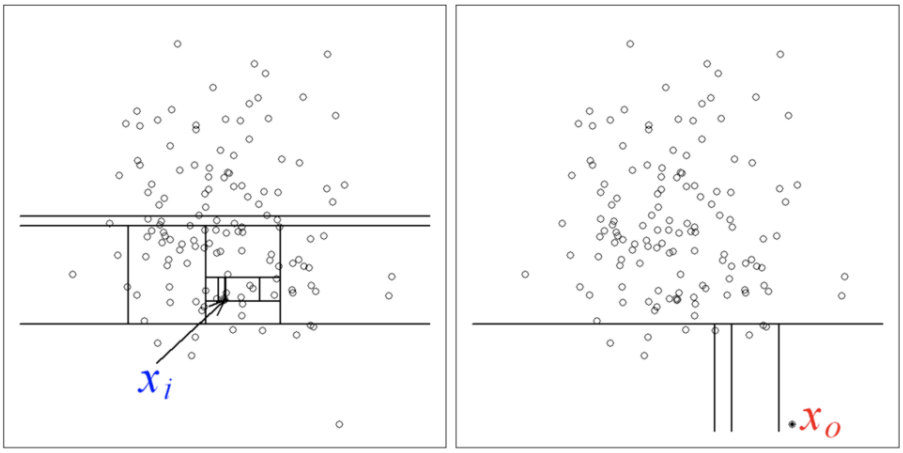


Figure 2.20: **Isolation Forest.** Identifying normal vs. abnormal observations.

An anomaly score is required to check if the data point is an outlier or not. Score calculation (cite) for Isolation Forest is given below:

$$s(x, n) = 2^{\frac{-E(h(x))}{c(n}} \tag{2.12}$$

In the above equation (2.12), h(x) refers to the path length of an observation x, c(n) refers to the average path length of an unsuccessful search in a Binary Search Tree, and lastly, n refers to the number of external nodes.

Each case (sample) will be assigned an anomaly score based on the above equation (2.12), where a score closer to 1 indicates an anomalous sample, score much below than 0.5 will represent a normal sample. If all sample scores are near to 0.5, then the entire data set does not seem to have clearly distinct anomalies.

2.6.10 Feature Bagging

The idea is to build various detectors based on randomly selected features. Basically, In Feature Bagging [48], Outlier detection (e.g. Local Outlier Factor) is executed in several random feature subsets (subspaces). Then, Combine results in an ensemble. Feature bagging can provide efficiency gains by performing computations on random feature subsets (subspaces). It also provides gains in model effectiveness through an ensemble technique. It is not a specific approach for high-dimensional data, but its application to high-dimensional data with the improved combination was introduced [49] in 2010.

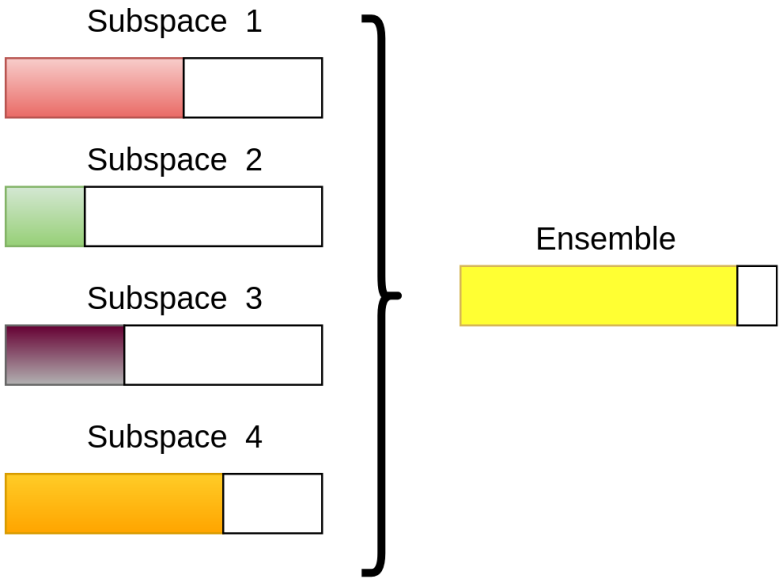


Figure 2.21: **Feature Bagging.** Outlier detection is performed on subspaces and results are combined in an ensemble.

2.7 Autoencoders

An autoencoder [50] is an artificial neural network, and it used to learn a representation (encoding) for a set of data points. Autoencoder learns an approximation of the

identity function of the dataset, i.e. $Id : \mathcal{X} \rightarrow \mathcal{X}$.

Feedforward is the simplest form of an autoencoder, i.e., a non-recurrent neural network similar to a multilayer perceptron (MLP). This form of an autoencoder contains one or more hidden layers connecting an input and an output layer. There are two major differences between an autoencoder and MLPs, i.e. the output layer has the same number nodes as that of the input layer and an autoencoder is trained to reconstruct its own inputs $\mathcal{X}$, whereas MLPs are trained to predict target value $\mathcal{Y}$ given input $\mathcal{X}$. Therefore, Autoencoders are called unsupervised learning models.

An autoencoder has two parts, i.e. an encoder and the decoder, where encoder and decoder can be defined as transitions ϕ and ψ, such that:

$$\phi : \mathcal{X} \rightarrow \mathcal{F} \tag{2.13}$$

$$\psi : \mathcal{F} \rightarrow \mathcal{X} \tag{2.14}$$

$$\arg\min_{\phi,\psi} ||X - (\psi \circ \phi)X||^2 \tag{2.15}$$

An autoencoder takes the input $\mathbf{x} \in R^d = \mathcal{X}$ and maps this onto $\mathbf{z} \in R^p = \mathcal{F}$. The number of nodes d in the input layer should be less the number of nodes p in a hidden layer, i.e. (p < d). The following is the simplest case.

$$\mathbf{z} = \sigma_1(\mathbf{W}\mathbf{x} + \mathbf{b}) \tag{2.16}$$

The above equation (2.16) is referred to as latent representation. σ_1 refers to an element-wise activation function. Activation functions used are ReLU (aka. rectifier), tanh (hyperbolic tangent), or sigmoid. Then, $\mathbf{z}$ is mapped onto the reconstruction $\mathbf{x}'$ which is of the same shape as $\mathbf{x}$:

$$\mathbf{x}' = \sigma_2(\mathbf{W}'\mathbf{z} + \mathbf{b}') \tag{2.17}$$

Generally, the squared errors loss function is used with autoencoders and is given as below:

$$\mathcal{L}(\mathbf{x}, \mathbf{x}') = ||\mathbf{x} - \mathbf{x}'||^2 = ||\mathbf{x} - \sigma_2(\mathbf{W}'(\sigma_1(\mathbf{W}\mathbf{x} + \mathbf{b})) + \mathbf{b}')||^2 \tag{2.18}$$

Conventional deep learning models blow up the feature space to learn non-linearities and identify subtle interactions in the data, whereas autoencoders aim at reducing feature space and keeping essential features. Autoencoders represent nonlinearities, whereas PCA can only represent linear transformations. Also, the autoencoders can be used for finding a low-dimensional representation of the input data $\mathcal{X}$. Some of the features may be redundant or correlated due to which much time is wasted in processing the model and also introduces overfitting. Therefore, It is ideal to use the feature which is essential. If the reconstruction of $\mathcal{X}$ is nearly accurate, then it can be interpreted that the low-level representation is good.

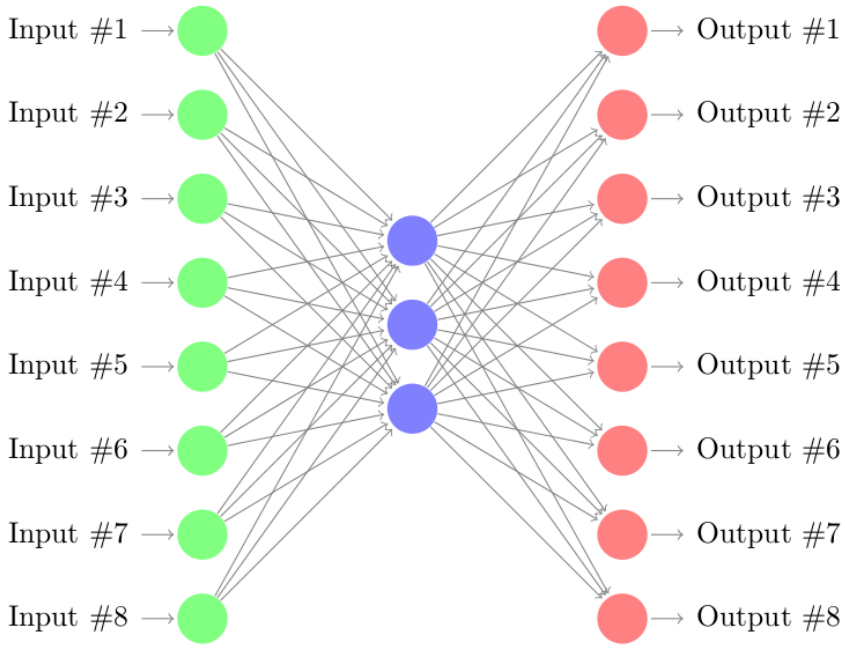


Figure 2.22: **Autoencoder Schematic Structure.** Input vector length d=8 and an attempt to reduce the dimensionality of the inputs to p=3.

Autoencoders for Anomaly Detection

Firstly, An autoencoder is trained on $\mathcal{X}_{train}$ with good regularization (i.e. use recurrent if $\mathcal{X}$ is a time series data). Then, Evaluate the model on the validation set $\mathcal{X}_{val}$ and

visualize the sorted reconstruction error plot.

2.8 Is the Research Problem Solved?

There is no definite solution to the problem/application that we are addressing in this Thesis. There are various solutions available but none of them is a perfect solution for this type of problem. We have solutions ranging from Semi-Supervised learning to Unsupervised machine learning methods. The first is to perform descriptive analysis of the sensor data and then apply the relevant methods inspired by the work done by Markus Goldstein [51].

2.9 Speaking with other Researchers

Rob Romijnders, Machine Learning Researcher from Eindhoven University of Technology suggested me some methods which are given below in the order of increasing complexity:

1. Treat all sensor readings as a bag of sensor readings. Average them per time series. On all those averages, We fit a Naive Bayes Classifier. The NBC will tell us the probability of a new sample. The probability will be high for a typical input and low for an anomaly.

2. Collapse all sensor readings and fit a Gaussian Mixture model to it. Again, the probability of a new sample will tell us about the the outlier.

3. We fit a Hidden Markov model to the time series. The hidden markov model takes into account all the autocorrelations in your time series. (We can also consider a Kalman filter or another Linear Dynamical Systems (LDS) if we prefer). Again, the probability of a new sample will tell us about the outlier.

4. We can fit a neural network in the form of an auto encoder. Researchers use the reconstruction loss as an indication for an outlier. The research says that this is the least stable option. It might give us good results, but it has a high chance of failing. Even if we decide to implement this method, We should use base models such as NBC and GMM to compare our results.

Pankaj Malhotra, Researcher at Tata Consultancy Services Innovation Labs, recommended first to visualise the data using Plotly [52] and to build ML models like RNN/LSTM possibly using Keras [53] library. He suggested that Semi-supervised learning may not be the best idea for anomaly detection. Since there are usually few anomalous samples, then the safest way to use semi-supervised learning would be to define a threshold over an anomaly score. Moreover, Anomaly score is obtained from a model trained on normal data. At last, Building an LSTM autoencoder on complete data was recommended where the data points with highest reconstruction error are most likely to be anomalous samples [54].

Rami Krispin, Data Scientist at iCloud, recommended me to use TStudio package (cite) for analysing the time series data. Later, I discovered that the TStudio package works with only monthly or quarterly sampled data for now. Also, Performing correlation analysis using ACF and PACF functions was recommended.

Chapter 3

Data Analysis

3.1 Dataset Description

The dataset contains data from multiple locations from the same site. Each location in a site is monitoring the level of gases such as Methane (CH_4), Carbon Dioxide (CO_2), and Oxygen (O_2). The importance of measuring the level of gases correctly and making sure that the data is anomaly-free is provided in the section 1.3. The dataset description was made clear to us by Dr Fiachra Collins, Ambisense Ltd. From the domain knowledge, It is clear that the behaviour of methane and carbon dioxide is opposite to the behaviour of oxygen. It means that If oxygen is rising, then methane and carbon dioxide level falls. The behaviour of methane is dependent on the atmospheric pressure. Atmospheric pressure (BaroPres field in the dataset) is ratiometric, i.e. it is dependent on the battery level readings. The electrical output signal of the pressure sensor depends on the battery supply voltage. This is a common feature for unamplified sensors and sensors that do not have built-in power-supply regulation. The span output of an unamplified device is essentially ratiometric (or proportional) to the excitation voltage applied. More information about all the parameters in the dataset and their expected behaviour and inter-dependencies are provided in the below tables (cite). Only the parameters such Methane (CH_4), Carbon Dioxide (CO_2), Oxygen (O_2), BaroPres (atmospheric pressure) and Batt (battery level) are considered for a machine learning model, as the other parameters are not significant.

Parameters	Description
senseDate	Time-stamp of data
CH_4	Methane concentration
CO_2	Carbon dioxide concentration
O_2	Oxygen concentration
Temp	Temperature sensor (of sampled gas)
Humidity	Humidity (of sampled gas)
GaugePres	Gauge pressure (difference in pressure between inside of well and atmosphere)
PumpPres	Gauge pressure when pumping
BaroPres	Atmospheric pressure (not adjusted for altitude)
Batt	Battery level
Packet	Sample index number
metTemp	Outdoor air temperature
metRH	Outdoor humidity
windDir	Wind direction
windKPH	Wind average speed
windGustKPH	Wind gust speed
baroMB	Atmospheric pressure (adjusted for altitude, sea-level compensated)
precip1Hr	Rainfall in last hour
precipToday	Rainfall in last 24 hours
BHflow_peakVal (L/hr)	Flow-rate of gases between well and atmosphere, peak reading in 10min sample
BHflow_stStVal (L/hr)	Flow-rate of gases between well and atmosphere, steady-state reading after 10min

Table 3.1: **Sensor & Parameter Description.** Describes the usage of different parameters in the system.

Parameters	Unit	Range
senseDate	dd/mm/yyyy hh:mm	
CH_4	%vol	0-100
CO_2	%vol	0-100
O_2	%vol	0-22
Temp	C	-50 to +50
Humidity	%RH	0-100
GaugePres	mB	-120 to +50
PumpPres	mB	-120 to +50
BaroPres	hPa = mB	850 - 1150
Batt	V	5.5 - 6.8
Packet		3 - 1599
metTemp	C	-50 to +50
metRH	%RH	0-100
windDir		0-360
windKPH	km/h	0-200
windGustKPH	km/h	0-200
baroMB	hPa = mB	850 - 1150
precip1Hr	mm	0-100
precipToday	mm	0-2400
BHflow_peakVal (L/hr)	L/hr	-7 to 60
BHflow_stStVal (L/hr)	L/hr	-7 to 60

Table 3.2: **Sensor & Parameter Range.**

Parameters	Expected Behaviour
senseDate	Not applicable
CH_4	Fluctuates, often related to CO_2 levels, often correlated with atm pressure
CO_2	Fluctuates, often related to CH_4 levels, often correlated with atm pres
O_2	Behaves opposite to CH4 & CO2 due to air displacing the other gases
Temp	Found to fail in the field: step-change to > 80
Humidity	Occasionally fails alongside the temperature sensor
GaugePres	Typically no huge variation in these applications
PumpPres	Typically -30mB less thangauge pressure due to suction pressure induced by pump
BaroPres	Varies with the weather. Fixed offset from "baroMB" due to altitude
Batt	Decreases as power is consumed, increases when solar charging or fresh battery replacement
Packet	Increments one per sample taken
metTemp	Acquired from nearby weather station
metRH	Acquired from nearby weather station
windDir	Acquired from nearby weather station
windKPH	Acquired from nearby weather station
windGustKPH	Acquired from nearby weather station
baroMB	Acquired from nearby weather station
precip1Hr	Occasionally dubious data as weather station doesn't have sufficent quality rain gauge
precipToday	Occasionally dubious data as weather station doesn't have sufficent quality rain gauge
BHflow_peakVal (L/hr)	High flows may coincide with higher CH4 or CO2 levels due to gas emission from subsurface
BHflow_stStVal (L/hr)	Steady-state typically lower values than peak readings

Table 3.3: **Sensor & Parameter Expected Behaviour.**

Parameters	Dependencies
senseDate	Not applicable
CH_4	GaugePres & AtmPres (therefore battery), also CO2 compensation
CO_2	GaugePres, AtmPres (therefore battery)
O_2	None
Temp	None
Humidity	None
GaugePres	Battery, ratiometric sensor when power < 5.5V
PumpPres	Battery, ratiometric sensor when power < 5.5V
BaroPres	Battery, ratiometric sensor when power < 5.5V
Batt	None
Packet	None
metTemp	None
metRH	None
windDir	None
windKPH	None
windGustKPH	None
baroMB	None
precip1Hr	None
precipToday	None
BHflow_peakVal (L/hr)	None
BHflow_stStVal (L/hr)	None

Table 3.4: **Sensor & Parameter Dependencies.** Description of inter-dependencies between different sensors is provided.

3.2 Time Series Analysis

3.2.1 Descriptive Analysis

The dataset chosen for Descriptive Analysis is BH11D site (see Appendix). The dataset contains many parameters such as CH_4, CO_2, O_2, Batt, BaroPres and others. Only CH_4, CO_2, O_2, Batt and BaroPres variables were analysed as they are the main contributing variables to detect for anomalies. Weather-related data in the dataset cannot be trusted, As they were from a third-party service. Sometimes, the readings for the weather data is faulty. Jitter plot and Box plot for each of these parameters is visualised and studied. Jitter plot is used to infer the density of the data.

Methane (CH_4) Sensor

The CH_4 parameter has a mean value of 0.82, a median of 0.30, a standard deviation of 1.68, and lastly minimum and a maximum value of 0.00 and 54.82. Definitely, the maximum value methane concentration is 54.82 which seems to be anomalous statistically, as the mean value of the parameter values is 0.82.

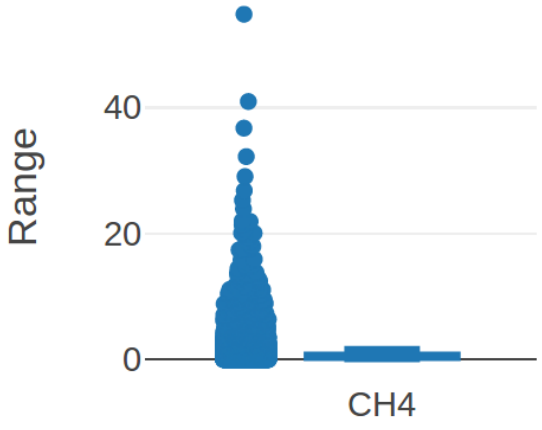


Figure 3.1: CH_4 **Concentration.** Jitter plot and Box plot for CH_4.

Carbon Dioxide (CO_2) Sensor

The CO_2 parameter has a mean value of 0.89, a median of 0.86, a standard deviation of 0.99, and lastly minimum and a maximum value of 0.04 and 30.73.

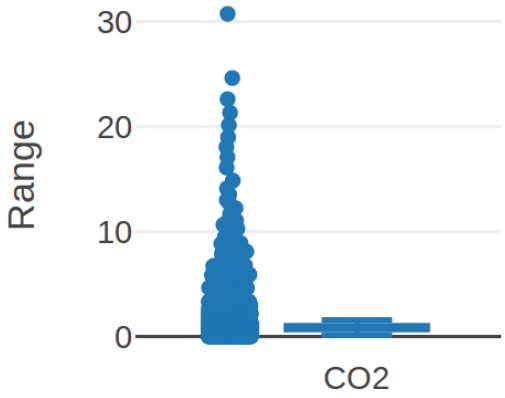


Figure 3.2: CO_2 **Concentration.** Jitter plot and Box plot for CO_2.

Oxygen (O_2) Sensor

The O_2 parameter has a mean value of 18.44, a median of 18.42, a standard deviation of 1.26, and lastly minimum and a maximum value of 3.76 and 21.87.

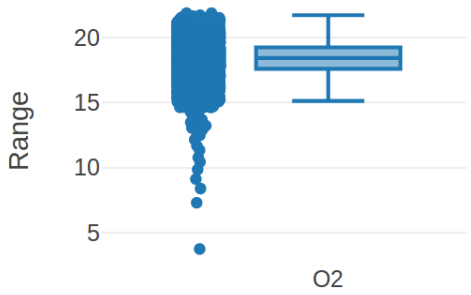


Figure 3.3: O_2 **Concentration.** Jitter plot and Box plot for O_2.

Atmospheric Pressure (BaroPres) Sensor

The BaroPres parameter has a mean value of 1015.66, a median of 1016.65, a standard deviation of 12.72, and lastly minimum and a maximum value of 819.22 and 1042.76.

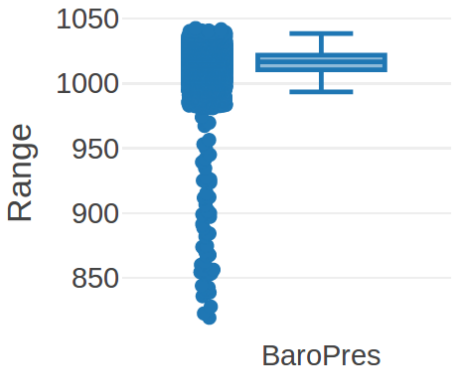


Figure 3.4: **BaroPres (Atmospheric Pressure).** Jitter plot and Box plot for Atmospheric Pressure.

Battery Parameter

The Batt parameter has a mean value of 6.14, a median of 6.02, a standard deviation of 0.34, and lastly minimum and a maximum value of 4.10 and 7.33.

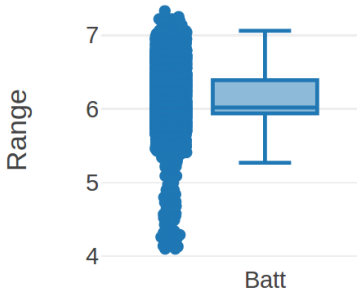


Figure 3.5: **Battery Readings.** Jitter plot and Box plot for Battery readings.

3.2.2 Individual Sensor Time Series Plots

Methane (CH_4) Sensor

The changes in the behaviour of the methane gas production for the BH2D site (see Appendix) is clearly visible in the figure 3.6. There are sudden spikes in the readings from the CH_4 sensor which are probable anomalies.

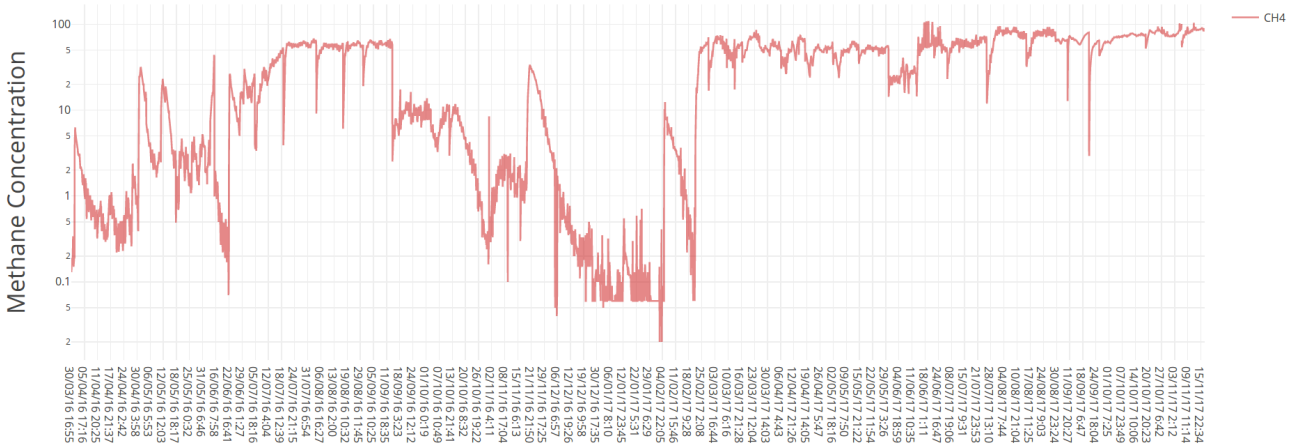


Figure 3.6: **Methane (CH_4) Sensor Time Series Visualization.**

Carbon Dioxide (CO_2) Sensor

The changes in the behaviour of the carbon dioxide gas production for the BH2D site is provided in the figure 3.7.

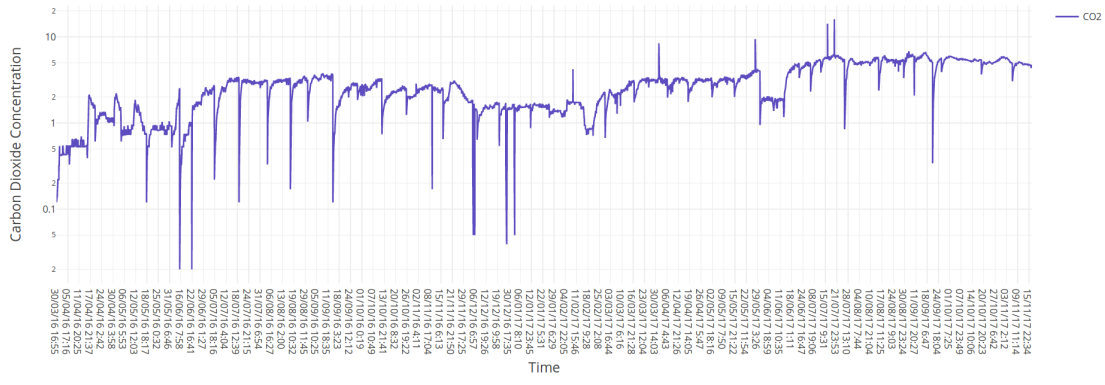


Figure 3.7: **Carbon Dioxide CO_2 Time Series Visualization.**

Oxygen (O_2) Sensor

The changes in the behaviour of the oxygen gas production for the BH2D site is provided in the figure 3.7.

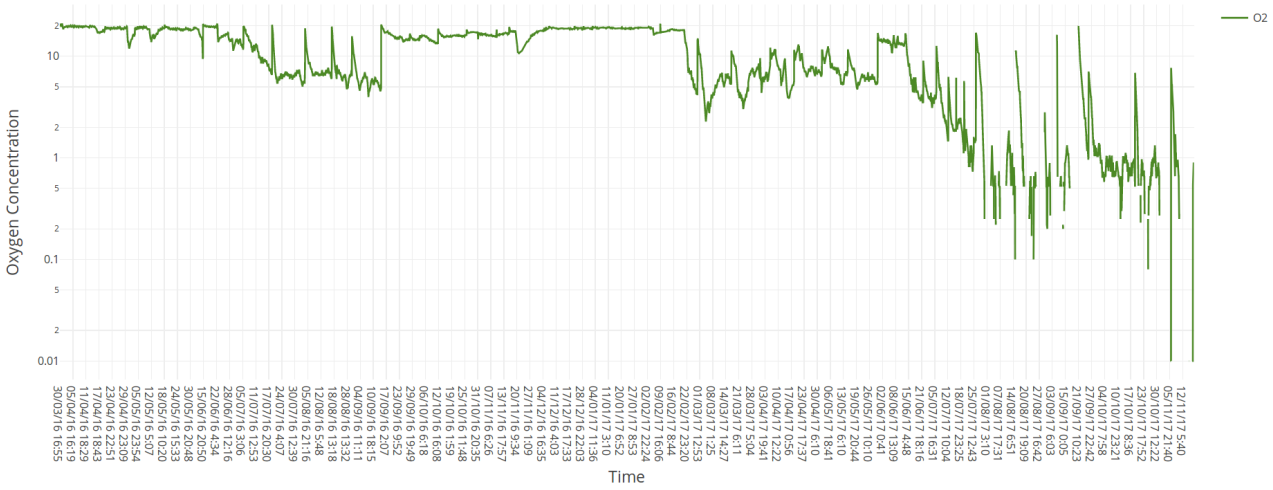


Figure 3.8: **Oxygen O_2 Time Series Visualization.**

BaroPres (Atmospheric Pressure) Sensor

The changes in the level of atmospheric pressure for the BH2D site is provided in the figure 3.10.

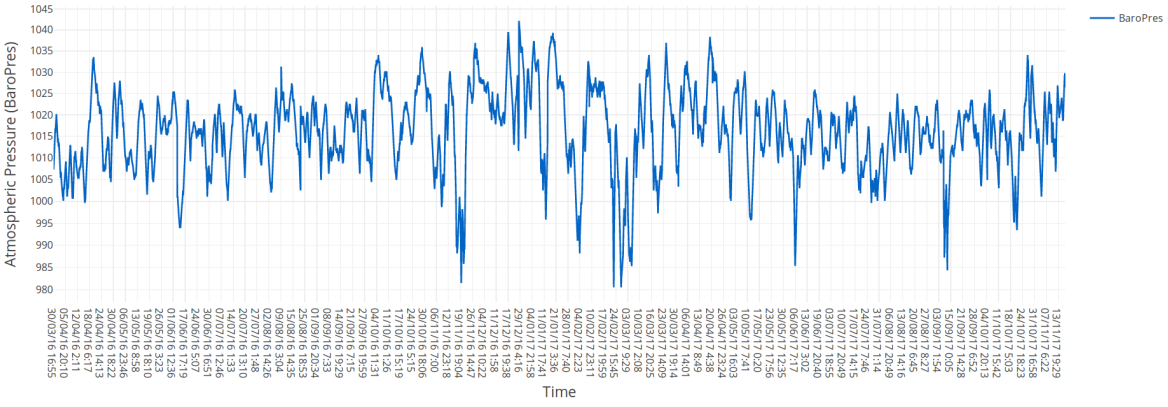


Figure 3.9: **BaroPres (Atmospheric Pressure) Time Series Visualization.**

Battery Level Readings

The changes in the battery level readings for the BH2D site is provided in the figure 3.10. The battery mostly has the values between 6.5V to 7V and the continuous drop in the battery readings indicates a need for battery replacement.

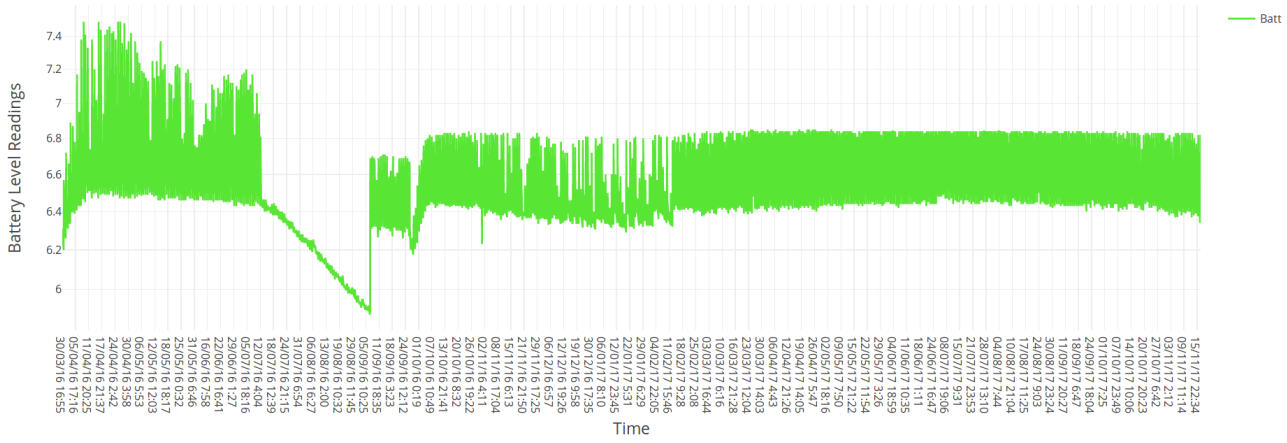


Figure 3.10: **Battery Level Time Series Visualization.**

3.2.3 Sensor System Behaviour Analysis

As given in the figure 3.11, the system is observed to go through various phases. For the system behaviour analysis, BH2D site (see appendix) is considered. The phases are as follows:

- Phase A: There is an increasing trend in the parameter CH_4.
- Phase B: The system is entering into a new normal. That is, the normal behaviour of the system is changed. This change applies to each parameter, i.e. CH_4, CO_2, O_2 are changing their behaviour in this particular phase.
- The parameter CH_4 is showing a decreasing trend. Continuously rising or falling levels of CH_4 is a concern for the landfill site. Continuously falling levels applies to other sensors also such as CO_2, O_2 and BaroPres.
- Phase D: Now, the CH_4 again entered into a new normal. That is, the normal behaviour of the system is changed.

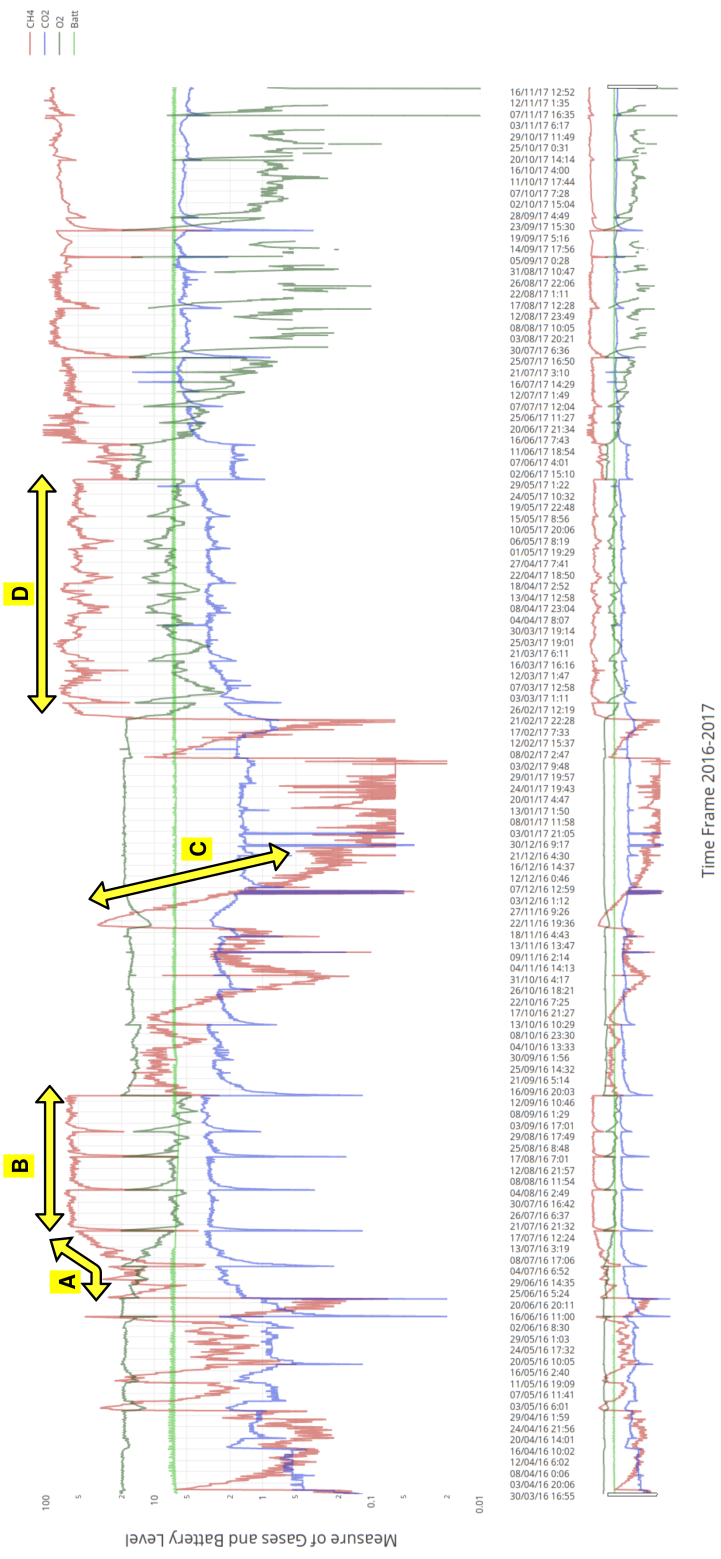


Figure 3.11: **Sensor System Behaviour.** As observed the parameter CH_4 is going through A, B, C and D phases.

3.2.4 Moving Average Analysis

Moving average analysis uses a window size of 72. As there is a need of at least previous 3-day data to be able to decide whether a sample is anomalous or not. Each day approximately 24 samples are collected.

Methane (CH_4) Sensor

A moving average with window size 72 is fitted to methane concentration for the BH2D site (see Appendix) is provided in the figure 3.12.

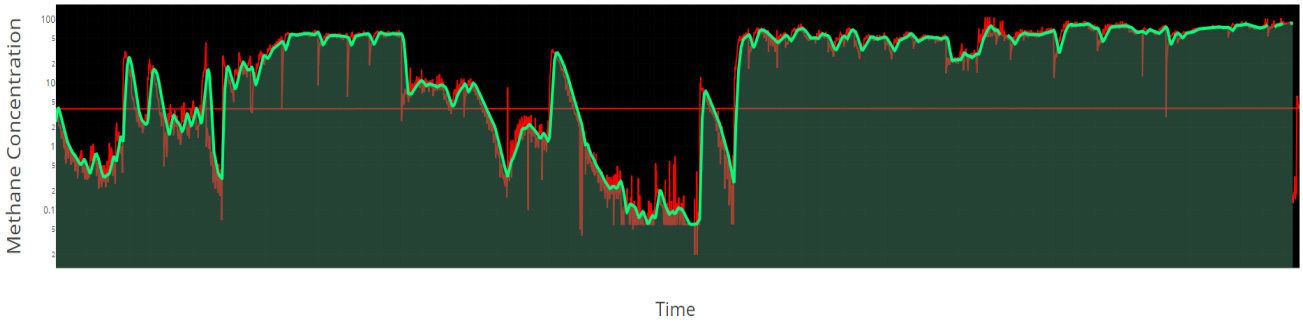


Figure 3.12: **Methane (CH_4) Moving Average Analysis.** Red line plot denotes the actual methane concentration levels and green line plot denotes the moving average.

Carbon Dioxide (CO_2) Sensor

A moving average with window size 72 is fitted to carbon dioxide concentration for the BH2D site and is provided in the figure 3.13.

Oxygen (O_2) Sensor

A moving average with window size 72 is fitted to oxygen concentration for the BH2D site and is provided in the figure 3.14.

BaroPres (Atmospheric Pressure) Sensor

A moving average with window size 72 is fitted to atmospheric pressure readings for the BH2D site and is provided in the figure 3.15. The sudden drop in moving average denotes missing data.

Figure 3.13: **Carbon Dioxide CO_2 Moving Average Analysis.** Red line plot denotes the actual carbon dioxide concentration levels and green line plot denotes the moving average.

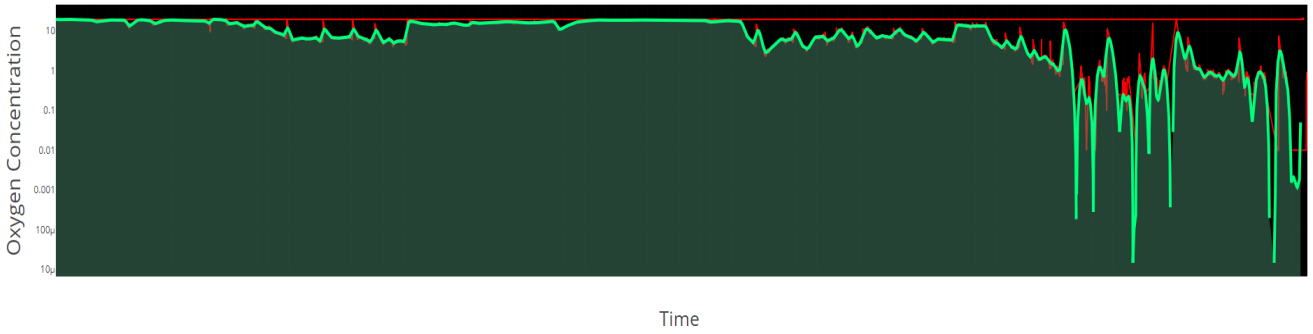


Figure 3.14: **Oxygen O_2 Moving Average Analysis.** Red line plot denotes the actual carbon dioxide concentration levels and green line plot denotes the moving average.

Battery Level Readings

A moving average with window size 72 is fitted to battery level readings for the BH2D site and is provided in the figure 3.16.

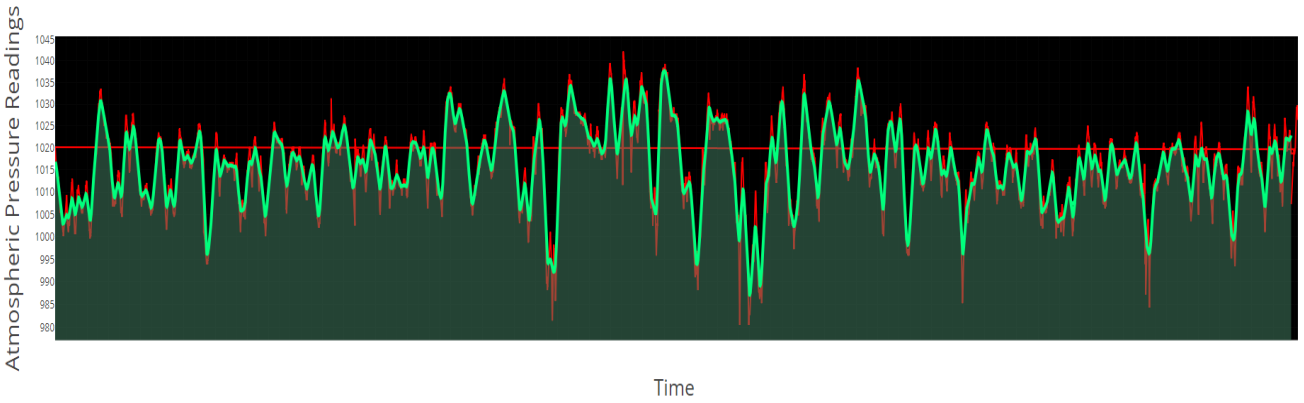


Figure 3.15: **BaroPres (Atmospheric Pressure) Moving Average Analysis.** Red line plot denotes the actual atmospheric pressure levels and green line plot denotes the moving average.

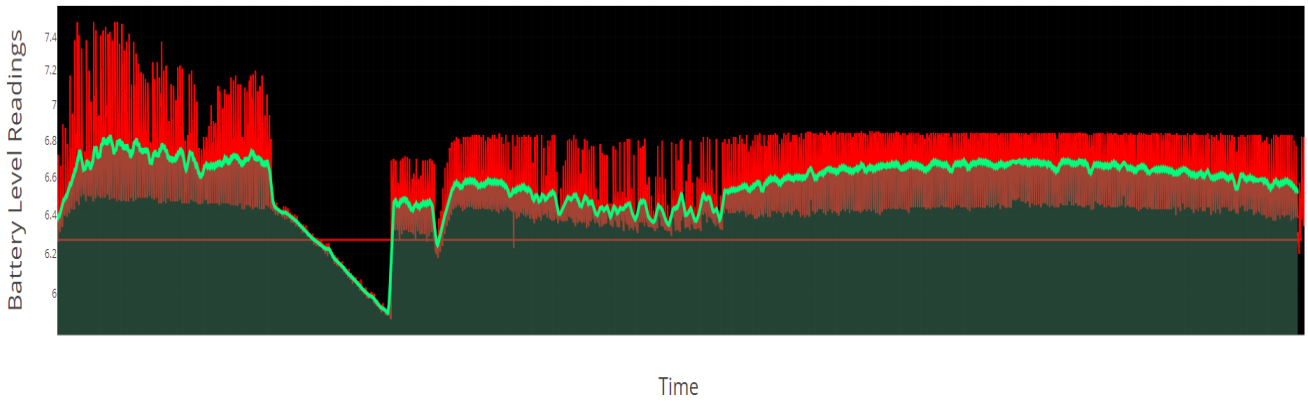


Figure 3.16: **Battery Level Moving Average Analysis.** Red line plot denotes the actual battery voltage levels and green line plot denotes the moving average.

3.3 Environmental and Instrument Anomalies

Typically, An 'anomaly' means a deviation in sensor data away from the expected. There is a question mark for what all to expect from the sensor data. Hence, there is a demand from the end-clients to understand the gas behaviour. End-clients are in need of predictive analytics.

Furthermore, the anomalies are into two categories: environmentally-driven anomaly and instrument-driven anomaly. The concept of environmentally-driven anomaly and

instrument-driven anomaly are made clear to us by Dr Fiachra Collins, Ambisense Ltd.

3.3.1 Environmental-driven Anomalies

A significant relative change in sensor readings over time indicates the possibility of occurrence of an environmental-driven anomaly. The timestamps of the sensor readings are essential and need to be considered as the changes in different sensor readings typically coincide in time due to the environmental driver affecting them. See the highlighted yellow sections in the figure 3.17. Note the anomaly may not be specific to a single individual datapoint but instead, manifest over a short series of data points.

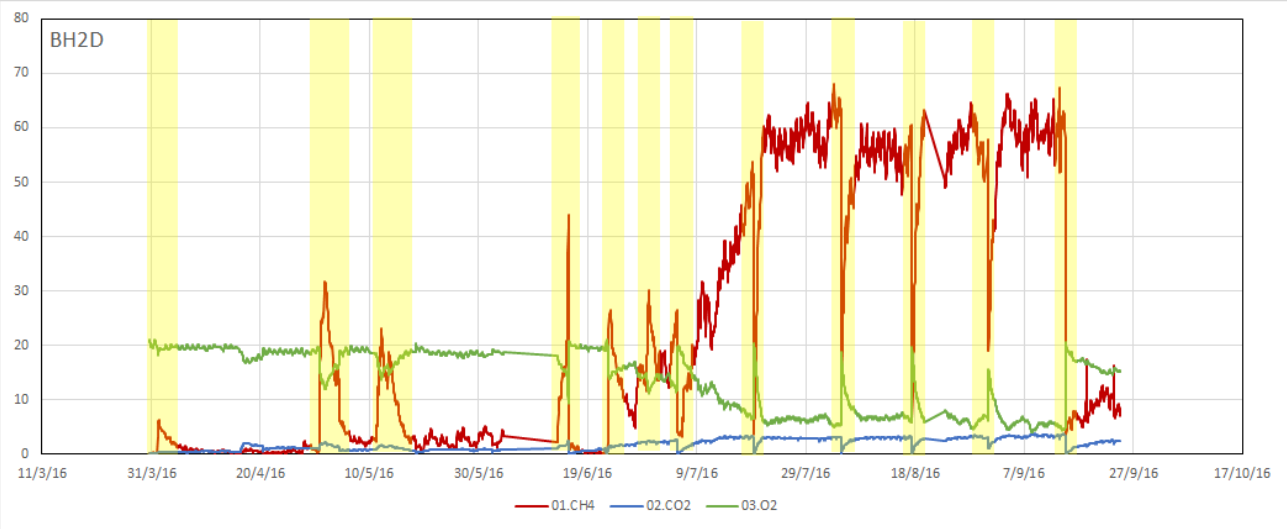


Figure 3.17: **Environmental-driven Anomaly Example.** Changes in sensor readings are due to the environmental changes in a landfill site.

3.3.2 Instrument-driven Anomalies

Instrument-driven anomalies are more difficult to identify from the sensor data due to the aforementioned environmental influences. Some parameters are easy to use for detecting instrument anomalies. For example, battery voltage < 5.5V or barometric pressure < 850 or >1150. However, these are simply achieved using thresholds.

However, Consider the graph below in the figure 3.18. As the concentration of CH_4 and CO_2 increases and displace the air, the level of O_2 should drop. Therefore there should be a negative straight line correlation between these parameters. Some anomalies to this expected correlation are highlighted with circles (shaded yellow region) in the figure 3.18.

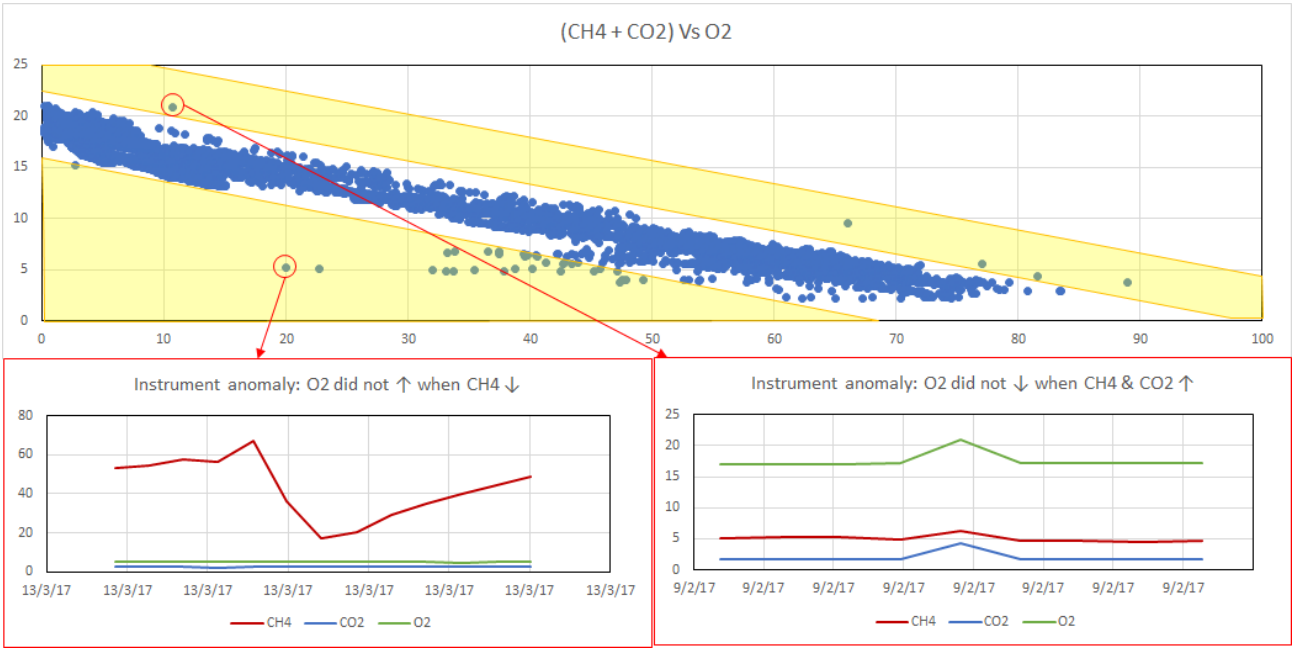


Figure 3.18: **Instrument-driven Anomaly Example.** Anomalies in the expected negative straight line correlation between $(CH_4 + CO_2)$ vs. O_2.

Chapter 4

Results and Evaluation

4.1 Gaussian Mixture Model

Gaussian Mixture Model [55] is a probabilistic model which represents normally distributed subpopulations with an overall population. A Gaussian Mixture Model does not require the information regarding a data point belonging to a subpopulation (cluster). As the subpopulation assignment of a data point is not known, this describes the nature of a Gaussian Mixture Model as unsupervised learning.

A Gaussian mixture model is parametrised by the values of the mixture component weights and the component means and variances or covariances. In a k-component Gaussian mixture model, the k^{th} component has a mean of μ_k and a variance of σ_k for the univariate case. For a multivariate case, It has a mean of $\overrightarrow{\mu}_k$ and a covariance matrix $\sum_k$.

- Univariate Model

$$p(x) = \sum_{i=1}^{K} \phi_i \mathcal{N}(x \mid \mu_i, \sigma_i) \tag{4.1}$$

$$\mathcal{N}(x \mid \mu_i, \sigma_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right) \tag{4.2}$$

$$\sum_{i=1}^{K} \phi_i = 1 \tag{4.3}$$

- Multivariate Model

$$p(x) = \sum_{i=1}^{K} \phi_i \mathcal{N}\left(\overrightarrow{x} \mid \overrightarrow{\mu_i}, \sum_i\right) \tag{4.4}$$

$$\mathcal{N}(\overrightarrow{x} \mid \overrightarrow{\mu_i}, \sum_i) = \frac{1}{\sqrt{(2\pi)^K \mid \sum_i \mid}} exp\left(-\frac{1}{2}(\overrightarrow{x} - \overrightarrow{\mu_i})^T \sum_{i^{-1}} (\overrightarrow{x} - \overrightarrow{\mu_i})\right) \tag{4.5}$$

$$\sum_{i=1}^{K} \phi_i = 1 \tag{4.6}$$

4.1.1 Labelling the Dataset

In this case, the Gaussian mixture model is used to determine whether a data point is an outlier or not. As manually labelling a dataset takes much time. The Gaussian mixture model is applied to determine the optimum number of components/clusters in the BH11D site dataset.

An optimum number of components k is selected by plotting a graph of BIC vs Number of components k. The Bayesian information criterion (BIC) or Schwarz criterion (also known as SBC, SBIC) [56] is a criterion used for model selection among a finite set of models where the model with the lowest BIC is preferred. BIC depends on the likelihood function, and it is closely related to the Akaike information criterion (AIC). Generally, it is possible to increase the likelihood by adding parameters, but it may result in overfitting of the model. Both Bayesian information criterion (BIC) and Akaike information criterion (AIC) help to reduce overfitting using a penalty term. In BIC, the penalty term larger than the penalty term of AIC.

4.1.2 Results

BIC vs Components for Different Covariance Types

Gaussian Mixture Models are compared with spherical, diagonal, full covariance matrices in the increasing order of performance. It is expected that a "full" covariance to perform better than other types of covariances. However, A full covariance is susceptible to overfitting on small-size datasets and fail to generalise well on the test data.

In selecting a best GMM model, It is better to have the lowest BIC value as possible. So, BIC vs Covariance Types graph helps select the best model. The graphs for BIC vs Covariance Type are given in the figures 4.1, 4.2, and 4.3. Moreover, the selected best model is given in the figure 4.4.

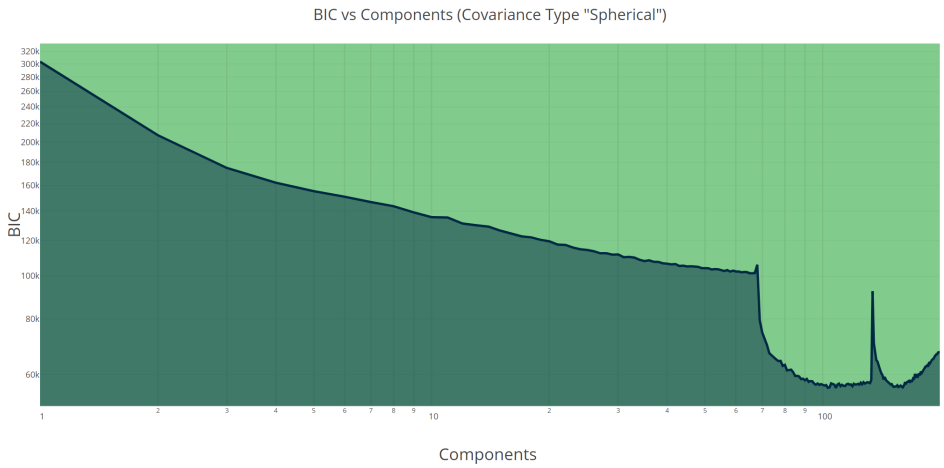


Figure 4.1: **BIC vs Components for Spherical Covariance Type.** BIC vs Components for covariance type "Spherical". As observed after component number 100, the value of BIC started to increase which refers to overfitting of GMM model.

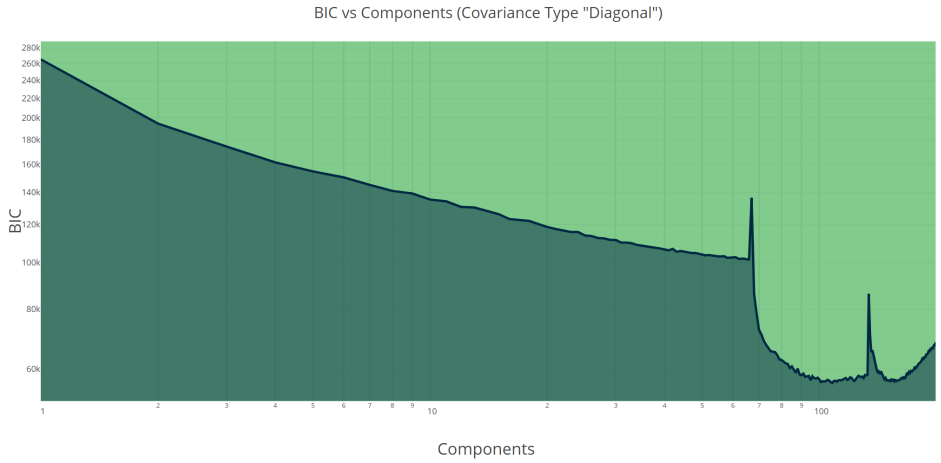


Figure 4.2: **BIC vs Components for Diagonal Covariance Type.** BIC vs Components for covariance type "Diagonal". As observed after component number 100, the value of BIC started to increase which refers to overfitting of GMM model.

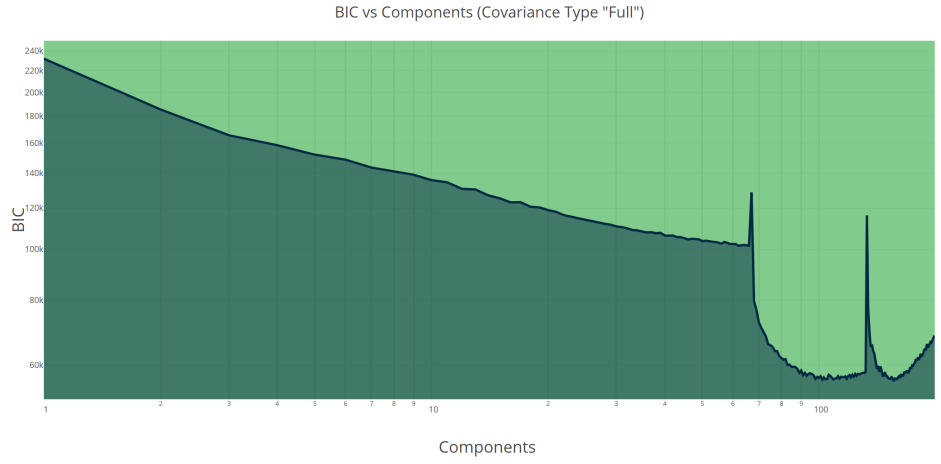


Figure 4.3: **BIC vs Components for Full Covariance Type.** BIC vs Components for covariance type "Full". As observed after component number 100, the value of BIC started to increase which refers to overfitting of GMM model.

Predicted Component Probabilities for Data Points

The figure 4.5 describes the predicted probabilities for all the data points in a dataset. The figure 4.5 also describes the stable and unstable component probability prediction. Unstable probabilities describe that the nature of data is changed. There is a possibility that the data points with low probabilities might be an outlier. It is always better

```
GMM(covariance_type='full',
init_params='wmc',
min_covar=0.001,
n_components=70, n_init=1,
n_iter=100, params='wmc',
random_state=None,
tol=0.001, verbose=0)
```

Figure 4.4: **Final GMM Model with 70 Components.** Selected GMM Model is of 'Full' covariance type.

to use GMM in conjunction with other models.

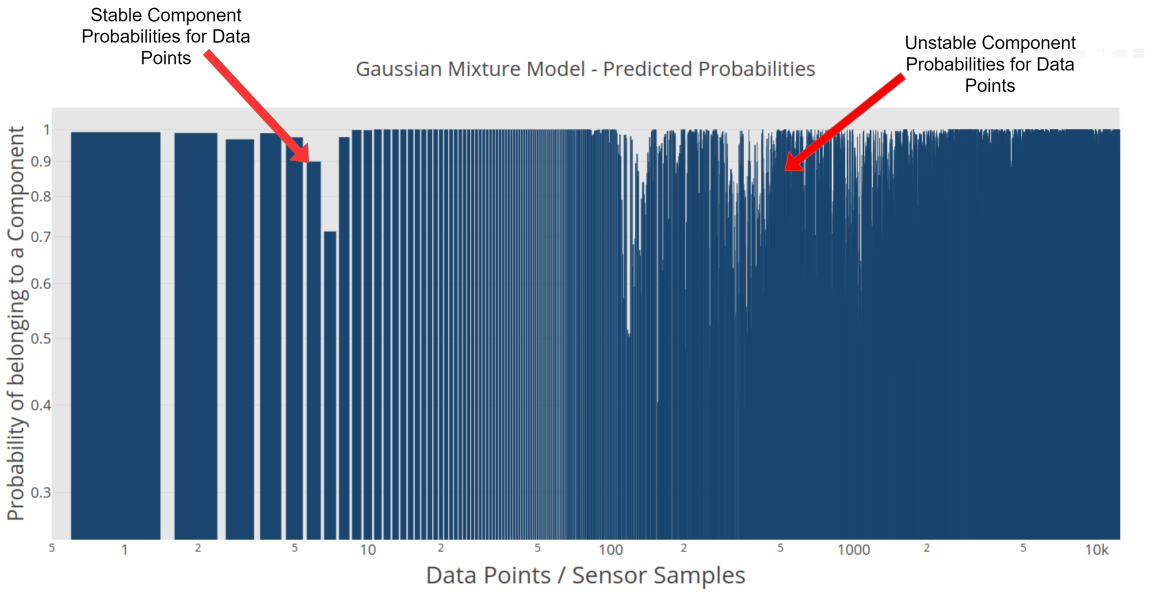


Figure 4.5: **Data Points with Component Probabilities.** Predicted probabilities for a data point gives a hint for an outlier.

Stable vs Unstable Components

The same above idea can be extended to stable components and unstable components. The stable components are those where the most data points belong to and vice versa. The figure 4.6 describes the phenomenon clearly.

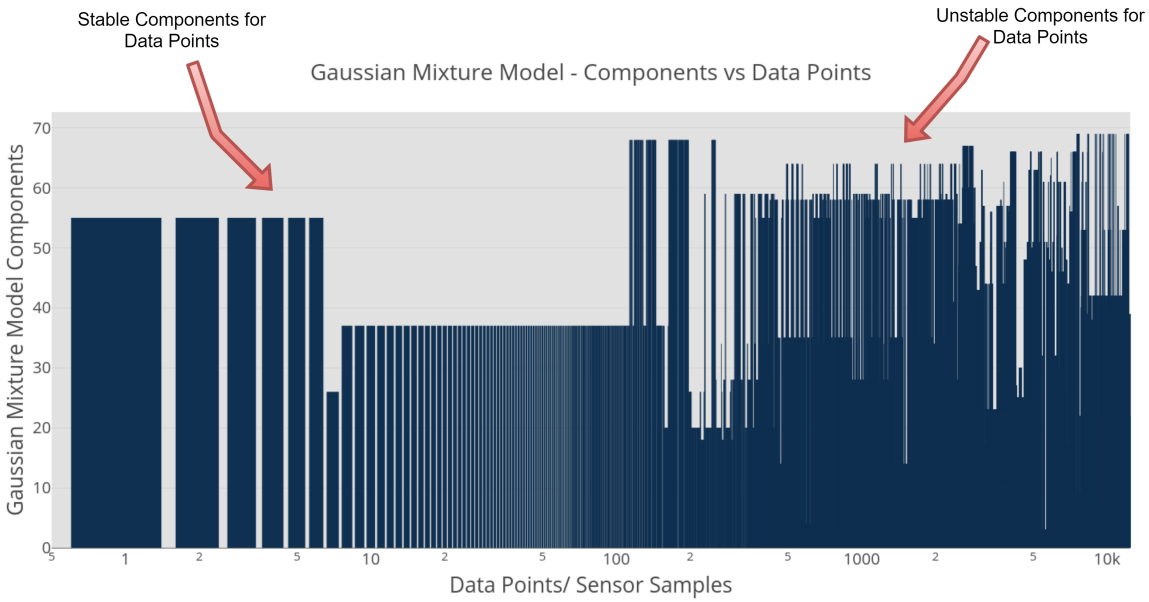


Figure 4.6: **Stable vs Unstable Components.** Unstable components define a possibility of an outlier component.

4.2 PyOD Benchmark Analysis

The BH11D labelled dataset is split into 60% as training data and 40% as testing data. Each algorithm from PyOD package is applied 20 times independently using different samplings. For each algorithm, the mean of 20 runs/trials helps in deriving the final result.

Following algorithms are tested for the BH11D site, referred to as Benchmark analysis. It helps in gauging the ability of existing algorithms to detect anomalies in the BH11D dataset.

1. Angle-based Outlier Detector (ABOD)
2. Cluster-based Local Outlier Factor
3. Feature Bagging
4. Histogram-based Outlier Detection (HBOS)

5. Isolation Forest
6. K Nearest Neighbors (KNN)
7. Local Outlier Factor (LOF)
8. Minimum Covariance Determinant (MCD)
9. One-class SVM (OCSVM)
10. Principal Component Analysis (PCA)

Evaluation Metrics

Three evaluation metrics used to determine the performance of an algorithm are below:

1. The area under the receiver operating characteristic (ROC) curve
2. Precision @ rank n (P@N)
3. Execution time

ROC Performance

The ROC performances of 10 different algorithms (average of 20 independent trials) are provided in the below figure 4.1. The Minimum Covariance Determinant (MCD) performed the best as it has higher AUC (Area Under the Curve) score, whereas Histogram-based Outlier Detection (HBOS) method performance is poor.

Precision@N Performance

For any given anomaly detection algorithm, there is a need to characterise how well an algorithm identifies all and only anomalies [57]. Precision measures how well an algorithm identifies only anomalies. It may be a percentage or a rate. Recall measures how well an algorithm can identify all anomalies and is a percentage. Precision is more important than recall when it is okay to have less False Positives in a trade-off to having more False Negatives.

Data	BH11D_labelled
#Samples	12424
#Dimensions	5
Outlier Percentage	0.8773
ABOD	0.81198
CBLOF	0.73849
FB	0.500465
HBOS	0.3722
IForest	0.754605
KNN	0.79902
LOF	0.539105
MCD	0.949455
OCSVM	0.77253
PCA	0.76573

Table 4.1: **PyOD ROC Performance.** The ROC performance for different algorithms (average of 20 independent trials).

Data	BH11D_labelled
#Samples	12424
#Dimensions	5
Outlier Percentage	0.8773
ABOD	0.049905
CBLOF	0.00898
FB	0.00838
HBOS	0.01829
IForest	0.012475
KNN	0.026755
LOF	0.03886
MCD	0.018555
OCSVM	0.00898
PCA	0.00898

Table 4.2: **PyOD Precision@N Performance.** The Precision@N performance for different algorithms (average of 20 independent trials).

Precision@N performances (average of 20 independent trials) for different algorithms are provided in the figure 4.2. The Angle-based Outlier Detector (ABOD) method has better precision whereas Feature Bagging method has the lowest precision.

Execution Time

The figure (cite) describes the Time Complexity (average of 20 independent trials) of different algorithms. Principal Component Analysis (PCA) method has the lowest execution time, whereas One-class Support Vector Machine (OCSVM) method has the highest execution time.

Data	BH11D_labelled
#Samples	12424
#Dimensions	5
Outlier Percentage	0.8773
ABOD	1.550585
CBLOF	0.03533
FB	0.887875
HBOS	0.03534
IForest	0.521135
KNN	0.279755
LOF	0.14081
MCD	2.03366
OCSVM	2.514095
PCA	0.007475

Table 4.3: **PyOD Execution Time.** The Precision@N performance for different algorithms (average of 20 independent trials).

Original Outliers	109
Z_Score Method	24
Modified Z-Score Method	611
IQR Method	573

Table 4.4: **Outlier Count via Autoencoder.**

4.3 AutoEncoder Result

An autoencoder from H2O library was built (see appendix for code link) and the dataset was divided into 70%/30% split. The idea to decide outliers is to assume a distribution on the reconstruction errors. Then, Based on the reconstruction error data, Following threshold functions were built and executed. Threshold functions were built using fundamental statistical methods, i.e. Z-Score Method, Modified Z-Score Method and Inter-Quartile Range Method [58].

Based on the figure 4.4, It can be derived that there is a need to create a threshold function which is domain-specific, as the above threshold functions are not performing well.

Chapter 5

Security Perspective

Access to close real-time information has turned out to be of fundamental importance for human well-being and security which fills in as an early-cautioning framework for hazardous ecological conditions, for example, poor air and water quality (e.g., [59], [60]), and cataclysmic events, for example, fires (e.g., [61]),floods (e.g., [62]), and seismic tremors (e.g., [63]). The gathered information is regularly conveyed to end clients without performing information/data quality checks or assessments.

The company Ambisense designs and manufactures smart, field deployable, monitoring instruments and networks. The solution developed by Ambisense is deployable field instruments which can track a multitude of parameters in real-time and report the data back through the cloud [64]. In the following section, We will discuss various security and privacy issues involving these systems.

5.1 Security Risks

Extensively, In the ecological monitoring situation security dangers are connected to all dangers that can:

- Harm the foundation of the monitoring framework.
- Damage correspondence channels associating distinctive segments of the monitoring framework.

- Enable unapproved parties to interrupt/intrude into the monitoring framework for vindictive purposes.

Damages to the system infrastructure. Any sabotaging effort with the point of physically harming the monitoring framework can put in danger the secrecy, integrity, and accessibility of the gathered ecological information.

Violation of the communication channels. All communication channels interfacing the distinctive parts of a sensor system can become a possible target for an enemy. Specifically, a foe may not initiate an attack directly but monitor data that she would not have the capacity to get to, that is, she could deliberately alter information transmitted on such channels. These two situations design two "traditional" security attacks, which can naturally damage the privacy and respectability of the information. Other than such attacks, a foe can likewise be occupied with checking the accesses performed on the information by the approved parties, to find some sensitive data about them.

Unauthorized access. Ecological information should be made accessible just to authorized parties and users approved by the information/data proprietor. Plainly the access restrictions should be in place until they are publicised. Unapproved ingress can gain access to the database containing vital information such as coalesce data from sensors, analysis or sensitive information about sensor nodes. The data store can be a on-premise server managed by the data owner, or an external , third-party storage server. The first case can be argued to be more trustworthy provided that there is stringent access controls. In the second case, the external third-party server isnt viewed as trusted and in this case there is a need to make sure that the server should not be able to access the data on its own. An enemy interfering with the sensor nodes can be keen on updating the raw sensor data, or to infuse false information with the goal that altered information is sent to the processing node.

5.2 Wireless Telemetry Risks

This section deals with the Security Risks with respect to Wireless Telemetry used in the Ambisense Systems. Data breaches and hacking put companies in jeopardy of exposing critical business information or can cause damage to company operations. Attacks on wireless sensing and control networks cause an unintended system operation, affecting production loss or safety issues. While the most visible cases relate to computer databases, data transmitted over any network can be captured by an outside intruder [65].

Above figure represents a typical wireless remote monitoring and control system which consists of a gateway and remote nodes that interface with sensors to provide field data. A sequence of security checks is necessary to protect the data as it passes from point to point. The wireless network, itself, must be responsible for protecting this information. The wireless data is carried by a low power wireless sensor network. Then, the standard Ethernet security practices can be applied once the wireless data reaches a gateway. These practices are usually consistent with company IT policies.

A variety of data intrusions can interrupt a wireless sensor network, jeopardizing the reliability of output and even the production process. For example, in a replay attack, a device captures an encrypted message and retransmits it later to the network, with potentially serious effects.

Best Practices:

Wireless sensor network security must include more than just encryption. Device authentication and replay prevention are also important security features necessary to secure a system [65].

- **Data Encryption**: This process obfuscates data to ensure that recipients with adequate access can read it. For wireless sensor networks, AES encryption is typically chosen because it provides a high level of security. AES can be implemented on low power sensor devices.

- **Replay Prevention**: this process tries block messages that are resent by the adversary.
- **Device Authentication**: To verify that the node joining the network belongs this network, and is not malicious node. Additionally, this process allows only authenticated nodes to connect to the network.

5.3 Privacy Risks

Privacy dangers are identified with all threats that can enable an enemy to deduce sensitive data from the gathered ecological information. Such derivations can inferred by observing the information gathering process (e.g., an enemy watching production rejects can find private points of interest on the production procedures of an organization), or indirect (e.g., studies done on the presence of polluting substances in geological regions or work environments can be correlated with the relationship between relating pollutants and diseases, uncovering conceivable diseases of people living in those territories). Gathered sensitive data can include personally identifiable information which can be tracked down to people. The risks are also associated with ecological monitoring system components on which information has been gathered.

5.4 Data Anonymization

Data anonymization refers to the process of either encrypting or removing personally identifiable information from data sets, so that the people/assets whom the data describes about will remain anonymous. It is type of information sanitization whose intent is privacy protection [66].

The European Union has introduced a law i.e. General Data Protection Regulation (GDPR) which demands that stored data regarding people in the EU should undergo either an anonymization or a pseudonymization process [67].

In conclusion, The security challenges were minimized as the data was anonymized by Ambisense Ltd before giving the data to us for analysis.

Chapter 6

Conclusion

The problem of landfill will still persist, but an attempt to detect anomalies in the sensor data has been carried out. No one method can guarantee the success of the system which detects anomalies. An attempt to label the data in an unsupervised way, i.e. using Gaussian Mixture Model was made. PyOD package was utilized to perform benchmark analysis, i.e. applying the existing techniques to gauge their abilities to detect anomalies in an environmental sensor data. An Autoencoder was built, A neural network approach which is helpful in learning non-linear relationships in the dataset. The threshold function for an autoencoder needs to be improved. Meanwhile, the research gives many possibilities of solving this problem. Finally, An Ensemble approach has the ability to take the best out of all models and do a better job at detecting anomalies.

6.1 Future Scope

6.1.1 Ensemble is the Solution

Outlier ensemble approach from PyOD package and an autoencoder can go hand-in-hand to solve this problem. Building an ensemble for this problem would be an interesting work.

6.1.2 Training the machine learning models keeping in mind the temporal dependency in the dataset.

6.1.3 Model Engineering

One way to train models would be to use 3-day window time, i.e. build models which have a context of 3-days only.

Bibliography

[1] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin, "Habitat monitoring with sensor networks," *Communications of the ACM*, vol. 47, no. 6, pp. 34–40, 2004.

[2] J. Porter, P. Arzberger, H.-W. Braun, P. Bryant, S. Gage, T. Hansen, P. Hanson, C.-C. Lin, F.-P. Lin, T. Kratz *et al.*, "Wireless sensor networks for ecology," *AIBS Bulletin*, vol. 55, no. 7, pp. 561–572, 2005.

[3] S. L. Collins, L. M. Bettencourt, A. Hagberg, R. F. Brown, D. I. Moore, G. Bonito, K. A. Delin, S. P. Jackson, D. W. Johnson, S. C. Burleigh *et al.*, "New opportunities in ecological sensing using wireless sensor networks," *Frontiers in Ecology and the Environment*, vol. 4, no. 8, pp. 402–407, 2006.

[4] "Time series analysis and applications — intechopen," https://www.intechopen.com/books/time-series-analysis-and-applications, (Accessed on 08/31/2018).

[5] H. M. Hashemian and W. C. Bean, "State-of-the-art predictive maintenance techniques," *IEEE Transactions on Instrumentation and measurement*, vol. 60, no. 10, pp. 3480–3492, 2011.

[6] "Reactive maintenance: Advantages & disadvantages — fiix," https://www.fiixsoftware.com/maintenance-strategies/reactive-maintenance/, (Accessed on 08/31/2018).

[7] T. Kuzin and T. Borovicka, "Early failure detection for predictive maintenance of sensor parts." in *ITAT*, 2016, pp. 123–130.

[8] J. Emberton and A. Parker, "The problems associated with building on landfill sites," *Waste Management & Research*, vol. 5, no. 1, pp. 473–482, 1987.

[9] D. Ganesan, A. E. Cerpa, W. Ye, Y. Yu, Y. Zhao, and D. Estrin, "Networking issues in wireless sensor networks," 2003.

[10] D. J. Hill and B. S. Minsker, "Automated fault detection for in-situ environmental sensors," in *Proceedings of the 7th International Conference on Hydroinformatics*, 2006.

[11] J. D. Olden, J. J. Lawler, and N. L. Poff, "Machine learning methods without tears: a primer for ecologists," *The Quarterly review of biology*, vol. 83, no. 2, pp. 171–193, 2008.

[12] E. W. Dereszynski and T. G. Dietterich, "Spatiotemporal models for data-anomaly detection in dynamic environmental monitoring campaigns," *ACM Transactions on Sensor Networks (TOSN)*, vol. 8, no. 1, p. 3, 2011.

[13] "Weka 3 - data mining with open source machine learning software in java," https://www.cs.waikato.ac.nz/ml/weka/, (Accessed on 08/31/2018).

[14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[15] "Matlab - mathworks - matlab & simulink," https://uk.mathworks.com/products/matlab.html, (Accessed on 08/31/2018).

[16] D. P. Solomatine and A. Ostfeld, "Data-driven modelling: some past experiences and new approaches," *Journal of hydroinformatics*, vol. 10, no. 1, pp. 3–22, 2008.

[17] J. L. Campbell, L. E. Rustad, J. H. Porter, J. R. Taylor, E. W. Dereszynski, J. B. Shanley, C. Gries, D. L. Henshaw, M. E. Martin, W. M. Sheldon *et al.*, "Quantity is nothing without quality: Automated qa/qc for streaming environmental sensor data," *BioScience*, vol. 63, no. 7, pp. 574–585, 2013.

[18] "Iot connectivity - comparing nb-iot, lte-m, lora, sigfox, and other lpwan technologies — iot for all," https://www.iotforall.com/iot-connectivity-comparison-lora-sigfox-rpma-lpwan-technologies/, (Accessed on 08/31/2018).

[19] "Intel galileo vs. raspberry pi — mouser," https://www.mouser.in/applications/open-source-hardware-galileo-pi/, (Accessed on 08/31/2018).

[20] "Cisco visual networking index: Forecast and methodology, 20162021 - cisco," https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html, (Accessed on 08/31/2018).

[21] A. Zimek and E. Schubert, *Outlier Detection*. New York, NY: Springer New York, 2017, pp. 1–5. [Online]. Available: https://doi.org/10.1007/978-1-4899-7993-3_80719-1

[22] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, pp. 85–126, 2004.

[23] "An experiment with the edited nearest-neighbor rule," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-6, no. 6, pp. 448–452, June 1976.

[24] M. R. Smith and T. Martinez, "Improving classification accuracy by identifying and removing instances that should be misclassified," in *The 2011 International Joint Conference on Neural Networks*, July 2011, pp. 2690–2697.

[25] "Anomaly detection with the normal distribution - anomaly," https://anomaly.io/anomaly-detection-normal-distribution/, (Accessed on 09/02/2018).

[26] "14.1 - autoregressive models — stat 501," https://onlinecourses.science.psu.edu/stat501/node/358/, (Accessed on 09/03/2018).

[27] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "Stl: A seasonal-trend decomposition," *Journal of Official Statistics*, vol. 6, no. 1, pp. 3–73, 1990.

[28] "Forecasting: Principlesandpractice," https://otexts.org/fpp2/stl.html, (Accessed on 09/03/2018).

[29] "Sax timeseries," https://www.slideshare.net/goyalnikita277/saxtimeseries, (Accessed on 09/04/2018).

[30] "Tsrepr: Time series representations in r," https://cran.r-project.org/web/packages/TSrepr/vignettes/TSrepr_representations_of_time_series.html, (Accessed on 09/04/2018).

[31] "Welcome to the sax," http://www.cs.ucr.edu/~eamonn/SAX.htm, (Accessed on 09/04/2018).

[32] "Python outlier detection (pyod), pypi," https://pypi.org/project/pyod/, (Accessed on 09/04/2018).

[33] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[34] H. Hotelling, "Analysis of a complex of statistical variables into principal components." *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933.

[35] "Brief visual explanation of pca - analytix," http://www.ahmetcecen.tech/blog/class/2016/10/04/PCA/, (Accessed on 09/05/2018).

[36] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang, "A novel anomaly detection scheme based on principal component classifier," MIAMI UNIV CORAL GABLES FL DEPT OF ELECTRICAL AND COMPUTER ENGINEERING, Tech. Rep., 2003.

[37] M. Hubert and S. Verboven, "A robust pcr method for high-dimensional regressors," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 17, no. 8-9, pp. 438–452, 2003.

[38] P. J. Rousseeuw and K. V. Driessen, "A fast algorithm for the minimum covariance determinant estimator," *Technometrics*, vol. 41, no. 3, pp. 212–223, 1999.

[39] J. Hardin and D. M. Rocke, "Outlier detection in the multiple cluster setting using the minimum covariance determinant estimator," *Computational Statistics & Data Analysis*, vol. 44, no. 4, pp. 625–638, 2004.

[40] J. Ma and S. Perkins, "Time-series novelty detection using one-class support vector machines," in *Neural Networks, 2003. Proceedings of the International Joint Conference on*, vol. 3. IEEE, 2003, pp. 1741–1745.

[41] "Local outlier factor — turi machine learning platform user guide," https://turi.com/learn/userguide/anomaly_detection/local_outlier_factor.html, (Accessed on 09/05/2018).

[42] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recognition Letters*, vol. 24, no. 9-10, pp. 1641–1650, 2003.

[43] M. Goldstein and A. Dengel, "Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm," *KI-2012: Poster and Demo Track*, pp. 59–63, 2012.

[44] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *ACM Sigmod Record*, vol. 29, no. 2. ACM, 2000, pp. 427–438.

[45] F. Angiulli and C. Pizzuti, "Fast outlier detection in high dimensional spaces," in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 2002, pp. 15–27.

[46] H.-P. Kriegel, A. Zimek *et al.*, "Angle-based outlier detection in high-dimensional data," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 444–452.

[47] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 413–422.

[48] A. Lazarevic and V. Kumar, "Feature bagging for outlier detection," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 2005, pp. 157–166.

[49] H. V. Nguyen and V. Gopalkrishnan, "Feature extraction for outlier detection in high-dimensional spaces," in *Feature Selection in Data Mining*, 2010, pp. 66–75.

[50] "Anomaly detection in time series using auto encoders," http://philipperemy.github.io/anomaly-detection/, (Accessed on 09/07/2018).

[51] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PloS one*, vol. 11, no. 4, p. e0152173, 2016.

[52] "Plotly," https://en.wikipedia.org/wiki/Plotly, (Accessed on 09/07/2018).

[53] "Keras," https://en.wikipedia.org/wiki/Keras, (Accessed on 09/07/2018).

[54] M. S. alDosari, "Unsupervised anomaly detection in sequences using long short term memory recurrent neural networks," Ph.D. dissertation, 2016.

[55] "Gaussian mixture model — brilliant math & science wiki," https://brilliant.org/wiki/gaussian-mixture-model/, (Accessed on 09/11/2018).

[56] "Bayesian information criterion - wikipedia," https://en.wikipedia.org/wiki/Bayesian_information_criterion, (Accessed on 09/11/2018).

[57] "Evaluating anomaly detection algorithms with precision-recall curves," https://medium.com/wwblog/evaluating-anomaly-detection-algorithms-with-precision-recall-curves-f3eb5b679476, (Accessed on 09/12/2018).

[58] "Three ways to detectoutliers - colin gorrie's data story," http://colingorrie.github.io/outlier-detection.html#iqr-method, (Accessed on 09/12/2018).

[59] H. B. Glasgow, J. M. Burkholder, R. E. Reed, A. J. Lewitus, and J. E. Kleinman, "Real-time remote monitoring of water quality: a review of current applications, and advancements in sensor, telemetry, and computing technologies," *Journal of Experimental Marine Biology and Ecology*, vol. 300, no. 1-2, pp. 409–448, 2004.

[60] B. Normander, T. Haigh, J. S. Christiansen, and T. S. Jensen, "Development and implementation of a near-real-time web reporting system on ground-level ozone in europe," *Integrated environmental assessment and management*, vol. 4, no. 4, pp. 505–512, 2008.

[61] M. Hefeeda and M. Bagheri, "Forest fire modeling and early detection using wireless sensor networks." *Ad Hoc & Sensor Wireless Networks*, vol. 7, no. 3-4, pp. 169–224, 2009.

[62] P. Young, "Advances in real–time flood forecasting, philos," TR Soc. Lond., 360, 1433–1450, Tech. Rep., 2002.

[63] J. K. Hart and K. Martinez, "Environmental sensor networks: A revolution in the earth system science?" *Earth-Science Reviews*, vol. 78, no. 3-4, pp. 177–191, 2006.

[64] "Environmental monitoring is often a complex undertaking," http://ambisense.net/services/, (Accessed on 08/31/2018).

[65] "Data security practices can ensure a more reliable and secure wireless remote monitoring and control system — remote magazine," http://www.remotemagazine.com/main/articles/data-security-practices-can-ensure-a-more-reliable-and-secure-wireless-remote-monitoring-and-control-system/, (Accessed on 08/31/2018).

[66] "Data anonymization - wikipedia," https://en.wikipedia.org/wiki/Data_anonymization, (Accessed on 08/31/2018).

[67] "Data science under gdpr with pseudonymization in the data pipeline," https://www.dativa.com/data-science-gdpr-pseudonymization-data-pipeline/, (Accessed on 08/31/2018).

includeappendix1

Appendix A

Software

A.1 Python

Python is a programming language that is widely used among programmers and in the industry.

A.1.1 Other Libraries and Frameworks

A number of python libraries were used like PyOD, tensorflow, numpy, scikit-learn. Tableau tool was extensively used for visualization. The other third party libraries were also used.

A.2 Experimental Setup and Code

The experiment was performed in linux environment (Ubuntu 16.04). The code and datasets can be found at the following link, https://github.com/AshwathSalimath/AnomalyDetectionThesis.