Machine Learning Based Failure Detection in Data Centers

Machine Learning Based Failure Detection in Data Centers

BY

NEGIN PIRAN NANEKARAN, M.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

© Copyright by Negin Piran, August 12, 2020

All Rights Reserved

Master of Applied Science (2020) McMaster University

Electrical and Computer Engineering Hamilton, Ontario, Canada

TITLE: Machine Learning Based Failure Detection in Data Centers

AUTHOR: Negin Piran Nanekaran,

B.Sc., (Electrical Engineering)

M.Sc. (Electrical Engineering)

McMaster University, Hamilton, Canada

SUPERVISOR: Dr. Mehdi Narimani

NUMBER OF PAGES: xi, 150

Abstract

This work proposes a new approach to fast detection of abnormal behaviour of cooling, IT, and power distribution systems in micro data centers based on machine learning techniques. Conventional protection of micro data centers focuses on monitoring individual parameters such as temperature at different locations and when these parameters reach certain high values, then an alarm will be triggered. This research employs machine learning techniques to extract normal and abnormal behaviour of the cooling and IT systems. Developed data acquisition system together with unsupervised learning methods quickly learns the physical dynamics of normal operation and can detect deviations from such behaviours. This provides an efficient way for not only producing health index for the micro data center, but also a rich label logging system that will be used for the supervised learning methods. The effectiveness of the proposed detection technique is evaluated on an micro data center placed at Computing Infrastructure Research Center (CIRC) in McMaster Innovation Park (MIP), McMaster University.

Acknowledgements

I would like to thank my husband Mohammad for constant support and encouraging me to do what I love. I believe without him; I could not continue my studies. Also, I would like to thank my supervisor Dr. Mehdi Narimani for his continual support, advice and feedback in each step of my study journey at McMaster University. I would also like to thank Cheryl Gies, Graduate Administrative Assistant, for her helps that were clearly going extra miles. And lastly, it has been very rewarding to work with such an excellent and dedicated research team at McMaster Innovation Park (MIP).

Table of Contents

List of Figures

List of Tables

Chapter 1

1 Introduction

Data Centers are the essential parts of today's technology, since IT-based companies such as banks, hospitals and many governmental institutes depend on the reliable operation of data centers. Especially by growing online businesses, data centers became to be the pillar of today's IT society. People are using internet for different purposes such as socializing, shopping, sharing information, gaming, attending in online courses, etc. On the other hand, mobile technology made an evolution and has changed the way people use the internet which led to the rapid growth in the number of internet users. Therefore, web search became an everyday necessity, which leads to an obvious increase in the internet usage and the demand for real-time data transmission. Persistent increase in the number of customers lead to a significant increase in the number of data centers. Figure 1-1 shows an evident growth in the number of data centers from 570 in 2020 to 628 in 2021 (Cisco, 2017).

Figure 1-1: Global data center growth (Cisco, 2017)

Data centers are categorized into three groups, micro data centers, mid-size data centers and large size data centers. Well known large IT-based corporations like Amazon, Facebook, Microsoft and Google are using large size data centers which host more than thousands of computers to be able to handle the large number of requests. These corporations are growing their data centers in terms of both size and number. For example, Amazon has 22 data centers all over the world and Figure 1-2 shows the location of these data centers. The number of computers in mid-size data centers and micro data centers are less than thousands and less than hundreds respectively. With the growing number of internet users, computers and other networking equipment which handles these requests have created a huge need for the maintenance equipment, inventories, and technicians. Traditional reactive or scheduled maintenance is not economical anymore and data center owners are looking for better ways of maintenance to shave off immense downtime costs.

The smarter approach is achievable by predictive modeling of normal and abnormal operations in data centers which will be discussed in detail in chapter 2.

Figure 1-2: Amazon web services (AWS) data centers worldwide locations (Singh, 2019)

1.1 Data Center Operation

Data center's infrastructure categorized into three subsections, power, cooling, and IT. In this configuration, servers are processing the incoming service request from the web and the resulting heat is removed by cooling unit and both units are fed though power unit. Following subsections, describe each unit in more details.

1.1.1 Information Technology (IT)

In a data center, an IT room is allocated to store servers on premises and can comprise thousands of servers lined up together. Figure 1-3 shows a typical server room located in

Google Iowa data center. Routers and switches are used to transport traffic between these servers in the data center and the core network. The quality of these components is very important in terms of both increasing the speed of transportations and decreasing failure rate in the IT racks. The large number of components in the IT room means that failure ratio is high because any given component may have small but non-zero failure rate. Therefore, racks should be analyzed carefully in terms of infrastructure and security. In large size data center, hundreds of thousands of switches are used especially in the IT room and their failure, imposes a big lose for IT-based businesses each year. For instance, Microsoft has lost more than 400 switches each year. That is why data center owners prefer to use high quality switches and cables to minimize these kinds of failures. On the other hand, Fortunately, because of increasing server capacity and network bandwidth, server's failures are very rare compared to other components in the IT room. The common failure that could happen in IT racks is the partial failure of servers where the state of a networking component is nominally healthy, but a subset of traffic is either lost or delayed. This type of failure which should be analyzed in the server level is not in the scope of this research.

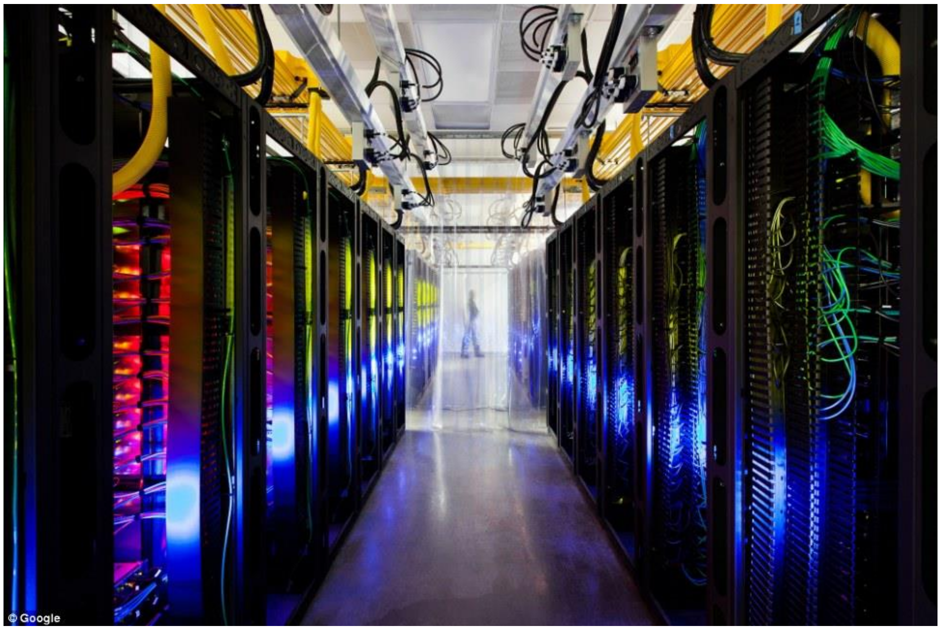

Figure 1-3: Google Iowa campus network room (News, 2015)

1.1.2 Power Distribution

Power distribution system feeds both cooling and IT units in a data center through electrical circuits, electronic converters, and an energy storage system. In the following subsections, common approaches to distribute power in a data center is discussed.

A. Traditional PDU distribution

In traditional data centers, the low power density of server racks required a low volume of under-floor air, and typically one or more branch circuits connects the server racks to the main PDUs placed along the side of the server room. Traditional distribution is typically used for designs from 50 kVA to 500 kVA (Torell, 2014). Figure 1-4 shows typical PDU within a data center.

Figure 1-4: Traditional PDU (Torell, 2014)

B. Panelboard distribution

In this method, the main power is distributed to separate panelboards, which usually ratings from 1.5 kVA to 75 kVA. Low cost and flexibility in sizing the breakers and cables are advantages of panel board distribution method. On the other hand, such flexibilities are prone to human error in the installations. Figure 1-5 shows typical Panelboard PDU within a data center.

Figure 1-5: Panelboard PDU (Torell, 2014)

C. Modular distribution

Modular power distributions are flexible, more reliable, and more efficient ways of delivering power to server racks. Modular distribution has higher initial investment costs but because of more reliability and efficiency, it decreases the total cost of ownership (TCO). Figure 1-6 shows typical Modular PDU within a data center.

Figure 1-6: Modular PDU (Torell, 2014)

Modular panel can be connected to the distribution systems in three different ways, under floor, overhead, and floor mounted that are shown in Figure 1-7.

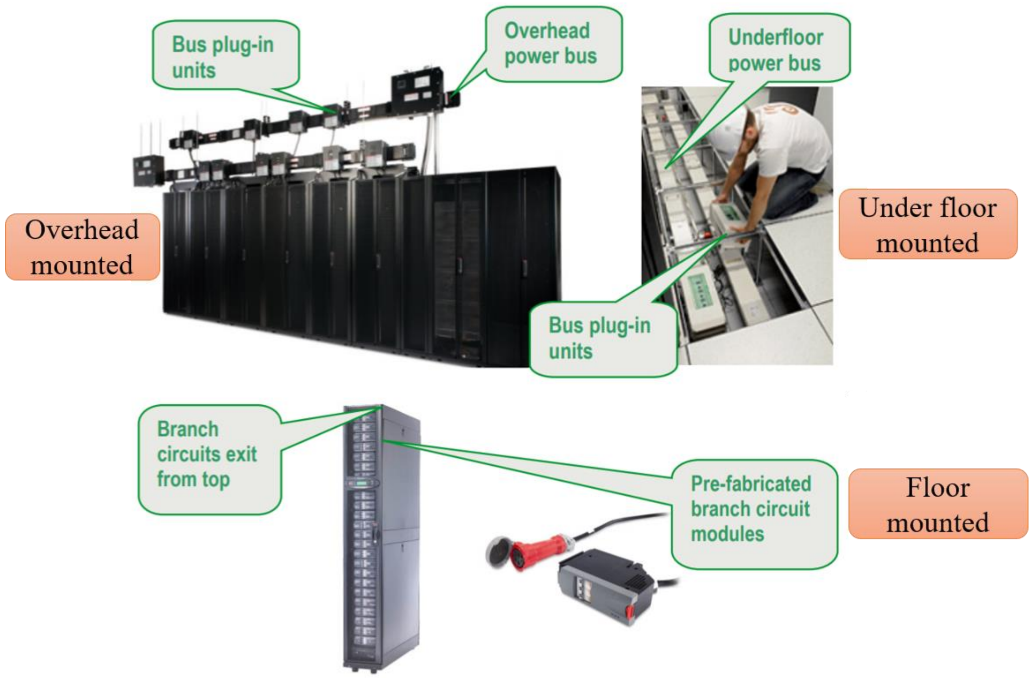

Figure 1-7: Examples of Under floor, Overhead and Floor mounted within a data center (Torell, 2014)

1.1.3 Cooling Unit

Servers are generating heat which should be removed from the server racks and IT rooms. Cooling system is the key component that consumes a large portion of the incoming power in a data center to avoid overheating the components (Ahuja, 2013). Therefore, even small improvements in efficiency of cooling unit, could lead to considerable savings in energy and power consumption. Air-cooled and liquid cooled systems are common architectures for cooling systems in data centers. In the following subsections cooling architectures and their applications has been discussed.

A. Liquid cooled system

These units typically use water or glycol as coolant liquids in the system. Figure 1-8 illustrates the schematic view of a chilled-water system (Evans, 2012). In these systems cooled water is piped throughout the building and the air handlers to cool down the IT equipment. The main drawback of this systems is the maintaining of water in an acceptable level since it could be vaporized during daily operations of cooling unit.

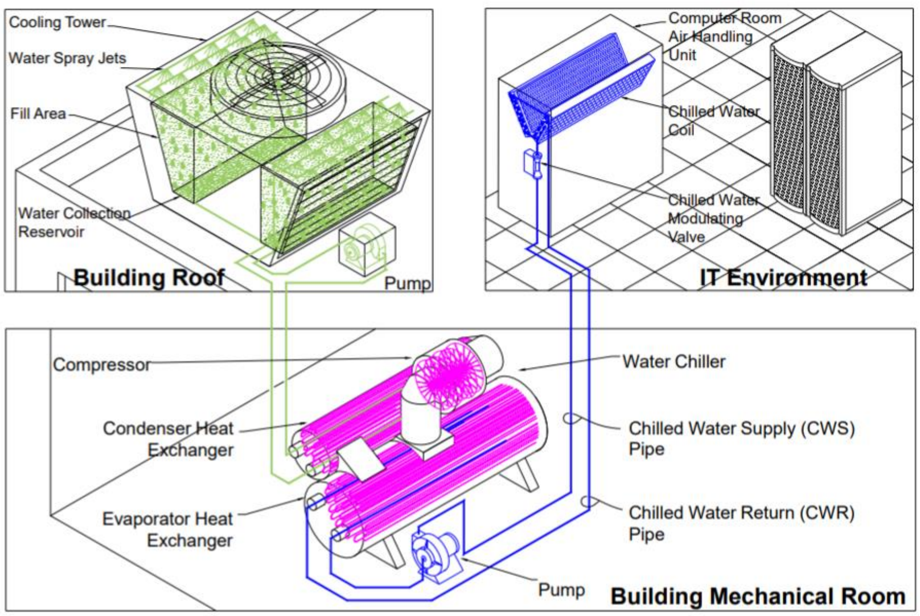

Figure 1-8: Water-cooled chilled water system (Evans, 2012)

B. Air cooled system

Air cooled systems are widely used in cooling IT infrastructure of a data center where low cost implementation and easy maintenance are priorities (Naguib, 2011). Higher power consumption time consuming maintenance downtimes and difficulties in scaling are the main disadvantages of air-cooled systems. Figure 1-9 shows the layout of an air-cooled system with four major components, condenser, evaporator, expansion valve and compressor.

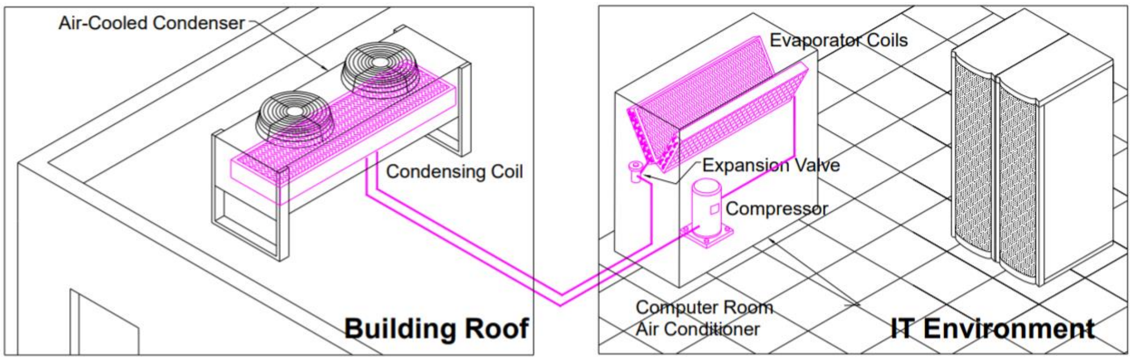

Figure 1-9: An air-cooled system (Evans T. , 2012)

1.2 Data Center Maintenance

Data center downtime costs could lead to huge financial losses. These losses can be reduced by proactively maintaining the equipment. Generally, maintenance management is divided into planned or unplanned programs (Mobley, 2002). Different types of planned and unplanned maintenance are shown in Figure 1-10 and have been described in the following subsections.

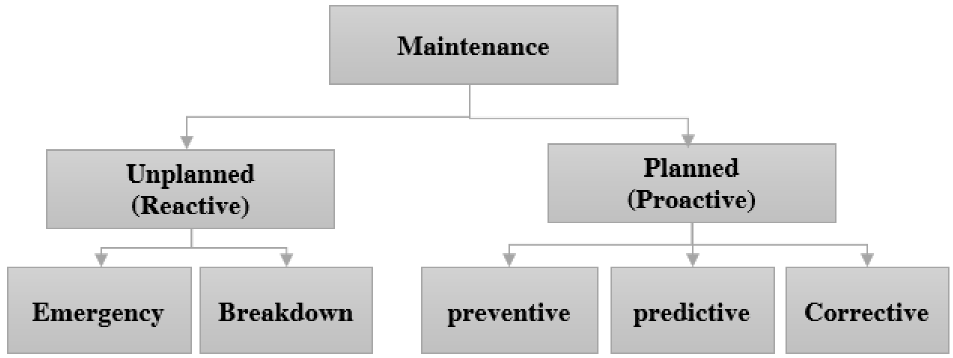

Figure 1-10: Maintenance methodologies (Fiix, 2019)

1.2.1 Unplanned Maintenance

An unplanned maintenance, limits expenditure on proactive maintenance and seems to have lower costs but in fact could lead to very high financial losses because of the process interruptions. Finding the root cause of a breakdown could be hard since there is no maintenance strategy and most probably can put technicians in the high risk. Corrective maintenance and Reactive maintenance are the two types of unplanned maintenance.

- Corrective maintenance consists of repairing an equipment that is still running within acceptable ranges but near to the unsafe operation region. Replacing a partially exposed wire is an example of this type of maintenance.
- Reactive maintenance is performed when equipment is completely failed. This type of maintenance should be performed immediately due to high cost of interruption in the process or safety hazards that it could create (such as fixing a broken engine).

1.2.2 Planned Maintenance

Planned maintenance program is divided into three types, planned breakdown, preventive and predictive maintenance.

- In planned breakdown which is also called run-to-fail failure, equipment would be in service until it stops due to problem in any of its components. This method is different from reactive maintenance since there is a plan for the failures and replacing the failed equipment.
- Preventive maintenance tries to lower down the failures and interruptions by regular maintaining of the equipment. This maintenance can be scheduled in fixed time intervals or based on the amount of usage in a specific asset. The main drawback of this method is the over/under maintaining of the equipment because of the inaccurate estimate for the time that maintenance should be performed on the equipment.
- Predictive maintenance is a type of planned maintenance where predictive models try to estimate the health of different parts of a system. The goal is to perform the maintenance just in time and avoid over spending on maintenance and system interruptions at the same time.

1.3 Micro Data Centers

Emerging new technologies such as self-driving cars or pervasive healthcare systems require online processing of huge amount of data instantly. Processing massive data with very low processing time requires edge processing where data processor units are close to IT loads. Micro data centers are new solutions to this problem and these days many organizations that need to share and analyze a quickly growing amount of data are turning to localized micro data centers installed on the place that is close to their central office.

To improve the reliability of any system, first thing to do is truly understanding the model, operation and design of that system. In this section the components of data center and their functions are analyzed. Information technology (IT), Cooling system and power supply are the three main parts of micro data centers. As it mentioned, micro data centers are smaller versions of a traditional data centers and are a good fit for office spaces which is located as close as possible to the end users. In normal operation, input power to servers (IT racks) will be transformed to heat and should be removed by cooling system. Figure 1-11 shows the holistic model of a micro data center. Cooling system includes chiller unit and air handler unit (AHU). And the shown pump in Figure 1-11 circulates water between these two units. In this figure a model of air cooled chiller is considered since in the setup of this research a packaged air cooled is used. Following are the four main air cooled chiller components and their functions:

- The **evaporator** allows the low pressure liquid refrigerant to absorb heat from the indoor air of the IT racks.
- The **compressor** takes the low pressure vapor from the evaporator and compresses it into high pressure vapor and also circulates the refrigerant in the system.
- The **condenser** condenses the refrigerant from air to water.

- The **expansion valve** performs two functions. It drops the pressure of the refrigerant liquid and separates the high and low pressure sides of the system while controlling the flow of liquid refrigerant between the condenser and the evaporator.

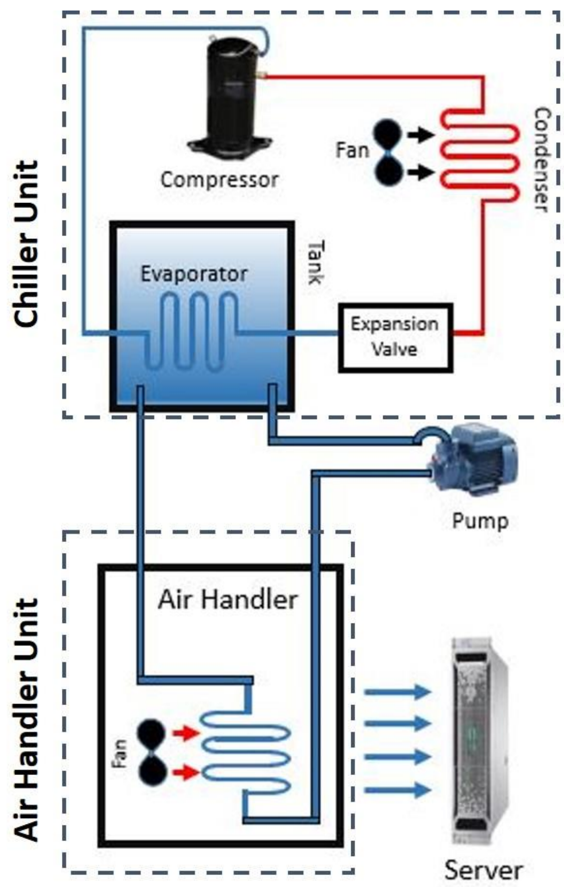

Figure 1-11: The holistic model of a micro data center

The above process is repeating for removing heat generated by servers in a rack till the indoor temperature in the rack drops below the set temperature value (usually around 10°C). If the rack temperature goes below than the minimum temperature set point, cooling units stop from working till the temperature goes up to a higher temperature set point (usually around 15°C). Now, if for any reason such as fan failure in the AHU, the AHU fails or the circulation of coolant slows down due to the pump abnormal behavior, then the cooling system will not be able to remove the heat from the server racks effectively. This will disrupt the normal operation of the IT system which generates heat, and cooling system cannot remove the heat properly. Data center failures not only costs of recovery time but also lost revenue, lost productivity and lost customers are other outcomes and average cost of unplanned data center failure is profound (Ponemon L. , 2016). While micro data centers have provided solutions for big data analysis of newly introduced industries, maintaining them is the main challenge for the owners of such facilities. This challenge originates from the fact that not only it is not economical to have expert technicians on-board for maintaining physical assets in micro data centers, but also number of qualified experts would be very limited with the proliferation of micro data center usage. These days, by growing the technology and increasing internet users, the reliability of data centers and micro data centers has taken more important than in the past.

1.4 Micro Data Center Predictive Maintenance and Fault detection

Total cost of ownership in data centers could be grouped into the following two categories,

- Capital expenditure (CapEx): Includes costs related to commercial space and equipment installed in the data center.
- Operational expenditure (OpEx): Costs related to energy usage (electricity and water), equipment maintenance, and staff salaries are the top operational costs in data centers.

While some of these costs are fixed and cannot be optimized, there is good opportunity in preventing financial losses using smart maintenance through predictive modeling.

Data center failures not only costs recovery time, but also leads to lose revenue, productivity, and customers (Patterson D. A., 2002). According to the study by Ponemon Institute, the average cost of a data center outage was 8,851 dollars per minute in 2016 which shows data centers outages could lead to huge financial losses. Emerson (Ponemon, Ponemon Institute, 2013) proposed the causes of data center outages. In their study, battery failure, cybercrime, and human error are recognized as the top three reasons for data center failures. With the growing demand of micro data centers, as they are the integral parts of today's technology, it is necessary to have a real-time anomaly detection and detect abnormalities before causing serious failures in the system. Thus, learning the correlations between IT, cooling, and power units will create unique opportunities for the predictive models to learn normal and healthy operations. These models can flag any deviations from the normal situation in a very nascent time which prevents cascading failures in the system. The details will be given in chapter 3.

1.5 Research Objectives

In this thesis, a new approach to fast detection of abnormal behavior of cooling and IT systems in micro data centers based on machine learning techniques is proposed. In particular, the thesis attempts to tackle the following objectives:

- Consolidate and analyses the fault mechanisms in micro data centers

- A data acquisition system (DAQ) is developed to collect electro-mechanical signals from micro data center (discussed in chapter 4).
- The real-time anomaly detection method detects abnormalities before causing serious failures in the system and has the following main features:
  - **Using Current Sensors:** Current data centers are using only temperature sensors to detect, predict and localize the anomalies. In this research, beside the temperature sensors we used current sensors which have low correlation with the existed temperature sensors. The low correlation between current and temperature sensors helps machine learning approaches to have fast and accurate detection and localization of the failures.
  - **Holistic View in Detection:** Decision making based on independent variables (such as temperatures or power consumptions) would lead to suboptimal results. Our method looks at a holistic view of the entire data center and learns normal operations inside such complex environment and flags abnormal situations even if individual meters are showing normal readings.
  - **Non-invasive Data Collection:** Data is collected from temperature and current sensors in data center in a non-invasive way that eliminates costly downtimes for implementing smart monitoring solution developed in this research.
  - **Costs Effective:** Parts used in the design makes the DAQ system very affordable for any micro data center owner. This cost is under 100 CAD in 2019 to log and process 8 different channels. The low-cost hardware makes it easier

to collect more signals and add new features to machine learning algorithms that eventually will increase their accuracy.

1.6 Thesis Organization

The following is an overview of the contents of each chapter in this thesis. Chapter 2 reviews the related work on improving the data center efficiency and reliability. In this chapter, predictive maintenance method is introduced for enhancing the reliability of the data centers. Machine learning based predictive maintenance methods are innovative ways of predicting or localizing failures and are discussed in chapter 3. Chapter 4 presents the mathematical presentations of proposed techniques and the experimental results and their effectiveness in predicting and localizing failures. And finally, in Chapter 5, we conclude our findings in this thesis.

Chapter 2

2 Review of Efficiency and Reliability in Data Centers

Data centers are centralized or distributed facilities responsible for processing daily web requests and thus play vital role in continuity of service for many industries. They retain different subsystems (cooling units, IT room, PDU) and each subsystem has various components. Figure 2-1 shows the major components of a typical data center.

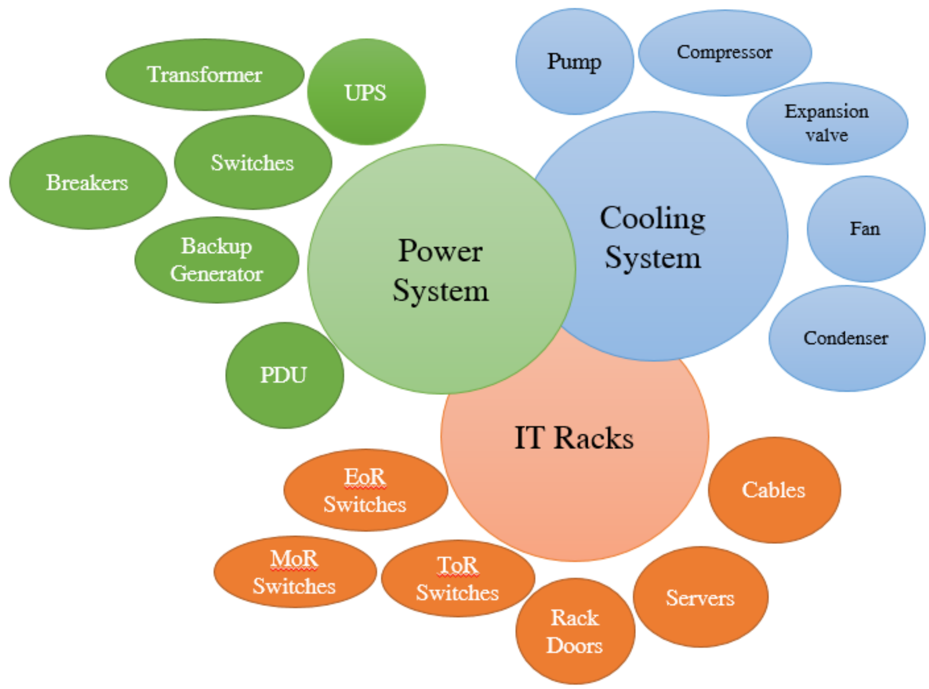

Figure 2-1: Major components of a typical data center (Goiri, 2015), (Wang T. S., 2014)

Proliferation of such complex systems has created extensive research grounds to increase safety through improving reliability and decrease financial losses through improving efficiencies. The following subsections have been dedicated to review related researches to efficiency (Section 2.1) and reliability (Section 2.1.22.2) in data centers.

2.1 Data Center Efficiency

In a data center electrical, mechanical and control system all are working together. Therefore, improving data center efficiency is a multidimensional challenge that is requiring a concerted effort to optimize power distribution, cooling infrastructure, IT equipment and IT output. Hence, energy efficiency can be analyzed in different parts of a data center separately.

2.1.1 Efficiency Related Metrics in Data Centers

Following metrics are commonly used in data center operation and management,

A. Power usage effectiveness (PUE):

Incoming power to the data center can be lost in different parts of it. Figure 2-2 shows the major components with high power losses ratio in a typical data center.

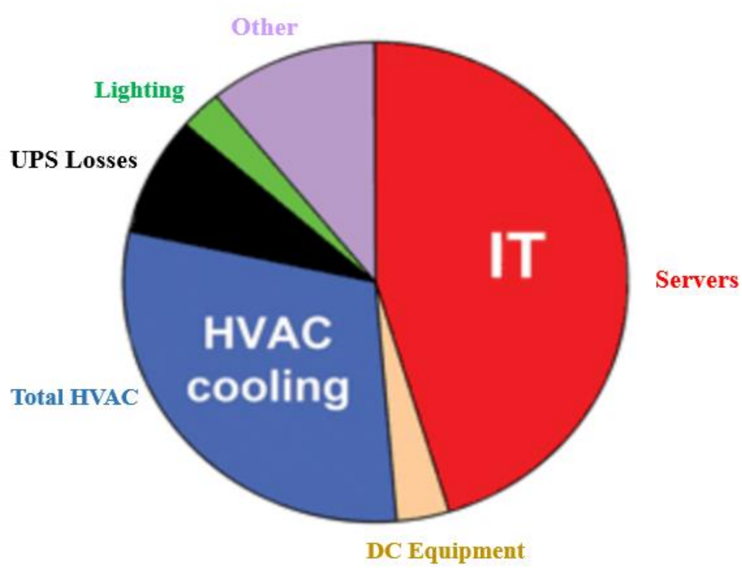

Figure 2-2: power losses in a typical data center (Tiwari, 2011)

PUE is the total power consumed by the data center divided by the power used by the servers (Avelar, 2010). This quantity shows how much power delivered to a data center is lost before it consumed by servers.

$$\text{PUE} = \frac{\text{Power delivered to data center}}{\text{Power consumed by servers}} = \frac{P_{in}}{P_{ICT}} \quad (2\text{-}1)$$

The lower the value of the PUE, the better the energy efficiency of the data center. A PUE value of 1 would represent the optimal data center efficiency which means that all the energy consumption in the data center is only consumed by the computing servers. Typically, operators are measuring PUE several times in a week to calculate the average weekly PUE of their data centers. The average PUE is 1.8 according to a study conducted by the Uptime Institute, (Miller, 2011). However, the average PUE in Google data centers is 1.11 (Google, n.d.).

B. Energy reuse effectiveness (ERE)

This metric called also as a green tool for data centers. ERE of a data center can be calculated according to the equation (Patterson M. K., 2012).

$$\text{ERF} = \frac{P_{reuse}}{P_{in}} \quad (2\text{-}2)$$

Where P_{reuse} is the amount of waste energy from the data center that can be used elsewhere.

C. Total cost of ownership (TCO)

Low TCO (Barroso, 2009) is very important especially for IT-based companies.

$$TCO = CapEx + OpEx \quad (2\text{-}3)$$

Where CapEx and OpEx stand for capital expenses and operational expenses respectively which are described in the previous section. Cost-effective topology design is important to data center to decrease CapEx, while using high efficiency equipment as well as monitoring data center to detect and predict failures can decrease OpEx.

D. Server compute efficiency (SCE)

It is important to determine whether or not a server is being used as a primary server, secondary server, etc. Conventionally, only CPU usage is used to determine whether or not a server is being used for primary services. But todays the SCE metric (Blackburn, 2010) is a new parameter which can determine the type of servers in a data center. Beside CPU usage, SCE measures disk and network I/O, incoming session-based connection requests and interactive logins to determine if the server is providing primary services or not. Because looking only at CPU usage, the sever may sometimes misclassified into a failed server although in reality it is working as a secondary server. SCE can be a better metric for correctly classifying the servers (Blackburn, 2010).

E. Computational efficiency:

Computational efficiency (Uddin, 2014) of servers is a quantity that is used to calculate server's efficiency.

$$\text{Computational Efficiency} = \frac{\text{average compute rate}}{\text{average energy consumption rate}} \tag{2-4}$$

2.1.2 Infrastructure Efficiency in IT room

By increasing web search and internet demand, there is an exponential growth in the number of servers in data centers. Considering the high cost of building data centers, designers always try to use as much servers as possible that can be supported by a given infrastructure installation. Techniques such as power capping (The row-level power budget is enforced by physical circuit breakers and fuses) seek to increase infrastructure efficiency by increasing the number of server supported by a given installation (Wang G. W., 2016). Increasing number of servers in a data center with a fixed power budget increases the computation capacity, and high computational capacity together with low power usage are important factors when determining the efficiency of data center (Shehabi, 2016).

Servers are one of key building blocks in a data center and improving their efficiency could have a great impact on data center overall efficiency. According to a study by Emerson Network (Power, 2009) any reduction in power consumption of IT will also reduce losses in the server power supply and the power distribution. Both of these reductions will reduce the amount of cooling energy required. This cascaded effect makes IT room more important in terms of saving energy in the whole system. There are many rack level studies for data centers that try to find the optimal server configuration and operation. Followings are the typical methodologies and the examples:

- **Turning servers off when they are not in use:** This is a very basic but effective way to control servers in a data center (Cole, 2011). There are always non-used or lightly-used servers which consume 70-85% of the power of a server running at

100% CPU load but produce no useful output. These types of servers also called as "Ghost servers" which turning them off will decrease the total amount of power drawn by IT racks. CPU utilization is a good metric to identify ghost servers. But the problem is that considering only the amount of utilization of servers can be misleading in determining secondary servers (like servers which are responsible to backup emails) and turning them off. That is where SCE can make a huge improvement in classifying servers into the right groups. Even if the ghost servers are identified correctly, the problem with this approach is often the setup time, the time required to turn a server off and back on (Chen G. H., 2008). Time is a very important factor especially for some applications such as autonomous driving cars or sensitive medical devises that require almost online responses from servers.

- **Replacing traditional servers with efficient servers:** New advancements in semiconductor industry, provided the opportunity to build efficient processors that are able to align their energy usage with the load level that they are serving. The new generations of servers have multi-core processors that are significantly energy efficient compared to the traditional single-core servers. Having high processing power in these servers also helps data center designers to increase the capacity of IT racks and thus creates extra spaces in server rooms (Cole, 2011).
- **Servers consolidation:** Combining legacy servers into a virtual system creates an IT environment that resources can be shared. This method is usually called server consolidation and can improve efficiencies in many traditional data centers. Figure X shows the consolidation servers.

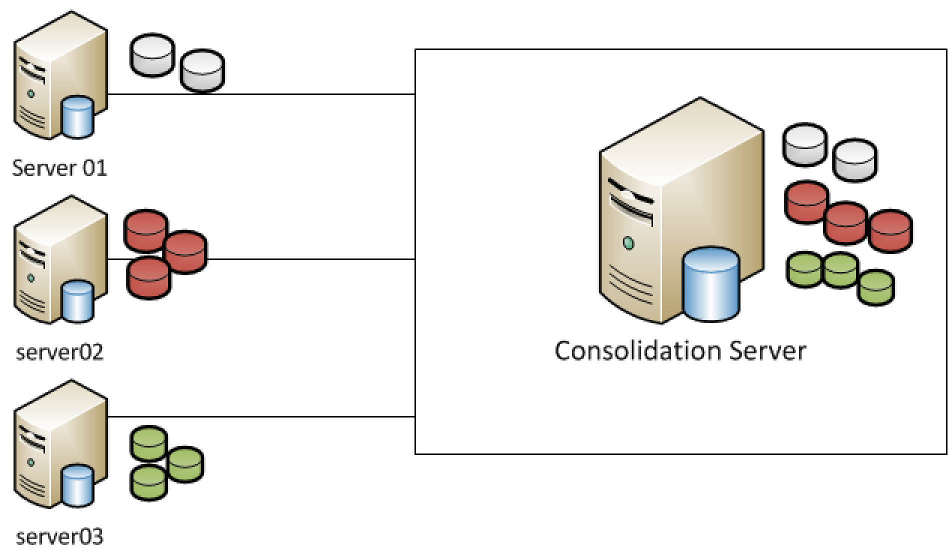

Figure 2-3: Server consolidation technique (Dickason)

In a consolidation server, additional servers will be used only when demand increases and using this methodology, server CPU usage can be increased by 40-60%. There are different studies to improve the efficiency of consolidation servers. For instance, in (Marty, 2007) they developed a CMP memory systems for server consolidation. This memory will help to maximize shared memory accesses serviced within a Virtual Machines (VMs) by supporting content-based page sharing among VMs.

- **Load balancing:** Load balancing is the technique which distributes workload uniformly among all the servers in a data center to achieve evenly distributed power in order to reduce hot spots and decrease latency (Sharma, 2005). Having hot spots in server rooms not only increases the damages to equipment, but also causes inefficient cooling scenarios.

2.1.3 Infrastructure Efficiency in Cooling Units

In a data center, cooling system is the key component that avoids overheating of components and consumes a large portion of the incoming power to the data center (Ahuja, 2013). A significant operational efficiency can be gained from how to choose to cool the data center. An insufficient cooling system can lead to overheating of the resources, reducing system reliability and device lifetime which all will make overhead costs. To avoid these costs and improve the reliability of the data center, it is essential to make the data center more efficient. Trade-off between (high) temperature setting and (low) power consumption is a challenging optimization problem. Followings are the summary of present cooling unit approaches which help to increase the overall data center efficiency.

- **Multiple cooling units:** Overcooling is a common issue in a data center where some parts of the data center being cooled more than the requirements. Using multiple cooling units in a data center is one of the basic ideas which can avoid overheating and overcooling. In data centers with multiple cooling units, each of the cooling units can be controlled independently or they can communicate in order to create a distributed cooling system which has higher efficiency than central cooling units.
- **Dynamic smart cooling** (DSC): Monitoring a network of temperature sensors at the air inlet and exhaust of equipment racks is the conventional method of controlling the cooling unit. Data from the sensors is fed to a controller where it is evaluated, and the controller can independently manipulate the supply air

temperature and airflow rate of each computer room air conditioning (CRAC) in the data center. In order to accomplish this efficiently, the impact of each CRAC in the data center must be evaluated with respect to each sensor. The result of such an evaluation will define the regions of influence for each CRAC unit. This information is used to determine which CRACs to manipulate when a given sensor location requires more or less cool air. In this case the efficiency of cooling unit can be calculated by the following equation (Wemhoff, 2013).

$$\text{Coefficiency of performance (COP)} = \frac{\text{amount of heat that is removed by the CRAC (Q)}}{\text{total amount of energy (E)}} \qquad (2\text{-}5)$$

2.1.4 Infrastructure Efficiency in Power Distribution Units

Looking at Figure 2-2, around 10 % of losses in a data center relates to losses in power units. If we can reduce the losses in the power system, more of the incoming power to the data center will feed the IT racks. This will improve the energy efficiency and keep PUE close to the optimal point. There are some techniques that are useful to improve the infrastructure efficiency in PDU for a data center;

- **Oversubscribing power:** This technique is oversubscribing a data center's power infrastructure with more servers than it can support (Sakalkar, 2020). Figure 2-4 shows the diagram of an oversubscribed power delivery system in a data center. This method can maximize resource utilization, increase the operator profit, and reduce the tenant costs. However, without an appropriate power control mechanism, oversubscription can lead to expensive unplanned service outages. In this method, power should never exceed the rated power capacity because it can throw a circuit breaker and take a section of the data center offline. Also, overloading power can damage the component permanently.

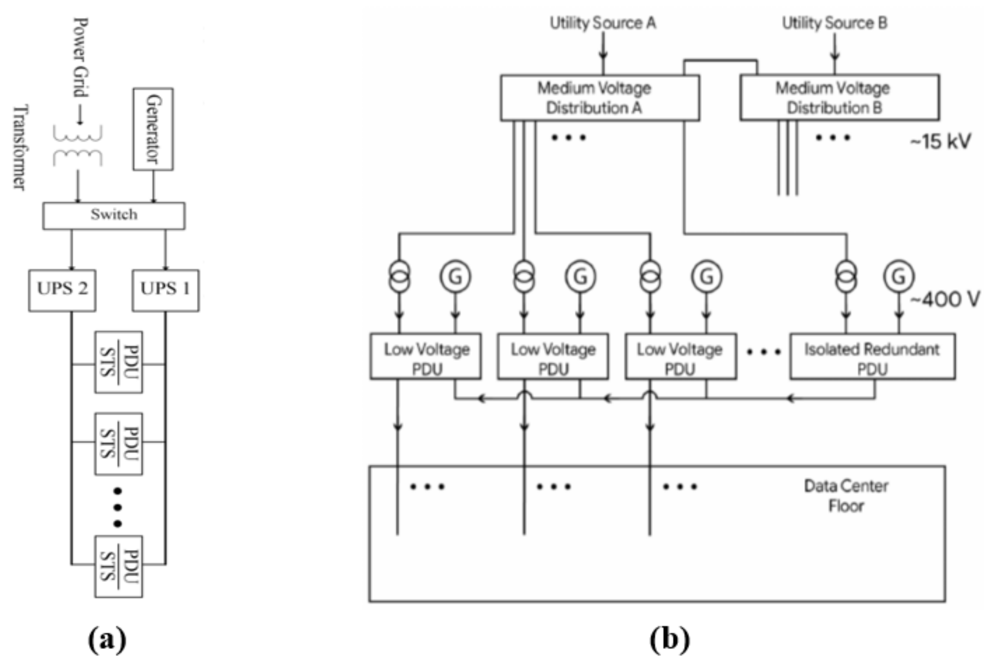

Figure 2-4: Schematic of a) a traditional power distribution system with N+1 redundancy at the PDU level, b) an Oversubscribing power delivery system (Sakalkar, 2020) (Fu, 2011)

- **power capping:** Power capping is like a safety mechanism that is used to ensure maximum power levels are not exceeded and circuit breakers are not tripped (which was the major problem of oversubscribing power in data center). In this method, throttling server power (via frequency/voltage scaling) is used as the safety mechanism to Techniques such as power routing seek to increase infrastructure efficiency by reducing the amount of power infrastructure equipment needed (Kontorinis, 2012).

- **Power routing:** Capital costs can be further reduced by using Power Routing (Liserre, 2016), which allows load to be shifted among PDUs during imbalances.

It should be also mentioned that all of these techniques require software mechanisms to track and predict peak power, to manage power budgets at each server, circuit, and PDU, while minimizing performance throttling. Though peak power could be tracked with explicit metering and logging, assessing peak power directly from operating system-level metrics can drastically reduce costs.

2.2 Data Center Reliability

Data centers are an increasingly vital infrastructure and their failures can be expensive due to business disruption and lost revenue. Failures can happen in any part of a data center. Figure 2-5 shows the failure causes in data centers where “battery failure” and “UPS capacity exceeded”, are recognized as the top two failures (Ponemon Institute, 2013).

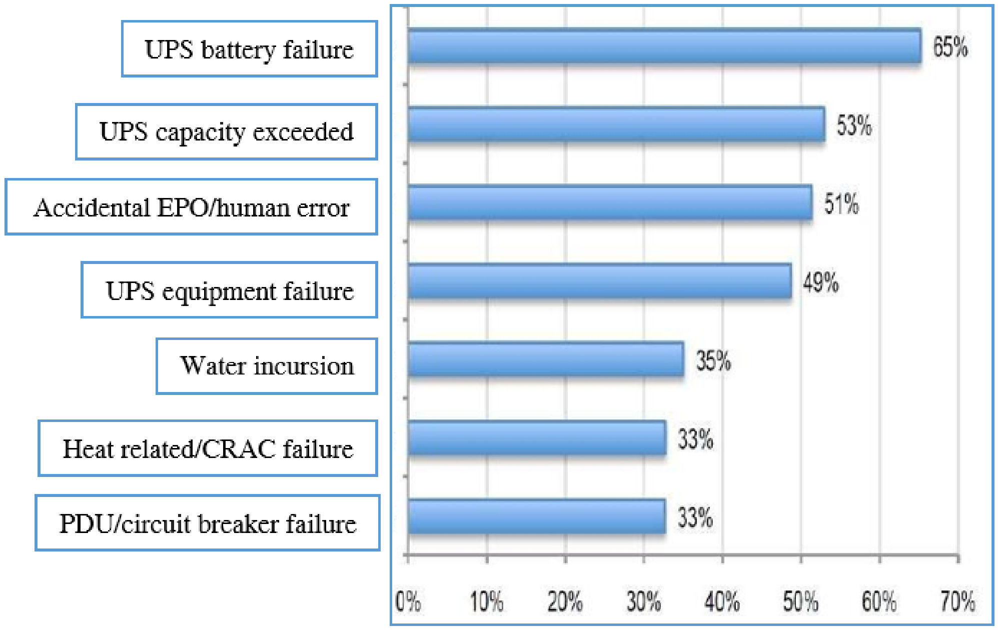

Figure 2-5: Causes of downtime in Data Centers (Ponemon Institute, 2013)

With the recent growth of online businesses, improving data center reliability has gained lots of attractions among researchers and data center designers. This section is reviewing the methods that are useful to enhance the data center reliability and availability.

2.2.1 Reliability Related Metrics in Data Centers

Following metrics are commonly used in data center operation and reliability management,

A. Failure rate

Failure rate (α) can be defined for any equipment (see equation 2-6). Which is assumed to be constant over component's lifetime (Heising, 2007).

$$\alpha = \frac{1}{MTBF} \tag{2-6}$$

where, MTTF is the mean time between consecutive failures of a component and usually is expressed in hour per failure.

B. Repair rate

Repair rate (μ) can be defined for the components with repair distribution (see equation 2-7). Similar to failure rate, μ is assumed to be constant over component's lifetime too (Heising, 2007).

$$\mu = \frac{1}{MTTR} \tag{2-7}$$

where, MTTR is the mean time to repair the faulty component.

C. Reliability

For one component with a constant failure rate α, the reliability of the system can be calculated according to the following equation (Heising, 2007).

$$R(t) = e^{-\alpha t} \tag{2-8}$$

The reliability of a system can be calculated according to the series or parallel status of its components. Reliability block diagram (RBD) is a typical way of showing the connectivity of components in a system. For example Figure 2-6 shows two components that are connected to each other in a series (see Figure 2-6 a) and in parallel configuration (see Figure 2-6 b).

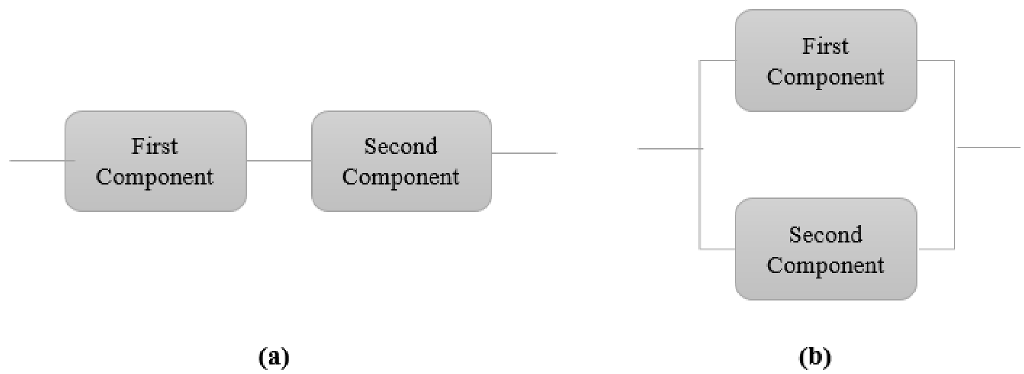

Figure 2-6: Two components series system (a), parallel system (b)

The reliability for above mentioned series and parallel connections can be derived from:

$$R(t)_{series} = e^{-\alpha_1 t} \times e^{-\alpha_2 t} \quad (2\text{-}9)$$

$$R(t)_{parallel} = e^{-\alpha_1 t} + e^{-\alpha_2 t} - (e^{-\alpha_1 t} \times e^{-\alpha_2 t}) \quad (2\text{-}10)$$

Where α_1 and α_2 are failure rates for the first and second component respectively. In the series system, the system reliability will decrease as the number of components is increased. However, having more parallel components in the system will increase reliability of the system. Therefore, to enhance the reliability, connecting redundant components in parallel would increase the overall reliability. It should be mentioned that in a system like data center which there is many components, RBD will become more complex.

D. Availability

Availability is the ratio expressed as the percentage of time a system or a component can perform its required function. Availability could be formulated with the following equation (Heising, 2007):

$$A = \frac{UpTime}{UpTime + DownTime} = \frac{MTBF}{MTBF + MTRR} \quad (2\text{-}11)$$

If the faulty component can be repaired instantly (MTTR = 0), then it wouldn't matter what the MTBF is, and the availability of system would be 100% (=1) all the time.

2.2.2 Redundancy

Backup components will be a great help to avoid costly periods of downtime in a data center. Redundancy is one of the possible methods to enhance the reliability and availability of the data centers. It refers to a system design where a component is duplicated, tripled, etc. A four-tier ranking system is proposed by Uptime institute for determining the reliability of data centers (Turner IV, 2006) which grouping data centers according to their properties (see Figure 2-7).

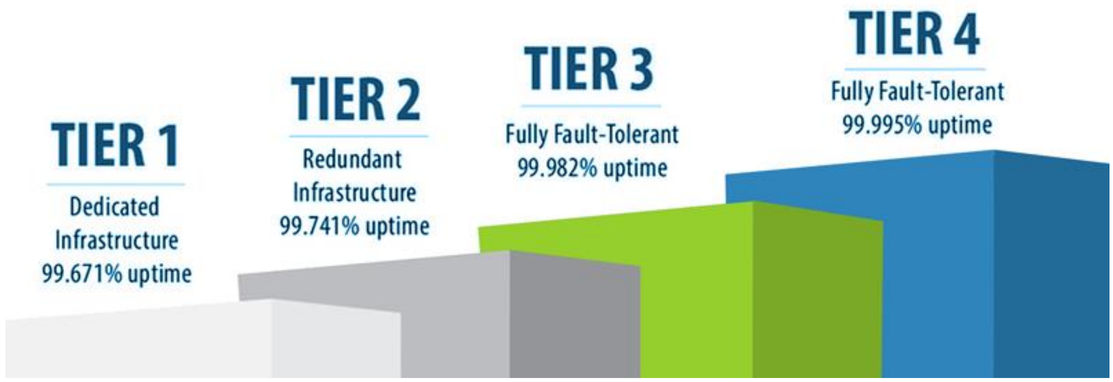

Figure 2-7: Tires of data centers according to Uptime institute (Turner IV, 2006) (Volico, 2018)

Figure 2-8: Topologies for different data center tiers Figure 2-8 shows the different topologies for data center tiers. Tiers are progressive, meaning that each tier incorporates the requirements of all the lower tiers. Tier 1 and 2 are suitable for small, cheap and short-term businesses with low reliability and availability, tire 3 (2N configuration) and 4 (2(N+1) configuration) are good for big and long-term businesses with highest reliability and availability. While high level tiers are costly, they lead to saving more money in the long term by decreasing the downtime of data center. There is a compromise between reliability of a system and total cost (Wiboonrat, 2008). Roy and collogues had an analysis on power architectures to identify in which component of data center power system redundancy is most needed (Roy, 2001).

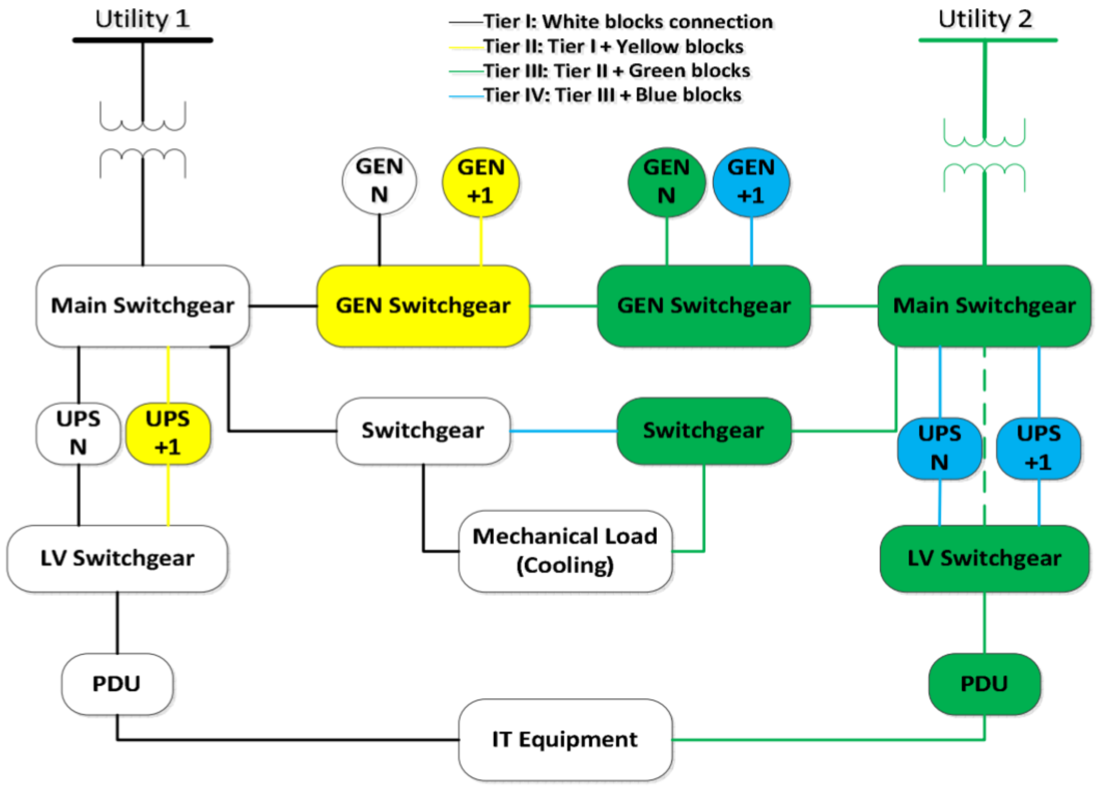

Figure 2-8: Topologies for different data center tiers (Chalise, 2015)

In the high level tiers, RBD calculations will be more complicated if the components are assumed to be repairable because the mathematical calculations become more complex. Using direct current (DC data centers) to feed data center components instead of AC power was a good idea to increase the reliability of data centers. Since, DC data centers are using less number of conversions which is an important cause of reliability improvement and decreasing the downtime of data center. Beside the high reliability and availability, high energy efficiency, less carbon footprint, lower installation cost and less maintenance cost, scalability, easier integration of renewable energy, lower operational cost and safety are advantages of DC data center over AC data centers.

2.2.3 Maintenance Programs

Not having enough redundancies for many components in a data center, could increase the failure ratio since any component may have small but non-zero failure rate. Even a small failure could have cascading effect and could interrupt the data center and the company's core business, resulting in significant loss especially for large IT-based companies. In such situation, a planned maintenance can improve the reliability of system by early detecting the abnormalities and avoiding major failures. Over the years, maintenance has evolved to embrace the concepts of reliability. In total, three types of planned maintenance can be identified, namely corrective, preventive, and predictive maintenance (Azadeh, 2016). Figure 2-9 shows the plans of a predictive maintenance which can be grouped into three categories, knowledge based condition monitoring, failure detection and failure prediction (Wang J. Z., 2017). Predictive maintenance uses predictive analytics to analyze the data gathered through the various sensors in order to predict the behavior of components and equipment in the future (LaRiviere, 2016).

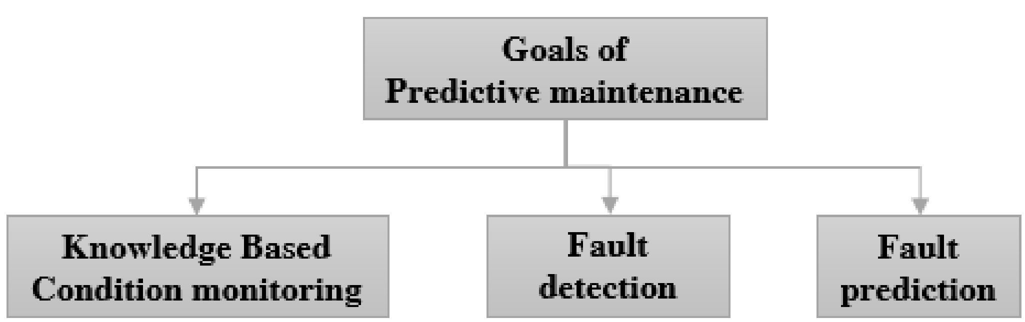

Figure 2-9: Predictive maintenance plans (Wang J. Z., 2017)

There are multiple modeling strategies for predictive maintenance which can increase the reliability of a system among which, classification, anomaly detection, and regression are the top three used methods for different industries. Following is a brief explanation of each:

- Classification models are used to predict the class that new observation belongs to. For example, predicting if a failure will occur within a given time window or not, is a binary classification problem. Another example is to look at the sensor readings and identify if the operation belongs to normal or abnormal classes.
- Regression models are used to predict continues values (and not categorical) for a target variable. For example, predict the remaining useful life (RUL) is a regression problem that tries to perfume the prediction by observing degradation process in different assets.
- Anomaly behavior detection: This method is able to flag anomalies when there is no clear historical data about the failures. Since it is assumed that most of operating data is collected in normal operation, these methods usually define the normality by measuring the distance of current operation from the center of historical data points.

The concept of predictive maintenance has existed for many years, but recently emerging new technologies such as IoT and big data analytics, make the predictive maintenance widely accessible (Nguyen, 2015).

- **Internet of Things (IoT):** Predictive maintenance has given rise to adaption of IoT platforms. IoT technology allows different assets and systems to connect, share, analyze, and act on the data (Atzori, 2010). Captured data from data center's different assets together with IoT technique will be used to monitor component's behavior continuously and provides an online health status of the equipment. This technology has made it possible to gather information from smaller sections of the data centers which previously were not focused on. Beside the advantages that IoT based predictive maintenance has, possible security problems in sensors should be considered by the designers.

- **Big data analytics for data preprocessing tasks:** Data collected from the system is not always usable before preparation. Big data preparation steps such as transformations and new feature extractions are essential in many applications (Russom, 2011). Figure 2-10 shows some of data preparation techniques with big data analysis which will be explained in detail in chapter 3.

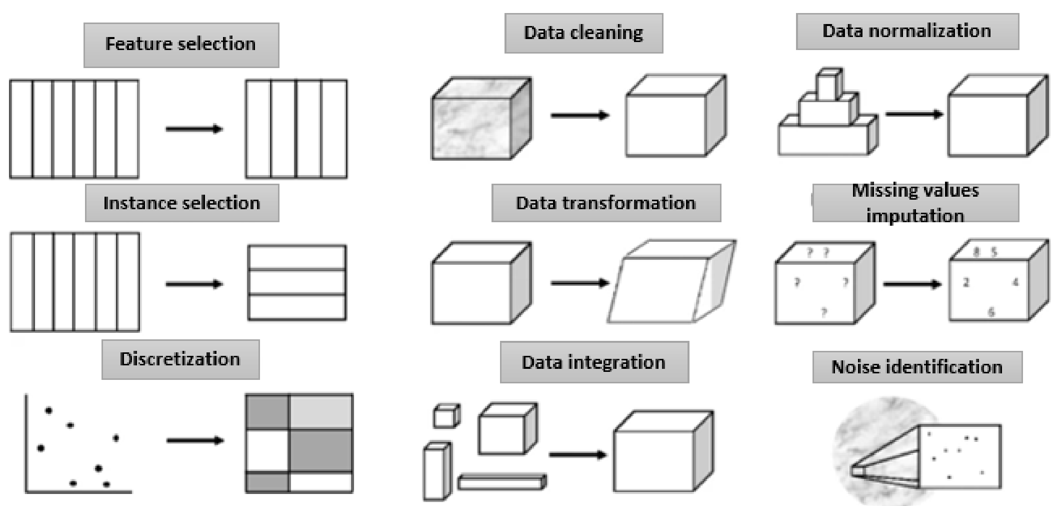

Figure 2-10: Data preprocessing tasks (García, 2016)

Reducing maintenance cost and time, reducing downtime of the data center, increasing operator safety, and increasing production quality are other advantages of the predictive maintenance (Nadabaica, 2012). Because of these advantages of predictive maintenance provides for the businesses, the global predictive maintenance market is growing and it is expected to reach 10.7 billion USD in 2024 (see Figure 2-11).

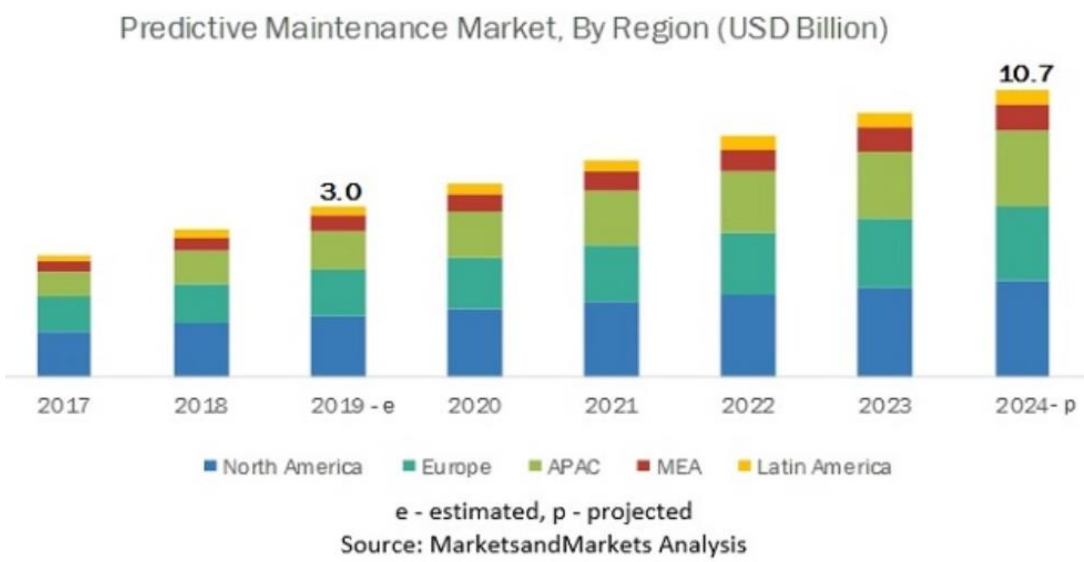

Figure 2-11: Global forecast on predictive maintenance market from 2017 to 2024 (Market, Jun 2019)

Predictive maintenance uses different strategies to detect/predict failures. Major subcategories are knowledge based condition monitoring, traditional machine learning (ML) based and deep learning (DL) based approaches (see Figure 2-12).

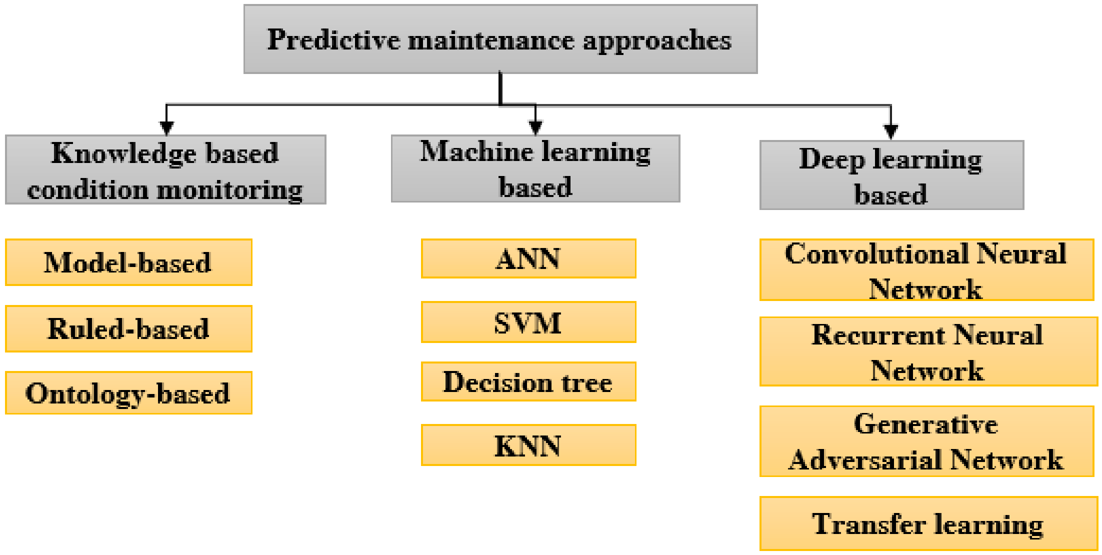

Figure 2-12: predictive maintenance approaches

A. knowledge based predictive maintenance

Knowledge based predictive maintenance uses statistical analysis for making predictions. A mathematical-based model usually requires a domain specialist that understands the asset and its environment. The creation of a mathematical model for an asset is a specific task; its main drawback is that the same model cannot be used for different assets. Defining the rules by human make this approach noise sensitive. To decrease the noise sensitivity and improve the results, most of the time, the combination of knowledge based methods with other algorithms like Kalman filter can be helpful. knowledge based predictive maintenance is also called knowledge based condition monitoring.

Collecting data in form of signals from measurable attributes of the asset to show the information about current circumstances of an active asset is condition monitoring (Marfatia., 2013). To gain the information such as temperature, voltage, vibration, current, etc. from a component different sensors may be used and actions will be recommended depends on the current situation. Condition monitoring in industrial robots (Borgi, 2017) and industrial boilers (Cauchi, 2017) are some of these studies.

B. Machine learning based predictive maintenance

Machine learning based predictive maintenance minimizes failure rates by learning from historical data. One of the advantages of ML-based predictive maintenance over knowledge-based is the less noise sensitivity (Zare Mehrjardi, 2018). It is because machine learning can dynamically define decision-rules (conditions or thresholds) to predict when a machine is fine or need to be maintained and does not need an expert to predefine the regulations. There are many machine learning algorithms where artificial neural network (ANN), decision tree (DT), support vector machine (SVM), and k-nearest neighbors (k-NN) are the most popular ones in the area of predictive maintenance (Sorsa, 1993), (Chen M. Z., 2004), (Chen K. Y., 2011), (Tian, 2014). Since machine learning based predictive maintenance is in the scope of this research, a detailed explanation is given in chapter 3. In the following, some of the published researches in predictive maintenance are reviewed:

Fault diagnosis in bearings is done by Samanta and colloquies (Samanta, 2003). They used time domain features for training the ANN algorithm to detect the fault of rolling element bearings. On the other hand, prognosis of bearing acoustic emission signals using supervised ANN is used to predicting the RUL of bearings (Elforjani, 2017). ANN algorithms have high accuracy in the expense of high computational resources. Also when training the model, the many weight parameters should be tuned properly to avoid overfitting since ANN is prone to over-fitting.

Decision tree is another machine learning algorithm which has gain considerable attraction recently since the input-out relations is easy to understand compared to other black-box relations in other machine learning algorithms. This approach uses recursive splitting of the training data and assigns a class label to leaf by the most frequent class observed and then prunes a subtree with a leaf or a branch if lower training error is obtained. But generally, trees are less accurate compared to the other machine learning approaches because of overfitting problems and should be used with extra care and testing. In (Li, 2018) fault detection of a refrigerant flow system is analyzed. Zheng and friends used decision tree machine learning algorithm for predicting lithium-ion battery failures (Zheng, 2019).

Support vector machine (SVM) performs well in small to medium size datasets but it is not a good tool for big data training since it a complex and heavy algorithm to train large datasets. Cooling failures are common in data centers and many of them are hard to detect especially partial failures such as valve blockage etc**.** About 32% of the system errors are caused by hardware and cooling problems (Gray, 1986) and a large portion of the failures are related to the abnormal air flows (Choo, 2013). SVM performs well in detecting the failures of chiller (Han, 2019) and can be also used to predict the RUL. Lithium-ion batteries RUL is predicted by SVM in (Wei, 2017). The structure of this algorithm and the way it predicts the clusters will be discussed in the next.

KNN algorithm is a machine learning technique which is easy to implement but does not have good accuracy when data is unbalanced (Susto, 2014). This is mainly due to using only a distance-based measure to classify different observations. Like the other machine learning algorithms, it can be used for both diagnosis and prognosis purposes. When detecting and predicting fault, K should be tuned depends on the data and output that should be predicted. Detecting faults of bearing based on KNN is analysed by Appana and his friends (Appana, 2017). On the prognosis side, remaining useful life estimation of insulated gate biploar transistors (IGBTs) based on a KNN is analysed (Liu Z. M., 2017).

C. Deep learning based predictive maintenance

Recently, deep learning has shown superior ability in feature learning, fault classification and fault prediction with multilayer nonlinear transformations. In recent decade, lots of deep learning approaches (LeCun, 2015) are invented to extract the complex relationships from data which other methods cannot find such a complex relationships in the data. These methods need huge datasets to train complex neural networks and if the volume of the historical data is not considerable, their performances could be less than traditional machine learning algorithms.

D. Predictive maintenance in data center

With the expansion of datacenters demands, the need for reliable data centers has continuously increased. Nlyte Software, Emerson Network Power, Schneider Electric and Device42 are popular vendors of services to increase the reliability of data center IT racks (Sverdlik, 2016). They all have condition monitoring and failure detection and prediction of IT racks using different approaches (knowledge, machine learning and deep learning based predictive maintenance). But it should be mentioned that they do not have condition monitoring, failure detection and prediction in a data center level. Ina data center different subsystems such as PDU, IT racks and Cooling unit are analyzed separately and often by different vendors and these vendors do not have holistic monitoring of a data center. That is where our research can detect some other and new failures just because of the abnormality between in the correlation of different subsystems of a data center which will be discussed in details in chapter 4.

2.3 Conclusion

This chapter provides an overview of present works on improving the efficiency and reliability of data centers and micro data centers. Considering the different subsystems of a typical data center, the efficiency improvement for different subsystems (IT room, cooling system, and power distribution unit) are explained. And on the reliability side, the effect of having backup components (redundancy) and maintenance programs are discussed. While conventional methods try to improve the reliability simply by adding more backup components, modern data centers use maintenance programs to achieve high reliabilities. This thesis focused on developing the predictive maintenance models and to this end, the advantages and drawbacks of each approach are discussed. Among these approaches, machine learning based predictive maintenance algorithms will be used in this thesis. In the next chapter machine learning based failure detection methods will be explained in more details.

Chapter 3

3 Machine Learning Based Fault Detection

3.1 Introduction

Traditionally, data center managers try to increase the reliability simply by adding redundant units such as servers, cooling or power units. As discussed in section 2.2.2, high reliability is achievable by having high level of redundancy (such as dual modular redundancy, triple modular redundancy and etc.) which imposes extra costs to the system. Unlike conventional health monitoring, smart monitoring minimizes failure rates by learning from historical data and without imposing above-mentioned overhead costs. One possibility in smart monitoring is predictive maintenance as it discussed at section 2.2.3, where the historical data is used to detect upcoming failures and schedule maintenance actions. Machine learning based predictive maintenance tries to prevent asset failures by analyzing production data and identifying patterns that could be indicative of the failures. The idea is coming from the fact that past changes provide the basis for future predictions.

3.2 Machine Learning Based Predictive Maintenance

The success of predictive maintenance models strongly depends on the framing of the problem, having the right data, and evaluating the predictions properly. The process of the predictive modeling in most of the recent researches (summarized in the section 2.2.3B) is similar and the block diagram of the steps is shown in Figure 3-1.

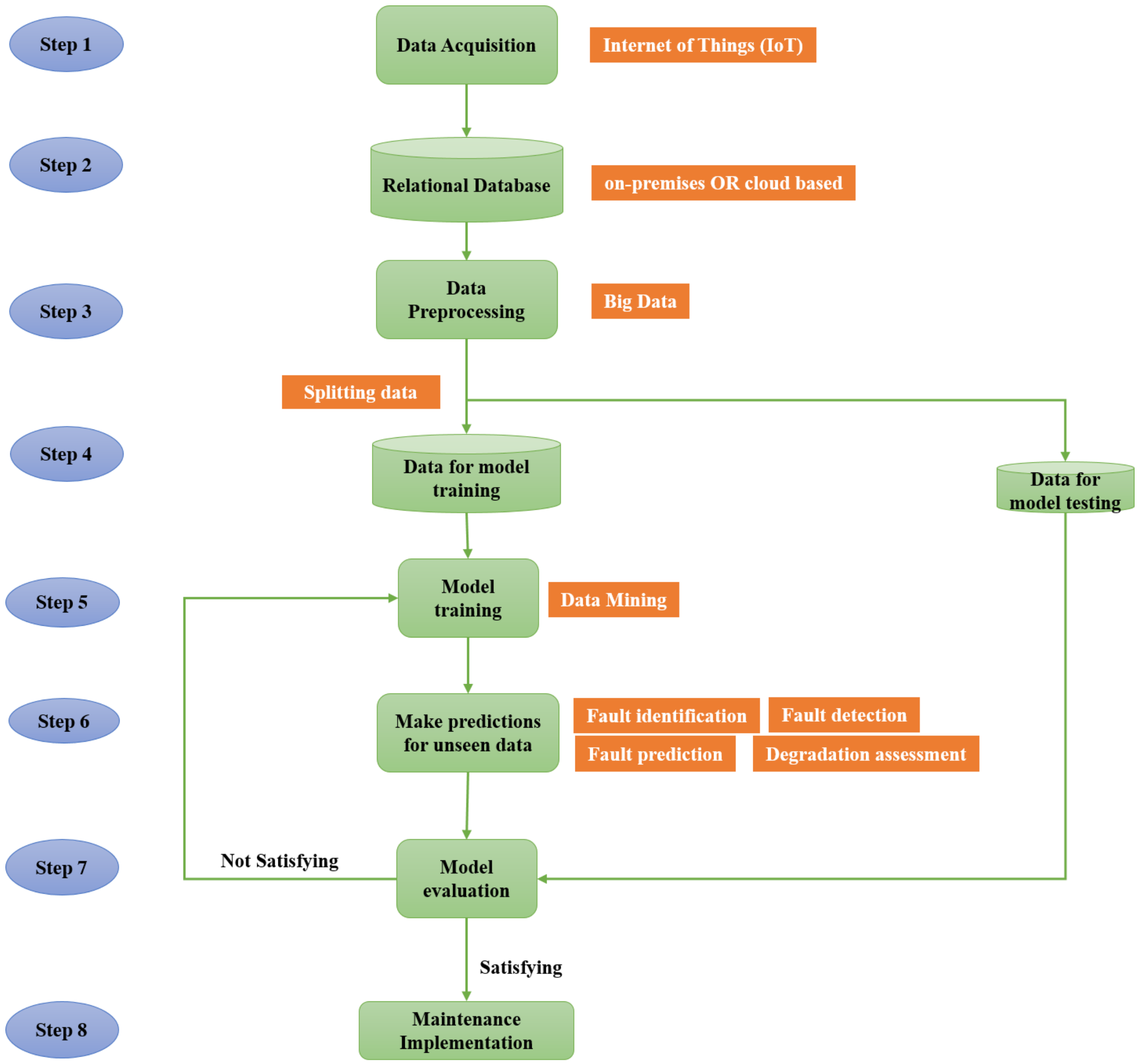

Figure 3-1: System architecture for an intelligent predictive maintenance (Ran, 2019)

Following is the brief explanation for each step:

- Step 1 – Data collection: Firstly, critical assets should be identified and then data should be collected by adding specific sensors in the appropriate locations. Servers, transformer, generator, UPS, cooling tower, CRAC units, chillers components, and pumps are some of the critical equipment in a typical data center. The process of data collection and the connectivity of these equipment to the data acquisition system are the main reasons why this step is usually called data collection by IoT devices.

- Step 2 – Creating a relational database: Collected data is time labeled signals and should be stored in relational (or tabular) databases. These data bases can be hosted using either cloud services (such as services offered by Google, Amazon, and Microsoft) or in the physical servers owned and managed by each company (on-premise solution). This work utilizes an on-premises solution hosted in a Raspberry pi-3 single-board computer which can be placed inside the cooling unit of a data center.
- Step 3 – Data Preprocessing: preprocessing of the data is an important step specially when working with sensor data which often contains errors, and noise values or has missing values. This is explained in detail in section 3.3.
- Step 4 – After preprocessing, the data will be split into train and test datasets. Typically, 70-90% of the collected original dataset is used for training and 10-30% for the test dataset. Train dataset will be used to extract the information and learn the signature of the system and train the predictive model. Where the test dataset will be used to evaluate and validate the trained model.
- Step 5 – Model training: The target of this step is to develop a predictive model by extracting the information from training (historical) dataset. This historical data can be labelled or unlabelled where, labeled data is a group of samples that have been tagged with one or more labels. Labels are tags for different classes of data (normal

and failure scenarios in the area of predictive maintenance) that machine learning algorithms try to predict. If the training data is labeled, it means we have prior knowledge about the categories that samples belong to, however, unlabelled data does not have any tags or labels associated with the observations. Depending on using labelled or unlabelled data, there are mainly two approaches to train predictive models:

- Unsupervised approach, where there is no need for labelled training samples and thus less human intervention requires in both training and testing stages of the model (Xiaojin, 2002)
- Supervised approach, where labelled data is required for training and testing stages of the model and thus necessitates more human interventions (Kotsiantis, 2007).

Since the labels are expensive to collect in many applications, unsupervised learning approach can be helpful by avoiding such expenses.

- Step 6 – Making predictions using test dataset: in this step a portion of the data which was not used in the training phase (which is usually called test dataset) will be fed to the trained model and the predicted outputs will be populated.
- Step 7 – Evaluating the trained model: evaluating the performance of a trained model is one of the core stages and can be done by calculating the accuracy of predictions on the test dataset shows if the trained model has acceptable skills in making predictions or not. In fact, steps 5 and 6 should be repeated with different

algorithms and hyper parameter settings until the results in step 7 meets a minimum accuracy requirements defined by domain experts. Generally, these minimum requirements are defined in a way that predictive models are able to outperform human based predictions in terms of accuracy, costs, or implementation scales (Van der Aalst, 2010).

In the following sections data preprocessing and model development are explained in more details.

3.3 Data Preprocessing

Collected raw data has to be processed to be ready for a machine learning algorithm. This step is important and has a vital effect on the performance of the data driven model. Data preprocessing techniques can be categorized into two group, simple and advance. While simple techniques impute missing values and scales different sensor readings, advanced preprocessing methods extract new features or drop less informative features to avoid overfittings and lower the training times (Lee J. B., 2014).

3.3.1 Feature Selection

Reducing the number of features is necessary to decrease the computational efforts and avoid overfittings in model training. Theoretically, a greater number of sensors could capture richer datasets that lead to higher accuracy in failure detection. In practical applications more sensors introduce new challenges from more storage capacity and processing time requirements to physical limitations in sensor placement. Also, some information may confuse the model and decrease the accuracy. Feature selection techniques by selecting the most relevant and important sub-features can overcome above mentioned problems. This process can be done in few different ways such as correlation analysis, principal component analysis and entropy calculation. Also, machine learning techniques such as generic algorithms and decision trees can been used for feature selection.

A. Correlation analysis

Correlation analysis (Hair, 2006) evaluates the relationship between features. High correlation means that two or more features have a strong relationship with each other, while low correlation means the opposite. Often correlations higher than 0.9 shows that one of the features doesn't have significantly new information for the predictive model and thus it could be removed from the training dataset. At the same time very low correlations between any feature and the target variable shows that such feature is not a good predictor of the target variable and could be removed from the training set. Followings are common ways in defining correlations among variables:

- Covariance: The relationship between two features can be summarized by calculating the average of the product between the values from each sample, where the values have been centered (Brownlee, 2016).

$$cov(x, y) = \frac{1}{n-1}\left(\sum_{x}\sum_{y}(x-\bar{x})(y-\bar{\bar{y}})\right) \tag{3-1}$$

Where n is the number of pairs of data (observations), $\bar{x}$ and $\bar{y}$ are the means of samples for features x and y, respectively. This technique does not have a good performance if the amount of observations is low or if there is non-linear correlation between the features or if the features do not have Gaussian distribution (Brownlee, 2016).

- Pearson's Correlation: since covariance matrix is challenging to explain, Pearson correlation is a good way to handle this problem (Rumsey, 2009), (Brownlee, 2016). It is in fact the normalized version of the covariance matrix where, covariance matrix is divided by the product of the standard deviations of the related features. Pearson's Correlation is easy to interpret because the results are between -1 and +1 with +1 being the strongest possible positive correlation and -1 being the strongest possible negative correlation. While zero shows there is no relationship between the related variables.

$$Pcor = \frac{cov(x,y)}{s_x s_y} \tag{3-2}$$

where s_x and s_y are standard deviations of features x and y respectively. Again, like covariance if the features do no not have Gaussian distribution or there is non-linear relationship between variables this technique will not have accurate results.

- Spearman Correlation: To deal with nonlinear relationship between features and non-Gaussian distributions, Spearman correlation is introduced (Brownlee, 2016).

$$Scor = \frac{cov\big(rank(x), rank(y)\big)}{s_{rank(x)}} \tag{3-3}$$

Where $rank(x)$ and $rank(y)$ are sorted values of the features x and y respectively. This is where the values are ordered and assigned an integer rank value.

B. Principal component analyses (PCA)

PCA is an algorithm to transform the data from higher to lower dimensions without losing significant information and variability (Wold, 1987). It is a linear projection method which transforms the original s dimensional dataset into less dimensional space where, s is the number of features in dataset. After projecting data points into the new low dimensional space, (new) variable with the greatest variance is called the first principal component, the second (new) variable with the high variance is called second principal component and so on. PCA can be used for reducing dimensionality by selecting the first few principal components (in most applications only two first ones) and eliminating the remaining principal components. Figure 3-2 shows the basic steps of a PCA.

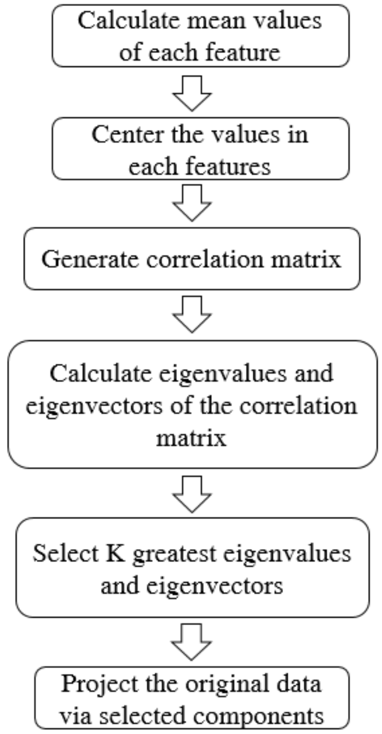

Figure 3-2: Block diagram of steps of principal component analyses technique (Wold, 1987)

Assume a dataset as a matrix X with $n \times s$ dimension where n is the number of observations and s is the number of features (number of sensors in this research). PCA steps are as follows;

- Data standardization: calculating the mean values of the features and then centering the all values is called data standardization. Since in most cases the dataset contains variables with various units and scales, it is necessary to standardize the data to give each variable the same influence. After standardization each variable has variance of one. Equation 3-4 which is done by subtracting the mean of the data from each data point calculates the standardized version (matrix D) of the original dataset (matrix X);

$$D = \begin{bmatrix} x_{11} - \bar{X}_1 & \dots & x_{1s} - \bar{X}_s \\ \vdots & \ddots & \vdots \\ x_{n1} - \bar{X}_1 & \dots & x_{ns} - \bar{X}_s \end{bmatrix} \qquad (3-4)$$

where $\bar{X}_i$ is the average of ith feature in the original dataset.

- Correlation matrix: The next step is calculating the correlation between each pairwise features and generate the covariance matrix for the whole dataset which is discussed in the previous section. This matrix is always square and symmetric.
- Calculate eigenvalues and eigenvectors: The next step is calculating the eigenvalues and eigenvectors of the created correlation matrix. The idea is that the eigenvectors with the largest eigenvalues correspond to the dimensions that have the strongest correlation in the dataset. To find the eigenvalues and eigenvectors, matrix decomposition methods like singular value decomposition (Golub, 1971) can be used. And then by choosing first k greatest eigenvalues and eigenvectors, data can be projected into the new low dimensional subspace.

$$PC = selected\ greatest\ eigenvalues\ and\ eigenvectors$$

- Data projection:

$$PCA = (PC)^T.X \qquad (3-5)$$

Where X is the original dataset before projection, $(PC)^T$ is the transpose of the chosen principal components and PCA is the projection of the original dataset. Therefore, by selecting the largest eigenvalues and using them to project dataset into the new space, the dimensionality of data can be reduced without losing significant information.

C. Entropy

It is used to quantify how much information there is in a feature. Equation 3-6 calculates the entropy (function H) of feature x (Shannon, 1949).

$$H(x) = -\sum_{i=1}^{n} p(x_i)\, I(x_i) \qquad (3-6)$$

Finding H(x) is the same as calculating the information for the probability distribution of the events for the random variable x. It gives the average number of bits required to represent an event drawn from the probability distribution for the random feature. Where n is the number of events (normal and failure sceneries in the application of predictive maintenance), x_i is a random variable, $p(x_i)$ is the probability that the event x occurs. $\mathrm{I}(x_i) = \log_2\left(\frac{1}{p(x_i)}\right)$ shows the information content of event x_i and is 0 for the definite contents and is increasing if the event is more rarely to happen. Therefore,

$$\begin{gathered} high\ E\ for\ the\ feature\ \leftarrow all\ events\ have\ same\ probability \\ low\ E\ for\ the\ feature \leftarrow more\ the\ information\ concentrates\ on\ few\ events \end{gathered}$$

So whenever a feature has low entropy, it means that it is a valuable feature in the process of model training.

D. Machine learning based feature selection approaches

Machine learning approaches, such as genetic algorithms and decision trees can also be used for feature selection (Swiniarski R. W., 2003). The goal in this approach is still fining subset of highly informative features that can be used for predictive model training.

3.3.2 Feature Extraction

Some features are easy for human to understand but typical machine learning methods have difficulties to learn from them such as time stamps that a sensor measures a physical input from its environment. While human has a solid and easy understanding of such data, machine learning algorithms usually prefer to have days, weeks, or months as separate features in the training dataset. Feature extraction methods transform the original feature space to new feature space (usually with higher dimensions) and streamline the learning process for predictive algorithms.

3.4 Machine Learning Based Predictive Models

There are different machine learning algorithms which use different structures to detect/predict anomalies (data center failures in our case) in the system. Choosing the right algorithm is one of the challenging steps in this process. Type of problem, size of training set, accuracy, training time, linearity, number of parameters, number of features are the important considerations when choosing machine learning algorithms (Ravanshad, 2018). Beside these factors, storage capacity is another important factor that should be considered. As it is mentioned earlier, depending on the type of training dataset, type of problem can be classified as supervised (Xiaojin, 2002), unsupervised (Schölkopf B. W., 2000) or reinforcement learning (Sutton, 1998).

3.4.1 Unsupervised Machine Learning Models

As it is mentioned earlier, when the historical data is unlabelled, the problem is unsupervised. In the real word, most of the time it is hard and sometimes impossible to create the labelled data. Unsupervised algorithms are useful methods in such cases and can be used for classifying unlabeled data into groups (such as normal and failure scenarios in this research). Figure 3-3 visualizes the process of unsupervised learning in a simple way (Goldstein, 2016). First step is training a machine learning algorithm to create a data driven model. Applying the trained model to the test data and evaluating the performance is the next step. If the results are not satisfying, the model should be improved (either by trying different algorithms or hyper parameters) again.

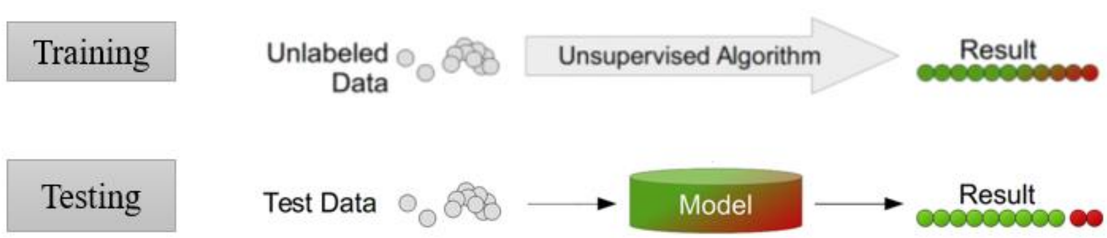

Figure 3-3: Unsupervised anomaly detection technique (Goldstein, 2016)

Unsupervised anomaly detection approaches can be grouped into four families (Falcão, 2019), distance-based, and classification-based approaches (see Figure 3-4). Between these groups distance-based and classification-based approaches are the most useful methods in the area of anomaly detection and predictive maintenance. Looking at Figure 3-4, the most widely used unsupervised algorithms especially in the area of predictive maintenance are given for these two group. To make a decision about which one works better, their performance should be checked in the specific application. Following is the explanation of these methods and have a comparison at the end to determine which one more probably performs well in which kind of datasets.

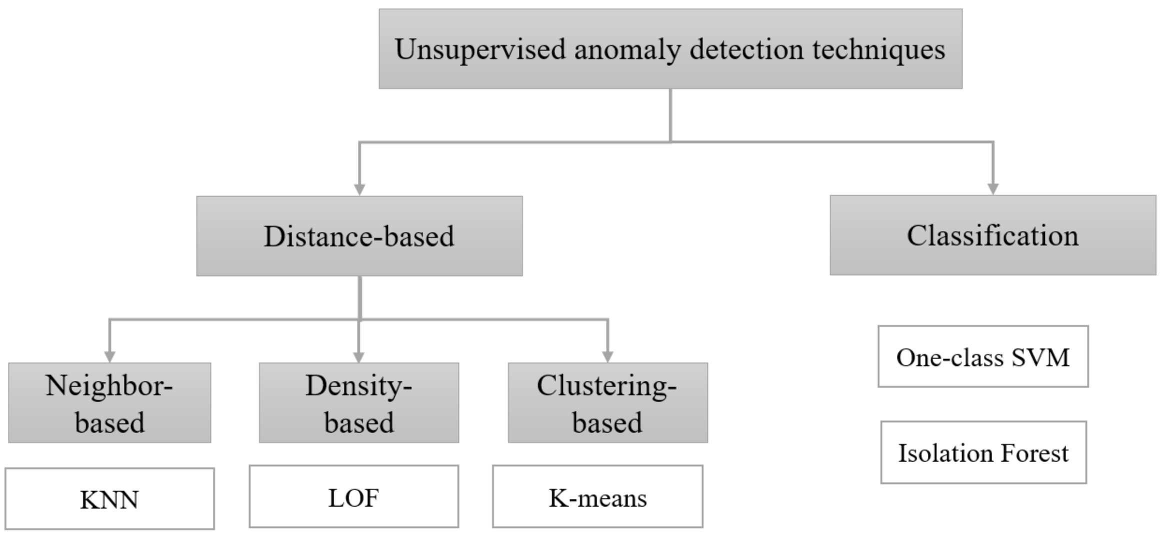

Figure 3-4: Famous anomaly detection machine learning algorithms (Falcão, 2019)

A. Distance-based Approaches

Distance measures play an important role in machine learning. There are three different distance-based machine learning algorithms, neighbor based, density based, and clustering based. It also should be mentioned that sometimes combination of these algorithms may improve the performance of the data driven model. For example, combination of a neighbour-based algorithm with an angle based algorithm may lead to better predictive models (Kriegel, 2008). The most widely used algorithms of distance based approaches in the area of predictive maintenance are k-nearest neighbor (KNN) and K-Means which will be discussed in the following sections.

- Neighborhood Based Clustering, finds clusters based on the neighborhood characteristics of the data (Zhou, 2005). KNN is the most common approach in this group. While this approach is known as a supervised learning method, it also can be used as an unsupervised technique (Prerau, 2000). In this approach by assuming that the training dataset belongs to only one cluster (normal), there is the potential to detect failure situation of the system without any prior knowledge of their existence. In an unsupervised KNN, the whole process is learning the unlabelled training dataset and then try to label a new unseen data point by looking at the K-closest labeled data points. This approach uses Euclidean distance (Zhang Z. H., 2006) or other distance metrics such as Manhattan, Minkowski, or Hamming distance to calculate the distance between a typical data point A and its neighbors. If this distance is larger than a threshold, it shows point A is far from its neighbors and thus, it could be flagged as anomalous point. Equation 3-7 calculates Euclidean distance between two n dimensional points (Zhang Z. H., 2006).

$$Edist(a,b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_n - b_n)^2} \qquad (3-7)$$

where a_i and b_i refer to the values of the i^{th} feature for observations a and b. The number of neighbors (K) should be define by human. For example, if K = 1, then the case is simply assigned to the class of its one nearest neighbor but if K is higher than 1, the average should be considered. In most of the cases the accuracy will be checked for different values of K to find the optimal number.

Unsupervised K nearest neighbor algorithm tries to find the distance between any point of a dataset with the K nearest neighbors itself. Now, imagine the average distance of a typical point A with its K neighbors is bigger than a specific threshold. This would be a sign of anomalousness since the point A is far from its neighbors. Finding the nearest neighbor is not trivial task and the most naïve way is to calculate the distances for all pairs of point A and the rest of the data point (Brute force method). In order to reduce this complexity, algorithms such as BallTree, and KDTree limit the search space by dividing the feature space to smaller subspaces (J. Goldberger, 2005). Table 3-1 below shows the time complexity of these methods where D and N are the number of features and observations respectively.

Table 3-1: Time complexity for different algorithms of KNN

	Brute-based	**KD Tree**	**Ball Tree**
Time complexity	$O[DN^2]$	$small\ D: O[D\log(N)]$ $large\ D: O[DN]$	$O[D\log(N)]$

Where in low N (N<30 or so), brute based approaches can search the whole feature space and would result in better outcomes. Data structure is also having large impact on the time complexity when using tree-based approaches. Also, number of K which is user defined constant, has high impact on the runtime of tree based approaches because trees will become slower when they require to check larger portion of the parameter space.

- Clustering based approaches: K means (Marroquin, 1993) is the most widely used algorithm in this family which takes the unlabeled data points and tries to group them into K number of clusters. The first step is randomly initializing k cluster centroids and assigning the first training data point to the nearest cluster centroid (training step). This distance can be calculated by Euclidean distance (equation 3-7). Then, new train data point is assigned to the cluster which has the shortest distance to the cluster centroid. After that initialized cluster centroids will be recalculated and modified once again new data points will be assigned to the clusters. This process is repeating till the centroids are not changing anymore. This method is very easy to understand but the main drawbacks are high running time and the need to specify the number of clusters in the data. That is why this algorithm should be tried for different K values which again imposes extra running time in the training step.
- Density based: Any machine learning algorithm which uses the concept of density reachability and density connectivity to cluster unlabeled data points is categorized into density-based algorithms in unsupervised learning. Local outlier factor (LOF) (Breunig, 2000) is the most widely used density based unsupervised machine learning approach for the application of anomaly detection and predictive maintenance. Figure 3-5 shows LOF step in 6 stages. The problem is calculating

LOF score for each point to find the outliers (failures in the area of predictive maintenance) based on the scores.

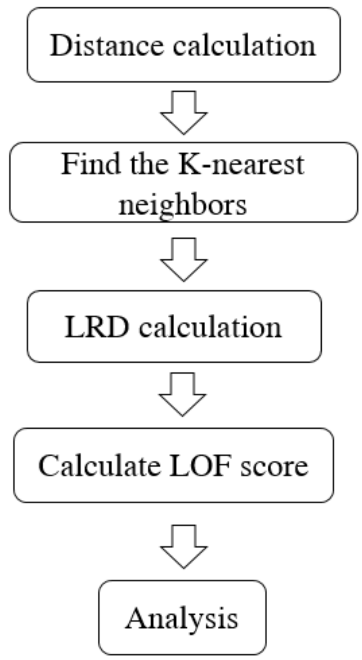

Figure 3-5: Local outlier factor steps (Breunig, 2000)

First, the distance between all pairs of observations is calculated. By initializing a value for K and finding the K nearest neighbor for each point, the distance between the observation and k-nearest neighbor will be calculated. The next step is calculating local reachability density (LRD). In other words, LRD is an optimal distance in any direction from a data point to its neighbors and illustrates how far is the point from its neighbors. Mathematically it can be formulated by equation 3-7 which is the inverse of the average reachability distance.

$$LRD = 1 \Big/ \frac{\sum_{i=1}^{k} reach\_distance(neighbor_i \leftarrow point)}{K} \qquad (3-7)$$

Where $reach\_distance(neighbor_i \leftarrow point)$ is the maximum of distance(point, kth neighbor) and distance(point, ith neighbor). To measure this distances Manhattan technique is used. Equation 3-8 formulates this technique for two n dimensional data points (S., 2011).

$$Manhattan\ distance(A, B) = \sum_{i=1}^{n} |a_i - b_i| \qquad (3-8)$$

After finding LRD for the data points, these values are used to calculate local outlier factor (LOF) and tells how likely a certain data point is an outlier/anomaly. In other word, LOF score of a data point shows the LRD of that point compared to the LRD of its neighbors. LOF for a specific point can be calculated by equation 3-9 which is the average LRD of the neighbors divided by the point's own local reachability density.

$$LOF = \frac{\sum_{i=1}^{k} \frac{LRD(neighbor_i)}{LRD(point)}}{\mathrm{k}} = \frac{\sum_{i=1}^{k} LRD(neighbor_i)}{\mathrm{k} * LRD(point)} \qquad (3-9)$$

and if the density of the point is smaller than the densities of its neighbors, LOF will be much higher than 1 and illustrates the point is far from dense areas (normal points in our case) and can be detected as an outlier data point (faulty situation). But to cluster a point as a normal or abnormal group, a value as a threshold should be defined which is a drawback of this technique. Setting this threshold mainly depends on data. While LOF value of 1 or less is a good indicator of an inlier (normal situation), if the dataset is clean, threshold can be 1.01 but if not, LOF value of 2 or more could be indicative of an inlier. It is obvious that if there are fluctuations in the density of the train dataset, this technique might not be the best detection method.

B. Classification-based Approaches:

- One-class SVM is an unsupervised algorithm which was first introduced by Schölkopf et al. (Schölkopf B. P.-T., 2001) for estimating the support of a high-dimensional distribution. The aim of using any unsupervised anomaly detection method, as mentioned in the previous section, is building a model for normal data points (class number one), and then flag points that do not follow the created model as anomalies (class number two). One-Class SVM tries to find an optimal closed boundary around the normal data points which is known as decision boundary. Collected data from the most of the real-world applications has non-linear characteristics and thus, non-linear decision boundaries are needed to separate the normal data points from faulty points. Finding the optimal decision boundary can be formulated into an optimization problem. The goal is to find a ball (hypersphere) to be as small as possible while at the same time, including most of the training data (normal data points). To find the optimal hyper-sphere in an n dimensional feature space, One-Class SVM (Schölkopf B. W., 2000) separates a desired fraction of the data points from the origin ρ with maximum margin. Because hyper-sphere cannot be always found in the original feature space, thus mapping data $\Phi(x)$ via a kernel function is needed (see Figure 3-6). There are three common kernel functions, linear, polynomial and Gaussian.

- Linear kernel; $K(x, y) = (x.y)$
- Polynomial kernel; $K(x, y) = (x.y + 1)^d$ where d is the degree of polynomial
- Gaussian RBF[1] kernel; $K(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}$ where σ^2 is the variance.

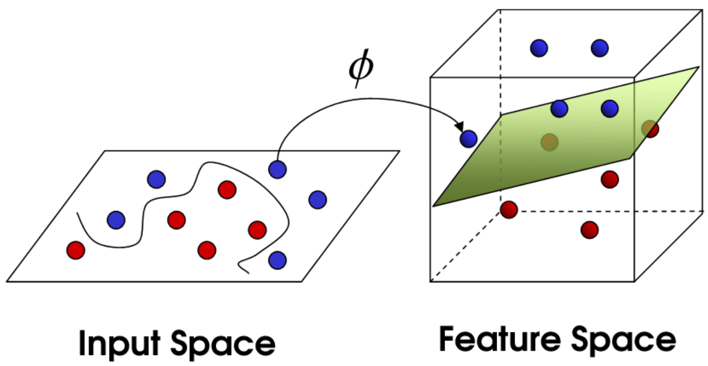

Figure 3-6: Mapping original data from input space to feature space (Wilimitis, 2018)

Because most of the real datasets are not linearly separable, a nonlinear kernel is needed (polynomial or Gaussian RBF kernels) to map the dataset from original input space to the new high dimensional feature space. Figure 3-7 illustrates the feature space with a nonlinear kernel function. The goal is finding the smallest hypersphere enclosing the data (green circle) which is equivalent to finding the hyperplane that separates the data from the origin (yellow line). The distance

[1] radial basis function

between the hyperplane and the origin is $\rho/||w||$, where w is the normal vector to the hyperplane and R corresponds to the radius of hypersphere.

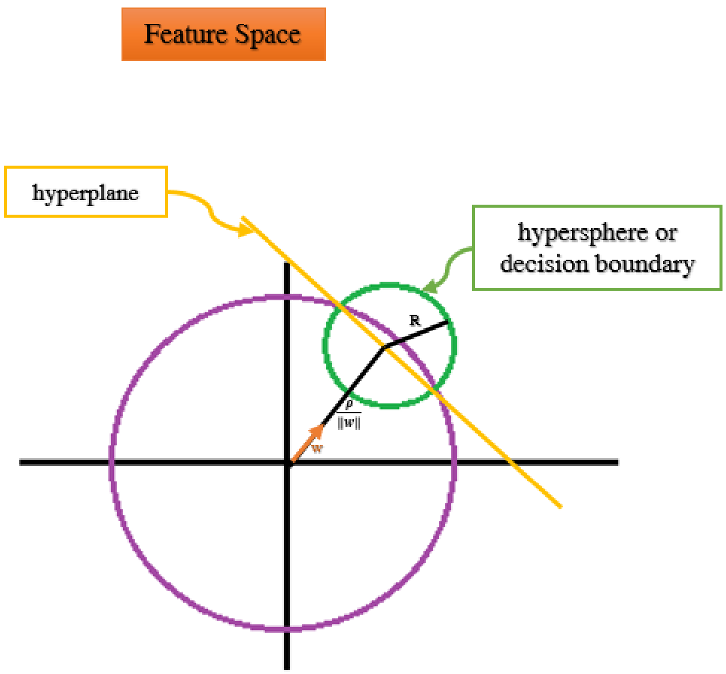

Figure 3-7: Illustration of the One Class SVM with nonlinear kernel (Mourão-Miranda, 2011)

This hyper-sphere can be written as following equation:

$$\|\varphi(x) - \rho\|^2 = R^2 \tag{3-12}$$

One-class SVM tries to minimize the radius R while keeping the majority of the observations inside the hyper-sphere. This condition can be formulated as following equation (Schölkopf B. W., 2000),

$$\underset{R,\varepsilon_i,\rho}{\text{minimize}}\, R^2 + \frac{1}{vn}\sum_{i-1}^{n} \varepsilon_i \qquad (3-13)$$

Subject to the following constraints:

$$\|\varphi(x) - \rho\|^2 \leq R^2 + \varepsilon_i \quad for\ i = 1, \ldots, n$$

$$\varepsilon_i \geq 1 \quad for\ i = 1, \ldots, n$$

Table 3-2 shows the parameters and their meaning in optimization.

Table 3-2: Unsupervised Optimization Parameters

Parameter	**Meaning of the parameter**
φ	Feature map
(x_i)	Training samples
$\varphi(x_i)$	Gaussian kernel for mapping x_i to feature space
$\boldsymbol{\rho}$	Center of hyper-plane
$\boldsymbol{R}$	Radius of hyper-plane
$\boldsymbol{\varepsilon_i}$	Slack variables used to penalize observations
v	Trade-off parameter ∈ [0,1]
n	Number of training patterns

With small values of v, the algorithm tries to put more data into the hyper-sphere and with larger v, it tries to minimize the size of the hyper-sphere. The optimization problem (Equation 3-13) is a constrained optimization problem and can be solved by the Lagrangian multiplier method. Figure 3-8 shows the process in a 2-dimensional feature space. Looking at the input space, green points can be classifying as one group (normal in this case) and rest can be classifying into the failure group. The target here is finding the best hyper-sphere and hyperplane after mapping data points into the new feature space. Looking at the feature space, green points represents normal while orange ones illustrates the abnormalities. This is how one-class SVM can detect failures in the application of predictive maintenance.

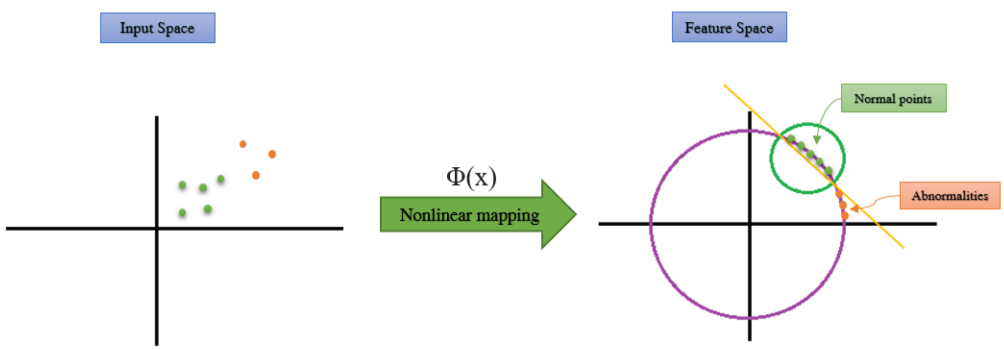

Figure 3-8: Illustration of detection normal and faulty points using one-class SVM algorithm

After finding the optimal hyper-sphere (decision boundary), the distance between any new given point and this boundary is defined as anomaly score.

$$f(x_i) = \|\varphi(x) - \rho\|^2 - R^2 \qquad (3-14)$$

for a given x_i, negative $f(x_i)$ will be classified as a normal point, otherwise, it will be classified as an abnormal point. $f(x_i)$ is also called as a decision function. In this research we will use the above-mentioned decision boundary to define the health index $H(x_i)$ of the system which will be discussed in Chapter 4.

- Another most widely used classification method is iForest technique (Liu F. T., 2008) which can be used in both supervised and unsupervised problems. The idea is generating tree by creating partitions repetitively (by first randomly selecting a feature and then selecting a random split value between the minimum and maximum value of the selected feature) till every single observation isolated. By stopping the algorithm from partitioning, the tree is generated. In the generated tree, every leaf represents the observation and the distance from the root node to every observation (every leave) defines an anomaly score for the observation (see equation 3-14). Because this tree will be generated for so many times to make it robust against randomness of whole process, every time only a fraction of samples will be used and at a very end the average of these trees will be the final tree. This algorithm has low linear time-complexity and requires small memory thus could be very useful in online anomaly detection algorithms. Equation 3-14 (Liu F. T., 2008) shows the decision function $H(x_i)$,

$$H(x_i) = 2^{-\frac{E_i(h(x))}{c(n)}} \qquad (3-14)$$

Where $h(x)$ is number of edges from leaf node to the root node. If it is close to 1, it indicates anomalies, and when it is close to 0, it is safe to consider the point and normal observation. $\frac{E_i(h(x))}{c(n)}$ is average of $h(x)$ for observation i and $c(n)$ is the average of $h(x)$ and can be calculated by equation (3-15) (Liu F. T., 2008).

$$c(n) = [2\ln(n-1) + 1.154] - \frac{2(n-1)}{n} \qquad (3-15)$$

3.4.2 Supervised Machine Learning Models

If we have prior knowledge of what the output values for our samples should be (having labels for the samples), the problem is a supervised one. Supervised machine learning algorithms are easy to understand compared to the unsupervised however, creating labelled dataset is always expensive and sometimes impossible. Supervised learning problems can be grouped into Regression and Classification problems. If the output variable is a category the problem is a classification problem wherein prediction of real values as a target, is called regression problem. Figure 3-9 shows the list of commonly used supervised machine learning algorithms in both regression and classification problems (Brownlee, machine learning mastery, 2016). For both regression and classification techniques, ensemble learning method is also introduced to improve the accuracy in detecting abnormalities in the system (DataQuest, 2019). In most of the cases combining the predictions of multiple machine learning models avoids overfittings and improves the performance.

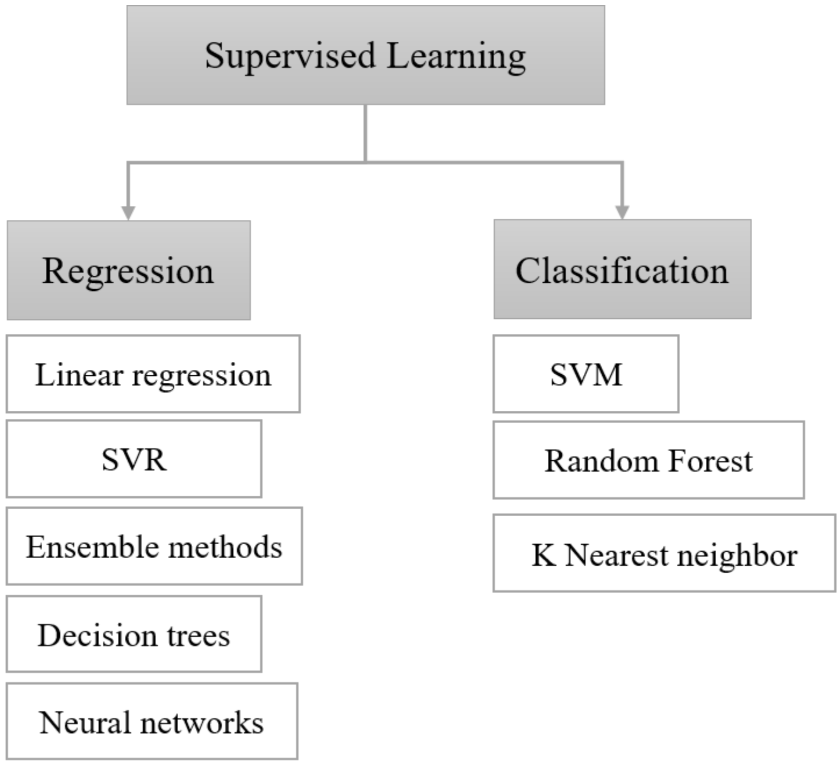

Figure 3-9: Supervised learning machine learning algorithms (Brownlee, machine learning mastery, 2016)

A. Regression

A regression problem is when the output variable is a real or continuous value. Many different models can be used, the simplest is the linear regression. It tries to fit data with the best hyper-plane which goes through the points and has minimum distance from each observation. Support vector regression (SVR), ensemble methods, decision trees and neural networks are other famous algorithms in this group (Roman, 2019).

B. Classification

A classification problem is when the output variable is a category (Brownlee, machine learning mastery, 2016). Given one or more observation from each category to the algorithm and train the model will allow the model to classify new unseen data into the right categories. Support vector machine (SVM), random forest, and nearest neighbors are most popular machine learning algorithms to solve supervised classification problems. In this research, our problem is defined as classification problem which tries to classify observations into normal and different failure clusters.

- Support vector machine: Most of the time it is hard to find decision boundaries in the real space, so SVM algorithm is used to first transform the original data into new space (higher dimensional space). This transformation will lead to gain linearly separation even if the data is not basically linearly separable. The kernel function which defines inner products in the transformed space is used in this research as transform function. The optimal decision boundary can be defined from the optimization problem 3-17.

$$\underset{\omega,\varepsilon}{\text{minimize}} \sum_{i=1}^{n} \omega_i^2 + c \sum_{i=1}^{n} \varepsilon_i \qquad (3-17)$$

Subject to the following constraints:

$$y_i(w.x_i + b) \geq 1 - \varepsilon_i \qquad for\ \ i = 1, \dots, n$$

where Table 3.2 shows the parameters and their meaning in supervised optimization problem (equation 3-17). C is a hyper parameter that decides the trade-off between maximizing the margin and minimizing the mistakes when drawing the decision boundary. when C is small, miss-classifications are given less importance and focus is more on maximizing the margin, whereas when C is large, the target is to avoid miss-classification at the expense of keeping the margin small.

Table 3.2: Supervised optimization parameters

Parameter	**Meaning of the parameter**
$\boldsymbol{x_i}$	Training samples
$\boldsymbol{\omega}$	Radius of hyper -plane
$\boldsymbol{\varepsilon_i}$	Slack variable used to penalize observations
$\boldsymbol{C}$	Trade-off parameter $\in [0,1]$
$\boldsymbol{n}$	Number of training samples
$\boldsymbol{y_i}$	Labels for the training samples

- K-Nearest Neighbor: Supervised KNN works similar to unsupervised KNN which is discussed in the previous section. The only difference is in the training step where there are samples of different classes while unsupervised KNN training dataset has only normal data points and the target is clustering new unseen points into normal and faulty groups. In the supervised KNN by looking at the K nearest neighbors of the new observation, it can be assigned to the group which most of the neighbors belong to.

- Random Forest: Tree based algorithms are other powerful machine learning algorithms. Random forest (Breiman, 2001) is a tree based approach and is useful method in classifying data into the groups. Random forests are an ensemble learner built by decision trees. So, to understand a process which random forest has, decision trees should be discussed first (Quinlan, 1986). Decision tree splits the input training dataset to create a binary tree. Each time splitting the data, subsets of variables randomly will be selected, so the process will produce different trees. Decision trees learn how to best split the dataset into smaller and smaller subsets to predict the class which a new observation most likely belongs to. When producing a decision tree, splitting the data will be continued till the maximum depth of the tree is reached. And each individual created tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction. One of the important factors in well organizing a tree is a measure called "gini index" which shows the impurity criterion for the two descendent nodes is less than the parent node.

3.4.3 Reinforcement Machine Learning Models

The reinforcement learning technique is a type of machine learning technique which tries to detect/predict results for the input data while there is no training dataset to train the model. Therefore, the machine learning algorithm is based on a trial and error scheme (Sutton, 1998). In the beginning the model will be created in respect to some random aspects. This created model predicts results and then the comparison between these results and the expected results will be used to improve the model. This is why this kind of algorithms are useful in combination with other learning algorithms to enable an efficient adaptive control system for the created data driven model which learns from its own experience and behavior. In this research we used supervised and unsupervised learning methods, so reinforcement learning is only cited and not described in details.

3.5 Conclusion

This chapter provided an overview and classification of different types of failure and relative predictive maintenance models with their advantages and drawbacks. As established in Chapter 2, this thesis focuses on the machine learning to learn health and unhealthy signature of data center operation. Since the streaming data from sensors are prone to noises and interruptions (missing values), data preparation methods are discussed to prepare the raw data for machine learning algorithms. And then, the different machine learning strategies that are relevant to predictive maintenance (detecting and predicting the failures) are introduced. The learning process can be done using two different types of datasets, a dataset with having labels (tags for normal and failure scenarios) or a dataset without labels. Since collecting labelled data is usually an expensive process especially in the area of predictive maintenance which the labels are different failure types, a process of unsupervised learning which leads to collecting labels and ultimately supervised learning will be discussed in the next chapter.

Chapter 4

4 The Proposed Technique to Detect Anomaly Operation of a Micro Data Center and Fault localization

4.1 Introduction

Data centers are the pillars of today's technology and the global data center market is expected to have dramatic growth in the next few years. There are different sizes of data centers in the world serving as enterprise or colocation facilities. Beside the large corporations with large data centers, small and new start-ups also are trying to have their own (micro) data center. As it is worth to notice again, in today's industrial world, unplanned data center outages can be expensive. Fortunately, the reliability of data centers can be improved by the application of predictive maintenance. Detecting failures and localizing them are two main goals in using predictive maintenance. There are various ways to detect and localize failures which are discussed in Chapter 3 and between the mentioned approaches this thesis targets machine learning based methods to detect and localize the failures in the data center.

Machine learning algorithms first should be trained and then the trained model can be used to detect and localize the failures. By locating different sensors in different places of the data center, both train and test data can be collected. The trained datasets can be labelled or unlabelled. If the data is labelled, supervised machine learning algorithms can be used while using unlabelled datasets, unsupervised machine learning algorithms will be used to train predictive models. It is important to note that collecting labelled data is usually expensive. Therefore, since the labeled data is expensive to collect, in this research, data center anomaly operation first will be identified by an unsupervised machine learning method. And then the fault will be localized using a supervised machine learning algorithm. Figure 4-1 shows this process where Figure 4-1 (a) produces health indexes using unsupervised learning method and creates labeled dataset for the supervised learning algorithm in Figure 4-1 (b).

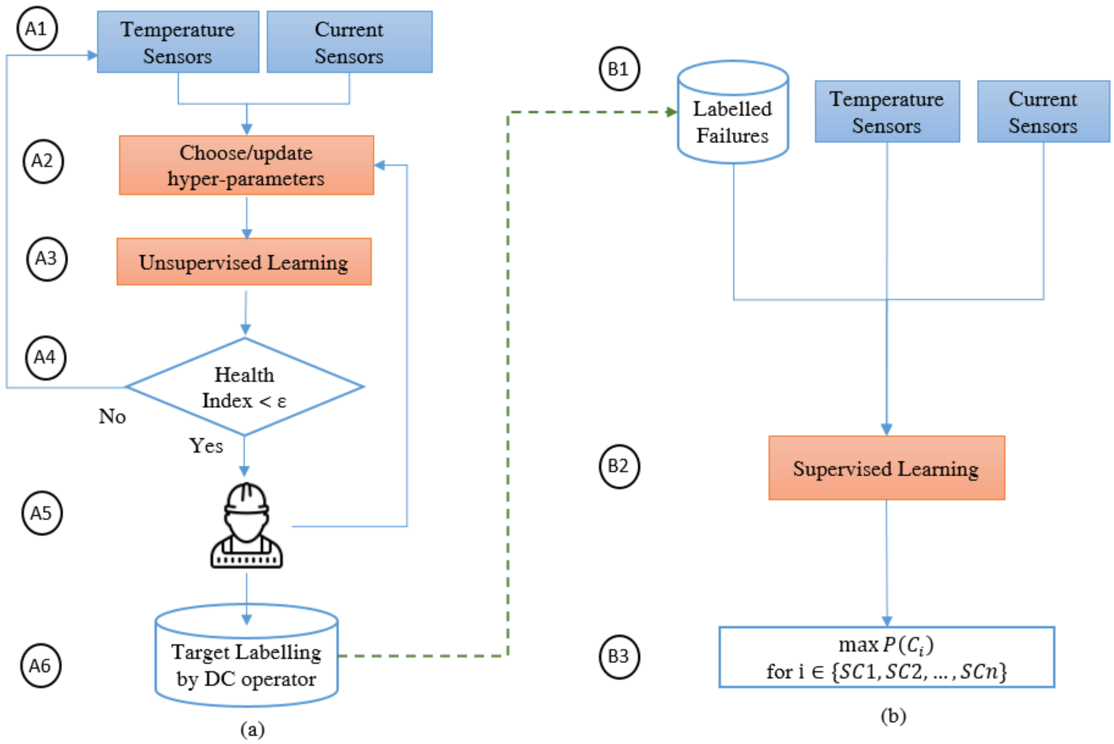

Figure 4-1: (a): unsupervised learning to define health index and assign new given data point to normal or abnormal clusters (b): supervised learning to localize failures.

Temperature and current sensors collect raw data from different subsystems in micro data center (A1 in Figure 4-1). Next, we choose typical values for hyper-parameters of unsupervised method (A2 in Figure 4-1) which will be updated based on the feedback received from the technician (A5 in Figure 4-1). In step A3, collected data from normal operation is supplied to unsupervised learning methods, where the decision boundary for separating normal versus rest of the dataset (abnormal portion) will be learned. Since the majority of observations are from normal operation, the contamination percentage of unsupervised algorithms can be set to smaller values. Most probably, the flagged contaminations have been originated from transient behaviours where a specific normal operation is transitioning to new normal operation. If the trained models detect anomalies, a technician will be notified and s/he needs to clear the anomaly and report the incident (labeling the data in A6). If s/he observes more false positives, the thresholds and hyper-parameters should be tuned (A5 in Figure 4-1).

The process of labeling should be continued until enough data is collected for the training of supervised algorithms (B1 in Figure 4-1). In this thesis, we have intentionally created failure scenarios explained in Table 4-1 to be used as the data labeled by technician in real world application. Supervised algorithms will be used for specifying the type of the failure which usually points out to the location of the failure as well (B2 in Figure 4-1). Classification methods produce probabilities of belongship to different operating scenarios (B3 in Figure 4-1) and will help technicians to quickly locate and clear the failures or anomalous operations.

Table 4-1: Failure scenarios tested on micro data center located at McMaster Innovation Park

Scenario	**Definition (# of samples – Duration (min))**
SC1	Normal operation of micro data center (1097 - 60)
SC2	Valve blockage of cooling system (202 – 10.77)
SC3	Compressor electrical failure (110 – 5.8)
SC4	Pump electrical failure (202 – 10.77)
SC5	Server rack door left open (90 – 4.8)
SC6	Instant server reduction to 60% (142 – 7.57)
SC7	Simultaneous failures of SC2 and SC5 (51 – 2.72)
SC8	Simultaneous failures of SC2 and SC5 (61 – 3.2)
SC9	Simultaneous failures of SC2 and SC5 (70 – 3.73)
SC10	Simultaneous failures of SC2 and SC5 (67 – 3.57)
SC11	Compressor failure at rest (138 – 7.36)
SC12	Simultaneous failures of SC5 and SC6 (51 – 2.73)

4.2 Defining Data Center Health Index

Different unsupervised anomaly detection methods have different structures to detect anomalies as discussed in previous chapters. After training data any driven model, it can give a value for each new unseen data points which indicates the chance of being normality or abnormality and each method has different mathematical way to calculate anomaly scores for new unseen data points. In this thesis, these scores are used to define the health index of micro data center. This equation maps the anomaly scores $f(x_i)$ to values between 0 and 1 and is defined in each specified subsections. In this research we will trigger anomaly events if health index stays below 0.8 for more than 30 seconds. These thresholds are for unsupervised method and have been selected after observing overall behaviour of normal operation for the micro data center. In practical applications, these numbers would be refined after few weeks (usually less than a month) operation.

4.3 Developed Data Acquisition System

As Figure 4-1 shows, the first step in training machine learning algorithms is collecting data from the sensitive sections of the physical environment. In this research, data is collected from 13 different sensors placed in power, cooling, and IT units of micro data center. Optimal number of these sensors can be studied considering the performance of unsupervised and supervised methods and will be explained in section 4.5.4. Table 4-1 shows the type and the location of these sensors. Among the sensors shown in this table, S5-S8, and S9 were already available in the micro data center and the rest of the sensors were added to the specified locations.

Table 4-2: Sensor type and location

Sensor	Type and Location of the sensor
S1	Input power to IT Rack (A)
S2	Input power to Compressor (A)
S3	Input power to Pump (A)
S4	Input power to Condenser Fans (A)
S5	IT Rack Temperature (Back, Down) (°C)
S6	IT Rack Temperature (Front, Up) (°C)
S7	IT Rack Temperature (Front, Down) (°C)
S8	IT Rack Temperature (Front, Middle) (°C)
S9	Condenser Temperature (°C)
S10	IT Rack Temperature (Back, Up) (°C)
S11	Ambient Temperature (°C)
S12	Cold Water Temperature (into IT rack) (°C)
S13	Hot Water Temperature (out of IT rack) (°C)

Another contribution of this study is the development of a data acquisition system (DAQ) which is designed in the form of an electronic shield for the raspberry-pi3. The developed electric shield is shown in Figure 4-2 and has an analogue to digital converter (MCP-3008) which is able to log 8 channels to RP3.

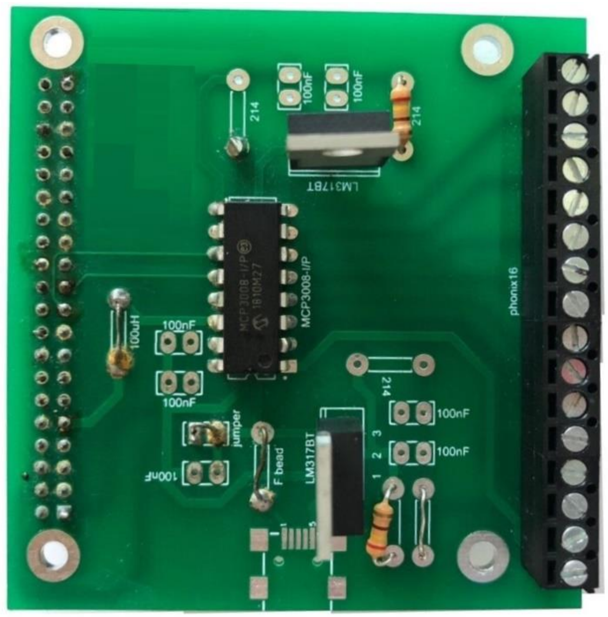

Figure 4-2: Developed analogue to digital converter system

Raspberry-pi3 (see Figure 4-3) acts as a local storage and processor of the collected data. Below are the major features of the raspberry-pi3 used in this study (RUNDLE, 2016):

- Size: 85.60mm × 56.5mm
- Weight: 45g
- CPU: Quad-core 64-bit ARM Cortex A53 clocked at 1.2 GHz
- GPU: 400MHz Video Core IV multimedia
- Memory: 1GB LPDDR2-900 SDRAM (i.e. 900MHz)
- USB ports: 4
- Video outputs: HDMI, composite video (PAL & NTSC)
- Network: 10/100Mbps Ethernet and 802.11n Wireless LAN
- Peripherals: 17 GPIO plus specific functions, and HAT ID bus
- Bluetooth: 4.1
- Power source: 5 V via Micro USB or GPIO header

Figure 4-3: Raspberry-pi3

Figure 4-4 shows the developed data system which the upper side is the developed shield and the bottom side is the raspberry pi 3 which are connected through 40 GPIO (general purpose input output) header. This system not only collects real-time data (no perceivable delay in the data collection process), but also cleans and processes them to have an end to end hardware/software solution for anomaly detection in micro data centers.

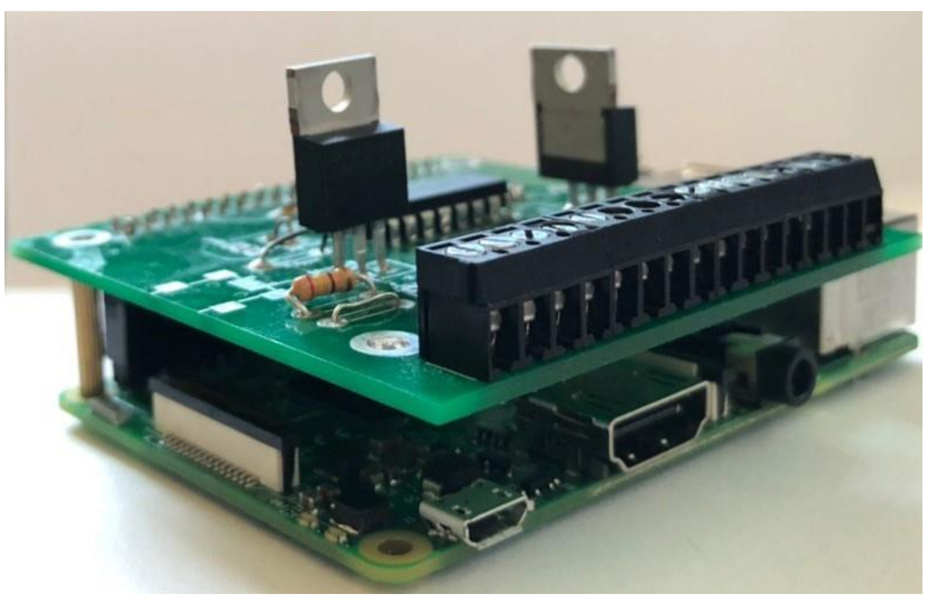

Figure 4-4: The developed data acquisition system

It is important to note that collected raw data is usually polluted with noises and outliers that can bias the training of machine learning algorithms. To avoid above mentioned problems, anomaly quantities have been removed from individual sensor values whenever they are greater than $Q_3 + 1.5 \times (IQR)$ or less than $Q_1 - 1.5 \times (IQR)$. In these constraints, Q_1 and Q_3 are quartile 1 and 3 respectively and interquartile range (IQR) is defined as $IQR = Q_3 - Q_1$. Measured quantities can have very different scales which in fact can introduce unfair advantages for some of the input features. Thus, the input data should be scaled to map all the data points to similar ranges. We have used $Z = \frac{X-u}{s}$ for scaling sensor data where X, u and s represent sensor data, mean and variance of x. These stages prepare the collected raw data for the unsupervised and supervised machine learning methods.

4.4 Unsupervised Failure Detection

The aim of this work is to make a system to detect anomalies in real time thus, a very fast process is needed. Therefore, beside the high accuracy and low memory requirement, having low processing time is one of the important factors for this work. As described previously, 13 sensors have been placed in a micro data center located at McMaster Innovation Park (MIP) to log measured current and temperature into the designed data acquisition (DAQ) system (see Figure 4-2). Around 1192 observations collected with 3 seconds intervals. During these intervals, RMS (root mean square) values of the current sensors will be recorded. This data is collected in a normal operation of data center. Figure 4-5 shows the PCA of the normal data which includes both transient (blue points) and steady state (orange points) observations. It should be mentioned that all the figures are plotted by using the " Sklearn, seaborn, and Matplotlib" packages of python. Which "Sklearn" is a library dedicated to the machine learning algorithms in python and this research uses KNN, K-means, One-Class SVM, LOF and isolation Forest libraries which are used on Raspberry pi3 for model training and failure detection.

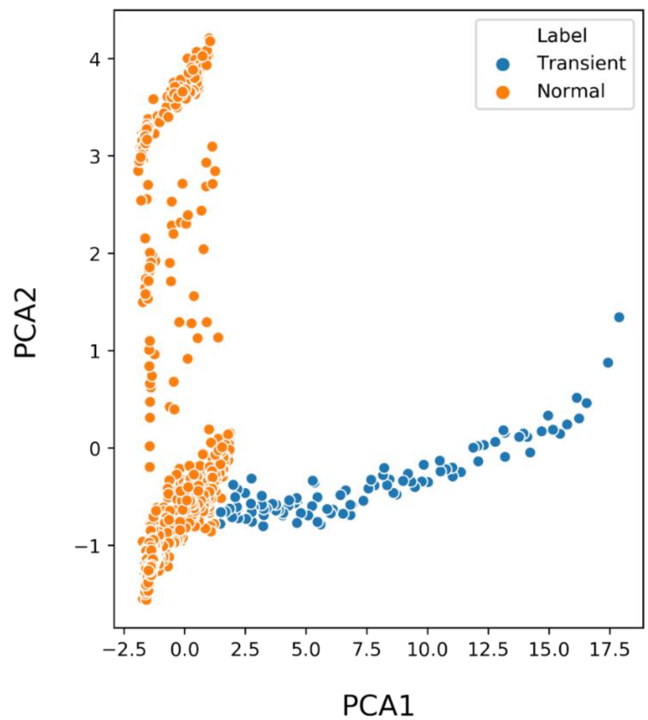

Figure 4-5: PCA of normal data includes transient (blue) and steady state (orange) data points

It is recommended to remove the transient part of the data during the training of the decision boundary for the normal operation.

There are different approaches for unsupervised machine learning where each could have different performances in different datasets and applications. Unsupervised machine learning algorithms can be classifying into two different approaches;

- Distance based methods
- classification based methods

These are discussed in section 3.4.1. In the following sections the collected data will be supplied to various unsupervised machine learning approaches and their performance are compared. It is worth to notice that despite having labels for different scenarios, these labels are not used in the training of unsupervised algorithms and will be used only to verify the performance of unsupervised algorithm:

4.4.1 K Nearest Neighbor

Unsupervised K nearest neighbor (KNN) algorithm is one of the simplest methods in detecting anomalies which is a distance based approach and tries to find the distance between any point of a dataset with its K nearest neighbors. As it mentioned earlier in section 3.4.1, KNN has three different approach which is useful in different kind of datasets; brute based, KDtree, and Ball Tree based approaches. For this specific application, since the number of training dataset is greater than 30 (this is a rule of thumb and can be different with different processing power in training phase) brute based KNN approach is not suitable for this case. This is because in the brute based method, the time complexity is calculated by $O[DN^2]$ where "N" and "D" are number of samples and dimension of dataset respectively and can be too expensive for datasets with bigger N. The results in Table 4-3 also verifies this issue where the processing time using brute based is very higher than tree based methods in any combination. Therefore, in terms of the processing time, tree based approaches will be suitable for this application.

Beside the processing time, the accuracy in detecting the anomalies should be considered in selecting the optimal solution. In order to have the best decision boundaries and find the optimal solution (high accuracy within acceptable run-time), KNN algorithm is trained with different combination of its hyper-parameters. This process is called Hyper-parameter tuning and the results are shown in Table 4-3. The hyper-parameters are number of neighbors (K) and type of the algorithm. As discussed in 3.4.1A, K will define the number of neighbors to be checked for measuring the distance and the algorithm (Ball tree and KDtree) will limit the search space in finding nearest neighbors (brute based algorithm searches the whole space). The highlighted setting has the best performance in terms of both processing time and the accuracy in finding the anomalies in the selected search space.

Table 4-3: Unsupervised KNN hyper parameter tuning

Hyper-Parameters		Results	
Number of neighbors (K)	Algorithm	Accuracy (%)	Processing time (ms)
1	Ball tree	100	35.90
1	**KDtree**	**100**	**25.95**
1	brute	100	179.5
3	Ball tree	97.31	39.89
3	KDtree	97.31	31.91
3	brute	97.31	215.4
6	Ball tree	94.88	27.92
6	KDtree	94.88	27.92
6	brute	94.88	187.4
10	Ball tree	94.29	47.87
10	KDtree	94.29	35.90
10	brute	94.29	207.4

As it is mentioned, the trained model will be used for classifying a new unseen data point (test data points) into the normal versus faulty clusters. For a new data point x_i, its average distance to its K nearest neighbors could be viewed as the outlier score ($f(x_i)$). Where if the data point on average is far away from the K neighbors, the data point is more likely an outlier (anomaly). The average distance between the observation and its K nearest neighbors is defined as anomaly scores in this application. In this thesis, the health of the system is defined by mapping these distances to values between 0 and 1, where 1 shows the healthy situation and 0 shows the unhealthy situation. The following steps are used to map the outlier score ($f(x_i)$) to health index defined in this thesis (Eq. (4-1)),

1. reversing the dataset ($-f(x_i)$)
2. shifting it to non-negative values ($-f(x_i) + |\max(f(x_i))|$)
3. scaling it to values between 0 and 1 ($\frac{-f(x_i)+|\max(f(x_i))|}{max(f(x_i))+|\min(f(x_i))|}$)

Eq. (4-1) summarizes these steps to introduce the health index used in this research.

$$H(x_i) = \frac{-f(x_i) + |\max(f(x_i))|}{max(f(x_i)) + |\min(f(x_i))|} \qquad (4-1)$$

Figure 4-6 shows the health of system with its K=1 neighbors while using KDtree method. BallTree method and different K values resulted in very similar graphs so we have shown the distance figure with only one setting (K=1 neighbors and KDtree method). As Figure 4-6 shows, micro data center operates under normal scenarios for 3500 seconds and then a specific failure happens. Right after the failure, the health index is going to drop below 0.8 very quickly and it will stay below this threshold for much longer that 30 seconds. Thus, all of the failure scenarios will be detected with the selected thresholds ($H_{threshold} = 0.8$ and $Time_{threshold} = 30\ seconds$). The title of each plot in Figure 4-6 indicates the failure type and if it is a single failure or combinations of two failures (for safety reasons, we limited the number of failure scenarios to the ones shown in Figure 4-6).

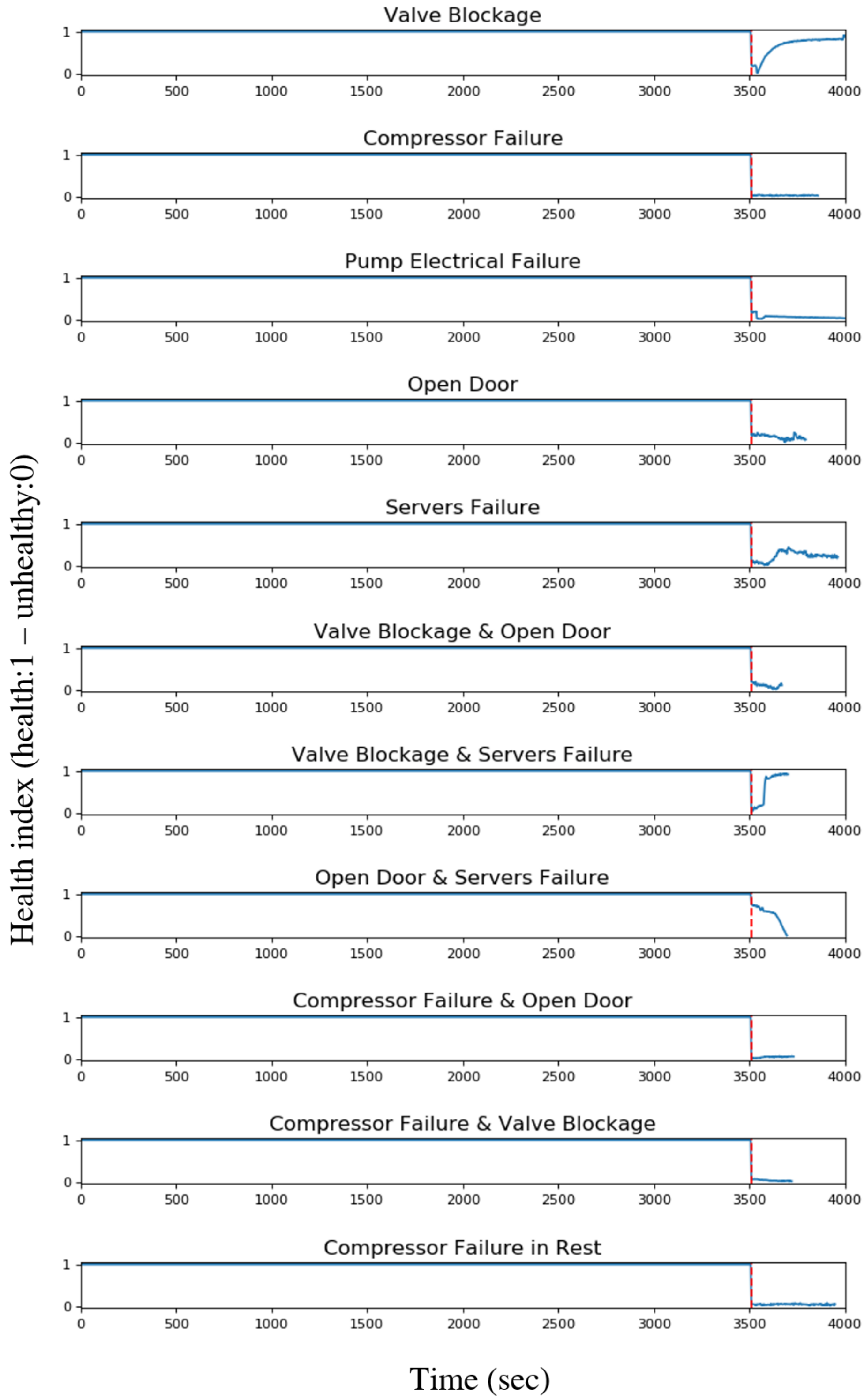

Figure 4-6: Health Index of micro data center using unsupervised KNN

4.4.2 K-means

K-means is an unsupervised clustering based machine learning. In the training step, this technique tries to partition a training dataset into K clusters where K is user defined constant. Because we have used only normal data points to train the machine learning algorithm, so K should be set to 1 (K=1). In the following figure blue points are the training data points which are collected during normal operation of micro data center and the red point is the centroid of the cluster.

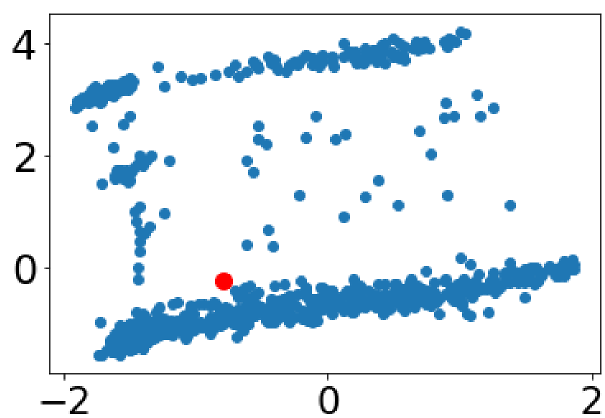

Figure 4-7: Normal cluster data points and centroid

After training the model using K=1, K-means can detect abnormalities based on their distance from the cluster centroid by looking for any values beyond certain thresholds. Setting this threshold similar to the previous unsupervised methods depends on the application and its dataset. In this application and with looking at health index curve in Figure 4-8, the threshold values have been set to ($H_{threshold} = 0.8$ and $Time_{threshold} = 30\ seconds$). These threshold values would lead to the accuracy of 92.8 % and runtime of 544 milliseconds.

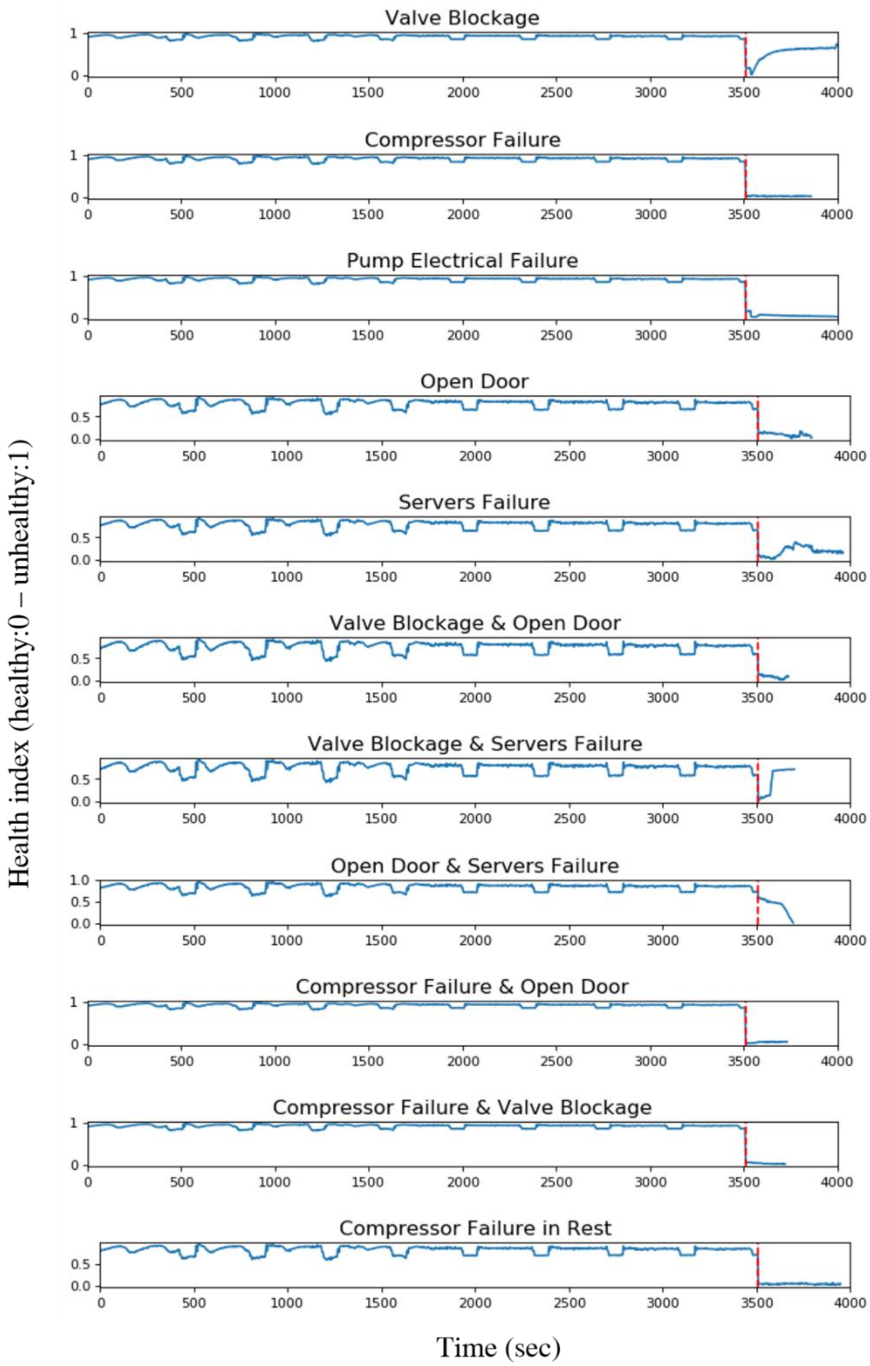

Figure 4-8: K-means clustering health index for failure scenarios

Figure 4-8 shows the health index which is defined by mapping the distance function to numbers between 0 and 1. This process is explained in deriving (Eq. (4-1)) in previous section.

4.4.3 One Class Support Vector Machine

One class support vector machine (one-class SVM) is a classification based unsupervised machine learning approach (section 3.4.1B). SVM tries to find the best decision boundaries that separate normal versus abnormal data points. Hyper-parameters are variables that should be set before training the model. In order to create the best decision boundaries and find the optimal solution (high accuracy within acceptable run-time), unsupervised algorithm is trained with different combination of these hyper-parameters. This process is called Hyper-parameter tuning and the results are shown in Table 4-4. The highlighted setting has the best performance in the selected search space.

Table 4-4: One class SVM hyper parameter tuning

Hyper-Parameters			Results	
Kernel type	Kernel coefficient	Degree	Accuracy (%)	Processing time (ms)
RBF	**0.1**	**-**	**98.74**	**31.23**
RBF	0.8	-	98.74	33.26
Polly	0.1	1	14.93	4.68
Polly	0.1	4	1.25	6.56
Polly	0.5	1	14.93	6.16
Polly	0.8	1	1.25	6.23

One of the important parameters is defining the kernel type and since the collected dataset is not linearly separable, a nonlinear kernel (such as polynomial or Gaussian RBF kernels) would perform better than linear kernels. Kernel coefficient (v in equation 3-13) will define the trade-off between overfitting and under-fitting. So, increasing v causes under fitting while decreasing it causes overfitting the model.

$$\begin{cases} small\ v \ \rightarrow overfitting \\ large\ v \ \rightarrow under\ fitting \end{cases}$$

Training with optimal hyper parameters leads to the decision boundary shown Figure 4-9. This boundary surrounds the normal dataset in lower dimensions (projected with principle component analysis) for the visualization purposes.

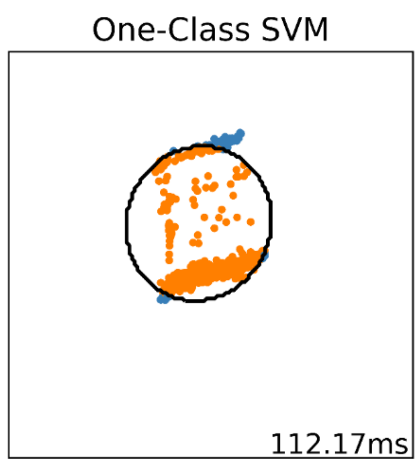

Figure 4-9: Decision boundary for normal operation using one class SVM

In Figure 4-10: Location of faulty data with respect to normal operation decision boundary shows the position of test dataset relative to the trained decision boundary. Since none of the failures are inside the decision boundary, accuracy of one-class SVM will be high in this application.

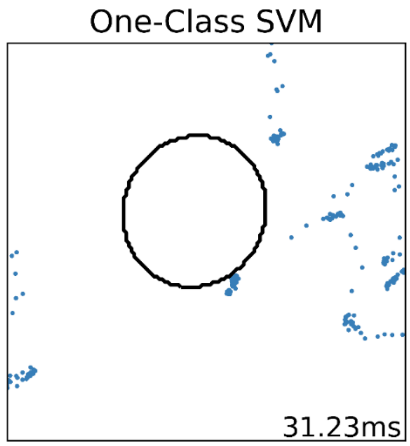

Figure 4-10: Location of faulty data with respect to normal operation decision boundary

As it is mentioned in Chapter 3, one-class SVM can identify the abnormalities by using decision function for calculating the anomaly score for any new data point. The calculated scores are the distances of any new data point to the separating hyperplane learned by the model. The created method uses these scores by interpreting positive and negative distances as normal and faulty data points respectively. In this study, equation (4-2) is used to map the anomaly scores to numbers between 0 and 1 where 1 shows the healthy situation while 0 shows the unhealthy operation of the micro data center. The following steps are used to map the outlier score ($f(x_i)$) to health index defined in this thesis (Eq. (4-2)),

1. reversing the dataset ($f(x_i)$)
2. shifting it to non-negative values ($f(x_i) + |\min(f(x_i))|$)
3. scaling it to values between 0 and 1 ($\frac{f(x_i)+|\min(f(x_i))|}{max(f(x_i))+|\min(f(x_i))|}$)

Eq. (4-2) summarizes these steps to introduce the health index used in this research.

$$H(x_i) = \frac{f(x_i) + \left|\min\left(f(x_i)\right)\right|}{max\left(f(x_i)\right) + \left|\min\left(f(x_i)\right)\right|} \qquad (4-2)$$

As Figure 4-11 shows, by selecting detection threshold of $H_{threshold} = 0.8$, most of the failures can be detected within few seconds. For example, after pump fails at t=3500 seconds, health index drops to almost zero in less than one second. Model behaves similarly in detecting all failures by selecting $H_{threshold} = 0.8$ and $Time_{threshold} = 30\ seconds$. The title of each plot in Figure 4-6 indicates the failure type and if it is a single failure or combinations of two failures (for safety reasons, we limited the number of failure scenarios to the ones shown in Figure 4-11).

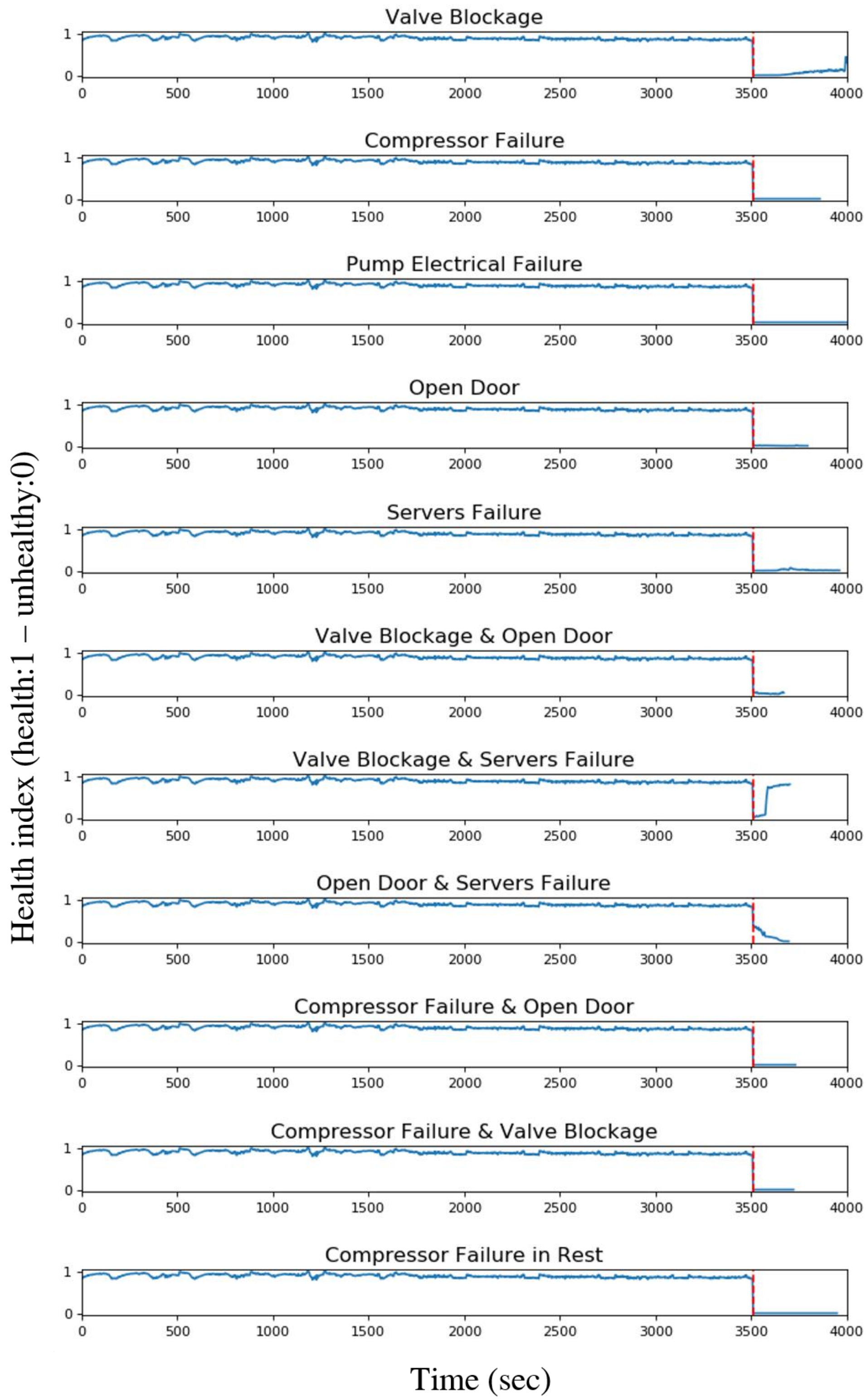

Figure 4-11: decision function results for one-class SVM

4.4.4 Isolation Forest

Isolation Forest is another powerful classification based approach which is discussed in section 3.4.1B. Figure 4-12 shows the boundary drawn by isolation forest for the normal operation. The blue points are identified as anomalies while they are in fact normal data points. Isolation forest always classifies the non-dense part of the data as abnormalities since these are the close leaves to the root node during tree creation process.

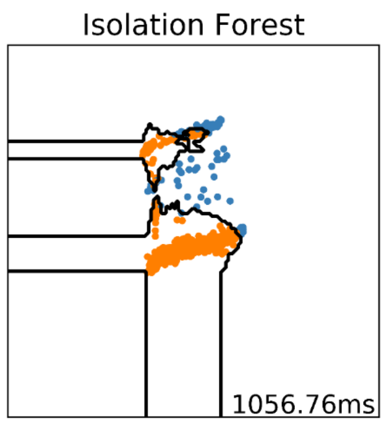

Figure 4-12: Decision boundaries for normal operation with unsupervised Isolation Forest)

After the model is trained, the anomaly score can be calculated for any new unseen data point. Isolation forest can calculate these scores in a way which is discussed in section 3.4.1C and will return the values in the range of [-1, 1] where negative and positive numbers correspond to abnormal and normal states of the observations. The measure of normality of an observation is the depth of the leaf containing this observation, which is equivalent to the number of splits required to isolate this point. Figure 4-13 shows the predicted clusters (blue: anomalies and orange: normal) after applying the test dataset (faulty dataset) to the trained model. Where again there is some miss-classification (orange points). Because of the over fitted decision boundary shown in Figure 4-12, this algorithm will not be used for obtaining the health of system.

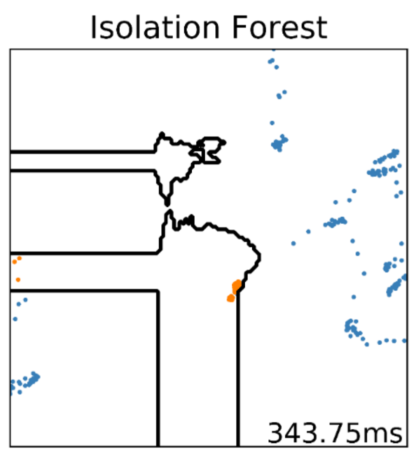

Figure 4-13: Location of faulty data with respect to normal operation decision boundary

4.4.5 Local Outlier Factor

Local outlier factor (LOF) is a density based machine learning algorithm where the core concept of this method is the application of reachability distance in detecting outliers (discussed in section 3.4.1). The idea here is using the reachability distance for each data point to calculate the local reachability density (LRD). Using LRD measurements, LOF(x) can be computed for each data point. As mentioned earlier, LOF computes the local density deviation of a given data point with respect to its neighbors and as a result if portion of a dataset has lower density than their neighbors, they will be flagged as anomalous operating points. Figure 4-14 shows the results of LOF on both training (Figure 4-14 (a)) and the test datasets (Figure 4-14 (a)). In this figure, orange points are inliers (normal) and blue points are outliers (abnormal). What can be concluded is that in both steps LOF algorithm does not perform well which could be because of low density of the data in parts of the train and test datasets. Because of aforementioned problem, this algorithm will not be used for obtaining the health of system in this application.

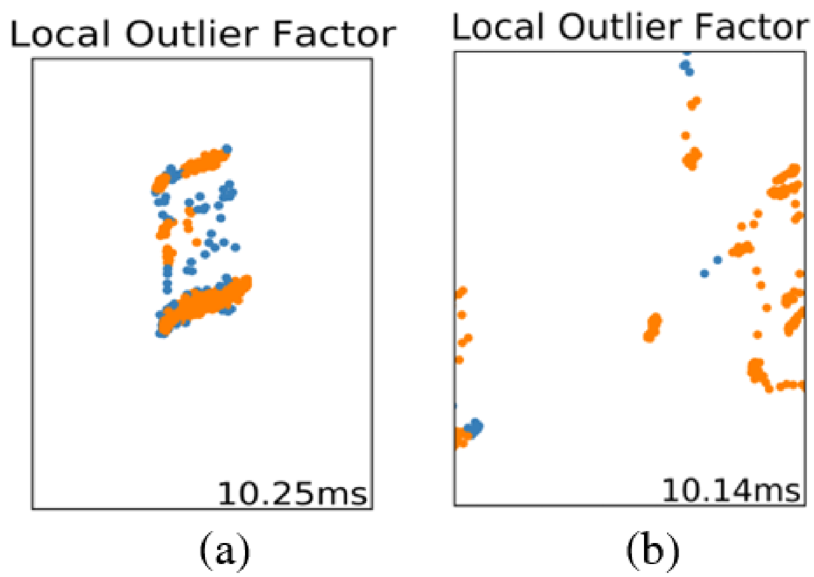

Figure 4-14: a) LOF for normal training dataset with 10 percent contamination, b) applying the faulty test dataset to the trained model

4.4.6 Summary

The anomalies predicted by the two different groups of machine learning approaches, distance based, and classification based are resulted in different performances.

- Because of having low density in parts of the normal data, isolation Forest and LOF consider these normal but low density portions as anomalies and lower down the overall performance drastically.
- KNN, K-means and one-class SVM show good performance in describing normal and abnormal distributions.

Table 4-5 compares the accuracy and processing time of the above mentioned unsupervised algorithms.

Table 4-5: Accuracy and processing times of unsupervised algorithms

	KNN	**Isolation Forest**	**LOF**	**K-means**	**One-class SVM**
Accuracy (%)	100	failed	failed	92.8	98.74
Processing time (ms)	25.95	failed	failed	544	31.23

As it is mentioned earlier, the current data centers are using only temperature sensors to detect and predict the anomalies. In order to have fast and accurate failure detection, in this research we added 4 current sensors which will give new and different angles to machine learning methods. Machine learning algorithms have better performance with more independent features. Since adding current sensors have low correlations with temperature sensors, they can provide independent information for predictive modes to learn from.

4.5 Supervised Failure Localization

As it is explained in the previous section, unsupervised techniques are used to monitor the health of micro data center, and any drop in the health status could be the sign of an abnormality in the system that should be checked by the operator. Operator will label the faulty observation with the failure type anytime that unsupervised algorithm shows a significant drop in health index and this labelled data will be used to localize the failures in the future (see Figure 4-1 (b)). Having enough labels (logged by the operator) for any type of failure gives the opportunity to the supervised learning methods for classifying new observations into the right group and localizing the failures. In this research labeled data is generated by creating 11 different types of failures in the micro data center. These failure scenarios are given in Table 4-1 (SC2, SC3, ..., SC12 and SC1 for normal operation). Table 4-1 also shows the duration of each scenario and the number of collected samples for each failure scenario which in total 1192 faulty observations have been collected with 3 sec interval recording. Thus, the dataset contains 13 features (number of placed sensors) and 2391 observations (normal data points + faulty data points). Figure 4-15 illustrates the class distribution by assuming all the faulty data points belong to the failure class and a detailed description of classes are shown in Figure 4-16. It should be mentioned that all the figures are plotted by using the " Sklearn, seaborn, and Matplotlib" packages of python. Which "Sklearn" is a library dedicated to the machine learning algorithms in python and this research uses SVM, KNN, Random Forest libraries which are used on Raspberry pi3 for model training and failure detection.

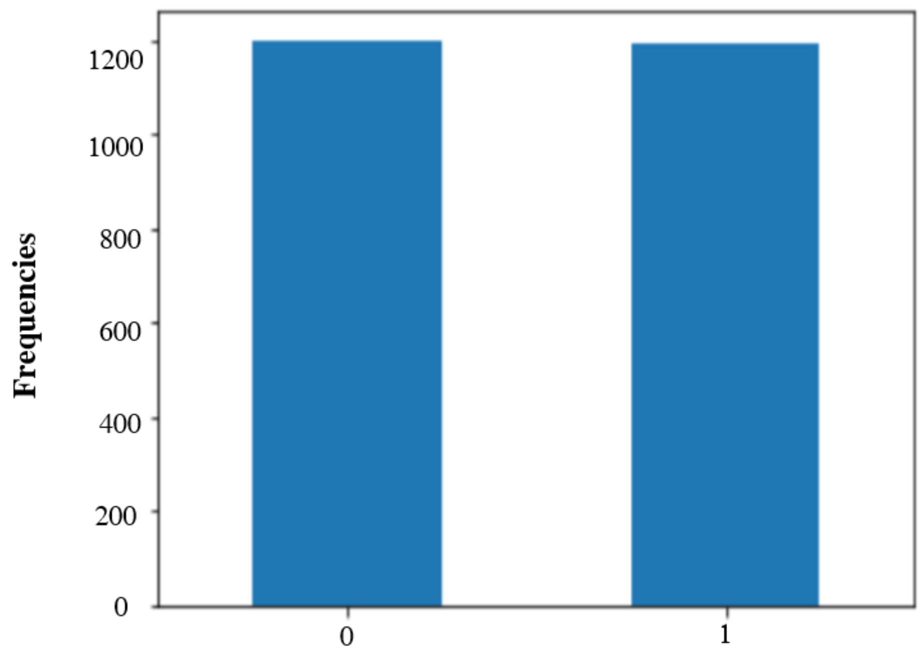

Figure 4-15: class distribution of collected dataset (0: normal, 1: faulty)

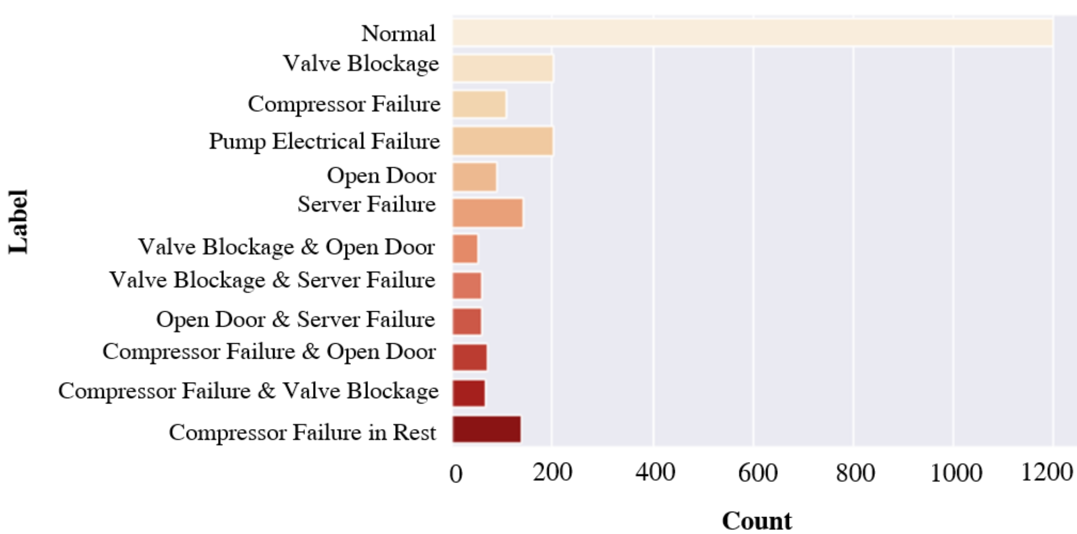

Figure 4-16: A detailed class distribution of collected dataset

In the following sections, collected data first is visualized and then is prepared for training supervised methods for the failure localization. In order to visualize the collected dataset which is originally in 13-dimension, the principal component analysis (PCA) with two principal components is used to show dataset in 2-dimension space. Figure 4-17 shows the distribution of the different scenarios in a 2D space. This figure shows that while some of the classes like Scenario 4 (SC4) which does not have overlaps with other classes is easy to identify, some of the other classes (like SC10 and SC11) are very close to each other and a full separation of them is not possible.

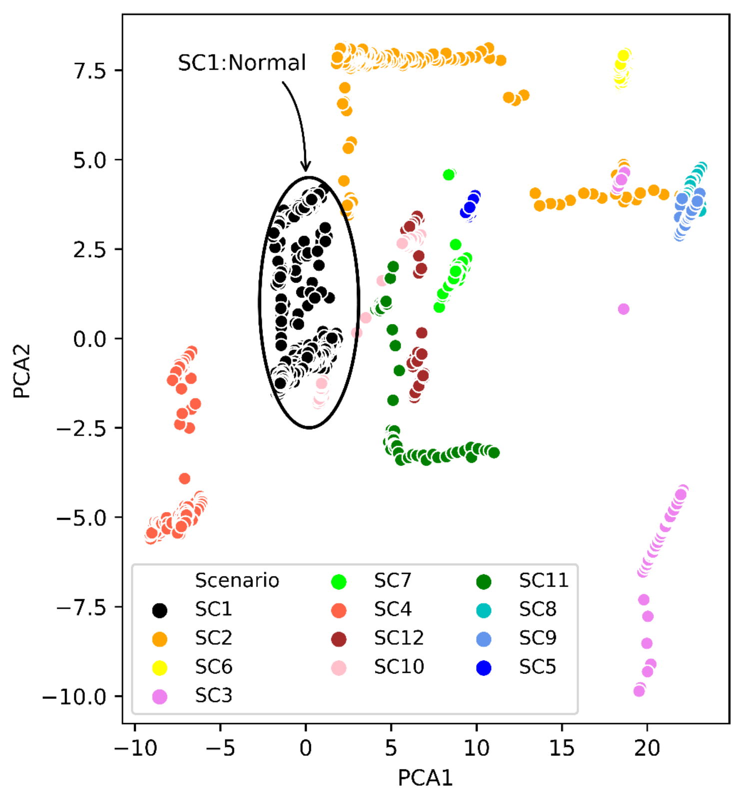

Figure 4-17: PCA of data (normal and faulty)

It is also interesting to know that if there are any significant correlations between the predictors. A heatmap is a two-dimensional graphical representation of data where the individual values that are contained in a matrix are represented as colors. Figure 4-18 shows the correlation between features in collected dataset. In this figure, blue means positive correlation and red means negative correlation and the stronger the color is, the larger the correlation magnitude is. The heatmap shows,

- Strong positive correlation between the temperature of front-upper side of the rack, with middle-lower side of it.
- Strong positive correlation between the temperature of back-upper side of the rack, with back-lower side of it.
- Strong negative correlation between Rack front down, condenser and ambient temperatures.
- Features with high correlations don't have new and independent information for machine learning algorithms and in many cases, it is recommended to replace the correlated features (multiple features) with their average (single feature). This will also help to reduce the dimension of the original dataset.

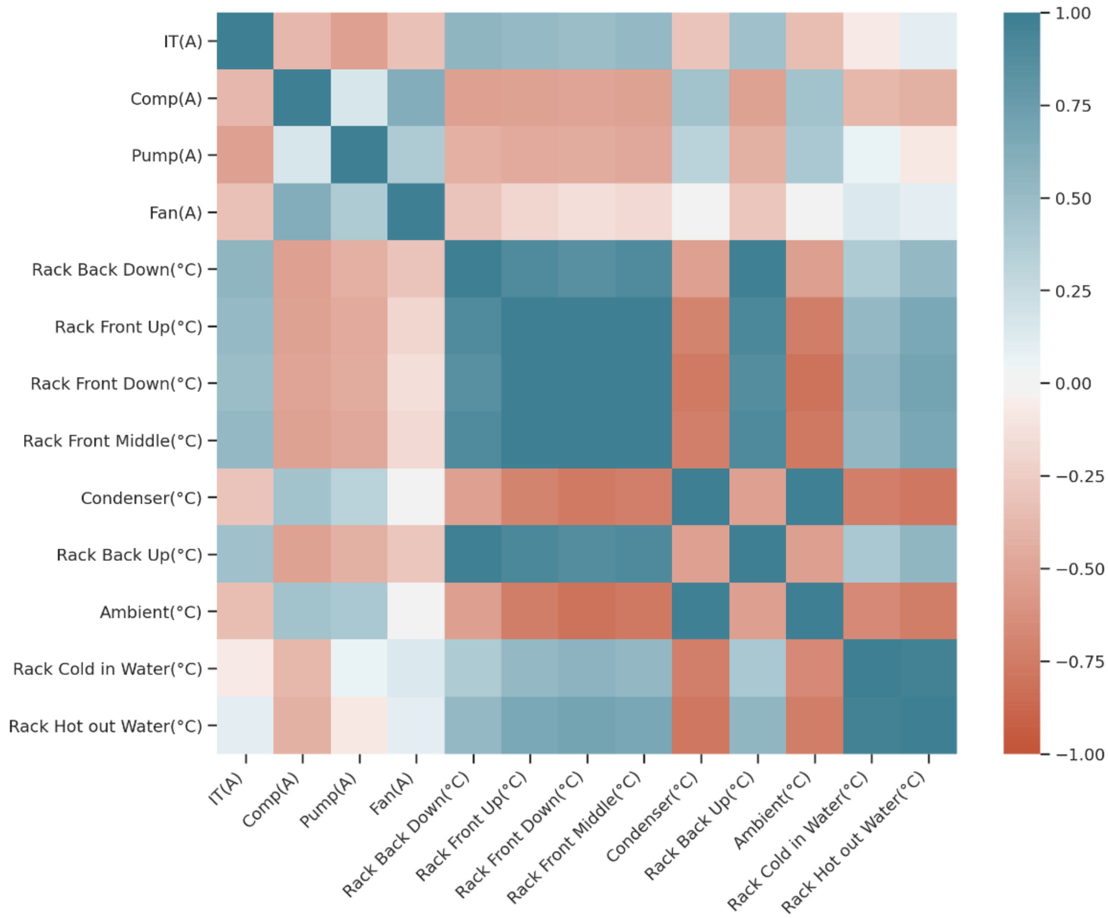

Figure 4-18: Correlation matrix heatmap for the dataset

Looking at the heatmap plot, the added 4 current sensors have low correlation with temperature sensors. Since machine learning methods have better performance with independent features, therefore, the newly added current sensors will improve the performance in failure localization.

Another good way to visualize to understand the dataset is looking at the probability density of each sensor respect to the normal and failure operations. The below subplots show the distribution of each feature (sensor readings) in normal and failure operations. Table 4-6 shows the sensor averages for both normal and failure scenarios.

Table 4-6: Sensor averages for both normal and failure data states

Sensor	Average value for failure	Average value for normal
Input power to IT Rack (A)	188.47	179.07
Input power to Compressor (A)	61.03	168.13
Input power to Pump (A)	38.20	45.04
Input power to Condenser Fans (A)	36.05	47.00
IT Rack Temperature (Back, Down) (°C)	35.73	30.53
IT Rack Temperature (Front, Up) (°C)	26.99	20.71
IT Rack Temperature (Front, Down) (°C)	27.42	19.81
IT Rack Temperature (Front, Middle) (°C)	27.42	20.41
Condenser Temperature (°C)	25.65	26.70
IT Rack Temperature (Back, Up) (°C)	32.61	27.02
Ambient Temperature (°C)	25.95	26.98
Cold Water Temperature (into IT rack) (°C)	14.72	11.92
Hot Water Temperature (out of IT rack) (°C)	16.56	13.38

This table compares the average values for all of the placed sensors in both normal and failure states. As this figure shows, the probability distribution for not all of the sensors are easily separable (their mean for normal and failure states are not far from each other in the graph). For example, while the average current drawn by compressor (Figure 4-19 (d)) is 168.13 and 61.03 for normal and failure scenarios respectively, for other sensors the mean values are closed to each other and a wide variance creates a considerable overlap between normal and failure states. This is main reason why sensors cannot detect all failure scenarios on their own (individually) and machine learning algorithms should be trained for having a holistic view of the micro data center.

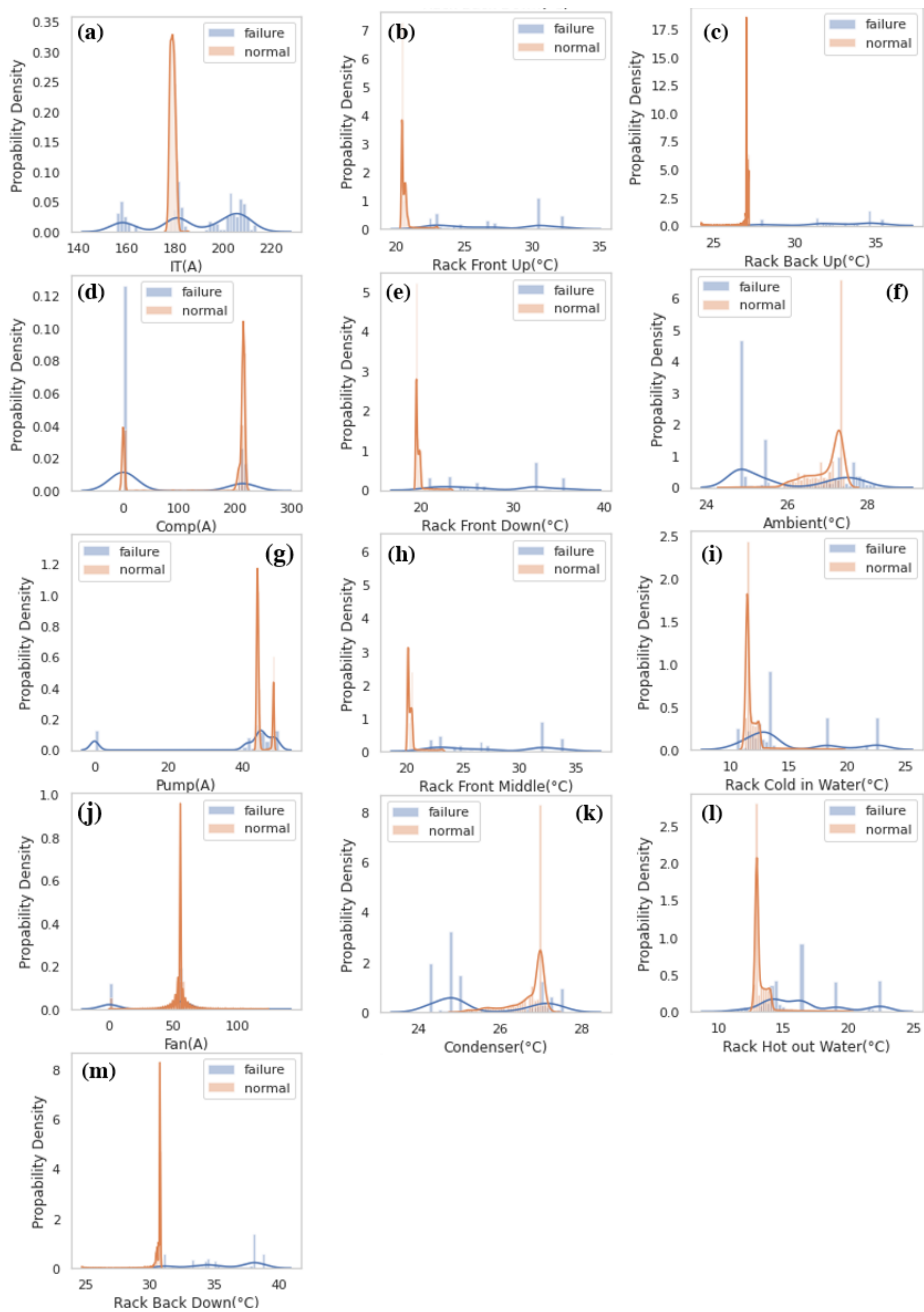

Figure 4-19: Distribution of sensors in normal and failure scenarios

The target of using supervised learning methods is creating a line or a hyperplane (linear or non-linear boundaries) which separates the data into distinct classes. Drawing better decision boundary will lead to better classifying of the new unseen. In this case there are

12 different groups including one normal and 11 different failures. Like unsupervised learning methods, there are many algorithms which can be used in solving the supervised problems which are discussed in section 3.4.2. To determine which one works well in this application, SVM, random forest, and K-nearest neighbors are applied to the collected data. Around 70 percent of the collected data is used for training the model and the rest for testing and validating the created model. Percentages of correctly classified scenarios are shown in the confusion matrix of Figure 4-20 where, equation (4-3) is used as the measure of accuracy:

$$Accuracy\ of\ class\ C = \frac{\#\ correctly\ classified\ samples\ in\ class\ C}{\#\ all\ samples\ in\ class\ C} \times 100\% \quad (4-3)$$

Since number of samples in anomalous and normal datasets are unbalanced in this application, general accuracy measures can not be used directly. The following equations are recommended ways of measuring accuracies in unbalanced datasets,

- Sensitivity or recall which gives true positive rate (Powers, 2011):

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4-4)$$

Where TP is true positives (truly detected anomalies), FN is false negative (anomaly points which are mislabeled as normal points).

- Precision which gives the probability of predicting a True Positive from all positive predictions (Powers, 2011):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4-5)$$

Where FP is false positive (normal operating points which are mislabeled as abnormal points).

- F1 score which gives the harmonic mean of precision and sensitivity (Powers, 2011):

$$\text{F1} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \quad (4-6)$$

4.5.1 Support Vector Machine

As discussed in section 3.4.2B, support vector machine classifies different classes by maximizing the gap between the observations in each class. The following grid, is the hyper parameter space to be searched by `GridSearchCV` function of `Sklearn` package in python:

```
Regularization parameter 'C': [1,10,100,1000],
Kernel coefficient 'gamma': [1,0.1,0.001,0.0001],
kernel type: ['linear','rbf']
```

The optimal hyper parameters obtained by `GridSearchCV` will be,

```
Optimal regularization parameter 'C' = 1000,
Optimal kernel coefficient 'gamma'= 1,
```

Optimal kernel type: `'rbf'`

Using the tuned hyper parameters above, the final SVC classifier is trained and the following confusion matrix shows the result for this optimal configuration,

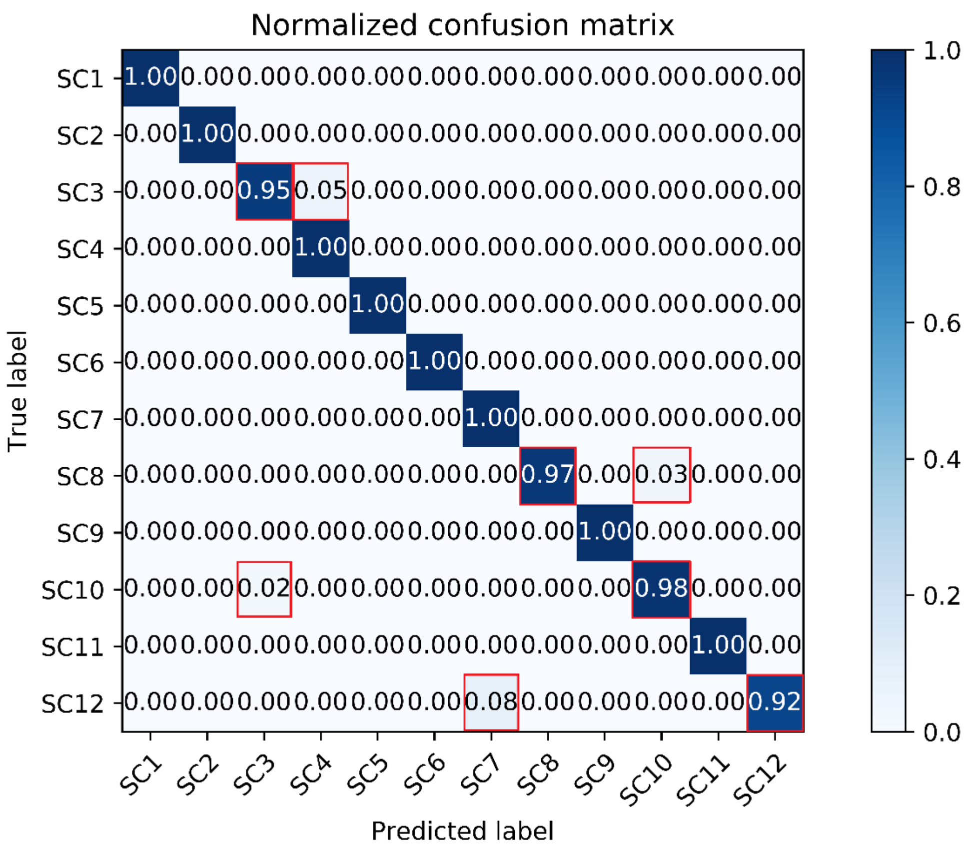

Figure 4-20: Normalized Confusion matrix using SVM classifier

The confusion matrix shows that supervised SVC is able to localize different fault scenarios with high accuracy. For example the errors for true label (vertical axis) and predicted labels (horizontal axis) are zero for many scenarios in Figure 4-20. Few miss-classifications shown in this matrix (such as SC8 instead of SC10) could be because of the transient behaviors during the failures. After this transient state, the operating point usually settles close to the center of a specific failure cluster. The duration of these transient behaviors depends on the thermal inertial and the severity of electrical faults. For example, short circuit faults will have different signature and duration than electrical overload faults or large data center will have bigger thermal capacity. The effect of these transient behavior can be studied for different size and configuration of micro data centers in the future studies.

4.5.2 Random forest

As discussed in section 3.4.2B, random forest creates multiple trees to classify different observations (by selecting random set of features and observations and splitting them to reach pure leaf nodes). The following grid, is the hyper parameter space to be searched by `GridSearchCV` function of `Sklearn` package in python:

```
Number of estimators: [200, 500],

Max features: ['auto', 'sqrt', 'log2'],

Max depth: [4, 5, 6, 7, 8],

Criterion: ['gini', 'entropy']
```

The optimal hyper parameters obtained by `GridSearchCV` will be,

```
Number of estimators= 500,

Max features= 'auto',

Max depth= 8,
```

```
Criterion= 'gini'
```

Using the tuned hyper parameters above, the final random forest classifier is trained and the following confusion matrix shows the result for this optimal configuration,

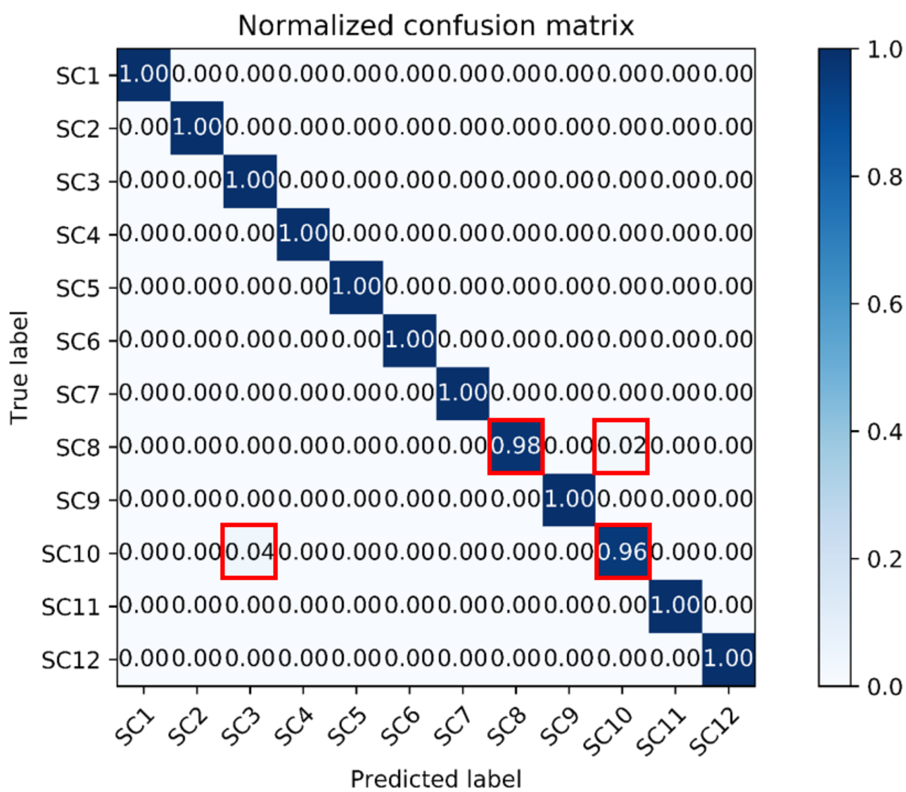

Figure 4-21: Normalized Confusion matrix using random forest classifier

The confusion matrix shows that random forest can localize different fault scenarios with high accuracy. For example the errors for true label (vertical axis) and predicted labels (horizontal axis) are zero for many scenarios in Figure 4-21. Few miss-classifications shown in this matrix (such as SC8 instead of SC10) could be because of the transient behaviors during the failures and the operating point usually settles close to the center of a specific failure cluster in steady state operation.

4.5.3 K nearest neighbor (KNN)

As discussed in section 3.4.2B, supervised KNN obtains the labels of K nearest neighbors for a typical observation A and assigns the most frequently observed labels to the point A. The following grid, is the hyper parameter space to be searched by `GridSearchCV` function of `Sklearn` package in python:

```
Number of neighbors: range (1,50),
Type of algorithm: ['ball_tree', 'kd_tree', 'brute'],
Max depth: [10, 20, 30 40, 50]
```

The optimal hyper parameters obtained by `GridSearchCV` will be,

```
Number of neighbors= 1,
Type of algorithm= 'ball_tree',
Max depth= 10
```

Using the tuned hyper parameters above, the final KNN classifier is trained and the following confusion matrix shows the result for this optimal configuration,

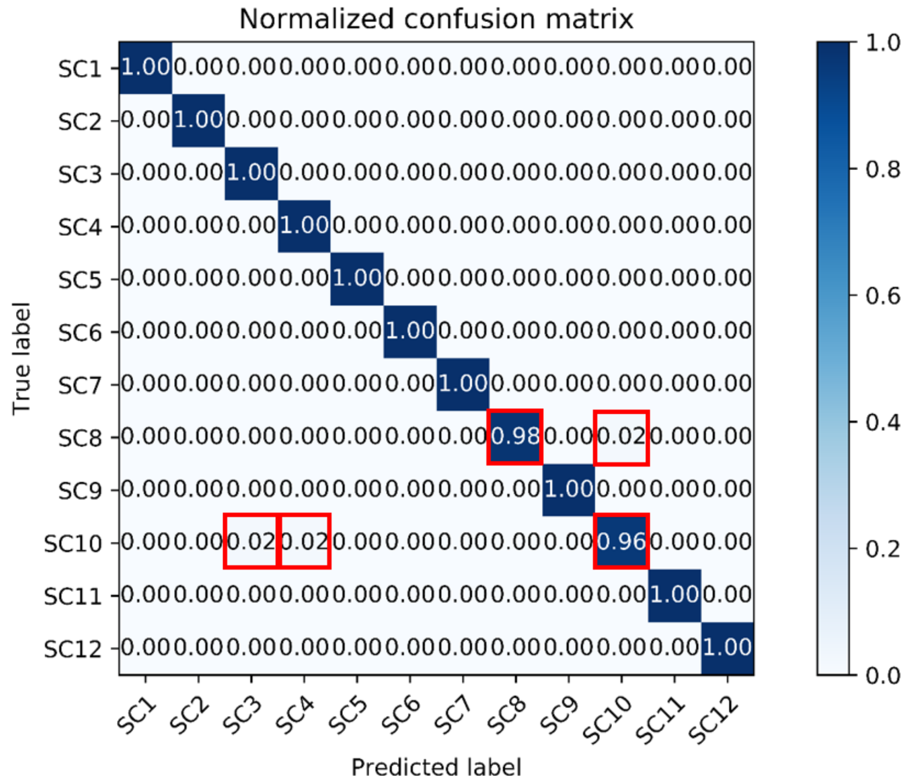

Figure 4-22: Normalized Confusion matrix using KNN classifier

The confusion matrix shows that KNN has high performance in localizing different failures. Miss-classifications shown in this matrix (such as SC8 instead of SC10) could be because of the transient behaviors during the failures.

4.5.4 Feature selection

Many factors can affect the success of machine learning on a given task. Theoretically, having more features should result in more discriminating power, however, in lower observations this could lead to overfitting of trained model. Another problem with having lots of features is to increase hardware requirements to store and process the data specially in online applications. Developed data acquisition system uses a raspberry pi module as storage and processing unit, thus selecting most informative features (sensors) could be important in practical applications. Feature selection is discussed in section 3.3.1 and Figure 4-23 shows the importance of features (sensors) used in this research individually using recursive feature elimination (RFE) method. This analysis helps to identify important sensors (to maintain them properly) and readjust the number of features to be used in the training of predictive models.

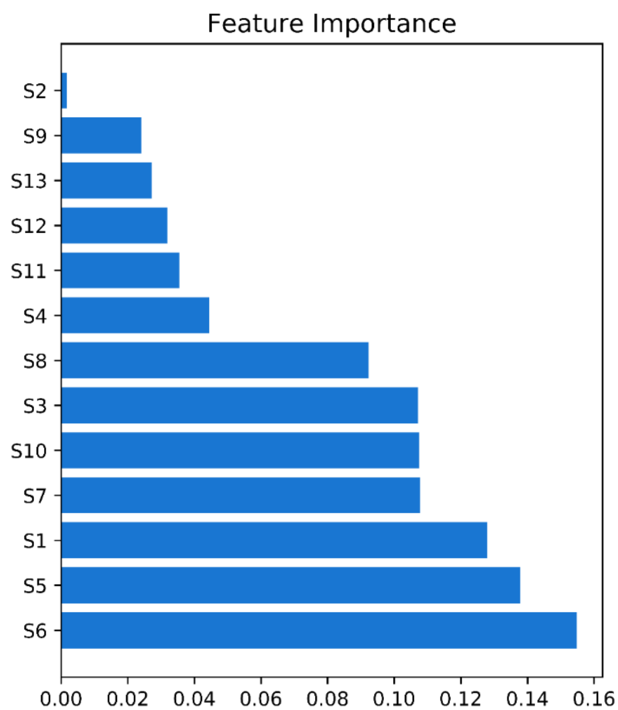

Figure 4-23: Sensor importance in supervised learning

After finding the most important features, other features can be removed without significantly loosing information and hurting the performance of the clustering algorithm. As Figure 4-252 shows the accuracy would remain around 97 percent even when we use only 8 important sensors (features).

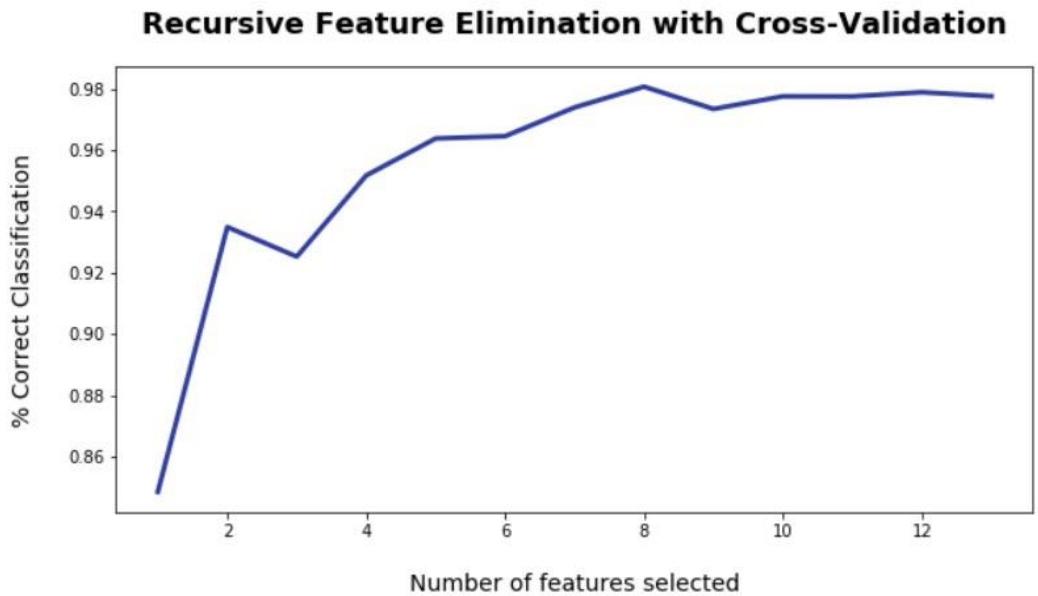

Figure 4-24: Accuracy calculation when using only important features in training the supervised machine learning algorithms (SVC)Summary

4.5.5 Summary

The application of unsupervised learning helps creating labels for different failure scenarios and thus the type and location of future failures can be classified with supervised learning methods. The performance of three supervised algorithms have been analyzed and the results of their F1 scores are given in Figure 4-25. High F1 scores in many scenarios shows the good separation between the failure scenarios and between the trained models, supervised SVC performs better than other predictive models. Also, the most important features have been studied and the result showed that using 8 sensors out of 13 existed ones can reduce the computational complexity and at the same time increase the accuracy in the failure localization.

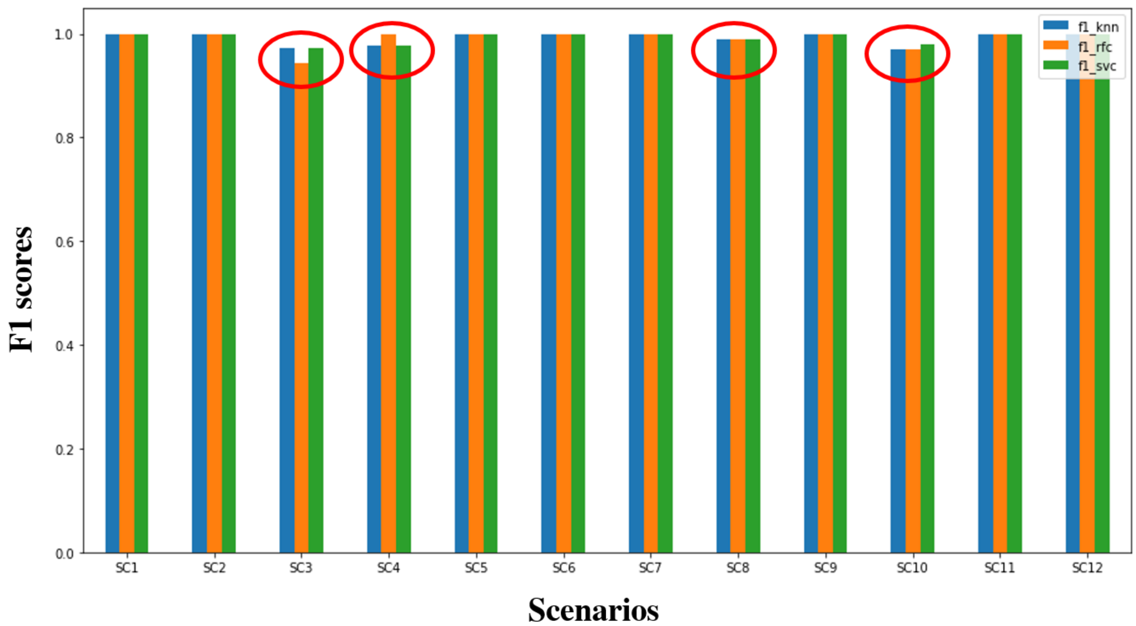

Figure 4-25: Comparison between f1 scores for different classes using three different supervised classifier (blue: KNN, orange: random forest and green: support vector machine classifier)

Chapter 5

5 Conclusion and Future Work

Data center outages are very expensive and could even impose security risks to operators in such facilities. Predictive maintenance is a mathematical approach to study the historical operations to decrease the unplanned outages, associated costs and security hazards. The current hardware and software (containing predictive algorithms) are mainly covering large datacenter applications and are not suitable for micro data centers. This study proposes a new approach to real-time monitoring and predictive maintenance of micro data center infrastructure by developing affordable and non-invasive hardware together with quickly trainable predictive models. This research accomplished:

- **Failure detection and localization in MDCs**: A new approach has developed to detect and localize the failures in MDCs quickly. Numerical results show the effectiveness of proposed method in the micro data center placed at McMaster Innovation Park (MIP):
    - ***Failure Detection***: Since collecting labelled data is usually expensive especially in the industrial world, unsupervised machine learning methods are used to learn the signature of the normal operation for micro data center. The developed machine learning based failure detection model could identify abnormalities in a few seconds. In the first step, holistic health status of system will be defined where any drop in the health status should be considered as a sign of a failure/abnormal situation in the micro data center. This is verified by creating 6 single failures and also 5 combined failures (totaling to 11 failure scenarios) where the trained model could detect all the failures very quickly.
    - ***Failure Localization***: Any consistent drop in health index (below 0.8 for more than 30 seconds in this research) will notify the operator by triggering an alarm. Operator will clear the event by logging the actual label of the event (false or true positive) in the database which these labelled data will be used to localize the failures by using supervised machine learning algorithms. The numerical results given in chapter 5 show the skills of the trained model in classifying the failures into the right groups with 98% accuracy.

- **Hardware design:** Also a data acquisition system (DAQ) is developed to collect electro-mechanical signals from micro data center (discussed in chapter 4). The developed data acquisition system is affordable (the cost was under 100 CAD in 2019) and easy to implement in micro data centers that usually don't have permanent access to technicians. This system collects data from the micro data center in a non-invasive approach, thus there is no need to create costly downtimes for implementing smart monitoring solution developed in this research.

This research could be continued and expanded in the following directions,

- Sensor failures are common specially in harsh environment and thus the bias in their readings can effect the performance of predictive models. While there are good studies in analysis sensor anomalies, their specific behaviour in micro datacenters could be studied in the future works.
- Micro data centers usually have high dynamics because of their physical size and capacity. High dynamics could lead to considerable transient behaviour between different operating scenarios and could be studied in the future researches.

Publication:

Conference Paper:

N. Piran Nanekaran, M. Esmalifalak and M. Narimani (2020), *Fast Anomaly Detection in Micro Data Centers Using Machine Learning Techniques*, IEEE Intonation Conference on Industrial Informatics (INDIN), 2020.

Bibliography

ABB. (2013). Datacenters. *The corporate technical journal*, 1–84.

Ahmed, N. D. (2012). Condition monitoring in the management of maintenance in a large scale precision CNC machining manufacturing facility. *IEEE In 2012 IEEE International Conference on Condition Monitoring and Diagnosis*, 842-845.

Ahuja, N. R. (2013). Real time monitoring and availability of server airflow for efficient data center cooling. *In 29th IEEE Semiconductor Thermal Measurement and Management Symposium*, 243-247.

AlLee, G. &. (2012). Edison redux: 380 Vdc brings reliability and efficiency to sustainable data centers. . *IEEE Power and Energy Magazine, 10(6),* , 50-59.

Andrae, A. S. (2015). On global electricity usage of communication technology: trends to 2030. pp. 117-157.

Appana, D. K. (2017). Reliable fault diagnosis of bearings using distance and density similarity on an enhanced k-NN. . *In Australasian Conference on Artificial Life and Computational Intelligence*, 193-203.

Atzori, L. I. (2010). The internet of things: A survey. . *Computer networks, 54(15),*, 2787-2805.

Avelar, V. (2010). *Guidance for Calculation of Efficiency (PUE) in Data Centers. .* Schneider Electric, Rueil Malmaison, France, White Paper, 158.

Azadeh, A. &. (2016). An integrated fuzzy analytic hierarchy process and fuzzy multiple-criteria decision-making simulation approach for maintenance policy selection. *Simulation, 92(1),* 3-18.

BABIKER, M. K. (2019). A hybrid feature-selection approach for finding the digital evidence of web application attacks. . *Turkish Journal of Electrical Engineering & Computer Sciences, 27(6),*, 4102-4117.

Bao, S. Z. (2017). Data-based line trip fault prediction in power systems using LSTM networks and SVM. *IEEE Access 6*, 7675-7686.

Barroso, L. A. (2009). The datacenter as a computer: An introduction to the design of warehouse-scale machines. . *Synthesis lectures on computer architecture, 4(1),* , 1-108.

Bash, C. B. (2006). Dynamic thermal management of air cooled data centers. *In Thermal and Thermomechanical Proceedings 10th Intersociety Conference on Phenomena in Electronics Systems. ITHERM.*

Blackburn, M. (2010). *The green grid data center compute efficiency metric: DCcE.* . The Green Grid, White Paper-34.

Borgi, T. H. (2017). Data analytics for predictive maintenance of industrial robots. *In 2017 International Conference on Advanced Systems and Electric Technologies (IC_ASET), IEEE.*, 412-417.

Breiman, L. (2001). Random forests. *Machine learning 45.1*, 5-32.

Breunig, M. M. (2000). LOF: identifying density-based local outliers. . *In Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 93-104.

Brownlee, J. (2016 , March 16). *machine learning mastery*. Retrieved from Machine Learning Algorithms: https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/

Brownlee, J. (2018, April 27). *How to Calculate Correlation Between Variables in Python*. Retrieved from Machine learning mastery: https://machinelearningmastery.com/how-to-use-correlation-to-understand-the-relationship-between-variables/

Brownlee, J. (2018, April 27). *How to Calculate Correlation Between Variables in Python*. Retrieved from machine learning mastery: https://machinelearningmastery.com/how-to-use-correlation-to-understand-the-relationship-between-variables/

Bujak, J. (2008). Mathematical modelling of a steam boiler room to research thermal efficiency. *Energy, 33(12),* , 1779-1787.

Candés, E. L. (2010). Robust principal component analysis?: Recovering low-rank matrices from sparse errors. . *In 2010 IEEE Sensor Array and Multichannel Signal Processing Workshop. IEEE.*, 201-204.

Cauchi, N. M. (2017). Model-based predictive maintenance in building automation systems with user discomfort. . *Energy, 138,* , 306-315.

Chalise, S. G. (2015). Data center energy systems: Current technology and future direction. . *In 2015 IEEE Power & Energy Society General Meeting*, 1-5.

Chan, H. Y. (2012). *Washington Patent No. 8,275,642.*

Chen, C. W. (2018). Detecting Data Center Cooling Problems Using a Data-driven Approach. . *In Proceedings of the 9th Asia-Pacific Workshop on Systems*, 1-8.

Chen, G. H. (2008). Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services. *In NSDI, Vol. 8,* 337-350.

Chen, K. Y. (2011). Using SVM based method for equipment fault detection in a thermal power plant. . *Computers in industry, 62(1)*, 42-50.

Chen, M. Z. (2004). Failure diagnosis using decision trees. . *In International Conference on Autonomic Computing, 2004. Proceedings, IEEE.*, 36-43.

Choo, K. G. (2013). Measured and simulated energy consumption analysis of a data center on an academic campus. . *In 29th IEEE Semiconductor Thermal Measurement and Management Symposium, IEEE.*, 231-236.

Cisco. (2017). *Forecast and methodology, 2016–2021. White Paper.* Cisco Global Cloud.

Cole, D. (2011). *Data center energy efficiency–looking beyond pue.* . No Limits Software, White Paper, 4.

Covell, M. &. (1991). A new, efficient structure for the short-time Fourier transform, with an application in code-division sonar imaging. *In icassp, IEEE.*, 2041-2044.

DataQuest. (2019). *The 10 Best Machine Learning Algorithms for Data Science Beginners*. Retrieved from DATAQUEST: https://www.dataquest.io/blog/top-10-machine-learning-algorithms-for-beginners/

Dhilon, B. (2006). Maintenability, Maintenance, and Reliability for Engineers. *CRC Press, Boca Raton.* Florida.

Dickason, P. (n.d.). Retrieved from https://community.spiceworks.com/people/pdickason/projects/consolidate-servers

Elforjani, M. &. (2017). Prognosis of bearing acoustic emission signals using supervised machine learning. . *IEEE Transactions on industrial electronics, 65(7), ,* 5864-5871.

El-Sayed, N. S. (2012). Temperature management in data centers: why some (might) like it hot. *In Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems*, 163-174.

Evans, T. (2012). *The different technologies for cooling data centers.* APC white paper, 59.

Falcão, F. Z. (2019). Quantitative comparison of unsupervised anomaly detection algorithms for intrusion detection. *In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, (pp. 318-327).

Fiix. (2019). *Breakdown maintenance*. Retrieved from Fiix: https://www.fiixsoftware.com/breakdown-maintenance/

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. . *Annals of eugenics, 7(2),* , 179-188.

Fu, X. W. (2011). How much power oversubscription is safe and allowed in data centers. . *In Proceedings of the 8th ACM international conference on Autonomic computing*, 21-30.

García, S. R.-G. (2016). Big data preprocessing: methods and prospects. *Big Data Analytics, 1(1), 9.*

Gill, P. J. (2011). Understanding network failures in data centers, measurement, analysis, and implications. (pp. 350-361). In Proceedings of the ACM SIGCOMM 2011 conference.

Gill, P. J. (2011). Understanding network failures in data centers: measurement, analysis, and implications. *In Proceedings of the ACM SIGCOMM 2011 conference*, (pp. 350-361).

Gill, P. J. (2011). Understanding network failures in data centers: measurement, analysis, and implications. 350-361.

Goiri, Í. H. (2015). Matching renewable energy supply and demand in green datacenters. . *Ad Hoc Networks*, 520-534.

Goldstein, M. &. (2016). *A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data.* . PloS one, 11(4), e0152173.

Golub, G. H. (1971). Singular value decomposition and least squares solutions. *In Linear Algebra, Springer, Berlin, Heidelberg.*, 134-151.

Google. (n.d.). *Efficiency: How we do it [Online]*. Retrieved from https://www.google.com/about/datacenters/efficiency/

Gray, J. (1986). Why do computers stop and what can be done about it? *In Symposium on reliability in distributed software and database systems*, 3-12.

Hair, J. F. (2006). *Multivariate data analysis (Vol. 6).*

Hamann, H. F. (2010). Thermal zones for more efficient data center energy management. *In 2010 12th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems, IEEE.*, 1-6.

Han, H. C. (2019). Least squares support vector machine (LS-SVM)-based chiller fault diagnosis using fault indicative features. . *Applied Thermal Engineering, 154,* , 540-547.

Heising, C. (2007). IEEE recommended practice for the design of reliable industrial and commercial power systems. IEEE Inc., New York. *IEEE Inc., New York.*

Ins., P. (2013). *Study on data center outages.* Retrieved from Ponemon Institute, sponsored by Emerson Network Power.: Available: www.emersonnetworkpower.com/documentation

J. Goldberger, S. R. (2005). Neighbourhood Components Analysis. *Neighbourhood Components Analysis*, 513-520.

J., E. R. (2016). *Deepmind AI reduces Google data centre cooling bill by 40%.* DeepMind blog.

Karidis, J. M. (2009). rue value: assessing and optimizing the cost of computing at the data center level. *In Proceedings of the 6th ACM conference on Computing frontiers*, (pp. 185-192).

Kohavi, R. &. (1995). Feature Subset Selection Using the Wrapper Method: Overfitting and Dynamic Search Space Topology. *In KDD* , 192-197.

Kontorinis, V. Z. (2012). Managing distributed ups energy for effective power capping in data centers. *In 2012 39th Annual International Symposium on Computer Architecture (ISCA), IEEE.*, 488-499.

Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. . *Emerging artificial intelligence applications in computer engineering, 160,* , 3-24.

Kriegel, H. P. (2008). Angle-based outlier detection in high-dimensional data. *In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 444-452.

Kriegel, H. P. (2008). Angle-based outlier detection in high-dimensional data. . *In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining,* 444-452.

Lahiri, R. N. (2008). Importance of strategic maintenance management for Indian utility industry. *IEEE In 2008 IEEE Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century,*, 1-5.

Lal, T. N. (2006). Embedded methods. *In Feature extraction* (pp. 137-165). Berlin: Springer.

LaRiviere, J. M. (2016). Where predictive analytics is having the biggest impact. *Harvard business review, 25.*

LeCun, Y. B. (2015). Deep learning. *nature, 521(7553),* , 436-444.

Lee, J. B. (2014). Cyber-integrated big data analytics agent for industry 4.0 applications. *de Agenten im Umfeld von Industrie, 4.*

Lee, Y. L. (2017). DC-Prophet: Predicting Catastrophic Machine Failures in DataCenters. *In Joint European Conference on Machine Learning and Knowledge Discovery in Databases* , (pp. 64-76).

Li, G. C. (2018). An improved decision tree-based fault diagnosis method for practical variable refrigerant flow system using virtual sensor-based fault indicators. *Applied Thermal Engineering, 129*, 1292-1303.

Lin, C. C. (2005). A neural network application for reliability modelling and condition-based predictive maintenance. *The International Journal of Advanced Manufacturing Technology, 25(1-2),*, 174-179.

Liserre, M. A. (2016). Power routing in modular smart transformers: Active thermal control through uneven loading of cells. *IEEE Industrial Electronics Magazine, 10(3),*, 43-53.

Liu, F. T. (2008). Isolation forest. *In 2008 Eighth IEEE International Conference on Data Mining, IEEE.*, 413-422.

Liu, Z. M. (2017). Remaining useful life estimation of insulated gate biploar transistors (IGBTs) based on a novel Volterra k-nearest neighbor optimally pruned extreme learning machine (VKOPP) model using degradation data. *Sensors, 17(11),* , 2524.

Marfatia., S. D. (2013). *Predictive maintenance for dummies.* . Retrieved from hoboken: http://cdn2.hubspot.net/hub/215445/file-534377792-pdf/Predictive_Maintenence_forDummies-Operations.pdf

Market, M. (Jun 2019, June). *Predictive Maintenance Market*. Retrieved from Predictive Maintenance Market: https://www.marketsandmarkets.com/Market-Reports/operational-predictive-maintenance-market-8656856.html

Marroquin, J. L. (1993). Some extensions of the k-means algorithm for image segmentation and pattern classification (No. AI-M-1390). . *MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTELLIGENCE LAB.*

Marty, M. R. (2007). Virtual hierarchies to support server consolidation. (pp. 46-56). In Proceedings of the 34th annual international symposium on Computer architecture.

Mazumdar, S. &. (2016). Forecasting data center resource usage: An experimental comparison with time-series methods. . *In International Conference on Soft Computing and Pattern Recognition, Springer, Cham.*, 151-165.

Miller, R. (2011). Uptime institute: The average pue is 1.8. . Data Center Knowledge, 10(05).

Mobley, R. K. (2002). An Introduction to Predictive Maintenance. *Butterworth-Heinemann.*

Mourão-Miranda, J. H.-T. (2011). Patient classification as an outlier detection problem: an application of the one-class support vector machine. . *Neuroimage, 58(3),* 793-804.

Nadabaica, D. C. (2012). STUDY OF THE ADVANTAGES OF PREDICTIVE MAINTENANCE IN THE MONITORING OF ROLLING BEARINGS. *Environmental Engineering & Management Journal (EEMJ), 11(12).*

Naguib, R. (2011). The changing landscape of chillers' energy efficiency. *Energy engineering*, 25-45.

News, F. (2015, October 21). *Google, Yahoo hacked by US government*. Retrieved from Fox News: https://www.foxnews.com/tech/google-yahoo-hacked-by-us-government

Nguyen, K. A. (2015). Multi-level predictive maintenance for multi-component systems. . *Reliability engineering & system safety, 144,*, 83-94.

Patel, C. D. (2005). *Cost model for planning, development and operation of a data center.* Hewlett-Packard Laboratories Technical Report.

Patterson, D. A. (2002). A Simple Way to Estimate the Cost of Downtime. . *In LISA (Vol. 2)*, 185-188.

Patterson, M. K. (2012). Energy efficiency metrics. *In Energy efficient thermal management of data centers* (pp. 237-271). Boston, MA.: Springer.

Ponemon. (2013). *Ponemon Institute*. Retrieved from Emerson Network Power: www.emersonnetworkpower.com/documentation

Ponemon. (2016). *Cost of Data Center Outages.* Retrieved from www.emersonnetworkpower.com/documentation

Ponemon Institute, s. b. (2013). *Emerson Network Power.* Retrieved from Emerson Power: www.emersonnetworkpower.com/documentation

Ponemon, L. (2016). *Cost of Data Center Outages.* Ponemon Insitute, Data Center Performance Benchmark Serie.

Power, E. N. (2009). *Energy logic: Reducing data center energy consumption by creating savings that cascade across systems. .* White paper, Emerson Electric Co, 9.

Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.

Prerau, M. J. (2000). *Unsupervised anomaly detection using an optimized K-nearest neighbors algorithm.* Columbia : Undergraduate Thesis, Columbia University.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning 1.1,* 81-106.

Ran, Y. Z. (2019). A Survey of Predictive Maintenance: Systems, Purposes and Approaches. . *arXiv preprint arXiv:1912.07383.*

Rasmussen, N. (2014). *Raised Floors vs Hard Floors for Data Center Applications.* White Paper 19, Revision 3. Schneider Electric.

Ravanshad, A. (2018, April 27). Retrieved from How to choose machine learning algorithms?: https://medium.com/@aravanshad/how-to-choose-machine-learning-algorithms-9a92a448e0df

Rish, I. (2001). An empirical study of the naive Bayes classifier. . *In IJCAI 2001 workshop on empirical methods in artificial intelligence (Vol. 3, No. 22)*, 41-46.

Roman, V. (2019, Jan 15). *Supervised Learning: Basics of Linear Regression*. Retrieved from towards data science: https://towardsdatascience.com/supervised-learning-basics-of-linear-regression-1cbab48d0eba

Roy, S. (2001). Reliability considerations for data centers power architectures. *Twenty-Third International Telecommunications Energy Conference. INTELEC 2001*, (pp. 406 – 411).

Rumsey, D. J. (2009). *How to Calculate a Correlation*. Retrieved from dummies: https://www.dummies.com/education/math/statistics/how-to-calculate-a-correlation/

RUNDLE, M. (2016, February 29). *Raspberry Pi 3 is the first with built-in Wi-Fi*. Retrieved from Wired: https://www.wired.co.uk/article/raspberry-pi-three-wifi-bluetooth-release-price-cost

Russom, P. (2011). Big data analytics. *TDWI best practices report, fourth quarter, 19(4),* , 1-34.

S., C. (2011). Manhattan Distance. *In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA.*

Saha, S. &. (2017). Data centre temperature monitoring with ESP8266 based Wireless Sensor Network and cloud based dashboard with real time alert system. *In 2017 Devices for Integrated Circuit (DevIC), IEEE*, 307-310.

Sakalkar, V. K. (2020). Data Center Power Oversubscription with a Medium Voltage Power Plane and Priority-Aware Capping. . *In Proceedings of the Twenty-Fifth*

International Conference on Architectural Support for Programming Languages and Operating Systems., 497-511.

Samanta, B. &.-B. (2003). Artificial neural network based fault diagnostics of rolling element bearings using time-domain features. . *Mechanical systems and signal processing, 17(2),* , 317-328.

Sánchez-Maroño, N. A.-B.-S. (2007). Filter methods for feature selection–a comparative study. , . *In International Conference on Intelligent Data Engineering and Automated Learning* (pp. 178-187). Berlin: Springer.

Sankar, S. S. (2011). Impact of temperature on hard disk drive reliability in large datacenters. *In 2011 IEEE/IFIP 41st international conference on dependable systems & networks (DSN). IEEE.*, 530-537.

Sasser, J. (2014). *A look at data center cooling technologies.* Retrieved from https://journal. uptimeinstitute. com/a-look-at-data-center-cooling-technologies

Schölkopf, B. P.-T. (2001). Estimating the support of a high-dimensional distribution. . *Neural computation, 13(7),* , 1443-1471.

Schölkopf, B. W. (2000). Support vector method for novelty detection. *In Advances in neural information processing systems*, 582-588.

Schölkopf, B. W.-T. (2000). Support vector method for novelty detection. . *In Advances in neural information processing systems*, 582-588.

Schroeder, B. P. (2009). DRAM errors in the wild: a large-scale field study. *ACM SIGMETRICS Performance Evaluation Review, 37(1)*, 193-204.

Shannon, C. E. (1949). *The Mathematical Theory of Communication, by CE Shannon (and Recent Contributions to the Mathematical Theory of Communication),.* illinois : University of illinois Press.

Sharma, R. K. (2005). Balance of power: Dynamic thermal management for internet data centers. *IEEE Internet Computing, 9(1),* 42-49.

Shehabi, A. S. (2016). *United states data center energy usage report (No. LBNL-1005775). Lawrence Berkeley National Lab.(LBNL),.* United States: Berkeley, CA.

Singh, S. (2019, September 23). *Cloud Data Center Locations for Top 3 CSPs (AWS, Azure & Google)*. Retrieved from TECHYAZ : https://techyaz.com/cloud/cloud-data-center-locations-for-top-3-csps-aws-azure-google/

Song, Z. M. (2014). A dynamic compact thermal model for data center analysis and control using the zonal method and artificial neural networks. *Applied thermal engineering, 62(1),*, 48-57.

Sorsa, T. &. (1993). Application of artificial neural networks in process fault diagnosis. *Automatica, 29(4),* 843-849.

SRIBLOG. (2019, march 5). *ABOD and its PyOD python module*. Retrieved from SRI BLOG: http://hari-sriblog.blogspot.com/2019/03/abod-and-implementation-pyod-python.html

Su, C. J. (2018). Big Data Preventive Maintenance for Hard Disk Failure Detection. . *International Journal of Information and Education Technology, 8(7).*

Susto, G. A. (2014). Machine learning for predictive maintenance: A multiple classifier approach. . *IEEE Transactions on Industrial Informatics, 11(3),* , 812-820.

Sutton, R. S. (1998). *Introduction to reinforcement learning.* Cambridge: MIT press.

Sverdlik, Y. (2016, October 19). *Datacenter Knowledge*. Retrieved from Here are the Top DCIM Software Vendors, According to Gartner: https://www.datacenterknowledge.com/archives/2016/10/19/here-are-the-top-dcim-software-vendors-according-to-gartner

Swiniarski, R. W. (2001). Rough sets as a front end of neural-networks texture classifiers. . *Neurocomputing, 36(1-4),* , 85-102.

Swiniarski, R. W. (2003). Rough set methods in feature selection and recognition. . *Pattern recognition letters, 24(6),*, 833-849.

Tan, C. J. (2019). Netbouncer: active device and link failure localization in data center networks. . *In 16th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 19),* 599.

Tian, J. A. (2014). Anomaly detection using self-organizing maps-based k-nearest neighbor algorithm. . *In Proceedings of the European Conference of the Prognostics and Health Management Society*, 1-9.

Tiwari, S. (2011). Need of green computing measures for Indian IT industry. *Journal of Energy Technologies and Policy , 1(4).*

Torell, N. R. (2014). Comparing Data Center Power Distribution Architectures. White Paper, 3:1–14.

Turner IV, W. P. (2006). *Tier classification define site infrastructure performance.* . Uptime Institute.

Uddin, M. S. (2014). Metrics for Computing Performance of Data Center for Instigating Energy Efficient Data Centers.

Van der Aalst, W. M. (2010). Process mining: a two-step approach to balance between underfitting and overfitting. . *Software & Systems Modeling, 9(1),* , 87.

Vaseghi, S. V. (2008). *Advanced digital signal processing and noise reduction.* John Wiley & Sons.

Verge, J. (2013, December 03). *Data center knowledge*. Retrieved from Data Center Downtime Costs: https://www.datacenterknowledge.com/archives/2013/12/03/study-cost-data-center-downtime-rising

Volico. (2018, December 4). Retrieved from https://www.volico.com/what-are-the-major-differences-between-data-center-tiers/

Wang, G. W. (2016). Increasing large-scale data center capacity by statistical power control. *In Proceedings of the Eleventh European Conference on Computer Systems*, 1-15.

Wang, J. Z. (2017). A new paradigm of cloud-based predictive maintenance for intelligent manufacturing. *Journal of Intelligent Manufacturing, 28(5)*, 1125-1137.

Wang, T. S. (2014). Rethinking the data center networking: Architecture, network protocols, and resource sharing. *IEEE access, 2,* 1481-1496.

Wei, J. D. (2017). Remaining useful life prediction and state of health diagnosis for lithium-ion batteries using particle filter and support vector regression. . *IEEE Transactions on Industrial Electronics, 65(7),* 5634-5643.

Wemhoff, A. P. (2013). Thermodynamic modeling of data center cooling systems. *In ASME 2013 International Technical Conference and Exhibition on Packaging* *and Integration of Electronic and Photonic Microsystems. American Society of Mechanical Engineers Digital Collection.*

Wiboonrat, M. (2008). An optimal data center availability and investment trade-offs. *In 2008 Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, IEEE.,* (pp. 712-719).).

Wilimitis, D. (2018 , Dec 12). *The Kernel Trick in Support Vector Classification*. Retrieved from towards data science: https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f

Wold, S. E. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems, 2(1-3),*, 37-52.

Xiaojin, Z. a. (2002). Learning from labeled and unlabeled data with label propagation.

York-ACSC. (2015, June 30). AIR-COOLED SCROLL CHILLER. Johnson Controls.

Zare Mehrjardi, Y. (2018). Application of multivariate control charts for condition based maintenanc. *e. International Journal of Engineering, 31(4),* 597-604.

Zhang, X. C. (2010). Experimental investigation and mathematical modelling of wood combustion in a moving grate boiler. . *Fuel Processing Technology, 91(11), ,* 1491-1499.

Zhang, Z. H. (2006). Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. . *In 18th International Conference on Pattern Recognition (ICPR'06) Vol. 3, IEEE.,* 1135-1138. Retrieved from Wikipedia.

Zheng, Z. P. (2019). A Novel Method for Lithium-ion Battery Remaining Useful Life Prediction Using Time Window and Gradient Boosting Decision Trees. *n 2019 10th International Conference on Power Electronics and ECCE Asia (ICPE 2019-ECCE Asia), IEEE.,* 3297-3302.

Zhou, S. Z. (2005). A neighborhood-based clustering algorithm. In Pacific-Asia Conference on Knowledge Discovery and Data Mining. *In Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 361-371). Berlin, Heidelberg.: Springer.

ZhuΓ́, X. &. (2002). *Learning from labeled and unlabeled data with label propagation.* Technical Report. Carnegie Mellon University:School of Computer Science, Pittsburgh PA, USA.