

COMPARING TIME SERIES WITH MACHINE LEARNING BASED PREDICTION APPROACHES FOR VIOLATION MANAGEMENT IN CLOUD SLAs

*A Project Report submitted in partial fulfilment of the requirements
for the award of the degree of*

**Bachelor of Technology
In
Computer Science and Engineering**

By

Nikhil Bansal	(161500353)
Nikunj Gupta	(161500361)
Rishabh Agarwal	(161500446)
Tushar Verma	(161500588)

Under the Guidance of

Mr.Saurabh Singhal, Asst. Professor

**Department of Computer Engineering &
Applications**

Institute of Engineering & Technology



**GLA University
Mathura- 281406, INDIA
May, 2020**



Department of Computer Engineering and Applications
GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,
Chaumuhan, Mathura – 281406 U.P (India)

Declaration

We hereby declare that the work which is being presented in the B.Tech. Project “**Comparing time series with machine learning based prediction approaches for violation management in Cloud SLAs**”, in partial fulfillment of the requirements for the award of the *Bachelor of Technology* in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of our work carried under the supervision of **Mr.Saurabh Singhal, Assistant Professor, Department of CEA, GLA University, Mathura.**

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Sign _____

Name of Candidate: Nikhil Bansal
University Roll No.:161500353

Sign _____

Name of Candidate: Nikunj Gupta
University Roll No.:161500361

Sign _____

Name of Candidate: Rishabh Agarwal
University Roll No.:161500446

Sign _____

Name of Candidate: Tushar Verma
University Roll No.:161500588

Certificate

This is to certify that the above statements made by the candidates are correct to the best of my knowledge and belief.

Supervisor

Mr.Saurabh Singhal (Assistant Professor)
Department of CEA, GLAU, Mathura

Project Coordinator

(Mr. Mayank Srivastava)

Program Co-ordinator

(Mr.Saurabh Anand)

Date:

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B.Tech Major Project undertaken during B.Tech Final Year. This project in itself is an acknowledgement to the inspiration, drive and technical assistance contributed to it by many individuals. This project would never have seen the light of the day without the help and guidance that we have received.

Our heartiest thanks to Dr. (Prof). Anand Singh Jalal, Head of Dept., Department of CEA for providing us with an encouraging platform to develop this project, which thus helped us in shaping our abilities towards a constructive goal.

We owe special debt of gratitude to Mr. Saurabh Singhal, Assistant Professor, Department of CEA, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. He has showered us with all his extensively experienced ideas and insightful comments at virtually all stages of the project & has also taught us about the latest industry-oriented technologies.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for the kind guidance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Sign _____

Name of Candidate: Nikhil Bansal

University Roll No: 161500353

Sign _____

Name of Candidate: Nikunj Gupta

University Roll No: 161500361

Sign _____

Name of Candidate: Rishabh Agarwal

University Roll No: 161500446

Sign _____

Name of Candidate: Tushar Verma

University Roll No: 161500588

ABSTRACT

In cloud computing, service level agreements (SLAs) are legal agreements between a service provider and consumer that contain a list of obligations and commitments which need to be satisfied by both parties during the transaction. From a service provider's perspective, a violation of such a commitment leads to penalties in terms of money and reputation and thus has to be effectively managed. In the literature, this problem has been studied under the domain of cloud service management. One aspect required to manage cloud services after the formation of SLAs is to predict the future Quality of Service (QoS) of cloud parameters to ascertain if they lead to violations. Various approaches in the literature perform this task using different prediction approaches however none of them study the accuracy of each. However, it is important to do this as the results of each prediction approach vary according to the pattern of the input data and selecting an incorrect choice of a prediction algorithm could lead to service violation and penalties. In this, we test and report the accuracy of time series and machine learning-based prediction approaches. Our analysis helps the cloud service provider to choose an appropriate prediction approach (whether time series or machine learning based) and further to utilize the best method depending on input data patterns to obtain an accurate prediction result and better manage their SLAs to avoid violation penalties.

List of Figures

Fig 3.1	System Modules	9
Fig. 3.2	Use Case Diagram	10
Fig. 3.3	Level 0 Data Flow Diagram	11
Fig. 3.4	Level 1 Data Flow Diagram	12
Fig 3.5	Gantt Chart	13
Fig 4.1	Front GUI	22
Fig 4.2	Log Files	23
Fig 4.3	Data Analysis Code (1)	24
Fig 4.4	Data Analysis Code (2)	25
Fig 4.5	Data Analysis Graph	26
Fig 5.1	Data Analysis Graph	27

CONTENTS

Declaration	ii
Certificate	ii
Acknowledge	iii
Abstract	iv
List of figures	v
CHAPTER 1 Introduction	1
1.1 Overview and Motivation	1
1.2 Objective	1
1.3 Scope	2
CHAPTER 2 Software Requirement & Analysis	3
2.1 Feasibility Study	3
2.2 Requirement Analysis	3
2.2.1 System Hardware and Software Specification	3
2.3 Cloud SLAs	3
2.4 Importance of Cloud SLAs	4
2.5 Violation in Cloud SLAs	5
2.6 Penalties on SLA Violation	5
2.7 SLA Levels	6
2.8 Machine Learning	6
2.9 Machine Learning Approaches	7
2.10 Applications of Machine Learning	7
2.11 Machine Learning Methods	8
2.12 Limitations of Machine Learning	8
2.13 Pandas Library	8

CHAPTER 3 Software Design	9
3.1 System Modules	9
3.2 Use Case Diagram	10
3.3 Level 0 Data Flow Diagram (DFD)	11
3.4 Level 1 Data Flow Diagram	12
3.5 Gantt Chart	13
 CHAPTER 4 Implementation and User Interface	 14
CHAPTER 5 Conclusion	27
CHAPTER 6 Summary	28
APPENDICES	
Appendix 1. References	

Chapter 1

Introduction

1.1 OVERVIEW & MOTIVATION

Cloud computing is being adapted by a growing number of individuals and enterprises due to its wide range of services, including the elastic scaling of resources, automatic service deployment and virtualized resources with its benefits of being economical and easily manageable in nature. Due to these features, cloud computing has become the first choice for many small to large organizations. Gartner Research states that cloud computing is a rapidly emerging technology on which organizations spent an estimated \$677 billion from 2013 to 2016. According to a survey Conducted by an IT decision maker for large companies, more than half of the respondents (68%) expected that 50% of their I.T. resources would be migrated to cloud platform.

In cloud computing, service level agreement(SLAs) are legal agreements between a service provider and consumer that contain a list of obligations and commitments which need to be satisfied by both parties during the transaction.

Violation of such a commitment leads to penalties in terms of money and reputation and thus has to be effectively managed.

An SLA is a legal contract which includes service obligations, deliverability, service objectives and service violation penalties. An SLA is not only used to measure the performance of the provider, it also helps to resolve disputes regarding consumer duties. An SLA comprises one or more objectives, called service level objectives (SLO), which comprise one or many low-level metrics.

Violations in SLAs leads to penalties in form of money and loss of reputation and thus it needs to be effectively managed. Our analysis will help to test and report the accuracy of time series based machine learning prediction approaches and help the cloud service provider to choose an appropriate prediction approach and utilize the best method depending on input data patterns.

1.2 OBJECTIVE

We are going to propose a new framework in order to obtain the best method for predicting the QoS based on the input and output pattern, CPU, and memory. This approach ranks the different prediction approaches according to its predicting accuracy, and this determines the SLA

violations. Using this method, the service providers select the method since incorrect adoption causes SLA violation and penalties.

We will test and report the accuracy of the time series and machine learning based prediction approaches. Our analysis helps the cloud service provider to choose an appropriate prediction approach and further to utilize the best method depending on input data patterns to obtain an accurate prediction result and better manage their SLAs and to avoid violation penalties.

1.3 SCOPE

The project will determine the SLA Violations between Cloud Service Provider and Cloud Consumer and on that basis we will calculate the penalties that cloud service provider will have to pay due to the violation of SLA terms and conditions. For more precise results time series algorithm of Machine Learning will be used to calculate the penalties that service provider will pay for the violation of SLA (Service Level Agreement) terms and conditions. This project can be widely used in IT industry to ensure service availability to consumer with respect to all terms and conditions and it will result in increased productivity.

Chapter 2

Software Requirements & Analysis

2.1 FEASIBILITY STUDY

The project is feasible in all aspects as it can be developed easily and is easy to use. We have developed a server and a client which is implemented by using Flask framework. In this project client requests, server responds. Whether server responds or fails to responds, in each case, log file is created displaying whether the server was up or down. So various time series based machine learning algorithm can be applied to log file data to predict SLA violation. On the basis of SLA Violation, Cloud Service Provider will be liable to pay the compensation to its customer.

2.2 REQUIREMENT ANALYSIS

2.2.1 System Hardware and Software Specifications Hardware Specifications:

A PC with following configurations was used for this purpose:

CPU

1. RAM: 4 GB
2. No of cores:4
3. No of threads:8
4. Clock Speed: 2.13GHz
5. Instruction Set: 64 bit

Software Specifications:

1. Python3.6
2. Flask
3. Panda Library

2.3 CLOUD SLAs

A cloud SLA (cloud service-level agreement) is an agreement between a cloud service provider and a customer that ensures a minimum level of service is maintained. It guarantees levels of reliability, availability and responsiveness to systems and applications, while also specifying who will govern when there is a service interruption.

A cloud infrastructure can span geographies, networks and systems that are both physical and virtual. While the exact metrics of a cloud SLA can vary by service provider, the areas covered are uniform: volume and quality of work -- including precision and accuracy -- speed, responsiveness and efficiency. The document aims to establish a mutual understanding of the services, prioritized areas, responsibilities, guarantees and warranties provided by the service provider.

An SLA will commonly use technical definitions that quantify the level of service, such as mean time between failures or mean time to repair, which specifies a target or minimum value for service-level performance.

A typical compute and cloud SLA articulates precise levels of service, as well as the recourse or compensation the user is entitled to should the provider fail to deliver the service as described. Another area to consider carefully is service availability, which specifies the maximum amount of time a read request can take; how many retries are allowed; and soon.

The SLA should also define compensation for users if the specifications aren't met. A cloud storage service provider usually offers a tiered service credit plan that gives users credits based on the discrepancy between SLA specifications and the actual service levels delivered.

2.4 IMPORTANCE OF CLOUD SLAs

- Overview, dates of contract, numbers, cost, and contact information, etc.
- Definition of the service you're receiving.
- The performance you can expect.
- How processes and performance levels will be monitored.
- How data privacy or compliance concerns will be manage.
- How to report any issues to the provider.
- How long it takes to resolve issues.
- A process for how to handle contract changes.

2.5 VIOLATION IN CLOUD SLAs

Cloud prefers or are obliged to have a certain Service Level Agreement (SLA) in place before delivering any service to its valuable customers. The SLA sets a definite time frame in which services are to be delivered to the valuable customers. If they are not delivered within the proposed time frame, SLA violation occurs and Service Provider is imposed by a penalty.

2.6 PENALTIES ON SLA VIOLATION

Service level agreement penalties will vary from contract to contract. When they are being drafted, several parameters for these penalties should be considered. These are:

Service availability: - This involves factors like network uptime, database availability, and data center resources. Penalties should be included as a deterrent against service downtime, as such downtime could negatively affect business productivity.

Service quality: - This involves the guarantee of performance, the number of defects or errors allowed in a product or service, process gaps, and other quality issues. One approach to consider is to levy a penalty for every failure that was made to meet these objectives.

There are a variety of penalties that may be incurred from service level violations. The three most common are:

Financial penalties: - With these, the vendor will be required to pay back to the customer the amount of damages that was agreed upon in the contract. This may not amount to a full reimbursement of the service fee paid by the customer for the job.

Service credits: - With these, the vendor will reimburse the customer for the cost of the work that was done or offer credit for future work to be done. In either event, actually funds are not being transferred.

License extension or support: - With this, the vendor will be required to extend the license's term or offer further support to the customer without charge, which may include development and maintenance.

Such penalties must be set out in the language of the service contract; otherwise, they will not be enforceable. Furthermore, of these penalties, the service credit and license extension penalty may not be considered adequate compensation by some, as some might question the value of receiving the continued services of a provider that fails to meet its quality levels. Rather, employing a combination of penalties may be a better approach, while at the same time including an incentive like a monetary bonus for satisfactory or beyond satisfactory work.

2.7 SLA LEVELS

Service level agreements are also defined at different levels:

- **Customer-based SLA**: An agreement with an individual customer group, covering all the services they use. For example, an SLA between a supplier (IT service provider) and the finance department of a large organization for the services such as finance system, payroll system, billing system, procurement/purchase system, etc.
- **Service-based SLA**: An agreement for all customers using the services being delivered by the service provider. For example:
 - A mobile service provider offers a routine service to all the customers and offers certain maintenance as a part of an offer with the universal charging.
 - An email system for the entire organization. There are chances of difficulties arising in this type of SLA as level of the services being offered may vary for different customers (for example, head office staff may use high-speed LAN connections while local offices may have to use a lower speed leased line).
- **Multilevel SLA**: The SLA is split into the different levels, each addressing different set of customers for the same services, in the same SLA.
 - **Corporate-level SLA**: Covering all the generic service level management (often abbreviated as SLM) issues appropriate to every customer throughout the organization. These issues are likely to be less volatile and so updates (SLA reviews) are less frequently required.
 - **Customer-level SLA**: covering all SLM issues relevant to the particular customer group, regardless of the services being used.
 - **Service-level SLA**: covering all SLM issue relevant to the specific services, in relation to this specific customer group.

2.8 MACHINE LEARNING

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as e-mail filtering and computer vision, where it is difficult or infeasible to develop conventional algorithms to perform the needed tasks.

Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data Mining is a related field of study, focusing on exploratory analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to

help the machine develop its own algorithm, rather than have human programmers specify every needed step.

The discipline of machine learning employs various approaches to help computers learn to accomplish tasks where no fully satisfactory algorithm is available. In cases where vast numbers of potential answers exist, one approach is to label some of the correct answers as valid. This can then be used as training data for the computer to improve the algorithm(s) it uses to determine correct answers. For example, to train a system for the task of digital character recognition, the MNIST dataset has often been used.

2.9 MACHINE LEARNING APPROACHES

Early classifications for machine learning approaches sometimes divided them into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system. These are:-

Supervised Learning: The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs.

Unsupervised learning: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

Reinforcement learning: A computer program interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle or playing a game against an opponent) as it navigates its problem space, the program is provided feedback that's analogous to rewards, which it tries to maximize.

Other approaches or processes have since developed that don't fit neatly into this three-fold categorization, and sometimes more than one is used by the same machine learning system. For example topic modeling, dimensionality reduction or meta learning. As of 2020, deep learning had become the dominant approach for much ongoing work in the field of machine learning.

2.10 APPLICATIONS OF MACHINE LEARNING

There are many applications for machine learning, including:-

- Agriculture
- Anatomy
- Adaptive Websites
- Affective Computing
- Banking
- Bioinformatics
- Cheminformatics
- Citizen Science
- Computer Networks
- Computer Vision

- Data Quality
- Information Retrieval

2.11 MACHINE LEARNING METHODS

- Regression
- Classification
- Clustering
- Dimensionality Reduction
- Ensemble Methods
- Neural Nets and Deep Learning
- Transfer Learning
- Reinforcement Learning
- Natural Language Processing
- Word Embeddings

2.12 LIMITATIONS OF MACHINE LEARNING

Although machine learning has been transformative in some fields, machine-learning programs often fail to deliver expected results. Reasons for this are numerous: lack of (suitable) data, lack of access to the data, data bias, privacy problems, badly chosen tasks and algorithms, wrong tools and people, lack of resources, and evaluation problems.

In 2018, a self-driving car from Uber failed to detect a pedestrian, who was killed after a collision. Attempts to use machine learning in healthcare with the IBM Watson system failed to deliver even after years of time and billions of investment.

2.13 PANDAS LIBRARY

Pandas is quite a game changer when it comes to analyzing data with Python and it is one of the most preferred and widely used tools in data munging/wrangling if not THE most used one. Pandas is an open source, free to use (under a BSD license) and it was originally written by Wes McKinney .

What's cool about Pandas is that it takes data (like a CSV or TSV file, or a SQL database) and creates a Python object with rows and columns called data frame that looks very similar to table in a statistical software (think Excel or SPSS for example. People who are familiar with R would see similarities to R too). This is so much easier to work with in comparison to working with lists and/or dictionaries through for loops or list comprehension.

Chapter 3

Software Design

3.1 SYSTEM MODULES

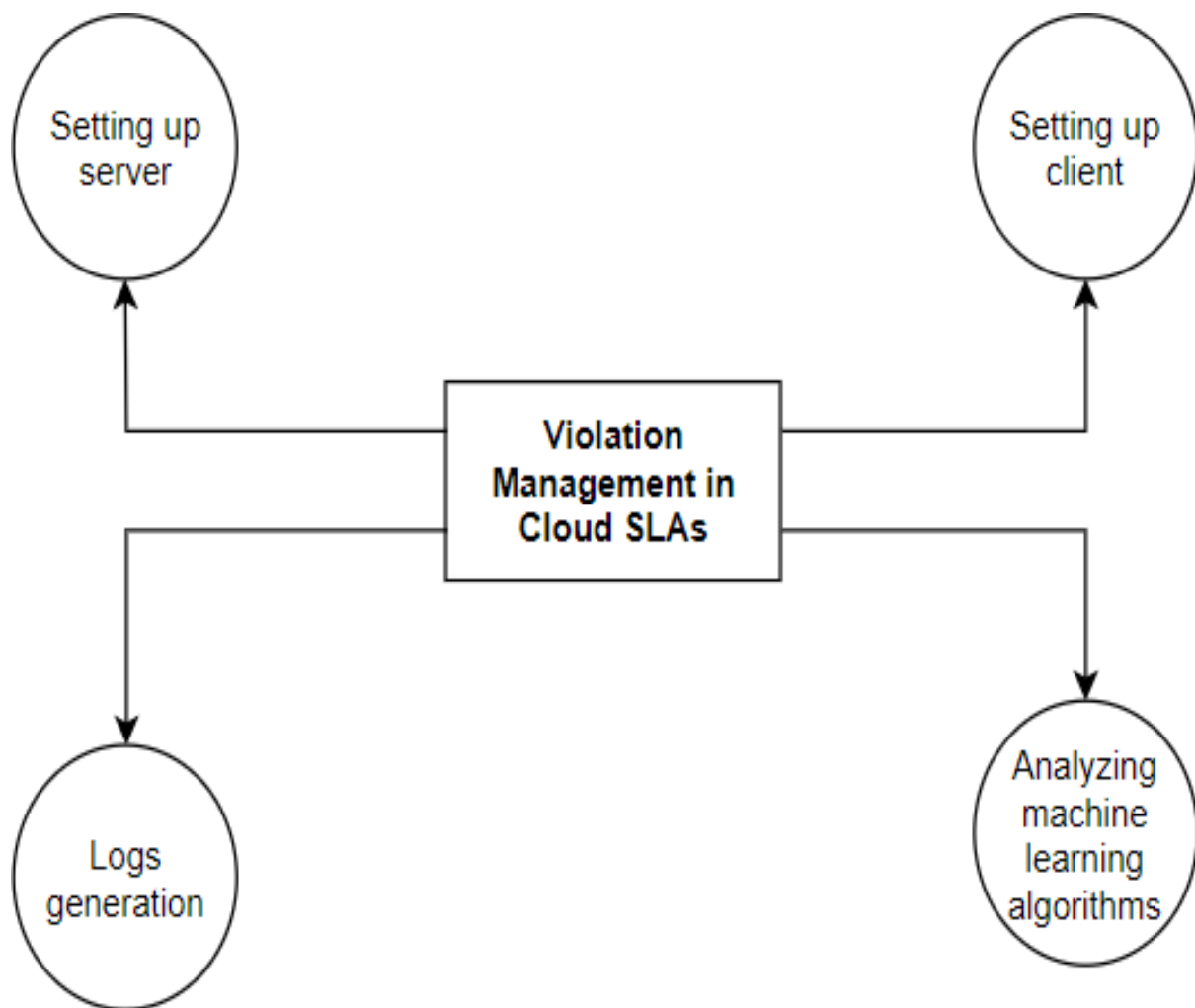


Fig 3.1: System Modules

3.2 USE CASE DIAGRAM

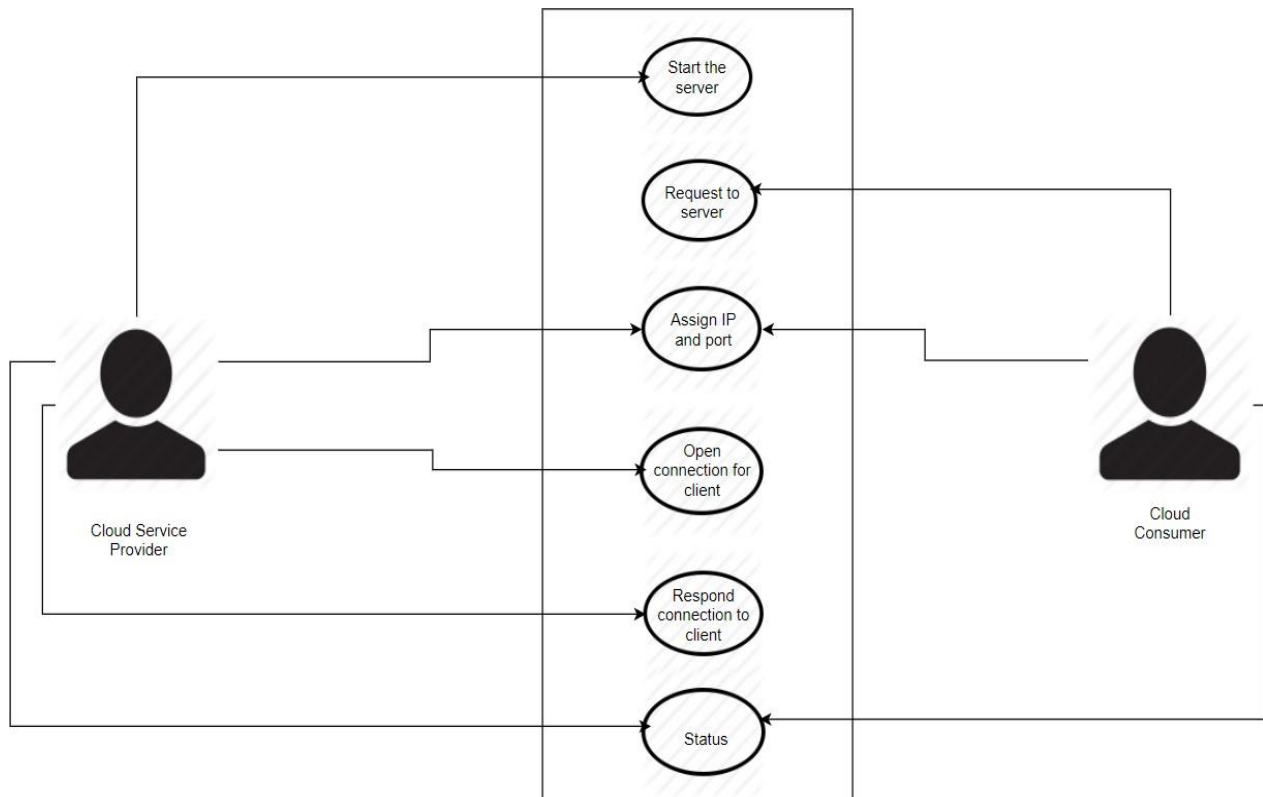


Fig. 3.2: Use Case Diagram

3.3 LEVEL 0 DATA FLOW DIAGRAM (DFD)

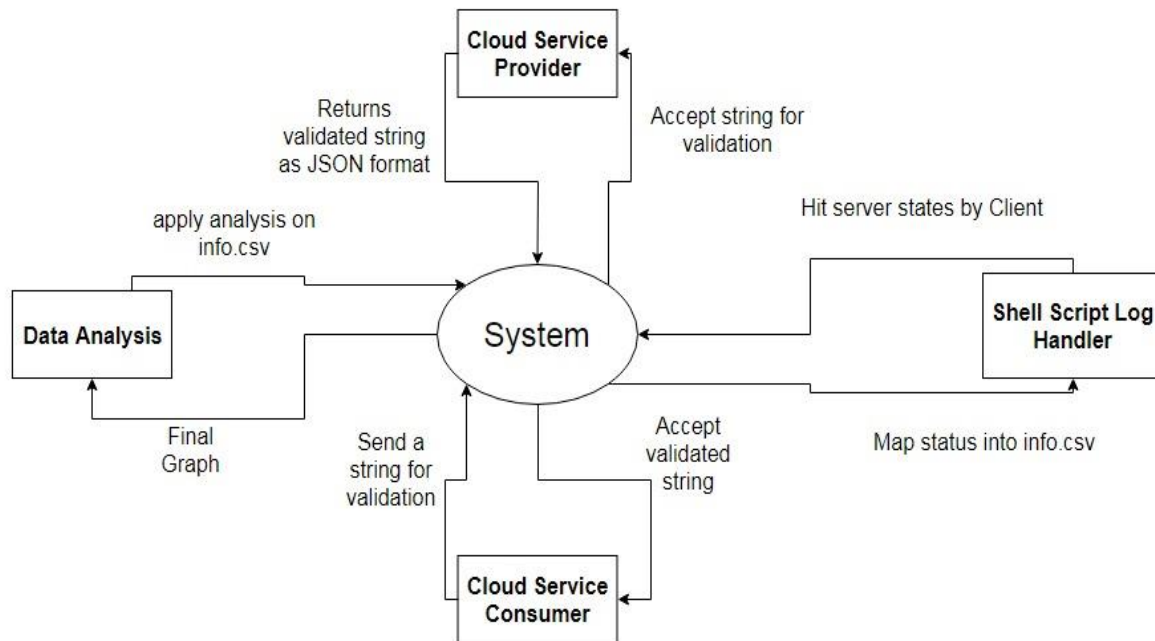


Fig. 3.3: Level 0 Data Flow Diagram

3.4 LEVEL 1 DATA FLOW DIAGRAM (DFD)

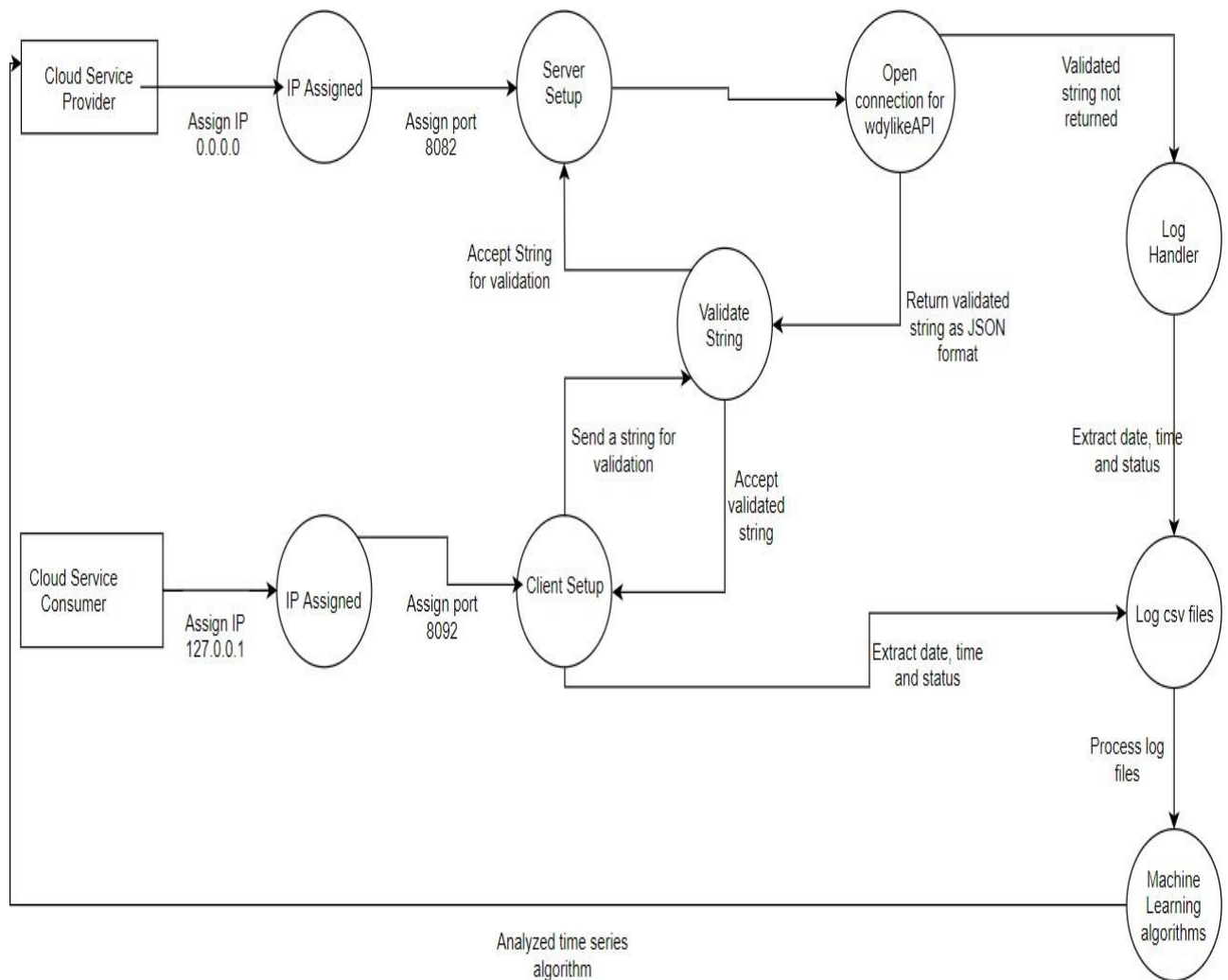


Fig. 3.4: Level 1 Data Flow Diagram

3.5 GANTT CHART

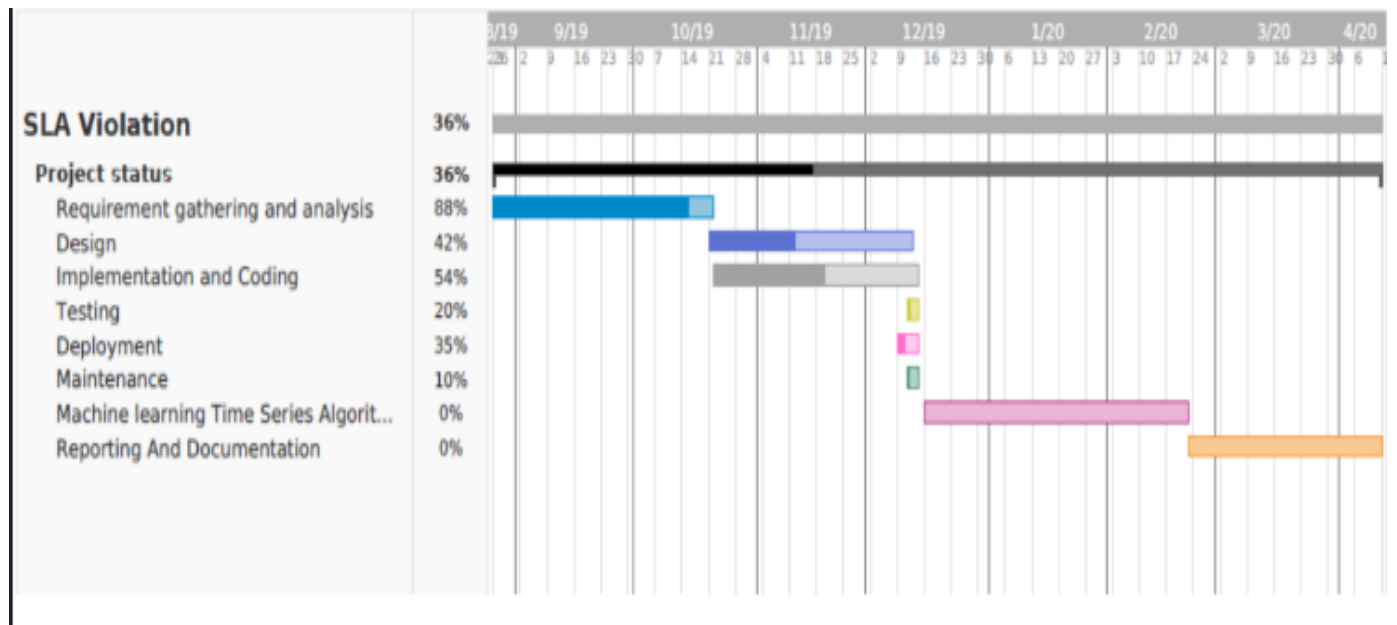


Fig 3.5: Gantt Chart

Chapter 4

Implementation & User Interface

Server Code: (Cloud Service Provider)

Framework Used

Flask Version 1.1.2

Modules Used:

-
1. urllib request: for JSON response operations
 2. regular expression: for string operations

Routes:

-
1. home: renders index.html where user enter the string for validation
 2. document: renders documentation.html the user guide for integration
 3. filtered: return the validate string response to the user

Templates:

-
1. index.html
 2. documentation.html

Server Configuration:

-
1. IP Address Host: 0.0.0.0
 2. IP Address Port: 8080
 3. Status of Debug Mode: True

Input:

String Entered by the user.

Output:

return censored string to the user by removing abusive words.

Server Activate: (Cloud Service Provider)

Server Configuration:

1. IP Address Host: 0.0.0.0
2. IP Address Port: 8080
3. Status of Debug Mode: True

Activate the virtual environment

-> Activate the Virtual Environment by following commands

 \$ source bin activate

 (env) \$

->By this we enable our server code to access the pre-installed dependencies.

Server Running: (Cloud Service Provider)

Server Configuration:

1. IP Address Host: 0.0.0.0
2. IP Address Port: 8080
3. Status of Debug Mode: True

Activate the virtual environment

```
->$ source bin activate  
->(env) $
```

Command to run server

```
->$ python3 Index.py
```

Other Information

Type of server loading: Lazy Loading
Running on: 0.0.0.0:8080
Environment: Production
Type of server: development
Debugger Pin: 223-091-036

Server output: (Cloud Service Provider)

End point for accessing the service

0.0.0.0:8082/filtered?inputpara=

0.0.0.0 : IP Address

8082 : Port number

filtered : Route

input para: Variable for get method

Example

0.0.0.0:8082/filtered?inputpara=this+is+shit

Output

Format: JSON

```
{  
  "message": "this is ****"  
}
```

JSON key: "message"

JSON value: "this is ****"

Client Code: (Cloud Service Consumer)

Framework Used

Flask Version 1.1.2

Modules Used:

-
1. urllib request : for JSON response operations
 2. regular expression : for string operations

Routes:

-
1. home : renders index.html where user enter the string for validation
 2. documen : renders documentation.html the user guide for integration
 3. filtered : return the validate string response to the user

Templates:

-
1. index.html
 2. documentation.html

Client Configuration:

-
1. IP Address Host: 127.0.0.1
 2. IP Address Port: 8092
 3. Status of Debug Mode: True

Response Status Code:

-
1. Equal to 200: Server Up with numerical status 1
 2. Not Equal to 200: Server Down with numerical status 2

Log Data Information:

Format Used: "%{asctime}\$, %{message}\$" , "%Y-%m-%d,%H:%M:%S"
No of Columns: 4
Column Details: Date_Stamp, Time_Stamp, Server_Alpha_Status, Server_Numerical_Status
Backup Count: 1

Client Activate: (Cloud Service Consumer)

Client Configuration:

1. IP Address Host: 127.0.0.1
2. IP Address Port: 8092
3. Status of Debug Mode: True

Activate the virtual environment

-> Activate the Virtual Environment by following commands

\$ source bin activate

(env) \$

->By this we enable our client code to access the pre-installed dependencies.

Client Running: (Cloud Service Consumer)

Client Configuration:

1. IP Address Host: 127.0.0.1
2. IP Address Port: 8092
3. Status of Debug Mode: True

Activate the virtual environment

->\$ source bin activate
->(env) \$

Command to run server

->\$ python3 client.py

Other Information

Type of client server loading: Lazy Loading
Running on: 127.0.0.1:8092
Environment: Production
Type of server: development
Environment: Production
Debugger Pin: 223-091-836

Running server, client and shell script:-

Client Configuration:

1. IP Address Host: 127.0.0.1
2. IP Address Port: 8092
3. Status of Debug Mode: True

Server Configuration:

1. IP Address Host: 0.0.0.0
2. IP Address Port: 8082
3. Status of Debug Mode: True

Client Running Command

->\$ python3 client.py

Server Running Command

->\$ python3 Index.py

Shell Script Running Command

->\$./script.sh

Sample Log Entry of client application

```
shit
[2019-12-13 13:06:33,095] WARNING in client: serverup, 1
127.0.0.1 -- [12/DEC/2019 13:06:33 "GET/plaintext?inputtext-shit HTTP/1.1" 200
```

Sample Log Entry of server application

```
shit
127.0.0.1 -- [12/DEC/2019 13:06:33 "GET/plaintext?inputtext-shit HTTP/1.1" 200
```

Sample Log Entry of Shell Script

```
{
  "message": "***"
}
```

Front GUI:-

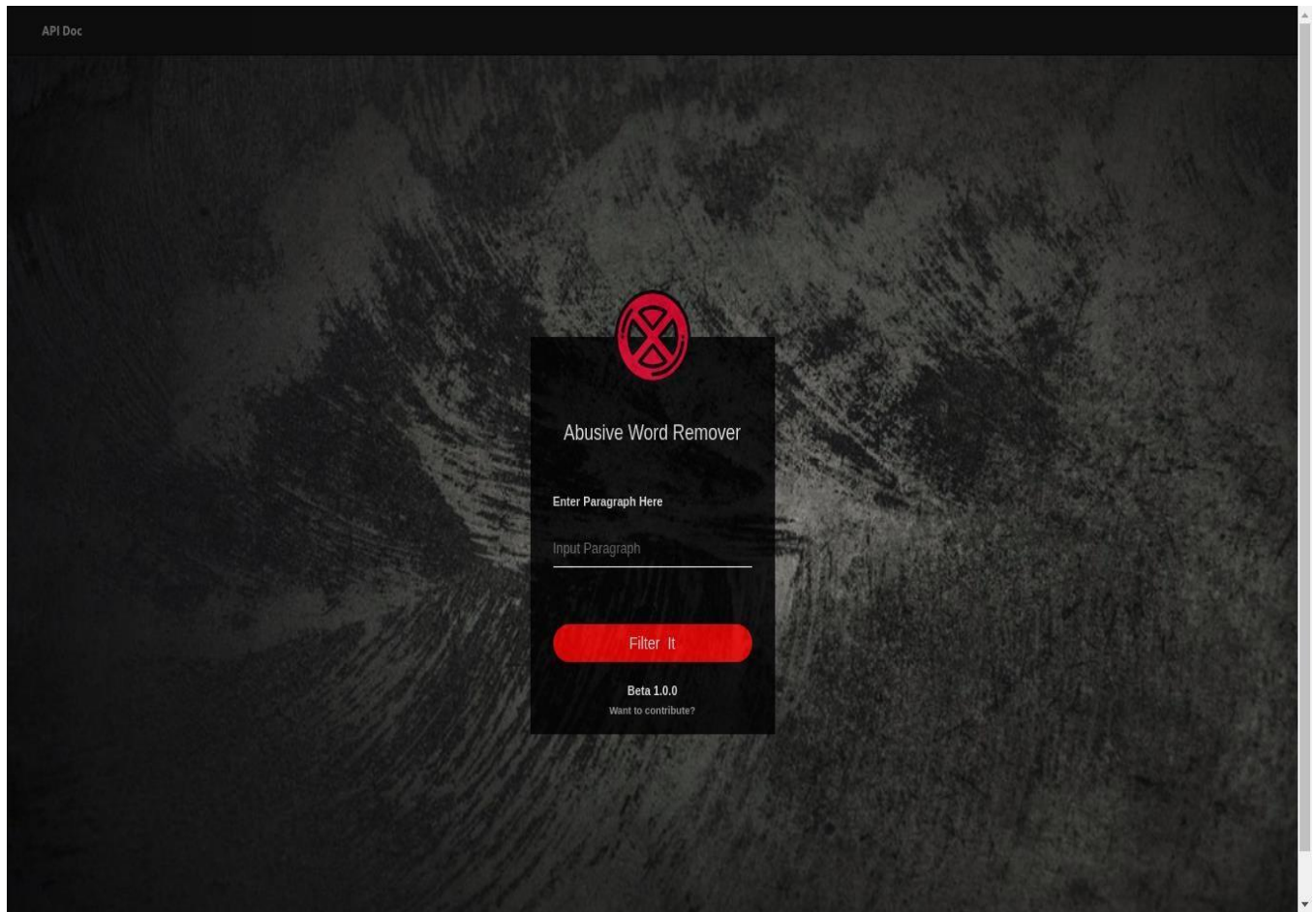


Fig 4.1: Front GUI

Log Files:-

	Date_Stamp	Time_Stamp	Server_Alpha_Status	Server_Numerical_Status
1	2019-12-13	1:10:14	server up	1
2	2019-12-13	1:10:14	server up	1
3	2019-12-13	1:10:15	server up	1
4	2019-12-13	1:10:15	server up	1
5	2019-12-13	1:10:16	server up	1
6	2019-12-13	1:10:16	server up	1
7	2019-12-13	1:10:17	server up	1
8	2019-12-13	1:10:18	server up	1
9	2019-12-13	1:10:18	server up	1
10	2019-12-13	1:10:18	server up	1
11	2019-12-13	1:10:19	server up	1
12	2019-12-13	1:10:20	server up	1
13	2019-12-13	1:10:20	server up	1
14	2019-12-13	1:10:21	server up	1
15	2019-12-13	1:10:21	server up	1
16	2019-12-13	1:10:22	server up	1
17	2019-12-13	1:10:22	server up	1
18	2019-12-13	1:10:23	server up	1
19	2019-12-13	1:10:23	server up	1
20	2019-12-13	1:10:23	server down	0
21	2019-12-13	1:10:23	server down	0
22	2019-12-13	1:10:23	server down	0
23	2019-12-13	1:10:24	server down	0
24	2019-12-13	1:10:24	server down	0
25	2019-12-13	1:10:24	server down	0

Fig 4.2: Log Files

Data Analysis Code (1):-

```
In [9]: import pandas as pd
df = pd.read_csv('/content/cloud_server.csv')
print(df)
```

	Date_Stamp	Time_Stamp	Server_Alpha_Status	Server_Numerical_Status
0	12/13/2019	1:10:14	server up	1.0
1	12/13/2019	1:10:14	server up	1.0
2	12/13/2019	1:10:15	server up	1.0
3	12/13/2019	1:10:15	server up	1.0
4	12/13/2019	1:10:16	server up	1.0
...
1048570	NaN	NaN	NaN	NaN
1048571	NaN	NaN	NaN	NaN
1048572	NaN	NaN	NaN	NaN
1048573	NaN	NaN	NaN	NaN
1048574	NaN	NaN	NaN	1.0

```
[1048575 rows x 4 columns]
/usr/local/lib/python3.6/dist-packages/IPython/core/interactiveshell.py:2718: DtypeWarning: Columns (0,1,2)
have mixed types.Specify dtype option on import or set low_memory=False.
interactivity=interactivity, compiler=compiler, result=result)
```

```
In [7]: import matplotlib.pyplot as plt
server_status = df.groupby('Server_Alpha_Status')['Server_Numerical_Status'].count()
print(server_status)
```

```
Server_Alpha_Status
server down      207
server up        526
Name: Server_Numerical_Status, dtype: int64
```

```
In [3]: server_status_unique = list(df.Server_Alpha_Status.unique())
server_status = server_status.reindex(server_status_unique,axis=0)
print(server_status)
```

```
Server_Alpha_Status
server up      526.0
server down   207.0
NaN           NaN
Name: Server_Numerical_Status, dtype: float64
```

```
In [10]: server_status = server_status.to_frame()
print(server_status.head(12))
server_status.reset_index(level=0,inplace=True)
```

	Server_Numerical_Status
Server_Alpha_Status	
server down	207
server up	526

Data Analysis Code (2):-

```
In [11]: plt.figure(figsize=(13,10))
plt.bar(server_status['Server_Alpha_Status'],server_status['Server_Numerical_Status'],color=(0.5,0.1,0.5,
0.6))
plt.suptitle('Cloud Up Time / Down Time Analysis',fontsize=10)
plt.title('Using Data from 13 DEC 2019 to 14 DEC 2019 (734 Rows | 4 Cols)',fontsize=20)
plt.xlabel('Server Status',fontsize=20)
plt.ylabel('Server Status Time In Seconds',fontsize=20)

for i,num in enumerate(server_status['Server_Numerical_Status']):
    plt.text(i,num,num,ha='center',fontsize=12)
```

Cloud Up Time / Down Time Analysis

Fig 4.4: Data Analysis Code (2)

Data Analysis Graph: - Following Data Analysis Graph is depicting the server availability and server unavailability for a given period of time (1 Day i.e. 13 Dec 2019 - 14 Dec 2019) using Pandas Library of Python and on that basis violation, determination of SLA Violation can be made and Cloud Service Provider will have to pay specified penalties to its customer (end user) as per SLA Terms and Conditions.

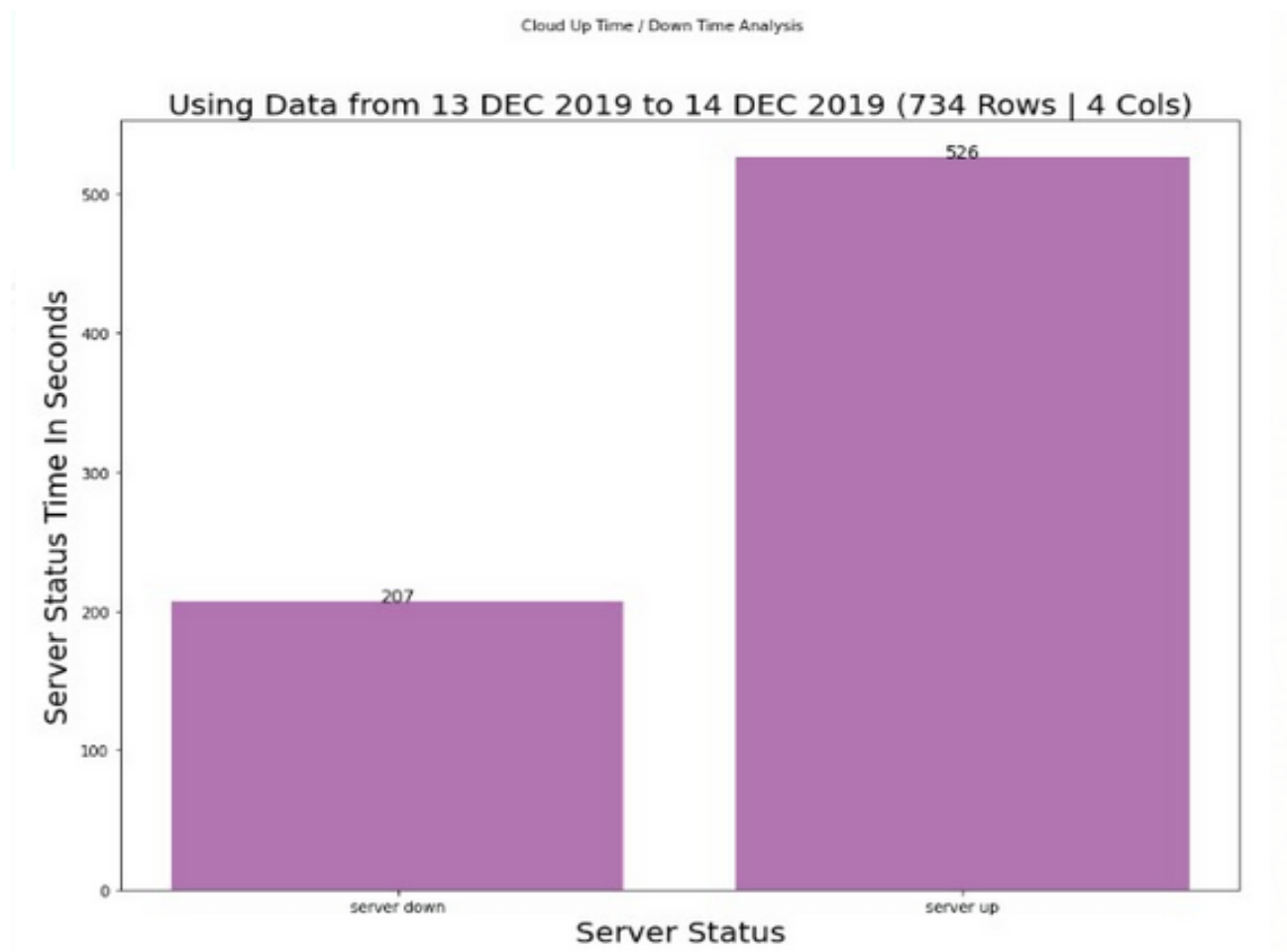


Fig 4.5: Data Analysis Graph

Chapter 5

Conclusion

The project determines the SLA Violations between Cloud Service Provider and Cloud Consumer using the Pandas Library and on that basis we will calculate the penalties that cloud service provider will have to pay due to the violation of SLA terms and conditions. For more precise results time series algorithm of Machine Learning can be used to calculate the penalties that service provider will pay for the violation of SLA (Service Level Agreement) terms and conditions. This project can be widely used in IT industry to ensure service availability to consumer w.r.t. all terms and conditions and it will result in increased productivity of the specific organization.

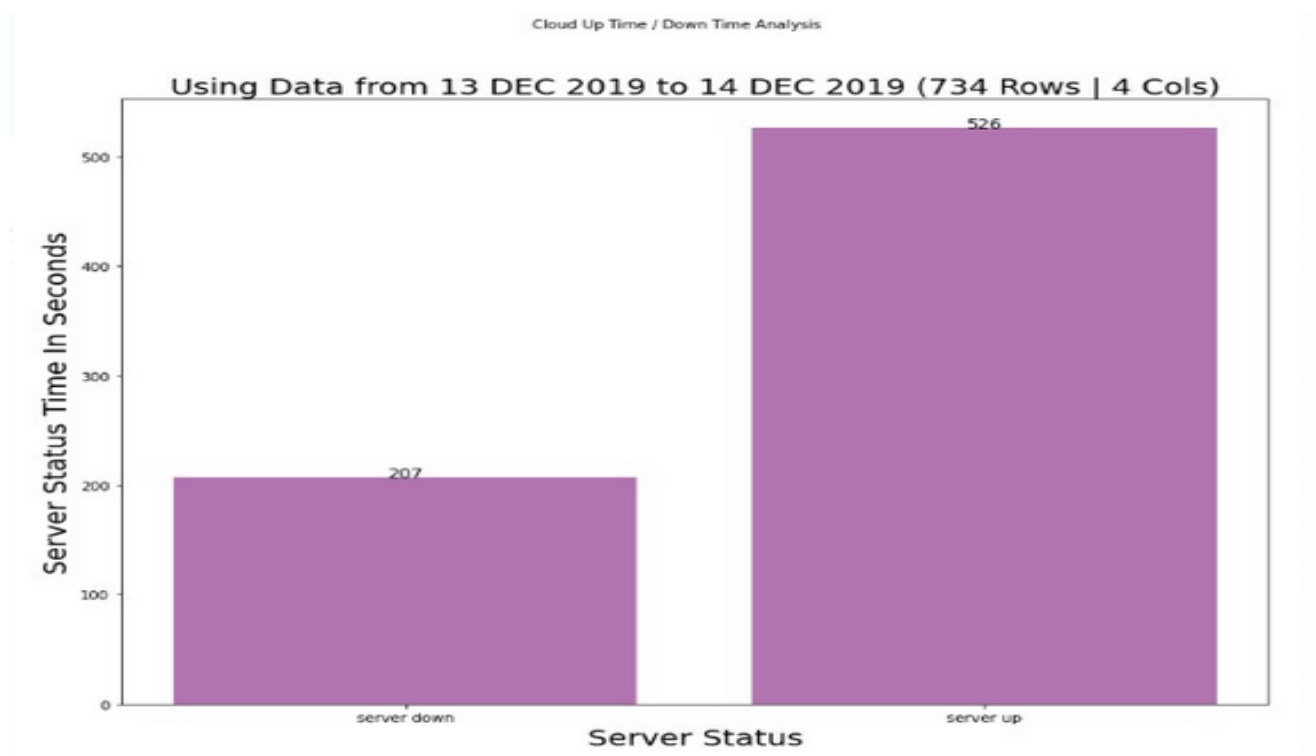


Fig 5.1: Data Analysis Graph

Following Data Analysis Graph is depicting the server availability and server unavailability for a given period of time (1 Day i.e. 13 Dec 2019 - 14 Dec 2019) using Pandas Library of Python and on that basis violation, determination of SLA Violation can be made and Cloud Service Provider will have to pay specified penalties to its customer (end user) as per SLA Terms and Conditions.

Chapter 6

Summary

This prediction based concept is very important and can be utilized and applied for accurate service management of service like in:-

• Cloud of Things (CoT) environment

In the recent past, cloud computing in combination with the Internet of Things has given rise to a new and dynamic area, namely the Cloud of Things (CoT) for service delivery. Despite the various benefits such a paradigm provides, it also brings with it challenges that need to be managed under a dynamic environment to achieve the service aims. This dynamism in QoS is not only observed during the formation of the SLAs but also at run-time. Hence, frequent changes in QoS which are both expected according to a pattern or dependent on other external conditions need to be captured and managed to avoid service violations. To manage the QoS according to a pattern, QoS prediction is one of the critical tasks. Furthermore, in the CoT, as different services from different regions are amalgamated to achieve the required service, the predicted QoS should not only be for individual services but also for the combined ones. But before this can take place, QoS attributes such as response time and service availability need to be predicted over a period both before and after service formation to proactively manage the risk of service violations. To achieve this, service providers need to choose an appropriate prediction approach which, according to the past characteristics of the input's QoS, gives the most accurate future QoS values.

• Proactive healthcare management

With the increase in the population and the strain it places on the health care system, the focus these days is on transforming from a reactive sick care to a proactive health care system. In a proactive model, the objective is to identify various factors such as at-risk individuals based on their current health record data, predict the onset of diseases and predict the risks of individuals being exposed to certain chronic conditions. To achieve these goals, predictive and descriptive types of data analytics have been utilized to predict and categorize patients in these risk profiles. Having such insights is also critical for better government planning and management so that resources can be allocated appropriately. To achieve the same, the recent focus on healthcare has shifted towards predictive analytics. The objective is to use statistical methods to predict outcomes for specific patients in certain conditions. The objective is not to replace the main role of the physician but to provide him with superior tools and methods that will help them to better and more proactively manage a patient's health. To assist these goals, the predicted results need to be accurate hence, using the correct algorithm is key.

• Stock market prediction

Stock markets are volatile and investors need appropriate sophisticated prediction techniques that will pre-determine how the markets will behave. Such techniques are also beneficial to the regulators in helping them to take corrective measures. To achieve this, using prediction methods that can capture the existing patterns and trends in the previous patterns and use these to forecast

future trends is critical. However, a comparison of the methods to determine the accuracy of each is missing. The presented analyses can assist in addressing this gap and thereby helping in the accurate prediction of stock market patterns and trends.

References

- Comparing time series with machine learning-based prediction approaches for violation management in Cloud SLAs.
 - WALAYAT HUSSAIN, School of Systems, Management and Leadership, Faculty of Engineering and Information Technology, Australian University.
 - FAROOKH KHADEER HUSSAIN, Centre for Artificial Intelligence, School of Software, Faculty of Engineering and Information Technology, University of Technology Sydney, NSW, Australia
 - MORTEZA SABERI, School of Business, University of New South Wales, Canberra, Australia.
 - OMAR KHADEER HUSSAIN, School of Business, University of New South Wales, Canberra, Australia.
 - ELIZABETH CHANG, School of Business, University of New South Wales, Canberra, Australia.
- SLA Violation Prediction in Cloud Computing: A Machine Learning perspective
 - REYHANE ASKARI HEMMAT, ABDELHAKIM HAFID.