Internet of Things and Neural Network Based Energy Optimization and Predictive Maintenance Techniques in Heterogeneous Data Centers

by

Vishal Kumar Singh

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Information Systems Engineering)
University of Michigan - Dearborn
2016

Doctoral Committee:

Associate Professor Jinhua Guo, Chair
Assistant Professor Jian Hu
Associate Professor Shengquan Wang
Associate Professor Weidong Xiang

DEDICATION

I would like to lovingly dedicate this dissertation to my parents.

ACKNOWLEDGEMENTS

This dissertation would not have been successfully completed without the contributions of many people. First of all, I would like to thank my thesis advisor, who has been the driving force in navigating me throughout my PhD training. I am very grateful to my family, friends and colleagues to constantly encourage me to stay on the course and relentlessly support me through ups and downs. I would like to express my gratitude to all my PhD committee members for suggestions on improving my work. Finally, I would like to thank the leadership of a midsize data center to offer their facility for my research work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Rapid growth of cloud-based systems is accelerating growth of data centers. Private and public cloud service providers are increasingly deploying data centers all around the world. The need for edge locations by cloud computing providers has created large demand for leasing space and power from midsize data centers in smaller cities. Midsize data centers are typically modular and heterogeneous demanding 100% availability along with high service level agreements.

Data centers are recognized as an increasingly troublesome percentage of electricity consumption. Growing energy costs and environmental responsibility have placed the data center industry, particularly midsize data centers under increasing pressure to improve its operational efficiency.

The power consumption is mainly due to servers and networking devices on computing side and cooling systems on the facility side. The facility side systems have complex interactions with each other. The static control logic and high number of configuration and nonlinear interdependency create challenges in understanding and optimizing energy efficiency. Doing analytical or experimental approach to determine optimum configuration is very challenging however, a learning based approach has proven to be effective for optimizing complex operations. Machine learning methodologies have proven to be effective for optimizing complex systems.

In this thesis, we utilize a learning engine that learns from operationally collected data to accurately predict Power Usage Effectiveness (PUE) and creation of intelligent method to validate and test results. We explore new techniques on how to design and implement Internet of Things (IoT) platform to collect, store and analyze data.

First, we study using machine learning framework to predictively detect issues in facility side systems in a modular midsize data center. We propose ways to recognize gaps between optimal values and operational values to identify potential issues.

Second, we study using machine learning techniques to optimize power usage in facility side systems in a modular midsize data center. We have experimented with neural network controllers to further optimize the data suite cooling system energy consumption in real time.

We designed, implemented, and deployed an Internet of Things framework to collect relevant information from facility side infrastructure. We designed flexible configuration controllers to connect all facility side infrastructure within data center ecosystem. We addressed resiliency by creating reductant controls network and mission critical alerting via edge device. The data collected was also used to enhance service processes that improved operational service level metrics.

We observed high impact on service metrics with faster response time (increased 77%) and first time resolution went up by 32%. Further, our experimental results show that we can predictively identify issues in the cooling systems. And, the anomalies in the systems can be identified 30 days to 60 days ahead. We also see the potential to optimize power usage efficiency in the range of 3% to 6%. In the future, more samples of issues and corrective actions can be analyzed to create practical implementation of neural network based controller for real-time optimization.

Chapter One – Introduction

1.1. Introduction

Extant literature reports that the increasing use of technology has caused a drastic increase in power capacity and density requirements. Annual consumption of electricity by US datacenters is projected to be roughly 138 billion kilowatt-hours by 2020[1]. The gap between buildings and need of IT industry has been increasing. The current needs of the IT industry include better processing performance [2], increase in available storage space [3], and improvement in access [4] and reduced latency [5]. Additionally, there is also a need to optimize solutions provided by IT vendors who are under constant pressure of meeting deadlines while dealing with a constantly shrinking budget.

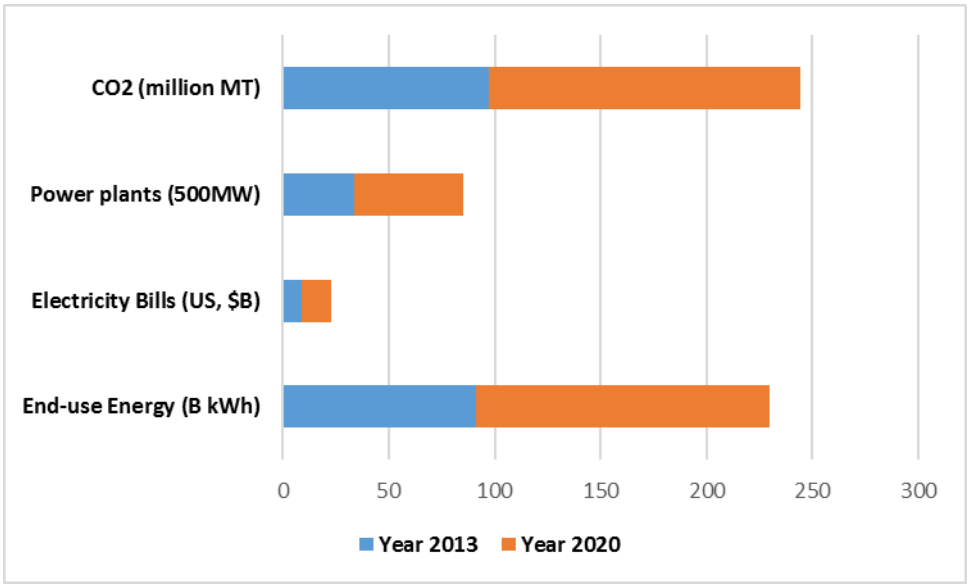


Figure 1: Shows the estimated power usage (in billions of kilowatt-hours), and the cost of power used, by U.S. data centers in 2013 and 2020, and the number of power plants needed to support the demand. Top bar shows carbon dioxide (CO2) emissions in millions of metric tons. (Source: NRDC)

Improving energy efficiency within a data center has been a key element of research in the last decade [1]. A prominent approach, which has been adopted, is the optimization of facilities through aisle containment [5]. Miller [6] contends that energy utilization and efficiency can further be improved through the management of virtual server loads. The use of

this approach has been effective in the management of energy through the combination of building automation and virtualization [7].

The historical as well as future trends of electricity use have been illustrated in Figure 1. In order to increase the efficiency of the data center, a high sense of urgency is required. There is a large volume of servers deployed at global scale in data centers. The size of the existing data centers is now being increased on a regular basis and there is more frequent building of new data centers. To some extent a parallel can be drawn between the scale of data centers and an electricity grid [8]. This study focuses on energy management in a modular data center, which is a key aspect of the scalability of data center. Of the total, 35% of the energy costs are entailed by the total cost of ownership (TCO) [9]. Since higher watts per square foot are required by high density data centers, they are driven by powerful servers.

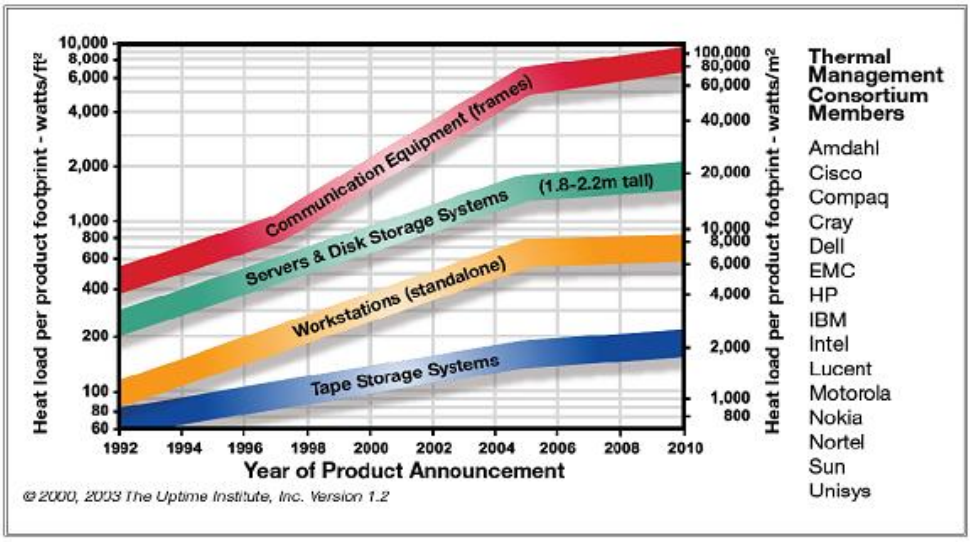


Figure 2: Heat Load – 2000, 2003 the Uptime Institute.

Higher heat per square foot will generate higher watts per square foot. Thus more energy will be required for removing heat. Heat load trend has been illustrated in Figure 2 [10]. In the industry, average energy efficiency is only 50% and electricity costs are high compared to server costs [10].

The point of modularity in data centers, which is more efficient and can be deployed in various environment settings, has arrived after more than 30 years. The issue of energy efficiency of data centers is getting immense attention. Green data centers are getting high attention and a compound annual growth of 26.35% during 2014 -2019 is forecasted for US markets [4]. The current study aims at defining the scope of the problem, identifying the research questions and discussing approaches for addressing these questions.

Generally, hundreds of thousands of servers are used across the data centers throughout the world for operating large scale information technology [5]. Over 300,000 servers run in the Chicago data center of Microsoft [6], while there are over a million servers running in Google [11]. With the growth of cloud computing, larger scalability will be required by data centers [11] and with the growth of competencies and infrastructure, most of the data storage and computation power of the world will be hosted by few infrastructure providers. As the competition for leadership positions is increasing, high energy efficiency will be a prerequisite for these larger data centers. Since minor energy savings can lead to cost savings for users, it will be a strategic competitive advantage to be energy efficient.

With gradual increase in the number and size of data centers, the environmentalists are increasingly becoming greatly concerned. Throughout the world, electricity use by data centers and telecommunications network is pegged at number 5 position and a 12% growth rate is expected. The global greenhouse emissions of data centers are over 2% and electricity produced from coal is used by a majority of the data centers [12]. Technologies like cloud storage and smart grid can be used for reducing the energy consumption worldwide [12]. High-energy capacity constraints will arise if data centers maintain the current energy trends. Thus, in order to enhance the unlimited potential of cloud, it is critical that operational efficiency and design of green data center be emphasized.

With data centers spread across the world, it is quite surprising that average energy efficiency of the industry is still less than 50% [11], which is questionable and mandates a review. A brief history of the data center has been discussed in the next section. After which the ideal data center energy profile is described and a path is identified for achieving it. Datacenter can be perceived as a warehouse-sized computer because of the convergence of networking, storage computing and power [13]. Data centers are primarily of two types. The first is the brick-and-mortar facility, which is mostly a co-location taking longer time to build but houses larger number of servers. The second is a flexible data center that can be categorized as: 1) Containerized - which involves the pre-population of a shipping container with a few thousand servers that can be combined with support infrastructure and takes less than a month for transportation and deploying [14]. 2) Modular stick built - a prefabricated (standard bill of materials) design which can be built in an environment like a building shell or a warehouse.

Data centers can be either shared or private. In private data centers, an entire building is used for the purpose of hosting applications under the proprietorship of the building owner.

In shared data centers, the owner to several entities [14] leases portions of the building out. Following services can be provided by shared data centers:

- Area-as-a-service, wherein square footage is leased out by the data center including power and cooling as well as rack layout;
- Infrastructure-as-a-Service (IaaS) wherein physical servers are used for leasing out;
- Platform-as-a-Service (PaaS) wherein virtual disks and CPUs are leased out;
- Software–as-a-Service (SaaS) wherein hosted software is leased out [14].

Majority of the large data centers right now are private. The trend of large shared data centers is increasing. It is natural for servers to reside in a multi-tenant heterogeneous data centers. A strong need was presented by Facebook for shortening the time required for bringing new data centers online. Facebook worked with Emerson Network Power that pioneered "rapid deployment data center" (RDDC) an innovative approach to the construction of data centers. In RDDC a modular prefabrication of the entire custom-designed, freestanding facility is carried out which can then be assembled onsite lowering the time required in bringing the new data centers online, without any premium paid for faster deployment [15]. Large data centers like Google, Microsoft and Amazon have homogenous standard systems as compared to smaller privately held multi-tenant data centers that have heterogeneous (non-standard) systems.

Can the entire industry benefit from this trend or is it only a niche practice that only hyper scale operators can use?

Although all containerized data centers are prefabricated, the reverse may not always be true. It is rather quick to deploy or relocate containerized systems package infrastructure and IT systems in standard form factor containers to be deployed in a freestanding, all-in-one data center container or a large data center. Containerized data centers are an effective solution for modular expansion in freestanding data centers situated in remote areas or in large facilities [16].

The traditional data center construction practices are being replaced by prefabricated data centers, wherein the prefabrication of an entire facility is done offsite and the shipping is done in modules and the assembly is done onsite [15]. The prefabrication of a data center is custom made according to the business objectives, IT applications, technology profile, climate and geography which comes with the advantage of economy and speed of prefabrication and

modular design. These results in a tightly integrated state-of-the-art facility that can be deployed at a lower cost and faster compared to a traditional facility [16].

The mainframes of 1960s mark the beginning of data. These large machines would occupy an entire room. They were very powerful and were mainly used for critical processing by military initiatives, space and large size companies [17]. The implementation of these machines was very lengthy as they were highly customized. Additionally, custom programs were required for running them and frequent errors were reported. In the 1970s, came the personal computers and their availability to home and small businesses increased. These machines were equipped with standardized operating systems and hardware which made it easy to develop software [17]. Next came Ethernet, through which distributed computing was enabled [18]. This marked the evolution of modern data center. Two trends were observed: 1) High-performance computing, wherein mainframes were replaced with low cost large clusters of servers [19]. Large companies made a shift to the servers in data centers from the mainframe systems.2) With the use of personal computing, online application usage went high as data and computing could now be executed on third party servers [20]. In order to gain economies of scale, service providers began collocating their servers in large facilities, of which some were third-party shared data centers while the others were privately owned.

This marks the journey from mainframes to the warehouse-sized server farms used currently. Over the last four years, these large data centers have developed. The existing data centers cannot handle the pace of computing and innovation growth. It is often found that for incremental expansion, old designs with minor enhancements are used by data centers [21]. Cloud computing is shaping into a universal technology with datacenters as its foundation. The scalability of data centers is facing the challenge of electricity costs. The major challenges that data center environments are facing today include power delivery, heat management and electricity consumption. Most data centers do not manage energy optimally and on average, the data center efficiency is less than 50%. The enormity of this challenge is widely acknowledged in addition to the high sense of urgency for resolving it [21].

According to evidences in literature, data center availability and 100% uptime SLA (Service Level Agreements) are basic requirements which need preventative maintenance and also look for predictive maintenance. The focus of the following section of the introduction is to revisit the need for predictive maintenance in heterogeneous data center facilities.

In 1999, a British entrepreneur Kevin Ashton put forth the Internet of Things (also referred as IoT) so that internet-based information architecture can be characterized. IoT is defined as a network of things or physical objects that are inserted with network connectivity, sensors, software and electronics that facilitate these things in the collecting and exchanging data [22]. It mainly represents the concept of network connectivity in everyday objects. So, in addition to computers and smart phones being connected to the internet, just about anything like street-lights, televisions, thermostats, washing machines and cars will also have internet connectivity. According to Ning and Wang [22], the IoT will be a massive network that will connect billions of things as well as encompass heterogeneous networks.

A significant rise in internet traffic is expected with the increasing number of devices connected to the Internet. According to Cisco in 2014 the Internet traffic generated by non-PC devices amounted up to 40%and this is expected to increase to 67% in 2019. Cisco further identified that 24% of all connected devices in 2014 was contributed by "Machine to Machine" ("M2M") connections like IoT such as automotive, healthcare, home and industrial verticals and this contribution is expected to grow to 43% in 2019 [23].

Monitoring of datacenter critical equipment's have given rise to adaption of IoT platform. With rise of Internet of things (IoT), strength of machine to machine interaction is making an impact on asset management and field services. We can use machine sensors to monitor their behavior continuously. The integration of IoT has made it possible to significantly augment asset instrumentation allowing IP based management and the ability to gather and analyze fine-grained data from uninterruptible power supply (UPS), computer room air conditioning (CRAC), circuits, power distribution unit (PDU) etc. Furthermore, the use of IoT based technology has made it possible to gather information from smaller sections of the data centers which previously were not focused on. These sections can include data pods, suites etc. Thus, it is possible to increase the interconnectivity of the complex systems such as software systems, mobile devices and people.

Data centers include major building systems (end devices) like Automatic Transfer Switch, Transformer, Generator, Cooling Tower, CRAC, Chillers, UPS and Pumps. Data centers also comprises of the devices that are mechanical and electrical systems of a customer data suite. Figure 3 shows major components of the data center. All systems need to be connected and monitored in data centers.

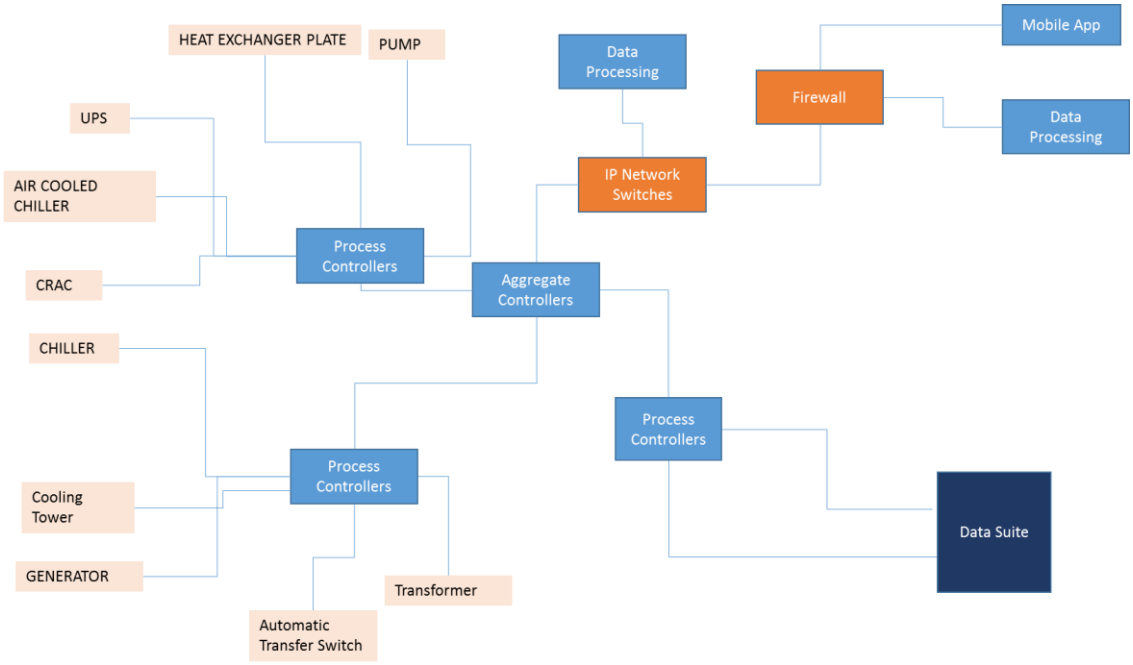


Figure 3: Major components of a datacenter

IoT architecture can be broken into 4 layers [24] [25] [26]. As shown in figure 4, they are perception layer, network layer, middle-ware layer and application layer.

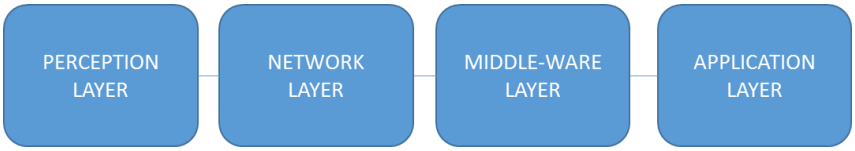


Figure 4: Layers of IoT architecture

In the data center ecosystem, the end devices are connected to controllers via variety of machine communication protocols including but not limited to Modbus, Lon Works etc. Multiple controllers are connected to aggregate controllers. The aggregate controllers can also function as edge devices that perform basic alerting and analytics and data processing. The aggregate controllers are connected to more powerful data processing servers on-site or in the cloud. This IoT framework can be utilized to monitor, control, manage and store data from the data center ecosystem. In the next section, we discuss specific challenges and problems in a heterogeneous data center.

1.2. Challenges in a heterogeneous data center

In a data center there are challenges with limitations of the ability of controllers connecting to all end devices, predictive maintenance of systems and power optimization. We will discuss problems for each of these areas.

1.2.1. Problems in IoT systems

There are four major problems in a midsize heterogeneous data center.

a. **Inability of controllers to connect all end devices in the midsize heterogeneous datacenter ecosystem:** Data centers have mixed legacy and new electrical and mechanical systems. Midsize Data centers often find it formidable to bear the cost of replacing legacy systems. This can be attributed to the high cost of bringing them online in addition to the lacking control logic and sensors in controllers connecting them. Mission critical facilities such as data centers are highly affected by this blindness to end devices. It is imperative that all equipment be monitored at the same level of service levels to customers are to be maintained. Thus, there is no need for all the systems in Data centers such as legacy systems to be connected. The main problem is the way of connecting the various end devices in a heterogeneous datacenter for the collection of the relevant information to be monitored, maintained and optimized. The development of an IoT platform for collecting pertinent information from all end devices, analyzing it and storing it, will result in outcomes that can be further utilized by the right person for conducting predictive maintenance or for tackling any problems that may emerge. This may result in enhanced customer service level agreements.
b. **Lack of security in sensors:** Hackers can target all end devices and sensors sitting in Data centers for stealing information, falsifying data or disabling or corrupting the device. The vulnerability of Software based security is high in certain types of attacks. If a device or sensor is accessible, it can possibly be cloned or altered in behavior by loading malware. Through this, unauthorized users can take control of the device and carry out different activities like disabling the device or providing information to hostile parties. The problem here is finding ways of protecting the end devices and sensors from physical security breaches.

c. **IoT network resiliency:** Heterogeneous data centers are mission critical systems in the need of high redundancy and reliability. End devices and sensors are monitored and controlled by controllers. In the case of midsize data centers, IoT systems are generally single paced with a sole controller and a sole aggregate controller. In the case of a mission critical data center, the result of controller failure may be many hours of down time or complete blindness of network operations center (NOC) to real time monitoring information from the end devices.

d. **Alerting at the edge:** In cloud IoT systems, sensors and end devices connect to servers that are in the cloud or data centers for performing alerting and analytics. As a result of this, two major issues are raised for the IoT system. The first issue is with bandwidth constraints in pushing forward the data. The second issue is the high reliance on cloud for receiving mission critical alerts by the local NOC. Mission critical alerts will not be received by NOC if latency or service down time of network connectivity to cloud servers causes a network issue. The issue here is the way reliance of mission alerting from servers is mitigated in the remote datacenter or the cloud.

1.2.2. Predictive maintenance problem in heterogeneous data centers

Data center facility side infrastructure is critical. This mission critical nature of data center has demanded the data center operating with 100% uptime. This requires the facility site infrastructure maintenance to deliver 100% service level agreements (SLA). There have been several steps taken to perform preventative maintenance (PM). These are standard checklists performed at certain frequency generally quarterly or yearly basis. The purpose of this is to stay on top of upkeep and avoid costly emergency maintenance repairs. There are also data available on devices on failure rates based on run times. These methods help with predictively catching issues and resolve it. The data center facility side devices have complex interactions with each other. Traditional methods have limitation and to analyze these interactions among facility infrastructures to predictively identify issues. The IoT platform allows us to evaluate significant volumes of operational data and develop predictive insights which can then be applied immediately in real time to identify and resolve issues.

A typical data center requires a number of sub components such as powerful generators, HVAC systems, power delivery and transfer circuits, uninterruptible power supplies and liquid chiller loops. In addition, operators of data centers would also need tools and the ability to

measure parameters on a customer-by-customer basis. In the present scenario, a majority of the systems issues and failures are taken care of after the issues have occurred.

Data center customers demand quick responses, especially those customers who have agreed upon an aggressive service level agreement (SLAs) [27]. The complexity of modern data center architecture makes it very difficult to provide rapid responses and to minimize false alarms. Modern data centers have a highly heterogeneous mix of infrastructures, which in turn has led to the demand for home grown IoT solutions that can provide significantly enhanced levels of integration with the various systems. Having or developing such a platform opens innovative new avenues that were not technically or economically feasible in the past.

Machine-learning algorithms can be used to predict failures which can be preemptively checked and resolved before asset efficiency erodes. This ability raises a few questions such as:

Can one detect fault patterns and monitor equipment against them to predict future performance degradation in a heterogeneous data center?

In this thesis, we discuss the use of IoT framework to collect data from a heterogeneous modular data center and propose several novel methods to optimize energy consumption and audit systems operations using machine-learning techniques to predictively identify issues.

1.2.3. Power optimization problem of data center facilities

Data centers require large amounts of power. Therefore, it is very important that the available power is used efficiently. There are certain steps that can help with conserving and utilizing available power in an efficient manner such as Aisle containment, which improves efficiency of cooling systems, and load balancing of virtual servers which improves power consumed by servers (IT Load) [28]. Modern data centers are continuously looking for methods to help increase efficiency and lower their power needs. Since data centers require huge amounts of power, even very small increases in efficiency translates to significant savings in the amount of power being utilized. These in turn results in notable cost savings and reduce carbon emissions. PUE (power usage effectiveness) is still a leading metric in a data center. According to [29], a number of metrics have been developed, like CADE (Corporate Average Data Center Efficiency), which are being used to accurately determine the amount of power

being used by each and every subsystem within the data centers. This will benefit data centers by highlighting areas that are being underutilized within the data centers.

Data centers take a lot of time and effort to be set up and hence, it is not possible to provide customers with space within the data centers immediately. This has become a business issue since most customers want immediate access to data center space [27]. Thus, the idea of modular data centers and containerized data centers were developed. A number of organizations started developing a shell for standalone portable containers that can be used anywhere or they have started developing modular frameworks which can be placed within any existing building [27,30].

According to [27], the Internet of Things (IoT) is rapidly growing with projected $7.1 trillion by 2020. This has made it possible to significantly augment asset instrumentation allowing IP based management and the ability to gather and analyze fine-grained data from UPS, CRAC, circuits, PDUs etc. The demand to further optimize facility side infrastructure energy usage of data centers and nature of complex interactions between them have driven to explore machine-learning techniques. This discussion raises the question:

Can we further optimize facility side infrastructure energy usage in a heterogeneous modular data center using machine learning techniques?

In this thesis, we discuss the use of IoT framework to collect data from a heterogeneous modular data center and propose several novel methods to optimize energy usage using machine learning techniques and neural network (NN) controllers

1.3. Research Problems and Objectives

This thesis tackles the research challenges in the aforementioned three problems:

• How to design a robust IoT platform that securely connects all end devices in a data center?

• How to detect fault patterns and monitor equipment against them to predict future performance degradation?

• How to optimize facility side infrastructure energy usage in a heterogeneous modular data center using machine learning techniques?

In the first problem, this thesis studies limitations of controller firmware logic and protocols to connect all the end devices in midsize heterogeneous data center.

• How to incorporate all the protocols in a single controller?

• How to incorporate all control logic in a controller for all end devices in data center ecosystem?

• How to SLA can Improve by connecting and alerting on all devices in the data center?

In the second problem, this thesis studies using machine learning techniques to predictively identify issues in facility side systems in a heterogeneous modular data center. In particular, the following research problems are investigated:

• How to predictively identify gaps of operational and optimal values?

In the third problem, this thesis studies using machine learning techniques to optimize power usage effectiveness in facility side systems in a data center. Also, we have experimented with neural network controllers to optimize the data suite cooling system leading to interesting future work. In particular, the following research problems are investigated:

• How to select factors for neural network training?

• How to validate and test a neural network model?

• What are the challenges of implementing a neural network controller?

1.4. Contributions

The contributions of this thesis can be broadly divided into following categories: survey and analysis of the power saving strategies, evolution of fractal modular data center, review IoT platform, machine learning techniques, review of predictive maintenance and neural network controllers. Our Unique contributions include.

- Creation of all-inclusive firmware and protocol in a controller and protocol to connect all end devices in a data center.
- Design and implementation of trusted sensor, controller resiliency and alerting at the edge device.

- Utilizing machine learning to predictively identify gaps of operational and optimal values to detect anomalies in a resolve them
- Utilizing IOT platform improve midsize datacenter SLA's.
- Application of machine learning framework power optimization.

1.5. Thesis Organization

In Chapter 2, we present the survey of traditional approaches to power utilization management and overall understanding of IoT platform. We review predictive maintenance, machine learning based techniques to power utilization management. We also review controller simulation and data center simulation methods.

In Chapter 3, we study the following areas utilizing a data suite facility of a midsize data center. We present the implementation of an IoT platform and methodology of collecting data using IoT framework. We present detailed characterization of data sets collected. Explaining of the neural network model selected. We discuss methods of data pre-processing, variable selection, data sampling, neural network model approach, predictive maintenance of cooling subsystem and experimental approach to neural network based controller.

In Chapter 4, we discuss results. We review the results on sensitivity analysis understanding behavior of each factor to PUE. We test the predictive accuracy of the neural network model. We measure neural network energy optimized value to operational value to predictively identify issues in cooling systems. We also discuss improvements in operational metrics to improve service level agreements.

In Chapter 5, we discuss conclusions for both the research problem along with service metrics improvements

Chapter Two: State of the Art IoT Platform in Data Center

2.1. Modular Data Center

The delivery of data center space to customers takes longer if traditional methods are used in constructing data center space. Datacenter operators find data center space as a critical business factor. As a result of this, modular data centers as well as containerized data centers were developed. Modular data center is built by many companies for inside building shell while for outside standalone pods are built [31]. Few examples have been illustrated in Figure 3.

Figure 5: Examples of containerized/modular datacenter

Stick built

These data centers are built of bill materials as well as full assembly each material on site. This follows the typical fabricated approach

Prefabricated module

The building of the data suite or the data center is done as modules. These modules are large sub-assemblies and it is on site where they are fully assembled

Containerized

These are manufactures pod or prebuilt like suites. They can come fully assembled to be dropped at a location.

Following are the benefits of modular data center:

- Speed of Deployment: Since time to deploy is reduced by prefabricated data centers, organizations looking to accelerate data center deployment should find them appealing.
- Scalability: The modular approach to design opted by prefabricated data centers makes them innately scalable that allow on-demand and streamlined capacity expansion [32].
- Cost Control: Economies of scale are leveraged by prefabricated data centers and their offsite assembly that facilitates lower total cost of ownership makes it possible to have streamlined processes.
- Design Flexibility: There are no innate restrictions to prefabricated data centers in terms of aesthetics or functionality as they are custom designed.
- Performance: When assembly is done in a factory-controlled environment fit, finish, and quality of workmanship can be controlled better and exhaustive testing and optimization can be carried out before delivery.
- Intelligence: When the management of integrated components is done together, more dynamic capacity adjustments and higher IT productivity is possible.
- Project Management and Service: Project specification and execution is simplified with preassembled, integrated systems that feature components from a sole vendor selected on the basis of compatibility, in addition to ensuring service efficiency and maintenance throughout the deployment life cycle [33].

2.2. IoT Frame work

In this section, we discuss IoT layers and broad challenges in the IoT and specifically elaborating challenges of connecting all midsize data center facility side systems with an IoT platform. According to extant literature there are multiple challenges that are faced by CIOs

and other analysts with existing IoT networks. We will specifically discuss four areas of problem review and current state of the art, its limitation and my proposed solution.

At present point solutions are sold by the device manufacturers with specific controllers like fan coil controls, roof to unit controls and heat pump unit controls. The most popular units are addressed by them and the rest are left. Majority of the major equipment manufacturers have closed systems supporting just their branded equipment. The solution presented is not for turnkey solution for the data centers [34]. The current limitation is the lack of vendor neutral gateways and controllers addressing the different configuration for connecting to the various end devices in heterogeneous data centers. The current study proposes an aggregated approach for the creation of all-inclusive configurations for addressing the link to the various end devices in datacenter ecosystems. This facilitates the connection of all the systems and is accompanied by pre-packaged sensors. For easy installation, all the protocols and drivers are prewritten and loaded into the system. All the control logic and protocols required for the datacenter ecosystem are supported by the controller.

Ever since digitization and automation of the different devices installed across the modern urban environments has increased, there has been an evident creation of fresh security challenges for various industries that may or may not be regulated [35].Ever since digitization and automation of the different devices installed across the modern urban environments has increased, there has been an evident creation of fresh security challenges for various industries that may or may not be regulated [35]. As already discussed, hackers can target even the smallest of devices sitting on the Internet for stealing information, falsifying data or disabling or corrupting the device. The vulnerability of Software based security is high in certain types of attacks. If a device or sensor is accessible, it can possibly be cloned or altered in behavior by loading malware. Through this, unauthorized users can take control of the device and carry out different activities like disabling the device or providing information to hostile parties. Another type of threat is posed by replay attacks by the poor interception of secured password exchanges between host and client and reuse of the password for gaining access to the network. The hardware Trusted platform module has been proposed in the heterogeneous data center IoT platform. Another type of threat is posed by replay attacks by the poor interception of secured password exchanges between host and client and reuse of the password for gaining access to the network. IoT does not have strong hardware authentication. It has been recently observed that reverse engineer is possible for the various types of Physically Unclonable Functions (PUFs). The current study proposes a hardware Trusted platform module in

heterogeneous data center IoT platform. All the pertinent components of a strong hardware trust are included in this trusted module.

Data Center Networking: Moderate bandwidth in Data centers is required by human interactions with applications [36]. IoT promises a dramatic alteration in these patterns through the transferring of huge amount of small message sensor data to the data center for it to be processed; this in turn brings dramatic rise in the requirements of the inbound data center bandwidth [37]. In heterogeneous Data centers, redundant controller and aggregate controller topology is not implemented [34]. The current study proposes redundant topology of controllers and aggregate controllers.

Ever since digitization and automation of the different devices installed across the modern urban environments has increased, there has been an evident creation of fresh security challenges for various industries that may or may not be regulated [35]. As already discussed, hackers can target even the smallest of devices sitting on the Internet for stealing information, falsifying data or disabling or corrupting the device. The vulnerability of Software based security is high in certain types of attacks. If a device or sensor is accessible, it can possibly be cloned or altered in behavior by loading malware.

Through this, unauthorized users can take control of the device and carry out different activities like disabling the device or providing information to hostile parties. Another type of threat is posed by replay attacks by the poor interception of secured password exchanges between host and client and reuse of the password for gaining access to the network. The hardware Trusted platform module has been proposed in the heterogeneous data center IoT platform.

Another type of threat is posed by replay attacks by the poor interception of secured password exchanges between host and client and reuse of the password for gaining access to the network.

IoT does not have strong hardware authentication. It has been recently observed that reverse engineer is possible for the various types of Physically Unclonable Functions (PUFs). The current study proposes a hardware Trusted platform module in heterogeneous data center IoT platform. All the pertinent components of a strong hardware trust are included in this trusted module.

Moderate bandwidth in data centers is required by human interactions with applications [36]. IoT promises a dramatic alteration in these patterns through the transferring of huge amount of small message sensor data to the data center for it to be processed; this in turn brings dramatic rise in the requirements of the inbound data center bandwidth [37]. In heterogeneous Data centers, redundant controller and aggregate controller topology is not implemented [34]. The current study proposes redundant topology of controllers and aggregate controllers.

The needs of the Internet of Things (IOT) can be met by making use of cloud-based architecture. In this approach, application intelligence and storage are centralized within server wire centers. However, there can be a breakdown when there are other requirements like real-time requirements associated with the control loop, the presence of large volume of data, the lack of available network bandwidth within the deployment model. This has supported the need for decentralized processing and resulted in the emergence of the 'edge computing' construct. In such an approach, the aim is not just to create an aggregation of available sensed physical data which looks similar to a gateway, but also promote distributed intelligence. To create such a platform, there is a need for deterministic and real-time processing to implement specific functionalities.

Storage, networking and computing capabilities are included in handheld devices and the endpoint devices like intelligent actuators and sensors. These are significant in intricate IoT systems. Nevertheless, there are certain security, bandwidth, power and space constraints in the endpoints. The IoT gateways as well as the intermediate layers within networking equipment function as ideal platforms for hosting IoT processing systems. These require collection of data from various sources including sensors, distributed databases and other elements. Servers in cloud or Data centers are connected by IoT systems. The risk of being blind to critical alerts is posed by the failure of connectivity to cloud. As a result of this limitation, crisis can occur in Data centers [38]. The current study proposes that application layer on the edge devices (aggregator controllers) be created with residing intricate logic of alerting. Additionally, it delivers alerts even if the close loop IoT platform network.

Following are additional IoT challenges, although they are not main concerns addressed in this thesis.

With the installation of numerous devices, significant security challenges will persist and the security complexity will drastically increase [36]. Consequently, the availability

requirements will be impacted as a result of which will be risk associated with personal safety and real-time business processes [39].

Similar to the case of increasingly digitized automobiles and smart-metering equipment, huge amount of data will be available on information pertaining to the personal use of the device by users, which if not secured properly may result in breach of privacy [25]. This is quite difficult as the information that IoT generates is crucial in bringing better services and managing devices like these [36].

IoT has a double impact on the types of data to be stored. The first is the consumer driven personal data and the second is the enterprise driven big data. As apps are used by consumers, devices gather significant knowledge about the user which generates significant data [40].

Another significant factor that adds to the increasing demand for more storage capacity is the impact of IoT on storage infrastructure. There is a need to address this factor as this data will become increasingly prevalent. Storage capacity should be the main focus now in addition to the IoT data being harvested and used by businesses in a cost-effective manner [36].

The focus of the impact that IoT has on the server market will mainly be on increased investment in major vertical industries as well as the organization's associated to the industries where IoT adds profit and a considerable value [41].

Even though there is no agreement regarding a specific and standardized architecture of IoT, a well-known architecture with three layers is commonly accepted; the first layer is the Perception Layer; the second layer is the Network Layer; and the third layer is the Application layer.

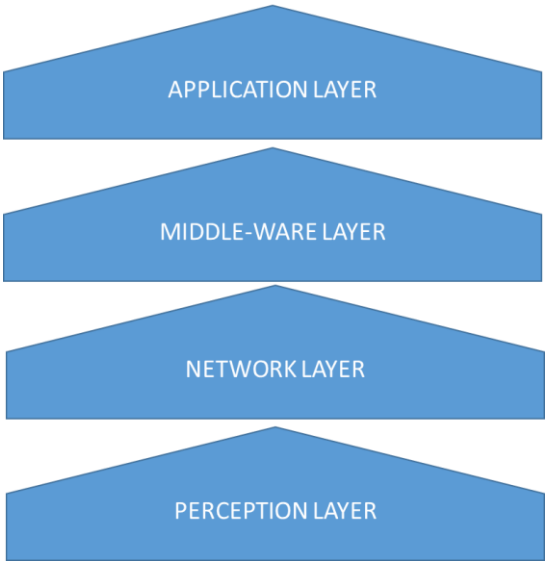


Figure 6; Key layers of IoT

2.2.1. Details for Each Layer

1. Perception Layer: This layer mainly looks at perceiving physical properties like speed, location and temperature through various sensing devices and taking this information and further converting it into digital signals that are easy to store and transmit through digital communication networks. This layer comprises of end-devices, actuators and sensors [24].
2. Network Layer: The role of this layer mainly includes the transmission of the data obtained from the perception layer to a processing center, server or a database. The technologies mainly used for realizing this layer include ZigBee, Bluetooth, Wi-Fi and cellular technologies 2G/3G/LTE (Long Term Evolution). According to [25], even though there are a wide variety of technologies available for radio access, the interconnection of all of these is possible through IPv6 at the transport layer in addition to addressing the numerous anticipated things that will connect in the future. This layer comprises of Application layer protocols, Message queues, Operating systems, Gateways, M2M (Machine-to-Machine communication network) servers and Communication protocols [25].
3. Middle-ware Layer: Included in this layer is the information processing systems that undertake automated actions on the basis of results of the processed data while linking the database and the system through storage capacity for collected data can

be obtained. Since this layer is service oriented, it makes sure that a similar service type exists between the connected devices [26].

4. The Application Layer: This is the layer where the information provided by the Network Layer is stored processed and analyzed. Through these the various end-user applications are facilitated such as safety, identity, location based services and building automation [25]. In this layer applications are provided for different types of technological challenges like controlling end device values, processing and monitoring. The Internet of Things is promoted by these applications. This layer comprises of software applications and IoT cloud platforms.

The data center can meet its goals of consistent improvement in the reliability and efficiency of mission critical systems by the acceptance of an IoT philosophy and the integration of physical building infrastructures with the various technology system infrastructures as well as internal and external people and mobile devices [42].

This implies the meshing of different systems for which meshing was not possible like the power delivery and circuits that the data center requires, liquid chiller loops, HVAC systems, uninterruptible power supplies and generators in addition to the ability of measuring consumption for each customer. In order to mesh the systems with personnel, this system was created by the data center within new IoT platform, wherein all the data was fed through a single funnel for filtering, analyzing and pushing back out like target alerts for the suitable parties [43].

(i) End Devices: These include major building systems like Automatic Transfer Switch, Transformer, Generator, Cooling Tower, CRAC, Chillers, UPS and Pumps. The other set comprises of the devices that are mechanical and electrical systems of a customer data suite. Figure 5 illustrates examples of the points.

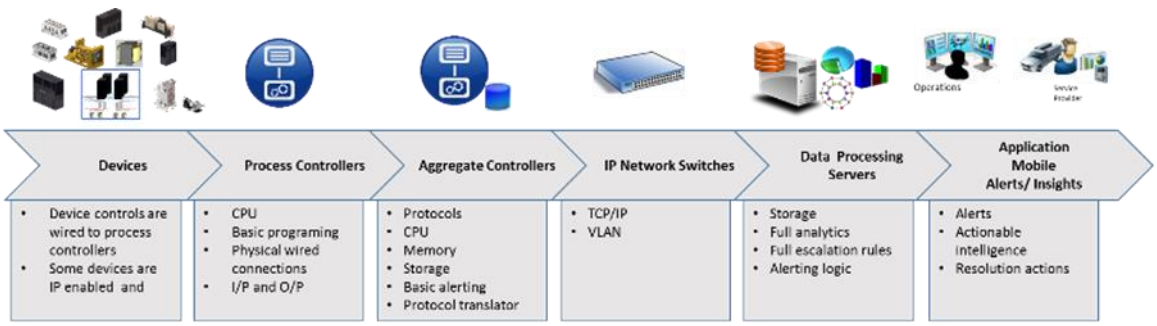


Figure 7: Variables monitored at the data suite

(ii) Process Controllers: This is the node where the physical wires connect to the end devices. The physical wiring outputs and inputs are managed at this node.

(iii) Aggregate Controllers: This node is a critical aspect in protocol translation. This is the stage where alerting logic and basic alert configuration can be programmed.

(iv) Network Switches: Firewall settings, VLAN and the TCP/IP traffic are managed by this node.

Data Processing Servers: The historical and real-time data for all measured points is stored in this server. Full analytics capability is present with this server. At this node, complex escalation rules can be implemented.

Mobile Alerts: In this node is the application receiving all the alerts in addition to the guide link on responding and resolving the issue.

Carefully crafted logic is used by the IoT platform technological components that are dependent on the minimization of false positives and feedback loops for constantly gathering and interpreting the data at Analytics Server and Aggregating Processor. This facilitates the system in learning from itself, consistently improving and reducing false alarms, which in turn increases redundancy and efficiency [44].

Another significant innovation is escalation and notification as the maintenance of speed of content and information delivery to the pertinent parties like vendors and external on-demand experts is desired by data centers. A communications platform was created in order to achieve this and it sits on top of the IoT alerting system and offers a push networking capability that is based on mobile apps, voice communications and e-mail. In this environment the people who have to provide a solution are automatically notified of the problem alert simultaneous to the service organization that initiates a conference call and a proactive response in obtained in minutes [44].

2.3. Traditional approaches to power utilization management

Over the past few years, there has been a great focus on making data centers energy efficient. Aisle containment has been used in an effort to optimize facilities [45]. Work has also been done on managing virtual server loads so that energy can be utilized efficiently [46]. Additionally, work has been done in energy management through the combination of virtualization and automation being built together [47]. New demands are emerging in relation to infrastructure and data power efficiency, as well as cloud computing. Moreover, more users are driving this change in data centers along with more data and even more dependence on the data center.

It has become more than important now to work with the apt data center optimization technologies as rapidly growing data and cloud technologies are leading the way in various technological categories. Improvement can be observed in any amount of gain in energy gains [8]. Lately machine learning techniques have been utilized by Google and Microsoft for energy optimization. Machine learning techniques are being explored by Google for energy optimization with data centers being used at a building level [48]. Microsoft is automating data center operations and measuring server workload spikes [49].

The current study is associated with data pods and suites in a multi-tenant facility that have heterogeneous server configurations [Figure 3]. This study aims at further optimizing micro facility environment associated with a data suite for a specific server load.

2.3.1. Optimizing Airflow and Temperature in DC using ANN

Reduced order models are significant in controlling energy usage in data center rooms so that the optimum operating conditions can be assessed in real-time and energy usage can be reduced. Here computational fluid dynamics (CFD) simulation-based Artificial Neural Network (ANN) models were developed and their application was done in cold aisle/ hot aisle data center configuration so that thermal operating conditions regarding a particular set of control variable can be predicted [50].

After training, the agreeable ANN-based model predictions were obtained in terms of the CFD results when the input variable had arbitrary values in specified limits. Additionally, a cost function based multi-objective Genetic Algorithm (GA) was combined with the ANN

model through which the inverse predictions for operating conditions could be made for a particular value of the output variable, for instance, server rack inlet temperatures.

In comparison to a fully CFD-based response surface optimization methodology, the total computation time is considerably reduced with ANN-GA optimization approach. As a result, operating conditions can possibly be predicted reliably in seconds, even in the case of configurations that are beyond the original ANN training set. It can be observed from these results that effective real-time thermal management design tool for data centers can be obtained from ANN based model.

2.3.2. Machine Learning Based Pre fetch Optimization for Data Center Applications

Even though performance tuning is complicated, it is essential for data centers. It is significant as there are a thousand of machines in a data center and power and cost reduction can be significantly reduced with a single-digit performance improvement. However, it is quite difficult due to the dynamic environments of the data centers, where application release is quite frequent and there is consistent up gradation of the servers [51]. The current study focuses on the various processor prefect configurations, in terms of their effectiveness, which can have a great influence on the performance of the overall data center including the memory system. A wide comparison gap is observed when the best configurations are compared ranging from 1.4% to 75.1%, for about 11 significant data center applications [52]. Following this, a tuning framework was developed that aims at predicting the optimal configuration on the basis of hardware performance counters. Performance can be achieved by the 1% of the best performance of any single configuration with identical application set [52].

2.3.3. Neural Switch using Open Flow as Load Balancing Method in Data Center

The bottlenecks caused by the increasing traffic in data centers are having an adverse effect in the performance, which leads to packet loss. Thus, in comparison to a single route option, multipath routes are rather better. Increase in the performance of data centers as well as low management cost can be achieved by using multipath to route traffic. Since the data

centers have similar information traffic, it is possible to implement Neural network inside switch/router through which routing can be composed between the destination and origin. Optimized routing as well as dynamic load balancing, devoid of SDN controller intervention, can be achieved from multipath routing with neural network [53].

2.3.4. Machine learning based Approaches for Power Utilization Management

The main aim is to use a utility function to the maximum. For instance, the given classifications may not be best suited for all classifications. One remarkable example is that misplaced use of the traditional knowledge of the game of backgammon when through unsupervised learning a series of computer programs (neuro-gammon and TD-gammon) learnt the game and became stronger than the best of human chess players by repeatedly playing with them. Certain principles were discovered by these programs that surprised the backgammon experts and gave a better performance that the backgammon programs who were given prior training on pre-classified examples [54].

Clustering is another type of unsupervised learning, which finds parallels in the training data. The basic assumption is that the clusters discovered will be a reasonable match with the intuitive classification. For example, clustering individuals on the basis of demographics may lead to the clustering of the poor ones on one group and the wealthy ones in the other. Inopportunely, the problem of over fitting the training data persists even in unsupervised learning. This problem cannot be avoided as it is necessary for any algorithm to be powerful to learn from its inputs [55].

Machine Learning and Data Mining is associated with gathering knowledge from data. Generally, this includes the creation of models or the discovery of patterns in examples from the prior aspects of system behavior with the minimum possible expert intervention [56]. In the current study, machine learning techniques have been used so that the resulting level of client satisfaction regarding the job and power consumption can be predicted, using the given set of jobs and machines, before the tasks are placed in machines and are moved across machines.

A move selection algorithm further uses these predictions for choosing destination machines that will produce good opportunities for consolidation and client satisfaction.

Suitable predictor algorithms need to be selected for this prediction process; these should be computationally light but with the ability of obtaining good results after training with data from different workloads. Additionally, a good validation set or training set is required; a training set includes data containing marked instances from archetypical executions [54].

If after training, the guesses of the predictors are near to the correct values obtained in the test set, the same is expected to be true for future real workloads. A Dynamic Backfilling scheduler is implemented by the machine learning aided policy, which then replaces the static decision maker. This is done with the use of the information that the user directly provides and the results of the power consumption and performance estimators as decision makers [57]. This implies that rather than directly fitting jobs in the host machines, the impact of the job in the potential host machine has been estimated in terms of power consumption and performance parameters.

In terms of Dynamic Backfilling, in each reschedule an attempt is made to empty low-used host machines so that nearly fully-booked ones can be fulfilled. Then, estimates are made if each movement will interrupt the resource requirements of all the different jobs in the machine. Additionally, estimates are also made regarding the new power consumption that the machine will have to compensate for any possible performance degradation. This allows for the obtaining of a rather strong and adaptive system where application specifications can be dynamic or imprecise [57].

Currently a negligible operation cost has been assumed but for any future work different factors including moving machine cost have been taken into consideration. Additionally, Dynamic Backfilling algorithm is rather costly, particularly when data collection processes are used. Thus, the use of reinforcement learning techniques and AI planning are being used so that decisions can be made in a cost effective yet accurate way.

2.4. Traditional approaches to Predictive maintenance

Several works on condition-based maintenance of repairable systems is applicable to statistically independent failure modes or a single failure mode. Apart from these works, the current study considers the challenge of predictive maintenance of repairable systems with resource constraints and dependent failure modes [58].

Assume that (i)two statistically dependent failure modes are caused by a repairable system, which affect each other bi-directionally, (ii) insufficient resources spent for maintenance are allocated to the two dependent failure modes so that the imperfect maintenance actions can be cooperatively executed, and (iii)future maintenance planned at the current time is dependent on the minimization of the projected maintenance cost rate determined in the long term and the projected number of future failures [59].

A new cooperative predictive maintenance model is suggested for resolving the above problem, which is based on the incorporation of effective age and hazard-rate function [60]. This model shows the statistic dependence of two failure modes in a manner that the hazard rate of one failure mode is dependent on the failures of the other failure mode collectively. The effect that imperfect maintenance has can be construed by how maintenance actions change the effective age and the hazard rate function. For each failure mode, the maintenance induced age reduction factor is deterministically associated to the degrees of resources allocated cooperatively for performing maintenance [59].

With the arrival of new monitored information, the decision variables in the maintenance policy can be recursively updated; these variables include the compliantly distributed degree of resources, the interlude between consecutive maintenance actions, and the number of maintenance actions that have to be performed. This approach is dependent on the minimization of the projected maintenance cost rate determined in the long term and the projected number of future failures [58].

As a result of the demanding cost, small scale and medium scale industries are unable to access majority of the predictive maintenance technologies. In the current study, a predictive maintenance policy has been proposed with the use of non-homogeneous Poisson process (NHPP) and failure mode effect and criticality analysis (FMECA) models which have a minimal requirement of sophisticated data acquisition systems and advanced monitoring technologies.

Long term reliability degradation in addition to recurrent overhauls is shown by majority of the repairable systems. The overall maintenance time of a system is predicted through the critical component of a system or machinery that exhibits a sad (deteriorating) trend. Firstly, the FMECA method is used in selecting the element to be used as an indicator for predictive maintenance; here the most critical component is selected [58].

Secondly, NHPP models are used for analyzing the failure data of the selected component; the relevant NHPP model is selected on the basis of the data analysis. In the end, the overall maintenance time for the system is decided by comparing the Mean Time Between Failure (MTBF) of the component and the threshold mean time between failure [MTBF(THz)] of the component. An overhead crane in a steel manufacturing company is used for validating the developed methodology [58].

The current study overviews the two maintenance techniques extensively discoursed in the literature; the first is the time-based maintenance (TBM) and the second is condition-based maintenance (CBM). The current study elucidates the working of TBM and CBM techniques for making decisions pertaining to maintenance. The study reviews recent research articles pertaining to application of these techniques. Then, the challenges of implementing each technique are compared from a practical point of view with a focus on the issues of decision making, data analysis/modeling, and data determination and collection. Additionally, the study presents considerable aspects for future research. In terms of industrial practice, each technique presented unique challenges, procedures and concepts/principles. It can thus be concluded that it is more realistic and valuable to apply the CBM technique. Nevertheless, it is imperative that further research on CBM be carried out so that it can be made more realistic for making decisions regarding maintenance. The current study provides useful information pertaining to the TBM and CBM application for the maintenance of decision making and exploring the practical challenges associated with the implementation of each technique.

2.4.1. Disadvantages of predictive maintenance

Compared to preventative maintenance, it is often that high cost is linked to the condition monitoring equipment that is necessary for predictive maintenance. Also, it is imperative to have wide experience and skill level for accurately interpreting the condition monitoring data [60]. Jointly these entail raised upfront cost of condition monitoring. Certain companies engage condition monitoring contractors in order to minimize the upfront costs of a condition monitoring program [61].

The criticality of the facility side infrastructure of data center is higher than ever. Downtime may lead to high penalty cost for midsize data center operators. With this mission critical nature of data center, a 100% uptime has been demanded from the data center operators.

Included in the facility infrastructures are mechanical and electrical systems interacting with each other intricately. With this, a 100% delivery of the service level agreements (SLA) is required from the facility side infrastructure maintenance. Several measures have been taken for performing preventative maintenance (PM).

These refer to standard checklist implemented at specific frequency annually or quarterly. The aim here is to keep a close check on upkeep so that high cost emergency maintenance repairs can be avoided. With the advent of IoT, instrumenting machine to gather operational data became easier. Remote machine sensors can be leveraged to consistently monitor machine behavior. These large amount of data can be used with machine learning technique.

2.4.2. Black Box Model of a Data center as a Temperature Predicting Tool

In order to achieve this, several approaches are being developed. For instance, control systems can be developed to run data centers in an energy efficient manner. The development of different theories has been undertaken across the world so that an optimal energy efficient state can be achieved [62].

However, while such control schemes are being synthesized, excessive time is being taken by the CFD simulations for plotting the map of such highly linear, dynamic and complex data center systems. The current study aims at developing and training artificial neural networks for a classic scaled setup of contemporary data center such as a Black Box Model through which the temperature at various points in the state space can be predicted across the room. This is a function of the CRAC fan speeds and the dissipating heat at those points at a given time [62]. Since the analysis time is significantly low in computational fluid dynamics, it is possible for the Black Box to predict temperatures at different points in the setup and in real time which makes optimization analysis faster. A massive set of data generated by CFD simulations through theoretical arrangements in a data center is used for training the neural model [63]. The current study also discusses the neural network training functions along with its training parameters. The study also makes comparisons of accuracy and computational time, and its justification.

The study also summarizes different suggestions for training dynamic and highly linear systems. Comparison is carried out between the CFD model output and the data generated so that the accuracy of the neural network can be predicted. The verification of the strength of the Black Bock in the training data limits has been illustrated through changing server heat and CRAC fan speed. In addition to being accurate, the Black Box tool is also very fast and enables its use in a dynamic learning setup or in a feed forward adaptive control setup or both. It is rather useful for the development of control systems for data centers to have a Black Box tool mimicking the CFD [63], used to learn complex interactions and predictively identify issues.

2.5. Iterative Optimization for the Data Center

Even though it is simple, iterative optimization is a power approach that seeks out the most suitable potential combination of compiler optimization for a particular workload. Consequently, several practical issues emerge in iterative optimization that restrict its wide use such as the significant overhead generated by the exploration process that the performance benefits have to compensate, the data set dependent nature of the process; and the large number of runs required for the identification of the best combination [51].

Thus, even though significant performance potential has been shown by iterative optimization, production compilers seldom use it [51]. The current study proposes Iterative Optimization for the Data Center (IODC), wherein context offered by servers and data centers is illustrated through which the challenges listed can be handled. The general idea is the generation of different combinations through workers in addition to the recollection of performance statistics at the master, through which the optimum combination of compiler optimizations is evolved.

IODC is evaluated using throughput compute-intensive server as well as Map Reduce applications. A large collection of datasets, ranging between 1000 to several million unique data sets per program, is gathered so that the large number of users interacting with the system can be reflected [64]. This was done for 568 days of CPU time for a total storage of 10.7TB. The throughput compute-intensive server applications reported an average performance improvement of 1.14× to 1.39×, while MapReduce applications reported an average performance improvement of 1.48×, and up to 2.08× [64].

2.5.1. Random Neural Network for Load Balancing in Data centers

When thousands of connected computer servers form a data center, it can be considered as a resource of processing capacity (CPU), disk space or memory. Different paths are used for distributing the jobs that arrive at the cloud data center to the different servers. Additionally, the internal traffic that can be found between servers inside the data center also has to be load balanced to various paths between them [65].

It is quite challenging to select the idle or underutilized paths for the traffic so that throughput optimality and load balancing can be achieved. The Random Neural Network (RNN) is a recurrent neural network, wherein neurons interact amongst themselves through the exchange of inhibitory and excitatory spiking skills. The RNN has proven to be an exceptional modeling tool for the different interacting entities due to the stochastic inhibitory and excitatory interactions in the network [51].

The same has been applied to several applications like classification, simulation pattern recognition, communication systems and optimization. It is proposed in the current study that Random Neural Network (RNN) be used for solving the issue of load balancing in data centers. Adaptive load balancing is achieved by RNN through online measurement of the path congestion that the network gathers [51].

2.5.2. SLA-based virtual machine management for heterogeneous workloads in a cloud data center

In cloud computing the efficient provisioning of resources is challenging because of its dynamic nature as well as the requirement for supporting heterogeneous applications. Although the concurrent running of workloads and the shared use of infrastructure is permitted by VM (Virtual Machine) technology, application performance is still not guaranteed by it [66]. Thus, at present, either performance guarantee is not offered by cloud data center providers or they make static VM allocation rather than dynamic; the result of which is inefficient resource utilization.

Furthermore, there may be different QoS (Quality of Service) requirements for the workload because of the execution of various types of applications like web and HPC, as a

result of which resource provisioning becomes even harder. Earlier studies have focused on resource usage patterns of applications like web applications, or on single type of SLAs (Service Level Agreements) due to which the data center resources have been utilized inefficiently [66]. The current study focuses on the resource allocation problem inside a data center wherein various application workloads are being executed, especially the transactional and non-interactive applications.

The current study suggests an admission control and scheduling mechanism, wherein the profit and resource utilization is maximized in addition to ensuring that QoS requirements of users are taken care of in accordance with the SLAs. In the current experimental study, the awareness of different SLAs was found to be important in addition to the mix of workloads and applicable penalties for better provisioning and utilization of the data center resources. Through the proposed mechanism, substantial improvement can be obtained over the static server consolidation in addition to reduction in SLA violations.

2.5.3. Towards energy-aware scheduling in data centers using machine learning

Since data centers and IT infrastructures identify energy-related costs as a major economical factor, it has become challenging for the research community and companies to find more efficient and better power-aware resource management strategies. “Green” IT is gaining growing interest, however, there still is a big gap that needs to be covered [57]. The current study proposes framework to obtaining an energy-efficient data center, wherein an intelligent consolidation methodology is provided that uses different techniques like machine learning techniques, power-aware consolidation algorithms, and turning on/off machines so that uncertain information can be handled while performance is maximized.

Models provided by previous system behaviors have been used for the machine learning approach so that scheduling decisions can be improved and predictions can be made pertaining to SLA timings, CPU loads and power consumption levels. This framework vertically encompasses cross-disciplinary, workload features and watt consumption [57]. These techniques have been evaluated with a framework wherein the entire control cycle of a real scenario is covered through the use of simulation with representative heterogeneous workloads. Additionally, the quality of the results has been measured according to a set of

metrics that are focused on traditional policies as well as goals. It is indicated by the results obtained that this approach is close to the optimal placement and works rather better when uncertainty levels increase [57].

2.6. Predictive Modeling and Simulation

Predictive modeling can be described as the process of the creation, evaluation and validation of a model so that the probability of an outcome can be best predicted. For this task the predictive analytics software solutions use many modeling methods such as statistics, artificial intelligence and machine learning. The detection theory is used for the testing, validation and evaluation of the model so that the probability of an outcome in a specific amount of input data can be guessed and the best model can be selected [67].

More than one classifiers can be used by managers for determining the probability of a set of data that belongs to another set. New information regarding the data as well as the development of the predictive model is facilitated through the modeling portfolio of predictive analytics software [68]. Every model is best suited to a specific types of problems depending on its strengths and weaknesses.

A model can be reused and is developed by training an algorithm through the use of historical data and saving the model so that it can be reused for sharing the common business rules that can are applicable to similar data so that the results can be analyzed through the use of the trained algorithm, without the historical data [68].

Simulation can be defined as the imitation of a real-world system or process operations over time. The development of a model is the initial requirement for the act of simulation. This model illustrates the main behaviors/functions or characteristics of the selected abstract or physical system or process. The model is the representation of the system itself, while the simulation is a representation of the system operation over time [69].

There are several contexts in which simulation can be used, like video games, education, training, testing, and safety engineering and performance optimization. Simulation models are often studied using computer experiments. Additionally, simulations are used with human systems or scientific or natural systems for getting details about their functions. The use of simulations can be done for illustrating the eventual effects of alternative courses of action and conditions. Additionally, the use of simulation has been found in the situation where a real

system cannot be engaged, or is not accessible, or maybe unacceptable or dangerous or may simply not exist [69].

The major challenges of simulations include validity and fidelity of the simulation results, the use of simplifying assumptions and approximations in the simulation and information regarding the selection of the main behaviors and characteristics. The protocols and procedures for model verification and validation is a continuing field of research and development, refinement and academic study in the simulations practice or technology, especially in the area of computer simulation [70].

2.6.1. Controller Simulation

The design of controller is such that it augments the quality of the control system so that the required control purpose can be served by the system. After the selection of the controlled object, the property and quality of the entire control system is dependent on the design of the controller. Thus, the entire control field is focused on the design of the controller and its analysis.

The control method that is used most commonly is the digital PID control, which is widely used in chemical, machinery, and metallurgy industry among others. In the simulated control system, PID controller refers to proportion of deviation (P), integral (I) and differential (D), which is the most widely used automatic controller.

The characteristics of PID controller include: simplicity of principle, ease of achievement, basic controller catering to most of the actual needs; the application of the controller is possible for various objects. The structural strength of the algorithm is string; and in many cases, the sensitivity of control quality is not high to the parameter and structure distresses of controlled object [71].

Nevertheless, the major disadvantage of PID control is its reliance on the controlled object in addition to the general prerequisite of knowing the mathematical model of the controlled object design. The characteristics of the controlled object like time variability and non-linear in practical industrial control make the establishment of accurate mathematical model difficult. Else, its application is limited due to the difficult online availability of the characteristic parameters [72].

Neural networks achieved the shortcomings of the PID controller for solving challenges in almost all the areas of technology and science. There are mainly two steps involved in neural network control: system identification and control.

In the neural network(NN) Feedback control output signals from a dynamical system or plant are measured and the difference between certain prescribed desired values and measured values are used for computing system inputs causing the measured values to track or follow the anticipated values. In feedback control design it is imperative that the boundedness or stability of all variables and the tracking performance of all variables be guaranteed. If this is not guaranteed, serious problems may be caused in the closed-loop system, such as unboundedness and instability of signals resulting in the destruction or failure of the system.

Werbos [73] and Narendra [74] first proposed the use of NN in control systems. The two major thrusts of NN control include NN in closed-loop feedback control and Approximate Dynamic Programming, wherein NN is used for approximately solving the optimal control problem.

Narendra and Parthasarathy [75] presented the various NN feedback control topologies. Some of these are derivative of standard topologies in adaptive control. Control signal flow loops are denoted by solid lines while tuning loops are denoted by dashed lines.

Feedback control topologies are mainly of two types; the first is direct techniques and the second is indirect techniques. In direct control the parameters of an adjustable NN controller are directly tuned and it is more effective. Indirect NN control has two functions: an identifier block wherein the NN is tuned so that the dynamics of the unknown plant can be learnt; and the controller block wherein information is used for controlling the plant.

The limitation of the use of NN for feedback control purposes is the selection of a suitable control system structure in addition to the demonstrating the way NN weights can be tuned with the use of mathematically acceptable techniques so that the performance and stability of closed-loop can be guaranteed. This article illustrates the different methods of NN controller design through which performance for systems with varying complexity and structure can be guaranteed. Several researchers have contributed to the development of the theoretical foundation for NN in control applications [76].

The following methods can be used for the implementation scenario of the neural network based controller

1. Normal Loop for PUE optimization:

Real time inputs that the data center sensor ports send to neural network trained controller are used. This produces and optimizes values that represent the need of a manual intervention in the system.

2. Feedback Fan Power loop for PUE optimization

Real time inputs from data center, which is an input from the feedback result from the model output is used for optimizing only the Fan Power. As a result of this, an automated regulated fan power is formed to the neural network so that an optimized PUE can be obtained.

3. Feedback Chiller Power & Fan Power loop for PUE optimization

Real time inputs from data center, which is an input from the feedback result from the model output is used for optimizing only the Chiller Power. As a result of this, an automated regulated Chiller Power is formed to the neural network so that an optimized PUE can be obtained.

2.6.2. Data Center Simulation

Up until now, the current study seems to be the first work that provides an exhaustive methodology for data center-level simulation. While numerous works have been leveraged in statistics, stochastic modeling and queuing theory, this is referred to as in-line. Attempts have been made in previous studies for parallelizing discrete-event simulations through the simultaneous execution of the various sections of the modeled system [77]. In general, parallelization like this is difficult as a consistent state is required by the system in addition to explicit locking and/or communication of data structures.

The parallelization strategy on the other hand distributes generation of autonomous observations for the sampled output metrics, without any synchronization, which in turn reduces communication overhead and design complexity. Alternatively, hierarchical models have been used by studies for representing data center systems [78]. The use of such models can be done in place of simulators or for complimenting them.

Lastly, the current work is quite similar to the architectural simulators sampling techniques [79] and/or statistical simulation [80] is used. Additionally, these methods provide considerable reduction in simulation time by only simulating the events that are essential for the desired statistical confidence level or simulating with a statistical abstraction.

Chapter 3: Methodology

3.1. Introduction

The modern DC has a wide variety of mechanical and electrical equipment, along with their associated set points and control schemes. Machine learning is well-suited for the DC environment given the complexity of plant operations and the abundance of existing monitoring data. With the rise of IoT, adaptation of controls to collect machine data has been gaining popularity. Intelligence derived from this helps in predictive maintenance and auditing of existing systems operations to identify and resolve issues proactively.

In this chapter, we will discuss the following:

- Modular Data Center Design
- Detailed design and implementation of an IoT platform in a midsize data center to collect and store data
- Characteristics of all the data points collected
- Data Pre-processing methods and variables selection criteria
- Selection of machine learning, training and testing its prediction accuracy
- Selection of variables to optimizing cooling power and using it to predictively maintain issues.

3.2. Modular data center

In this section we explain the modular data center setup in a midsize data center that is used to collect the data. The following schematic shows the modular data center setup in a midsize data center with aisle containment.

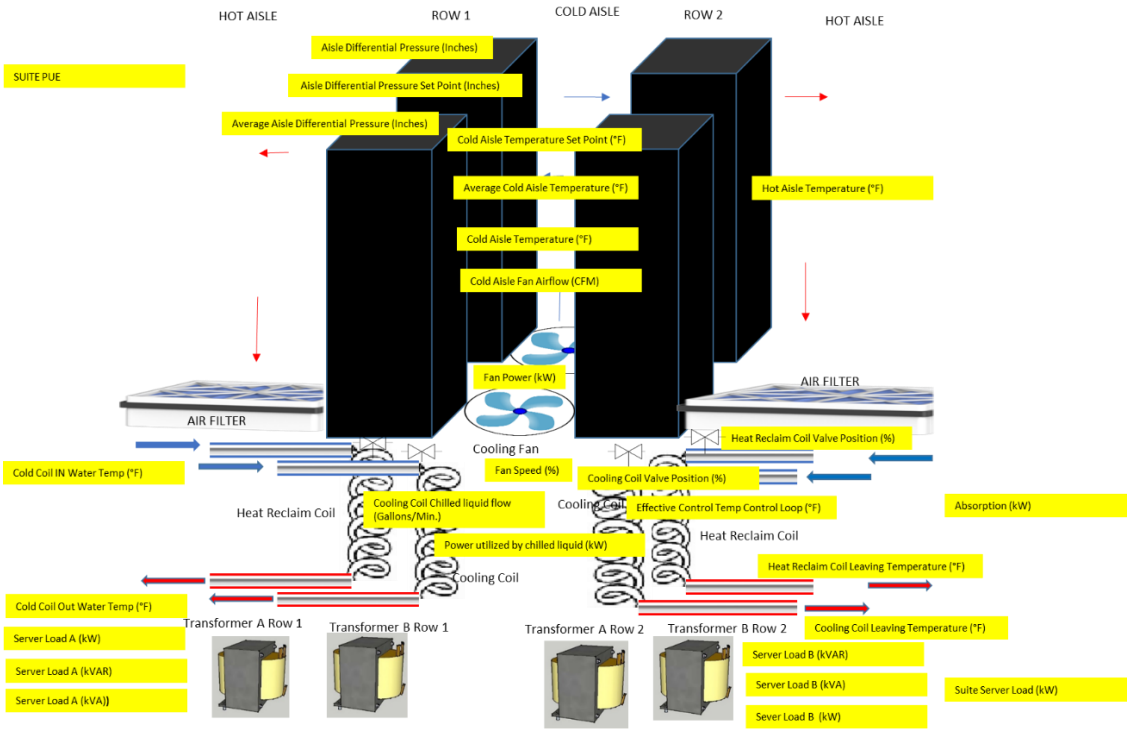


Figure 8: Two-dimensional heterogeneous DC experimental setup design

This is a fractal design with a common cold aisle and dedicated hot aisles. Each row contains 10 server racks.

- Thermally and electrically with min 300 watts incrementally upgradeable to 600 watts per sq. foot without disrupting live load.
- Redundant cooling coil and fans. The cooling coil are in series and can operated in two modes. Mode one is heat reclamation and mode two is fully redundant.
- Closed loop contained environment
- Redundant overhead bus bar distribution
- Redundant controls and instrumentation

3.3. Characterizing the Data Center Senor Ports Dataset

In this section we detail the description of data attributes that are collected. The frequency of the data collected is at the 10 sec interval using the designed IoT infrastructure**.**

- Average Cold Aisle Temperature: Average of two temperature thermostats. The average is inlet to the servers measured at 5 feet above the finished floor

- Cooling Coil Leaving Temperature: Average sensor temperature at the discharge of each coil
- Cooling Coil Valve Position: Opening percentage from 0 to 100%. The opening is proportional to the amount chiller water that flows through the coil.
- Fan Speed: Percent of max fan rpm from 0 to 100.
- Hot Aisle Temperature: Average of two temperature thermostats. The average is outlet of the servers measured at 5 feet above the finished floor.
- Heat Reclaim Coil Leaving Temperature: Average sensor temperature at the discharge of each coil.
- Cooling Coil Chilled liquid flow: The amount chilled water moving through the coil expresses in gallons/min.
- Coil Energy Absorption: The amount of thermal energy extracted from data suite in kW.
- Fan Power: The power consumed by the fan in kW.
- Power utilized by chilled liquid: The amount of power in kW to generate the chilled liquid by the central plant.
- Cold Coil IN Water Temp: Temperature in degree Fahrenheit of the chiller water entering the coil.
- Cold Coil Out Water Temp: Temperature in degree Fahrenheit of the chiller water leaving the coil.
- Server Load A: Power consumed by servers in kW for side A of dual corded servers.
- Server Load B: Power consumed by servers in kW for side B of dual corded servers.
- Suite Server Load: Total power consumed by the suite side A plus side B.

3.4. Specific Design Methods of the IoT Platform

In this section, we will discuss design methods used for the IOT platform in the heterogeneous data center. First, we explain the flexible controller configuration and multiple protocol design and implementation to address the challenges of connecting all systems in a data center ecosystem. Second, we discuss the hardware design method and logic for trusted sensor module. Third, we discuss the controller network resiliency architecture. Finally, we describe the alerting architecture on edge device of the IoT platform.

3.4.1. Design Goals and Key Features

In this section, we will discuss design methods used for the IOT platform in the heterogeneous data center. First, we explain the flexible controller configuration and multiple protocol design and implementation to address the challenges of connecting all systems in a data center ecosystem. Second, we discuss the hardware design method and logic for trusted sensor module. Third, we discuss the controller network resiliency architecture. Finally, we describe the alerting architecture on edge device of the IoT platform.

1. Flexible controller configuration:
    a. Design goals:
        i. Flexibility: The controller should have flexible configuration (logic and protocol) to setup/connect a particular end device in the heterogeneous data center ecosystem. The controller needs to be vendor neutral to support heterogeneous end devices manufactured by different manufactures.
        ii. Reliability: The configuration code should be reliable and stable when running with end devices in the heterogeneous data center ecosystem.
        iii. Modifiability: Packaged configuration should be modifiable as new end devices are introduced to the market that cater to the datacenter ecosystem.
    b. Key features:
        i. Ability for developers to create packaged configuration to be deployed on the controller hardware.
        ii. Ability for the user at the data center to pick from configuration (logic & protocol) to setup communication with each end device.
    c. Challenges:
        i. Working with various manufactures to build drivers/logic to all end devices in the heterogonous datacenter ecosystem.
        ii. Testing with live systems for reliability of code to be production ready.

iii. Inclusion of all connectivity logic and protocol into a single configurable controller that could be configured easily by the end user to deploy for each end device in a heterogeneous data center.

2. Trusted Sensors:
    a. Design goals:
        i. Security: The sensor should be secure, it avoids any non-authorized physical swap of sensors and remote terminal unit (RTU).
        ii. Fault Tolerance: Any swap of the sensors will be alerted and the new sensor/RTU will not work if the hardware trust is not compatible.
    b. Key features:
        i. Incorporation of Trusted Platform Module (TPM) circuitry for hardware authentication on the sensor /RTU at the end device and the controller it is connected to.
        ii. Selecting the trusted module logic used in the microcontroller.
    c. Challenges:
        i. Testing with live systems while in operation.
        ii. Selection and testing of trusted module logic.

3. Controller Network Resiliency
    a. Design goals:
        i. Fault Tolerance: ability to accurately switching to another controller when the active controller fails.
        ii. High availability: Ability to provide 100% controller uptime via the redundant configuration.
    b. Key features:
        i. Redundant controller that is directly connected to the end device
        ii. Redundant aggregate controller that is connected to other controllers.
    c. Challenges:
        i. Several iterations of creating a fault tolerant handshake.
        ii. Limited networking capability of controllers.

4. Controller Alerting at the Edge
    a. Design goals:
        i. High availability: Ability to provide 100% uptime in providing alerting and real time monitoring using an edge device like an aggregate controller.
        ii. Fault Tolerance: Ability for the edge device to be fully functional even when connection is lost to the datacenter/cloud servers that provide alerting and analytics.
    b. Key features:
        i. Alerting logic built into the aggregate controllers that supports all end devices in the heterogeneous data center ecosystem.
        ii. Creation of alerting and escalation logic for alerting to the right resources using a mobile application
    c. Challenges:
        i. Creation of process to incorporate external resources as the first response to a critical operational issue.
        ii. High resource consumption by the edge device (aggregate controller) due to more work load.

3.4.2. Flexible Controller Configuration

As discussed in previous chapters, there is no single IoT platform deployed in controllers with all configurations that will support full ecosystem of a heterogeneous data center ecosystem. We discuss the design methodology to create an IoT platform to select from flexible configuration to support end devices for the heterogeneous data center ecosystem. Currently, there are no standards in implementing an IoT platform that includes a single “standard” network protocol, control programs, distributed architecture or field bus. The IoT software framework is designed from ground up to adopt that there will never be one standard in the immediate future for deploying an IoT Platform in a heterogeneous data center. As shown in Figure 9, the software framework core components consist of the core operating system, middle ware applications, device interface protocols, external API and user interface.

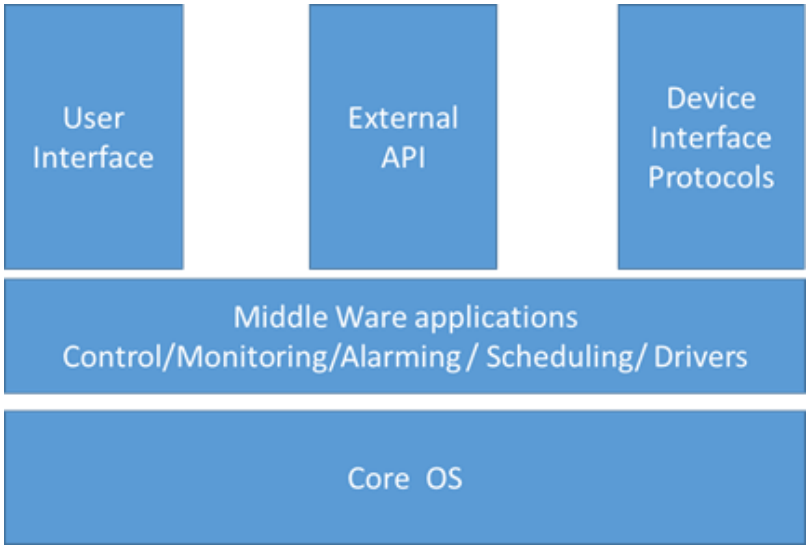


Figure 9: Software framework for the IoT platform

(i) User interface: This consists of graphical interface for reporting and programming for users and developers. This can be a client application or a web browser. This includes all the toolkits required for programming.

(ii) External API: This includes all the interfaces to communicate to databases including oracle, DB2, SQL, mail servers, security attenuation software etc.

(iii) Device interface protocols: This supports interfaces to communicate with multiple protocols like BACnet, Modbus, LonWorks etc. In the next section, we will discuss the detailed list protocol implementation.

(iv) Middle ware application: In this layer all the logic and configuration reside for controlling, monitoring, alarming, scheduling. This is the layer all the programming configuration resides that allows communication legacy systems. All the logic that supports the end devices in a heterogeneous datacenter resides in this layer. We will describe the detailed workflow of creating and deploying flexible configuration application to connect end devices for the heterogeneous datacenter ecosystem.

1. **Flexible controller logic creation method:**

In this section we will explain the workflow deploying configurations on controllers. All the configurations required to connect end devices in a data center are preinstalled in the controller to pick from. In this step we write logic for monitoring and controlling all devices in the heterogeneous data center ecosystem. This is the part of the standard controller that can be

configured to connect to any remote terminal unit (RTU), current transformers (CT's) and other sensors in the data center ecosystem.

.

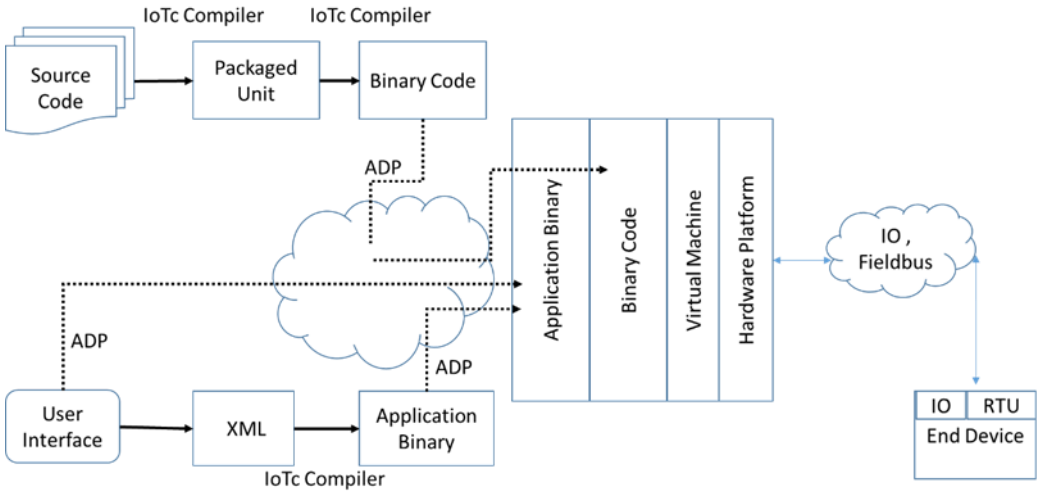


Figure 10: Work flow for creating and implementing the IoT framework to controllers and aggregate controllers

As shown in Figure 10 source code block is a programming language used for writing the IoT Framework applications. Java and C# are the basis for the programming language in this implementation. The implementation uses an object oriented programming model inclusive of inheritance that permits a class or object to be the basis of another class or object. Additionally, it uses polymorphism that is able to process objects differently based on their classes or data types.

Similar to C#, Java or C++ the software is arranged into classes. Further these classes are arranged into a packaged unit. Also, a Packaged Unit is the basic unit of distribution, versioning, and naming in the IoT framework. Normally, a Packaged Unit is like a .NET DLL or like a JAVA JAR file.

The storage of Packaged Units is done as a single file. The file in itself is a standard archive file that can be opened with the use of any 'zip' tool. There is an XML manifest that specifies meta-data regarding the Packaged Unit like description, vendor, version and name. All the component types of the Packaged Unit available for constructing applications are also enumerated in the manifest.

The compilation of Packaged Units is done from IoT language source code with the use of IoTc compiler. In the process of compilation, validity is checked for the various classes in

the Packaged Unit and then for transitional representation they are represented into a special format. The format of transitional representation is a text based "assembly language". The nature of this code is portable, which implies that it can be applied for all formats.

The format of transitional representation is not ideal for machine execution. Thus, the Packaged Unites are further compiled in a single file known as Binary Code image. Binary code is a rather compact binary depiction of the code designed such that it can be directly executed by the IoT virtual machine.

A set of Packaged Units compiled into a Binary Code with the use of IoTc compiler. The compilation process involves layout of method code, fields and reflection meta-data in memory. Then it entails optimization for big-endian or little-endian and native platform pointer size. Finally, link method calls to their memory locations.

In order to maximize performance, the Binary Code is optimized by the IoTc for the requirement of a specific platform. This implies that it is not possible that the Binary Code image be ported to other platforms.

After the compilation of a set of Packaged Units is done into a Binary Code file, it runs on the IoT Virtual Machine (IoTVM). The Binary Code is interpreted by the IoTVM and IoT programs are brought to the full functional state. The language used for IoTVM is ANSI C and it is compiled for a target controller platform.

Graphical programming tools are used for the creation of application binary so that applications can be designed using assembling and linking component instances and packaged units. The model of an IoT framework is like a tree of components. The use of links is done for establishing relationships between components for specifying data and event flow in the application.

The ultimate goal of the IoT Framework is the creation of programmable smart devices for providing support to end devices in a heterogeneous data center ecosystem. Everything comes together at this stage. Typically, a process of IoT Framework-enabling a device includes Transferring the IoTVM SVM to the designated controller, connecting to the end device. Then run the IoT VM. Finally, commission the controller with binary code image and load the application binary file.

As illustrated in Figure 10, the IoT framework uses the Authenticated Datagram Protocol (ADP) for communication. Host secure channel establishment protocol is hosted by ADP and

in the network hierarchy it is handled at the sub transport level. Discussed below is the operation of ADP between two hosts H1 and H2. An ADP channel is established by the two hosts with the use of a Public key encryption PKE-to- Single Key encryption SKE bootstrapped protocol. Agent-to-agent channels are built by H1 and H2 on their ADP channel by sending PKE-based certificates of their respective agents to each other. The PKE-based certificates sent and received are cached by the two hosts, which in turn reduce the overhead of building agent-to-agent channels over their ADP [81].

2. **Flexible protocol configuration**

Several protocols are bundled with the IoT framework to provision, program, and communicate with IoT framework enabled devices over various network topologies. As the IoT is maturing, there is no one standard protocol. The IoT platform is designed to handle all protocols required by the data center ecosystem. Additional protocols can be added to the IoT framework as needed. Following are the list of protocols included in the IoT platform to support current needs of the heterogeneous data center ecosystem.

Advantech, BACnet, Belimo-Energy-Valve, Bitpool, DGAPI, DNP3, DQL, DSA Over COAP, Digital Ocean, Elios4you, Enocean, Google plus, HTML 5, haystack, IRC, JDBC, kafka, MQTT, Mango, Modbus, Mongo DB, Motion Jpeg, MySQL, OPC, REST server, RSS feeds, Rasberry Pi, Relayr, RethinkDB, SNMP,IOT, Slack, Solaredge, Splunk, Traccar, Twilio, UPnP, WeMo, Weather, Webctri, Zwave, Zabbix, ZoneMinder and adding others as needed.

In summary, the workflow typically is used to deploy applications to controllers and aggregate controllers. The configurations and protocols required to map all the data coming from the end devices in a heterogeneous data center is deployed. The appropriate configuration including the protocol is selected for a specific end device during the final implementation.

3.4.3. Trusted Sensors

As already discussed, hackers can target even the smallest of devices sitting on the Internet for stealing information, falsifying data or disabling or corrupting the device. The

vulnerability of Software based security is high in certain types of attacks. If a device or sensor is accessible, it can possibly be cloned or altered in behavior by loading malware. Through this, unauthorized users can take control of the device and carry out different activities like disabling the device or providing information to hostile parties. Another type of threat is posed by replay attacks by the poor interception of secured password exchanges between host and client and reuse of the password for gaining access to the network. The hardware Trusted platform module has been proposed in the heterogeneous data center IoT platform.

In this section, we discuss the design methods of hardware trusted computing network. We detail the design of Trusted Platform Module (TPM), which is implemented in a standalone secure microcontroller. TPM performs measurements on system firmware, software and configuration data before execution begins, and compares the measurements with expected values stored securely on the chip. The software or firmware is allowed to run only if the respective sets of values match. If a mismatch is detected, the system may roll back the module in question to a last-known good state.

1. **Hardware trusted module topology:**

In an IoT platform the controller is connected to the sensor and device. As shown in Figure 11, the TPM Module is a security controller integrated circuit. The TPM Module is incorporated in the circuitry of the controller and also on the end device/ sensor.

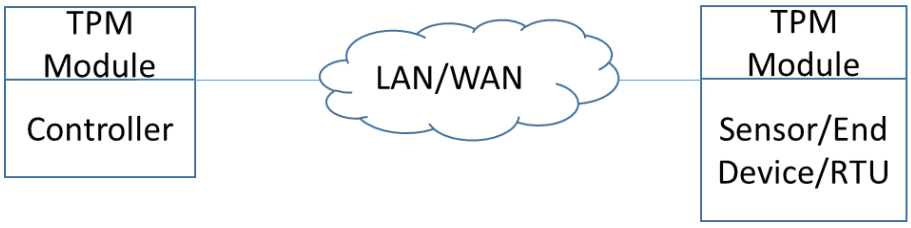


Figure 11: Hardware trusted platform module for IoT

2. **Design components of TPM**

The TPM circuitry along with software and firmware provides the root of trust platform. Components of the platform extends to other components by building a chain of trust, where each linkage of the components extend to next one. This section discusses the TPM architecture components that are used in the heterogeneous data center IoT platform. The components as well as its flow is illustrated in Figure 12.

Input and Output: The flow of information over the communications bus is managed by I/O component. TPM used Low Pin Count (LPC) bus interface to connect to the chipset. It carries out protocol encoding/decoding ideal for communication over internal and external buses. It directs messages to suitable components. Access policies associated to Opt-In component and the other TPM functions that require access control to be enforced by the I/O component. A specific I/O bus is not required by the main specification. Issues around a particular I/O bus are the purview of a platform specific specification.

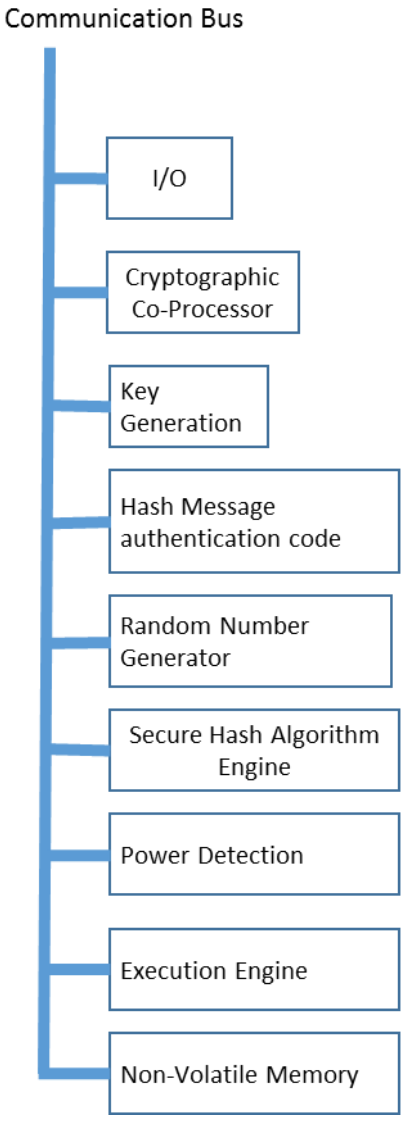


Figure 12: Components of IoT TPM

Cryptographic Co-Processor: Cryptographic operations within the TPM are implemented by the cryptographic co-processor. Conventional cryptographic operations are employed by the TPM in conventional ways.

These operations include Random number generation (RNG), Hashing (SHA-1), Asymmetric encryption/decryption (RSA), Hashing (SHA-1) and Asymmetric key generation (RSA). These capabilities are used by the TPM for generating random data, generating asymmetric keys and the signing and confidentiality of the data stored. Symmetric encryption maybe used by TPM for internal use but no algorithm is exposed to any symmetric algorithm functions for general TPM users. Additional asymmetric algorithms may be implemented by the TPM. Different algorithms may be used for signing and wrapping in the TPM devices implanting different algorithms.

Key Generation: RSA key pairs along with symmetric keys are created by the Key Generation component. No minimum requirements are placed by TCG on key generation times for asymmetric or symmetric keys.

Hash Message authentication code (HMAC): Two pieces of information are provided by the HMAC engine to the TPM. The first is the proof of knowledge of the Authenticated Data (AuthData) and the second is the proof of the authorization of the request arriving and that the command in transit has not been modified. The definition of HMAC is only for HMAC calculation. The mechanism or order of the data transported from the caller to the actual TPM is not specified by it. The HMAC creation is order dependent.

There are specific items for each command that are parts of the HMAC calculation. RFC 2104 is the initial point of the actual calculation. In order to properly define the HMAC in use, two parameters need to be selected by the RFIC 2104. The first value is the key length and the second value is the block size. This specification requires a key length of 20 bytes as well as a block size of 64 bytes. The basic construct is H (K XOR opad, H (K XOR ipad, text)) where

H = the SHA1 hash operation

K = the key or the AuthData

XOR = the xor operation

opad = the byte 0x5C repeated B times

B = the block length

ipad = the byte 0x36 repeated B times

text = the message information and any parameters from the command

Random Number Generator: In the TPM, randomness is provided by the Random Number generator. These random values are used by the TPM for randomness in signatures, key generation, and nonces. The RNG includes a state-machine that mixes and accepts unpredictable data in addition to a postprocessor that has a one-way function (e.g. SHA-1).

The rationale for the design is that a TPM can be a good source of randomness regardless of a genuine source of hardware entropy. The state of a state-machine can be non-volatile, which is initialized with unpredictable random data when TPM is manufactured before the TPM is delivered to the customers. In order to salt the random number, entropy or (unpredictable) data can be accepted by the state-machine at any time. The source of such data can be both hardware or software, for instance, monitoring of random mouse movements or keyboard strokes or from thermal noise.

After every TPM reset, a reseeding is required by the RNG. Compared to a software source, entropy is supplied at a higher baud rate by a true hardware source. When entropy is added to the state-machine it must be ensured by the process that the new state of the state-machine is not visible to the outside source. After the TPM has been shipped, the state of the state machine should not be deducible by the manufacturer or the owner of the TPM.

The output of the state-machine is condensed by the RNG post-processor into data that is uniform and sufficient entropy. Compared to the output produced, more bits of input data should be used by the one-way function. The current definition of the RNG allows a Pseudo Random Number Generator (PRNG) algorithm to be implemented. Nevertheless, for the devices that have a hardware source of entropy available, it is not necessary to implement PRNG.

Similar to RNG mechanism this specification refers to RNG as well as PRNG implementations. Distinguishing between the two is not required at the TCG specification level. On each cell, 32 bytes of randomness should be provided by the TPM. If enough randomness is not available, it is possible that larger requests may fail.

Secure Hash Algorithm Engine (SHA-1) Engine: The TPM primarily uses the SHA-1 hash capability because it is a trusted implementation of a hash algorithm. In order to support measurement taking in the platform boot phases and for allowing access of hash functions to

environments. The TPM is not a cryptographic accelerator. Minimum throughput requirements for TPM hash services is not specified by TCG.

Power Detection: The TPM power states are managed by the power detection component together with platform power states. It is mandated by the TCG that all the power state changes be notified by the TPM. Command-execution may be restricted by TPM in the periods when there are physical constraints for the platform operation. In a PC, generally the operational constraints occur in the power-on self-test (POST) and the Operator input is then required through the keyboard. Access to certain commands maybe allowed by the TPM when it is in a boot state or a constrained execution mode. The state changes affecting the TPM command processing modes may be notified by the TPM at some crucial point in the POST process.

Execution Engine: Program code is run by the execution engine for executing the TPM commands sent by the I/O port. The execution engine is crucial in making sure that shield locations are protected and operations are properly segregated.

Non-Volatile Memory: The use of non-volatile memory component is required for storing persistent identity and state related to the TPM. Items like Attestation Identity Key or the Endorsement Key are included in the NV area and it is available for use and allocation by the entities which the TPM Owner has authorized.

3.4.4. Controller Network Resiliency

As discussed in chapter 1 and chapter 2, heterogeneous data centers are mission critical systems that require high reliability and redundancy. In this section, we discuss the redundancy topology of controllers and aggregate controllers.

1. **Redundant Controller Architecture:**

A control system in a mission-critical environment such as heterogeneous data center is required to have the high-level of reliability because a single failure in such control systems may result in a huge catastrophe. Hence, the controllers are specially designed using a fault-tolerant architecture with multiple redundancy to maintain the high reliability. In fig. 13, we show a basic architecture used in the design. Controller C1 and C2 are connected to the end device/sensor/RTU via a relay switch. The relay switch is listening both C1 and C2. Single

controller runs at one time. In case of one controller failure, the relay switches to the other one. For example, if the steady state was running with controller C1, in case of C1 failure fault signal is sent to relay to switch C2.

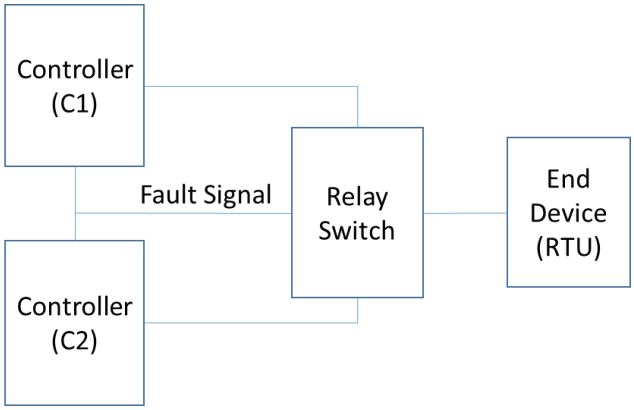


Figure 13: Redundant controller architecture

2. Hardware trusted module topology:

As discussed in Chapter 2, controllers are connected to aggregate controllers in the heterogeneous data center IoT topology. The topology shown in Fig.14, each controller is redundantly connected in a daisy chain. In this architecture AC1 and AC2 software is always running, but the network port of only one aggregate controller is active. For example, If AC1 is running is in steady state only AC 1 is sending IP traffic via IP back bone. AC2 listening AC1 failure. If AC1 fails, AC2 opens its network port to send data over the IP back bone.

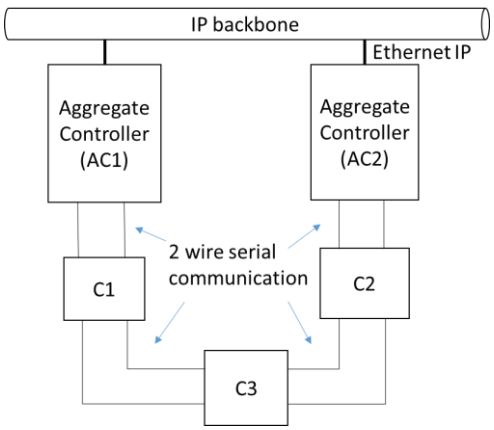


Figure 14: Redundant aggregate controller architecture

3.4.5. Controller Alerting at the Edge (Edge Fog Layer)

As already discussed, in cloud-based architecture, the storage and intelligence of application are centralized in server data centers. This architecture caters to the need of most of the Internet of Things (IoT) applications, however, tends to break down with limited network bandwidth, high volume of data and real-time requirements. Additionally, a crucial role is played by latency. There is an emerging need of decentralized processing, which are also known as the "edge computing" [38]. The perfect places for hosting IoT processing for systems are the intermediate layers of aggregate controllers and IoT gateways; these require the collection of data from various sources like bases, distributed data and sensors. They are located on the data path and their processing capabilities now allow running complex IoT applications and critical alerting.

1. **Alerting Logic:**

With thousands of data points collected using the IoT platform, it is not efficient to alert on every point. It is important to understand dependencies and correlation of each point in a sub ecosystem of the heterogeneous data center and creating actionable intelligence at the aggregate level. Our design addresses creation of each ecosystem. For example, let's consider HI-FOG fire suppression system in a heterogeneous data center. The high pressure allows the water mist to infiltrate into fire in a liquid form. This results in targeted evaporation at targeted locations were the liquid is needed most. High pressure water mist also successfully penetrates hard to reach space and provides higher level of cooling, hence protecting data center equipment and other structures.

In the above there is a single container filled with liquid that services three zones, zone 1 (z1), zone 2 (z2), and zone 3 (z3) in the data center. Liquid is pumped using a pump at certain pressure (P). The logic is created and alerted as follows.

a. If any of the Zone, z1, z2, z3, start the pump.

b. Calculate the desired pressure of the system

c. Maintain the desired pressure of the of full ecosystem

d. Alert when the pump fails

e. Alerts when cumulate desired pressure fails

f. Alert when cumulative liquid level is the tank is below desired level.

Each of the following alerts are mapped on escalation path to direct to an internal or external resource to be respond to resolve the issue. All the above is done using a user interface shown in figure 15. Alarm classes are created with high, medium, low status. The escalation logic is how the alert should be communicated and routed. Finally, the alert is delivered via email of push notification to mobile app.

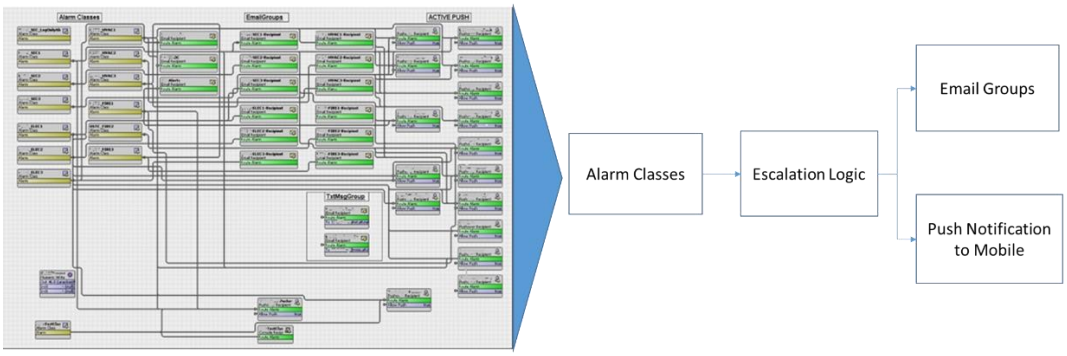


Figure 15: Programmed logic and escalation for alerting

2. Design motivation for the IoT edge computing:

Scalability. If several sensors and end devices are present in a heterogeneous data center, the centralized approach may not be sufficient for handling this increasing volume of end devices as well as its geographical specificities. The relevancy or safety of data is highest if the processing is done close to the edge of the network.

Network resource conservation. The network bandwidth required for carrying this newly created information is directly impacted by the volume of data generated by the various sensors. Distributed processing enables the relieving of the restrictions on the network by only sending the necessary information to the cloud or the operation center. This can also be done by carrying out majority of the data processing like mission critical alerting and analytic at the remote site, which is closer to the source of the data.

Latency. In order to create stable behavior in real-time systems, low latency is required. The large delays observed in overloaded cloud server farms and the various multi-hop networks prove to be unacceptable. Additionally, latency and timing jitter can be minimized by local, high performance nature of distributed intelligence. In terms of jitter

and latency the requirements of data center applications are very high. The most stringent requirements are satisfied only by local processing.

Resilience. In the case of a heterogeneous data center that is in a network blackout and is completely disconnected from the cloud, mission critical tasks like critical controls, analytics, alerting are to be performed at the local station. Rather than being a recommendation, an architecture based on distributing processing is generally the sole valid solution.

Thus the design is based on control at the edge location and performing mission critical alerting. In the current design, this has been implemented in the aggregate controllers. As illustrated in Figure 16, the fog layer is found in the aggregate controllers in the IoT framework. It is illustrated in Figure 17 that the middle layer of the IoT software framework is comprised of the fog layer alerting and analytics logic.

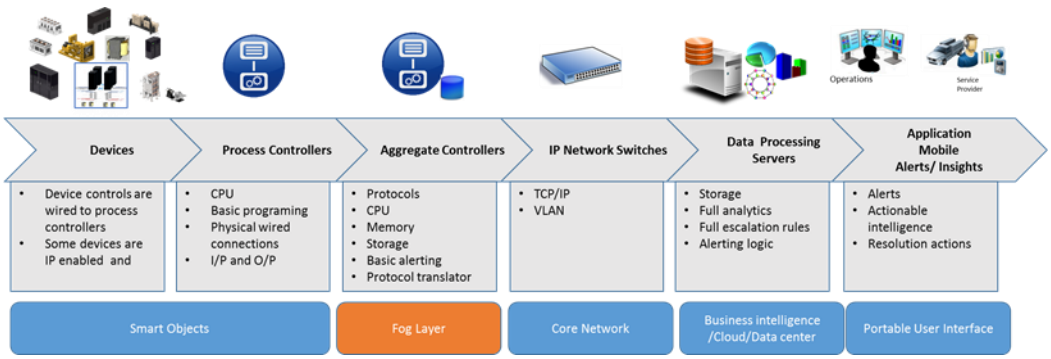


Figure 16: Fog Layer/ edge computing with IoT framework

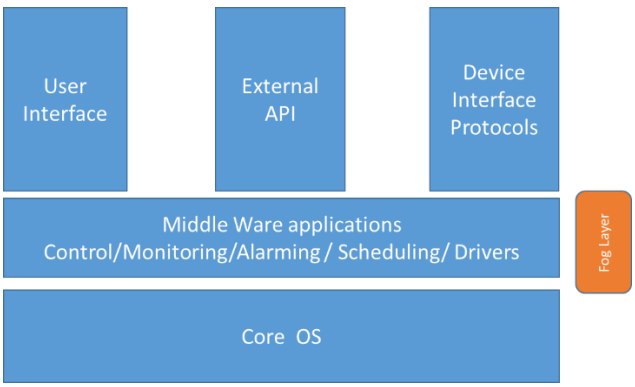


Figure 17: Fog computing resides in the middle application layer of the IoT software framework

The following section discusses the full end to deployment of the data center including all layers of the IoT Platform. It entails how end devices are connected to controllers which intern is connected to aggregate controllers. Further aggregate controllers are connected to database and application servers. This explains the dashboards and alerting implementation along with network topology and mobile application.

3.5. IoT Framework Implementation

In this section we explain the implementation of IoT platform used to collect and store data. The design consists of five layers. In this section we explain implementation of each IoT layer.

1. **Perception layer**: The main function of this layer is to obtain the various types of static / dynamic information of the real world through various types of sensors and to share with Internet access. The end devices several devices including electrical circuits, uninterrupted power supplies, cooling coils, chilling systems, fans, temperature sensors, pressure sensors etc. Some devices come with controls equipped to communicate via Modbus, BACnet, LonWorks, TCP/IP. Some of them have controls with no communication and need to be wired in custom methods to connect to process controllers. The connection can have established via dry contacts on machines to process controllers. Custom programs are written to bring the points into aggregate controllers. Following real examples from the IoT platform setup.

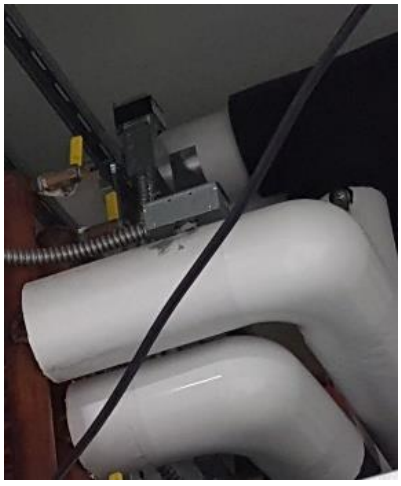


Figure 18: Cooling Coil Valve Instrumentation

Figure 18 shows cooling coil valve that manages amount of chilled water in the system. This an actuator. That is connected via two 22 gauge wires to programmable process controllers where the logic is programmed to measure position in percentage and communicated out on Lon works FT-10 network to aggregate controllers. The aggregate controllers convert the communication protocol to IP.

Figure 19: Process controllers connecting power meters

Figure 19 shows the process controllers connection to all the power meters. The end device is connected via to RS485 serial communication to process controller. The protocol is Modbus between process controller and aggregate controller. The aggregate controllers convert the communication protocol to IP.

Figure 20: On board chiller control board

Figure 20 shows process controller on the chiller with connectivity on the control broad. It communicates over BACNET MSTP (Multiple Spanning Tree Protocol) serial

communication that goes to the aggregate controllers. The aggregate controllers convert the communication protocol to IP.

Figure 21: Temperature sensors for hot and cold coils

Figure 21 shows the temperature sensors on hot and cold coils. These measure the water temperature. These are resistive thermistors. This is connected via two 22 gauge wires to programmable process controllers where the logic is programmed to measure temperature in degrees Fahrenheit and communicated out on LonWorks protocol to aggregate controllers. The aggregate controllers convert the communication protocol to IP.

Figure 22: Thermostat that measures hot and cold aisle temperature

Figure 22 shows the Thermostat that measures air temperature in hot and cold aisles. These are resistive thermistors. This is connected via two 22 gauge wires to programmable process controllers where the logic is programmed to measure temperature is degrees Fahrenheit and communicated out on LonWorks protocol to aggregate controllers. The aggregate controllers convert the communication protocol to IP.

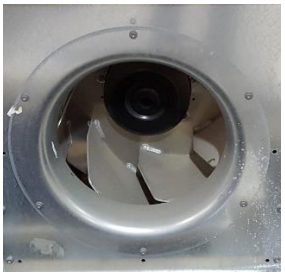


Figure 23: Fan in the cold aisle

Figure 23 shows a variable speed fan that is used in the cold aisle to maintain temperature and the air pressure in the cold aisle. This is connected via two 22 gauge wires to programmable process controllers where the logic is programmed to measure fan speed and communicated out on LonWorks protocol to aggregate controllers. This allows for continuous monitoring how much power is consumed by the fan in real-time. If the end device is expensive and highly mission critical the trusted module platform is deployed for higher security.

Figure 24: UPS Ethernet connectivity: Modbus IP

Figure 24 shows the UPS Ethernet port connection. The UPS have on board Ethernet communication port that communicates using Modbus IP.

2. **Access layer**: The main function of this layer is to send the perception layer information to the Internet through the various communication networks. Due to security reasons of wireless communication, this design entails wired connection. Therefore, the end devices are connected via wired connection. The devices that only have hot points connect using RS232 or RS485 to programmable process controllers. Following is the figure of a programmable process controller.

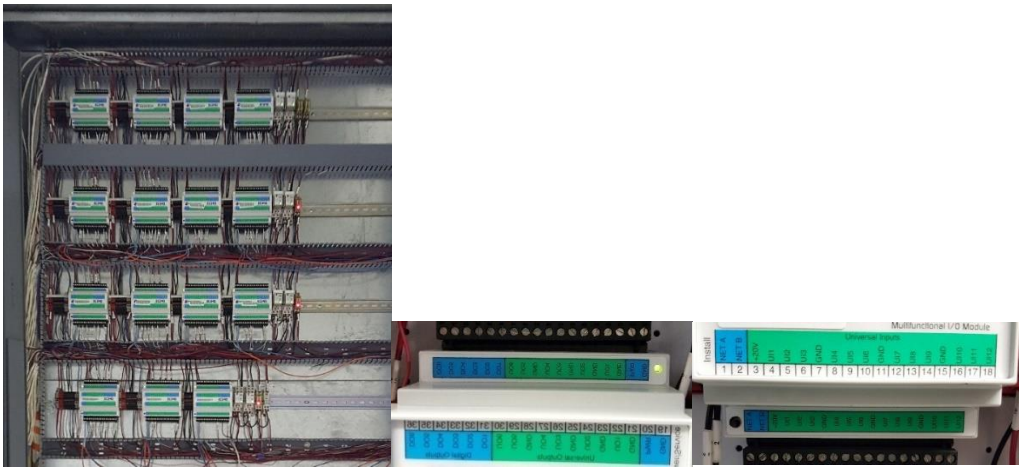


Figure 25: Programmable process controllers

Figure 25 shows programmable process controllers. The points are pulled in to process controllers and logics are programmed to measure the points. Some data points are readily available and for some data points a set of sensors are deployed to collect information legacy end devices in a heterogeneous data center. Process controllers have limited I/O. Based on points that need to be collected from the end device, it may require to deploy multiple process controllers. If the end device is mission critical and requires 100% uptime, then the process redundant controllers are deployed as discussed in section 3.4.4.

Figure 26 shows the schematic of configurable logic programmed in the programmable process controller. The configurable logic is based on understanding of all the data points that can be gathered from an end device. The logic is created based on the design methods detailed in 3.4.2. The configurable logic and protocol resides in the middle layer of IoT platform software framework shown in Figure 9. Software workflow used in the creation of configuration logic as shown figure 10. This process created logic and protocol are developed and deployed in a single controller for all the end devices in heterogeneous data center. The controller then used by the user to pick the configurations to end device for final deployment.

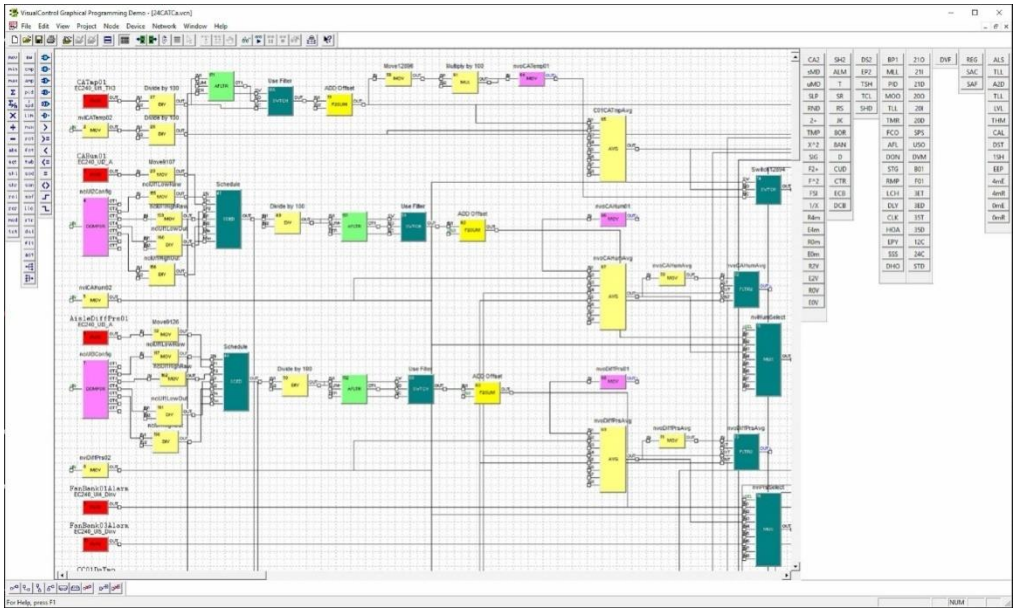


Figure 26: Programmed logic schema for process controllers

3. **Network layer**: The main function of this layer is to establish an efficient and reliable infrastructure platform for upper management and large-scale industrial applications with global Internet platform. Given the security concerns, the devices were directly connected by wires. Controls on the devices come via communication mechanisms. Majority of the machines of the machines talk over MODBUS, BACNET and LON protocols. The controllers are preprogrammed with all the protocols described in section 3.4.4.

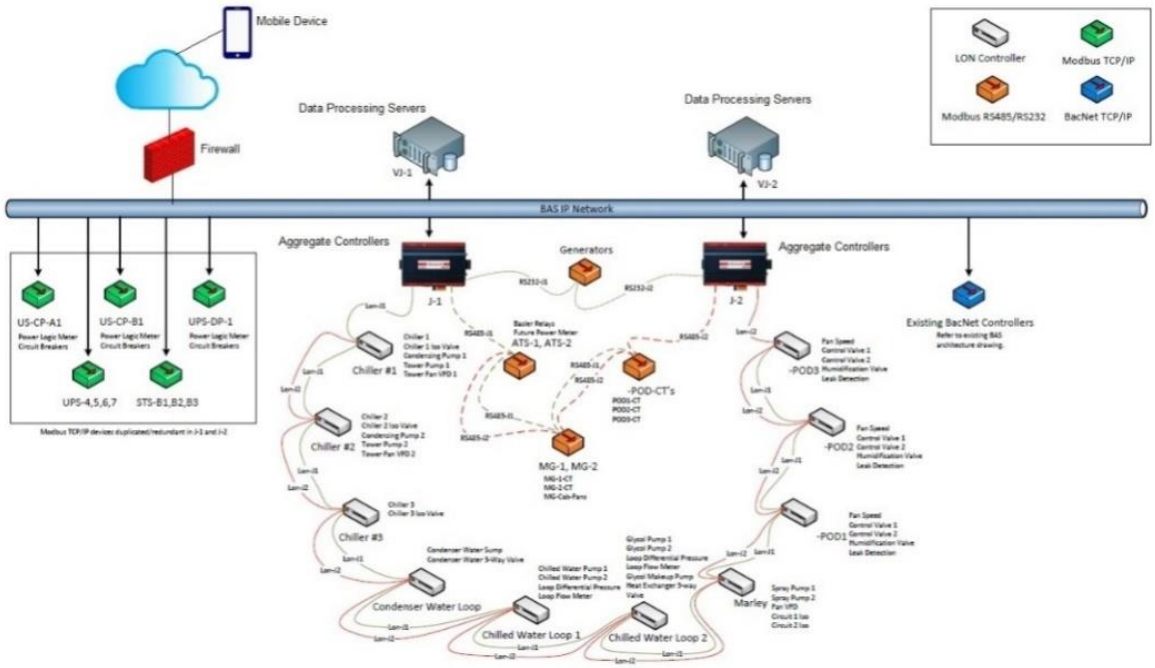


Figure 27: IoT network topology

Figure 27 shows the network topology of the IoT implementation. The network includes two aggregate controllers for network resiliency. Each controller is labeled showing which end device it is connected to. On the top right of figure 27 shows the legend for how the controllers are communicating. As shown, there are LonWorks, Modbus TCP/IP, Modbus RS485/RS232 and BACnet TCP/IP. The controllers are daisy chained and connected in redundant fashion to aggregate controllers as described in section 3.4.4. Figure 28 shows an aggregate controller. The aggregate controller is a gateway unit that the controllers are connected to. They have core operating system, user interface, external API, device interface protocols and middle ware applications. The Software framework is as shown in Figure 9. The aggregate controller acts as protocol translator to convert all the various protocols coming from controllers to Internet protocol (IP). Once the it is converted to IP, the communication up stream to analytical servers in the data center or the cloud over IP. The edge fog layer resides at the aggregate controller. The detailed design of the alerting and the edge is explained in section 3.4.5.

Figure 28: Aggregate controllers

The IP traffic is connected via cat 5 cables to switch stack and all the intermediary firewalls. For security for purpose it is best separating IoT traffic from other traffic. All IoT traffic is separated via dedicated VLAN. Figure 29 is the image of switch stack configured in master slave configuration to maintain network redundancy.

Figure 29: Switch stack

4. **Service management layer**: This layer exists at the edge device and in servers in the data centers and cloud. The servers in the data center and cloud are mainly responsible for getting real-time control and management of the huge amounts of data. Big data analytics and analytics reside at this layer. This layer also provides the interface to upper layer application with a good user interface. These interface is deployed as when dashboards or a mobile application. Also, alarming and management resides in a skinny fashion in aggregate controllers. The detail design of alerting at the edge device in explained section 3.4.5. The edge alerting provides resiliency at local data center in case of loss connection the servers in the data center. Figure 15 shows a programming logic schema for alerting and escalation. Point coming from the end devices and these can be programmed to change the priority of alerts and escalations to responsible individuals.

5. **Application layer**: The key purpose of this layer is the integration of the underlay system function in addition to building a practical application for midsize heterogeneous data center. This layer has the advanced alarm management, reporting analytics and dashboards. The left side of figure 30shows the cooling subsystem dashboard. It provides real time monitoring to network operations center. It has three major components cooling tower, chillers and heat exchanger. All relevant points are shown on the dashboard and anomalies are shown by flashing numbers. The right of figure 30 is a graphical representation of a data suite in heterogeneous data center. This displays values of all the point monitored for the data suite. Also, All the data collected is stored in database for historical trending, predictive analytics and ad-hoc analysis.

Figure 30: Control Dashboards

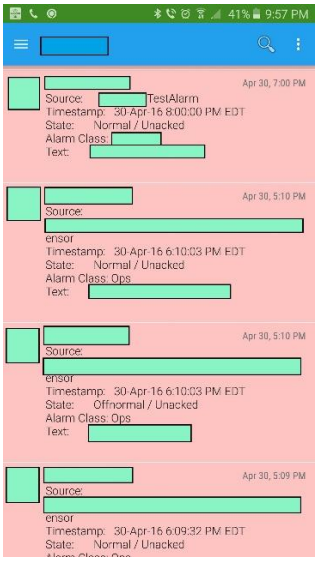


Figure 31: Mobile App: Schematic of the alerts/notifications received

Figure 31 shows the schematic of alerts received on the mobile phones. These alerts are received based on the rules in the programmed logic for alerting and escalation. This app list app with all internal and external resources. The app delivers the details of the issue right to your mobile app. There is a feature in this app that override all your phone configurations to deliver the mission critical alert.

3.6. IoT Platform Use Cases, Benefits and Results

It took about two years to deploy the IoT Platform. Since the first version of the solution was not stable, many iterations of testing were conducted on it. A major challenge in testing was that it had to be carried with end device running in production environment. There is a need of resources for overriding the issues detected. After the completion of the testing, the standard process is the new provisioning in production systems. The results and benefits attained after the deployment and running of the IoT platform for 10 months are listed below.

IoT Design and Implementation benefits.

Through flexible configuration, the data center is able to connect easily to the various old and new devices. With this, the data center is able to gather data from the various devices in the midsize heterogeneous data center. Once full data collection is possible on the same time stamp interval, it becomes feasible to use the learn machine learning approach for non-linear optimization problems with various service management, predictive maintenance and optimization.

Previously, it was possible for a contractor to replace sensors/RTU and NOC operations and go unnoticed. With the implementation of trusted sensors on critical systems, the replacement sensors could be controlled tightly. The system identifies any changed sensor and takes the change through apt change management processes for implementation. As a result, any issues regarding physical security breaches and unauthorized replacement of sensors can be resolved.

Since a data center has a mission critical nature, down time may occur due to a controller failure monitoring mission critical end device, which in turn affects the uptime service levels agreements of the data center. The data center has 2N configuration, due to the execution of controller network resiliency, resulting in 100% uptime for controllers as well as high availability.

If any previous loss of network connectivity to the data center suit or cloud results in the non-availability alerting server, the NOC will not have any of the critical alerting functions available to it. With the execution of the edge device alerting, the data center will be able to receive critical alerts from the aggregate controller (edge device), which refers to the IoT internal network working separately from the outside network connectivity servers.

Use cases with the IoT platform deployment:

When a problem alert is raised after the IoT platform has been implemented, the people responsible for deriving a solution are notified automatically even before the service organization organizes a conference call, giving them the opportunity of reacting proactively in minutes. With this more accountability, redundancy and speed is created and pressure is taken off the network operating center that was earlier susceptible as a single point failure.

- The biggest success of the datacenter was the ability of mobilizing the resources required for the management of such a highly automated and complex system. costs were effectively managed by the company with the collaboration of the numerous internal business units and the utilization of the strength of its expert partners and vendors on demand, in contrast to the hiring of full-time personnel.
- When an internal transformer failure occurs in the datacenter, an on call vendor technician directly receives the alert along with the exhaustive error codes. The ownership of the issue is taken by the vendor technician who follows the communication as well as the resolution protocols. With the elimination of intermediary communication delays, the response time of the vendor technician is lower and issues are resolved promptly. While in the past about 45 min to 60 min would have been required for responding to this issue, now it only requires less than 15 minutes.
- The vendor directly maintains the liquid levels in the data center. The vendor directly sends the levels. The precise levels for 100% SLA are proactively maintained by the. With this the internal resource for managing this is eliminated.
- With the real time alert, which is accompanied by detailed error codes provided to an expert vendor for an UPS failure, there is an increase in the first time resolution, compared to the numerous attempts made for resolving this issue before the implementation of the IoT platform.

Business results of the IoT platform deployment:

1) Lower response time, which decreased from 45 minutes to less than 10 minutes.
2) Higher SLA compliance, which increased from 84% to 100%.
3) Greater customer retention and acquisition, with 22% increase in new business and 100% renewal of contracts.
4) Decreased full-time, on-site personnel costs, with the reduction of 4 full time equivalent employees.

Better collaborative, efficient business/vendor relationship, with an increase from 63% to 95The implementation of IoT platform throughout the systems has led to positive outcomes for the datacenter. With the integration of the IoT platform, the processes were streamlined, which in turn reduced the alert response time from approximately 45 minutes to 10 minutes. In

addition to improvement in efficiency, faster response and compliance to SLA has also led to the improvement of customer relationships. Also, ever since the implementation 100 percent customer retention has been maintained and there has been a 22 percent increase in the improvement of acquisition.

The data center enabled the better integration of the external vendors in addition to streamlining and increasing the accuracy of the monitoring and response to alerts. As a result, the need for almost four Network Operations Centre (NOC) technicians was eliminated by the data center while taking pressure off the initial single failure point. The internal team/vendor relationships are better in the data center, increasing the efficiency of the NOC technicians with no need for troubleshooting disorganized alarms. The first time resolution was increased by the data center from 63% to 95%.

The goal of the datacenter to consistently improve the reliability and efficiency of mission critical systems was met by embracing an IoT platform as well as integrating diverse technology system infrastructures with internal and external people, mobile devices and physical building infrastructures.

This implied the meshing of various that did not mesh well, for instance, the circuits and power delivery required by the data center, liquid chiller loops, HVAC systems, uninterruptible power supplies and generators. Additionally, it implied to the ability of measuring power consumption on the basis of individual customer.

The data center was able to mesh the systems with personnel by the creation of this system inside the new IoT platform. Figure 32 illustrates the various components of a midsize heterogeneous datacenter that used one single funnel for feeding the data that needed to be filtered, analyzed and pushed back out as target alerts to the suitable parties.

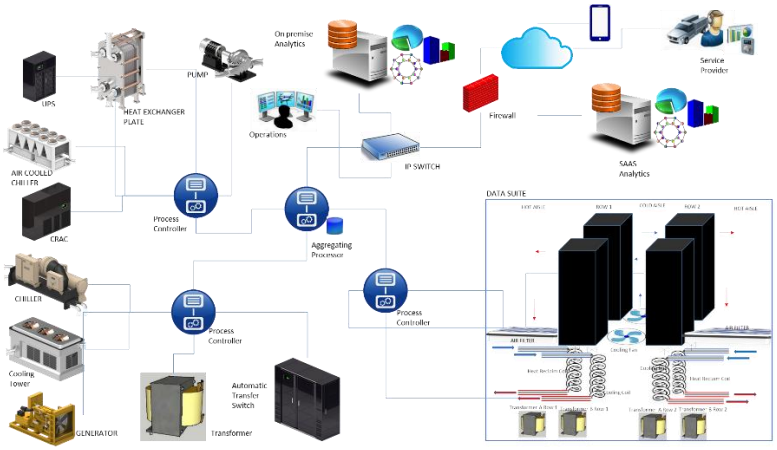


Figure 32: Shows all the components of a traditional mid-size data centers

3.7. Neural Network Model Methodology

The data collected using IoT deployment is the prerequisite to applying neural network methods. The IoT platform design and implementation discussed in the previous section has enabled to collect data to apply neural network methods. In the following sections we will discuss the methods machine learning techniques.

An Artificial Neural Network (ANN) can be defined as an information processing paradigm based on the way information is processed by biological nervous systems like the brain. The new framework of the information processing structure is the main element of this paradigm. It includes numerous highly interconnected processing elements (neurons) that work together to solve a particular problem. Similar to people, ANNs learn by example. The configuration of an ANN is done using a learning process for a particular application such as data classification or pattern recognition. In biological systems learning occurs through the adjustments between the synaptic connections existing between the neurons. Neural networks have a remarkable ability of deriving meaning from imprecise or complicated data, which facilitate the extraction of patterns and detection of trends that other computer techniques or humans may not be able to notice because of their complexity [82]. A trained neural network can be perceived as an expert in the given information category for analysis. Other advantages have been listed below.

1. Adaptive learning: This refers to the ability of learning the way to execute tasks on the basis of the data provided for initial experience or training.

2. Self-Organization: It is possible for an ANN to create its own representation or organization of the information received at the time of learning.
3. Real Time Operation: ANN computations can be carried out simultaneously and in order to maximize on this capability, the designing and manufacturing of special hardware devices is being carried out.
4. Fault Tolerance via Redundant Information Coding: The associated performance degrades when the network leads are partially destructed. Nevertheless, despite the major network damage, some network capabilities can be retained.

A significant role is played by the neural network model in the data center optimization, which Google DC has successfully implemented. The neural network has a global defense through which any real time problem can solved yielding better results.

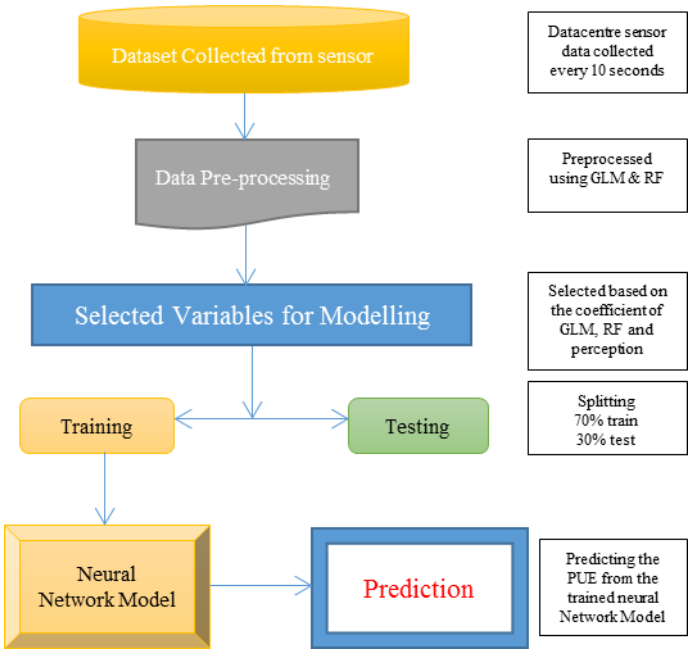


Figure 33: Neural Network Model Block Diagram

The block diagram as shown in Figure 33 represents the logic flow of neural network prediction model which invokes the transformation of initial dataset retrieved from the sensor ports to the final Prediction obtained from the neural network model. Initially it is started by pre-processing, Variable Selection, Data Sampling and Neural Network Prediction Model. The collected data is pre-processed through by some techniques such as filling missing variables, removing outliers, normalization and creating new variables. The variables are selected from the pre-processed data through positive skewness arrived with the target SUITE PUE with other dependent variables. The selection of variables is achieved through the Generalized Linear

Model (GLM), Random Forest (RF) and Experts Perception because of dependency. The sampling process is done for the selected variables choose for modeling, splitting into training, testing and validation dataset. The training data set are trained through neural network model and the testing data is used for the prediction of the data sets through which neural network is trained for the evaluation of SUITE PUE.

3.7.1. Data Pre-processing

Data pre-processing refers to a data mining technique wherein raw data is transformed into a comprehensible format. In general, real world data is likely to have more errors, likely to lack certain trends or behaviors, may be inconsistent and incomplete. Data pre-processing is a verified way of solving such issues. In data pre-processing raw data is prepared for further processing [83]. The use of data pre-processing can be done in rule-based applications such as neural networks and database-driven applications like customer relationship management.

- Data Cleaning: Different processes are used for data cleaning such as solving the discrepancies in the data, smoothing of the noisy data and filling in missing values.
- Data Integration: Refers to the bringing together of different data representations and the resolving of conflicts within the data.
- Data Transformation: Refers to the normalization, aggregation and generalization of data.
- Data Reduction: The aim of this step is to provide a reduced illustration of the data in a data warehouse.
- Data Discretization: In this step several values of a continuous attribute are reduced by dividing the attribute intervals range.

The data set used for pre-processing is total of 243 columns and 119421 rows. Described below are the techniques used for pre-processing, where R-Statistical software for the data pre-processing was used.

1. **Filling Missing Values**: In real time, certain variables may be missing values in observations because of technical problems or issues with the sensor system. The best approach is to completely discard those values, however due to the general lack if large

enough samples it is not affordable to lose data as there is certain information present o the non-missing entries as well.

In mean computation, the mean (average) of the available data for that specific numeric variable in the sample is used as a substitute. In the case of discrete variable. The value that is most often seen or is the most likely value is used as a substitute. In imputation by regression, an attempt at predicting the value of a missing variable is made using other variables whose values are known. A separate classification or regression is defined based on the type of data variable missing, which is then trained by data points for which values like this are known. If several variables are missing, the means are taken as the initial estimate and the process is repeated till the stabilization of the predicted values. The regression approach is considered to be equal to mean imputation if there is no high correlation between the variables [84].

2. Removing Outliers: Majority of the outliers are dealt in the same way as the methods for missing values such as statistical methods, imputing values, treating them as a separate group, binning them, transforming them and observations [85]. Common ways of dealing with outliers have been listed below:

(i) Deleting observations: Outlier values are deleted if the number of outlier observations is small, or if there is a processing error or a data entry error. Additionally, trimming at both ends can also be used for removing outliers.

(ii) Transforming and binning values: Outliers can also be eliminated by transforming variables. The variation that extreme values cause is reduced by the natural log of a value. Another form of variable transformation is binning. Through decision three algorithm outliers can be dealt with properly because of the binning of variables. Additionally, the process of assigning weights to different observation scan also be used.

(iii) Imputing: Similar to missing values, outliers can also be imputed. The mode, median and mean imputation methods can be used. Prior to imputing values, the analysis of an outlier as natural or artificial should be carried out. Values can be imputed if it is artificial. Additionally, statistical model can be used to predict outlier observation values, which can then be imputed with predicted values.

(iv)Treat separately: If the numbers of outliers are high, they should be treated separately in the statistical model. One approach that can be taken is treating both groups separately and building individual model for both groups after which the output is combined.

3. Creating New Variables: In real time the data may be large, some of the variable can be combined or split based on the data scenario. They can be of two types Splitting of variables and Joining of variables.

(i) Splitting of Variables: The data in a single column may contains values that can occupy five other columns. For Example, time format "04/11/2015 12:01:00 PM" can be split into "04/11/2015", "12", "01", "00" and "PM", using string splitting operation.

(ii) Joining of Variables: The data in multiple columns can be aggregated into a single column based on the dependency/ necessity. This approach uses AND or operation to achieve it, here we used summation Suite Flow (gal/min), Suite Absorption (kW), Suite Fan Power (kW), Suite Fan Airflow (CFM), Suite Absorption proportional to chiller power (kW).

4. Data Normalization: Normalization involves in rescaling the attributes which are numeric in the range [0,1]. In later section normalization is used for examine sensitivity analysis [79].

One possible formula is given below:

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

3.7.2. Variable Selection

The quantification of the variable importance is an important issue in many applied problems complementing variable selection by interpretation issues.

In the linear regression framework, making a distinction between various variance decomposition based indicators: “dispersion importance”, “level importance” or “theoretical importance” quantifying explained variance or changes in the response for a given change of each regressor.

In the random forests framework, increase in the mean of error of a tree in the forest is the most commonly used importance score of a given variable, particularly when the OOB samples have the random permutation of the observed values of the variable. Such random

forests are often referred to as permutation importance indices as opposed to total loss of node impurity measures.

Two crucial questions regarding the variable importance behavior are addressed here. The first is the importance of a group of variables and the second is the way it behaves when around highly correlated variables. The basis for variable selection process is the cooperation of variable importance for the classification and model estimation for generating, evaluating and comparing a family of models.

We use Generalized Linear Model, Random Forest and Experts perception in the below section to signify the importance of variables with all the methods.

3.7.3. Generalized Linear Model

Generalized linear model (GLM) refers to a flexibly generalizing ordinary linear regression that rather than normal distribution permits response variables with error distribution models. Linear regression is generalized by GLM by permitting the association between the linear model and response variable through link. Additionally, it can also be done by permitting the magnitude of each measurement's variance to be function of the values predicted. A stepwise procedure is a natural technique for selecting variables in terms of generalized linear models. [80].

3.7.4. Stepwise Procedure

The general working scenario of stepwise procedure is followed up with 3 rules such as Backward Elimination, Forward Selection and Stepwise Regression. The rules are incorporated in GLM variable selection model in R- Statistical Software as a package to follow up. The generalization of each rules are explained below.

1. Backward Elimination

This is the easiest process of variable selection and is easy to implement without any particular software. In situations where the hierarchy is complex, the execution of backward elimination can be done manually while taking into consideration the variables that are eligible for removal.

(i) Begin with all the predictors given in the model.

(ii) Eliminate the predictor whose p-value is higher than α_{crit}

(iii) After the model is refit, again go to step 2

4. Stop the process when p-values are lower than α_{crit}. The α_{crit} is also referred to as "p-to-remove" and does not have to be higher than 5%. If the goal is prediction performance, then it would be ideal to have 15-20% cut-off, even though preference should be given to methods designed rather directly for optimal prediction.

2.Forward Selection

This method is just the reverse of backward method.

(i) Begin with no variables in the model.

(ii) Check if the p-value of all the predictors in the model have been added. Select the one whose p-value is lowest compared to $\alpha_{crit.}$

(iii) Keep the process going till the addition of new predictors is possible.

3. Stepwise Regression

This method combines forward selection and backward elimination. This takes care of the situation where the addition or elimination of the variables is done early in the process and their change can be contemplated in the future. A variable can be added or eliminated at each stage and there are several variations in terms of how this can exactly be done. While computationally, the stepwise procedures are comparatively cheap, there are certain limitations associated with it.

(i) It is possible that at times the optimal model maybe missed due to the "one-at-a-time" nature of adding or eliminating variables.

(ii) It is imperative that the p-values used not be taken too literally. The validity is uncertain as so much multiple testing is taking place. The significance of the remaining predictors tends to increase when the less significant ones are increased. As a result, the significance of the remaining predictors is exaggerated.

(iii) Since the procedures are not directly associated to the final prediction objectives, it may not be helpful in solving the given problem. It is imperative to consider that when it comes to any variable selection method, the basic purpose of selection cannot be separated

from the model selection. The statistical importance of the remaining variables in the model is amplified due to variable selection trends. Correlation can still be drawn between the variables and the response. It would be wrong to say that there is no relation between the variables and the response, it is rather that no additional explanatory effect is provided by them beyond the variables that already are a part of the model.

(iv) In general, the models picked in stepwise selection smaller than required for the purpose of prediction. For instance, simple regression that only has one predictor available. If the slope for this predictor does not apparently have any statistical significance, there may not be enough evidence to relate it to y but it could still be used for predictive purposes.

The results obtained from the GLM model is explained in detail in the results and discussion section.

3.7.5. Random Forest Algorithm

Random forests refer to a concept of the general technique of random decision forests that comprise an ensemble learning method for tasks like classification and regression. Many classification trees grow in the random forests. The classification of a new object from an input vector can be done by putting each vector down every tree in the forest. A classification is provided by each tree and the tree is perceived to "vote" for that class. The classification that has most of the votes is selected by the forest.

Described below is how each tree is grown:

1. If N is the number of cases in the training set, the casing of sample N is done at random from the original data, but with replacement. This sample will be used as the training set for tree growth.
2. If the number of variables is M, each node is specified with a number m<<M, such that there is random selection of variables from M and the node is split using the best split on this during the forest growing, m is held constant.
3. The trees are not pruned and each tree is allowed to grow as much as possible.

There are two factors on which forest error rate is dependent. (i) The correlation among any couple of trees in the forest. The forest error rate increases with increase in correlation. (ii) The strength that each tree in the forest has. It is likely that a tree will be a strong classifier if

it has low error rate. The forest error rate can be decreased in the strength of individual trees is increased.

The strength and correlation reduces when m is reduced and vice versa. The optimal range of m is somewhere in between and is usually quite wide. Random Forest is also used to find the variables significance owing to its unexcelled precision. Discussed below is the general methodology of finding the variable importance and the results are elaborated in the results and discussion section.

The two data objects that random forests generate determine most of the options. (i) oob (out-of-bag) data: The derivation of the training set for the current tree is done through sampling and replacement after which almost one-third of the cases are removed from the sample. The use of this oob (out-of-bag) data is done to obtain a running impartial estimate of the classification error as the forest is added with trees. Additionally, it is used in getting estimates of variable significance. (ii) Proximities: After the building of each tree, the entire data is run down the tree and the computation of proximities is done for each pair of cases. If the same terminal node is occupied by two cases, the proximity goes one up. When the run ends, the normalization of the proximities is done by dividing it by the number of trees. The use of proximities is done in the production of illuminating low-dimensional views of the data, locating outliers, and replacing missing data [86].

3.7.6. Variable importance

In every tree grown in the forest, put down the oob cases and count the number of votes cast for the correct class. Now randomly permute the values of variable m in the oob cases and put these cases down the tree. Subtract the number of votes for the correct class in the variable-m-permuted oob data from the number of votes for the correct class in the untouched oob data. The average of this number over all trees in the forest is the raw importance score for variable m.

If the values of this score from tree to tree are independent, then the standard error can be computed by a standard computation. The correlations of these scores between trees have been computed for a number of data sets and proved to be quite low, therefore we compute

standard errors in the classical way, divide the raw score by its standard error to get a z-score, ands assign a significance level to the z-score assuming normality.

If the number of variables is very large, forests can be run once with all the variables, then run again using only the most important variables from the first run.

For each case, consider all the trees for which it is oob. Subtract the percentage of votes for the correct class in the variable-m-permuted oob data from the percentage of votes for the correct class in the untouched oob data.

3.7.7. Data Sampling

The total input size is 119421, which is separated into 70% (training), 15% (testing) and 15% (validating). This is dataset which is imputed for the analysis purpose.

Through sampling, data analysts, predictive modelers and data scientists are able to use small, wieldy amount of data for building and running analytical models rather quickly, while still generating precise results. the use of sampling is especially significant that are too large for a complete efficient analysis for instance, big data applications. However, size of the required data sample is a significant consideration. In some cases, most of the information regarding a data set can be obtained from a very small sample. In other cases, the use of a larger sample can increase the probability of accurate representation of the data as a whole, even though the large size of data sample may obstruct the ease of interpretation and manipulation. In any case, large and almost complete data sets are mainly used for drawing samples.

Samples can be drawn from data in many ways, determined by the situation and the data set. Probability is the basis for sampling, wherein random numbers corresponding to points in the data set are used. It is ensured by this approach that the points selected for the sample are not correlated. Additional variations in probability sampling include multi-stage cluster sampling in addition to systematic, stratified and simple sampling. Once a sample is generated its use can be done in predictive analytics [87]. The classification of the sample dataset used in the neural network model is done into validation, testing and training.

1. Training

The training dataset refers to the data used for constructing or discovering a predictive relationship. Majority of the approaches searching through training data for establishing empirical relationships have a tendency of over fitting the data, which means that apparent links in the training data that are not often found can be identified.

Training set: A set of examples used for learning: to fit the parameters of the classifier in the MLP case, we would use the training set to find the "optimal" weights with the back-prop rule

2. Validation

Validation is carried out for estimating the training (depending on the input, the value that needs to be predicted and the size of data) of the model as well as its model properties (IR-models precision and recall, classification errors for classifiers and mean error for numeric properties).

Validation set: This refers to a set of examples used for tuning the classifier parameters. In reference to the MLP case, validation set is used for determining a stopping point for the back-propagation algorithm or identifying the optimal number of hidden units.

3. Testing

A test set refers to a data set that is not dependent on the training data, but permits the probability distribution equal to the training data. Over fitting is minimal is a model fit to the training set is also a fit for the test set. Over fitting is possible if compared to the test set the training set has a better fit.

Test set: This refers to a set of examples that are used only for assessing a fully-trained classifier in terms of its performance. In relation to the MLP case, the test would be used for estimating the error rate after the final model (MLP size and actual weights) has been selected. After the final model has been assessed on the test set, the model should not be tuned any further for obtaining a better accuracy.

The reason of having validation and test dataset separately because error rate estimate of the final model on validation data will be biased (smaller than the true error rate) since the validation set is used to select the final model. After assessing the final model on the test set, model can be tuned to get better accuracy by varying parameters. The dataset used for neural network model is summarized in the result and discussion section.

3.8. Solution Approach: Neural Network Model

Data Mining terms like computational learning theory and machine learning are often used for denoting the application of classification algorithms or classification algorithms for predictive data mining. Conventional statistical data analysis is generally associated with the estimation of population parameters through statistical inference, however, data mining generally emphasizes on precision of the prediction, irrespective of the explicability or comprehensibility of the of the techniques or models used for generating the prediction. The application of this type of technique to predictive data mining can be illustrated through meta-learning techniques or neural networks like boosting. Generally, these methods include the fitting of very complex "generic" models that are not associated to any theoretical or reasoning understanding of basic causal processes; rather accurate classifications or predictions can be generated from these techniques in cross validation.

Neural networks are referred to as analytic techniques inspired from (hypothesized) learning processes in the cognitive system and the neurological brain functions with the capability of predicting new observations from other observations after the execution of the so-called learning process from the existing data. Neural Networks is another data mining technique.

The primary step is the designing of a specific network architecture, wherein a particular number of layers are included, each of which consists of a particular number of neurons. It is imperative that the network size and structure matches the nature of the phenomenon under investigation. Since at an early stage the latter is not well known, this task is rather difficult and involves numerous trials and errors. However, there is neural software makes artificial intelligence techniques applicable in this difficult task to ascertain the best suited network architecture.

The training process is then carried out for the new network. In this phase, an iterative process is applied by the neurons to the several inputs so that the weights of the network can be adjusted for optimal prediction of the sample data for performing the data. After the use of existing data in the learning phase, the new network is ready and can be utilized for generating predictions.

The network thus developed in the learning process is a representation of a pattern observed in the data. Thus, the network in this approach is the functionally equal to a model of

relations among the variables in the conventional model building approach. Nevertheless, opposite to the traditional models, the relations in the network cannot be articulated in the usual terms used in methodology or statistics for describing relations among variables; for instance, A and B are positively correlated only when the value of D is high and C is low. Certain neural networks are capable of producing highly precise predictions, however, they represent a classic a-theoretical research approach (also referred to as "a black box"). This approach is associated only with practical considerations, which implies to the solution's predictive validity and its applied relevance and is not associated with the nature of the basic mechanisms or its pertinence for any theory of the fundamental phenomena.

Nevertheless, it should be noted here that the use of Neural Network techniques can also be done as an aspect of analyses designed for building explanatory models because Neural Networks can facilitate the exploration of data sets for the identification of pertinent variables or variable groups. The results thus obtained further help in the process of model building. Furthermore, there is a neural network available now that makes the use of sophisticated algorithms for searching the most relevant input variables, thereby making a potential contribution directly to the process of model building.

A major advantage of neural networks noted is that, hypothetically, they have the capability of approximating any constant function; as a result of which no theories regarding the fundament model are required by the researcher and probably to some extent, certain variables may also not be required. However, a significant disadvantage here is that the final solution is dependent on the initial network conditions and as already discussed, the interpretation of the solution in the conventional analytic terms is not possible, particularly the ones used in building theories explaining the phenomena.

It is emphasized by some authors that massively parallel computational models are used or are expected to be used by neural networks. For instance, according to Haykin [88] neural network can be defined as a hugely parallel distributed processor with a natural inclination towards the storage of experiential knowledge that can be made available for use. There are two ways in which it resembles brain: 1) the acquisition of knowledge is done through a learning process; and 2) knowledge is stored using interneuron connection strengths referred to as synaptic weights.

3.8.1. Multi-Layer Perceptron

The neural network Algorithm used multi-layer perceptron, which are well applicable when modeling functional relationships. A Multi-Layer Perceptron (MLP) has the fundamental structure of a directed graph, which means that in comprises of directed edges and vertices, referred to as synapses and neurons in this particular case. The connection of synapses is only possible with subsequent layers. Covariates in separate neurons comprise the input layer while the response variables comprise the output layer. The layers in between are known as hidden layers as they cannot be directly observed. The input layers as well as the hidden layers have a constant neuron connecting for synapses interception, implying the synapses that the covariate does not influence directly. A neural network is illustrated in Figure 34, showing one hidden layer consisting of three hidden neurons. The base for this neural network is the way the two covariates A, B are related to the response variable Y. In theory, any number of covariates and response variables can be used. Nevertheless, convergence difficulties can occur when a vast number of covariates and response variables are used.

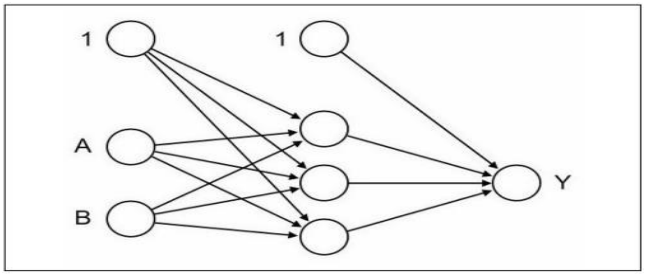


Figure 34: Example of a neural network

A weight is attached to each of the synapses, which indicates the effect that corresponding neuron has and the data passes through the neural network in the form of signals. The so-called integration function first processes all the signals through the combination of all the incoming signals after which the so-called activation function transforms the output of the neuron.

The simplest multi-layer perceptron (also known as perceptron) consists of an input layer with n covariates and an output layer with one output neuron.

It calculates the function

$$o(x) = f\left(w_o + \sum_{i=1}^{n} w_i x_i\right) = f w_o + w^T x$$

where w_o denotes the intercept, **w** = (w_1,..., w_n) the vector consisting of all synaptic weights without the intercept, and **x** = (x_1,..., x_n) the vector of all covariates.

3.8.2. Supervised Learning

Learning algorithms at the time of learning that are focused on supervised learning algorithms fit the neural networks to the data. The characterization of these learning can be done on the basis of the use of a specific output, which is compared to the projected output and through the adaptation of all parameters in accordance with this comparison. Usually the initialization of all weights is done with random values taken form a standard normal distribution. The steps listed below are repeated during an iterative process.

1. For current weights and given inputs x output o(x) is calculated by the neural networks. If the training process is yet to be completed, there will be a variation in the predicted output o and the observed output y.
2. An error function *E*, like the sum of squared errors (SSE).

$$E = \frac{1}{2}\sum_{l=1}^{L}\sum_{h=1}^{H}(o_{lh} - y_{lh})^2$$

Or cross entropy

$$E = -\sum_{l=1}^{L}\sum_{h=1}^{H}(y_{lh}\log(o_{lh}) + (1 - y_{lh})\log(1 - o_{lh}))$$

Measures the difference between predicted and observed output, where $l = 1,L$ indexes the observations, i.e. given input-output pairs, and $h = 1,H$ the output nodes.

3. The rule of a learning algorithm is used for adapting all weights. The process stops after the fulfillment of pre-specified criterion; for instance, if the given threshold is higher than all absolute partial derivatives of the error function with respect to the weights ∂E/∂w. The resilient back propagation algorithm is a widely used learning algorithm.

3.8.3. Back propagation and Resilient Back propagation

The basis for the resilient back propagation algorithm is the conventional back propagation algorithm, wherein the weights of a neural network are modified so that a local minimum of the error function can be identified [26]. Thus, the calculation of the gradient of the error function ∂E/∂w is done with respect to the weights so that the root can be found. The weights are particularly modified going in the direction opposite to the partial derivatives till the time a local minimum is obtained. Figure 35 roughly illustrates this basic idea for a univariate error-function.

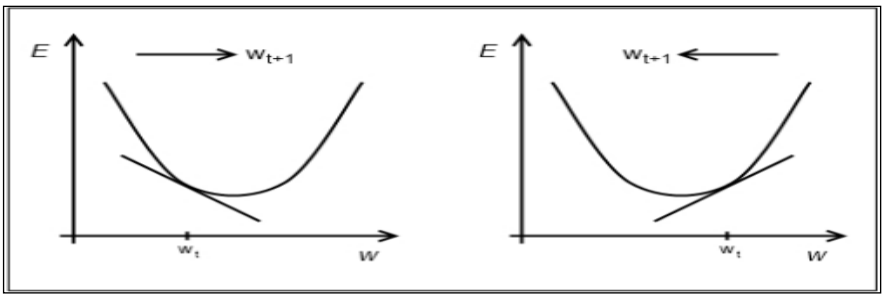


Figure 35: Univariate Error Function

The weight is increased if the partial derivative is negative and the weight is decreased if the partial derivative is positive. This makes sure that a local minimum is obtained. The chain rule is used for calculating all partial derivatives as the calculated function of a neural network mainly comprises of activation and integration functions.

In resilient back propagation, the traditional back propagation algorithm is used for each weight along with a separate learning rate η_k that can alter during the process of training. This takes care of the issue of defining an over-all learning rate that is suitable for the entire network and the entire training process. Additionally, the weights are updated using their sign and not the magnitude of the partial derivatives. This makes sure that the learning rate has an equal influence over the entire network. The following rule is used for adjusting weights, where k indexes the weights and t the iteration.

$$w_k^{t+1} = w_k^{(t)} - \eta_k^{(t)}.sign(\frac{\partial E^{(t)}}{\partial w_k^{(t)}})$$

where *t* indexes the iteration steps and *k* the weights.

In order for convergence to speed up in shallow areas, there will be an increase in the learning rate if the sign of the corresponding partial derivative is kept. Alternatively, if the sign of the partial derivative of the error function is changed, the learning rate will be decreased. This is because a missed minimum because of a large learning rate is indicated by a changing sign. If weight backtracking is not used, it may occur several times that algorithm will jump over the minimum.

A resilient back propagation is performed by the globally convergent version introduced with extra modification of learning rate associated to all other learning rates. Usually, it is the smallest learning rate (indexed with i) or the learning rate related to the smallest absolute partial derivative that is changed in accordance with the dataset.

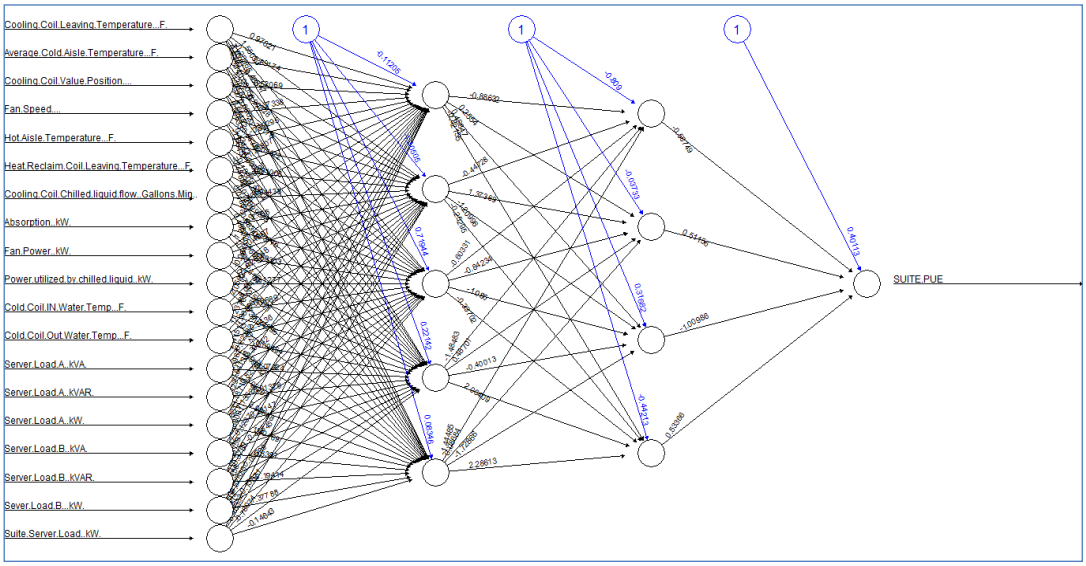
$$\eta_i^{(t)} = -\frac{\sum_{k;k\neq i} \eta_k^{(t)} \cdot \frac{\partial E^{(t)}}{\partial w_k^{(t)}} + \delta}{\frac{\partial E^{(t)}}{\partial w_i^{(t)}}}$$

If $\frac{\partial E^{(t)}}{\partial w_i^{(t)}} \neq 0 \; and O < \delta \ll \infty 0.$

3.8.4. Power Optimizing Framework

The machine learning algorithm used is Neural Network. The neural network model utilizes 2 hidden layers and 0.01 as the regularization parameter as shown in the Figure 36. The training dataset contains 19 input variables and one output variable (the Suite PUE). The total size of the data samples used is 119421 rows, which was collected from a heterogeneous datacenter sensor ports. Of the total, training uses 70% of the dataset while cross-validation and testing use the remaining 30%. Before splitting, random shuffling of the chronological order of the dataset is done so that biasing the testing and training sets on older or newer data can be avoided. The result and discussion section simulates the neural network model predicted PUE vs actual PUE graph.

Figure 36: Block diagram of Neural Network Modeling



3.9. Sensitivity Analysis:

Sensitivity analysis denotes to the study of the apportioning of the uncertainty in the output of a mathematical model or system (numerical or otherwise) to different uncertainty sources in its inputs [89] [90]. Uncertainty analysis is an associated practice which focuses more in propagation of uncertainty and uncertainty quantification. Ideally, sensitivity and uncertainty analysis should be run together. There are several advantages of the process of recalculation of outcomes under different assumptions so the influence of variable under sensitivity analysis can be determined.

- To test the strength of the system or model results while uncertainty is present.
- Better comprehension of the link between input and output variables in a model or system.
- Uncertainty reduction: Identification of model inputs causing considerable uncertainty in the output. It should thus be increasingly focused on so that the strength can be increased/
- Searching the model for errors by facing unanticipated links between inputs and outputs.
- Model simplification – Identification and removal of repetitive parts of the model structure or fixing model inputs that do not affect model structure.
- Enhancement of communication from modelers to decision makers by making more persuasive, compelling, understandable and credible recommendations.
- Identifying regions in the space of input factors for which the model output meets the optimum criterion, or is minimum or maximum/
- When models are calibrated with large number of parameters, the calibration stage can be eased by a primary sensitivity test by maintaining the focus on the sensitive parameters. If sensitivity of parameters is not known, it is possible that time will be wasted on non-sensitive ones [91].

The analysis is done in the dataset for analyzing the behavior of dependent variables (cold aisle, fan power, etc..) over the target Suite PUE. For sensitivity analysis, the data is initially normalized, and results are simulated with explanation in the result and discussion section [92]. Data normalization is performed given its broad range of raw feature values.

The values of a feature vector z are mapped to the range [¬1, 1] by:

$$z_{norm} = \frac{z - mean(z)}{\max(z) - \min(z)}$$

The 19 dependent variable used for modeling is as follows

Dependent Variables	Dependent Variables
Cooling Coil Leaving Temperature (°F)	Power utilized by chilled liquid (kW)
Average Cold Aisle Temperature (°F)	Cold Coil IN Water Temp (°F)
Cooling Coil Valve Position (%)	Cold Coil Out Water Temp (°F)
Fan Speed (%)	Server Load A (kVA)
Hot Aisle Temperature (°F)	Server Load A (kVAR)
Heat Reclaim Coil Leaving Temperature (°F)	Server Load A (kW)
Cooling Coil Chilled liquid flow (Gallons/Min.)	Server Load B (kVA)
Absorption (kW)	Server Load B (kVAR)
Fan Power (kW)	Sever Load B (kW)
	Suite Server Load (kW)

Table 1: Selected Variables for modeling

Note that meta variables derived from individual sensor data are many of the inputs representing totals and averages.

3.10. Solution Method: Cooling Power Simulation

In this section, we describe selection of variables for simulation of cooling power optimization in predictive fashion. This uses a small block in the whole architecture of heterogeneous modular data center. This uses Cold Aisle Temperature, Chiller power, Server Power, Fan Power and Suite PUE for doing a compact analysis, which proves that it can be applicable for multiple number variable/block for whole data center optimization.

The below neural network model Figure 37 uses a sample use case, which uses five variables for training among that three are input variable and two are target variables. The simulation results by varying Cold Aisle Temperature and Fan Power to Chiller Power is discussed in detail on the result and discussion section

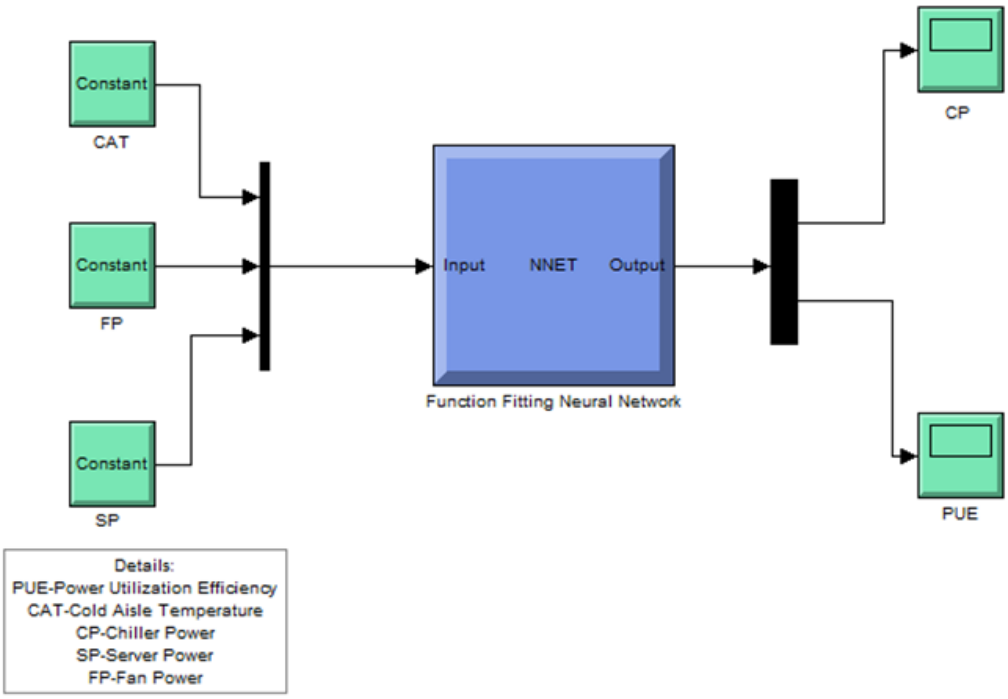


Figure 37: Neural Network model for predicting Chiller Power and PUE

Chapter 4: Results and Discussions

4.1. Introduction

In this chapter we discuss the results and outputs of methods used. First, we discuss results of how important variables are selected. We discuss training results and validation of machine learning. Further, we compare optimized machine learning output to actual operation value to discuss variances in the systems to predictively identify anomalies in the operating systems. Second, we discuss the results of improvements in operation metrics. Lastly, we propose further work with neural network controller to learn and perform corrective action to maintain optimal operations.

4.2. Variable Importance Approaches

4.2.1. Generalized Linear Model

The results below report the coefficient obtained from the GLM Algorithm, which explains the importance of the variable (Fan power, Chiller power etc..) associated with the target variable (Suite PUE). The tool used for the analysis in R- Statistical Software, below are the explanation based on the results obtained from the algorithm

- Input used 27 variables, explain in detail
- Output 19 as important

Probability Measure(p)

The Significates codes represent the probability measure (p), indicates whether the variable is significant or not. The p-value greater than 0.05 is represented as non-significant/not important variable. The results tabulated in Table 2 signifies that 15 as important and 12 variables as not valid, based on the probability measure.

Deviance

Measure of goodness of fit of a generalized linear model is called deviance. Or rather, it's a measure of badness of fit. If the numbers are higher that indicates worse fit.

Deviance reporting by R-reports are done two forms– The first one is the null deviance and the second one is the residual deviance. As shown in table 2, we have a null deviance value of 8.40 on 23858 degrees of freedom. Residual deviance of 0.0 points on 23859 degrees of freedom, which is a substantial reduction in deviance. The above numbers indicated that the Residual Deviance has reduced by 0.0 with a loss of 8 degrees of freedom.

Fisher Scoring

Fisher's scoring algorithm is a derivative of Newton's method for solving maximum likelihood problems numerically. As indicated in Table 2, two iterations were needed to perform the fit.

Information Criteria

The Akaike Information Criterion (AIC) provides a method that assesses the quality of the model through comparison of related models. AIC based on the Deviance, but if the model is complicated it penalizes you. Much like adjusted R-squared, it's intent is to prevent you from including irrelevant predictors [93].

However, unlike adjusted R-squared, the number is not meaningful by itself. If you find yourself dealing with more than one similar candidate models (where all of the variables of the simpler model occur in the more complex models), then the one with smallest AIC is selected. It is very useful for comparing models, but cannot be interpreted just as by itself.

Deviance Residuals:				
Min	1Q	Median	3Q	Max
-1.08E-03	-7.90E-05	-9.73E-06	7.11E-05	1.27E-03
Coefficients: (3 not defined because of singularities)				
	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.27E+00	1.07E-03	1190.166	< 2e-16 ***
Aisle Differential Pressure	8.86E-04	9.92E-04	0.893	0.371606
Aisle Differential Pressure set Point				
Cooling coil leaving air temprature	2.91E-05	4.71E-06	6.186	6.28e-10 ***
Cold aisle average temparature	1.71E-03	3.70E-03	0.461	0.644729
Aisle Differential average Pressure	-6.19E-04	4.47E-04	-1.386	0.165785
Cold aisle temparature	2.81E-04	1.62E-04	1.735	0.082669 .
Cold aisle temparature set point				
Cooling coil valve position	9.83E-05	2.94E-06	33.429	< 2e-16 ***
Control temprature	-2.05E-03	3.71E-03	-0.553	0.580245
Fan Speed	6.61E-05	4.23E-05	1.564	0.117854
Hot aisle temp	5.30E-05	1.39E-05	3.811	0.000139 ***
Heat Reclaim coil leaving temprature	-5.18E-05	1.29E-05	-4.001	6.33e-05 ***
Heat Reclaim coil valve position				
Cooling Coil Chilled liquid flow	-6.66E-05	1.33E-06	-49.941	< 2e-16 ***
Absorption	-5.04E+00	8.05E+01	-0.063	0.950103
Fan Power	5.71E-03	1.34E-05	424.667	< 2e-16 ***
Fan Airflow	-1.30E-07	6.87E-08	-1.893	0.058320 .
Absorption Chiller power	2.36E+01	3.76E+02	0.063	0.95009
Cold coil entering water temprature	1.88E-04	7.31E-06	25.724	< 2e-16 ***
Cold coil leaving water temprature	-1.11E-04	4.47E-06	-24.765	< 2e-16 ***
Server Load A (kVA)	1.16E+02	4.11E+01	2.808	0.004996 **
Server Load A (kVAR)	-8.85E-05	1.12E-05	-7.88	3.40e-15 ***
Server Load A (kW)	-4.72E-05	1.37E-05	-3.449	0.000564 ***
Server Load B (kVA)	1.14E+02	4.05E+01	2.808	0.004996 **
Server Load B (kVAR)	5.71E-05	1.11E-05	5.14	2.76e-07 ***
Sever Load B (kW)	1.53E-04	2.12E-05	7.227	5.08e-13 ***
Suite Server Load (kW)	-1.11E+02	3.97E+01	-2.808	0.004995 **
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1				
(Dispersion parameter for gaussian family taken to be 2.828426e-08)				
Null deviance: 8.40667722 on 23883 degrees of freedom				
Residual deviance: 0.00067483 on 23859 degrees of freedom				
AIC: -347320				
Number of Fisher Scoring iterations: 2				

Table 2: GLM Variable importance report

4.2.2. Random Forest Algorithm

The random forest algorithm, signifies the importance of the variable based on the Node purity. Random forest consists of a number of decision trees. Each node in the decision trees is a condition on a single feature, designed to split the dataset into two so that similar response values end up in the same set. For the analysis purpose, set ntree=100 & maxnodes = 100. The graph Figure 38 and the report in Table 3 signifies the importance of variable.

Note: IncNodePurity means increase in node purity, Total decrease in node impurities from splitting on the variable, averaged over all trees. Impurity is measured by residual sum of squares. Impurity is calculated only at node at which that variable is used for that split. Impurity before that node, and impurity after the split has occurred. Also higher the IncNodepurity lower its MSE.

- Input used 27 variables, explain in detail
- Output 22 as important, because the target variable PUE had impact with the dependent variable and listed in ascending order.

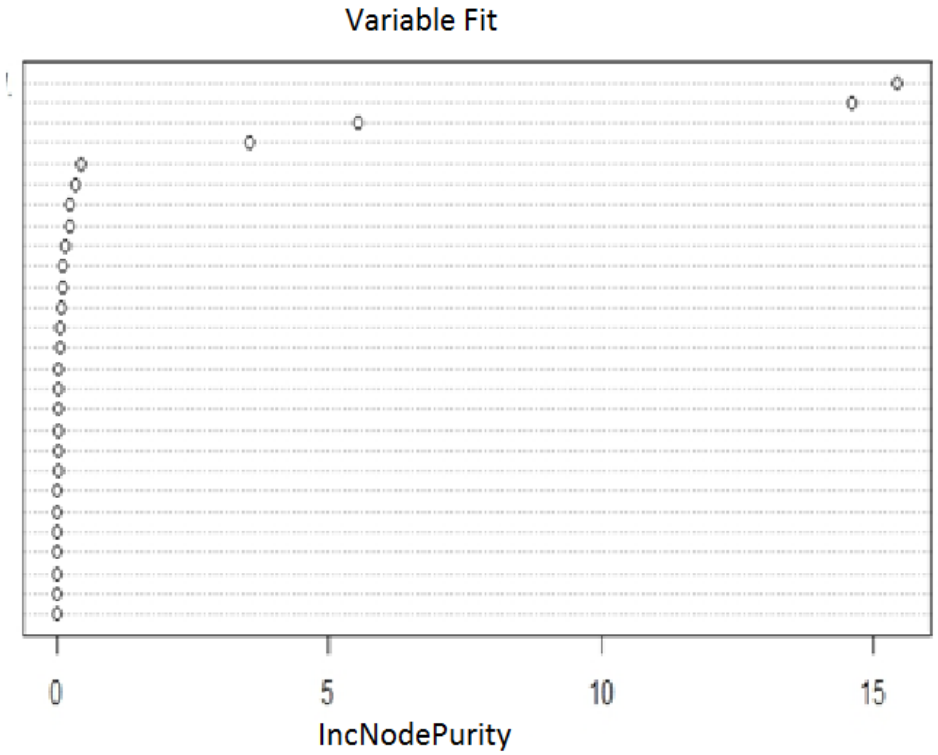


Figure 38: Random Forest Variable Importance Graph

Variables	IncNodePurity
Absorption Chiller power	15.44297253
Absorption	14.59729198
Cooling Coil Chilled liquid flow	5.544058326
Cooling coil valve position	3.536483804
Cooling coil leaving air temprature	0.454430588
Cold coil leaving water temprature	0.329052662
Server Load B (kVA)	0.234556198
Sever Load B (kW)	0.223578545
Cold aisle average temparature	0.150899916
Control temprature	0.115939492
Suite Server Load (kW)	0.104115687
Server Load B (kVAR)	0.072819523
Heat Reclaim coil leaving temprature	0.059305632
Hot aisle temp	0.044140633
Fan Power	0.033339934
Server Load A (kVAR)	0.022739893
Fan Speed	0.021319433
Cold coil entering water temprature	0.02071094
Fan Airflow	0.020497607
Server Load A (kVA)	0.019540539
Server Load A (kW)	0.014609945
Cold aisle temparature	0.005741113
Aisle Differential Pressure	0
Aisle Differential Pressure set Point	0
Aisle Differential average Pressure	0
Cold aisle temparature set point	0
Heat Reclaim coil valve position	0

Table 3: Random Forest Variable Importance report

4.2.3. Perception Approach

This approach evolved from the subject expertise of data center

Expertise Choose variables
Cooling coil leaving air temprature
Cold aisle average temparature
Aisle Differential average Pressure
Cooling coil valve position
Cooling Coil Chilled liquid flow
Absorption
Absorption Chiller power
Fan Speed
Fan Power
Heat Reclaim coil leaving temprature
Server Load A (kVA)
Server Load B (kVA)
Server Load A (kVAR)
Cold coil entering water temprature
Cold coil leaving water temprature

Table 4: Perception Table

4.2.4. Final Variables chosen for modeling

The selected variables for modeling is derived from GLM, RF and Perception approach, the logic used is

$$(GLM \cap RF) \cup PA$$

GLM->Generalized Linear Model

RF-> Random Forest

PA-> Perception Approach

The intersection of GLM and RF, union with PA gives the important variables which are used for the modeling and tabulated in Table 5.

Variable finally selected for modelling
Absorption Chiller power
Absorption
Cooling Coil Chilled liquid flow
Cooling coil valve position
Cooling coil leaving air temprature
Cold coil leaving water temprature
Server Load B (kVA)
Sever Load B (kW)
Cold aisle average temparature
Suite Server Load (kW)
Server Load B (kVAR)
Heat Reclaim coil leaving temprature
Hot aisle temp
Fan Power
Server Load A (kVAR)
Fan Speed
Cold coil entering water temprature
Server Load A (kVA)
Server Load A (kW)

Table 5: Selected Variables for modeling

Data Sampling:

The data sampling is the process of separating the dataset finally chosen for modeling into taking training, testing and validating dataset. The total input size is 119421, which is separated into 70% (training), 15% (testing) and 15% (validating). This is dataset which is imported for the analysis purpose.

4.3. Machine learning training results

Neural Network Diagram:

The below Plot of a trained neural network including trained synaptic weights and basic information about the training process. It produces the structure of the trained neural network, i.e. the network topology. As shown in Figure 39, The plot includes by default the trained synaptic weights, all intercepts as well as basic information about the training process like the overall error and the number of steps needed to converge. The neural network model uses one input layer, two hidden layer with 9 neurons and one output layer.

Input size(row) is 119421

There are 20 variables, among which 19 are input variables and 1 output/target variable.

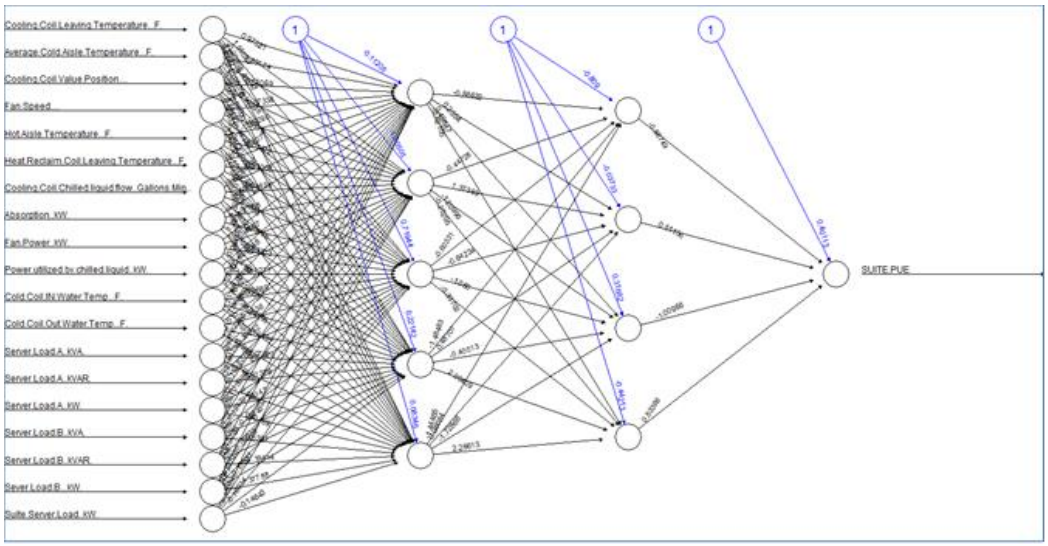
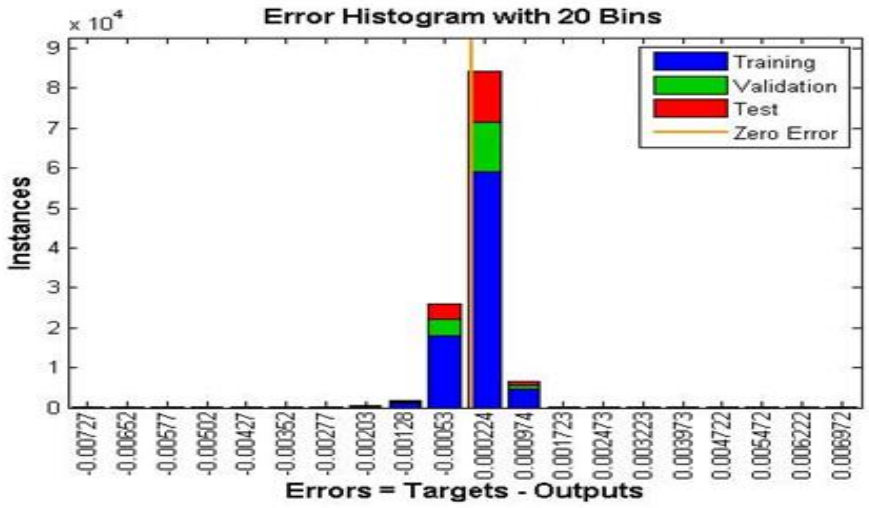


Figure 39: Neural Network node graph

The below figure 40 represents the error histogram obtained from the neural network model where blue bars represent training data, the green bars represent validation data, and the red bars represent testing data. The histogram can give you an indication of outliers, which are data points where the fit is significantly worse than the majority of data. In this case, you can see that while most errors fall between -0.00128 and 0.00974. This indicates that the error rate is very low and model building on this dataset will yield good result.



Figure

Figure40: Neural Network model error histogram

The performance of the neural network model is represented in the below figure 41. This represents the property epoch indicates the iteration at which the validation performance reached a minimum. The training continued for 6 more iterations before the training stopped. This figure does not indicate any major problems with the training. The validation and test curves are very similar. If the test curve had increased significantly before the validation curve increased, then it is possible that some over fitting might have occurred.

The next step in validating the network is to create a regression plot, which shows the relationship between the outputs of the network and the targets. If the training were perfect, the network outputs and the targets would be exactly equal, but the relationship is rarely perfect in practice

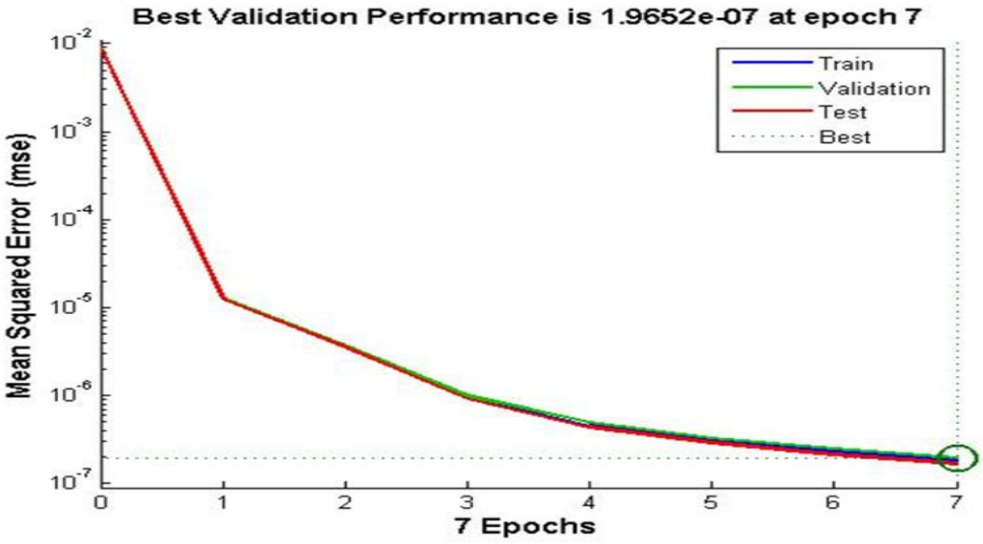


Figure 41: Performance Graph of neural network model

The four plots represent the training, validation, testing and all data. The dashed line in each plot represents the perfect result – outputs = targets. The solid line represents the best fit linear regression line between outputs and targets. The R value is an indication of the relationship between the outputs and targets. If R = 1, this indicates that there is an exact linear relationship between outputs and targets. If R is close to zero, then there is no linear relationship between outputs and targets.

For this example, the training data indicates a good fit. The validation and test results also show R values that greater than 0.9. The scatter plot is helpful in showing that certain data points have poor fits.

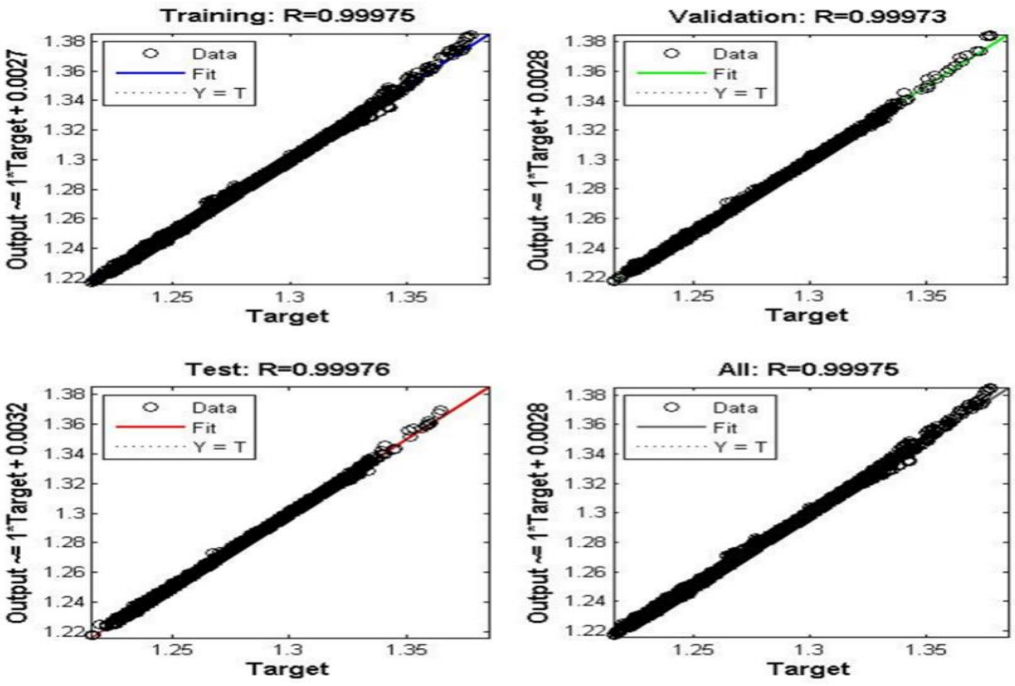


Figure 42: Regression analysis for Training, Validation and Test dataset

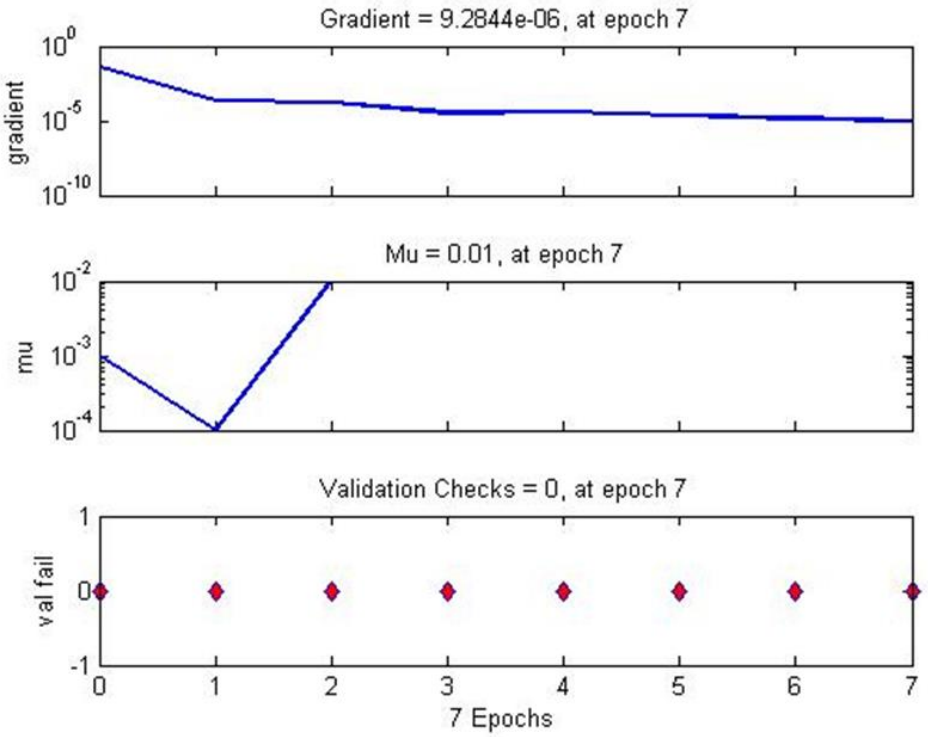


Figure 43: Neural Network model training state

The above figure 43 represents the training state of the neural network model. As an illustration of how the training works, consider the simplest optimization algorithm — gradient descent. It updates the network weights and biases in the direction in which the performance function decreases most rapidly, the negative of the gradient. Gradient is a value of back propagation gradient on each iteration in logarithmic scale. 9.2844e-06 means, reached the bottom of the local minimum of the goal function.

Themu is the control parameter for the algorithm used to train the neural network. Choice of mu directly affect the error convergence. In case of least mean squared LMS algorithm, mu is dependent on the maximum Eigen value of input correlation matrix, plot signifies that the mu value is less 0.01.

The number of validation checks represents the number of successive iterations that the validation performance fails to decrease. If this number reaches 6 (the default value), the training will stop. In this run, you can see that the training did stop because of the number of validation checks arrived is zero.

For understanding epoch: In MATLAB an epoch can be thought of as a completed iteration of the training procedure of your artificial neural network. That is, once all the vectors

in your training set have been used by your training algorithm one epoch has passed. Thus, the "real-time duration" of an epoch is dependent on the training method used (batch vs. sequential, for example).

epoch - Presentation of the set of training (input and/or target) vectors to a network and the calculation of new weights and biases. Note that training vectors can be presented one at a time or all together in a batch.

Mat lab allows you to set a maximum number of epochs after which to terminate the training procedure. This is used to stop the training in case the solution of the training algorithm does not converge, to prevent infinitely running the training [93].

4.3.1. Actual VS Predicted PUE results validation

The plot figure 44 below shows the variation between the Actual PUE calculated from the sensor data and Predicted PUE arrived from neural network. Also each PUE signal is predicted with the frequency of 10 seconds.

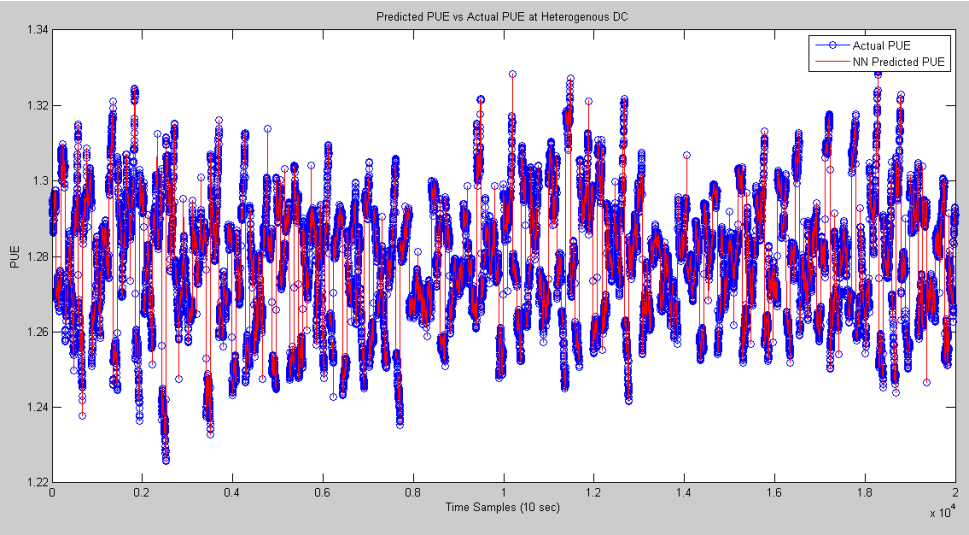


Figure 44: Predicted vs. Actual PUE

The neural network detailed in this paper achieved a mean Square error of +0.004 and standard deviation of 0.001 on the test dataset. The model accuracy for those PUE ranges is expected to increase over time as additional data are collected on heterogeneous DC operations.

4.4. Sensitivity Analysis

The following graphs reveal the impact of individual operating parameters on the DC PUE. We isolate for the effects of specific variables by linearly varying one input at a time while holding all others constant. Such sensitivity analyses are used to evaluate the impact of set point changes and identify optimal set points. All test results have been verified empirically.

Figure 45 & 46 represents that Fan power and Fan Speed are directly proportional to SUITE PUE, where Figure 45 signifies a linear variation between the PUE and Fan Power but Figure 36 depicts that there is an optimization in fan speed through an upper sloppy curve which creates a positive impact in the fan power for controlling the PUE without exceeding drastic change in the power consumption.

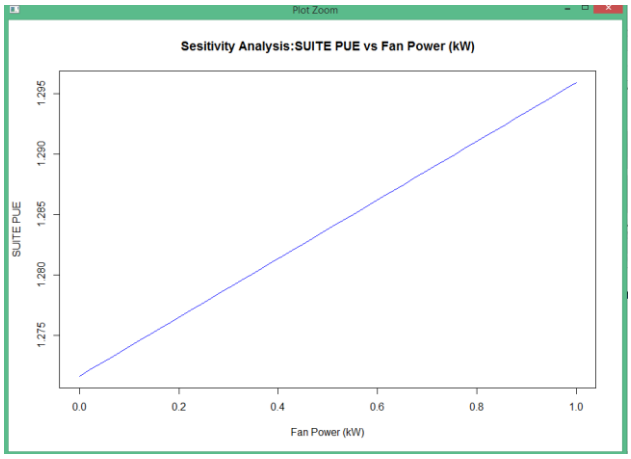


Figure 45: SUITE PUE vs. Fan Power

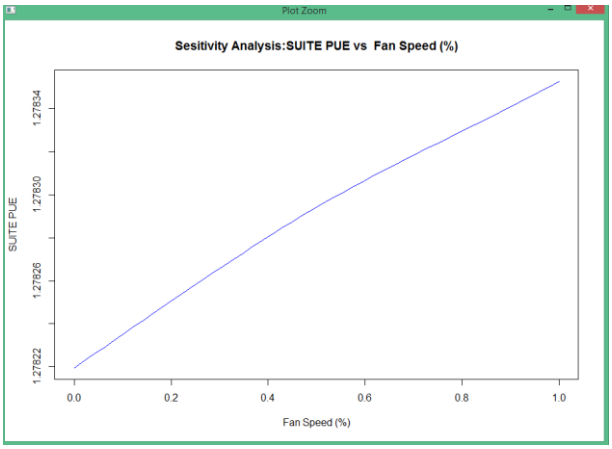


Figure 46: SUITE PUE vs. Fan Speed

Figure 47 signifies that the Absorption (KW) which is the chiller power varies inversely to PUE, as chiller power increases PUE drops. It concludes that it creates a great impact in PUE, which relatively stabilized through the fan power and server load for better synchronization.

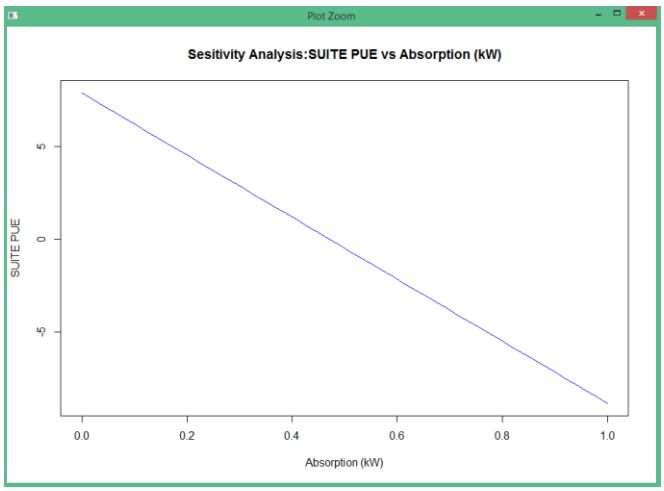


Figure 47: SUITE PUE vs. Absorption

Figure 48 specifies the variation of PUE with Suite Server Load (KW) is linear, which states that the PUE decreases exponentially as the server load decreases. Eventually as per the data samples trained most of the power in the heterogeneous DC station is consumed by server load 78%.

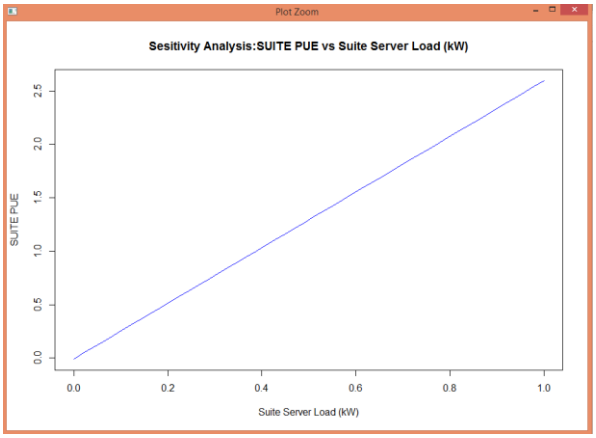


Figure 48: SUITE PUE vs. Suite Server Load

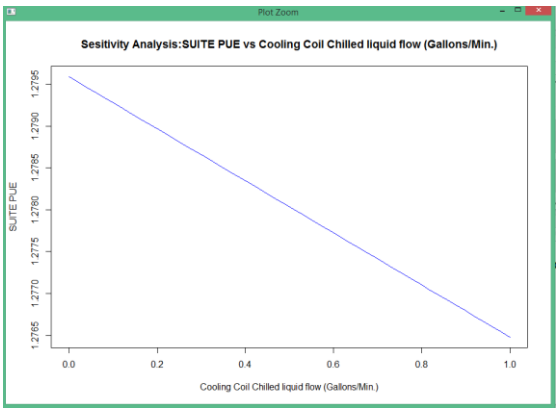


Figure 49: SUITE PUE vs. Cooling Coil Chilled Liquid Flow

Figure 49 represents that as the cooling coil chilled liquid flow increases significantly the SUITE PUE decreases so there is an inversely proportional to each other.

Figure 50 represents a slightly sloppy curve for the SUITE PUE versus Heat reclaim coil leaving temperature (°F), says that PUE is in stabilized state when the temperature is in the optimal stage and also shows that they are inversely proportional as the temperature increases the PUE drops out.

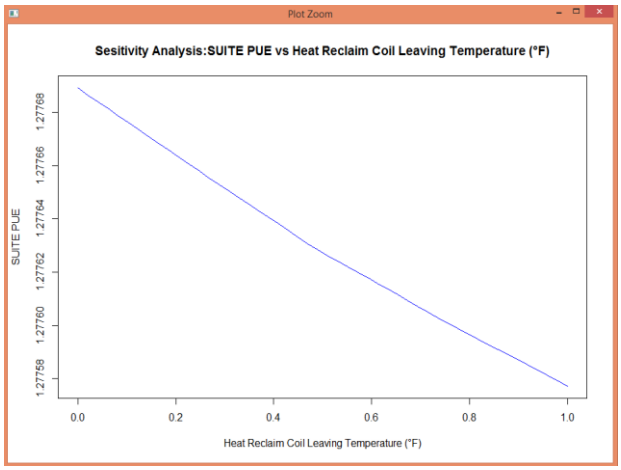


Figure 50: SUITE PUE vs. Heat Reclaim Coil Leaving Temperature

Figure 51 represents that as the Cooling coil leaving temperature (°F) increase the PUE decreases, so the DC should be maintained with increasing the cooling coil leaving temperature with stabilizing other variables and making PUE more effective to reduce the cost. Similarly Figure 52 suggests that the providing a system with cold aisle temperature (°F) over a period of time under different circumstance, the variation in the PUE is linearly increased as the cold aisle temperature decreases.

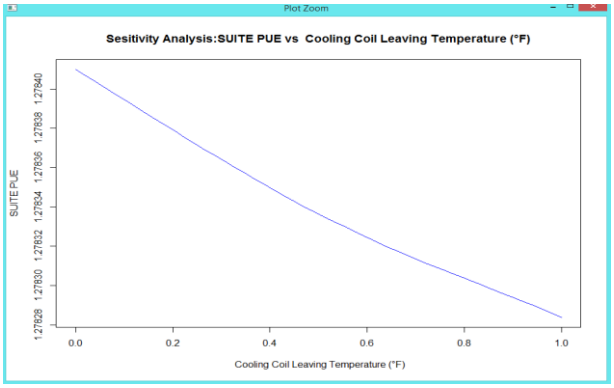


Figure 51: SUITE PUE vs. Cooling Coil Leaving Temperature

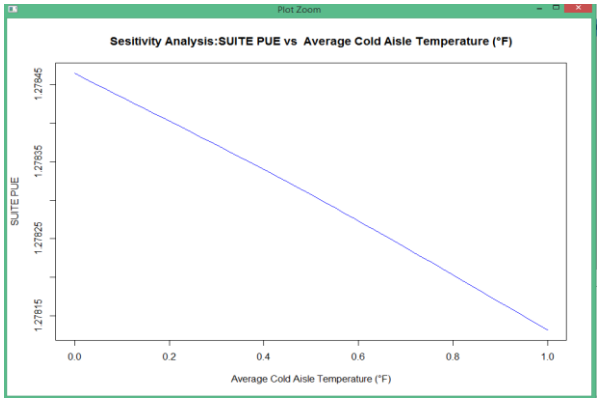


Figure 52: SUITE PUE vs. Average Cold Aisle Temperature

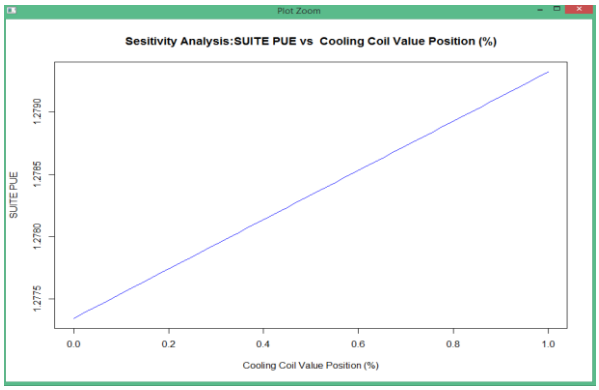


Figure 53: SUITE PUE vs. Cooling Coil Valve Position

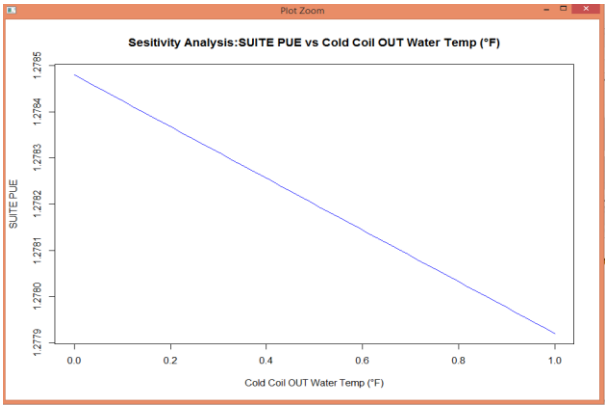


Figure 54: SUITE PUE vs. Cold Coil Out Water Temp

Figure 53 represents a linear variation as the Cooling coil valve position increases the PUE also increases, as it is directly proportional so usage of power is more as it becomes big. Figure 54 indicates that when Cold coil out water temperature decreases eventually the PUE increases, so the temperature for this scenario is optimized and they are inversely proportional to each other.

4.5. Predictive Model for the cooling system

In this section we explain using simulation model to predict optimized output of the cooling system from the neural network trained model as discussed in the section 3.7 Solution Method: Cooling Power Simulation. The figure 55 is plotted chiller Power versus Time samples (10 sec), by keeping constant the Cold Aisle Temperature at 68 ^{o}F. And also a plot with PUE versus time samples (10 sec), by keeping constant the Cold Aisle Temperature at 68 ^{o}F. This optimized value is compared with operational value. In this particular analysis we find that system is operating 6% less efficient.

Similarly, for determining the PUE at same constant cold aisle temperature 68^{o}F, the plot on figure 56 represents the variation in of actual PUE versus the predicted PUE. The difference is 11% gap between the actual versus predicted PUE.

The variation in the Figure 55 and Figure 56 shows large difference between optimized value verses actual operations. In the specific example, it is an unusual behavior; some malfunction has occurred in the heterogeneous DC. The large difference signifies anomalies in the data center systems. Manual intervention identified issues with faulty valve operation. Similarly, other issues can be solved by investigating the datacenter.

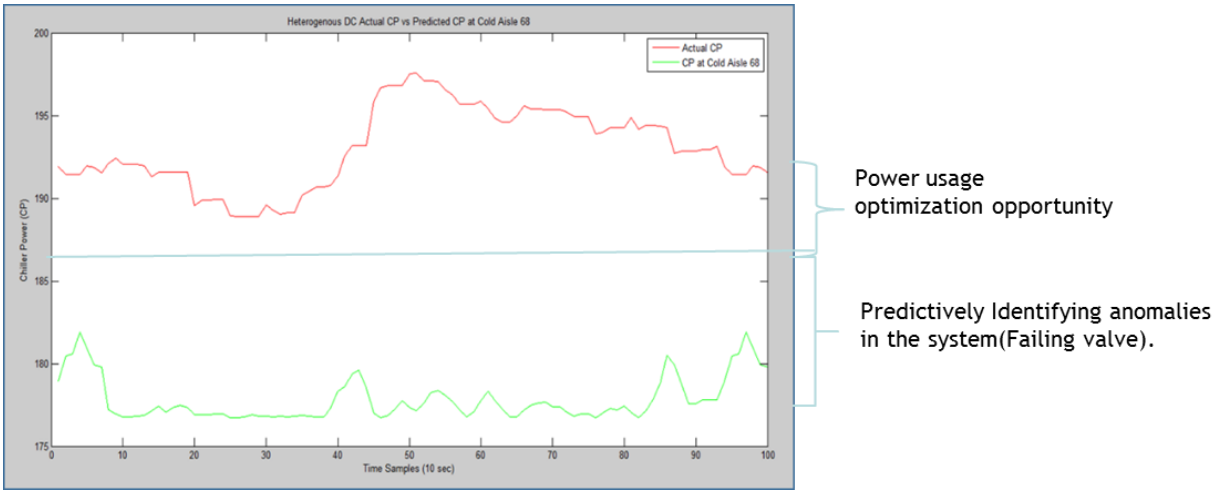


Figure 55: Actual vs. Predicted CP at constant cold aisle 68 degrees F

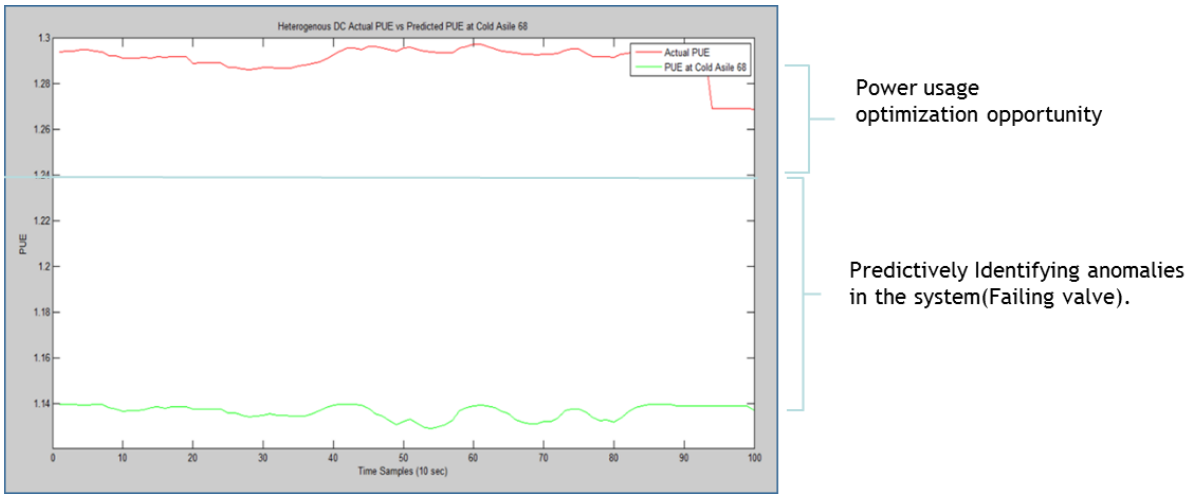


Figure 56: Actual vs. Predicted PUE at constant cold aisle 68 degrees F

The precise and robust PUE model offers many benefits for heterogeneous DC operators and owners. For example, in real time comparison of actual vs. predicted heterogeneous DC performance for any given set of conditions can be used for automatic performance alerting, real-time plant efficiency assessing and troubleshooting.

Comprehensive DC efficiency model enables operators to simulate the DC operating configurations without making physical changes. Currently, it's very difficult for an operator to predict the effect of a plant configuration change on cooling power prior to enacting the changes. This is due to the complexity of modern DCs, and the interactions between multiple control systems. A machine learning approach leverages the plethora of existing sensor data to develop a mathematical model that understands the relationships between operational parameters and the holistic energy efficiency. This type of simulation allows operators to

virtualize the DC for the purpose of identifying optimal plant configurations and predictively identify issues while reducing the uncertainty surrounding plant changes.

4.6. Limitations

- Back propagation neural networks (and many other types of networks) are in a sense the ultimate 'black boxes'. Apart from defining the general architecture of a network and perhaps initially seeding it with a random number, the user has no other role than to feed it input and watch it train and await the output. In fact, it has been said that with back propagation, "you almost don't know what you're doing". Some software freely available software packages (NevProp, bp, Mactivation) do allow the user to sample the networks' progress' at regular time intervals, but the learning itself progresses on its own. The final product of this activity is a trained network that provides no equations or coefficients defining a relationship (as in regression) beyond its own internal mathematics. The network 'IS' the final equation of the relationship.
- Back propagation networks also tend to be slower to train than other types of networks and sometimes require thousands of epochs. If run on a truly parallel computer system this issue is not really a problem, but if the BPNN is being simulated on a standard serial machine (i.e. a single SPARC, Mac or PC) training can take some time. This is because the machines CPU must compute the function of each node and connection separately, which can be problematic in very large networks with a large amount of data [94]. However, the speed of most current machines is such that this is typically not much of an issue.

4.7. Future work: Experimental NN Chiller Power Controller

With repeated experiments and we will be able identify varieties of issues and its resolution. This confidence, the future work will allow to build neural network controllers that can take corrective actions automatically to optimize power. The neural network predictive controller that is implemented in neural network model of a nonlinear plant to predict future plant performance. The controller then calculates the control input that will optimize plant performance over a specified future time horizon. The first step in model predictive control is

to determine the neural network plant model (system identification). Next, the plant model is used by the controller to predict future performance.

This is a functional block diagram shown in figure 57 of Neural Network Controller for getting optimized chiller power, contains 5 blocks such as CP input controller, NN CAT controller, CAT Controller, NN CP controller and CP Output.

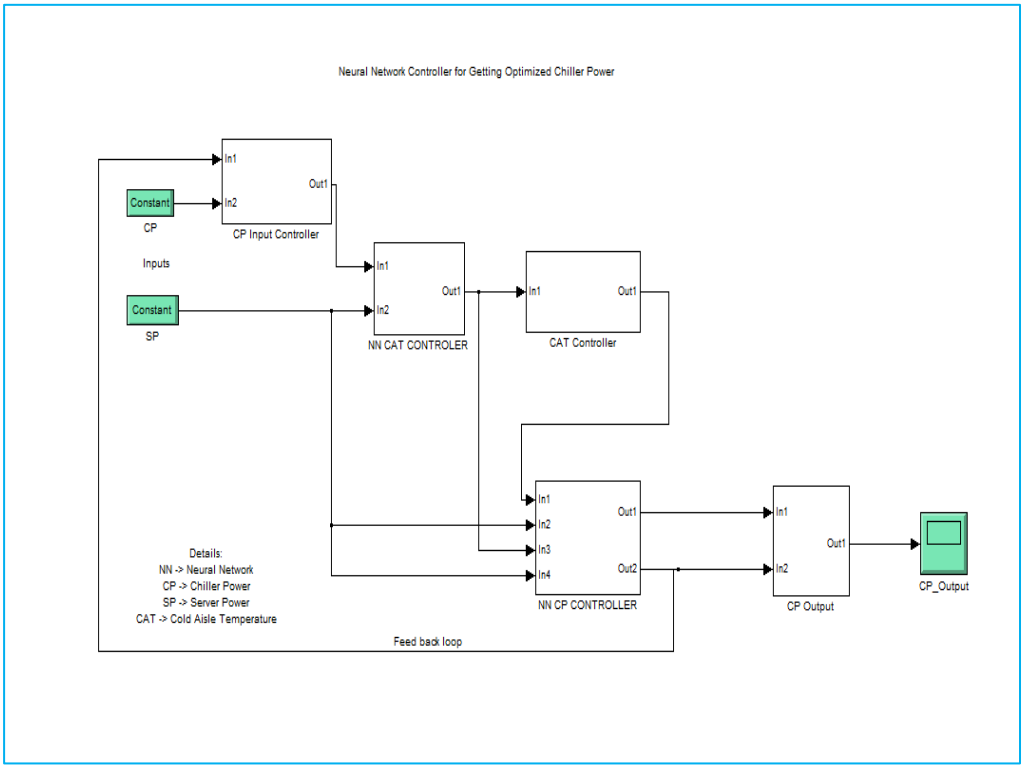


Figure 57: Example chiller power controller

The functions of each block is explained below:

1. The CP Input controller which controls the chiller power

2. NN CAT Controller, a neural network trained model which give optimized Cold Aisle Temperature

3. CAT Controller controls the Cold aisle temperature.

4. NN CP Controller, a neural network trained model which give optimized chiller power

5. CP Output is optimized output obtained.

This controller seems to be not close enough for the deployment, needs additional neural network prediction modules to jointly make it as an excellent framework. This will be a real time controller in the future.

Chapter 5: Conclusions

Accelerating growth in data center complexity and scale is making energy efficiency optimization increasingly important yet difficult to achieve. Additionally, maintaining 100% service levels is a requirement for midsize data centers. Unplanned failure of facility side equipment is detrimental to data center operations. Predictively identifying and preventing issues is becoming more critical for data centers. Fast response to alerted issues and resolving issues in the first attempt is more important than ever.

This thesis presented practical implementation IoT framework deployment in a modular heterogeneous midsize data center, and analyzed data collected from the IoT platform for predictive maintenance, power management and improving operational metrics.

First, thesis reviewed the modular datacenter design and introduces practical design and implementation of the IoT platform. We proved that it is practical to design and implement perception layer, network layer, middle-ware layer and application layer of the IoT framework.

Second, the thesis investigated the use of machine learning techniques. We selected statistically valid variables, train, test and validate the learning engine. We are able to predict DC PUE within 0.0004 +/- 0.0005. We used learning model to optimize power usage. The implementation showed that comparing optimized output with actual operational data predictively alerts on anomalies in the data center facility side systems.

Third, the thesis showed the use of data collected, intelligent alerting and partnering with experience vendors can reduce full time employee overhead. Intelligent alerting to the expert vendors improved response times on addressing issues. Delivering detail machine error codes via alerts can help technician resolve issues in their first attempt.

In conclusion, actual testing on heterogeneous DCs indicates that machine learning is an effective method of using existing sensor data to model DC energy efficiency, predictive alerting, and can yield significant cost savings. Model applications include DC simulation to evaluating new plant configurations, assessing energy efficiency, and identifying optimization opportunities.

The IoT system brings balance between people and technology, approving that compared to independent working the collective working of these two critical components of operation is better. The strengths of people include their agility, their flexibility and their ability to approach a problem from different angles. Similarly, the strengths of technology include its efficiency and its ability to accurately and quickly handle a large number of tasks. The data center aims at these strengths playing off each other through the use of technology for maximizing the accuracy, efficiency and speed of the response time of their personnel and simultaneously adapting logic for increasing the flexibility, agility, and ability to teach of the technology, just the same as their personnel. In an IoT framework, the collective working of people and technology is better.

The use of IoT framework for the collection of data paves way for further interesting work. It enables the creation of intelligent error detection and also a better understanding of the interesting interaction of multiple systems. Additionally, the collection of such large amount of data can be used for training machine learning engines and for the assured creation of machine learning based controllers for achieving real-time efficiencies in the ecosystem of the datacenter.

References

[1] NRDC issue brief. America's Data Centers Are Wasting Huge Amounts of Energy, August 2015 Available: http://www.nrdc.org/Accessed May. 1st, 2016.

[2] J. Timmer, Datacenter Energy Costs Outpacing Hardware Prices: ArsTechnica, 2009. [Online]. Available http://arstechnica.com/business/news/2009/10/datacenter-costs-outpacing-hardware-prices.ars Accessed May. 2rd, 2016.

[3] R. Miller, "How a Good PUE Can Save 10 Megawatts", *Data Center Knowledge*, 2010, vol.13.

[4] Green Datacenter Market in the US 2014-2019, Market to Reach $41 Billion Annually by 2015: Pike Research Press Release, 2015. [Online]. Available http://www.technavio.com/report/green-data-center-market-in-the-us-2015-2019. Accessed May. 1at, 2016.

[5] J. Hamilton, "On Designing and Deploying Internet-Scale Services", In *USENIX Large Installation Systems Administration (LISA)*, 2007.

[6] R. Miller, Microsoft: 300,000 servers in container farm, 2008.

[7] K, Vaid, "Datacenter power efficiency: separating fact from fiction", In invited *talk at the 2010 Workshop on Power Aware Computing and Systems*, 2010.

[8] J. Koomey, "Growth in data center electricity use 2005 to 2010", A report by Analytical Press, completed at the request of The New York Times, 2011.

[9] C. Belady, A. Rawson, J. Pfleuger, and T. Cader, "Green grid data center power efficiency metrics: PUE and DCIE", *Technical report, Green Grid*, 2008.

[10] K. G. Brill, "The invisible crisis in the data center: The economic meltdown of Moore's law", white paper, Uptime Institute, 2007.

[11] R. Miller, Who has the most web servers: Data Center Knowledge, 2009. [Online]. Available http://www. datacenterknowledge. com/archives/2009/05/14/whos-got-the-most-web-servers.

[12] G. Cook, and J. Van Horn, "How dirty is your data? A look at the energy choices that power cloud computing", Greenpeace International, 2012.

[13] X. Fan, W. D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer", in *ACM SIGARCH Computer Architecture News ACM.*, 2007, vol. 35, no. 2, pp. 13-23.

[14] C. Metz, Google admits data center podification, 2009.

[15] A. Vahdat, M. Al-Fares, N. Farrington, R. N. Mysore, G. Porter, and S. Radhakrishnan, “Scale-out networking in the data center”, *IEEE Micro.*, no.4, pp.29-41, 2010.

[16] S.Niles, Standardization and Modularity in Data Center Physical Infrastructure. Schneider Electric, pp.4, 2011.

[17] D. Hornby, B.Walker, and K. Pepple, Consolidation in the data center: simplifying IT environments to reduce total cost of ownership. Pearson Education, 2002.

[18] R. M. Metcalfe, and D.R. Boggs, “Ethernet: Distributed packet switching for local computer networks”, *Communications of the ACM.*, vol.26, no.1, pp.90-95, 1983.

[19] ComputerWeekly.com, Computerweekly.com, 2016. [Online]. Available http://www.computerweekly.com/blogs/database-notes/2010/06/ibm-believes-in-commoditisedhpc- for-bi.html.

[20] J. Erbes, H. R. Motahari-Nezhad, and S. Graupner, “The future of enterprise IT in the cloud”, *Computer*., no.5,pp. 66-72, 2012.

[21] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, and M. Zaharia, “A view of cloud computing”, *Communications of the ACM.*, vol.53, no.4, pp.50-58, 2010.

[22] H. Ning, and Z. Wang, “Future internet of things architecture: like mankind neural system or social organization framework?”, *Communications Letters, IEEE*., vol. 15, no.4, pp.46, 2011.

[23] P. Vlacheas, R. Giaffreda, V. Stavroulaki, D. Kelaidonis, V. Foteinos, G. Poulios, and K.Moessner, “Enabling smart cities through a cognitive management framework for the internet of things”, *Communications Magazine, IEEE.*, vol.51, no.6, pp.102-111, 2013.

[24] M. Yun, and B. Yuxin, “Research on the architecture and key technology of Internet of Things (IoT) applied on smart grid”, in *Advances in Energy Engineering (ICAEE), 2010 International Conference on IEEE*., 2010, pp. 69-72.

[25] H. Suo, J. Wan, C. Zou, and J. Liu, “Security in the internet of things: a review”, in *Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on IEEE*., 2012, vol. 3, pp. 648-651.

[26] F. Günther, and S. Fritsch, “Neuralnet: Training of neural networks”, *The R Journal*, vol.2, no.1, pp.30-38, 2010.

[27] IBM Interconnect 2015: A new way, 2015. [Online]. Available: http://www.slideshare.net/ibm/ibm-interconnect-2015-asset-management-and-the-internet-of-things Accessed May. 2rd, 2016.

[28] Integrated Approach to data Center Power Management Lakshmi Ganesh, Hakim Weatherspoon, Tudor Marian, Kem Birman Computer Science Department, Cornell University, 2012.

[29] Better Data Center Standardization through Pod Architecture Design, 2016. [Online]. Available: http://www.networkcomputing.com/data-centers/better-data-center-standardization-through-pod-architecture-design/55918907 2016 Accessed May. 2rd, 2016.

[30] Google, Machine Learning Applications for Data Center Optimization Jim Gao, 2014.

[31] C. Kelley, and J. Cooley, Deploying and Using Containerized/Modular Data Center Facilities. The Green Grid White Papers, 2011.

[32] Data centers – dcBLOX, dcBLOX, 2016. [Online]. Available: http://dcblox.com/about-us/data-centers-2/Accessed May. 2rd, 2016.

[33] J. Onisick, Better Data Center Standardization through Pod Architecture Design | Network Computing: Networkcomputing.com, 2012. [Online]. Available: http://www.networkcomputing.com/data-centers/better-data-center-standardization-through-pod-architecture-design/55918907 Accessed May. 2rd, 2016.

[34] Schneider Electric, "Andover Continuum - Schneider Electric", *Schneider-electric.com*, 2016. [Online]. Available: http://www.schneider-electric.com/en/product-range/6823-andover-continuum/?filter=business-2-building-management-and-security&parent-category-id=1200. [Accessed: 13- Sep- 2016].

[35] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey",*Computer networks*, vol.54, no.15, pp.2787-2805, 2010.

[36] Gartner, How to Put an Implementable IoT Strategy in Place, 2015. [Online]. Available:http://www.gartner.com/imagesrv/research/iot/pdf/iot-275309.pdf Accessed May. 9th, 2016.

[37] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges", *Ad Hoc Networks*, vol.10, no.7, pp.1497-1516, 2012.

[38] "Crash - Facilities Management Design & Construction Feature", *Facilitiesnet*, 2006. [Online]. Available: http://www.facilitiesnet.com/designconstruction/article/Crash-

Facilities-Management-Design-Construction-Feature--5390. [Accessed: 13- Sep-2016].

[39] F. Xia, L. T. Yang, L. Wang, and A. Vinel, "Internet of things",*International Journal of Communication Systems*, vol.25, no.9, pp.1101, 2012.

[40] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future internet: the internet of things architecture, possible applications and key challenges", in *Frontiers of Information Technology (FIT), 2012 10th International Conference onIEEE.*,2012, pp. 257-260.

[41] M. A. Chaqfeh, and N. Mohamed, "Challenges in middleware solutions for the internet of things", in *Collaboration Technologies and Systems (CTS), 2012 International Conference onIEEE.*, 2012, pp. 21-26.

[42] S., Singh, and N. Singh, "Internet of Things (IoT): Security challenges, business opportunities & reference architecture for E-commerce", in *Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on IEEE.*, 2015, pp. 1577-1581.

[43] R. H. Weber, "Internet of Things–New security and privacy challenges", *Computer Law & Security Review*, vol.26, no.1, pp.23-30, 2010.

[44] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions", *Future Generation Computer Systems*, vol.29, no.7, pp.1645-1660, 2013.

[45] J. Niemann, K. Brown, and V. Avelar, "Impact of hot and cold aisle containment on data center temperature and efficiency", *Schneider Electric Data Center Science Center*, vol.135, pp.1-14, 2011.

[46] R. Raghavendra, P.Ranganathan, V. Talwar, Z. Wang, and X. Zhu, "No power struggles: Coordinated multi-level power management for the data center", in *ACM SIGARCH Computer Architecture News.*, 2008, vol. 36, no. 1, pp. 48-59.

[47] L. Ganesh, H. Weatherspoon, T. Marian, and Birman, K, "Integrated approach to data center power management. Computers", *IEEE Transactions on.*, vol.62, no.6, pp.1086-1096, 2013.

[48] J. Gao, and R. Jamidar, Machine learning applications for data center optimization. Google White Paper, 2014.

[49] P. Bodik, Automating datacenter operations using machine learning, 2010.

[50] J., Zhang, and F.Haghighat, "Development of Artificial Neural Network based heat convection algorithm for thermal simulation of large rectangular cross-sectional area Earth-to-Air Heat Exchangers", *Energy and Buildings*, vol.42, no.4, pp.435-440., 2010.

[51] P. Liu, “Using Random Neural Network for Load Balancing in Data centers”, in *Proceedings on the International Conference on Internet Computing (ICOMP).*, 2015, The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), pp. 3.

[52] S. W.Liao, T. H. Hung, D. Nguyen, C. Chou, C. Tu, and H. Zhou, “Machine learning-based prefetch optimization for data center applications”, in *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis ACM.*, 2009, pp. 56.

[53] C. T. Yang, Y. W. Su, J. C. Liu, and Y. Y. Yang, “Implementation of load balancing method for cloud service with open flow”, in *Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on IEEE.*, 2014, pp. 527-534.

[54] A., Beloglazov, and R. Buyya, “Energy efficient resource management in virtualized cloud data centers”, in *Proceedings of the 2010 10th IEEE/ACM international conference on cluster, cloud and grid computing IEEE Computer Society.*, 2010, pp. 826-831.

[55] A.McGregor, M.Hall, P. Lorier, and J.Brunskill, “Flow clustering using machine learning techniques”, in *Passive and Active Network Measurement.*, 2004, Springer Berlin Heidelberg. pp. 205-214.

[56] R. Das, J. O. Kephart, C. Lefurgy, G. Tesauro, D. W. Levine, and H. Chan, “Autonomic multi-agent management of power and performance in data centers”, in *Proceedings of the 7th international joint conference on Autonomous agents and multi agent systems: industrial track.*, 2008, International Foundation for Autonomous Agents and Multi agent Systems, pp. 107-114.

[57] J. L. Berral, Í. Goiri, R. Nou, F. Julia, J. O. Fitó, J. Guitart, and J. Torres, “Toward Energy-Aware Scheduling Using Machine Learning”, *Energy-Efficient Distributed Computing Systems.*, vol. 215, 2012.

[58] N. K. Srivastava, and S. Mondal, “Predictive maintenance using modified FMECA method”, *International Journal of Productivity and Quality Management*, vol.16, no.3, pp.267-280, 2015.

[59] N. K. Srivastava, and S. Mondal, “Predictive maintenance using FMECA method and NHPP models”, *International Journal of Services and Operations Management*, vol.19, no.3, pp.319-337, 2014.

[60] R. K. Mobley, *An introduction to predictive maintenance*, Butterworth-Heinemann,2002.

[61] K. A. Kaiser, and N. Z. Gebraeel, "Predictive maintenance management using sensor-based degradation models*", Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol.39, no.4, pp.840-849, 2009.

[62] A. Bowling, Development of Artificial Neural Network based Black Box Model of a Data center as a Temperature Predicting Tool as a Function of Server Location. Dissipating Server Heat and CRAC fan speed, 2014.

[63] D. Coakley, P. Raftery, and M. Keane, "A review of methods to match building energy simulation models to measured data", *Renewable and Sustainable Energy Reviews.*, vol.37, pp.123-141, 2014.

[64] Y. Chen, S. Fang, L. Eeckhout, O. Temam, and C. Wu, "Iterative optimization for the data center", *ACM SIGARCH Computer Architecture News.*, vol.40, no.1, pp.49-60, 2012.

[65] A. Jindal, S. B. Lim, S. Radia, and W. L. Chang, *U.S. Patent No. 6,327,622*. Washington, DC: U.S. Patent and Trademark Office, 2001.

[66] S. K. Garg, A. N. Toosi, S. K. Gopalaiyengar, and Buyya, R. "SLA-based virtual machine management for heterogeneous workloads in a cloud datacenter", *Journal of Network and Computer Applications.*, vol.45, pp.108-120, 2014.

[67] P. M. Ferreira, A. E. Ruano, S. Silva, and E. Z. E. Conceicao, "Neural networks based predictive control for thermal comfort and energy savings in public buildings", *Energy and Buildings.*, vol.55, pp.238-251, 2012.

[68] D. Abbott, Applied Predictive Analytics: Principles and Techniques for the Professional Data Analyst. John Wiley & Sons, 2014.

[69] G. Fishman, Discrete-event simulation: modeling, programming, and analysis. Springer Science & Business Media, 2013.

[70] H. S. Sarjoughian, and F. E. Cellier, "Discrete event modeling and simulation technologies: a tapestry of systems and AI-based theories and methodologies", *Springer Science & Business Media*., 2013.

[71] M. Ebrahimi, and R. Whalley, "Analysis, modeling and simulation of stiffness in machine tool drives", *Computers & industrial engineering*., vol. 38, no.1, pp. 93-105, 2000.

[72] L. Luoren, and L. Jinling, "Research of PID control algorithm based on neural network", *Energy Procedia.*, vol.13, pp.6988-6993, 2011.

[73] P. J. Werbos, “Neural networks for control and system identification”, in *Decision and Control, 1989., Proceedings of the 28th IEEE Conference on IEEE.*, 1989, pp. 260-265.

[74] K. S. Narendra, “Adaptive control using neural networks”, *Neural networks for control.*, vol.3, 1990.

[75] K. S. Narendra, and K. Parthasarathy, “Gradient methods for the optimization of dynamical systems containing neural networks”, *Neural Networks, IEEE Transactions on.*, vol.2, no.2, pp.252-262, 1991.

[76] F. Lewis, and S. Ge, "Neural networks in feedback control systems", *Mechanical Engineer’s Handbook*, 2005. [Online]. Available: http://www.pdx.edu/sites/www.pdx.edu.sysc/files/media_assets/SySc576_FrankLewis NNsControl.pdf. [Accessed: 13- Sep- 2016].

[77] J. Walter, S. Spinner, and Kounev, S. “Parallel simulation of queueing petri nets”, in *Proceedings of the 8th International Conference on Simulation Tools and Techniques ICST.*, 2015, pp. 109-118.

[78] C. Delimitrou, S. Sankar, K. Vaid, and C. Kozyrakis, “Decoupling Datacenter Storage Studies from Access to Large-Scale Applications”, *Computer Architecture Letters.*, vol.11, no.2, pp.53-56, 2012.

[79] V. Sridharan, and D. Liberty, “A study of DRAM failures in the field”, inProceedings *of the International Conference on High Performance Computing, Networking, Storage and Analysis IEEE Computer Society Press.*, 2012, pp. 76.

[80] D. Meisner, J. Wu, and T. F. Wenisch, “BigHouse: A simulation infrastructure for data center systems”. in *Performance Analysis of Systems and Software (ISPASS), 2012 IEEE International Symposium on IEEE.*, 2012, pp. 35-45.

[81] P. VenkatRangan, Trust Requirements and performance of a Fast Subtransport-Level Protocal for Secure Communication: IEEE Transactions on Software Engineering, Vol.19 No.2, Feb.1993.

[82] C. Stergiou, and D. *Siganos, Neural Networks*. 2010.

[83] Technopedia, "What is Data Preprocessing? - Definition from Techopedia", *Techopedia.com*, 2016. [Online]. Available: https://www.techopedia.com/definition/14650/data-preprocessing. [Accessed: 13- Sep- 2016].

[84] A. Ethem, *Introduction to Machine Learning*, Second Edition, PHI, 2010.

[85] S. Ray, "A Comprehensive Guide To Data Exploration", *Analytics Vidhya*. N.p., 2016. Web. 21 May 2016.

[86] L. Breiman, and A. Cutler, *Random forests-classification description*, Department of Statistics, Berkeley, 2007.

[87] Search Business Analytics, "What is data sampling? - Definition from WhatIs.com", *SearchBusinessAnalytics*, 2016. [Online]. Available: http://searchbusinessanalytics.techtarget.com/definition/data-sampling. [Accessed: 13- Sep- 2016].

[88] S. Haykin, "*The Neural networks a comprehensive foundation*", New York: Macmillan College Publishing Company, 1994.

[89] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, S. Tarantola, *Global Sensitivity Analysis: The Primer*, John Wiley & Sons, 2008.

[90] D. J. Pannell, "Sensitivity Analysis of Normative Economic Models: Theoretical Framework and Practical Strategies", *Agricultural Economics*., vol.16, pp. 139–152, 1997.

[91] J. A. Freeman and D. M. Skapura, *Neural networks: Algorithm, Applications and Programming teaching techniques*, Addison Wesley, 1991.

[92] D. Lillis, "Generalized Linear Models In R, Part 2: Understanding Model Fit In Logistic Regression Output", Theanalysisfactor.com. N.p., 2016. Web. 21 May 2016.

[93] Ortixx, "What is an epoch in ANN's and how does it translate into code in MATLAB?",*Stackoverflow.com*, 2014. [Online]. Available: http://stackoverflow.com/questions/25887205/what-is-an-epoch-in-anns-and-how-does-it-translate-into-code-in-matlab. [Accessed: 13- Sep- 2016].

[94] University of Wisconsin, "A Basic Introduction To Neural Networks", *Pages.cs.wisc.edu*, 2016. [Online]. Available: http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html. [Accessed: 13- Sep- 2016].