Minor thesis

Keshav – Team – 01

A NOVEL ARCHITECTURE FOR STREAMLINED NETWORK TRAFFIC PROCESSING

Supervisor
Dr. Keshav Sood | Keshav.sood@deakin.edu.au
Mohammad Nosouhi | m.nosouhi@deaki.edu.au

Team
Rahul Ganji | rganji@deakin.edu.au
Vishal Yadav Allegala | vallegala@deakin.edu.au
Sampath Kumar Gudipudi | sgudipudi@deakin.edu.au

SAMPATH KUMAR GUDIPUDI

sgudipudi@deakin.edu.au | 218671901

Table of Contents

Table of Figures

A NOVEL ARCHITECTURE FOR STREAMLINED NETWORK TRAFFIC PROCESSING

Final report

Abstract:

Internet has been evolving and great advancements have come and many more are yet to come. Even with that vast availability of technologies, advanced machinery and powerful sensors we are way behind in their usage, data transmission and processing. So, in this project we are focusing on implementing some better ways for data collection, transmission and its processing. We have experienced many technologies from 2G, 3G, 4G and now 5G is rolling out in many major cities but there are many regional areas are still using 3G signals. We can easily differentiate the data speeds from 3G to 4G. So, with that low data transmission speeds people in those areas are unable to have proper communication and most of the connections drop out with low signals. In this paper we are mainly focusing on three aspects 1. Data collection, 2. Data transmission and Data traffic monitoring this is where Architecture comes into play and 3. Data processing. Data collection is from sensors that are placed in a particular area. Data transmission is via Cloud computing and Edge computing. With unaccepted increase in data consumption during these times we have to be readily equipped with latest technologies like edge computing which has an efficiency of 29.6% over cloud computing and 69.1 % of reduced latency over cloud computing. It has latest defensive systems against energy cloud system attacks (Jararweh, 2020). *Data processing is for anomaly detection for the data transmitted and to bring some conclusions and predictions for future use. Data processing is done using Machine learning algorithms. As we know that Machine learning algorithms are used in many industries like health, weather, IT and many more for data mining* (Yanshan Wang, 2020) *and finding out data patterns. In this we mainly focus on unsupervised algorithms like BRICH algorithm.*

Key words: - Data transmission, Data traffic monitoring, Cloud computing, Edge computing, Machine learning, Data mining, Unsupervised machine learning, Data patterns BRICH algorithm.

Introduction:

This report is mainly focused on data transmission, data monitoring and data processing in real time manner. We have been using cloud services for about quite a few years in data storage, services and computing and we have achieved many advancements and improvements to it. "A Cloud is a system of interconnected and interdependent physical and virtual computer and computing services which is operated based on negotiations between consumer and cloud service providers" (Arvindhan, 2019) Services like Amazon, Netflix, Google and many more have used cloud services and are still using the same technologies. It uses remote server over the internet and provides services on demand. From the time of cloud computing more consumers have added into the list of users as it was fast reliable and secure. It allows the users to access the data at any time and at any place. It has shown us the next stage of computing from its early ages and those stages have been increasing till this time. It benefits the users in outsourcing the services and provides better infrastructure for information (Ratten, 2020). It has reduced investment burden on organizations that relay on them by providing services like infrastructure, software and many more. Cloud computing provides mainly 3 models which are 1. Platform as a service, Infrastructure as a service and software as a service (R. Velumadava rao, 2015).

But now Edge has come into action where it greatly reduces latency, cost and computation and storage problems. Edge computing is concept of deploying IoT devices which are capable of doing all those services of storage, computation and transmission in a distributed way unlike cloud depending on centralised computing where the whole data has to be transferred a central server which far away in other countries, store them and there perform computation on them (Jararweh, 2020). The service and computation will be fast in Edge computing. It will also create jobs in the industry and increases government economy. Coming to Data processing We have seen many papers on static data processing methods using some supervised machine learning algorithms. But these type of methods can be useful to entertainment like services whereas coming to predicting a natural calamity like bush fires in

Australia, Earthquakes and Tsunami we have to have real time data so that the department people can warn the people and reduce the impacts and save people lives. So, we are mainly focussing on real time data transmission and data processing. For data transmission and monitoring we have worked on cloud computing and Edge computing and compared both the result to bring out better understandings and conclude the result. Edge offers many advantages over the cloud computing making it more reliable for service providers to use edge resources. It presents new services and products that will enable new business model for both service providers and systems owners of edge.

And also, on real time data processing and to detect anomalies in log and sensory data we have used Machine learning algorithm to find out anomalies and drew some patterns in the data. We mainly focused on unsupervised learning to detects pattens and anomalies in real time data without humans. As we cannot implement the procedure in real time in this short period of time. We have taken some assumptions and worked it. We have obtained good understandings and result from this thesis.

Rest of the Paper follows as related background work, Literature review, flow of work (Experimental setup), Detailed analysis, results, conclusion and future work and references.

Related Background work

1. Cloud computing vs Edge computing**.**

Most used definition of cloud computing which was introduced by U.S National Institute of Standards and Technology "*A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of services (for example, networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction*".

Edge computing refers to "Edge computing is a new paradigm in which the resources of an edge server are placed at the edge of the Internet, in close proximity to mobile devices, sensors, end users, and the emerging IoT" (Weisong Shi, 2019)

With vast increase in IoT devices the need for edge computing is also raising. One simple example for that is Netflix OTT services. Netflix is a cloud-based service and the only server it has is based in US Virginia. It is Australia's most subscribed OTT service with around 12 Million subscribers. So, every time user uses its services to watch movies or series it has to retrieve that data from US server 52.167.104.17. Let us assume that an average user uses two devices to stream Netflix that total no of devices will be around 24 Million. So, in this case the latency between the server and user will be greatly increased with increase in number of users. So, If Netflix uses Edge technology and deploy an edge server at central location of Australia the burden on US server decreases by large amount. Not only that Storage space can be reduced at centralised location, Security can be increased, Latency can be reduced by greater amount. Moreover, Job opportunities can be created.

Both Cloud and Edge are completely different technologies and they both have their own advantages and disadvantages. These technologies cannot be interchanged. Their requirement varies based on the need. If an organization need time sensitive processing it should use Edge computing where it equipped with intelligent and specialized devices. On the other hand, if Organization or companies which doesn't really bother about processing time uses Cloud technologies (Arora, 2020).

2. Supervised learning vs Unsupervised learning**.**

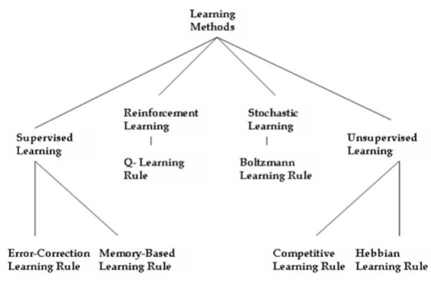


Figure 1 Machine learning models (Sathya.R, 2013)

Supervised learning is Training a data set with preassigned labelled data sets or classified data sets. These techniques are utilized by MultiLayer Perception Models (MLP). MLPs have 3 characters (Sathya.R, 2013)

1. The Hidden layers of data which are not part of input or output will enable the model to learn and find complex patters.
2. It has differentiable nonlinearity in its neuronal activities.
3. Higher degree of connectivity is exhibited by interconnection network model.

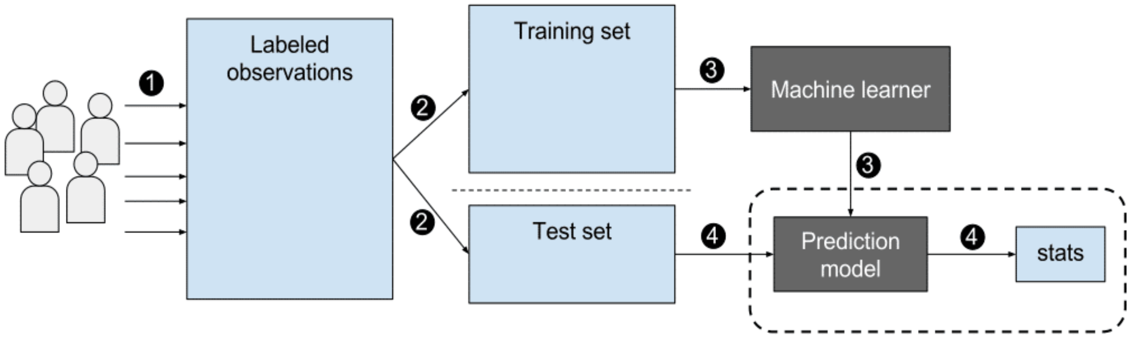


Figure 2 Supervised Learning Model (Salian, 2018)

Unsupervised Learning is based on identifying patterns in unlabelled data sets. It can organize and learn from the information without having an error signal for the solution. It has three phases 1. Competitive phase, 2. Cooperative Phase and 3. Adaptive Phase. (Sathya.R, 2013)

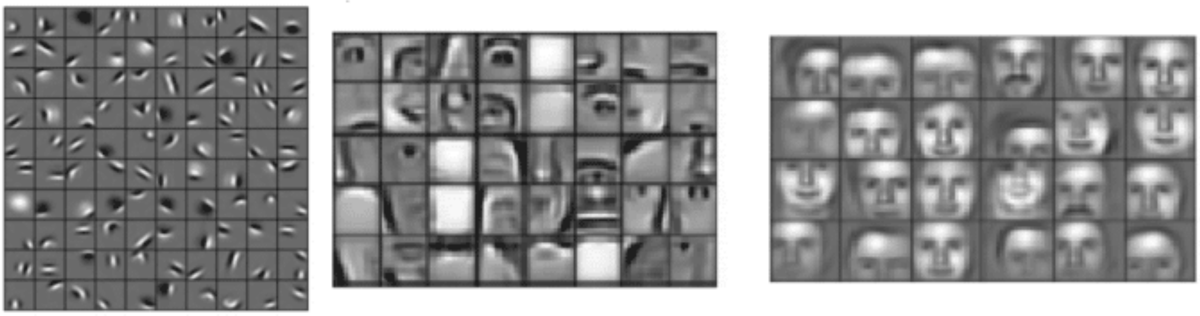


Figure 3 Unsupervised Learning model (Sathya.R, 2013)

It organizes the data in many different ways depending on the requirement 1. Clustering, 2, Anomaly detection, 3. Association and 4. Autoencoders. Yes, It will be difficult for the algorithms to measure accurate measurements or conclusions with unsupervised learning but research fields where data is find or to expensive we can use Unsupervised learning model to produce better quality results (Salian, 2018).

Literature review.

1. Introduction

Out there in the real world we walk across millions of IoT devices and sensors. These devices and sensors do collect huge amount of data with respect to their functionality. There by transferring them to different organizations for processing. We already knew that these IoT devices can be compromised by any intruder or they may collect false or wrong data depending the weather conditions around them. Let us consider a temperature and moisture sensors that are been kept in the soil to collect respective readings each day. There may be some of the sensors that can give produce wrong data due to some effects of the atmosphere around them. This can be caused only to certain number of sensors in a particular geographic area. It is called as error. If it happens with every sensor in that area, then it is called as Event. There has been lot of study going on in finding out these outliers. Not only with the faulty sensors there may be some sort of intruder intervention in the process of data transmission. Which can also be very likely to happen as IoT devices can be compromised very easily. We call those uncommon activities as Network anomalies. Those can be Data breaches, network failures or some sort of attacks on the networks. I want to use these techniques to determine any abnormal activity in the data collection and processing of forest fire data sets. This study can be used to predict some (not accurately) wildfire and help people in those regions to be careful and cautious during those times.

2. Literature Review

i. Outlier detection.

A vast amount of research is taking place in finding the relation of outliers from the collected sensory data using machine learning methods. There are numerous classifications in machine learning like clustering, neighbours using K-means etc. which can be used to identify abnormality in clustered data sets collected from various sensors. Once the data is collected, we divided that data into two parts one is of 80% and other 20% we use bigger data set to train the machine learning model and once the model learnt we pass the other data set to predict the abnormalities in the data set. The first phase of the process is called as training phase and the second is called testing phase. In testing phase, we have two more categories namely 1. Supervised learning and 2. Unsupervised learning. In supervised learning model we'll have prior knowledge about the behaviour on the input and output knowledge. Whereas unsupervised learning we only have knowledge on the input data that to we get it in the testing phase. The previous papers that I have taken to review has used static data sets to train and test datasets. They used fuzzy theory to cluster the data and thereby detect the outliers. They also used distance similarity which reduced communication overhead n the neighbouring nodes (Nashreen Nesa, 2018). Authors Nashreen, Tania and Indrajit have proposed an architecture for the detection of outliers in the IoT environment. They have given definitions for Error and Event. "*Error denotes any noisy data that may have originated from a faulty sensor. They are characterized by any abrupt change in the measurements that differ significantly from the rest of the sample data."* (Nashreen Nesa, 2018) and for the Event "*Event refers to any phenomenon that changes the consistent state of the real-world. Outliers caused by events are found to occur less frequently than that caused by errors [16]. However, the duration of this type of outlier once detected is longer.* (Nashreen Nesa, 2018)*"*

In their approach in find the outlier Error or Events they performed every operation 2 times for the error and one time for the Event because the occurrence of error is very higher than the Event. Even it occurs it's not the evidence that the data is misled or compromised. There are chances for the data to be corrupted but not always. Even though Event occurs less frequently it lasts for long. These authors used four classification models CART, RF, GBM and LDA. They have got great conclusions by performing those machine learning algorithms. (Nashreen Nesa, 2018)

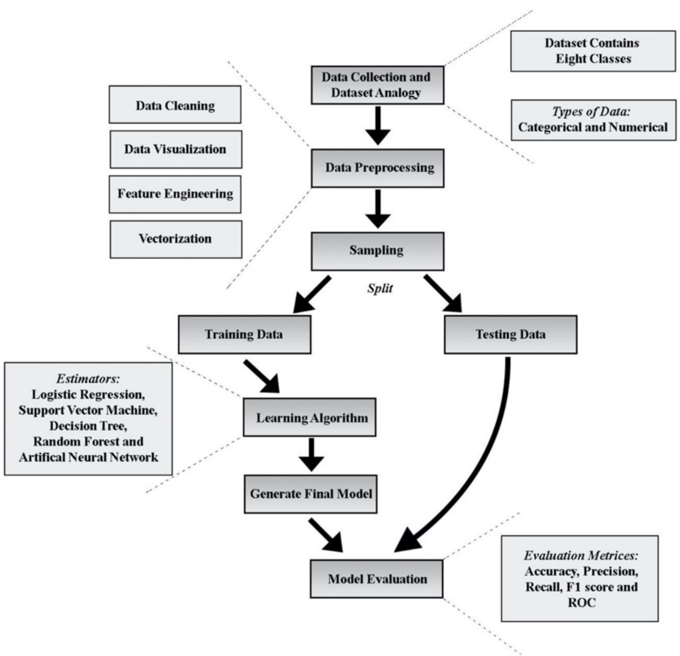


Figure 4 Overall framework for attack and anomaly detection in IoT (Hasan, 2019)

ii. Anomaly Detection

Several researchers have been working on this topic to find out new ways of detections and accurate findings. Phal et al. had worked on microservices and implemented K-means and BIRCH techniques and drew 96.3% of Overall accuracy by updating the clustering technique for the online learning model. DRNN has been introduced to provide clear explanation about smart home security breaches (Pahl, 2018). Liu el at worked on Industrial IoT services and introduced detector for On and Off traffic (X. Liu, 2018). Along with some associates they developed light probe routing mechanism to evaluate trust of every neighbour node in detecting anomaly. Ukil et al. worked on anomaly detection in healthcare based IoT. Likewise, there are numerous numbers of research carried out and will be carried in the future. In the paper I reviewed Mahmudul Hasan have used Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and Artificial Neural Network (ANN). (Hasan, 2019).

Well this also visualised and cleaned using machine learning. Where all the data set is divided in to 80% and 20 % fragments. Where 80% fragment is used to train the machine and 20% is used to testing purpose. The author used Five cross fold validation and came to conclude that RF and ANN have performed accurately and df has shown some similarities. Whereas LR and SVM were very weak in their experiment. In his article he explained very well about the parameters and some mathematical calculations as well. (Hasan, 2019)

iii. Cloud Computing.

It's been more than a decade that we are experiencing Cloud computing technologies. We have seen many advancements and vast acquisition have happened. It has been considered as innovative model for IT services. It enabled Businesses focus on their business and increase productivity. All the enterprises large, medium and small scale are rapidly moving towards cloud computing adoption. As mentioned, Cloud computing has three models 1. Software as a service, 2. Platform as a service and 3. Infrastructure as a service.

1. Software as a Service. It enables users to use providers applications on Cloud platform.
2. Platform as a Service. It enables consumers/users to deploy their applications on the Cloud platforms.
3. Infrastructure as a Service. It enables users to use cloud storage, networks and other computing resources.

Cloud computing services models have 5 essential and common characteristics. They are 1. On-demand self -service, 2. Broad network Access, 3. Resource pooling, 4. Rapid Elasticity and finally 5. Measured service. Also, it has four deployment models 1. Private cloud where cloud services are only for individual organizations, but it can be used my multiple businesses associated with the patent organization. These services can be managed by another third part services. 2. Community Cloud. Where the services that are part of Community cloud can be utilized by any business that comes under the community provinces. 3. Public Cloud. This service can be used by general public. And the last one is Hybrid cloud. It is combination of two are more individual services.

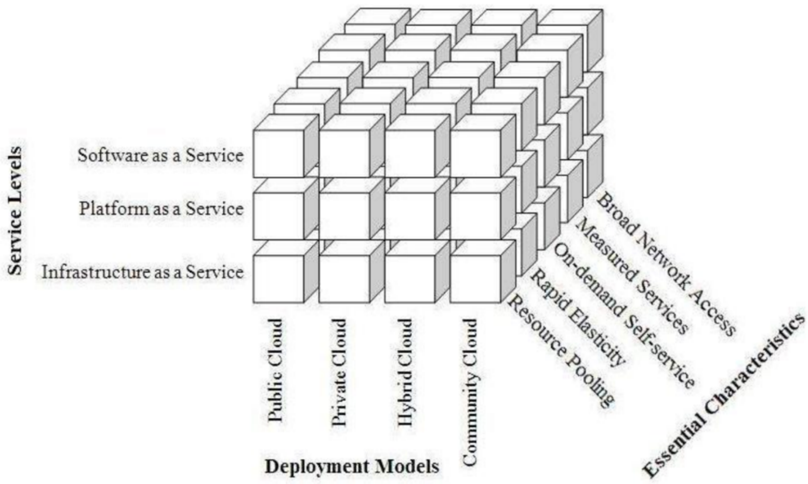


Figure 5 Cloud computing Anatomy (Yang, 2012)

iv. Edge Computing.

Besides Cloud computing Edge computing is also emerging as new technology which offers its services from edge server which is placed at the end of the internet close to its users or consumers. Edge server is not very close to its user but not too far away as cloud server. These edge services are called as Fog, Cloudlets and Micro data centres in which edge hardware is placed. Edge computing is defined as "*technologies allowing computation to be performed at the edge of the network, on downstream data on behalf of cloud services and upstream data on behalf of IoT services*" (Weisong Shi, 2019). The term "Edge" refers to "*computing and network resources along the path between data sources and cloud data centres, and edge is a continuum*" (Weisong Shi, 2019).

It answers all the shortcoming of Cloud computing 1. Latency, 2 Bandwidth, 3. Availability, 4. Energy and 5. Privacy and Security. In edge computing end devices serve as both data consumers and also data producers. These devices also do processing, catching data storage and data distribution services. It provides large amount of data processing at edge server rather than sending all the data to centralized cloud service where there will more latency, require large bandwidth to transfer data to and from in large quantities, there can be security challenges in the air (transmission mode) (Weisong Shi, 2019).

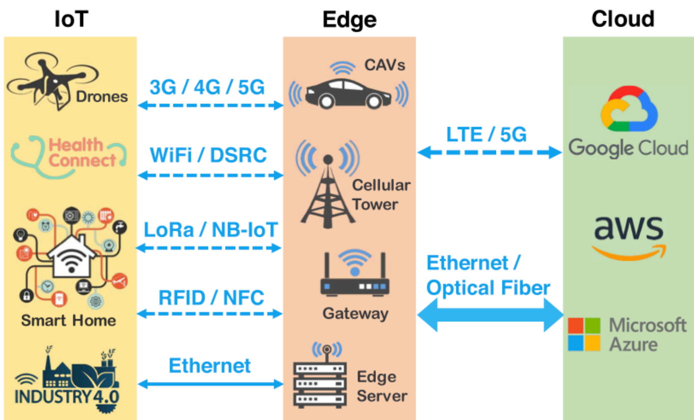


Figure 6 Three Tire edge computing model (Weisong Shi, 2019)

X. Chen et al. proposed a framework model for helpful Mobile Edge registering that joins neighbourhood gadget calculation and organized asset sharing. Dependable collaboration is accomplished between various clients (for example versatile and wearable gadgets clients) through their social connections created utilizing the gadget social chart model. Participation among gadgets helps in preparing and execution of various offloaded undertakings and organization asset sharing utilizing the bipartite coordinating-based calculation (Chen X., 2018). The handling intensity of nearby cell phones and accessible organization assets in closeness are used for an alternate sort of assignment executions plans including, neighbourhood and offloaded errands, for example, D2D, D2D helped cloud and direct cloud. The exploratory outcomes show cost decrease in calculation and undertakings offloading. Nonetheless, in reproduction, the greatest number of gadgets is set to be 500 and for every preliminary the re-enactments are run just multiple times, which is deficient for getting a solid outcome. For the effective reasonable usage of the proposed calculation, the convention must run for normal multiple times and include enormous social networks of gadgets (X. Liu, 2018).

R. Wang et al. proposed a knowledge-centric cellular network (KCE) design to augment the organisation asset usage in learning-based device-to-device (D2D) correspondence frameworks, for example, social and smart transportation D2D networking frameworks. The KCE architecture is of three layers, specifically the physical layer, the Knowledge layer, and the virtual administration layer. The physical layer gathers and deals with the client information, the Knowledge layer deals with the associations between clients in D2D correspondence connection, and the virtual administration layer is answerable for asset assignment in the D2D helped network. The proposed approach appears to be encouraging in vehicle-to-vehicle correspondence application (V2V) whereby data can be transferred through Mobile Edge hubs (Wang R., 2018).

M. Du et al. proposed two privacy-preserving algorithms called Output Perturbation (OPP) and Objective Perturbation (OJP). The authors have distinguished privacy issues in preparing data of machine learning algorithms handled and processed by Edge nodes. Privacy can be exposed using co-related datasets. The proposed algorithms safeguard preparing data privacy by utilizing differential privacy through machine learning. In their proposed method, they included noise in wire-less big data by applying the Laplace mechanism to edge. The proposed algorithms are contrasted and stochastic gradient descent (SGD) and private aggregation of teacher ensembles (PATE-G) utilizing the four datasets CIFAR-10, MNIST, STL-10, and SVHN and TensorFlow. Their proposed algorithms accomplished precise privacy preserving, which additionally implies that a big privacy financial plan is required and that the data utility becomes low (Du M., 2018).

Flow of work

The whole process of set was done under some assumptions as we cannot deploy all the sensors and stuff to collects the data in real time. We assumed our mobile phone as a sensor and some data is transferred from mobile which is considered as first stage of our process which is data collection. The second and third stage es are the main part of our project which is data transmission and data processing. For Cloud services in data storage and transmission we have considered two systems which are 300 kms apart and calculated latency reports for data transmission.

We have used ZeroTire distributed network Hypervisor built on the top of secure peer to peer network. It is one of the advanced and manageable network providers. It has original protocol which is similar to VXLAN and IPSec. But they are closely coupled in the sense of OSI model. One of it is Peer to peer network which is capable of Encryption, authentication and used to create virtual network on dynamic needs. It’s a preconfigured application which is ready to use and fast service. Using (ZeroTire, 2020)these any two devices in world can be connected and communicated instantly. Network configuration for cloud services is below.

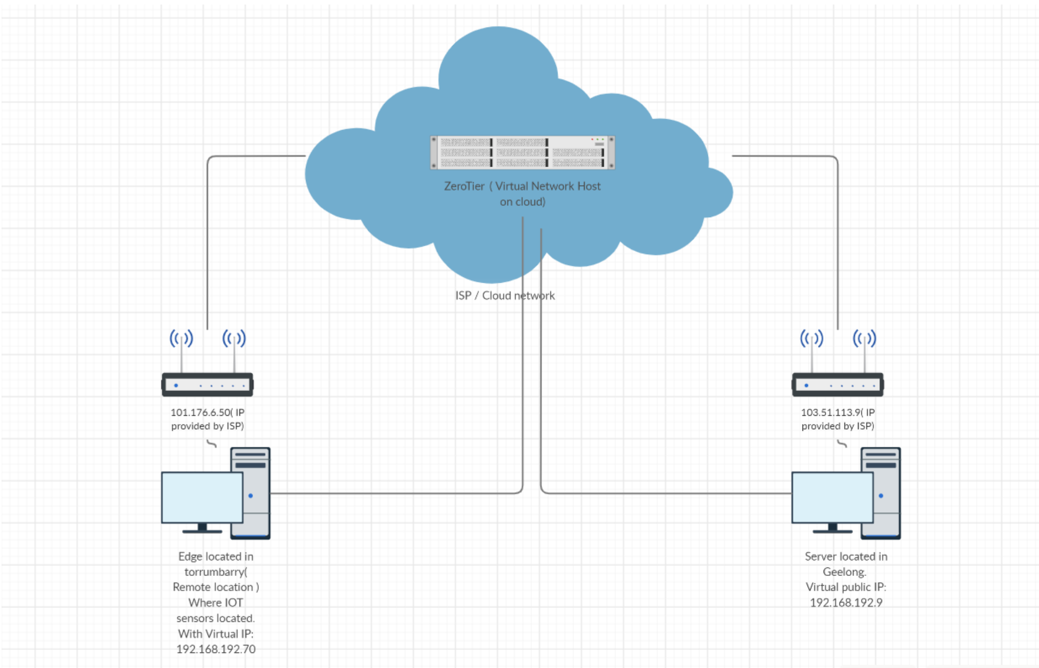


Figure 7 Cloud Network design

Edge Services setup. Its configuration is also similar, but an edge server is deployed in between the sensor and cloud server and all the data storing and processing services are placed at the end of their network. But the cloud server in this context is for providing some extra supports like services and to deal with any other complications. The network diagram for Edge service is given below.

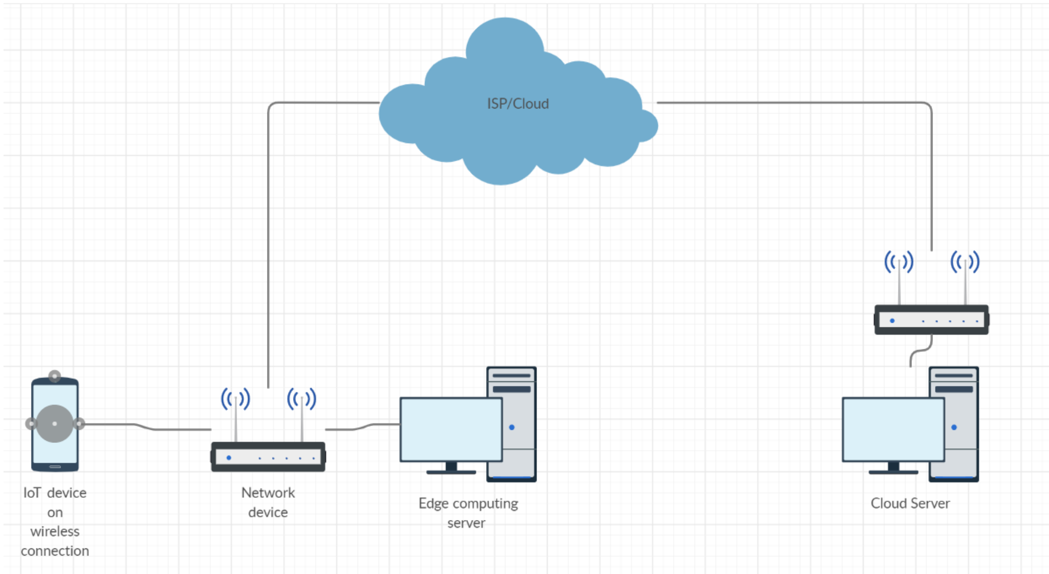


Figure 8 Edge network design

Except the servers all the remaining part is same as the first configuration. Where mobile phones are used as sensors and those data collected is stored and processed in the edge server. Thus, the result is made available to the cloud server so that its result or predictions and processing procedure can be exchanged by other edge servers connected to centralized cloud server. So, the main aim of edge services here is to reduce the processing and transmission (latency) times. Thus, time sensitive processes can be performed, and result can be available much earlier than regular cloud services.

The last phase of our process is data processing. Data processing is for two reasons 1. Find out anomalies in the data and if possible, remove them and also to find out patterns and use those patterns in predictions. We have used Unsupervised Machine Learning algorithm BRICH Algorithm". K-means is one of the most popular Machine learning algorithms but the problem with it is it will not be effective with large data sets as well as with the time factor and quality of the data sets (Alokesh, 2020).

BRICH over comes all these problems where BRICH is Balanced Iterative Reducing and Clustering using Hierarchies. The clustering process of this algorithm is it brings out small but compact summery of larger datasets and they by clusters this small and compact data summery instead of large datasets. The only drawback for this algorithm is that it can be used only to matric attributes that means attributes whose values can be represented in Euclidean space (Alokesh, 2020).

BRICH algorithm has two important features 1. Clustering Feature (CF) and 2. CF – Tree. According to the definition generating small but composed or dense summaries is called as Clustering Feature. It has 3 elements as entries which is called Ordered triple (N, LS, SS). N – number of cluster points, LS – Linear Sum of those data points and SS is Squared Sum of data points in the cluster. CF – Tree is compact representation for a dataset and leaf of the tree is a sub-cluster. CF- tree contains pointes to every child node for every entry (Alokesh, 2020).

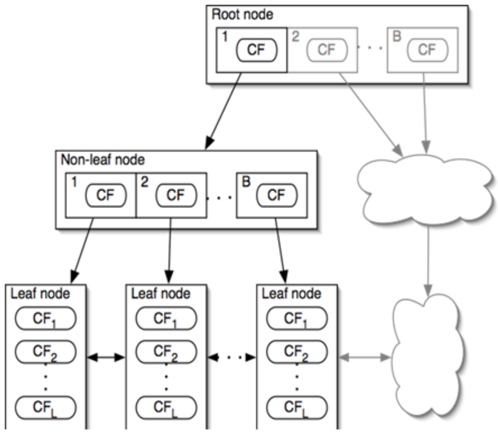


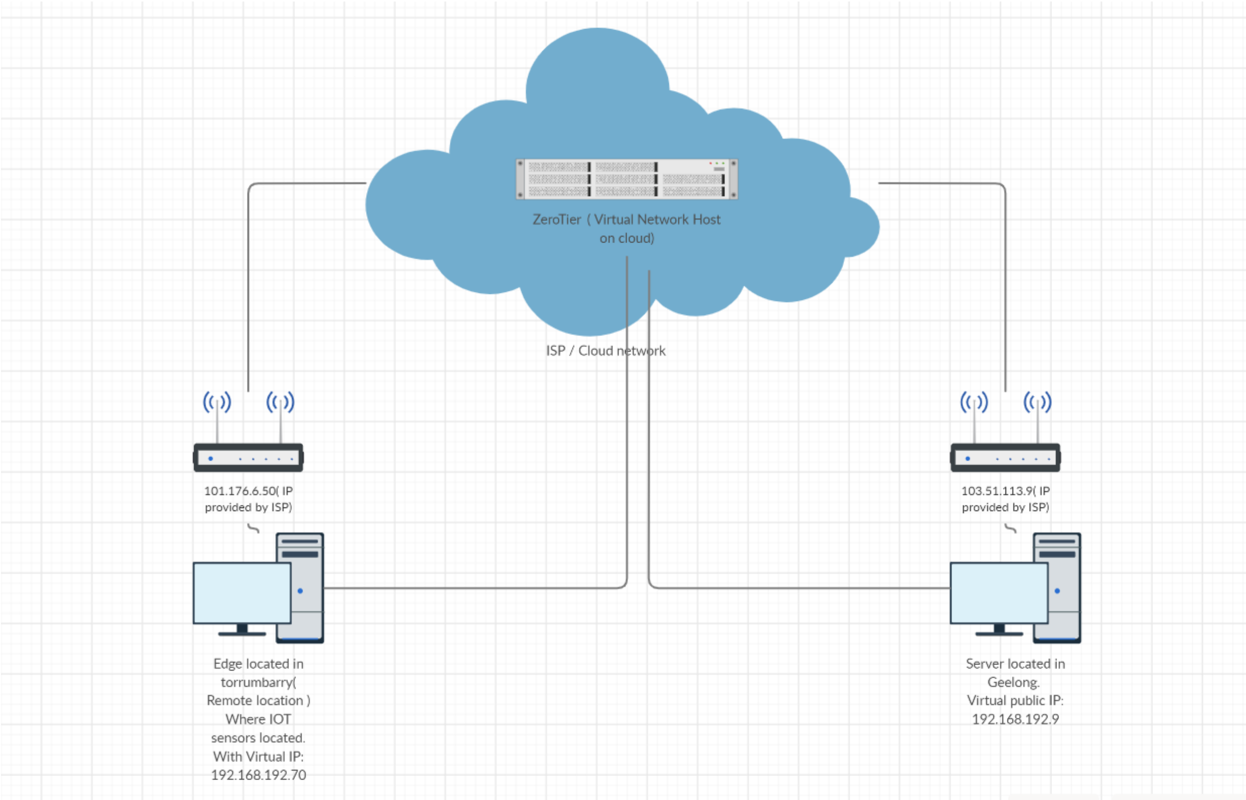
Figure 9 CF - Tree (Maklin, 2019)

BRICH Algorithms parameters are 1. Threshold. It is maximum of data points that CF tree can hold in a subcluster of a leaf node. 2. Branching factor. It is maximum number of sub-clusters in each internal node. And lastly 3. n_clusters. It is final clusters that remains towards the end of clustering (Alokesh, 2020).

Network design and network architecture

In our case we are demonstrating edge and cloud computation, where cloud computation process is already being in existence and still evolving. Where idea of Edge as been introduced in 2014 by cisco and multiple industries are trying to inherit in this system. In cloud all the computation process, storage process is performed at a fixed location and are centralized which is geographically located far from the actual incident process. Dropbox, cloud drives, Netflix are best examples for these services. Where Edge computation process always try to stay closer to actual incident process. In cloud services internet service providers play a key factor whereas in Edge internet service systems will just be a part based on application and operation requirement dependencies.

Architecture to demonstrate cloud services.



As mentioned earlier in cloud services data needs to travel from one geographical location to another geographical location to perform its computations or operations. As this two locations are geographically apart and running on different ISP's ideally there will be no communication between this two location, to establish routing services for traffic low we had created a virtual network configured on cloud using a third party application called ZeroTier and virtual public Ip's are assigned to this different location from the same network to ensure they are connection, even though the real traffic is through ISP's (internet service providers) even with change in hop count the latency between two locations will be the same as shown the below figures. At the same time any monitoring services or real time application services will try to send data in equal intervals of time to process the data. However, to ensure traffic between two locations are monitored through PRTG open source available tool.

```
C:\Users\Rahul Ganji>tracert 103.51.113.9

Tracing route to ip-103-51-113-9.mel.xi.com.au [103.51.113.9]
over a maximum of 30 hops:

  1     1 ms     2 ms     2 ms  TELSTRA-GATEWAY [10.0.0.138]
  2    57 ms    51 ms    53 ms  gateway.vb01.melbourne.asp.telstra.net [58.162.26.193]
  3    47 ms    46 ms    47 ms  ae251.win-ice301.melbourne.telstra.net [203.50.62.178]
  4    57 ms    45 ms    46 ms  ae20-20.lon-ice301.melbourne.telstra.net [203.50.61.130]
  5   110 ms   206 ms    99 ms  bundle-ether25.exi-core10.melbourne.telstra.net [203.50.61.128]
  6    60 ms    56 ms    61 ms  bundle-ether1.lon-edge902.melbourne.telstra.net [203.50.11.112]
  7    55 ms    56 ms    47 ms  voc1255684.lnk.telstra.net [139.130.110.30]
  8    62 ms    57 ms    54 ms  static-202.110.255.49.in-addr.VOCUS.net.au [49.255.112.202]
  9    54 ms    47 ms    48 ms  ip-103-51-112-241.mel.xi.com.au [103.51.112.241]
 10    65 ms    53 ms    47 ms  ip-103-51-113-9.mel.xi.com.au [103.51.113.9]
```

Figure 10 Routing path of traffic before assigning virtual IP's

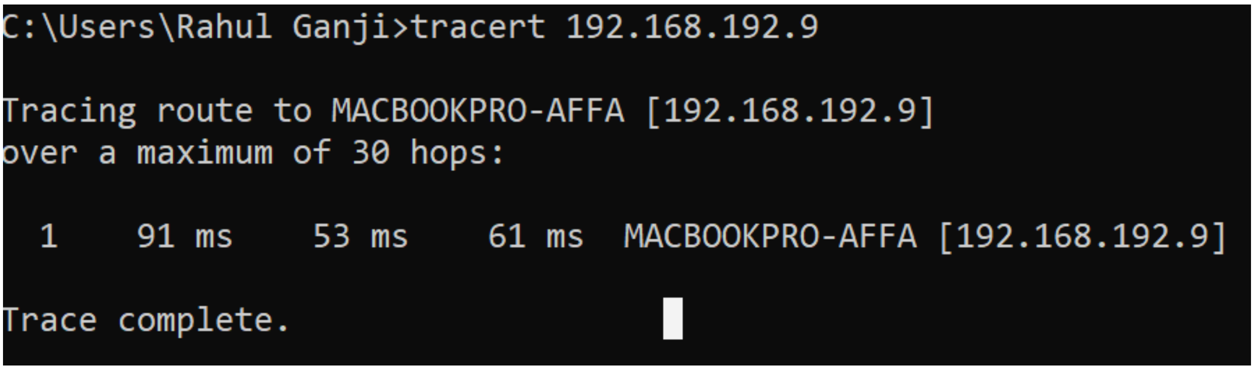


Figure 11 Routing path of traffic after assigning virtual IP's

Let us consider a group of testing centres located in a remote location where an organization has to send and receive from cloud to run their applications. In this case each and every requirement that each equipment has that should send a request to server which is located in cloud. In our test scenario cloud server is located in Geelong which is responsible for computing and storing process whereas actual incident location Is approximately 300 kilometres away from the central hub. As we know the latency between two location depends on the transferring capacity of channel over ISP which will continuously changes based on real time usage.

Let us consider one of the equipment needs some data from cloud, The average latency is around 76 milli seconds, which is ideally applicable for any services to run applications on cloud.

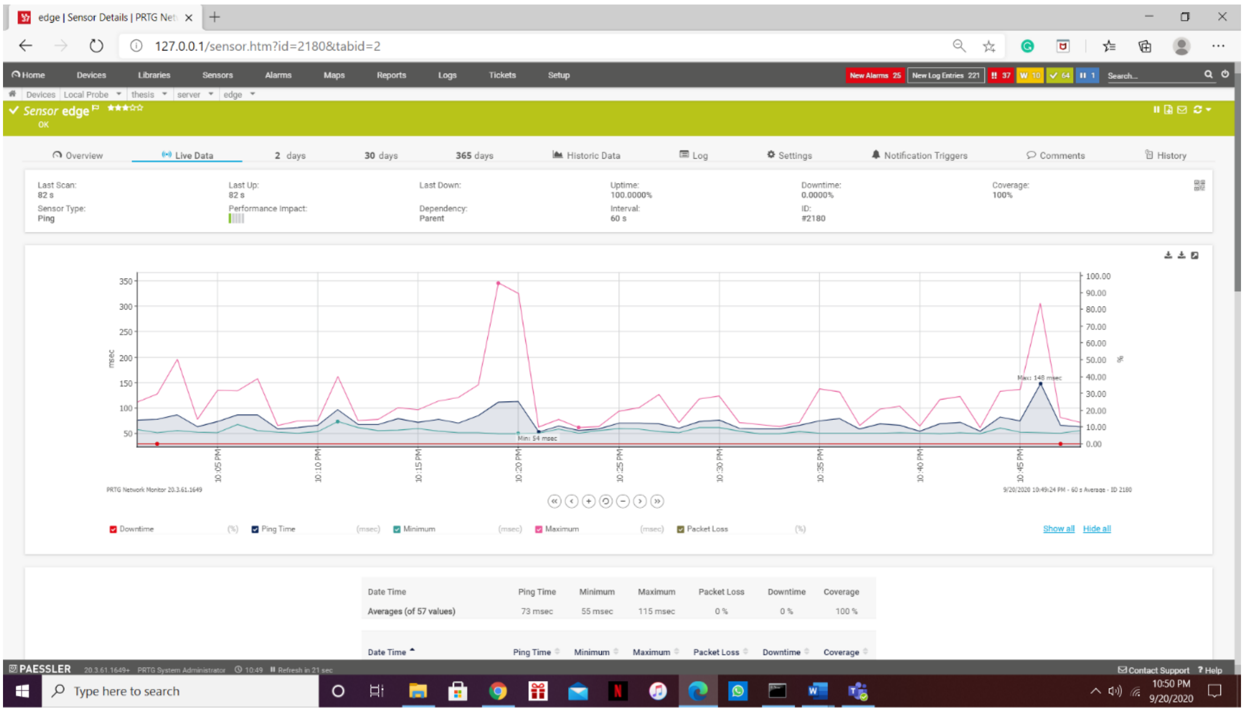


Figure 12 Single sensor to cloud latency graph

Let us consider around 25 equipment's are trying to reach the cloud server for some or other data or for computation process. The average latency is around 86 milli seconds.

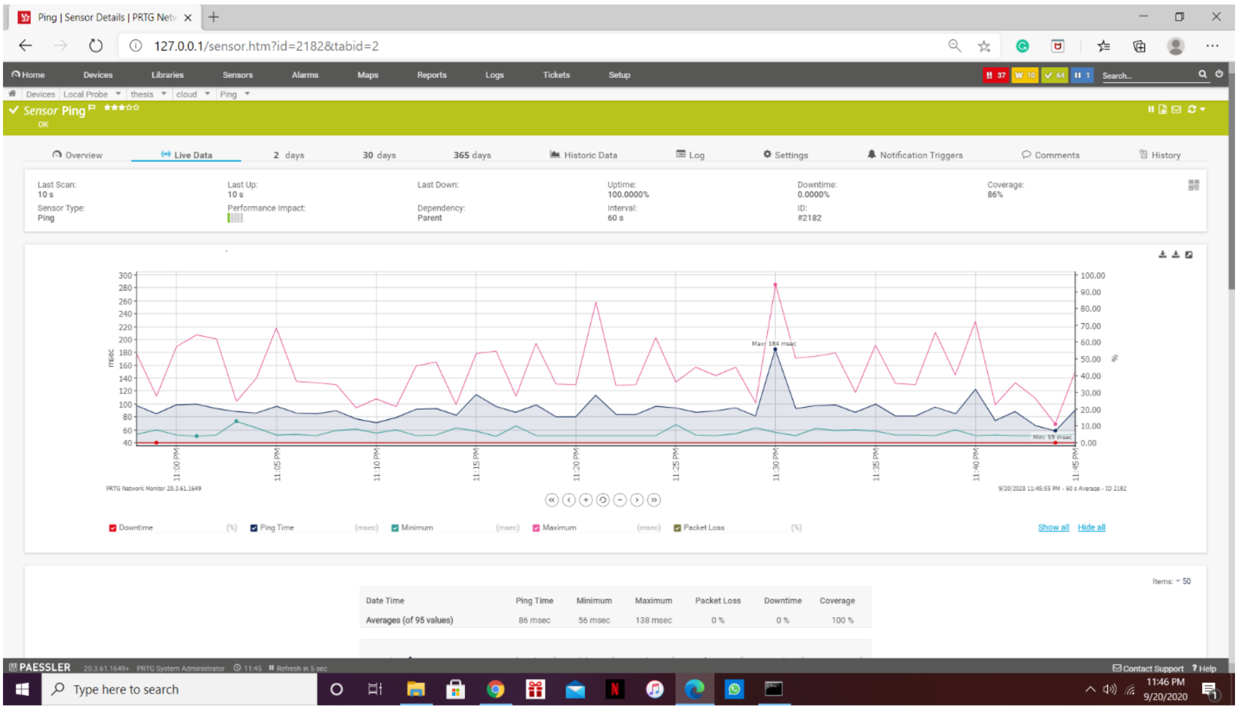


Figure 13 25 sensors data to cloud latency

Let us consider if 45 devices are trying to reach out cloud server at the same time for different sorts of data. The Average latency is around 95 milli seconds.

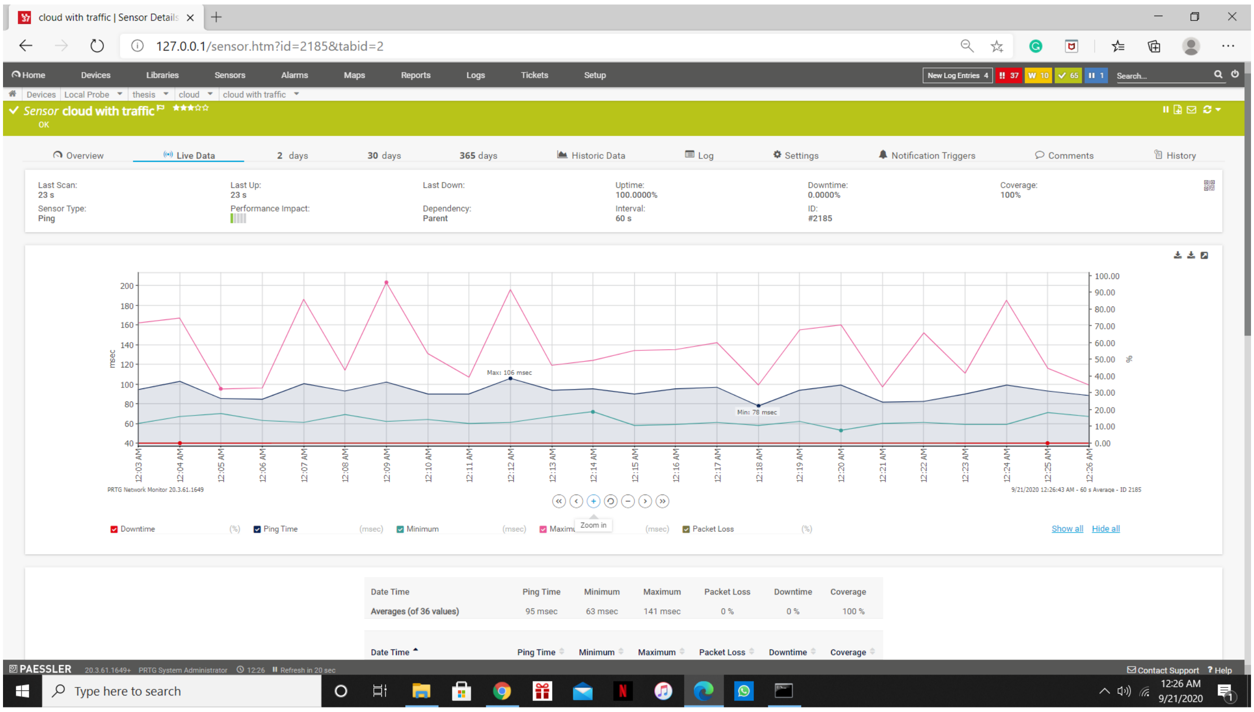
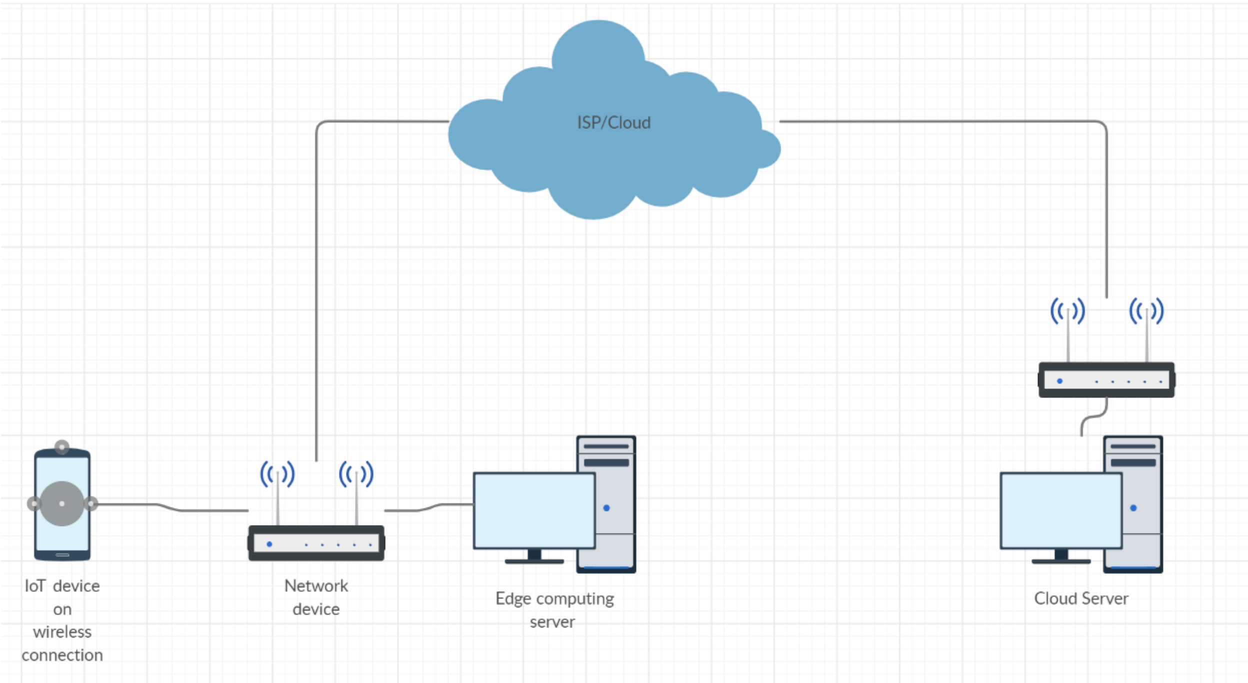


Figure 14 45 sensors data to cloud latency graph

Architecture to demonstrate Edge computing:



In Edge computation process data processing, storing and other services will stay closer to the incident location. Which will ideally rollout internet traffic in many cases or with less amount of data on air. Let us consider the same scenario as that of cloud where a set of service testing team has to send, receive or compute data from server to run operations. As shown in the above figure Edge server is closer or sometimes will be in the same network as that of incident location devices. In case of edge the traffic handling capacity will be fixed and will mainly independent of external factors unlike traffic through internet service providers.

Let us consider a single equipment has to send, receive or process data from edge. The average latency is around 12 milli seconds.

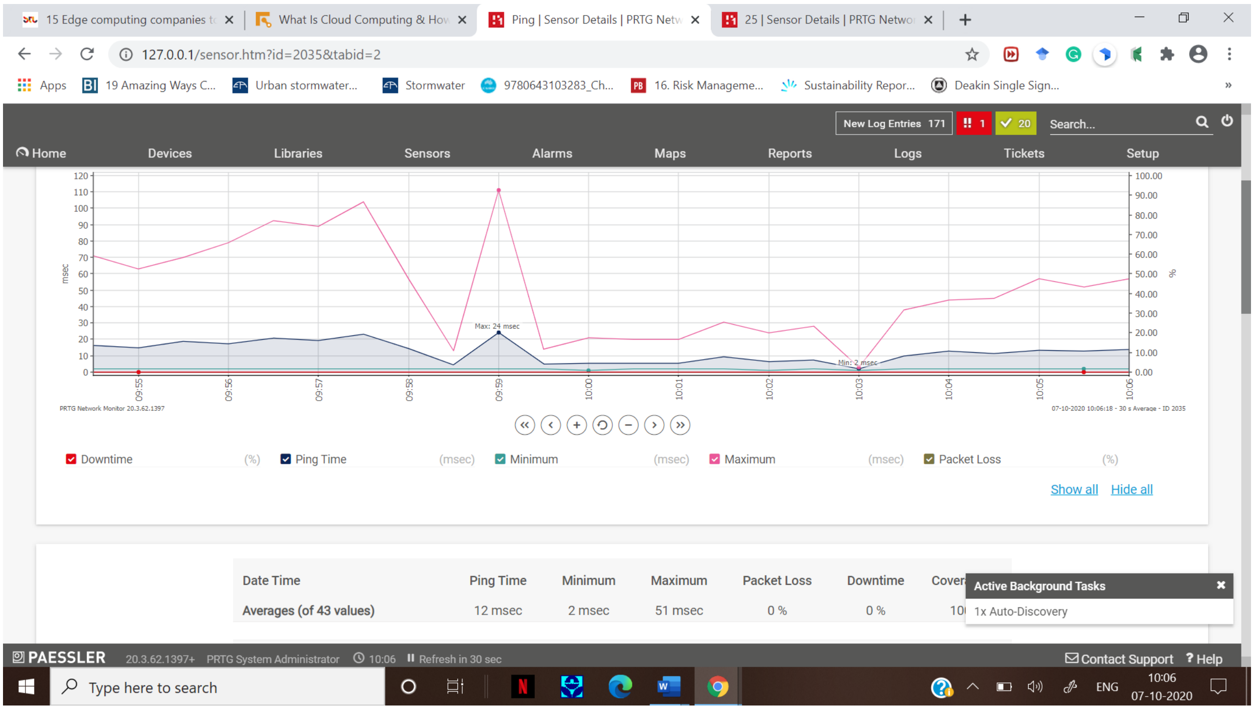


Figure 15 single sensor data to Edge latency graph

Let us consider a 25 equipment has to send, receive or process data from edge server at the same time. The average latency is around 15 milli seconds.

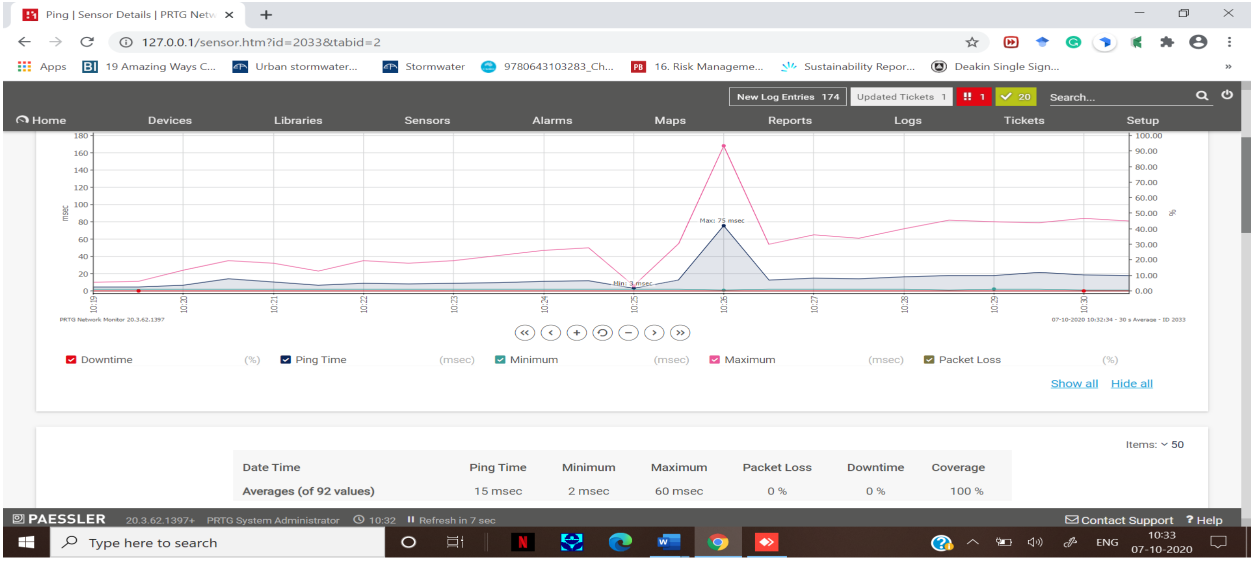


Figure 16 25 sensors to edge latency graph

Let us consider a 45 equipment has to send, receive or process data from edge server at the same time. The average latency is around 34 milli seconds.

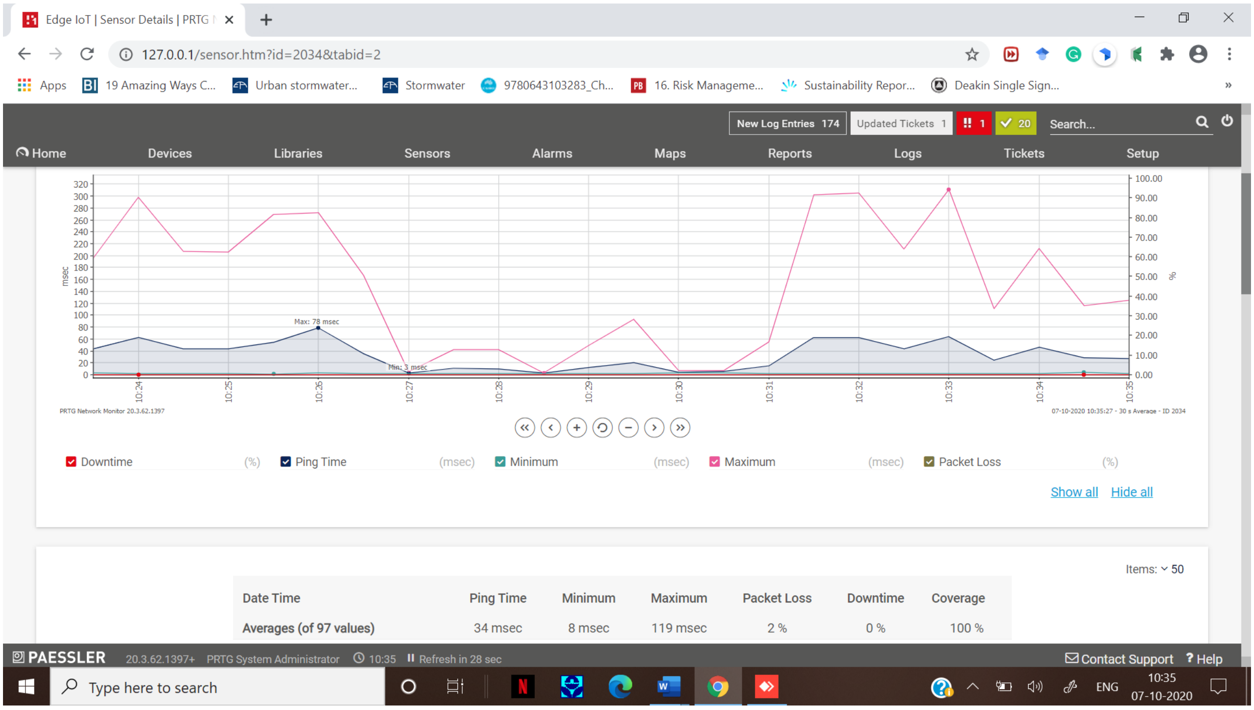


Figure 17 45 sensors data to edge latency graph

Analysis of Edge and cloud computing architecture based on considered scenario.

To perform testing in cloud and edge we had considered the same scenarios and testing is performed with same number of data signals.

Load	Average Cloud latency in milli seconds	Average Edge Latency in milli seconds
Single transmission unit	66	12
25 transmission units	86	15
45 transmission units	95	34

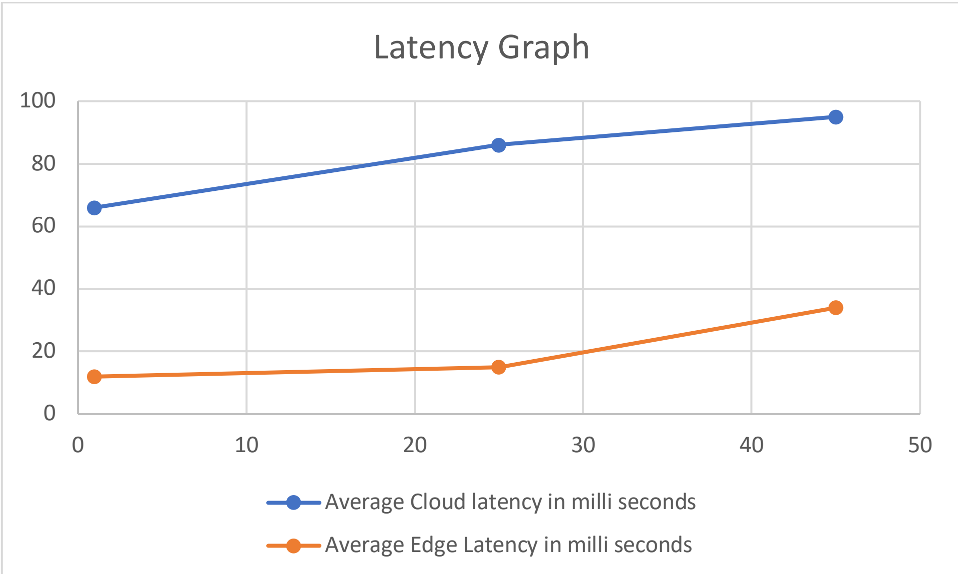


Figure 18 Average latency plots of cloud and Edge services

Based on the above results obtained, it is evident that the latency obtained for a same set of real time data that is processed is Edge and cloud network design are giving us different results and Edge is more efficient in terms if latency compared to cloud.

BRICH algorithm python code *(“I’m unable to submit the python code file in the submissions so I added my code here as text)*

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
import random
from sklearn.model_selection import train_test_split
#from sklearn.linear_model import SGDClassifier
from sklearn.cluster import MiniBatchKMeans
                                                        In [3]:
import warnings
```

```
warnings.filterwarnings("ignore")

sensor_number = 1
```

In [4]:

```
def add_noise(df, col_index, n_outliers, sigma, outlier_start_pos):

    for i in range (0, n_outliers):

        df_noisy = df

        sigma_T = sigma * df_noisy.iloc[outlier_start_pos + i, col_index]
        #noise_T = random.gauss(0, sigma_T)

        df_noisy.iloc[outlier_start_pos + i, col_index] += sigma_T

        ## Changing the target label to 1

        df_noisy.iloc[outlier_start_pos + i, 2] = 1

    return df_noisy
```

In [5]:

```
## Reading the dataset from the CSV file
dataset = pd.read_csv('Weather_dataset.csv', index_col = False)
```

In [6]:

```
df = dataset[dataset['sensor'] == sensor_number]
```

In [7]:

```
# Sorting the measurements based on the date and timestap
df.sort_values(by=['recording_date_time'], inplace=True)
```

In [8]:

```
df = df.drop(['recording_date_time', 'sensor'], axis = 1)
```

In [9]:

```
df.to_csv('new.csv', index = False)
```

In [10]:

```
## How many outliers to create in the dataset
n_outliers = 10
```

In [11]:

```
## First, create outliers for the temperature column
for i in range(5):
    temp_outlier_start_pos = random.randint(0, df.shape[0])
    df_temp_outliered = add_noise(df, 0, n_outliers, 2, temp_outlier_start_
pos)
```

In [12]:

```
## Then, create outliers for the humidity column

for i in range(5):
    hum_outlier_start_pos = random.randint(0, df.shape[0])
```

```
    df_hum_outliered = add_noise(df, 1, n_outliers, 1, hum_outlier_start_po
s)
```

In [13]:

```
## Merging the two outliered datasets into a single one
df_outliered = df_temp_outliered
df_outliered['humidity'] = df_hum_outliered['humidity']
df_outliered['target'] |= df_hum_outliered['target']

#df_outliered = df_outliered.drop(['date', 'time', 'month', 'hour'], axis =
1)
df_outliered.to_csv('new1.csv', index = False)
```

In [14]:

```
## Scaling the "temperature" and "humidity" columns
scaler = StandardScaler()
scaler.fit(df_outliered.iloc[:,0:2])
df_scaled = scaler.transform(df_outliered.iloc[:,0:2])
```

In [15]:

```
# The final dataset
df_outliered_scaled = df_outliered
df_outliered_scaled.iloc[:, 0:2] = df_scaled
```

In [16]:

```
df_outliered_scaled.to_csv('new2.csv', index = False)
```

In [17]:

```
## Splitting the dataset into train and test datasets
X_train, X_test, y_train, y_test = train_test_split(
    df_outliered.drop('target', axis=1), df_outliered['target'],
    test_size=0.2, random_state=0, shuffle = False)
```

In [18]:

```
print(X_train.values[0])
print(y_train.values[0].ravel())
print(X_train.shape)
```

```
[-1.34487797  1.27378057]
[0]
(7008, 2)
```

In [19]:

```
X = X_train.values[0]
print(X_train.values[0].reshape(1,-1))

X = np.array([[1,2]])
print("\nX: ", X)
```

```
[[-1.34487797  1.27378057]]

X:  [[1 2]]
```

In [20]:

```
X = df_outliered_scaled.drop('target', axis=1)

Y = df_outliered_scaled['target']
```

In [21]:

```
#cluster_centres = np.array([[10, 0.7], [25, 0.7], [500, 20]])
KM = MiniBatchKMeans(n_clusters = 5,random_state = 0, batch_size = 7)

l = int(X.shape[0]/7)
for i in range(0, l):
    KM.partial_fit(X.values[7*i:7*i+7])

y_pred = KM.fit_predict(X)
```

In [22]:

```
for i in range(0, X.shape[0]):
    if KM.predict(X.values[i].reshape(1,-1))[0] == 1 or y_test.values[i] ==
1:
        print("\nPrediction: ", KM.predict(X_test.values[i].reshape(1,-1)))
        print("Real: ", y_test.values[i])
```

```
Prediction:  [1]
Real:  0

Prediction:  [1]
Real:  0

Prediction:  [1]
Real:  0

Prediction:  [1]
Real:  0

Prediction:  [1]
Real:  0

Prediction:  [1]
Real:  0

Prediction:  [1]
Real:  0

Prediction:  [1]
```

In [23]:

```
from sklearn.cluster import Birch

BRC = Birch(n_clusters= 3, threshold = 0.6, branching_factor = 5000)

label_pred = []
for i in range(0, X.shape[0]):
    BRC.partial_fit(X.values[i].reshape(1,-1))
```

```
    ## Assigning the current measurement to a cluster and store the result
in
    ## label_pred
    label_pred.append(BRC.predict(X.values[i].reshape(1,-1)))

labels = np.array(label_pred)
```

In [24]:

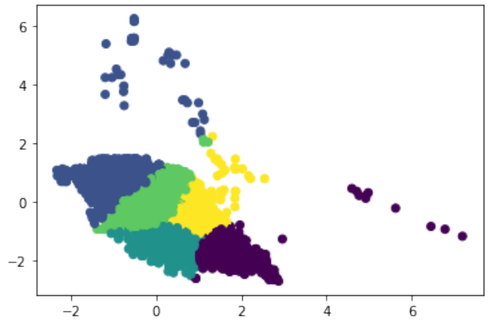
```
columns = ['Prediction']
pred = pd.DataFrame(labels, columns = ['Prediction'])
pred['Real Cluster'] = Y.values

pred.to_csv('Comparison.csv', index = False)


print(pred.head(10))
   Prediction  Real Cluster
0           0             0
1           0             0
2           0             0
3           0             0
4           0             0
5           0             0
6           0             0
7           0             0
8           0             0
9           0             0
```

In [25]:

```
_pred = BRC.predict(X)
plt.scatter(X.iloc[:,0], X.iloc[:,1], c = y_pred)
```

Out[25]:

```
&lt;matplotlib.collections.PathCollection at 0x7fed097df4d0&gt;
```



In [26]:

```
from sklearn.linear_model import SGDClassifier
SGD = SGDClassifier()
## In the first call of partial_fit(), we must mentioned the possible class
es
SGD.partial_fit(X_train.values[0].reshape(1,-1), y_train.values[0].ravel(),
classes = [0,1])
counter = 0
```

```
for i in range(1, X_train.shape[0]):
    SGD.partial_fit(X_train.values[i].reshape(1,-1), y_train.values[i].rave
l())


    y_hat = SGD.predict(X_train.values[i].reshape(1,-1))

    if y_hat[0] == 1 or y_train.values[i] == 1:
        print("Prediction: ", y_hat)
        print("Real: ", y_train.values[i])
        counter +=1


print(SGD.score(X_test, y_test))
print(counter)
Prediction:  [0]
Real:  1
Prediction:  [1]
Real:  1
Prediction:  [1]
Real:  1
Prediction:  [1]
Real:  1
Prediction:  [1]
Real:  1
Prediction:  [1]
Real:  1
Prediction:  [1]
Real:  1
1.0
90”
```

Conclusion

It is now clear that when to adopt Cloud computing and when to adopt edge computing. If there is any time sensitive process it is better to prefer Edge computing rather than Cloud computing. But it is not the only solution we still have to use cloud computing as well other side of Edge. Yes, there are many applications where Cloud overcomes Edge, so Cloud and Edge are two separate technologies. Only some of their drawbacks can be addressed or mitigated by the other technology. Also learnt better ways of using machine learning algorithms in predicting weather conditions to study its patters and changes. This type of studies can be applied to real time data to identify several behaviours in response to the conditions and complications.

Future Research.

It is just the beginning of Edge computing and there are many things to improve to provide better service for the customers. There are many fields that can be improved in Edge computing. Also, there are many challenges that have to addressed on behalf of edge computing. They are

1. Resource Management,
2. Perfect business model,
3. Architecture,
4. Dynamic Billing model,
5. Real time application support,
6. Privacy and security,
7. Storage issues can be addressed.

Even we have just done some assumptions in our experimental study so there is a chance of expanding our experiment to bring out new and real time results or solutions to some of the issues.

We are unable to get the real time data from the forest so the immediate research can be applying our process for the real time datasets. We can use these concepts of cloud and edge computing also Machine learning techniques to learn new patters in the everyday weather conditions, detect and warn people if there is any adverse change. If require we can take lifesaving decisions. Also, Energy of the system is also a new area in both Cloud and Edge computing to explore.

References.

Alokesh, 2020. *Geeks for Geeks.* [Online] https://www.geeksforgeeks.org/ml-birch-clustering/ [Accessed 8 10 2020].

Arora, S., 2020. *Webinar Wrap-up: Edge Computing Vs. Cloud Computing.* s.l.: Simple Learn.

Arvindhan, A. a., 2019. *Data Security and Privacy-Preserving in Cloud Computing Paradigm,* Panjab: IGI GLOBAL.

Chen X., Z. Z. W. W. W. D. Z. J., 2018. Socially-motivated cooperative mobile edge computing. *IEEE,* Volume 32, pp. 177-183.

Du M., W. K. X. Z. Z. Y., 2018. Differential privacy preserving of training model in wireless big data with edge computing. *IEEE.*

Hasan, M., 2019. *Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches,* Bangladesh: Science Direct.

Jararweh, Y., 2020. Enabling efficient and secure energy cloud using edge computing and 5G. *Journal of Parallel and Distributed Computing,* 145(ScienceDirect/ Elesevier), pp. 42-49.

Maklin, C., 2019. *Towards Data science.* [Online] https://towardsdatascience.com/machine-learning-birch-clustering-algorithm-clearly-explained-fb9838cbeed9
[Accessed 8 10 2020].

Nashreen Nesa, T. G. a. I. B., 2018. *Outlier Detection in Sensed Data using Statistical Learning Models for IoT,* West Bengal, India: IEEE.

Pahl, M.-O. P. M.-O., 2018. *All Eyes on You: Distributed Multi-Dimensional IoT Microservice Anomaly Detection,* Rome Italy: IEEE.

R. Velumadava rao, K. S., 2015. Data Security Challenges and Its Solutions in Cloud Computing. *Procedia Computer Science,* 48(Science Direct), pp. 204-209.

Ratten, V., 2020. *Cloud Computing Technology Innovation Advances: A Set of Research Propositions.* Chicago: IGI GLOBAL.

Salian, I., 2018. *blogs.nvidia.* [Online] https://blogs.nvidia.com/blog/2018/08/02/supervised-unsupervised-learning/
[Accessed 07 10 2020].

Sathya.R, A. A., 2013. Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification. *(IJARAI) International Journal of Advanced Research in Artificial Intelligence, ,* Volume 2, pp. 34-38.

Wang R., Y. J. W. D. W. H. Y. Q., 2018. Knowledge-centric Edge computing based on virtualized D2D communication systems. *IEEE communication,* Volume 56, pp. 32-38.

Weisong Shi, G. P. Z. X., 2019. *Edge Computing,* s.l.: IEEE.

X. Liu, Y. L. A. L. a. L. T. Y., 2018. *Defending ON–OFF Attacks Using Light Probing Messages in Smart Sensors for Industrial Communication Systems,* s.l.: IEEE.

Yang, H. a. T. M., 2012. A Descriptive Literature Review and Classification of Cloud Computing Research,. *Communications of the Association for Information Systems ,* Volume 31, pp. 36-61.

Yanshan Wang, Y. Z. T. M. T. E. J. A. A. P. T. N. Z. S. A. A. H. L. S. K. H. L., 2020. Unsupervised machine learning for the discovery of latent disease clusters and patient subgroups using electronic health records. *Journal of Biomedical Informatics,* 102(ELSEVIER).

ZeroTire (2020).