**Study and Performance Analysis of QoS Parameters in Cloud Networking Platform**

By

**Sagarika Hembram (18/ECE/038) (Leader)**

**Soumik Sil (18/ECE/020)**

**Shaheer Alam (18/ECE/029)**

**Sayan Mandal (18/ECE/031)**

**Neha Rani (18/ECE/060)**

A Project thesis Submitted to Haldia Institute of Technology

In Partial Fulfilment of the Requirements for Degree of

B. Tech in Electronics & Communication Engineering,

Maulana Abul Kalam Azad University of Technology,

Formerly known as West Bengal University of Technology

UNDER THE SUPERVISION OF

Dr. Suman Paul

Assistant Professor

Haldia Institute of Technology

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

HALDIA INSTITUTE OF TECHNOLOGY

2022





**HALDIA INSTITUTE OF TECHNOLOGY**

**(An Institution of ICARE)**

**(Approved by AICTE, New Delhi)**

(Affiliated to Maulana Abul Kalam Azad University of technology, West Bengal)

(Formerly known as West Bengal University of Technology-WBUT)

P.O- HIT, Haldia, Dist.- Purba Medinipur, West Bengal, Pin-721657, West , INDIA

certificate

This is to certify that the thesis entitled **“Study and Performance Analysis of QoS Parameters in Cloud Networking Platform”** submitted by Sagarika Hembram, Soumik Sil, Shaheer Alam, Sayan Mandal and Neha Rani are absolutely based upon their work under the supervision of Dr. Suman Paul, Assistant Professor and that neither this thesis nor any part of it has been submitted for any degree/diploma or any other academic award anywhere before.

----------------------------------- ---------------------------------

Dr. Chanchal Kumar De Dr. Suman Paul

HOD of Electronics & Communication Engineering, Assistant Professor

Haldia Institute of Technology Haldia Institute of Technology

**Ph.: (03224) 252850/253062, EPABX : 252900/253061/255615/255616/255617, Tele Fax: (03224) 252800/253062**

**E-mail : hithaldia@yahoo.com, admin@hithaldia.in Website: www.hithaldia.in**

**CERTIFICATE OF APPROVAL**

The foregoing thesis entitled **“Study and Performance Analysis of QoS Parameters in Cloud Networking Platform”** is hereby approved as a creditable study of an engineering subject carried out and presented in a manner satisfactory to warrants its acceptance as a pre-requisite for the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it is submitted**.**

Committee for evaluation of the project

--------------------------- ------------------------- ---------------------

ACKNOWLEDGMENT

Firstly, we would like to thank Dr. Suman Paul for his constant guidance and support in the project. Without his help the completion of this project would have been impossible. We would also like to share our sincere gratitude to Haldia Institute of Technology for providing the opportunity to work on this project. It helped us understand how to apply technical knowledge in real life situations, which would prove to be of immense importance in the future.

We extend our sincere thanks to one and all of HIT family and faculty members for the completion of this document on the project report format guidelines.

Sagarika Hembram (18/ECE/038)

Soumik Sil (18/ECE/020)

Shaheer Alam (18/ECE/029)

Sayan Mandal (18/ECE/031)

Neha Rani (18/ECE/060)

**i**

ABSTRACT

KEYWORDS: **Voice over IP, Quality of Service (QoS), Machine Learning**

Voice over IP (VoIP) is a real time application that allows transmitting voice through the Internet network. Recently, there has been amazing progress in this field due to the advances in hardware and breakthrough algorithms. As a result, the quality of VoIP calls has improved considerably. However, the quality of VoIP calls under extreme conditions of packet loss still remains a major problem that needs to be addressed. This project report concentrates in making an analysis of the effects that network QoS impairments such as delay, jitter, and packet loss have in the quality of VoIP calls. Additionally, we will use Machine Learning approach to solve this problem by developing a Machine Learning model which optimizes these QoS parameters to enhance the quality of VoIP calls.

**ii**

TABLE OF CONTENTS

|  |  |
| --- | --- |
| **TITLE** | **PAGE** |
| ACKNOWLEDGEMENT | i |
| ABSTRACT | ii |
| LIST OF TABLES | iii |
| LIST OF FIGURES | iv |
| ABBREVIATIONS | vi |
| 1. CHAPTER 1 : INTRODUCTION |  |
| 1.1 Background of Project | 1 |
| 1.2 Organization of Chapters | 2 |
| 2. CHAPTER 2 : CLOUD COMPUTING |  |
| 2.1 What is Cloud Computing | 3 |
| 2.2 Cloud Computing Services | 3 |
| 2.2.1 IaaS (Infrastructure-as-a-Service) | 3 |
| 2.2.2. PaaS (Platform as a Service) | 4 |
| 2.2.3. SaaS (Software as a Service) | 4 |
| 2.3. Types of Cloud | 5 |
| 2.3.1 Public Cloud | 5 |
| 2.3.2 Private Cloud | 6 |
| 2.3.3 Hybrid Cloud | 6 |
| 2.4 Benefits of Cloud | 6 |
| 3. CHAPTER 3 : QUALITY OF SERVICE |  |
| 3.1. What is Quality of Service (QoS) | 8 |
| 3.2. Traffic Characteristics | 9 |
| 3.2.1 Voice | 9 |
| 3.2.2. Video | 9 |
| 3.2.3. Data | 10 |
| 3.3. QoS Models | 10 |
| 3.3.1 Best Effort | 10 |
| 3.3.2. Integrated Services | 11 |
| 3.3.3. Differentiated Services | 12 |
| 3.3. QoS Tools | 13 |
| 4. CHAPTER 4 : VOICE OVER IP(VOIP) |  |
| 4.1. What is VoIP | 15 |
| 4.2. How VoIP works | 15 |
| 4.3. VoIP in Unified Communication | 16 |
| 4.4. VoIP telephone equipment | 16 |
| 4.5. Advantages and disadvantages of VoIP | 17 |
| 5. CHAPTER 5 : LITERATURE REVIEW |  |
| 5.1. Simple Regression | 18 |
| 5.2. Multi-parameter Regression | 18 |
| 5.3. QoE/QoS correlation with supervised ML | 19 |
| 6. CHAPTER 6 : APPROACH |  |
| 6.1. Overview | 20 |
| 6.2. Data Collection | 20 |
| 6.3. Data Preprocessing | 23 |
| 6.4. Data Modelling | 27 |
| 6.4.1. Linear Regression | 27 |
| 6.4.2. K-Nearest Neighbours | 28 |
| 6.4.3. Decision Tree | 29 |
| 6.4.4. Gradient Descent | 31 |
| 6.4.5. Voting Ensemble | 32 |
| 7. CHAPTER 7 : RESULT |  |
| 7.1. Comparative Analysis | 34 |
| 7.2. Discussion | 40 |
|  |  |
| CONCLUSION | 41 |
| APPENDIX A | 42 |
| APPENDIX B | 44 |
| APPENDIX C | 47 |
| APPENDIX D | 48 |
| APPENDIX E | 50 |
| APPENDIX F | 52 |
| APPENDIX G | 54 |
| REFERENCES | 55 |

LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **TABLE** | **TITLE** | **PAGE** |
| 3.1 | Voice Traffic Characteristics | 9 |
| 3.2 | Video Traffic Characteristics | 9 |
| 3.3 | Data Traffic Characteristics | 10 |
| 6.1 | QoS features acceptable range | 21 |
| 6.2 | Description of MOS values | 22 |
| 6.3 | Values of intercept and coefficients in LR | 27 |
| 6.4 | Values of intercept and coefficients in GD Regressor | 32 |
| 6.5 | R2 scores of base models | 33 |
| 7.1 | Comparison of models based on MAE | 34 |
| 7.2 | Comparison of models based on MSE | 36 |
| 7.3 | Comparison of models based on R2 score | 37 |
| 7.4 | Comparison of models based on Adjusted R2 score | 39 |

**iii**

LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **FIGURE** | **TITLE** | **PAGE** |
| 2.1 | Features of Cloud Computing | 3 |
| 2.2 | Cloud Computing Services | 5 |
| 3.1 | Bandwidth difference when QoS is applied and when not applied | 8 |
| 3.2 | IntServ mechanism | 11 |
| 3.3 | DiffServ mechanism | 12 |
| 3.4 | Sequence of QoS tools applied to packet flows | 14 |
| 4.1 | Working of VoIP | 16 |
| 6.1 | Block diagram of overall modelling process | 20 |
| 6.2 | First 5 rows of QoS dataset | 21 |
| 6.3 | Input Feature | 23 |
| 6.4 | Output Feature (MOS) | 23 |
| 6.5 | 3D scatter plot of input features and output feature | 24 |
| 6.6 | 2D scatter plot of packet loss rate and MOS | 24 |
| 6.7 | 2D scatter plot of jitter and MOS | 25 |
| 6.8 | 2D scatter plot of delay and MOS | 25 |
| 6.9 | Spearman’s correlation among the features | 26 |
| 6.10 | Plot for finding value of k | 29 |
| 6.11 | Structure of decision tree | 30 |
| 6.12 | Structure of decision tree with max\_depth = 2 | 30 |
| 6.13 | Feature Importance | 31 |

**iv**

|  |  |  |
| --- | --- | --- |
| 7.1 | Comparison of models based on MAE | 35 |
| 7.2 | Comparison of models based on MSE | 36 |
| 7.3 | Comparison of models based on R2 score | 38 |
| 7.4 | Comparison of models based on Adjusted R2 score | 39 |

**V**

ABBREVIATIONS

|  |  |
| --- | --- |
| **IaaS** | **Infrastructure as a Service** |
| **PaaS** | **Platform as a Service** |
| **SaaS** | **Software as a Service** |
| **QoS** | Quality of Service |
| **POTS** | Plain Old Telephone Service |
| **CSP** | Cloud Service Provider |
| **AWS** | Amazon Web Services |
| **GCP** | Google Cloud Platform |
| **VM** | Virtual Machine |
| **RSTP** | Real-Time Streaming Protocol |
| **UDP** | User Datagram Protocol |
| **VoIP** | Voice – over IP |
| **TCP** | Transmission Control Protocol |
| **BE** | Best Effort |
| **IntServ** | Integrated Services |
| **DiffServ** | Differentiated Services |
| **RSVP** | Resource Reservation Protocol |
| **PHB** | Per-Hop-Behaviour |
| **DSCP** | Differentiated Services Code Point |
| **ACL** | Access Control List |
| **NBAR** | Network Based Application Recognition |
| **WRED** | Weighted random early detection |
| **ML** | Machine Learning |
| **QoE** | Quality of Experience |
| **MOS** | Mean Opinion Score |
|  |  |
| **vi** | |
| **NB** | Naïve Bayes |
| **SVM** | Support Vector Machine |
| **KNN** | K-Nearest Neighbours |
| **LR** | Linear Regression |
| **DT** | Decision Tree |
| **RF** | Random Forest |
| **NN** | Neural Network |
| **MAE** | Mean Absolute Error |
| **MSE** | Mean Square Error |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**vii**

# : INTRODUCTION

## 1.1 Background of Project

Voice over IP (VoIP), the transmission of voice over the Internet, particularly the IP protocol, has gained popularity over the recent years. Businesses and institutions are transforming their current phone infrastructure from Plain Old Telephone Service (POTS) to VoIP. There is a notorious advantage of transmitting voice using the IP protocol and that is IP networks are low in cost.

VoIP also allows the integration of voice and data over the same channel. Voice mail can now easily be integrated into email, virtual conference rooms are being placed around the world, and services such as caller ID and call forwarding can be easily implemented in a packet switched network instead of the traditional circuit switched network. In the near future, we will see an overhaul of new services, such as wireless VoIP. In fact, VoIP is shaping the future of communications. [1]

Over the last few years there has been remarkable progress in the field of VoIP. The Telecom industry has concentrated on developing new voice codecs that perform better under conditions of packet loss and consequently increase the quality of VoIP calls to a certain extent. VoIP has also benefited from improvements in digital signal processing. However, the quality of VoIP calls under impairment conditions such as delay, jitter, and packet loss still remains a major problem that needs to be addressed for next generation of VoIP services.

Through this report, we present an analysis of the effects that these network QoS impairments have in the quality of VoIP calls and try to solve this problem using machine learning approach. We used Mean Opinion Score (MOS) to assess the quality of VoIP calls and developed a machine learning model which optimizes the QoS parameters, hence improving VoIP quality.

## 1.2 Organization of Chapters

This project report consists of the following chapters:

* Chapter 1 (Introduction) - discusses the background and motivation of the report
* Chapter 2 (Cloud Computing) – gives an overview of cloud computing
* Chapter 3 (Quality of Service) – gives a detailed description of QoS
* Chapter 4 (Voice over IP) – gives an overview of VoIP
* Chapter 5 (Literature Review) – provides a brief overview of current state-of-the-art ML techniques for improving QoS
* Chapter 6 (Approach) – discusses about the experimental setup of this report
* Chapter 7 (Results) – gives the results of the experiment

**2**

# : CLOUD COMPUTING

## 2.1 What is Cloud Computing

Cloud computing is on-demand access, via the internet, to computing resources—applications, servers (physical servers and virtual servers), data storage, development tools, networking capabilities, and more—hosted at a remote data center managed by a cloud services provider (or CSP). The CSP makes these resources available for a monthly subscription fee or bills them according to usage. The three largest public CSPs that have established themselves as dominant fixtures in the industry are AWS, GCP and Microsoft Azure. Other major CSPs include Apple, Citrix, IBM, Salesforce, Oracle etc.

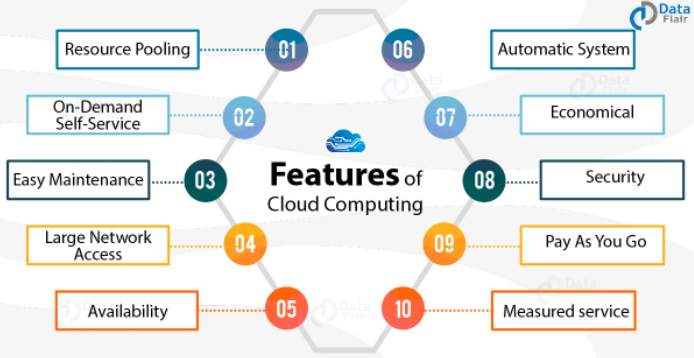


Fig. 2.1 Features of Cloud Computing

## 2.2 Cloud Computing Services

### 2.2.1 IaaS (Infrastructure-as-a-Service)

IaaS provides on-demand access to fundamental computing resources–physical and virtual servers, networking, and storage—over the internet on a pay-as-you-go basis. IaaS enables end users to scale and shrink resources on an as-needed basis, reducing the need for high, up-front capital expenditures or unnecessary on-premises or ‘owned’ infrastructure and for overbuying resources to accommodate periodic spikes in usage.

### 2.2.2 PaaS (Platform as a Service)

PaaS provides software developers with on-demand platform—hardware, complete software stack, infrastructure, and even development tools—for running, developing, and managing applications without the cost, complexity, and inflexibility of maintaining that platform on-premises.

With PaaS, the cloud provider hosts everything—servers, networks, storage, operating system software, middleware, databases—at their data center. Developers simply pick from a menu to ‘spin up’ servers and environments they need to run, build, test, deploy, maintain, update, and scale applications.

### 2.2.3 SaaS (Software as a Service)

SaaS—also known as cloud-based software or cloud applications—is application software that’s hosted in the cloud and that you access and use via a web browser, a dedicated desktop client, or an API that integrates with your desktop or mobile operating system. In most cases, SaaS users pay a monthly or annual subscription fee; some may offer ‘pay-as-you-go’ pricing based on your actual usage.

**4**

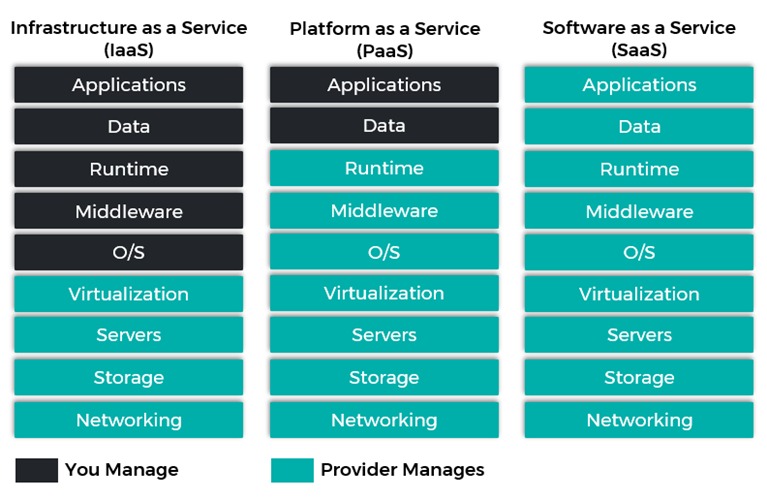


Fig. 2.2. Cloud Computing Services

## 2.3 Types of Cloud

### 2.3.1 Public Cloud

Public cloud is a type of cloud computing in which a cloud service provider makes computing resources available to users over the internet. These resources might be accessible for free, or access might be sold according to subscription-based or pay-per-usage pricing models. The public cloud provider owns, manages, and assumes all responsibility for the data centers, hardware, and infrastructure. It typically provides high-bandwidth network connectivity to ensure high performance and rapid access to applications and data.

**5**

### 2.3.2 Private Cloud

Private cloud is a cloud environment in which all cloud infrastructure and computing resources are dedicated to and accessible by one customer only. It is typically hosted on-premises in the customer's data center. Many companies choose private cloud over public cloud because private cloud is an easier way to meet their regulatory compliance requirements. Others choose private cloud because their workloads deal with confidential documents, medical records, financial data, or other sensitive data.

### 2.3.3 Hybrid Cloud

Hybrid cloud is a combination of public and private cloud environments. It connects an organization's private cloud services and public clouds into a single, flexible infrastructure. The goal of hybrid cloud is to establish a mix of public and private cloud resources that gives an organization the flexibility to choose the optimal cloud for each workload and to move workloads freely between the two clouds as circumstances change.

## 2.4 Benefits of Cloud

• Cost - Cloud computing eliminates the capital expense of buying hardware and software and setting up and running on-site data centers.

• Speed - Most cloud computing services are provided self service and on demand, so even vast amounts of computing resources can be provisioned in minutes.

• Global scale - It means delivering the right amount of IT resources right when it is needed and from the right geographic location.

• Productivity - On-site data centers typically require a lot of hardware setup, software patching, and other time-consuming IT management chores. Cloud computing removes the need for many of these tasks, so IT teams can spend time on achieving more important business goals.

**6**

• Reliability - Cloud computing makes data backup, disaster recovery and business continuity easier and less expensive because data can be mirrored at multiple redundant sites on the cloud provider’s network.

• Security - Many cloud providers offer a broad set of policies, technologies and controls that strengthen your security posture overall, helping protect your data, apps and infrastructure from potential threats.

**7**

# : QUALITY OF SERVICE

## 3.1 What is Quality of Service (QoS)

A stream of packets from a source to destination is called a flow. Quality of Service is defined as something a flow seeks to attain.  In connection oriented network, all the packets belonging to a flow follow the same order. In a connectionless network, all the packets may follow different routes.

The needs of each flow can be characterized by four primary parameters:

• **Reliability** - Lack of reliability means losing a packet or acknowledgement which entertains retransmission.

**• Delay** - Increase in delay means destination will find the packet later than expected.

• **Jitter** - Jitter is the variation in delay for packets belonging to the same flow.

• **Bandwidth** - Increase in bandwidth means increase in the amount of data which can be transferred in given amount of time.

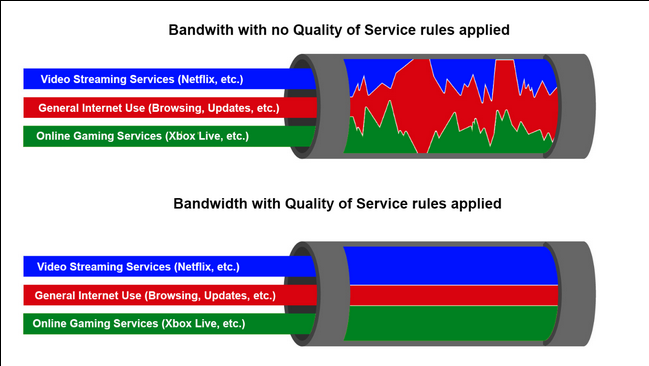


Fig. 3.1. Bandwidth difference when QoS is applied and when not applied

## 3.2 Traffic Characteristics

### 3.2.1 Voice

Voice traffic is predictable and smooth and very sensitive to delays and dropped packets.

* Voice packets must receive a higher priority than other types of traffic.
* Cisco products use the RTP port range 16384 to 32767 to prioritize voice traffic.
* Voice can tolerate a certain amount of latency, jitter, and loss without any noticeable effects. [2]

Table 3.1. Voice Traffic Characteristics [2]

|  |  |
| --- | --- |
| **VOICE TRAFFIC CHARACTERISTICS** | **ONE-WAY REQUIREMENTS** |
| * Smooth * Benign * Drop sensitive * Delay sensitive * UPD priority | * Latency <= 150 ms * Jitter <= 30 ms * Loss <= 1% Bandwidth (30 - 128 Kbps) |

### 3.2.2 Video

Video traffic tends to be unpredictable, inconsistent, and bursty. Compared to voice, video is less resilient to loss and has a higher volume of data per packet.

* The number and size of video packets varies every 33 ms based on the content of the video.
* UDP ports such as 554, are used for the Real-Time Streaming Protocol (RSTP) and should be given priority over other, less delay-sensitive, network traffic. [2]

Table 3.2. Video Traffic Characteristics [2]

|  |  |
| --- | --- |
| **VIDEO TRAFFIC CHARACTERISTICS** | **ONE-WAY REQUIREMENTS** |
| * Bursty * Greedy * Drop sensitive * Delay sensitive * UPD priority | * Latency <= 200 - 400 ms * Jitter <= 30 - 50 ms * Loss <= 0.1 – 1% * Bandwidth (384 Kbps - 20 Mbps) |

**9**

### 3.2.3 Data

Data applications that have no tolerance for data loss, such as email and web pages, use TCP to ensure that if packets are lost in transit, they will be resent.

* Data traffic can be smooth or bursty.
* Network control traffic is usually smooth and predictable. [2]

Table 3.3. Data Traffic Characteristics [2]

|  |
| --- |
| **DATA TRAFFIC CHARACTERISTICS** |
| * Smooth/Bursty * Benign/Greedy * Drop Insensitive * Delay Insensitive * TCP Retransmits |

## 3.3 QoS Models

There are three main models for providing QoS services in a network:

* Best Effort.
* Integrated Services (IntServ).
* Differentiated Services (DiffServ)

### 3.3.1 Best Effort

The Best Effort (BE) QoS model is the simplest of the three. It is the default QoS model used for Internet and it doesn’t implement any QoS mechanism at all, that is the reason why there isn’t any complexity associated to this QoS model.

BE does not allow for resource reservation or any other mechanism related to asking for some kind of special treatment to the network. For this reason, BE model does not work very well will any emerging application with real-time (RT) traffic demands.

**10**

### 3.3.2 Integrated Services

The Integrated Services (IntServ) model is also known as hard QoS model. It is a model based on flow, i.e., source and destination IP addresses and ports. IntServ model performs deterministic Admission Control (AC) based on resources requests vs. available resources.

With the IntServ model, applications ask to the network for an explicit resource reservation per flow. Network devices keep track of all the flows traversing the nodes checking if new packets belong to an existing flow and if there are enough network resources available to accept the packet. By reserving resources on the network for each flow, applications obtain resources guarantees and a predictable behaviour of the network.

The implementation of this model requires the presence of IntServ capable routers in the network and uses RSVP for end-to-end resource reservation. RSVP enables a host to establish a connection over connectionless IP Internet:

1. Applications request some level of service to the network before sending data.
2. The network admits or rejects the reservation (per flow) based on available resources.
3. Once cleared, the network expects the application to remain within the requested traffic profile.

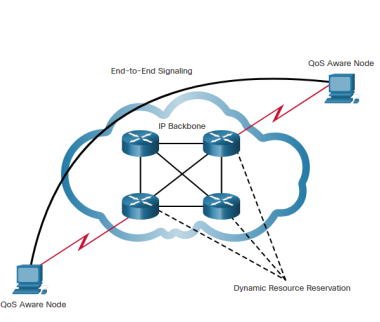


Fig. 3.2. IntServ mechanism [2]

**11**

### 3.3.3 Differentiated Services

Differentiated Services (Diffserv) model is also known as a soft QoS model. It’s a model based in service classes and per hop behaviours associated to each class. In this case, there is no need for an explicit request for resource reservation by applications to the network. Differentiated Services is based on statistical preferences per traffic class.

DiffServ allows end devices or hosts to classify packets into different treatment categories or Traffic Classes (TC), each of which will receive a different Per-Hop-Behaviour (PHB) at each hop from the source to the destination. Each network device on the path treats packets according to the locally defined PHB. PHB defines how a node deals with a TC. Network service policies can be specific to an entire QoS domain, some part of a network or even a single node.

 Priorities are marked in each packet using DSCP for traffic classification. This marking is performed per packet usually at the QoS domain boundary. The marking can be done at several levels of the networking layers (MPLS EXP, CoS).

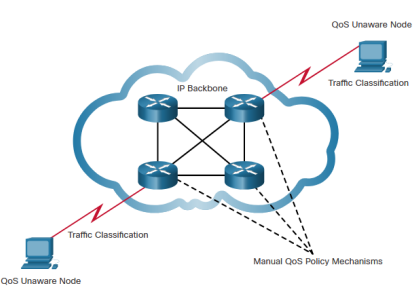


Fig. 3.3. DiffServ mechanism [2]

**12**

## 3.4 QoS Tools

There are three categories of QoS tools:

**1. Classification and marking tools** -

Before a packet can have a QoS policy applied to it, the packet has to be classified. Classification determines the class of traffic to which packets or frames belong. When the traffic class is determined, the packets are marked. Only after traffic is marked can policies be applied to it. How a packet is classified depends on the QoS implementation.

* Methods of classifying traffic flows at Layer 2 and 3 include using interfaces, ACLs, and class maps.
* Traffic can also be classified at Layers 4 to 7 using Network Based Application Recognition (NBAR). [2]

**2. Congestion avoidance tools** –

Congestion avoidance tools monitor network traffic loads in an effort to anticipate and avoid congestion at common network and internetwork bottlenecks before congestion becomes a problem. They monitor the average depth of the queue. When the queue is below the minimum threshold, there are no drops. As the queue fills up to the maximum threshold, a small percentage of packets are dropped. When the maximum threshold is passed, all packets are dropped. Some congestion avoidance techniques provide preferential treatment for which packets get dropped.

Weighted random early detection (WRED) allows for congestion avoidance on network interfaces by providing buffer management and allowing TCP traffic to decrease, or throttle back, before buffers are exhausted. WRED helps avoid tail drops and maximizes network use and TCP-based application performance. [2]

**13**

**3. Congestion management tools** -

Traffic shaping and traffic policing are two mechanisms provided by Cisco IOS QoS software to manage congestion.

* Traffic shaping retains excess packets in a queue and then schedules the excess for later transmission over increments of time. Traffic shaping results in a smoothed packet output rate. Shaping is an outbound concept; packets going out an interface get queued and can be shaped.
* Policing is applied to inbound traffic on an interface. Policing is commonly implemented by service providers to enforce a contracted customer information rate (CIR). However, the service provider may also allow bursting over the CIR if the service provider’s network is not currently experiencing congestion. [2]

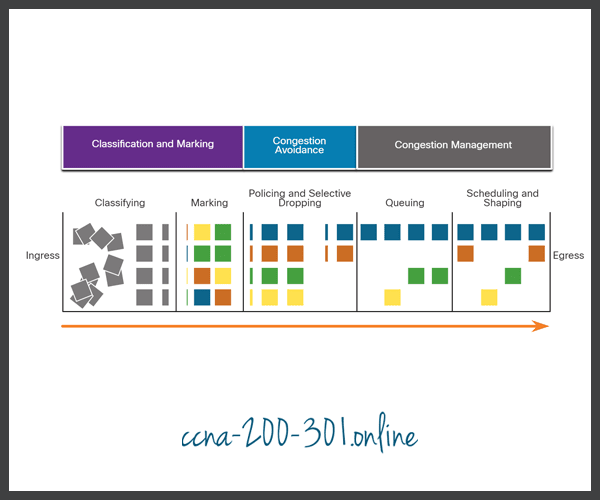
****

Fig. 3.4. Sequence of QoS tools applied to packet flows [2]

**14**

# : VOICE OVER IP (VOIP)

## 4.1 What is VoIP

VoIP (voice over Internet Protocol) is the transmission of voice and multimedia content over an internet connection. VoIP allows users to make voice calls from a computer, smartphone, other mobile devices, special VoIP phones and WebRTC-enabled browsers. VoIP is a technology useful for both consumers and businesses, as it typically includes other features that can't be found on common phone services. It is also helpful to organizations as a way to unify communications. The process works similarly to a regular phone, but VoIP uses an internet connection instead of a telephone company's wiring.

A VoIP service will convert a user's voice from audio signals to digital data, then send that data through the internet. If another user is calling from a regular phone number, the signal is converted back to a telephone signal before it reaches that user.

## 4.2 How VoIP works

VoIP services convert a user's voice from audio signals to digital data, in which that data is then sent to another user -- or group of users -- over Ethernet or Wi-Fi. To accomplish this, VoIP will use codecs. Codecs are either a hardware- or software-based process that compresses and decompresses large amounts of VoIP data. Voice quality may suffer when compression is used, but compression reduces bandwidth requirements.

The process of sending data to other users includes encapsulating audio into data packets, transmitting the packets across an IP network and un-encapsulating the packets back into audio at the other end of the connection.

Additional components of a typical VoIP system include the following: an IP PBX to manage user telephone numbers, devices, features and clients; gateways to connect networks and provide failover or local survivability in the event of a network outage; and session border controllers to provide security, call policy management and network connections.

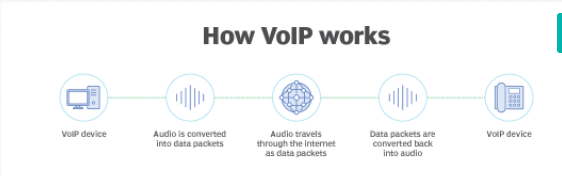


Fig 4.1: Working of VoIP

## 4.3 VoIP in Unified Communication

VoIP consolidates communication technologies into one unified system -- meaning that VoIP can allow for a number of audio, video or text-based communication methods. This can be particularly useful for businesses, so teams don't have to work with multiple different applications to communicate with one another effectively. VoIP creates this network by allowing users to make calls and hold web conferences using devices like computers, smartphones or other mobile devices.

Some common features might include:

* audio calls;
* video calls;
* voicemail;
* team chats;
* email;
* SMS texts;
* mobile and desktop apps
* mobile and local number portability

## 4.4 VoIP telephone equipment

The two main types of VoIP telephones are hardware-based and software-based.

A hardware-based VoIP phone looks like a traditional hard-wired or cordless telephone and includes similar features, such as a speaker or microphone, a touchpad and a caller ID display. VoIP phones can also provide voicemail, call conferencing and call transfer.

**16**

Software-based IP phones, also known as softphones, are software clients installed on a computer or mobile device. The softphone user interface often looks like a telephone handset with a touchpad and caller ID display.

## 4.5 Advantages and disadvantages of VoIP

Benefits of VoIP include:

* Lower cost - Price is lower than typical phone bills.
* Higher-quality sound - With uncompressed data, audio is less muffled or fuzzy.
* Access for remote workers - Good for employees who work remotely as they have a number of options to call into meetings or communicate to other teammates.
* Added features - These features include call recording, queues, custom caller ID or voicemail to email.
* Low international rates - When a landline makes an international call, it rents the wired circuit for the call to transfer overseas. VoIP doesn't require a wired line and uses the internet to make calls, which means it's treated like normal traffic and is less expensive.

Disadvantages of VoIP:

* Not all these services may connect directly to emergency services.
* VoIP needs a high-speed internet connection.
* Services will not work during power outages.
* There may be a lack of directory assistance depending on the VoIP service.

**17**

# : LITERATURE REVIEW

Statistical and ML techniques have been found useful in linking QoE to network- and application-level QoS, and understanding the impact of the latter on the former. Linear and non-linear regression (e.g. exponential, logarithmic, power regression) was used to quantify the individual and collective impact of network- and application- level QoS parameters (e.g. packet loss ratio, delay, throughput, round-trip time, video bitrate, frame rate, etc.) on the user’s QoE.

## 5.1 Simple Regression

In [3], Shaikh et al. study existing correlation between different network-level QoS parameters and MOS in the context of a web surfing. They show that a correlation does exist and that among 3 forms of regression (linear, exponential, and logarithmic), linear regression renders the best correlation coefficient between QoE and packet loss rate, exponential regression captures the correlation between QoE and file download time with highest accuracy, whereas logarithmic regression is the best fit for linking QoE to throughput.

Reichl et al. [4], in alignment with the Weber-Fechner law from the field of psychophysics, use logarithmic regression to quantify the correlation between available bandwidth and mobile broadband service users’ MOS.

## 5.2 Multi-parameter Regression

In order to grasp the impact of the global network condition on the QoE, Elkotob et al. [5] propose to map MOS to a set of QoS parameters (e.g. packet loss rate, frame rate, bandwidth, round trip time and jitter) as opposed to a single one

More complex regression and classification models based on supervised and unsupervised ML techniques (including deep learning) were also proposed and tested against real-life and trace-driven datasets.

## 5.3 QoE/QoS correlation with supervised ML

In [6], Mushtaq et al. applied six ML classifiers to model QoE/QoS correlation, namely NB, SVM, *k*-NN, DT, RF and NN. A dataset is generated from a controlled network environment where streamed video traffic flows through a network emulator and different delay, jitter, and packet loss ratio are applied. Opinion scores are collected from a panel of viewers and MOS are calculated. ML models are fed with nine features related to the viewers, network condition and the video itself, namely, viewer gender, frequency of viewing, interest, delay, jitter, loss, conditional loss, motion complexity and resolution. A 4-fold cross-validation is performed to estimate the performance of the models. Results show that DT and RF perform slightly better than the other models with a mean absolute error of 0.126 and 0.136 respectively, and a TPR of 74% and 74.8*%* respectively.

Demirbilek et al. [7] developed no-reference models to predict QoE for audio-visual services. These techniques include: decision tree ensemble methods (RF and BG), and deep learning (DNN). All models are trained and validated through 4∼10-fold cross-validation on the INRS dataset [8]. The dataset includes user-generated MOS on audio-visual sequences encoded and transmitted with varying video frame rates, quantization parameters, filters and network packet loss rates. 34 no-reference application- and network-level features are considered. Experiments with different feature sets show that, apart from the DNN model, all models perform better with the complete set of features, and hence do not require feature processing. On the contrary the DNN model performs better when trained only with 5 independent features, namely: video frame rate, quantization, noise reduction, video packet loss rate, and audio packet loss rate. The conducted experiments also show that all models perform quite well and that the RF model with complete set of features performs the best (lowest RMSE 0.340 and highest PCC 0.930). The video packet loss rate seems to be the most influential feature on the RF model.

**19**

# : APPROACH

## 6.1. Overview

This report attempts to optimize the QoS parameters by using supervised machine learning techniques. The block diagram of the overall prediction process is shown in Fig. 6.1. QoS dataset is taken which consists of VoIP traffic parameters which includes packet loss rate, jitter, delay, a, effective delay, R, final R and Mean Opinion Score (MOS). Data pre-processing is performed by removing the redundant features and scaling the features. Then this cleaned data is applied to 5 ML algorithms – Linear Regression, KNN, Decision Tree, Gradient Descent and Voting Ensemble to construct the models. The models were validated and their performances were compared. Finally, we predicted the MOS values based on the optimal machine learning model.

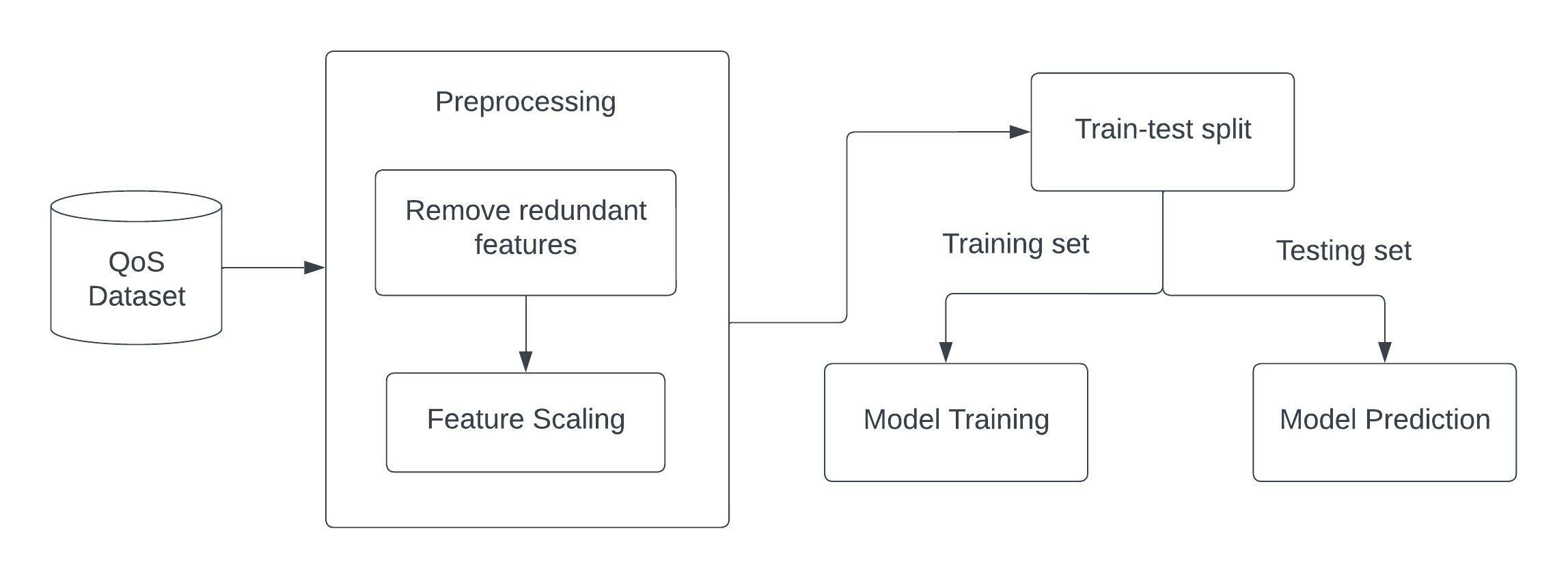


Fig 6.1 : Block diagram of overall modelling process

## 6.2. Data Collection

QoS dataset is generated which focuses on VoIP traffic parameters. It comprises of 204 samples and 8 features - packet loss rate, jitter, delay, a, effective delay, R, final R and Mean Opinion Score (MOS). Fig 6.2 shows the first 5 rows of QoS dataset.

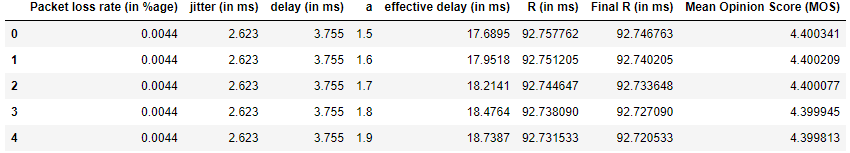


Fig 6.2: First 5 rows of QoS dataset

The samples for packet loss rate, jitter and delay ranges within the acceptable limits for VoIP as given in table 6.1 :

Table 6.1: QoS features acceptable range

|  |  |
| --- | --- |
| feature | range |
| packet loss rate | 0.001% to 0.03% |
| jitter | 0.01 ms to 5 ms |
| delay | 1 ms to 20 ms |

We combine delay and jitter into a metric called effective delay, which is measured in milliseconds. The calculation is as follows:

Effective delay = delay + a\*jitter + 10 ………….. (6.1)

‘a’ is a factor which varies between 1.5 to 2. We multiplied jitter with the factor of ‘a’ because the effect of jitter is high on the voice quality. We add a constant of 10 ms to account for the delay from the codecs.

R is a metric we calculated based on effective delay as follows:

For effective delay < 160 ms:

R = 93.2 - (effective delay)/40 …………… (6.2)

For effective delay >= 160 ms:

**21**

R = 93.2 - (effective delay - 120)/10 …….…… (6.3)

If the effective delay is less than 160 ms, the overall impact to the voice quality is moderate. For larger values, the voice quality drops more significantly, which is why R is penalized more.

Final R is calculated by taking packet loss rate into consideration as follows:

Final R = R - 2.5 \* packet loss rate ………..…… (6.4)

MOS (Mean Opinion Score) is the universal metric to measure and classify the conversation quality that happens over a network. It is based on the opinion of the user and ranges from 1.0 to 5.0 with the following classifications given in table 6.2

Table 6.2: Description of MOS values

|  |  |  |
| --- | --- | --- |
| **MOS** | **QUALITY** | **DESCRIPTION** |
| 5 | Excellent | Imperceptible |
| 4 | Good | Perceptible but not annoying |
| 3 | Fair | Slightly annoying |
| 2 | Poor | Annoying |
| 1 | Bad | Very annoying |

MOS is calculated with the following formula:

For final R < 0:

MOS = 1.0 ………..…… (6.5)

For 0 < final R < 100.0:

MOS = 1 + 0.035\*final R + 0.000007\* final R\*(final R-60)\*(100- final R) ……… (6.6)

For final R >= 100.0:

MOS = 4.5 …………..… (6.7)

**22**

## 6.3 Data Preprocessing

The redundant features (effective delay, R and final R) are removed from the dataset and only the independent features are taken for modelling.

The input features are packet loss rate, jitter, delay and a as shown in fig 6.3. The output feature is MOS and is shown in fig 6.4.

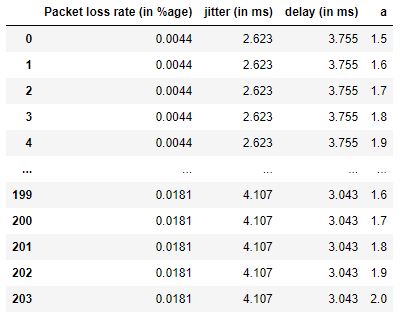


Fig 6.3: input features

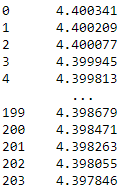


Fig 6.4 : output feature (MOS)

The relation between the input features and output feature is shown in the following fig 6.5 :

**23**

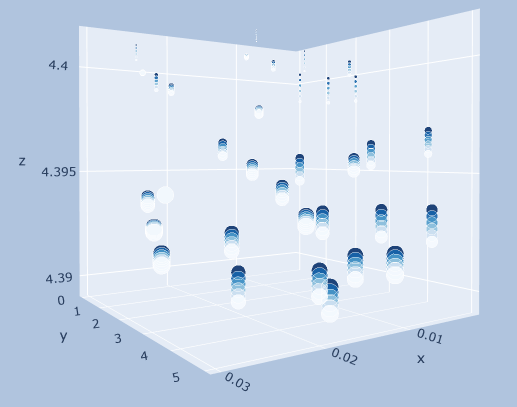


Fig: 6.5: 3D scatter plot of input features and output feature

Here , x-axis represents packet loss rate, y-axis represents jitter, z-axis represents MOS, size of the blob represents delay (smaller size means less delay) and color of the blob represents value of a (lighter shade means less value of a).

From fig 6.5, we can see that:

* MOS is inversely proportional to jitter (as jitter increases MOS decreases)
* MOS is inversely proportional to delay (as delay increases MOS decreases)

The relation between MOS and each of the input features is shown in fig 6.6, fig 6.7 and fig 6.8

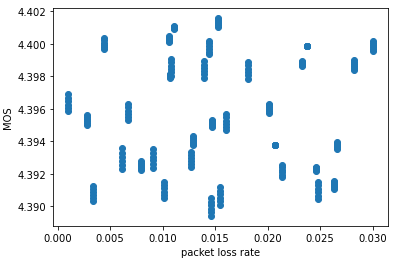


Fig 6.6: 2D scatter plot of packet loss rate and MOS

**24**

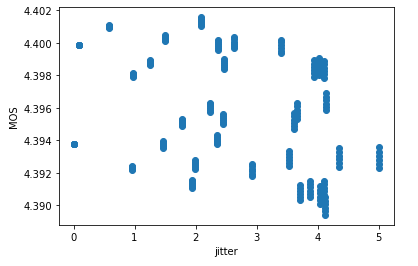


Fig 6.7: 2D scatter plot of jitter and MOS

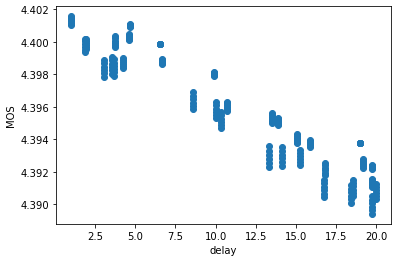


Fig 6.8: 2D scatter plot of delay and MOS

The source code for these plots are given in Appendix B.

Fig 6.9 shows the Spearman’s correlation among the features. It can be seen that there is little to no correlation among the input features, which suggests that the input features are independent of each other. Further, it is evident that delay has the highest value of negative correlation with MOS, indicating that delay is an important feature in predicting MOS.

**25**

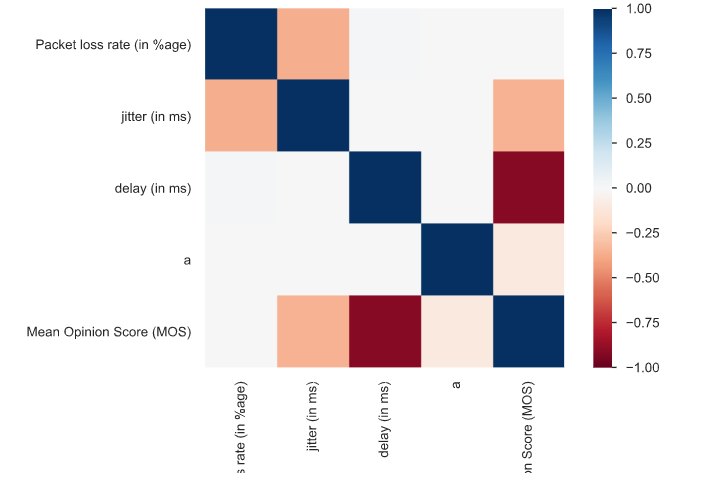


Fig 6.9: Spearman’s correlation among the features

Now the dataset is split into training set and testing set in 75:25 ratio. Therefore 153 rows goes into training set and 52 rows goes into testing set. The training set will be used to build the model and testing set will be used to validate the model.

Then both the training and testing set are standardized using the below formula. Standardizing involves rescaling the distribution of values so that the mean of observed values is 0 and the standard deviation is 1.

z = (x - u) / s ……………… (6.8)

where, u = mean of the training samples

 s = standard deviation of the training samples

The source code is provided in Appendix A.

**26**

## 6.4 Data Modelling

We used 5 different ML algorithms – Linear Regression, KNN, Decision Tree, Gradient Descent and Voting Ensemble to construct the models. Each of the model’s hyper-parameters were optimized to obtain the best result.

### 6.4.1 Linear Regression

Linear regression is a linear model, i.e, it assumes a linear relationship between the input variables (x) and the output variable (y). Mathematically, we can represent a linear regression as:

y = a0 + a1 \* x ……….…… (6.9)

Here,

y = Dependent Variable (Target Variable)

x = Independent Variable (predictor Variable)

a0 = intercept of the line

a1 = Linear regression coefficient (slope / scale factor to each input variable)

After applying LR to construct the model from our data, we obtained the following values of intercept and coefficients shown in table 6.3:

Table 6.3: Values of intercept and coefficients in LR

|  |  |
| --- | --- |
| Intercept | 4.395731072042288 |
| Coefficient of packet loss rate | -0.00039892 |
| Coefficient of jitter | -0.00118707 |
| Coefficient of delay | -0.00320489 |
| Coefficient of a | -0.00022948 |

So our Linear regression model can be represented as follows:

**27**

MOS = 4.3957 – (0.00039892 \* packet loss rate + 0.00118707 \* jitter + 0.00320489 \* delay + 0.00022948 \* a) …………..…… (6.10)

From the table 6.3, we can see that all the input features are negatively correlated with the output feature because their coefficients are negative. Delay is the most significant feature as it has the greatest negative coefficient and packet loss rate is the least significant.

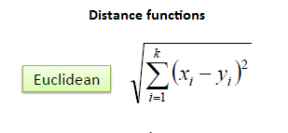
The source code of this model is given in Appendix C.

### 6.4.2 K-Nearest Neighbours

The K-Nearest Neighbours algorithm uses feature similarity to predict the values of any new data points. This means that the new point is assigned a value based on how closely it resembles the points in the training set.

Steps of KNN algorithm:

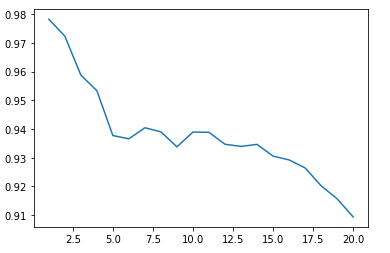
* First, the distance between the new point and each training point is calculated using the following formula:

 …………..…… (6.11)

* The closest k data points are selected (based on the distance).
* The average of these data points is the final prediction for the new point.

To find the optimal value of k, we plotted a graph of k values ranging from 1 to 20 and their corresponding model accuracies as shown in fig 6.10:

**28**



Accuracy

Values of k

Fig 6.10: Plot for finding value of k

From fig 6.10, we can say that k = 2 is the optimal value. So, we use k = 2 to construct our KNN model.

The source code of this model is given in Appendix D.

### 6.4.3 Decision Tree

Decision tree builds models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches, each representing values for the attribute tested. Leaf node represents a decision on the numerical target. The topmost decision node in a tree which corresponds to the best predictor called root node.

**29**



Fig 6.11 : Structure of decision tree

By hyper-parameter tuning, we found max\_depth = 5 gives the optimal result. So, we use max\_depth = 5 to construct the decision tree model. The following fig 6.12 shows the decision tree model with max\_depth = 2.

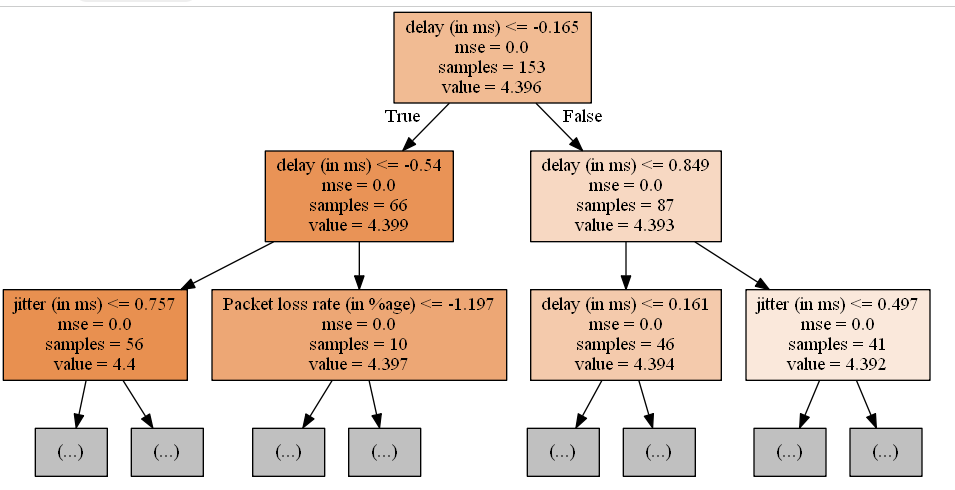
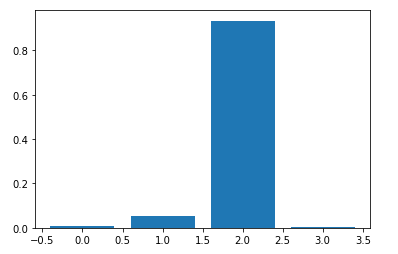


Fig 6.12: Structure of decision tree model with max\_depth = 2

**30**



importance

a

delay

jitter

Packet loss rate

Fig 6.13: Feature Importance

Fig 6.13 shows the importance of each individual feature in making the model. It can be seen that delay has the highest importance, followed by jitter, packet loss rate and a.

The source code of this model is given in Appendix E.

### 6.4.4 Gradient Descent

Gradient Descent is an algorithm that finds the best-fit line for a given training dataset in a smaller number of iterations. For some combination of m and c, we will get the least Error (MSE). That combination of m and c will give us our best fit line.

The algorithm starts with some value of m and c (usually starts with m=0, c=0). We calculate MSE (cost) at point m=0, c=0. Then we reduce the value of m and c by some amount (Learning Step). We will notice a decrease in MSE (cost). We will continue doing the same until our cost function is a very small value or ideally 0 (which means 0 error or 100% accuracy).

**31**

We created a user-defined class named ‘GD Regressor’ which takes epochs (number of iterations) and learning rate as input parameters.

1. It takes the initial values as m=0 and c=0.
2. Then it calculates the partial derivative of the Cost function (MSE) with respect to m and the partial derivative of the Cost function with respect to c.
3. Next it updates the current values of m and c.
4. It repeats step2 – step3 for the given number of iterations.

We have taken epochs = 70 and learning rate = 0.1, which gives the following values for m (coefficient) and c (intercept), as shown in table 6.4:

Table 6.4: Values of intercept and coefficients in GD Regressor

|  |  |
| --- | --- |
| Intercept | 4.395730348732736 |
| Coefficient of packet loss rate | -0.00033312 |
| Coefficient of jitter | -0.00112119 |
| Coefficient of delay | -0.00320633 |
| Coefficient of a | -0.00023265 |

The source code of this model is given in Appendix F.

### 6.4.5 Voting Ensemble

A voting ensemble combines the predictions from multiple other models. In the case of regression, this involves calculating the average of the predictions from the models. It is a technique that may be used to improve model performance, ideally achieving better performance than any single model used in the ensemble.

For our Voting model, we have taken 3 base models – Linear Regression model, Decision tree model and KNN model.

R2 scores for each of the base models using 5-fold cross-validation are given below in table 6.5:

**32**

Table 6.5 : R2 scores of base models

|  |  |
| --- | --- |
| **BASE MODEL NAME** | **R2 SCORE** |
| Linear Regression model | 1.0 |
| Decision tree | 0.82 |
| KNN | 0.91 |

Now, these base models are combined to form the Voting Ensemble model.

The source code of this model is given in Appendix G.

**33**

# : RESULTS

## 7.1. Comparative Analysis

Each model was assessed using the regression metrics - Mean Absolute Error (MAE), Mean Squared Error (MSE), R Squared (R2 score) and Adjusted R-Square.

* Mean Absolute Error (MAE) –

It is the average of the absolute differences between the actual value and the model’s predicted value. The mean absolute error (MAE) has the same unit as the original data. The bigger the MAE, the more critical the error is. It is robust to outliers.

However, it fails to punish the bigger error terms. MAE is not differentiable and therefore cannot be used as a cost function.

MAE = ……………… (7.1)

Table 7.1: Comparison of models based on MAE

|  |  |
| --- | --- |
| **Model** | **Mean Absolute Error (MAE)** |
| Linear Regression | 9.43E-05 |
| K-Nearest Neighbours | 0.00043 |
| Decision Tree | 0.00033 |
| Gradient Descent | 0.00012 |
| Voting Regression | 0.00012 |

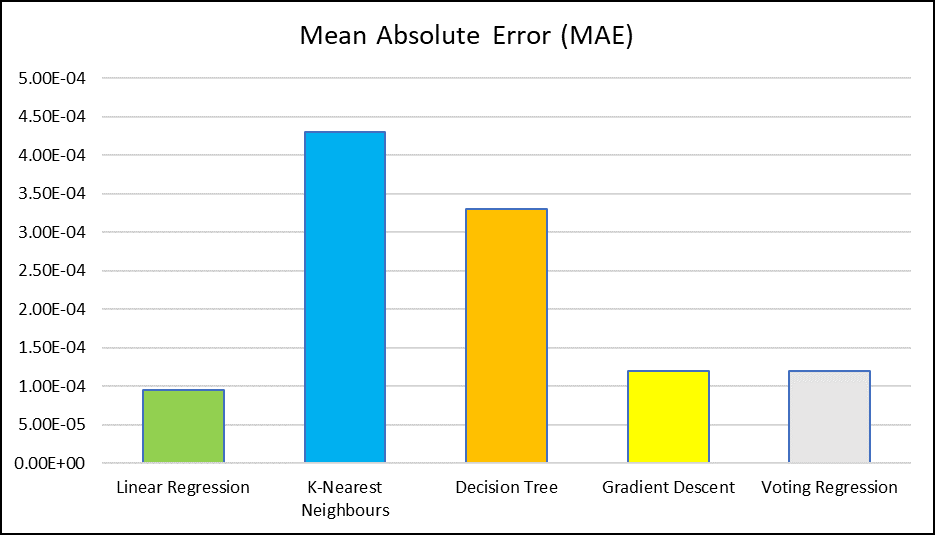


Fig 7.1: Comparison of models based on MAE

From the above fig 7.1, it is evident that Linear Regression model has the least errors and KNN model has the maximum errors.

* Mean Squared Error(MSE) –

It is the average of the squared differences between the actual and the predicted values. Lower the value, the better the regression model. MSE uses the square operation to remove the sign of each error value and to punish large errors.

As we take the square of the error, the effect of larger errors become more pronounced then smaller error, hence the model can now focus more on the larger errors. The graph of MSE is differentiable and therefore it can be used as a cost function.

MSE = 2 …………. (7.2)

**35**

Table 7.2: Comparison of models based on MSE

|  |  |
| --- | --- |
| **Model** | **Mean Square Error (MSE)** |
| Linear Regression | 1.49E-08 |
| K-Nearest Neighbours | 3.62E-07 |
| Decision Tree | 1.72E-07 |
| Gradient Descent | 2.20E-08 |
| Voting Regression | 2.20E-08 |

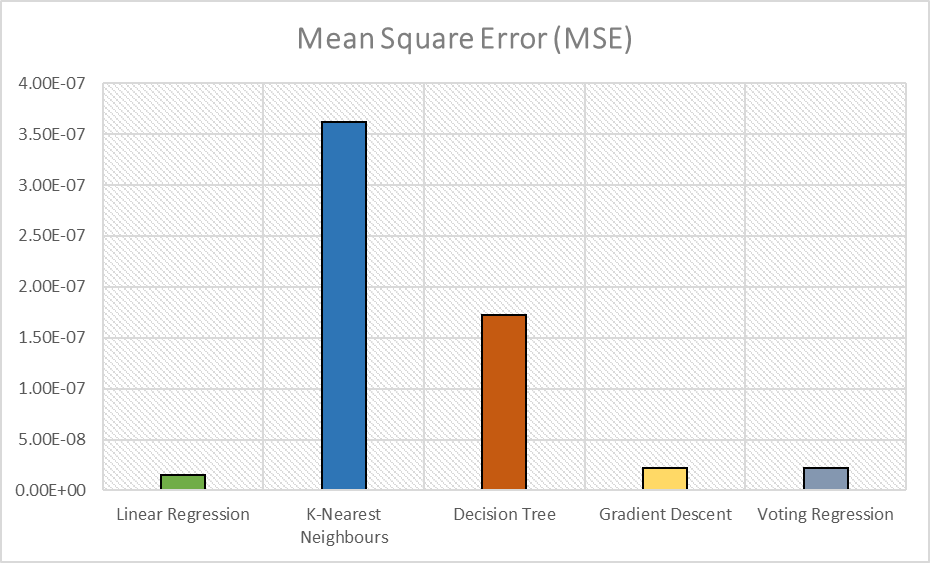


Fig 7.2: Comparison of models based on MSE

From Fig 7.2, it is evident that Linear Regression model has the least errors and KNN model has the maximum errors.

* R Squared (R2 score) –

It tells how close the actual data points are to the fitted line generated by a regression algorithm. It helps us to find the relationship between the independent variable towards the dependent variable.

**36**

It measures the proportion of variance of the dependent variable explained by the independent variable. R2 score ranges from 0 to 1. A larger R-squared value indicates a better fit. If R² is equal to 0, the model is not performing better than a random model. If R² is negative, the regression model is erroneous.

R2 Squared = 1 - …………… (7.3)

where,

SSr = squared sum error of regression line

SSm = squared sum error of mean line

SSR = 2 …………… (7.4)

SSM = 2 …………… (7.5)

Table 7.3: Comparison of models based on R2 score

|  |  |
| --- | --- |
| **MODEL** | **R2 SCORE** |
| Linear Regression | 99.88% |
| K-Nearest Neighbours | 97.23% |
| Decision Tree | 98.68% |
| Gradient Descent | 99.83% |
| Voting Regression | 95.6% |

**37**

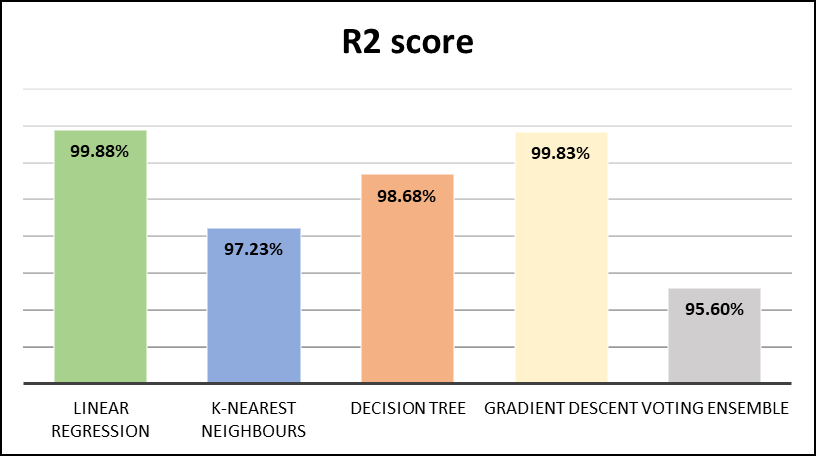


Fig 7.3: Comparison of models based on R2 score

It can be seen in fig 7.3 that LR model gives the highest R2 score.

* Adjusted R-Square –

Adjusted R² is the same as standard R² except that it penalizes models when additional features are added which don’t increase the explanatory power of the regression model. It measures the variation explained by only the independent variables that actually affect the dependent variable. The value of adjusted r-square is always less than or equal to the value of r-square. It ranges from 0 to 1, the closer the value is to 1, the better it is.

Ra2 = 1 - ………..…. (7.6)

where,

n = number of observations

k = number of independent variables

Ra2 = adjusted R2

**38**

Table 7.4: Comparison of models based on Adjusted R2 score

|  |  |
| --- | --- |
| **MODEL** | **ADJUSTED R2** |
| Linear Regression | 99.88% |
| K-Nearest Neighbours | 97.18% |
| Decision Tree | 98.65% |
| Gradient Descent | 99.82% |
| Voting Ensemble | 95.51% |

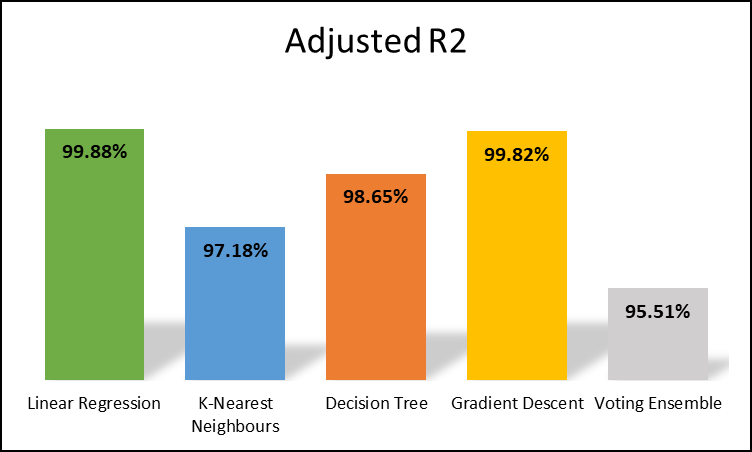


Fig 7.4: Comparison of models based on Adjusted R2 score

It can be seen in fig 7.4 that LR model gives the highest R2 score.

**39**

## 7.2 Discussion

It is evident from the above accuracy metrics, that Linear Regression model performs the best. It gives the lowest MSE (1.49E-08), MAE (9.43E-05) and highest R2 (99.88%) and adjusted R2 score (99.88%). The model gives accurate predictions because of the following reasons:

* There exists a negative linear relationship between the input features and output feature.
* There is very small multi-collinearity (correlation among the independent variables)

**40**

# CONCLUSION

This study used machine learning approaches to optimize the QoS parameters of VoIP traffic. The results revealed that Linear Regression model can accurately predict the MOS from the given input features. Meanwhile, the identification of feature importance based on Linear Regression indicated that delay has the highest significance in developing the model, closely followed by jitter and packet loss has the least significance. The accurate predictions made by the LR model can be attributed to the high negative correlation between MOS and delay and the absence of multi-collinearity.

These findings show that machine learning approaches have potential in determining the QoS parameters and hence can be used to improve the quality of VoIP calls.

**41**

APPENDIX A

import pandas as pd

import numpy as np

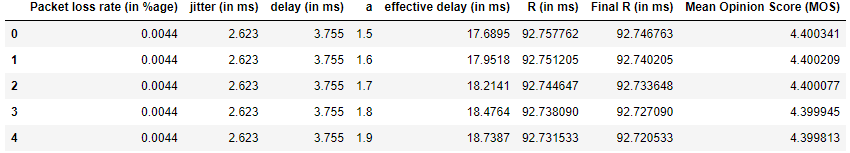
import matplotlib.pyplot as plt

import plotly

import plotly.graph\_objs as go

df = pd.read\_csv('QoS dataset.csv')

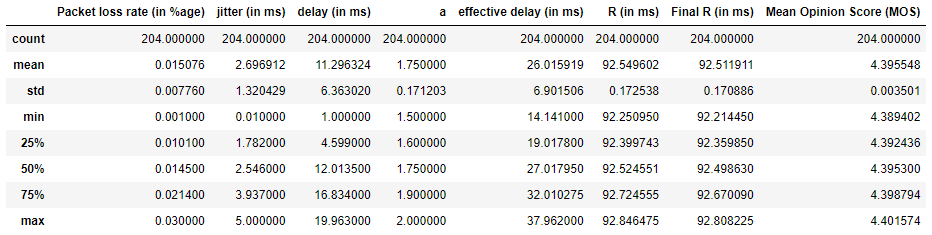
df.head()



df.shape

(204, 8)

df.describe()

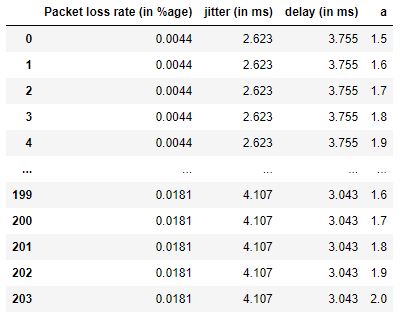


X = df.iloc[:,0:4]

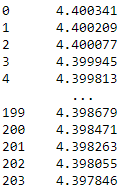
y = df.iloc[:,-1]

X

**42**



y



from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=4)

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

**43**

APPENDIX B

markersize = df.iloc[:,2]

markercolor = df.iloc[:,3]

fig = go.Figure(data=[go.Scatter3d(x=df.iloc[:,0],

y=df.iloc[:,1],

z=df.iloc[:,-1],

marker=dict(size=markersize,

color=markercolor,

opacity=0.9,

reversescale=True,

colorscale='Blues'),

line=dict (width=0.02),

mode='markers')])

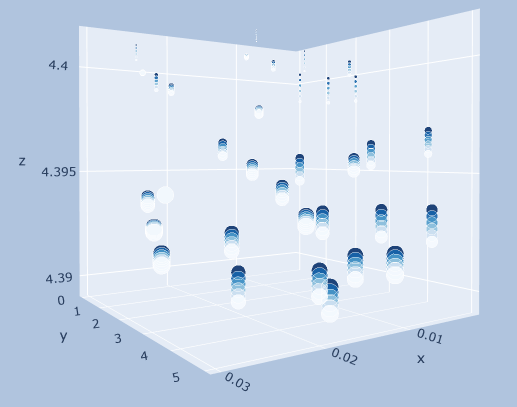
fig.update\_layout(

margin=dict(l=5, r=5, t=5, b=5),

paper\_bgcolor="LightSteelBlue",

)

fig.show()



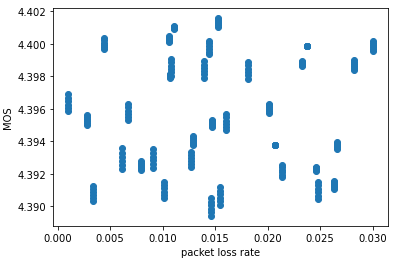
**44**

plt.scatter(df.iloc[:,0], df.iloc[:,-1])

plt.xlabel("packet loss rate")

plt.ylabel("MOS")

plt.show()

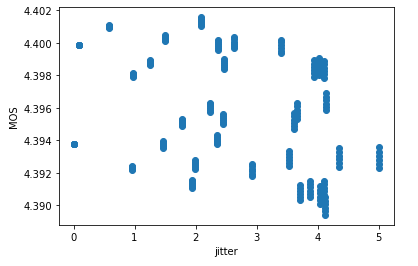


plt.scatter(df.iloc[:,1], df.iloc[:,-1])

plt.xlabel("jitter")

plt.ylabel("MOS")

plt.show()



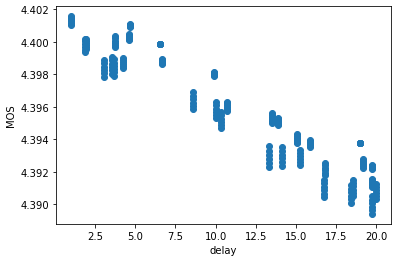
plt.scatter(df.iloc[:,2], df.iloc[:,-1])

plt.xlabel("delay")

plt.ylabel("MOS")

plt.show()

**45**



**46**

APPENDIX C

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import cross\_val\_score

from sklearn.metrics import mean\_absolute\_error,mean\_squared\_error,r2\_score

model1 = LinearRegression()

model1.fit(X\_train, y\_train)

y\_pred = model1.predict(X\_test)

print("Coefficient = ", model1.coef\_)

print("Intercept = ", model1.intercept\_)

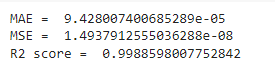


print("MAE = ",mean\_absolute\_error(y\_test,y\_pred))

print("MSE = ",mean\_squared\_error(y\_test,y\_pred))

r2 = r2\_score(y\_test,y\_pred)

print("R2 score = ",r2)



n = 204

k = 4

adj\_r2\_score = 1 - ((1-r2)\*(n-1)/(n-k-1))

print("adjusted R2 = ",adj\_r2\_score)



**47**

APPENDIX D

from sklearn.neighbors import KNeighborsRegressor

accuracy = []

for i in range(1,21):

knn = KNeighborsRegressor(n\_neighbors=i)

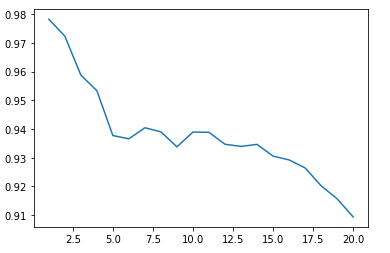
knn.fit(X\_train, y\_train)

y\_pred = knn.predict(X\_test)

acc = r2\_score(y\_test, y\_pred)

accuracy.append(acc)

plt.plot(range(1,21), accuracy)



model2 = KNeighborsRegressor(n\_neighbors= 2)

model2.fit(X\_train, y\_train)

y\_pred = model2.predict(X\_test)

print("MAE = ",mean\_absolute\_error(y\_test,y\_pred))

print("MSE = ",mean\_squared\_error(y\_test,y\_pred))

**48**

r2 = r2\_score(y\_test,y\_pred)

print("R2 score = ",r2)



n = 204

k = 4

adj\_r2\_score = 1 - ((1-r2)\*(n-1)/(n-k-1))

print("adjusted R2 = ",adj\_r2\_score)



**49**

APPENDIX E

from sklearn.tree import DecisionTreeRegressor

model3 = DecisionTreeRegressor(random\_state=4, max\_depth=5)

model3.fit(X\_train, y\_train)

y\_pred = model3.predict(X\_test)

!pip install graphviz

from sklearn import tree

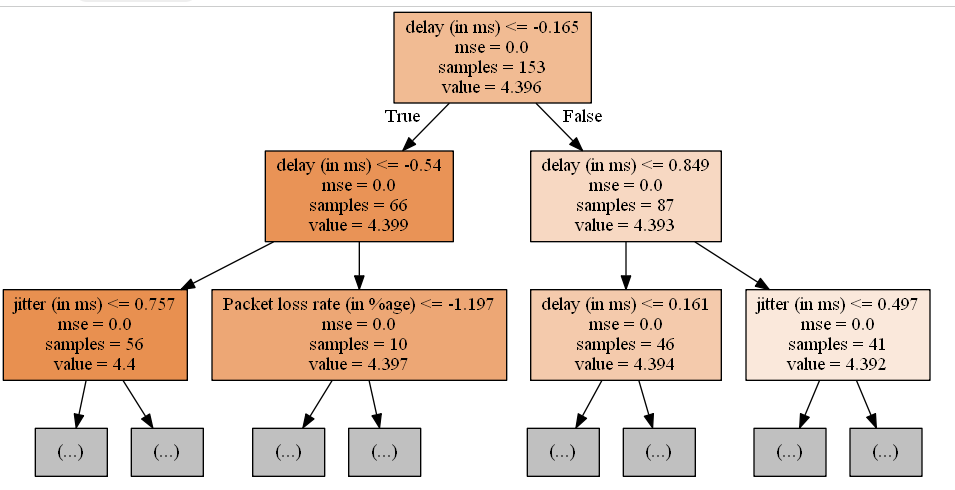
dt = tree.export\_graphviz(model3, out\_file='tree.dot', feature\_names=['Packet loss rate (in %age)','jitter (in ms)','delay (in ms)','a'], max\_depth=2, filled=True)

!dot -Tpng tree.dot -o tree.png

image= plt.imread('tree.png')

plt.figure(figsize=(15,15))

plt.imshow(image)

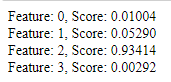


importance = model3.feature\_importances\_

**50**

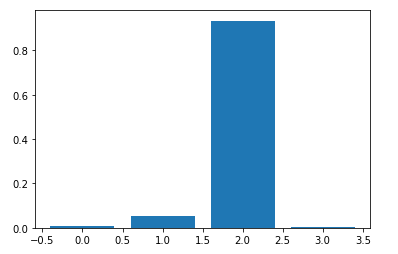
for i,v in enumerate(importance):

print('Feature: %0d, Score: %.5f' % (i,v))



plt.bar([x for x in range(len(importance))], importance)

plt.show

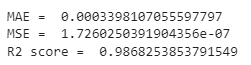


print("MAE = ",mean\_absolute\_error(y\_test,y\_pred))

print("MSE = ",mean\_squared\_error(y\_test,y\_pred))

r2 = r2\_score(y\_test,y\_pred)

print("R2 score = ",r2)



n = 204

k = 4

adj\_r2\_score = 1 - ((1-r2)\*(n-1)/(n-k-1))

print("adjusted R2 = ",adj\_r2\_score)



**51**

APPENDIX F

class GDRegressor:

    def \_\_init\_\_(self,learning\_rate=0.01,epochs=100):

        self.coef\_ = None

        self.intercept\_ = None

        self.lr = learning\_rate

        self.epochs = epochs

    def fit(self,X\_train,y\_train):

        self.intercept\_ = 0

        self.coef\_ = np.ones(X\_train.shape[1])

        for i in range(self.epochs):

            y\_hat = np.dot(X\_train,self.coef\_) + self.intercept\_

            intercept\_der = -2 \* np.mean(y\_train - y\_hat)

            self.intercept\_ = self.intercept\_ - (self.lr \* intercept\_der)

            coef\_der = -2 \* np.dot((y\_train - y\_hat),X\_train)/X\_train.shape[0]

            self.coef\_ = self.coef\_ - (self.lr \* coef\_der)

        print(self.intercept\_,self.coef\_)

    def predict(self,X\_test):

        return np.dot(X\_test,self.coef\_) + self.intercept\_

model4 = GDRegressor(epochs=70, learning\_rate=0.1)

model4.fit(X\_train,y\_train)

y\_pred = model4.predict(X\_test)

**52**



print("MAE = ",mean\_absolute\_error(y\_test,y\_pred))

print("MSE = ",mean\_squared\_error(y\_test,y\_pred))

r2 = r2\_score(y\_test,y\_pred)

print("R2 score = ",r2)



n = 204

k = 4

adj\_r2\_score = 1 - ((1-r2)\*(n-1)/(n-k-1))

print("adjusted R2 = ",adj\_r2\_score)



**53**

APPENDIX G

from sklearn.ensemble import VotingRegressor

lr = LinearRegression()

dt = DecisionTreeRegressor(random\_state=4)

knn = KNeighborsRegressor(n\_neighbors= 7)

estimators = [('lr',lr),('dt',dt),('knn',knn)]

for estimator in estimators:

  scores = cross\_val\_score(estimator[1], X, y, scoring='r2', cv=5)

  print(estimator[0], np.mean(scores))



model5 = VotingRegressor(estimators)

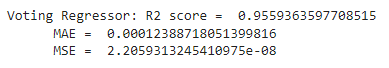
scores = cross\_val\_score(model5, X, y, scoring='r2', cv=5)

r2 = np.mean(scores)

print("Voting Regressor: R2 score = ", r2)

print("      MAE = ", mean\_absolute\_error(y\_test,y\_pred))

print("      MSE = ", mean\_squared\_error(y\_test,y\_pred))



n = 204

k = 4

adj\_r2\_score = 1 - ((1-r2)\*(n-1)/(n-k-1))

print("adjusted R2 = ",adj\_r2\_score)



**54**

REFERENCES

1. **Ribadeneira, Alexander F**., "An Analysis of the MOS under Conditions of Delay, Jitter and Packet Loss and an Analysis of the Impact of Introducing Piggybacking and Reed Solomon FEC for VOIP." Thesis, Georgia State University, 2007.
2. https://data.kemt.fei.tuke.sk/AplikaciePocitacovychSieti/Prednasky/ENSA\_Module\_9.pdf
3. **Shaikh, Junaid, et al**. “Quality of Experience from User and Network Perspectives.” *Annals of Telecommunications - Annales Des Télécommunications*, vol. 65, no. 1–2, 2009, pp. 47–57. https://doi.org/10.1007/s12243-009-0142-x.
4. **Reichl P, Egger S, Schatz R, D’Alconzo A**. The logarithmic nature of qoe and the role of the weber-fechner law in qoe assessment.IEEE; 2010, pp. 1–5.
5. **Elkotob M, Grandlund D, Andersson K, Ahlund C.** Multimedia qoe optimized management using prediction and statistical learning. In: Local Computer Networks (LCN),2010 IEEE 35th Conference on. IEEE: 2010. p. 324–7.
6. **Mushtaq MS, Augustin B, Mellouk A**. Empirical study based on machine learning approach to assess the qos/qoe correlation. In: Networks and Optical Communications (NOC), 2012 17th European Conference on. IEEE: 2012. p. 1–7
7. **Demirbilek E, Grégoire JC**. Machine learning–based parametric audiovisual quality prediction models for real-time communications. ACM Transactions on Multimedia Computing. Commun Appl (TOMM). 2017; 13(2):16.
8. **Demirbilek E, Grégoire JC**. INRS audiovisual quality dataset.ACM; 2016, pp. 167–71.