Addis Ababa University

College of Technology and Built Environment

School of Electrical and Computer Engineering

A Measurement-based Quality of Experience Model for Mobile Video Streaming: A Hidden Markov Model Approach

---

by

Amel Salem Omer

A dissertation submitted to the College of Technology and Built Environment in partial fulfillment of the requirement for the degree of **Doctor of Philosophy**

in

Electrical and Computer Engineering

Date: December 10, 2025

Advisors

Dr. -Ing. Dereje Hailemariam

School of Electrical and Computer Engineering

College of Technology and Built Environment

Addis Ababa University
Ethiopia

Dr. Tsegamlak Terefe

School of Electrical and Computer Engineering

College of Technology and Built Environment

Addis Ababa University
Ethiopia

Addis Ababa University
College of Technology and Built Environment
School of Electrical and Computer Engineering

A Measurement-based Quality of Experience Model for Mobile Video Streaming: A Hidden Markov Model Approach

by
Amel Salem Omer

Approval by Board Examiners:

(Chairman, School Graduate Committee)	*(Signature)*	*(Date)*
Dr. -Ing. Dereje Hailemariam *(Advisor)*	*(Signature)*	*(Date)*
Dr. Tsegamlak Terefe *(Co-advisor)*	*(Signature)*	*(Date)*
Dr. Yalemzewd Negash *(Internal Examiner)*	*(Signature)*	*(Date)*
Dr. Fikreselam Gared *(External Examiner)*	*(Signature)*	*(Date)*

Declaration

I, Amel Salem Omer, hereby declare that the dissertation represents solely my original work and does not incorporate any material from other educational institutions without proper acknowledgment. To the best of my knowledge, it does not contain previously published material by another person without recognition.

Amel Salem Omer _____ _____

(Signature) *(Date)*

This dissertation has been formally submitted for examination under my supervision and endorsement as university advisor.

Dr. -Ing. Dereje Hailemariam _____ _____

(Signature) *(Date)*

Dr. Tsegamlak Terefe _____ _____

(Signature) *(Date)*

Acknowledgments

First and foremost, I would like to thank Allah for providing His divine guidance and support throughout my life. He has been my true leader in everything that I do.

Words cannot express my gratitude to my advisor Dr. -Ing. Dereje Hailemariam for his invaluable guidance from start to finish. He has been pushing me forward through some difficult times, which has made a big difference in my personal life as well. I also could not have undertaken this journey without my co-advisor Dr. Tsegamlak Terefe who generously provided his support and motivation.

I am also thankful to Mr. Tesfaye Addisie, Mr. Abera Dibaba, and Ms. Bethelhem Getu for their collaboration in co-authoring publications related to the core themes of this dissertation and for their support in translating the research into practical applications. My special thanks also go to Dr. Fitsum Asamnew for his support in the development of iNET, the Android-based mobile application used to collect user experience and network service quality data essential for this dissertation.

I would like to express my heartfelt gratitude to my family, especially my mother Wro. Zerthun Amede, my siblings and my spouse, Mr. Ermias Haileyesus, for their unwavering support throughout this journey. Their belief has been a constant source of strength throughout my life. Lastly, I sincerely thank all those who have offered their support and assistance along the way.

Abstract

Demand for mobile multimedia services is rising; as a result, operators routinely monitor network performance using Quality of Service (QoS) metrics such as latency, throughput, and packet loss. While these metrics indicate network performance, they do not fully reflect end-user experience, particularly under fluctuating signal strength, bandwidth constraints, and diverse usage conditions. Though difficult to measure, Quality of Experience (QoE) captures users' perceived service quality, which determines their satisfaction and loyalty. The gap between measurable QoS and subjective QoE remains a key challenge, and bridging it is vital for improving service delivery.

This dissertation addresses the QoS–QoE gap using a data-driven, machine-learning approach based on data collected from a Mobile Network Operator (MNO). For mobile video streaming, it proposes a Hidden Markov Model (HMM)-based model to predict user-perceived QoE from measurable QoS parameters. A custom Android-based data collection mobile app, called iNET, was developed to collect synchronized user-side network metrics and subjective feedback from 550 users across different devices, usage times, and locations in Addis Ababa. The data are used to build a QoS to QoE mapping model. The modeling builds on our earlier experience in analyzing network accessibility, retainability, and congestion using Markov Chain and HMM-based models trained on real-world datasets from an operator's Network management system (NMS). This experience was vital for the proposed QoS-to-QoE mapping, which is the core of this dissertation.

Central to QoS–QoE mapping is identifying the hidden and observable state in the HMM. QoE is typically latent from an MNO perspective, which aligns with treating it as the hidden state in HMM terminology, while network-side QoS metrics as observations. However, in the real system, QoS conditions influence QoE, which can motivate modeling QoS as the underlying hidden state and QoE as the outcome. This creates a modeling paradox between adhering to HMM emission terminology and reflecting the causal behavior of the QoS–QoE process, and it similarly affects whether QoS should be treated as hidden or observable. Accordingly, we evaluated both formulations by alternately treating QoE and QoS as hidden states and comparing their empirical performance.

As the HMM requires a single sequence, we applied Principal Component Analysis (PCA) to reduce the dimensionality of QoS features and then used K-means to quantize them into a single cluster-label sequence. The HMM-based model achieved 98.04% accuracy, outperforming baseline Random Forest (96.15%) and Support Vector Machines (SVM) (49.03%) models. The results show that HMM-based modeling enables accurate QoE prediction and highlight the potential of the iNET measurement tool for use by MNOs.

Keywords: iNET, QoS, QoE, HMM, Markov Chain, Long Term Evolution (LTE), Mobile Network, Modeling, Prediction.

List of Publications

Article in Referenced Journals

1. Omer, A. S., Tufa, A. D., Debella, T. T., and Woldegebreal, D. H., "***Hidden Markov models for predicting cell-level mobile networks performance degradation***". e-Prime—Advances in Electrical Engineering, Electronics and Energy, 9 (September 2024). URL: https://doi.org/10.1016/j.prime.2024.100742

2. Omer, A. S., Yemer, T., and Woldegebreal, D. H., ***"Hybrid K-mean clustering and Markov chain for mobile network accessibility and retainability prediction***". Engineering Proceedings, 18(1), (June 2022). URL: https://doi.org/10.3390/engproc2022018009

Article in Refereed Conference Processing

1. Omer, A.S., and Woldegebreal, D.H., "***Review of Markov chain and its applications in telecommunication systems***". In e-Infrastructure and e-Services for Developing Countries, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 443. Springer, Cham. (2022). URL: https://doi.org/10.1007/978-3-031-06374-9_24

Acronyms

Symbols | A | B | C | D | E | F | G | H | I | K | L | M | N | O | P | Q | R | S | T | U | V

Symbols

2G Second Generation. 1

3D Three Dimension. 97

3G Third Generation. xxvii

4G Fourth Generation. 15

5G Fifth Generation. 1

6G Sixth Generation. 156

A

ABR Adaptive Bitrate Streaming. 22

APIs application programming interface. 25

AR Augmented Reality. 1

ARP Allocation and Retention Priority. 48

ARQ Automatic Repeat Request. 37

AV1 AOMedia Video 1. 36

AvBR Average Bitrate. 21

B

BBR Bottleneck Bandwidth and Round-trip. 35

BCH Broadcast Channel. 41

BGP Broader Gateway Protocol. 29

BLER Block error rate. 50

C

CAC Call Admission Control. 75

CBR Constant Bitrate. 21

CDNs Content Delivery Network. 27

CDR Call drop rate. 80

CNN-LSTM Convolutional Neural Network Long ShortTerm Memory. 156

CNNs Convolutional Neural Networking. 63

CPU Central Processing Unit. 33

CSSR Call Setup Success Rates. 76

D

Diffserv Differentiated Services. 46

DL Downlink. 38

DL-SCH Downlink Shared Channel. 41

DMC Discrete Markov Chain. 68

DMMs Discrete Markov Models. 67

DNS Domain Name system. 29

DRM Digital Rights Management. 23

DSCP Differentiated Services Code Point. 46

DSL Digital Subscriber line. 25

DTMC Discrete-Time Markov Chain. xxiii

E

E-UTRAN Evolved UMTS Terrestrial Radio Access Network. 39

eNodeB Evolved Node B. 40

EPC Evolved Packet Core. 39

ERAB Enhanced Universal Terrestrial RAN RAB. 80

F

FDD Frequency Division Duplex. 39

FEC Forward Error Correction. 37

FR Full-Reference. 58

G

GBR Guaranteed Bit Rate. 48

GPU Graphics Processing Unit. 33

GUI Graphical user interfaces. 2

H

HetNets Heterogeneous Networks. 114

HEVC High-Efficiency Video coding. 36

HLS HTTP Live Streaming. 22

HMM Hidden Markov Model. ix

HSS Home Subscriber Server. 40

HTTP Hypertext Transfer Protocol. 36

I

IntServ Integrated Services. 46

IOT Internet of Things. 1

IP Internet Protocol. 2

IPTV Internet Protocol Television. 9

ISP Internet Service Providers. 63

ITU International Telecommunication Union. 19

ITU-R International Telecommunication Union Radiocommunication. 57

ITU-T International Telecommunication Union Telecommunication. 7

K

KNN k-Nearest Neighbor. 62

KPI Key Performance Indicators. xxii

L

LAN Local area network. 25

LSTM-HMM Long Short Term Memory Hidden Markov Model. 156

LTE Long Term Evolution. ix

M

MBMS Multimedia Broadcast and Multicast Services. 42

Mbps megabits per second. 31

Mbps megabits per second. 31

MBSFN Multimedia Broadcast Single Frequency Network. 42

MC Markov Chain. 14

MCH Multicast Channel (MCH). 42

MIMO Multiple-Input Multiple-Output. 43

MISO Multiple-Input Single-Output. 43

ML Machine Learning. 62

MME Mobility Management Entity. 40

MMQ Modeling Media Quality. 9

MNO Mobile Network Operator. ix

MOS Mean Opinion Score. 9

MOVIE Mobile Video Information Extraction. 64

MPEG-DASH Moving Picture Experts Group Dynamic Adaptive Streaming over HTTP. 22

MT Maximum Throughput. 48

N

NAT Network Address Translation. 40

NFV Network Function Virtualization. 24

NMS Network management system. ix

NN Neural Networks. 8

NR No-Reference. 59

O

OFDMA Orthogonal Frequency Division Multiple Access. 38

OTT Over-the-top providers. 9

P

PCA Principal Component Analysis. ix

PCH Paging Channel. 42

PCRF Policy Control Function. 40

PF Proportional Fair. 48

PGW Packet Data Network Gateway. 40

PRB Physical Resource Block. 45

PRS Performance report system. 80

PSNR Peak Signal-to-Noise Ratio. 9

Q

QCI QoS Classes Identifier. xxvii

QoE Quality of Experience. ix

QoS Quality of Service. ix

QUIC Quick UDP Internet Connections. 64

R

R.H.S. Right-hand side. 106

RAB radio access bearer. 79

RACH Random-Access Channel. 42

RAN Radio Access Network. 40

RF Random Forest. 15

RMSE Root Mean Squared Error. 9

RNC Radio Network Controller. 80

RNNs Recurrent Neural Networks. 63

RR Reduced-Reference. 48

RRC Radio Resource Control. 42

RSRP Reference Signal Received Power. 12

RSRQ Reference Signal Received Quality. 26

RSVP Resource Reservation Protocol. 46

RTT Round Trip Time. 28

S

SAE System Architecture Evolution. 114

SDK Software Development Kit. 26

SDN Software-Defined Networks. 24

SGW Serving Gateway. 40

SIMO Single-Input Multiple-Output. 43

SINR Signal-to-Interference-Plus-Noise Ratio. 12

SISO Single-Input Single-Output. 44

SSIM Structural Similarity Index Measurement. 58

SSR setup success rate. 80

SVM Support Vector Machines. ix

T

TB Transport Block. 42

TCP Transmission Control Protocol. 26

TDD Time Division Duplex. 39

TFT Traffic Flow Templates. 48

TTI Transmission Time Interval. 39

TUQ Testing User-perceived QoS. 8

U

UE User Equipment. 40

UL Uplink. 38

UL-SCH Uplink Shared Channel. 42

UMTS Universal Mobile Telecommunications Service. 9

V

VBR Variable Bitrate. 21

VMAF Video Multi-dimensional Assessment Tool. 51

VoD Video on Demand. 21

VoIP Voice over IP. 20

VOS Video On Demand Service. 37

VQM Video Quality Metric. 58

VR Virtual Reality. 1

Contents

List of Figures

List of Tables

1 Introduction

This chapter lays the foundation for the dissertation. The background explains the importance of mobile network performance and user experience and defines the concepts of QoS and QoE. It also introduces the conventional approaches to QoS-to-QoE mapping, highlighting their limitations. The second section covers the motivation, which is to improve how user experience is assessed in mobile video streaming networks. A model is developed to link measurable network-side parameters to subjective user-perceived experience.

Section 1.3 presents the research objectives, both general and specific. Section 1.4 follows with a review of related literature, while Section 1.5 outlines the methodology used for data collection and modeling. Section 1.6 defines the scope and limitations of the research. The chapter concludes with Sections 1.7 and Section 1.8, which summarize the main contributions and provide an overview of the dissertation's structure.

1.1 Background

Mobile network services were initially designed for voice communication, but their capabilities have expanded considerably over recent decades. The introduction of Second Generation (2G) and 3G networks enabled data transmission and laid the groundwork for the proliferation of mobile-based services. With the development of LTE, network capacity and QoS delivery have improved substantially, supporting increased usage of services such as internet browsing, music streaming, and video calls, as shown in Figure 1.1 below [1][2].

More recently, the emergence of Fifth Generation (5G) networks has introduced further advancements, offering higher data rates, lower latency, and increased connection density [3]. These improvements are expected to support new classes of applications, including Virtual Reality (VR) and Augmented Reality (AR), and Internet of Things (IOT). Despite these advancements, the growing demand for data-intensive services has placed increasing pressure on mobile networks. This surge in demand often manifests as QoS degradation that negatively affect users' quality of experience, such as:

- **Network Congestion and Reduced Throughput:** A high volume of data traffic can

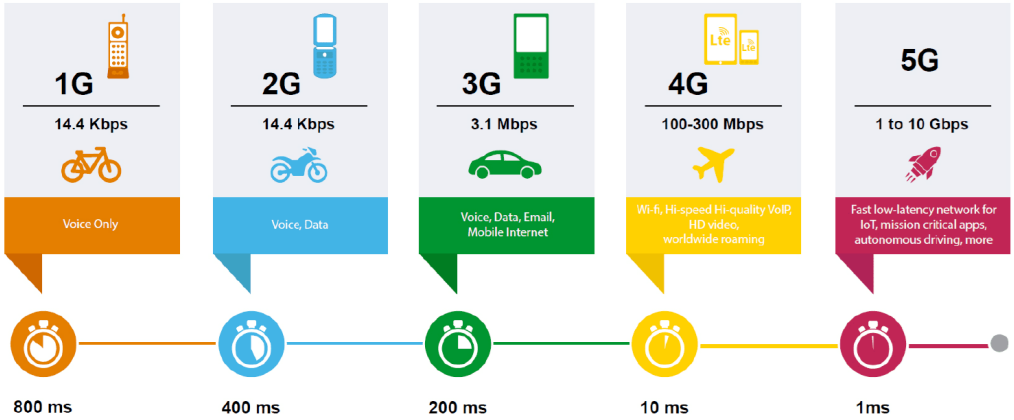


Figure 1.1: Evolution of Mobile Network [2].

cause congestion, leading to increased latency and reduced data rate. This often results in buffering during video playback or delays in interactive applications.

- **Dropped Calls and Connection Problems:** Congestion may also contribute to dropped calls and service interruptions, especially for users who rely on mobile connectivity for routine activities.

These issues highlight the importance of monitoring both QoS and QoE to maintain service quality in mobile multimedia environments.

- **QoS** refers to the technical performance metrics of a network, such as bandwidth, latency, and packet loss. These parameters are measurable from the network side and can be used by operators or service providers to detect and address performance issues [4].

- **QoE**, on the other hand, captures the user's perception of service quality, encompassing subjective factors such as visual clarity, responsiveness, and ease of use. Figure 1.2 illustrates how different components contribute to QoE across service layers. A network with high QoS may still result in poor QoE if the service suffers from frequent interruptions, buffering, or poor interface design [5].

Figure 1.2 presents a layered QoE framework in which user experience is shaped through several interconnected layers [6], ranging from content creation parameters such as audio quality, resolution, and compression to content delivery over Internet Protocol (IP) networks. It also incorporates interactivity elements, including Graphical user interfaces (GUI), semantics, haptics, and digital storytelling, as well as user-centered factors such as trust, satisfaction, and social interaction. At the highest level, the ecosystem layer addresses business models,

Figure 1.2: The QoE influencing factors affects each layer interface in the experience [6].

workflows, security, and system integration. Collectively, these layers form a holistic QoE toolbox, emphasizing that user experience depends on more than network performance alone. Therefore, in competitive service environments, operators must manage both QoS and QoE by combining network optimization with responsiveness to user feedback.

At the top, the ecosystem layer encompasses business models, economic considerations, workflows, security, and system integration. Collectively, these layers form a holistic QoE "toolbox," highlighting that user experience results from the combined influence of technical, interactive, human, and organizational factors rather than network performance alone.

In competitive service environments, managing both QoS and QoE is essential. Operators must not only resolve network-level issues promptly but also be responsive to feedback that reflects user satisfaction.

Traditionally, network operators relied on QoS metrics to guide optimization efforts. While technically measurable, these metrics do not fully account for user perception. In contrast, QoE assessment often requires collecting data directly from users through surveys or behavioral monitoring. This shift toward user-centric evaluation can support more targeted optimization and potentially improve service outcomes [7].

Early attempts to relate QoS and QoE have included basic machine learning approaches such

as fuzzy logic systems [8]. These methods aim to establish a direct mapping between network and user-side metrics. However, they often fall short of capturing the variability introduced by user context, expectations, or tolerance for delay .

As mobile networks become more complex, there is a growing need for advanced modeling techniques that can accommodate diverse user behaviors and dynamic conditions [9]. Data-driven models offer opportunities to:

- Analyze the impact of network conditions on perceived user experience;
- Predict performance degradation before it occurs;
- Optimize resource allocation based on real-time usage patterns.

Ensuring high QoE remains a persistent challenge, primarily because it depends on subjective perception and cannot be directly inferred from network statistics alone [10]. This gap between measurable QoS and subjective QoE forms the basis for the research presented in this dissertation.

1.2 Motivation

As discussed in the previous section, the evolution of mobile networks has changed how users engage with multimedia services. While traditional network optimization focuses on technical parameters such as bandwidth and latency, understanding these alone do not fully reflect user-perceived quality. A better understanding of user experience is required to improve service delivery. This section focuses on the importance of mapping QoS to QoE and the associated challenges and tool requirements.

1.2.1 Importance of QoE-to-QoS Mapping

The growing popularity of mobile multimedia applications, such as video streaming, has contributed to increased pressure on mobile networks. Although newer technologies like LTE and 5G have introduced higher capacity and throughput, service improvement depends not only on technical performance but also on how users perceive the service they get from the networks. QoE reflects the user's perception of the delivered service. It encompasses various factors such as responsiveness, reliability, video clarity, and ease of access, many of which are not captured through standard network measurements. A network that performs well on QoS

parameters may still result in low QoE if users experience frequent interruptions or unstable streaming performance [11].

A reliable QoE-to-QoS mapping process is valuable to MNO for several reasons[12]:

- **Customer Satisfaction and Loyalty:** Understanding how users perceive service performance enables MNO to identify and address specific service issues, leading to improved satisfaction and retention.
- **Resource Allocation Efficiency:** Mapping user experience to network parameters supports targeted adjustments and more effective use of available network resources.
- **Service Differentiation:** In a competitive service environment, maintaining consistent user experience can offer a strategic advantage. MNO that monitor and respond to user experience can deliver more consistent service quality.

1.2.2 Challenges in Measurement and Analysis

Mapping QoS to QoE presents several difficulties. The relationship is not only complex but also influenced by factors external to the network. Unlike QoS, which is based on observable technical parameters, QoE depends on user preferences, usage context, and device capabilities [11]. These factors include:

- **Subjectivity:** User experience is shaped by individual expectations. Some users are more sensitive to delay or buffering than others, even under the same network conditions.
- **Non-linear Response:** Improvements in individual QoS metrics may not always result in noticeable changes in QoE. A higher bitrate, for example, might have minimal effect beyond a certain threshold.
- **Network Variability:** Environmental factors such as user location, network load, or device differences can cause service degradation, complicating the process of defining a stable mapping.

Existing methods that rely on basic machine learning or regression models often fail to capture these complexities, especially across varying environments.

1.2.3 Tool Development Needs

Due to the measurement challenges outlined above, effective tools are required to support QoS-to-QoE mapping [13] [14]. These tools must integrate various types of data and support flexible modeling approaches. The following features are particularly relevant:

- **Large-scale Data Collection:** To characterize user experience more completely, tools should incorporate both subjective input, such as surveys or behavior patterns, and technical performance data.
- **Network Data Integration:** Collection of network metrics including latency, jitter, and throughput should be synchronized with user-side data to build comprehensive datasets.
- **Advanced Modeling Techniques:** Effective mapping requires the use of machine learning methods that can model complex and variable relationships across multiple conditions.

Developing such a tool could enhance the network operator's ability to understand and respond to user needs, enabling more adaptive and user-centered network management strategies. In conclusion, existing QoS-to-QoE mapping methods face several limitations, especially when applied in dynamic mobile environments. This research aims to address these gaps through the design and validation of improved data-driven approaches, as outlined in the following section.

1.3 Objectives

1.3.1 General Objective

The primary objective of this dissertation is to develop a data-driven HMM based framework for mapping QoS parameters to QoE mobile multimedia services, supported by a mobile collection tool called iNET that captures both network performance and user-perceived experience.

1.3.2 Specific Objectives

To achieve the general objective of investigating HMMs for QoE mapping, this research pursued the following specific objectives:

1.3.2.1 Define the Research Scope and Lay the Foundation

- Select appropriate mobile network types, service operation modes, and representative multimedia services that reflect current advancements in mobile networking.
- Identify relevant QoS and QoE parameters based on established International Telecommunication Union Telecommunication (ITU-T) standards for the selected service and network types [15][16].
- Review existing linear and non-linear QoS-to-QoE mapping techniques to identify valuable insights pertinent for the proposed HMM-based model.
- Study the fundamentals of Markov Chain, including its applications in mobile network performance prediction, drawing from previous work on joint accessibility and retainability prediction using Markov Chains and K-Means clustering.
- Implement a Markov Chain model to analyze mobile network accessibility and retainability, thereby reinforcing the foundational understanding necessary for transitioning to HMM.

1.3.2.2 Develop and Evaluate the HMM-based Framework and Mobile Application

- Design and implement HMM-based framework that maps QoS parameters to QoE metrics for mobile multimedia applications, capturing the complex relationship between network measurements and user experience.
- Gain a deeper understanding of the principles of HMMs in relation to user experience prediction, informed by prior research on cell-level performance degradation prediction using HMMs.
- Develop a mobile application, named 'iNET', capable of collecting user-perceived data (e.g., through surveys or user experience ratings) without operator intervention.
- Use the mobile application (iNET) to collect user-reported experience data and corresponding network measurements, which were later used to develop and evaluate the HMM-based QoE mapping model.
- Build and validate the proposed HMM-based QoE mapping model, assessing its effectiveness and accuracy through simulations and real-world experiments across diverse mobile network scenarios.

1.3.2.3 Analyze and Apply the Model's Insights

- Analyze the impact of individual QoS parameters (e.g., bandwidth, latency) and their interactions on perceived QoE using the developed HMM framework. This will generate valuable information on the complex relationship between network performance and user experience.

- Explore the potential applications of the HMM-based mapping framework in user-centric network management and multimedia service optimization. This will involve figuring out how network operators can use the knowledge gathered from precise QoE mapping to enhance user experience, which will ultimately increase customer loyalty and satisfaction.

1.4 Literature Review

This section reviews existing research on QoS-to-QoE mapping in mobile video streaming. It begins by examining conventional approaches for assessing QoE, highlighting their limitations in capturing the complex relationship between technical network parameters and user-perceived experience. The review then looks at recent studies that apply machine learning methods to this mapping task. Techniques such as SVM and Neural Networks (NN) have been explored as alternatives to traditional models [17][18]. It tries to identify knowledge gaps by critically examining the body of existing research, opening the door to the development of a more reliable mapping framework for mobile video streaming [19].

1.4.1 QoE Measurement Approaches for Mobile Streaming

Evaluation of user-perceived experience of mobile video streaming is a key research area, with direct implications for user satisfaction and service adoption [20]. Existing approaches for measuring QoE are generally classified as either subjective or objective, each with its strengths and limitations.

- **Subjective Techniques:** These techniques rely on user feedback to evaluate perceived quality. Common techniques include surveys, focus groups, and lab experiments under controlled network conditions, where the Testing User-perceived QoS (TUQ) method is one such example [21]. While these techniques offer valuable insights, they are costly, time-consuming, and sensitive to memory bias or social desirability effects, which may distort real-world applicability [22].

- **Objective Techniques:** These focus on measuring network-side technical parameters such as bandwidth, latency, packet loss, and video quality indicators like bit-rate or frame rate. Data is typically collected using client-side monitoring or network probes, as in the Modeling Media Quality (MMQ) approach [21]. However, mapping these parameters to perceived QoE is often non-linear. Traditional models struggle to capture this complexity, especially under dynamic mobile network conditions [23]. This highlights the need for more flexible and accurate approaches, such as those based on HMMs.

1.4.2 Machine Learning For QoS-QoE Mapping And Optimization

Reliable evaluation of user experience is essential for optimizing mobile services. To meet user expectations, network operators must go beyond traditional metrics and understand how technical performance translates to QoE. Recent work explores the use of machine learning to model this relationship, owing to its ability to capture complex, non-linear correlations between QoS and QoE [24].

Machine learning models outperform conventional linear techniques by adapting to diverse data patterns and variable network conditions. The availability of large datasets, efficient algorithms, and increased computational power has accelerated progress in this area.

- **Linear Regression:** This technique models direct relationships between QoS metrics and user satisfaction. Studies show it can map indicators like Peak Signal-to-Noise Ratio (PSNR) to Mean Opinion Score (MOS) [24]. However, performance varies by application and environment. For example, models trained on mobile operator-managed services performed poorly when applied to Over-the-top providers (OTT) content providers, where network variability is higher [25].

- **SVM:** SVMs offer robust capabilities for classification and regression tasks. They effectively capture non-linear mappings between QoS and QoE. A study using Universal Mobile Telecommunications Service (UMTS) and LTE data demonstrated accurate results, with Root Mean Squared Error (RMSE) of 11% for voice and 10% for web services [26] [24]. This suggests potential for real-time QoE prediction and network optimization.

- **NN:** NNs extend this potential further. They can model highly complex relationships and are well-suited for adaptive QoE modeling. For instance, one study predicted QoE in Internet Protocol Television (IPTV) services using a multi-layer network, achieving better results than decision trees or SVMs [27]. However, performance can be affected by data imbalance and overfitting [28].

Other models, including fuzzy logic, decision trees, and Kohonen maps, have also been explored [29] [30][14]. The choice of method depends on service type, available features, and data structure.

All things considered, the literature shows how machine learning can help overcome the drawbacks of conventional linear regression techniques and provide more reliable and accurate QoE modeling and mapping. However, further research is needed to develop comprehensive frameworks that can effectively capture the complex and dynamic relationship between network performance and user experience, particularly in the context of emerging mobile multimedia services.

1.5 Methodology

This section presents the methodological steps followed in this research. Two distinct but interlinked workflows were adopted: one focused on modeling mobile network performance using data collected from operator systems, and the other on building a HMM-based framework for mapping QoS parameters to QoE using user-side data collected via the iNET application. The subsections describe the tools, data collection procedures, cleaning and pre-processing steps, and modeling techniques applied in both workflows as shown in Figure 1.3.

1.5.1 Data Collection

1.5.1.1 Operator-based Data Collection

For the network performance modeling, log data was collected from a mobile network operator. The data included performance counters such as call setup success rate, call drop rate, handover success rate, signal strength indicators, and throughput statistics. These metrics were extracted over time from cell-level network management systems. The dataset supported modeling transitions in network behavior under real-world traffic and signal conditions, providing the foundation for Markov Chain and HMM-based modeling.

1.5.1.2 iNET-based Data Collection

To support the QoS-to-QoE mapping effort, a mobile-based application called *iNET* was developed. The app collected both objective QoS metrics (e.g., delay, jitter, packet loss) and subjective user responses during video streaming sessions. The data included timestamped logs of network performance and user ratings of playback quality, enabling the development of a data-driven

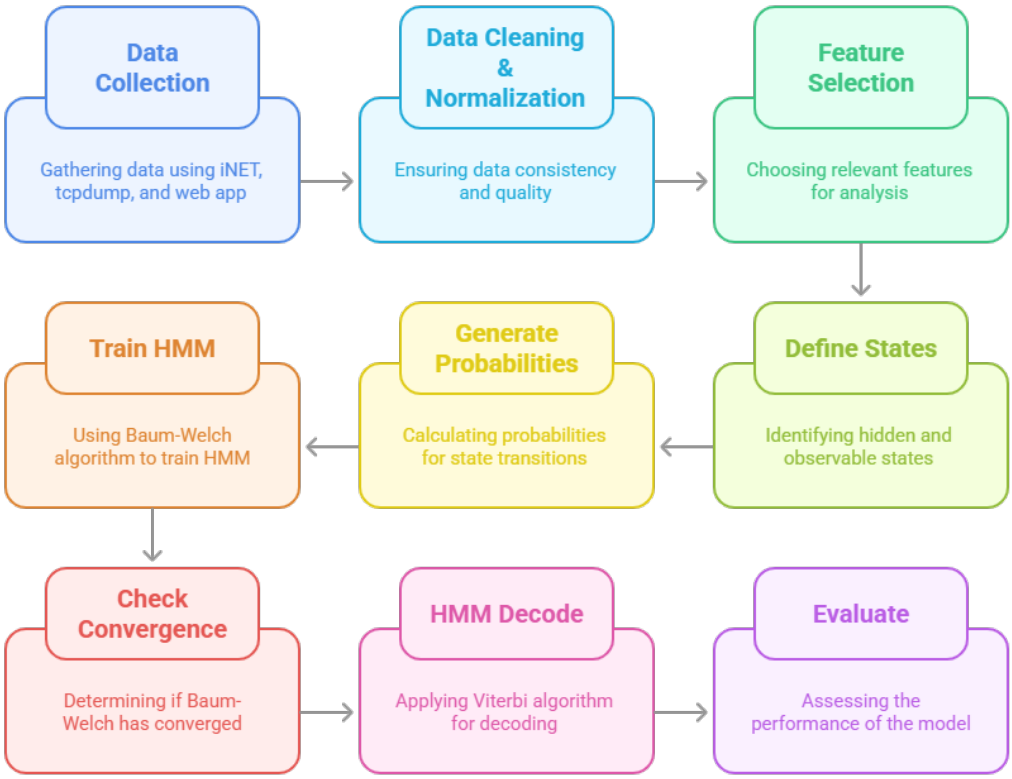


Figure 1.3: Methodology Work Flow

HMM framework for predicting perceived video experience from measurable network-side indicators. The collected data was monitored and exported for further processing through a web-based portal. Ethical protocols were followed throughout the process. All participants provided informed consent, and data was anonymized to ensure privacy and confidentiality.

1.5.2 Data Cleaning and Pre-processing

1.5.2.1 Operator Data Cleaning

The operator-collected data required cleaning to handle missing entries, anomalies, and inconsistencies in time-stamped logs. Redundant or inconsistent records were filtered out. Smoothing techniques and temporal alignment were applied to ensure that state transitions could be modeled correctly. The cleaned dataset preserved the performance variation over time while allowing accurate transition probability estimation.

1.5.2.2 iNET Data Cleaning

The iNET data involved both structured QoS logs and subjective QoE labels. Pre-processing steps included outlier removal, time alignment between network and user logs, and label consistency checks. Cases with incomplete sessions were excluded. The final dataset retained valid QoS-QoE pairs for modeling.

1.5.3 Feature Selection and Dataset Construction

1.5.3.1 Features for Network Performance Modeling

The Markov Chain-based modeling for accessibility and retainability prediction relied on key QoS indicators collected from operator-side measurements. Features included call setup success rate, call drop rate, radio resource utilization, handover success rate, Signal-to-Interference-Plus-Noise Ratio (SINR), and traffic volume. These parameters were transformed into discrete state categories based on performance thresholds and clustered using K-Means to model transitions across service quality states.

For the HMM-based cell-level performance degradation modeling, a distinct set of features was selected to reflect spatial and temporal variations in service quality. Variables such as Reference Signal Received Power (RSRP), uplink and downlink traffic volumes, and handover request rates were used. These metrics supported the learning of hidden state transitions that characterize gradual or abrupt degradation trends in specific network cells.

1.5.3.2 Features for QoS-to-QoE Mapping

The iNET dataset used end-to-end performance indicators such as round-trip time, jitter, and packet loss as input features. These were paired with discrete user-labeled QoE levels (e.g., poor, fair, good). Feature selection focused on variables with consistent patterns across multiple users and sessions, optimizing model robustness.

1.5.4 Markov Chain and HMM-based Modeling

1.5.4.1 Markov Chain Modeling

Markov Chain modeling was applied to the operator data to analyze transitions in network performance states. For example, a network cell might move between high, moderate, and poor accessibility states over time. Transition probabilities between states were computed based on

historical sequences. This approach captured temporal dynamics and performance volatility, laying the foundation for more complex hidden state modeling.

1.5.4.2 HMM-based Degradation Prediction

This stage applied an HMM model to identify hidden performance degradation states in mobile networks. The model was trained using operator-provided data reflecting radio access network behaviors.

Key modeling steps included:

1. **Defining hidden states:** Internal operational states were categorized into levels such as optimal, congested, and degraded, based on domain understanding of network instability patterns.

2. **Selecting observables and emissions:** Measurable indicators like call drop rate, SINR, throughput, and handover failure rate were used to estimate emission probabilities for each hidden state.

3. **Model training and inference:** The model was trained using sequences of observed events, enabling it to predict state transitions and proactively infer future degradation trends under fluctuating load and signal conditions.

This approach allowed proactive performance monitoring and deeper insight into evolving service reliability at the cell level.

1.5.4.3 HMM-based QoS-to-QoE Mapping

The final model focused on predicting perceived QoE using network-side QoS metrics, based on user-side data collected through the iNET mobile platform.

The process followed these steps:

1. **Defining hidden states:** Latent user experience conditions were modeled as states such as smooth playback, minor stalling, and frequent buffering, reflecting subjective user perception.

2. **Selecting observables and emissions:** Observable QoS metrics, such as packet loss, jitter, latency, and buffering duration, were used to determine emission probabilities for each experience level.

3. **Model training and application:** The HMM was trained on time-synchronized QoS-QoE sequences from the iNET app. Once trained, the model could infer perceived QoE based solely on network-side measurements, without continuous user feedback.

This data-driven mapping framework supported scalable QoE estimation and provided a basis for real-time service quality adaptation.

1.5.4.4 Methodological Integration

The modeling framework integrates two distinct but complementary perspectives. Operator-side modeling used Markov Chain (MC) and HMM to capture evolving patterns in mobile network performance. It focused on transitions in signal quality, traffic load, and handover efficiency. These models enabled prediction of service instability using historical patterns in network metrics.

User-side modeling, by contrast, estimated QoE based on observable QoS parameters collected via the iNET mobile platform. A dedicated HMM was trained to associate user feedback with real-time network measurements.

Bringing these two perspectives together allows for a layered understanding of mobile service quality. Network-level predictions support proactive infrastructure management, while user-level estimations provide insight into perceived experience. Their integration enhances reliability and contextual accuracy in multimedia performance assessment.

1.6 Scope and Limitations

1.6.1 Scope

The scope of this research is defined by the modeling focus, data sources, and network environment considered during experimentation and analysis.

- The research investigates three modeling approaches: Markov Chain-based modeling for network accessibility and retainability, HMM-based modeling for network degradation, and HMM-based QoS-to-QoE mapping.
- Two primary datasets were utilized: operator-collected network metrics and user-side data collected using the iNET Android application.
- The operator dataset captures real-world network performance over time, while the iNET data reflects user experiences during mobile multimedia usage in Addis Ababa.

- The network environment focused primarily on 4G LTE due to the unavailability of 5G services during the study period.
- The iNET data collection was crowd-sourced from over 500 Android users, enabling dual-sided analysis but limiting user diversity.
- Only mobile video streaming sessions were analyzed, targeting OTT services with high multimedia bandwidth demands.
- Markov-based models formed the core of the methodology, while SVM and Random Forest (RF) were included for benchmarking.
- Advanced deep learning techniques were not employed, as the research emphasized explainable probabilistic modeling frameworks.

1.6.2 Limitations

While the research provides useful insights into mobile network performance and QoS-to-QoE mapping, several limitations must be acknowledged regarding data generalizability, platform scope, and modeling constraints.

- The iNET data collection was limited to Android platforms; users on iOS and other operating systems were not included.
- The research was geographically restricted to Addis Ababa, which may limit the applicability of results to other urban or rural settings.
- The research exclusively considered video streaming traffic, which may not represent performance under other service types like voice or interactive applications.
- Only Fourth Generation (4G) LTE networks were analyzed; findings may not generalize to newer 5G infrastructures or legacy 3G environments.
- The QoS-to-QoE model relied on self-reported data and network statistics, introducing subjectivity and potential bias.
- While Markov Chain and HMM provided interpretable temporal models, more complex deep learning models may capture nonlinear dynamics more effectively.
- The focus on probabilistic modeling excludes alternative methods that could offer improved accuracy at the cost of interpretability.

1.7 Contributions

This research contributes to the field of mobile network performance modeling and QoS-to-QoE estimation for video streaming services by authoring three well-cited and peer-reviewed journal articles and conference paper.

- **Paper 1 and 2: Integrated Application of Markov Chains and HMMs [31] [32]:** The two publications review and apply Markov Chain and HMM-based modeling to two related domains: operator-side network performance prediction and user-side QoE estimation. It extends previous work by incorporating temporal patterns in network accessibility and retainability and by inferring hidden degradation trends. Building on this foundation, the research proposes a separate HMM that maps observable QoS metrics to user-perceived QoE levels. This dual application provides a comprehensive probabilistic framework for understanding and forecasting service quality.

- **Paper 3: Improved QoE Prediction accuracy [33]:** By training the QoS-to-QoE HMM on mobile-side data collected via the iNET application, the model achieves improved inference accuracy using only network-side metrics. Prior research on cell-level degradation prediction demonstrated that HMMs are effective for capturing latent quality states. This research applies similar principles to predict user satisfaction, enabling scalable estimation without continuous user feedback.

- **Development of a Mobile-Based Data Collection Framework:** In addition to the publications, the research developed a custom Android application, iNET, to collect synchronized user feedback and network performance data during live video streaming sessions. The app captures parameters such as buffering time, playback smoothness, latency, and packet loss, along with user ratings. This setup allows the construction of a training dataset suitable for QoS-to-QoE mapping in real-world conditions.

1.8 Organization

This dissertation comprises six main chapters. The introductory chapter outlines the background, motivation, objectives, and methodology that guide the research. A comprehensive literature review is provided in Chapter 2, covering foundational concepts of QoS and QoE, characteristics of multimedia services, and previous modeling approaches, including traditional and machine learning-based techniques.

Chapter 3 presents Markov Chain and HMM-based modeling techniques applied to mobile network performance data, particularly for analyzing accessibility, retainability, and degradation patterns. The user-side analysis is explored in Chapter 4, where an HMM-based mapping framework is developed using crowd sourced data collected via the iNET mobile application. Chapter 5 discusses the experimental results and evaluates the model performance using established metrics. The final chapter concludes the research, summarizing key contributions and outlining directions for future work.

2 QoS & QoE in Mobile Multimedia Services

This chapter examines the delivery of multimedia services over mobile networks, with a focus on QoS and QoE. It begins by classifying multimedia services based on content types, timing requirements, and delivery strategies, followed by an overview of the ecosystem involved in multimedia streaming. Key performance parameters such as latency, jitter, bitrate, and bandwidth are examined, along with the tradeoffs and adaptation mechanisms required to meet service constraints. The discussion then focuses on LTE networks, detailing their architecture, transport channels, and support for multimedia delivery.

QoS models, including QCI, scheduling, and traffic classification, are reviewed in the context of resource management. The chapter concludes with an overview of QoE concepts, measurement approaches, and recent research linking network conditions to user-perceived quality.

2.1 Multimedia Services: Types and Service Characteristics

Multimedia services are telecommunication services that integrate diverse content types and delivery mechanisms to enhance user engagement. According to International Telecommunication Union (ITU) [34], these services involve the combined use of content such as text, audio, image, and video delivered in an integrated and often synchronized manner. In addition to content, these services are characterized by their temporal behavior, including responsiveness to real-time constraints, and by their delivery strategies, such as constant, variable, or adaptive bitrate control. This section provides a classification of multimedia services based on these contents, temporal characteristics, and delivery mechanisms [35].

2.1.1 Multimedia Services Content Types

Multimedia services integrate various forms of media, including text, audio, video, and images. In many application contexts, these media forms are also referred to as content. This integration enhances communication, enriches user experience, and supports diverse applications such as entertainment, education, and information dissemination [36] [37] As shown in Figure 2.1, in multimedia services, the fundamental content types include the following:

- **Text:** A fundamental medium used to convey structured or unstructured information. It includes elements such as captions, metadata, subtitles, annotations, and user interface labels that support content comprehension and navigation.
- **Audio:** Includes speech, music, narration, and ambient sound, contributing to emotional tone, contextual awareness, and accessibility (e.g., audio guides or screen readers).
- **Image:** Static visual representations such as diagrams, photographs, and infographics, used to complement textual and auditory content and facilitate visual comprehension.
- **Video:** A dynamic media form combining motion and synchronized audio to deliver immersive communication. This includes recorded footage, live streams, and computer-generated animations, used across platforms such as streaming services, virtual classrooms, video conferencing systems, and interactive learning environments.

Figure 2.1: Common Media Formats in Multimedia Services [38].

2.1.2 Temporal Characteristics of Multimedia Services

Multimedia services can be classified based on their temporal characteristics, which describe how sensitive the service is to timing delays during interaction between sender and receiver. These characteristics distinguish between applications that require continuous, low-latency communication and those that tolerate delays and asynchronous delivery. Accordingly, multimedia services fall into two broad categories: real-time and non-real-time services.

- **Real-Time Services:** Typical examples for Real-Time Services include Voice over IP (VoIP), live video conferencing, live streaming, and online gaming. These services require low latency, minimal jitter, and continuous packet delivery to support synchronous interaction. Even brief transmission delays can lead to disruptions such as audio-visual

de-synchronization, lag, or conversational breakdowns due to their high sensitivity to network impairments. Real-time services therefore demand strict QoS guarantees to maintain interactive performance under varying conditions [39].

- **Non-Real-Time Services:** These services operate asynchronously and can tolerate transmission delays. They include Video on Demand (VoD), file downloads, email, and pre-recorded content streaming. Their design emphasizes reliable content delivery rather than immediacy, making use of buffering and retransmission mechanisms to accommodate packet loss or fluctuating bandwidth. As a result, non-real-time services are more robust to network variability and do not require dedicated timing control [40].

2.1.3 Bitrate Control and Delivery Modalities

Bitrate control is fundamental to multimedia service performance, affecting both the perceived QoE and the efficient use of network resources. Different bitrate models are applied either at the content encoding stage (i.e., streaming service provider) or dynamically during content delivery (i.e., by network service provider), with varying implications for latency, buffer management, and QoS provisioning [41]–[43].

- **Constant Bitrate (CBR):** This service delivery transmits media at a fixed bitrate regardless of scene complexity and network condition. While CBR simplifies encoder design and facilitates predictable network planning, it can lead to suboptimal performance. Complex scenes may suffer from visual degradation, while simple scenes may consume excessive bandwidth unnecessarily. CBR is often used in live streaming and real-time communications where timing predictability is critical.

- **Variable Bitrate (VBR):** VBR adjusts the bitrate according to content complexity, allocating more bits to dynamic or detailed scenes and fewer to static ones. This strategy improves compression efficiency and media quality while reducing average bandwidth usage. However, it can produce unpredictable bitrate peaks, requiring larger buffers and careful scheduling, particularly in constrained networks.

- **Average Bitrate (AvBR)**: AvBR encoding is a hybrid approach that targets a specific average bitrate over the entire content duration. It offers a compromise between quality consistency and predictable file size. This model is particularly useful for offline content preparation, such as pre-encoded VoD libraries with constrained storage or transmission quotas.

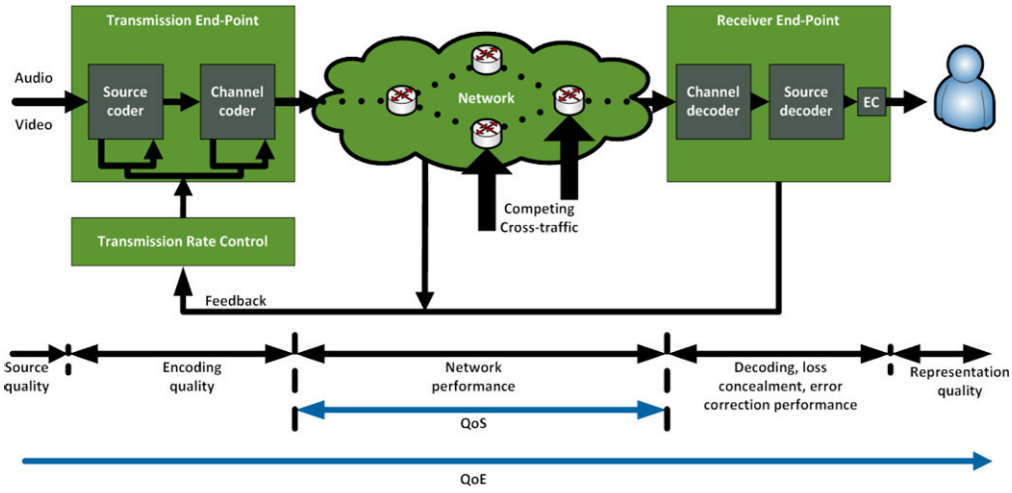


Figure 2.2: Key Stakeholders in the Multimedia Streaming Ecosystem [6].

- **Adaptive Bitrate Streaming (ABR)**: In ABR the client dynamically selects from multiple pre-encoded bitrate representations based on current network conditions and playback buffer status. Protocols such as Moving Picture Experts Group Dynamic Adaptive Streaming over HTTP (MPEG-DASH) and HTTP Live Streaming (HLS) enable seamless transitions between bitrate levels to maintain uninterrupted playback. ABR is essential in variable-bandwidth environments such as LTE networks, as it helps preserve QoE despite fluctuating throughput and latency.

Bitrate control occurs at both encoding and delivery stages. While encoding defines initial media characteristics, delivery-time adaptation ensures smooth playback under varying network conditions. Distinguishing between the two is key to effective multimedia system design.

2.2 Multimedia Streaming Ecosystem

The multimedia streaming ecosystem consists of multiple interconnected actors responsible for producing, delivering, and rendering media content across heterogeneous networks and devices. These actors span from content generators and encoding platforms to transport providers and end-user applications. Each layer of the ecosystem contributes to the end-to-end delivery pipeline, ultimately shaping both the technical performance, i.e., QoS, and the user-perceived experience, i.e., QoE. Figure 2.2 conceptually illustrates the key stakeholders involved in this ecosystem and their interactions.

2.2.1 Content and Platform Providers

Content and platform providers are responsible for initiating the multimedia delivery pipeline. Their responsibilities span content creation, aggregation, preparation, and platform-level access control. The structure, encoding, and availability of multimedia content at this stage directly influence downstream network performance and user experience. Outputs from this layer include raw and encoded media assets, such as multi-bitrate renditions suitable for adaptive delivery [44] [45].

- **Content Creators:** This group includes individuals, academic institutions, and commercial entities such as media studios and television networks. They produce original multimedia assets whose characteristics, resolution, format, and duration, directly impact encoding strategies and overall QoE.
- **Content Aggregators and Publishers:** Operating between content creators and platforms, they organize and curate materials from various sources. OTT services, educational repositories, and news platforms are common examples. Their publishing policies govern access models, monetization strategies, and content categorization.
- **Encoding and Transcoding Services:** Media encoding vendors, in-house processing pipelines, and cloud-based transcoding platforms fall under this role. These services generate multiple bitrate representations (e.g., CBR, VBR, ABR) to support adaptive streaming and device compatibility.
- **Digital Rights and Licensing Authorities:** Standards organizations, licensing boards, and digital certificate authorities are responsible for securing intellectual property and enforcing distribution constraints through mechanisms like Digital Rights Management (DRM), encryption, and access control.

2.2.2 Content Delivery Infrastructure

The content delivery infrastructure is responsible for the efficient and scalable distribution of multimedia streams between service platforms and end-user devices. This layer incorporates usually cloud-based technologies and systems that host, cache, route, and adapt content in response to variable network and device conditions [46].

- **Streaming Platforms:** These systems deliver audio and video content using adaptive bitrate streaming protocols such as MPEG-DASH and HLS [47]. They manage media

manifests, segment requests, and playback adaptation logic. Typical implementations involve cloud-based distribution architectures designed to support diverse network environments.

- **Audio Delivery Services:** Optimized for low-bitrate and continuous playback, these services rely on lightweight transport protocols, compression techniques, and predictive buffering. Examples include platforms serving music, podcasts, or speech-driven content.
- **Interactive Services:** Cloud-based real-time applications such as gaming and remote rendering platforms require ultra-low latency and precise synchronization. These services use high-performance edge computing infrastructure to render and transmit audiovisual content in response to user inputs with minimal delay.
- **Telemetry and Analytics Integration:** Delivery platforms often incorporate monitoring tools that track playback statistics, segment download times, bitrate transitions, and failure events. These measurements support real-time adaptation and long-term optimization of QoE across user populations.

2.2.3 Access Network Providers

Access network providers facilitate the core network and last-mile connectivity between streaming platforms and end-user devices. Their infrastructure and resource management strategies directly impact the delivery performance of multimedia content across mobile, fixed, and wireless access technologies.

- **Radio-based Access:** In mobile networks such as LTE and 5G [48], operators allocate radio spectrum and manage scheduling to ensure sufficient bandwidth, link stability, and continuous coverage, particularly under high user load or mobility. The core network supports these efforts through session management, IP connectivity, and efficient routing, all of which are essential for sustaining multimedia services.
- **QoS Enforcement:** Traffic prioritization is achieved through mechanisms such as QCI, dynamic resource allocation, and buffer provisioning [49]. These techniques enable the network to distinguish between service types and maintain the performance of delay-sensitive multimedia applications.
- **Resource Optimization and Virtualization:** Access networks increasingly employ Mobile Edge Computing, local caching, and virtualization technologies (e.g., Software-Defined Networks (SDN)/ Network Function Virtualization (NFV)) to reduce latency,

increase throughput, and offload central network components [50][51]. These enhancements are particularly effective for real-time and adaptive streaming services.

- **Access Variability Across Technologies:** Fixed broadband (e.g., fiber, Digital Subscriber line (DSL)), wireless Local area network (LAN) (e.g., Wi-Fi), and satellite links each exhibit distinct performance profiles [52]. Their influence on multimedia delivery depends on factors such as congestion, mobility, and coverage consistency [53].

2.2.4 End-User Devices and Playback Applications

The end-user layer comprises the devices and playback applications responsible for decoding, rendering, and interacting with multimedia streams. Although not involved in content production or delivery, this layer plays a pivotal role in determining the user-perceived QoE [54].

- **Device Capabilities:** The quality of playback is influenced by screen resolution, processing power, hardware acceleration, and codec support. Smartphones, smart TVs, laptops, and tablets exhibit substantial variation in performance and compatibility.
- **Media Player Applications:** Streaming clients such as mobile video apps or browser-based players manage playback logic, implement ABR algorithms, and monitor buffer status. Examples include open-source frameworks like ExoPlayer and AVPlayer integrated into native operating systems.
- **Operating System and Media Frameworks:** Client-side software environments provide the necessary application programming interface (APIs) for decoding, DRM enforcement, and network communication. These frameworks also report telemetry used for session-level quality assessment.
- **User and Environmental Context:** Factors such as device mobility (e.g., in-vehicle usage), multitasking, and ambient network conditions (e.g., Wi-Fi vs. cellular) influence adaptive logic and responsiveness.

2.2.5 Monitoring and Analytics Layer

Monitoring and analytics systems are essential for assessing the performance of multimedia streaming services from both network and user perspectives. These tools enable service providers and researchers to identify performance bottlenecks, optimize resource allocation,

and adapt delivery strategies. The tools encompasses multiple monitoring approaches, ranging from infrastructure-focused metrics to application-level quality tracking.

2.2.5.1 Network-Centric Performance Measurement

This category includes tools that collect essential connectivity metrics typically visible to end users, such as download and upload throughput, latency, and jitter. In addition to these, some tools measure deeper radio-level parameters, including packet loss, RSRP, and Reference Signal Received Quality (RSRQ) to assess signal strength, interference level, and link stability in mobile networks. However, most commercial applications emphasize basic indicators that reflect perceived network performance from the user perspective [55].

Measurement is typically conducted using active testing or crowd-sourced passive logging. Platforms such as *Ookla Speedtest* and *OpenSignal* are examples of widely used network-centric measurement tools shown in the Table 2.1. *Ookla* performs active, client-initiated speed tests using multiple Transmission Control Protocol (TCP) flows toward geographically distributed servers, reporting on throughput and latency [56]. *OpenSignal* uses background sampling across a wide user base to estimate signal strength, latency, jitter, and application-level scores such as video and gaming experience [57].

While these tools provide broad geographic and temporal coverage, they exhibit several limitations for scientific research. First, both platforms offer restricted access to raw measurement data, limiting reproducibility and analysis flexibility. Second, their proprietary aggregation methods and client-side sampling introduce potential biases due to user demographics, device types, and server proximity. Lastly, they do not capture media-specific indicators like buffer occupancy, rebuffering frequency, or perceived QoE, which are critical for end-to-end streaming evaluations.

2.2.5.2 Application-Centric QoE Analytics

Application-level analytics systems are designed to monitor playback-related metrics such as startup delay, rebuffering, resolution switching, and session failure. These systems often rely on embedded telemetry collected from media players via dedicated Software Development Kit (SDK)s. This layer supports personalized delivery optimization and service diagnostics. Aggregated QoE metrics help streaming platforms adjust bitrate selection algorithms, content placement strategies, and user engagement models. However, such data is typically proprietary and inaccessible for external research [58] [59].

Table 2.1: Comparison of Network Performance Measurement Tools.

Feature	Ookla Speed test	OpenSignal
Measurement Approach	Active (client-initiated speed test to nearest server)	Passive (background data collection on user devices)
Metrics Collected	Download and upload speed, latency, jitter	QoS and QoE proxies (video/gaming experience, latency, signal strength)
Geographic Coverage	Broad global presence via distributed test servers	High coverage through crowdsourced mobile app data
Bias & Limitations	Subject to test location bias, device and user selection effects	Affected by device types, user distribution, indoor/outdoor conditions
Access to Data	Aggregated reports; raw data via commercial API (restricted)	Summary reports public; raw data limited and commercial
Use Case Fit for Research	Useful for benchmarking; lacks application-layer detail	Valuable for coverage trends; lacks control and protocol-level data
Known Gaps	No media-specific metrics (e.g., buffering events, resolution drops)	Limited temporal granularity; lacks support for custom indicators

2.2.5.3 Hybrid Monitoring Systems

Hybrid systems integrate passive device-side logging with active test mechanisms and contextual metadata. These systems enable correlation between network-layer behavior and application-layer outcomes. Examples include client-side analytics frameworks integrated into mobile or embedded players, which log segment downloads, bitrate switching events, and playback errors. Such hybrid approaches facilitate root-cause analysis and adaptive model training by connecting observable user experiences with underlying transport dynamics.

2.2.5.4 Data-Driven Optimization and Feedback Loops

Advanced analytics pipelines apply machine learning to historical monitoring data to inform system adaptation. Predictive models guide streaming parameter adjustments, prefetching strategies, and dynamic Content Delivery Network (CDNs) selection. In real-time, feedback loops enable rapid error detection and mitigation, supporting consistency in QoE across diverse network conditions and device environments [60].

2.3 Technical Requirements and Delivery Tradeoffs

Multimedia services, especially video streaming, impose stringent and multifaceted technical requirements across communication networks. These requirements span from low-level packet delivery to high-level user experience metrics. Failure to meet them can result in noticeable performance degradation, including latency, stalls, and poor visual fidelity. This section outlines the key parameters that govern delivery quality, how they are measured in real-world systems, and how they influence user-perceived QoE.

2.3.1 Key Parameters in Multimedia Delivery

The following parameters are fundamental to the quality and efficiency of video streaming services. Each parameter is defined technically, described in terms of real-world measurement practices, connected to the multimedia ecosystem, and evaluated for its impact on end-user experience.

2.3.1.1 Latency (Delay)

Latency, or end-to-end packet delivery delay, is the time required for data to move from a content server to the user device. It is most commonly measured as Round Trip Time (RTT), which can be estimated using active probing tools (e.g., ICMP/TCP echo tests such as Ookla Speedtest) or passively through application-layer timestamps. Latency is a stochastic variable so that its variability necessitates reporting metrics such as the average (mean), minimum/maximum, and high-percentile values (e.g., 95^{th} or 99^{th} percentiles) to fully capture user experience degradation [61], [62]. There are two primary latency metrics:

- *One-way latency* refers to the time it takes for a packet to travel from the source to the destination in a single direction; it provides a more accurate assessment of transmission delay but requires synchronized clocks at both ends.
- *RTT latency* measures the total delay for a packet to travel from the source to the destination and back; it is easier to measure since it does not require clock synchronization, but it combines forward and return path delays.

High latency negatively affects the quality of video streaming, especially for live and interactive services. Research shows that latencies above 5–10 seconds significantly reduce user engagement in live video applications, and delays beyond 30–90 seconds render time-sensitive

content (e.g., sports) frustrating due to event-to-screen lag [63]. Furthermore, sub-second latency is critical in interactive applications such as cloud gaming and live auctions, where delay impairs responsiveness [64]. Latency sources are distributed across the delivery ecosystem:

- **Content Platform:** Processing tasks such as video encoding, transcoding, and chunk preparation at the content delivery platform can introduce several seconds of latency before playback begins.
- **Network Operator:** Network-related processes, including Domain Name system (DNS) resolution, TCP handshake initiation, Broader Gateway Protocol (BGP) route propagation, queuing at intermediate routers, and long-distance data transmission, all contribute to RTT delays.
- **End-User Device:** On the user's device, operations such as buffer initialization, manifest file parsing, and video rendering contribute to startup and playback latency, particularly on resource-constrained smartphones and set-top boxes.

Understanding latency distributions is essential because mean values often hide problematic tail behavior. Percentile reporting (e.g., 95^th percentile RTT) captures those high-latency spikes that frequently trigger playback stalls or resolution downgrades, even if average performance appears acceptable [59].

2.3.1.2 Jitter

Jitter refers to the variability in packet inter-arrival time, or in other words, how much delay fluctuates around its average. While latency reflects the average delay, jitter quantifies its inconsistency, typically measured as the standard deviation or variance of latency over time [63].

- **User Impact:** In video streaming, high jitter causes buffer underflows, stuttering, and playback instability. It is especially problematic in real-time or low-latency scenarios, where it manifests as frame drops or audio-video mismatches. Temporal unpredictability introduced by jitter makes it difficult for ABR algorithms to maintain stable playback quality [65]. Jitter is typically quantified using the following methods, which provide insight into the variability of packet arrival times.
  - Network monitoring tools, such as OpenSignal or Wireshark, which calculate jitter based on variations in packet arrival timestamps.

  - Application-level analytics that estimate jitter from the irregular timing of segment downloads.

  The main causes of jitter across the multimedia delivery ecosystem are outlined below:

  - *Network Layer:* Causes include queue instability, dynamic routing path changes, traffic shaping or policing mechanisms, and randomly occurring packet losses.
  - *User Device:* Inefficient buffering mechanisms and unsynchronized segment fetching cycles contribute to increased jitter at the client side.

- **Latency-Jitter Relationship:** Jitter complements latency by describing how much packet delay varies over time. A system with low average latency but high jitter can produce unstable video playback. Conversely, a system with moderate latency but low jitter tends to support smooth streaming, provided that buffers are properly dimensioned [45].

- **QoE Implication:** Empirical research indicates that jitter often has a more significant impact on user satisfaction than latency. Its role in causing rebuffering events and playback instability makes it a key factor in determining overall QoE [59], [63].

2.3.1.3 Packet Loss Rate

Packet loss rate refers to the proportion of data packets that are either lost or corrupted in transit and fail to reach the destination. In video streaming, even low packet loss can lead to severe degradation in playback quality, including frozen frames, pixelation, and audio distortions. Measurement is typically performed using:

- *Active probing tools* such as Ookla or RIPE Atlas that send test packets and count received acknowledgments.

- *Passive application logs* that detect missing video segments or TCP retransmission events.

- **User Impact:** Packet loss can disrupt the decoding of video streams, especially for compressed formats like H.264 or HEVC where lost frames may affect the decoding of subsequent frames [66].

- **Causes Across the Ecosystem:** The causes of packet loss across the ecosystem can be categorized as follows:

  - *Network Layer:* Congestion, buffer overflows, faulty routing paths, or wireless interference.
  - *Content/Application Layer:* Unreliable transport configurations or absence of error correction mechanisms.

- **QoE Implication:** Empirical evidence suggests that packet loss significantly contributes to playback interruptions, making it a primary driver of rebuffering events and subsequent user abandonment during video streaming sessions [67].

2.3.1.4 Bitrate

Bitrate is the volume of data used to represent video content per unit time, expressed in kbps or megabits per second (Mbps). It determines the detail and clarity of visual output. Bitrate is dynamically adapted in ABR systems based on estimated throughput.

- **Measurement:** Bitrate is calculated by dividing the size of a video segment by the download duration, using player analytics APIs or traffic capture tools.
- **User Impact:** High bitrates enhance visual fidelity but may cause buffer depletion in constrained bandwidth scenarios. Bitrate oscillation can also degrade perceived quality [68].
- **QoE Relevance:** Studies confirm a strong correlation between user satisfaction and stable, sufficient bitrate especially in HD/4K content [63].

2.3.1.5 Bandwidth

Bandwidth refers to the *maximum data transfer capacity* of a network link, typically measured in megabits per second (Mbps). It defines the upper limit on how much data can be transmitted over a connection within a given time, distinguishing it from bitrate, which represents the actual data consumption rate of a media stream. While bitrate adapts based on the needs of a video or audio stream, bandwidth reflects the potential capacity available on the network. Bandwidth is typically measured using tools such as iPerf, Speedtest, or by analyzing interface counters and throughput logs from media players [69].

In multimedia service delivery, limited or variable bandwidth is a common constraint, often resulting in rebuffering, delayed startup, or reduced video resolution. High bandwidth availability allows for stable delivery of high-quality streams, while insufficient or fluctuating bandwidth forces ABR algorithms to reduce video quality to maintain continuous playback.

Bandwidth availability is influenced by several factors, including radio spectrum allocation in wireless networks, backhaul capacity, network congestion, and physical interference. In mobile networks such as LTE, bandwidth is dynamically influenced by radio conditions, scheduling policies, and channel quality metrics such as RSRP and RSRQ [70].

2.3.1.6 Resolution

Resolution refers to the spatial dimension of video frames, such as 720p, 1080p, or 4K. Higher resolutions improve detail but demand higher bitrate and more robust network capacity.

- **Measurement:** Typically logged through playback metadata or ABR decision records.
- **User Impact:** Viewers perceive higher resolution as clearer and more immersive. However, high resolution combined with insufficient bitrate may cause visual artifacts (e.g., blurring, blockiness).
- **QoE Relevance:** ABR algorithms adjust resolution to maintain continuous playback under fluctuating network conditions [59].

2.3.1.7 Startup Delay

Startup delay is the interval between the user's request to play a video and the rendering of the first frame. It is a critical early-stage QoE metric.

- **Measurement:** Collected via playback SDKs or client logs that track user interaction and media start events.
- **User Impact:** Delays exceeding 2 seconds lead to increased abandonment. Empirical results show each additional second of delay results in up to 5.8% more users leaving [64].
- **Causes Across the Ecosystem:**
  - *Content Server:* Slow manifest or segment fetch.
  - *Network:* Latency in DNS resolution or CDNs lookup.
  - *Device:* Decoding lag, buffer threshold configuration.

2.3.1.8 Rebuffering Ratio/Frequency

Rebuffering refers to playback interruptions caused by empty buffers. Ratio indicates the fraction of time spent rebuffering; frequency counts stall events.

- **Measurement:** Player telemetry logs stall durations and frequencies during a session.
- **User Impact:** Rebuffering is among the most visible and frustrating impairments. It disrupts engagement and often leads to abandonment [67].
- **QoE Relevance:** Rebuffering frequency and severity are central in most QoE models, often weighted more than bitrate or resolution changes.

2.3.1.9 Resolution Switching

Resolution switching refers to the *dynamic change in video resolution* during a streaming session, often controlled by ABR algorithms in response to changing network or device conditions. This process allows the media player to optimize playback quality and avoid buffering by scaling video resolution up or down depending on the available bandwidth, buffer occupancy, and playback performance metrics. The switching behavior is typically recorded by media player logs and analytics dashboards that capture stream profile transitions.

From the user's perspective, resolution switching can affect the perceived visual quality of the content. Smooth and infrequent switches are generally tolerated, while ABR frequent changes lead to a degraded and unstable viewing experience. Resolution switching is triggered by several ecosystem factors, including fluctuating network throughput, congestion, and client-side resource limitations [71]. For instance, sudden drops in estimated available bandwidth may lead the ABR engine to downgrade resolution to avoid playback stalls. Device-related constraints such as Central Processing Unit (CPU) or Graphics Processing Unit (GPU) overload may also cause resolution adjustments even when the network condition remains stable.

2.3.1.10 Session Failure

Session failure denotes the *inability to initiate, sustain, or complete* a multimedia streaming session. This includes failures during startup (e.g., manifest or segment fetch errors), mid-session interruptions, or unexpected terminations before completion. Session failures are high-level reliability indicators and reflect systemic problems in the end-to-end streaming pipeline, from CDNs to client [72].

Users typically experience session failure as an inability to watch the requested content or as an abrupt termination of playback. These events often result in frustration, reduced engagement, and in many cases, complete abandonment of the service. Technically, session failures can originate from DNS resolution issues, transport-layer disruptions, expired authentication tokens, misconfigured player settings, or errors in fetching content from CDNs edge servers [73]. Sudden changes in network connectivity, such as Wi-Fi to cellular handovers or signal loss in mobile environments, also contribute. Diagnosing session failures requires coordinated logging from player SDKs, CDNs status analytics, and server-side delivery metrics.

2.3.2 Tradeoffs in Multimedia Delivery

Multimedia delivery systems must constantly balance conflicting service requirements to ensure satisfactory QoE. For instance, real-time applications prioritize minimal delay, while high-definition streaming demands maximum visual fidelity. However, network bandwidth, device capabilities, and user expectations do not always permit these objectives to be met simultaneously. As a result, service providers adopt adaptive strategies to manage tradeoffs, typically using bitrate control, resolution adaptation, and delivery modality switching to optimize playback under constrained or variable conditions [74].

Video streaming platforms such as YouTube and Netflix exemplify this balance: by providing multiple resolutions and allowing adaptive bitrate switching, they dynamically align quality with network and device conditions [75]. These mechanisms help minimize rebuffering and startup delay while preserving an acceptable level of visual quality.

Common tradeoff scenarios include:

- **Latency vs. Resolution:** Real-time applications such as video conferencing prioritize low latency over visual fidelity (e.g., high resolution). Such systems often reduce resolution to lower bitrate and buffering delay, and may disable Forward Error Correction to avoid added redundancy and recovery wait time, at the expense of reduced loss protection (i.e., higher risk of artifacts or brief stalls under packet loss)

- **Bitrate vs. Buffering Stability:** To prevent rebuffering, adaptive streaming systems lower the video bitrate in constrained networks, sacrificing sharpness for smooth playback.

- **Startup Delay vs. Initial Quality:** Fast startup is achieved by loading low-resolution segments first. Higher quality is gradually introduced after playback begins.

- **Energy Efficiency vs. Fidelity:** On mobile devices, streaming applications reduce resolution or frame rate to conserve battery power, particularly when running on low battery or in power-saving mode.
- **Network Utilization vs. User Fairness:** In shared networks, ABR clients may throttle video quality to reduce overall congestion and ensure fair bandwidth distribution among users.
- **Adaptability vs. Visual Consistency:** Frequent bitrate/resolution switching improves adaptability but causes noticeable quality fluctuations. Some services smooth transitions to avoid user-perceived instability.
- **Caching Efficiency vs. Personalization:** Edge caching optimizes delivery by serving popular content from nearby nodes. However, highly personalized or live content limits caching potential, increasing delivery delay.

These tradeoffs are not limited to content providers; network operators also participate through mechanisms such as traffic shaping, congestion control algorithms (e.g., Bottleneck Bandwidth and Round-trip (BBR)), and differentiated QoS policies. CDNs further support the balance by caching popular content closer to users, reducing latency and buffering. Ultimately, both application and infrastructure layers cooperate to optimize for service-specific QoE targets under varying conditions.

2.3.3 Streaming Techniques and Adaptation Mechanisms

To meet the stringent technical requirements of multimedia services while maintaining QoE, both service providers and network operators implement a variety of adaptive mechanisms and encoding techniques. These techniques help reconcile competing demands such as quality versus latency, resolution versus bandwidth, and stability versus responsiveness.

2.3.3.1 Streaming Techniques

Streaming techniques specify the architectural and protocol-level methods used to transport multimedia content from servers to end users.

- **Progressive download streaming** In this technique, media files are downloaded sequentially, and playback begins once sufficient data is buffered. While simple to implement, progressive download offers limited adaptability to fluctuating network conditions and may lead to inefficient bandwidth usage under variable throughput scenarios [40].

- **Live streaming** Live streaming delivers content in real time with strict latency constraints and is commonly used for events, gaming platforms, and video conferencing. Protocols such as low-latency variants of HLS and MPEG-DASH are widely used. This technique prioritizes timeliness over buffering stability, making it sensitive to packet loss and bandwidth variation.

- **On-demand streaming** VoD services deliver pre-recorded content that allows flexible buffering and adaptive bitrate selection. This technique supports higher compression efficiency and playback stability, making it suitable for services such as Netflix and YouTube.

- **Segment-based HLS streaming** Modern streaming systems divide video content into small time-based segments delivered over Hypertext Transfer Protocol (HTTP). Techniques such as MPEG-DASH and Apple HLS enable scalable and friendly delivery while supporting dynamic adaptation at the client side [76]. It also forms the foundation for most adaptive mechanisms in current multimedia services.

2.3.3.2 Service Provider-Side Adaptation

- **Video Codecs (H.264, H.265 / High-Efficiency Video coding (HEVC), AOMedia Video 1 (AV1)):** Video codecs compress raw video to reduce data size while preserving perceptual quality. The choice of codec directly affects bandwidth usage and error resilience. For instance, H.265/HEVC offers approximately 50% better compression efficiency than H.264 at the same quality level, while newer codecs like AV1 provide further gains and are being adopted by YouTube and Netflix. These techniques enable tradeoffs between video quality and resource usage, especially under constrained bandwidth conditions [77].

- **Adaptive Bitrate (ABR) Streaming:** Protocols such as MPEG-DASH and Apple HLS divide video content into multiple versions at varying resolutions and bitrates. The player dynamically selects the most suitable version based on real-time network conditions and device capability [76]. This mechanism ensures uninterrupted playback during bandwidth fluctuations. ABR streaming embodies the tradeoff between resolution and playback continuity.

- **Buffer Management:** Buffers are used to pre-load a portion of video content before playback to absorb jitter and bandwidth variation. The size and management of the

buffer depend on the type of service: Video On Demand Service (VOS) platforms like Netflix typically use larger buffers for stability, whereas real-time services like Twitch minimize buffer size to reduce latency. Thus, buffer sizing reflects the tradeoff between playback stability and responsiveness [78].

- **Error Control Mechanisms:** Forward Error Correction (FEC) adds redundancy that allows recovery from packet losses without retransmissions, used in latency-sensitive environments. Automatic Repeat Request (ARQ), by contrast, retransmits lost packets and is better suited for TCP-based applications where delay is less critical [79]. These mechanisms manage the tradeoff between added redundancy and stream continuity.

2.3.3.3 Network Provider-Side Adaptation

- **Congestion Control Algorithms:** Advanced congestion control algorithms like BBR (developed by Google) adjust sending rates based on estimated bandwidth and round-trip time [80]. These techniques outperform traditional TCP congestion control in maintaining lower delay and higher throughput, implementing tradeoffs between fairness and responsiveness in data delivery.
- **Traffic Prioritization and QoS Policies:** Network operators may prioritize certain types of traffic using mechanisms like Differentiated Services (DiffServ) or Deep Packet Inspection (DPI). Real-time traffic such as video conferencing may receive higher priority over bulk data transfers [81]. This reflects a tradeoff between equitable bandwidth distribution and meeting latency-sensitive service demands.
- **Edge Caching and CDNs Optimization:** CDNs and edge caching reduce latency by serving frequently accessed content from servers close to the user. Services like Netflix Open Connect and YouTube's CDNs infrastructure exemplify this strategy [79]. While edge caching consumes local storage, it significantly improves startup times and reduces backbone congestion, thus realizing the tradeoff between storage redundancy and delivery efficiency.

2.4 Multimedia Services in Mobile Networks

With the exponential growth of mobile multimedia services, the underlying network infrastructure must support diverse traffic types with varying sensitivity to delay, jitter, and bandwidth. This section examines how mobile networks, particularly LTE systems, are architected to meet

these demands. It outlines the core components of the LTE architecture, explains how QoS is provisioned through bearer services, and investigates the implications for multimedia traffic performance.

2.4.1 LTE: The 4G Standard for Mobile Multimedia

Wireless networks have evolved over the years to meet user demands for high-bandwidth services and mobility. LTE, the dominant 4G standard, addresses these demand by delivering major advancements over previous generations.

A key focus of LTE is user satisfaction, achieved through a combination of high data rates (up to 150 Mbps downlink and 50 Mbps uplink) and low latency (around 10 ms). However, the actual user experience depends on the available bandwidth and network configuration. As shown in Table 2.2, it is specifically designed with user satisfaction in mind, offering peak downlink speeds of up to 150 Mbps and uplink speeds of 50 Mbps, with typical latency as low as 10 ms. However, actual user experience is influenced by factors such as available bandwidth, device capability, and network configuration [82].

LTE addresses the challenge of combining high data throughput with low power consumption through a re-engineered physical layer that enables flexible modulation for both Downlink (DL) and Uplink (UL) communication. It also supports full-duplex operation, allowing simultaneous transmission and reception. Importantly, the system prioritizes downlink performance—leveraging the higher power of base stations to extend mobile device battery life [83].

Table 2.2: Comparison of Key Features between LTE and 3G Technologies[84]

Features	LTE	3G
Data rates (DL/UL)	Up to 150 Mbps / 50 Mbps	Up to 42 Mbps / 1.4 Mbps
Latency	10 ms	100 ms
Spectral efficiency	30% better	20% better
QoS Support	Multiple QoS classes	Single QoS classes

Compared to 3G systems [84], LTE, offers significantly improved data rate, redunced latency, and a greater spectral efficiency. These benefits are made possible through a set of advanced technologies, including:

- **Orthogonal Frequency Division Multiple Access (OFDMA):** This technique divides available bandwidth into multiple subcarriers, enabling parallel data streams for multiple users with differentiated QoS levels.

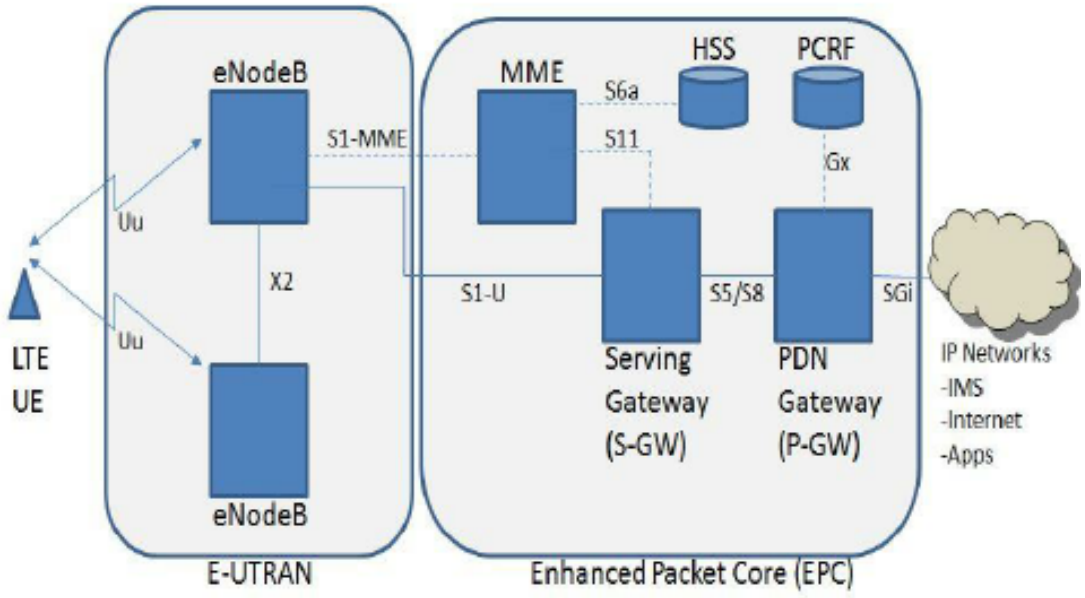


Figure 2.3: LTE Network Architecture [87].

- **Transmission Time Interval (TTI):** LTE utilizes a short 1 ms TTI allows the network to rapidly respond to changing channel conditions.
- **QoS classes:** LTE defines service classes that prioritize real-time applications (e.g., VoIP, video calls) while allowing more relaxed parameters for services like web browsing.
- **Frequency Division Duplex (FDD) and Time Division Duplex (TDD)**: These methods allow separation of uplink and downlink transmissions by FDD using distinct frequency bands and TDD alternating in time.

This overview establishes LTE as a robust foundation for mobile multimedia services, offering the speed, responsiveness, and flexibility needed to support data-intensive applications. The following sections explore the LTE network architecture and transmission modes in greater detail—laying the groundwork for analyzing how LTE's QoS mechanisms affect QoE, particularly in mobile video streaming.

2.4.1.1 LTE Network Architecture

The LTE network architecture [85] [86] forms the backbone of modern mobile communication, enabling fast, reliable, and scalable data transmission for a wide range of services. As shown in Figure 2.3, it adopts a layered architecture divided into two main domains: Evolved UMTS Terrestrial Radio Access Network (E-UTRAN) and Evolved Packet Core (EPC).

This separation allows specialized handling of radio access and core networking functions, enhancing performance and maintainability.

- **E-UTRAN:** Serves as the LTE Radio Access Network (RAN), acting as the interface between mobile devices and the core network. Key components within the E-UTRAN include:
  - **User Equipment (UE):** Devices such as smartphones and tablets equipped with LTE radios. They initiate communication with the network and are the endpoints of user data flow.
  - **Evolved Node B (eNodeB):** LTE base stations that handle direct communication with UEs. Each eNodeB manages radio resources (e.g., power, frequency, scheduling) and is responsible for mobility features such as handovers between cells to ensure continuous service during user movement.
- **EPC:** The EPC constitutes the IP-based core of LTE, managing user sessions, mobility, and policy enforcement.
  - **Mobility Management Entity (MME):** Controls signaling and mobility functions. It tracks UE location, manages session establishment, and facilitates handovers between eNodeBs.
  - **Serving Gateway (SGW):** Routes data packets between the UE and external networks. It applies packet fiLTEring, quality enforcement, and traffic prioritization based on network policies.
  - **Packet Data Network Gateway (PGW):** Acts as the gateway to the internet, performing IP address allocation and Network Address Translation (NAT) and supporting policy enforcement for service quality and charging.
  - **Home Subscriber Server (HSS):** A centralized database containing user subscription information, authentication credentials, and service profiles.
  - **Policy Control Function (PCRF):** Determines policy and charging rules. It instructs the PGW on how to treat different traffic types (e.g., real-time vs. non-real-time), ensuring QoS alignment and billing compliance.
- **Connectivity and Scalability:** The S1 interface connects E-UTRAN with the EPC, handling both control and user plane data. This modular design allows mobile operators to scale networks by deploying additional eNodeBs and upgrading EPC components independently. QoE is a central design consideration in LTE architecture. For example:

- **Handover Management** via MME ensures seamless service continuity during mobility.
- **Data Prioritization** via PCRF improves real-time application performance.
- **Address Translation and Session Control** via PGW enables uninterrupted access to external content.

In conclusion, understanding the distinct functionalities within the E-UTRAN and EPC, along with their interconnectedness through the S1 interface, provides a solid foundation for appreciating the complexities of managing a modern mobile network infrastructure. This knowledge will be instrumental in analyzing the relationship between network performance QoS and user experience QoE in the context of mobile multimedia services, which will be explored in subsequent sections.

2.4.2 Transport Channels and Transmission Modes in LTE

The LTE network architecture relies on a sophisticated interplay between transport channels and transmission modes to ensure efficient and reliable data transmission. These elements work in concert to carry user information and control signals within the network, catering to diverse service requirements and user equipment capabilities.

2.4.2.1 Transport Channels: Dedicated Pathways for Information Flow

Transport channels function as dedicated pathways for conveying information generated at higher network layers. Each channel is characterized by its specific coding scheme and information transportation method, enabling tailored data delivery based on the content type and network conditions. Here's a closer look at some key transport channels in LTE [88]:

- **Broadcast Channel (BCH):** This downlink channel serves as the foundation for network operation, transmitting essential system information for cell configuration and management. UEs within the cell leverage this information to identify the cell, understand the available system bandwidth, and synchronize their uplink and downlink frequencies. The BCH transmits in a fixed format, ensuring efficient decoding by all UEs within the coverage area.
- **Downlink Shared Channel (DL-SCH):** The DL-SCH is the workhorse for downlink data transmission. It carries a broad range of information, including user data (voice calls, video streams, web downloads), supplementary system information not covered by

the BCH (e.g., temporary cell identifiers), and paging messages used to notify idle UEs of incoming calls or messages. Data is segmented into Transport Block (TB) for transmission, with one TB generated per TTI (typically 1 ms). Depending on the chosen transmission mode and channel conditions, one or two TBs can be transmitted per sub-frame for each UE, enabling flexible bandwidth allocation based on user demands.

- **Paging Channel (PCH):** This downlink channel is dedicated to a specific task: transmitting paging messages to UEs. These brief messages alert UEs in the Radio Resource Control (RRC) IDLE state (powered on but not actively connected to the network) about incoming calls or data sessions, prompting them to transition to a connected state for communication.

- **Multicast Channel (MCH) (MCH):** The MCH caters to the specific needs of Multimedia Broadcast and Multicast Services (MBMS). This service allows for the efficient transmission of the same content (e.g., live sports broadcasts, traffic updates) to a large group of UEs simultaneously. The MCH operates only in designated subframes defined by the Multimedia Broadcast Single Frequency Network (MBSFN) standard, enabling efficient spectrum utilization for multicast services.

- **Uplink Shared Channel (UL-SCH):** The UL-SCH handles the critical task of transporting user data and control information from UEs to the eNodeB. This channel is utilized by all UEs in the network, regardless of their current state (active call, idle, etc.). UEs leverage the UL-SCH to transmit voice data during calls, upload files, or send control signaling to the network.

- **Random-Access Channel (RACH):** When a mobile device needs to establish network access, it transmits a random-access message on the RACH, which is an uplink channel. This message essentially serves as a "handshake" with the network, requesting resources for data transmission. The eNodeB acknowledges this message by allocating resources (uplink bandwidth and subframes) to the UE, enabling it to initiate communication. The mapping between these transport channels and their corresponding physical channels is illustrated in Figure 2.4. This mapping defines how the logical transport channels are translated into physical signals for transmission over the air interface.

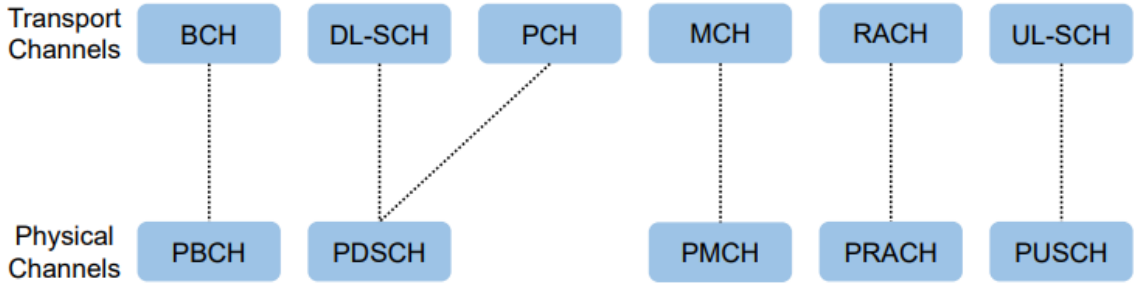


Figure 2.4: Mapping of the transport channels onto the physical channels in LTE [86].

2.4.2.2 Transmission Modes: Leveraging Multiple Antennas for Enhanced Performance

The transmission mode in LTE determines the multi-antenna configuration used for communication between the eNodeB and the UE. These configurations can be categorized into four types based on the number of antennas employed on both the transmitting and receiving ends [89]:

- **Multiple-Input Multiple-Output (MIMO)**: In this advanced configuration, both the transmission eNodeB and reception UE sides use multiple antennas. MIMO allows for the simultaneous transmission of independent data streams, significantly increasing the system's data capacity without requiring additional bandwidth. LTE commonly employs 2x2 or 4x4 MIMO setups, which improve both throughput and spectral efficiency. Furthermore, MIMO enhances signal robustness and reduces the impact of fading, which is particularly important for maintaining high QoE in bandwidth-intensive multimedia applications like HD video streaming.
- **Multiple-Input Single-Output (MISO)**: This configuration utilizes multiple antennas for transmission (eNodeB), while the receiving side (UE) has only one. MISO can improve transmission robustness against fading by exploiting spatial diversity at the transmit side. By strategically transmitting the same signal from multiple antennas with slight variations, the eNodeB can mitigate the effects of multipath fading and enhance signal reliability.
- **Single-Input Multiple-Output (SIMO)**: Here, the transmission side (eNodeB) employs a single antenna, while the receiving side (UE) benefits from multiple antennas. This setup can enhance reception through spatial diversity. By leveraging multiple receive

antennas, the UE can exploit variations in the signal caused by multipath fading to improve signal quality and potentially increase data throughput.

- **Single-Input Single-Output (SISO):** This basic configuration utilizes only one antenna for both transmission (eNodeB) and reception (UE). While functional, SISO offers limited potential for improving signal quality or data rates.

2.4.2.3 Key Constraints in LTE Multimedia Delivery

Despite its advances, LTE faces several technical and operational constraints that challenge its ability to deliver consistent, high-quality multimedia services. These limitations arise from the nature of wireless transmission, limited spectral resources, and the need to support user mobility across diverse network conditions [90][66].

- **Limited and Shared Bandwidth:** LTE relies on shared radio spectrum. In high-demand scenarios such as peak hours or urban areas, congestion leads to degraded throughput, affecting high-bitrate streaming and increasing startup delay or rebuffering risk.
- **User Mobility and Handover Delays:** Seamless service delivery is challenged when users move across LTE cells. Handover procedures, especially under high-speed mobility (e.g., trains or vehicles), introduce latency, jitter, or even packet loss, disrupting real-time applications.
- **Dynamic Channel Conditions:** Wireless links are susceptible to fading, shadowing, and interference. These variations can degrade link quality, resulting in video quality fluctuation, buffering, and unpredictable playback performance.
- **Network Congestion:** Increased user density and multimedia demand reduce per-user resource availability [91]. Congestion affects real-time services the most, where maintaining low latency and consistent throughput is critical.
- **Bandwidth Constraints:** Delivering high-resolution multimedia content under constrained spectrum availability necessitates efficient compression (e.g., via H.265) [92] and adaptive streaming strategies.

2.4.2.4 Technical Enablers and Optimization Opportunities

To address these constraints, LTE incorporates a range of protocol-level and physical-layer enhancements aimed at improving multimedia service delivery. These technical enablers help

optimize resource usage, enhance robustness, and adapt media quality dynamically in response to real-time network conditions [93].

- **ABR**: LTE enables ABR techniques such as MPEG-DASH and HLS allowing media clients to dynamically adjust stream quality based on real-time bandwidth and buffer status. While this enhances continuity, it can also cause noticeable resolution shifts.
- **QoS-Aware Scheduling:** LTE supports scheduling based on QCI, enabling the network to prioritize multimedia traffic with specific delay and loss sensitivity—crucial for VoIP and live video applications.
- **Support for MIMO and Carrier Aggregation:** These physical-layer enhancements boost spectral efficiency and aggregate available bandwidth, supporting the demands of bandwidth-intensive applications like 4K streaming or multi-user scenarios [94].
- **Dynamic Resource Allocation:** The eNodeB can allocate Physical Resource Blocks (Physical Resource Block (PRB)s) adaptively in response to user demands and service type, optimizing real-time traffic flow across variable radio conditions.
- **Enhanced Multimedia Broadcast Multicast Service (eMBMS):** eMBMS enables efficient content distribution to multiple users simultaneously, ideal for live events and group-based media services where multicast transmission reduces network load [95].

2.5 QoS Fundamentals and Models in LTE Networks

QoS is a foundational concept in mobile networking, directly influencing how multimedia services are delivered and experienced by end-users. In LTE networks, QoS frameworks are designed to maintain service-level performance even under dynamic traffic loads and varying radio conditions.

To support services such as video streaming, VoIP, and real-time conferencing, LTE specifies distinct QoS attributes with quantitative benchmarks [96]:

- **Latency:** End-to-end delay should be below 150 ms for voice services and under 400 ms for video conferencing.
- **Jitter:** Variations in delay must remain within 30 ms to preserve video and audio synchrony.

- **Throughput:** Video streaming typically requires 2–5 Mbps; HD or 4K may demand more.
- **Packet Loss:** For perceptual quality, packet loss must be below 1% for both voice and video streams.

2.5.1 ITU and LTE-Specific QoS Class Identifiers

To implement these requirements, LTE defines standardized QCIs. Each QCI is associated with a specific traffic type, defining parameters such as delay tolerance, priority, and acceptable loss rate. Table 2.3 outlines representative QCI configurations for multimedia delivery.

Table 2.3: Example QCI in LTE [97].

QCI	Priority Level	Packet Delay Budget (ms)	Packet Loss Rate
1	2	100	10^{-2} (Conversational Voice)
2	4	150	10^{-3} (Live Streaming Video)
6	6	300	10^{-6} (Buffered Video)
9	9	300	10^{-6} (Best Effort Services)

These identifiers allow LTE networks to map application flows to appropriate bearers, enabling predictable QoS behavior across varying radio and traffic conditions.

2.5.2 QoS Models in Network Design

In networking, QoS models define how traffic is treated and prioritized. While not exclusive to LTE, these models [98] [5] influence how mobile and IP core networks allocate resources:

- **Best-Effort:** No guarantees on throughput or latency. All packets are treated equally, suitable for non-critical applications.
- **Integrated Services (IntServ):** Based on per-flow resource reservation via Resource Reservation Protocol (RSVP), this model supports guaranteed service but lacks scalability for large networks.
- **Differentiated Services (Diffserv):** Uses Differentiated Services Code Point (DSCP) field markings in the IP header to classify packets into different QoS classes, enabling scalable service differentiation.

In LTE, a hybrid approach is used. Within the radio access and core network, QCI based bearers manage traffic classification and prioritization. For IP-level transport over the backhaul and internet, DiffServ may be applied for end-to-end service coherence.

2.5.3 Rate Adaptation Support in LTE Architecture

To cope with the variability of wireless conditions, LTE supports multiple rate adaptation techniques. These techniques have been discussed earlier under *Bitrate Control and Delivery Modalities*, but are briefly revisited here to show their technical implementation within LTE:

- **CBR** and **VBR** traffic can be transported using dedicated or default bearers, depending on QCI allocation and service criticality [99].
- **ABR** streaming is extensively supported in LTE, allowing client applications to select segment qualities based on real-time throughput estimates and buffer states.
- **Dynamic scheduling** at the eNodeB allocates PRBs based on radio quality and QoS constraints, enabling real-time adaptation to channel conditions [100].

While ABR mechanisms are implemented on the client side, their effectiveness relies on LTE's ability to sustain throughput levels and low jitter within the allocated bearer. Integration of ABR with QCI ensures a tight coupling between application-layer adaptation and network-layer resource control.

2.6 QoS in Practice: LTE Network-Centric Perspective

LTE employs a structured QoS framework that distinguishes traffic based on service requirements, user subscription levels, and network policy enforcement mechanisms. The cornerstone of service differentiation in LTE is the use of QCIs, which define key transmission characteristics such as packet delay budget and packet loss rate. Each bearer is associated with a specific QCI, allowing the network to assign differentiated handling and prioritization strategies [101][102]. For example:

- **QCI 1** is designed for conversational voice, with a packet delay budget of 100 ms and a packet error loss rate of 10^{-2}.
- **QCI 2** targets real-time video, with higher bandwidth and a slightly relaxed delay budget (150 ms).

- **QCI 6–9** are used for best-effort services like web browsing or file downloads, where delay is less critical.

These QoS levels are enforced using mechanisms such as Allocation and Retention Priority (ARP), Traffic Flow Templates (TFT) , and scheduling algorithms at the eNodeB level. LTE's architecture supports both Guaranteed Bit Rate (GBR)and Non-GBR bearers, enabling flexible resource allocation depending on service sensitivity and network load.

2.7 QoS Monitoring and Resource Management in LTE

Effective QoS assurance in LTE is not merely a matter of initial service configuration—it requires continuous monitoring and real-time adjustments. This section outlines the key mechanisms LTE networks use to monitor and manage resources for optimal service delivery.

2.7.1 Scheduler Design and Traffic Prioritization

LTE employs dynamic scheduling at the eNodeB level to allocate radio resources based on current channel conditions, service requirements, and QCI. Scheduling algorithms such as Proportional Fair (PF), Maximum Throughput (MT), and Reduced-Reference (RR) are implemented to balance efficiency and fairness[103].

- **PF:** Balances throughput and fairness by scheduling users with good channel conditions while ensuring minimum allocation for others.
- **RR:** Provides equal time slots to all users regardless of channel condition or service requirement.
- **MT:** Focuses on maximizing overall cell throughput but may starve users with poor channel quality .

2.7.2 Traffic Classes and QoS-Aware Resource Allocation

Effective QoS enforcement in LTE relies on the classification of traffic into predefined categories based on delay tolerance, loss sensitivity, and interaction requirements. This classification enables the network to dynamically allocate radio and core network resources according to application-specific performance needs, ensuring a consistent user experience across service types [104][97]. The main traffic classes include:

- **Conversational Traffic:** Characterized by strict latency and jitter requirements, this class includes services like VoIP and video conferencing where real-time interaction is essential.
- **Streaming Traffic:** Used for services such as live video and audio streaming, this traffic type tolerates minor jitter but remains sensitive to end-to-end delay and packet loss.
- **Interactive Traffic:** Found in applications like web browsing or online gaming, it requires moderately low delays to maintain responsiveness but is less affected by occasional jitter or loss.
- **Background Traffic:** Includes tasks such as email sync, file downloads, and software updates, which are delay-tolerant and can be deprioritized without impacting user-perceived quality.

Resource allocation mechanisms in LTE, including scheduling algorithms and bearer prioritization, leverage these classifications to ensure that delay-sensitive services receive preferential treatment under congestion or limited bandwidth conditions.

2.7.3 Admission Control and Load Balancing

Admission control mechanisms evaluate incoming traffic against available capacity to prevent overloading and ensure QoS compliance. Load balancing techniques redistribute traffic across cells or carriers to alleviate congestion.

2.7.4 Performance Monitoring and KPI Tracking

In LTE networks, continuous monitoring of KPIs is critical to evaluating service quality, diagnosing network issues, and ensuring that QoS targets are met. These KPIs serve as quantifiable metrics that guide resource allocation, traffic shaping, and policy enforcement across the radio access and core network layers [105] [106]. The most commonly tracked KPIs include:

- **Packet Loss Rate:** Measures the proportion of data packets that fail to reach their destination, directly impacting multimedia continuity and user experience.
- **Throughput per User:** Represents the average data rate successfully delivered to an individual user, reflecting both network capacity and fairness in resource distribution.

- **Block error rate (BLER):** Indicates the ratio of erroneous transport blocks at the physical layer; high BLER values can trigger retransmissions, increasing latency and degrading QoS [107].

- **Scheduling Delay:** Captures the time interval between a data request and its actual transmission, offering insight into resource allocation efficiency and congestion levels.

Tracking these indicators in real time enables LTE operators to proactively adjust scheduling algorithms, balance load across cells, and optimize user experiences under varying traffic conditions.

2.8 QoE in Multimedia Services

2.8.1 Definition and Dimensions of QoE

QoE, as defined by the ITU-T Recommendation P.10/G.100, is "*the overall acceptability of an application or service, as perceived subjectively by the end user*"[108]. This acceptability results from a complex interplay of technical performance, user expectations, and contextual factors. Unlike QoS, which focuses on network performance indicators, QoE centers on the end-user's perception of the service quality.

QoE can be broadly categorized into two dimensions:

- **Subjective QoE:** This involves human perception, feelings, expectations, and preferences, often assessed using user feedback tools like Mean Opinion Score (MOS)[109].

- **Objective QoE:** Involves algorithmic or model-based estimations that infer user experience based on quantifiable metrics such as bitrate, delay, jitter, and resolution [110].

2.8.2 Influencing Factors in Mobile QoE

QoE in mobile multimedia services is influenced by a wide range of variables. These can be grouped into three main categories:

- **User-centric Factors:** Device type and screen size, individual preferences, prior experiences, and mobility patterns significantly shape user expectations and perceived service quality [111].

- **Network Factors:** Key parameters include end-to-end delay, jitter, packet loss, and available bandwidth. Poor connectivity often leads to buffering, interruptions, or resolution downgrades [112].
- **Application and Contextual Factors:** These involve the codec efficiency, service content type, user location, and time of day. For example, a user streaming HD content during peak hours may have a different QoE than during off-peak hours due to congestion [113].

2.8.3 Measurement and Evaluation Approaches

QoE assessment can follow either subjective or objective methodologies:

- **Subjective Methods:** These include techniques like [114]:
  - **Mean Opinion Score (MOS):** A widely used metric where users rate the perceived quality on a scale (typically 1 to 5).
  - **Degradation Category Rating (DCR):** A comparison-based technique assessing how much the quality is degraded.
- **Objective Methods:** These include [115]:
  - **PSNR:** Measures signal fidelity compared to a reference.
  - **Video Multi-dimensional Assessment Tool (VMAF):** A machine-learning-based model that correlates strongly with human perception.
  - **Machine Learning Models:** These include regression models, SVMs, and deep neural networks that map QoS parameters to QoE outcomes.

2.8.4 Importance and Role of QoE in Service Optimization

QoE serves as a strategic metric for service providers aiming to improve customer retention and business outcomes [116]. Its importance lies in:

- **Business Value:** High QoE enhances user satisfaction, increases platform engagement, and reduces churn. This directly correlates with higher revenue and market competitiveness.

- **Integration with Network Planning:** QoE metrics support informed decisions in network provisioning, congestion control, and resource allocation. For instance, ABR streaming mechanisms leverage real-time QoE feedback to adjust media quality dynamically.
- **Competitive Advantage:** Providers offering superior QoE can differentiate themselves, especially in saturated markets where technical parameters alone are insufficient to gain user loyalty.

2.9 QoE for Multimedia Services

Unlike traditional quality measures solely focused on technical parameters like network bandwidth and response times, QoE takes a more comprehensive approach. It encompasses both objective and subjective factors that influence how users perceive a service or application. This broader perspective is crucial for delivering multimedia services that are not only technically sound but also meet user expectations and provide a satisfying experience.

2.9.1 Subjective and Objective QoE

Traditional quality measures focused solely on technical parameters like network bandwidth and response times. However, QoE takes a more comprehensive approach, encompassing both objective and subjective factors that influence how users perceive a service or application [114]. This broader perspective is crucial for delivering multimedia services that are not only technically sound but also meet user expectations and provide a satisfying experience.

2.9.1.1 Objective QoE

Objective QoE provides quantifiable insights into system performance through measurable technical parameters [115]. These metrics act as a foundation for network engineers and service providers to identify areas for improvement and ensure a smooth user experience. Here are some key objective QoE metrics:

- **Network Performance:** Measured parameters include latency (data transmission delay), jitter (variation in delay), and packet loss rate (percentage of data packets not reaching their destination). Consistent and predictable network performance is crucial for services like real-time video calls or online gaming, where even slight delays or disruptions can significantly impact QoE.

- **Application Performance:** This focuses on factors like response time (time taken for the application to respond to user input), uptime (percentage of time the application is available), and error rates (frequency of application crashes or errors). Reliable application performance ensures users can interact seamlessly with the service without encountering crashes or freezing issues.

While objective QoE factors are primarily quantitative, some user limitations, like color blindness, might introduce a qualitative element that can be quantified and considered within objective QoE metrics. This highlights the interconnectedness between objective and subjective QoE aspects. For example, changing human biological or cognitive parameters (e.g., temporary hearing loss due to loud music) can influence subjective perceptions and feelings about service quality. However, the primary focus of objective QoE remains on measurable technical parameters that directly impact system performance.

2.9.1.2 Subjective QoE

Subjective QoE delves deeper into the user's individual perception and feelings during service usage. Unlike objective factors, subjective aspects are intangible and vary greatly depending on the user's unique experiences and expectations. Understanding these subjective elements is essential for creating a user-centric service that fosters positive user experience [117]:

- **Expectations:** Prior beliefs about a service's capabilities shape user perception. A user expecting high-definition video streaming with minimal buffering will be disappointed if they encounter frequent interruptions. Setting realistic expectations through clear communication is vital for managing user perception. Service providers can utilize onboarding tutorials or informative product descriptions to manage user expectations and ensure alignment with actual service capabilities.

- **Prior Experience:** Past interactions with similar services influence user expectations. A history of positive experiences fosters trust and increases the likelihood of a good QoE in the present. Conversely, negative past experiences can lead to lower expectations and reduced satisfaction with the current service. Service providers can leverage positive user reviews and testimonials to build trust and address potential concerns stemming from past negative experiences with similar services.

- **Feelings and Thoughts:** Emotional responses during usage significantly impact QoE. Frustration with slow loading times or excitement during a high-quality video call

directly affect user perception. Service design should aim to minimize frustration and maximize positive emotions for a more enjoyable user experience. Techniques like progress indicators during loading times or visual feedback for successful actions can help manage user emotions and enhance QoE.

- **Context of Use:** The environment where the service is used plays a crucial role. For instance, a weak cellular signal in a remote location might hinder QoE for a video call, while a user on a stable Wi-Fi connection at home might experience a different level of satisfaction. Understanding the context of use allows for tailoring services to specific situations and user needs. Adaptive bitrate streaming for video services can adjust video quality based on available bandwidth, ensuring a smooth experience even on weaker connections.

By considering both objective and subjective aspects of QoE, service providers can gain a more holistic understanding of user experience. This allows them to identify areas for improvement across network infrastructure, application performance, and user experience design. Optimizing these factors while addressing user expectations and emotional responses is key to delivering high-quality QoE for multimedia services, ensuring user satisfaction and loyalty in a competitive market.

2.9.2 Factors Influencing QoE in Mobile Multimedia Services

While traditionally, research focused primarily on the impact of network and application performance (objective factors) on QoE, recent studies emphasize the growing importance of user-centric considerations. Here's a breakdown of the key factors influencing QoE in mobile multimedia services:

2.9.2.1 User-centric Factors

- **Internal Factors:** These encompass both subjective and objective aspects that influence a user's perception of service quality [118][119]:
  - **Subjective:** User expectations and preferences play a significant role. A user expecting seamless video streaming will be more frustrated by buffering compared to someone with lower expectations.
  - **Objective:** Cognitive factors like workload and attention levels also influence QoE. A user multitasking on their phone might be less sensitive to minor delays compared to someone focused solely on a video call.

- **External Factors Shaping QoE in Mobile Multimedia Services:** Beyond user expectations and cognitive factors, the broader context in which a service is used plays a significant role in shaping QoE. Here's a deeper dive into the key external factors influencing user perception:
  - **Technological Domain: The Foundation of Service Delivery**

    The technological domain encompasses the technical infrastructure that underpins service delivery. Here, various factors influence user experience:

    * **Network Performance:** The underlying network acts as the backbone for multimedia services. Factors like bandwidth, latency (data transmission delay), jitter (variation in delay), and packet loss rate (percentage of data packets not reaching their destination) directly impact user experience. Consistent and reliable network performance is essential for smooth streaming, uninterrupted video calls, and other real-time applications. Inconsistent bandwidth or frequent packet loss can lead to buffering, lag, and dropped calls, significantly hindering QoE [29].
    * **Device Capabilities:** The user's device acts as the interface for interacting with the service. Processing power, memory, screen resolution, and operating system all play a role. A lower-end device might struggle with high-definition video playback, complex applications, or resource-intensive tasks [120]. This can lead to stuttering playback, slow loading times, or application crashes, negatively impacting QoE.
    * **Software Design:** The design and functionality of the application or service itself come into play [59]. An intuitive user interface with clear navigation and optimized features for mobile devices contributes to a positive QoE. Efficient coding practices that minimize resource consumption and ensure smooth performance are also crucial. A poorly designed interface, buggy software, or features not optimized for mobile usage can lead to frustration and a negative user experience.

  - **Business Domain: Balancing User Needs and Service Offerings**

    Business practices and service offerings can shape user perception and impact QoE in several ways [121]:

    * **Pricing and Billing:** The cost of the service and data plans influence user expectations and satisfaction. A user on a limited data plan might be more

sensitive to buffering compared to someone with an unlimited plan. Transparent pricing models and clear communication about data usage limitations help manage expectations and avoid user frustration.

* **Marketing and Communication:** Clear and transparent communication about service capabilities and limitations is essential. Effective marketing can build trust and foster a positive brand image by setting realistic expectations. Overpromising features or failing to communicate potential limitations can lead to user disappointment and negatively impact QoE [122].
* **Customer Support:** Providing timely and helpful customer support is crucial for addressing user concerns and resolving issues that can negatively impact QoE. Responsive and knowledgeable support staff can help users troubleshoot problems, answer questions, and ultimately improve their overall experience with the service.

– **Contextual Domain: The Environment Matters**

The environment in which the service is used plays a significant role in shaping QoE [123]:

* **Location:** User location can significantly affect network performance. A user in a remote area with a weak cellular signal will experience a different QoE for a mobile video call compared to someone in an urban center with a strong signal. Service providers can consider offering network optimization features or recommending specific locations for better signal quality to enhance user experience in varying locations.
* **Time of Day:** Network congestion can vary depending on the time of day and user location. During peak usage hours, users might experience slower speeds and more buffering, impacting QoE. Service providers can utilize dynamic allocation of resources or implement strategies to manage peak loads, helping to ensure a consistent and positive user experience [124].
* **Social Situation:** The social context in which the service is used can influence expectations and perception. A user on a work video call might have different expectations for responsiveness and quality compared to someone on a casual video call with friends. Understanding these social contexts can help service providers tailor their offerings to meet user needs and optimize QoE for different scenarios.

By understanding these external factors in conjunction with user-centric considerations, service providers can create a more holistic picture of QoE. Addressing the technical infrastructure, business practices, and contextual elements that influence user experience allows for a more optimized and user-friendly service. This comprehensive approach ultimately contributes to increased customer satisfaction, loyalty, and a competitive advantage in the mobile multimedia services market.

2.9.3 QoE Measurement Approaches: Balancing Accuracy and Efficiency

QoE Measurement plays a crucial role in ensuring user satisfaction with multimedia services, particularly video applications. However, the ideal measurement approach strikes a balance between accuracy and efficiency. This section delves into the two main categories of QoE measurement methods: subjective and objective.

2.9.3.1 Subjective Methods: High Accuracy , High Cost

Subjective methods provide the most accurate assessment of QoE by directly capturing user perception. Standardized protocols like International Telecommunication Union Radiocommunication (ITU-R) BT.500-13 [125] and ITU-T Rec.P.910 [126] guide the conduct of subjective video quality measurements, ensuring controlled test environments, appropriate observer selection, and standardized assessment procedures. The Mean Opinion Score (MOS), a numerical rating from 1 (poor) to 5 (excellent), serves as the primary metric for subjective evaluation. While highly accurate, subjective methods present significant drawbacks:

- **Cost and Time:** Maintaining a controlled testing environment and recruiting participants can be expensive and time-consuming.
- **Scalability:** Subjective methods are difficult to automate and implement in real-time, limiting their scalability for large-scale monitoring.

2.9.3.2 Objective Methods: Efficiency for Real-world Application

Objective methods offer a faster and more cost-effective alternative to subjective assessments [127] [128]. These methods rely on mathematical models or comparative techniques to quantify video quality. Their key advantages include:

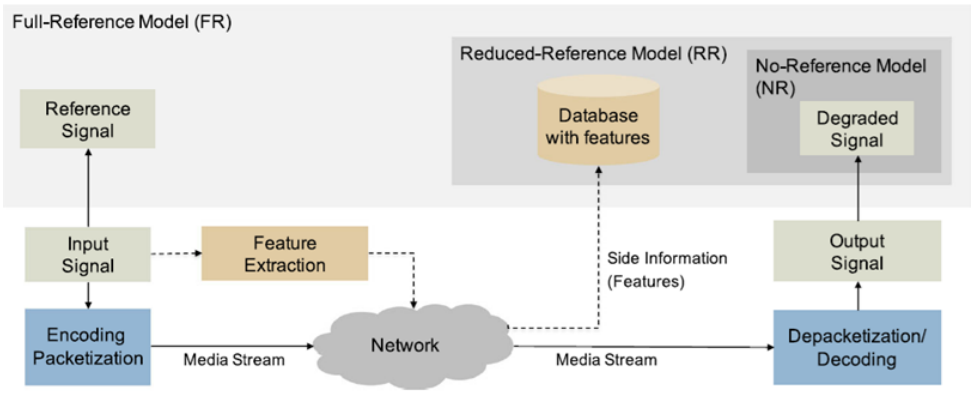


Figure 2.5: Classification of objective video quality models [129].

- **Automation:** Objective methods can be automated, enabling real-time implementation for network monitoring and service optimization.
- **Scalability:** The automated nature of objective methods makes them suitable for large-scale deployments.
- **Real-World Application:** Objective methods can be applied to in-service quality monitoring, network and terminal design, codec optimization, and selection.

However, objective methods often sacrifice some accuracy for efficiency. They aim to predict the subjective MOS that users would give but may not always perfectly capture user perception.

2.9.3.3 Objective Method Categories: Tailoring the Approach

Objective video quality measurement methods can be further classified into three categories based on their reference requirements as depicted in Figure 2.5.

- **Full-Reference (FR):** FR methods require access to the original, undistorted video signal. They perform pixel-by-pixel comparisons to quantify distortions, making them impractical for real-time applications where the original video might not be available. Examples of FR methods include Video Quality Metric (VQM), Structural Similarity Index Measurement (SSIM), and PSNR [130].
- **RR:** RR methods extract features from the original video for quality assessment. This allows for online monitoring but requires a side channel to transmit the extracted features [131].

- **No-Reference (NR):** NR methods estimate video quality without needing the original signal. This makes them ideal for monitoring live video traffic but can be less accurate compared to FR and RR methods [132].

Understanding the trade-offs between subjective and objective methods allows service providers and researchers to choose the most appropriate approach for their specific needs. While subjective methods provide the gold standard for accuracy, objective methods offer valuable tools for real-time monitoring, scalability, and cost-effectiveness. As research continues, the development of more accurate and efficient objective methods will further enhance QoE management for multimedia services.

2.9.3.4 Importance of QoE in Multimedia Services

The landscape of multimedia services has evolved significantly. While once solely concerned with technical parameters like network bandwidth and response times, a more holistic approach is now paramount. This is where QoE enters the scene as a critical factor directly impacting user satisfaction, loyalty, and ultimately, the success of multimedia services.

- **Beyond the Technical: A user-centric approach Multimedia**

  QoE transcends the realm of simply ensuring technical functionality. It embraces a user-centric perspective, encompassing both objective (measurable technical aspects) and subjective (user perception and feelings) elements that influence how users perceive a service. This broader understanding acknowledges that even technically sound services can fall short if the user experience falls flat. Here's why QoE plays a vital role in the success of multimedia services:

  - **Elevating User Satisfaction:** High QoE translates to a smooth, seamless user experience free from disruptions and frustrations. Users are more likely to be satisfied and engaged with a service that meets their expectations and delivers a positive experience. Imagine a video call where the picture quality is crisp, the audio is clear, and there's minimal lag. This fosters a sense of connection and satisfaction, encouraging continued use. Conversely, a call plagued by constant buffering, choppy video, and audio dropouts creates frustration and can quickly sour the user experience [133].
  - **Fostering Customer Loyalty:** Positive QoE experiences cultivate user loyalty. When users perceive a service as reliable, enjoyable, and fulfilling their needs, they're more likely to continue using it and potentially recommend it to others.

This loyalty translates to a stable user base and a competitive edge in the multimedia market. Consider a video streaming platform that consistently delivers high-quality content with minimal buffering and a user-friendly interface [134]. Users are likely to subscribe and become loyal customers, appreciating the smooth and enjoyable experience.

- **Combating Churn:** Poor QoE experiences are detrimental to user retention. Frustration with lagging video calls, dropped connections, or unreliable service can lead to churn (cancellation of service). By prioritizing QoE and addressing potential issues, service providers can minimize churn and retain their user base. Imagine a video conferencing platform plagued with frequent connection drops during important meetings. Users are likely to become frustrated and seek out a more reliable LTE, resulting in churn [135].

- **Differentiation in a Crowded Market:** Today's multimedia market is fiercely competitive [136]. QoE can serve as a powerful differentiator, allowing service providers to stand out from the crowd. By demonstrably delivering a superior user experience, providers can attract new customers and solidify their position. Consider two music streaming platforms: one offering high-fidelity audio, seamless playback, and personalized recommendations; the other experiencing frequent buffering and lacking intuitive features. Users are more likely to choose the platform that delivers a superior and enjoyable listening experience.

- **Data-Driven Service Improvement:** Understanding user experience through QoE measurement provides invaluable data for service providers [137]. This information allows them to identify areas for improvement, optimize resource allocation, and make informed decisions regarding service development. By analyzing user feedback and objective QoE metrics, providers can identify bottlenecks, improve service reliability, and tailor offerings to better meet user needs and expectations.

In today's multimedia landscape, focusing solely on technical specifications is a recipe for diminishing returns. By prioritizing QoE and taking a user-centric approach, service providers can deliver a satisfying and engaging experience that translates to user satisfaction, loyalty, and ultimately, the success of their multimedia offerings. By fostering positive QoE, multimedia services can not only survive but thrive in a competitive marketplace.

2.9.3.5 User Experience Considerations in LTE Video Delivery

From a QoE perspective, LTE video delivery must address both technical and perceptual dimensions [138][139][140][141][142]:

- **Startup Delay and Buffering:** Users expect minimal wait time before playback starts. Initial buffering and mid-playback stalls are among the most detrimental events to perceived video quality.
- **Resolution Consistency:** Frequent shifts in video quality due to ABR are often perceived negatively, even if playback remains uninterrupted. Users prefer consistent medium-quality playback over fluctuating high-quality streams.
- **Latency in Live and Interactive Services:** Delays beyond a few seconds in live content or milliseconds in RTC degrade user engagement and interactivity. LTE's lower round-trip times compared to 3G improve this but may still fall short for time-critical applications.
- **Device and Screen Size Considerations:** The impact of quality fluctuations may be more noticeable on larger or high-resolution displays. LTE must support efficient content scaling for various device types.
- **Context Awareness:** Factors such as user location, time of day, and content type (e.g., sports vs. news) influence QoE expectations. LTE-based streaming systems benefit from context-aware adaptation to optimize delivery.

2.10 Related Work in QoE/QoS Mapping for Mobile Video Streaming

Optimizing the user experience (QoE) of mobile video streaming remains a significant challenge due to the dynamic nature of wireless networks and the ever-increasing demands of high-resolution content. To address this challenge, researchers have explored various techniques for mapping between QoS, which focuses on technical network parameters, and QoE, which captures the user's subjective perception of the service. This section delves into the existing body of research on QoE/QoS mapping for mobile video streaming, highlighting both traditional techniques and the growing prominence of machine learning approaches.

2.10.1 Traditional Techniques and Limitations

Early efforts in QoE/QoS mapping for mobile video streaming employed traditional techniques like linear regression and simple machine learning algorithms. These methods attempt to establish a direct relationship between readily measurable QoS parameters (e.g., bandwidth, latency, packet loss) and perceived QoE metrics (e.g., stalling rate, rebuffering events, video quality). However, research has consistently shown that raw QoS measurements and their associated KPI are not always reliable predictors of QoE.

- **Linear Regression:** This statistical method establishes a linear relationship between QoS parameters and predicted QoE values. Although it offers a baseline approach, linear regression can be limited in its ability to capture complex nonlinear relationships between variables often observed in QoE/QoS mapping, particularly with mobile video streaming, where network conditions can fluctuate rapidly [143].

- **Simple Machine Learning Models:** Techniques such as decision trees and k-Nearest Neighbor (KNN) have also been explored. Decision trees classify QoE based on a series of decision rules derived from QoS parameters, while KNN predicts QoE by identifying similar past instances based on QoS metrics [144]. However, these methods can be susceptible to overfitting and may not generalize well to unseen data, especially in the context of dynamic mobile video streaming environments.

While traditional techniques provide a starting point, the limitations in capturing the intricacies of user experience necessitate more sophisticated approaches.

2.10.2 Machine learning Approaches for QoE/QoS Mapping

With the growing availability of user data and computational power, Machine Learning (ML) has emerged as a powerful tool for QoE/QoS mapping. These techniques aim to learn the complex relationships between QoS parameters and user perception, providing more accurate and dynamic QoE predictions specific to mobile video streaming. Some prominent examples include:

- **Support Vector Machines (SVM):** SVMs can effectively handle nonlinear relationships and provide robust predictions, demonstrating promising results in QoE/QoS mapping for mobile video streaming [145]. For example, the research by [145] employed SVM to map network QoS metrics from real UMTS and LTE data to user-perceived data QoE (measured by MOS) with high precision. This approach has the potential to improve

resource allocation and service optimization for mobile video streaming providers by providing a more realistic measure of user satisfaction in a dynamic mobile network environment.

- **Neural Networks (NNs):** Deep learning architectures, particularly Convolutional Neural Networking (CNNs) and Recurrent Neural Networks (RNNs), have shown significant potential for QoE/QoS mapping in mobile video streaming [146]. CNNs excel at handling spatial relationships within QoS data (e.g., identifying patterns in network congestion data), while RNNs can capture temporal dependencies (e.g., understanding how past network fluctuations might impact the future QoE). An example by [147] explored the use of a multi-layer neural network to predict QoE for IPTV viewing specifically focusing on mobile video streaming parameters. Their model achieved better accuracy than simpler approaches, highlighting the potential of NN for mobile video streaming QoE/QoS mapping. However, it's important to acknowledge that NN can be computationally expensive to train and require substantial amounts of labeled data, which can be a challenge to obtain in real-world mobile video streaming scenarios.

2.10.3 Knowledge Gaps and Opportunities for Improvement

While the field of QoE/QoS mapping for mobile video streaming has seen significant progress, there are still knowledge gaps and opportunities for improvement. Some key areas to consider include:

- **Incorporating Contextual Information:** Current models often focus primarily on network-centric QoS parameters. However, the user context (for example, device capabilities, network type, and application usage patterns) can significantly impact QoE in mobile video streaming. Integrating contextual information into mapping models opens avenues for more holistic and user-centric QoE predictions [16].
- **Addressing Data Scarcity and Bias:** Training effective ML models often requires large amounts of labeled QoE data, which can be challenging to obtain for mobile video streaming services. Techniques for data augmentation and transfer learning can help mitigate this issue [148]. Additionally, ensuring unbiased datasets is important.
- **Challenges for Internet Service Providers (ISP):** growing challenges in monitoring QoE due to rapidly evolving network architectures and shifting user expectations. A significant obstacle is the increasing use of application layer encryption by OTT services

like YouTube and Netflix, restricting ISP access to detailed traffic data that traditional QoE monitoring systems depend on. This requires a shift toward new measurement methods and a fundamental redesign of monitoring infrastructures. Furthermore, the adoption of Software-Defined Networks and Network Function Virtualization introducs of both opportunities and complexities. These technologies enable scalable and adaptable quality models suitable for dynamic application environments but also increase architectural abstraction. Moreover, the emergence of the Internet of Things brings new use cases for quality monitoring through virtualized sensing devices. Finally, the deployment of modern protocols such as HTTP/2 and Quick UDP Internet Connections disrupts conventional traffic modeling approaches, especially those related to video streaming and web performance, necessitating novel strategies for accurate QoE prediction.

- **Comprehensive Measurement Tools:** There is a recognized need for tools that provide multifaceted insights into mobile video streaming, beyond just network traffic. Mobile Video Information Extraction is an open-source, cross-platform measurement tool that provides multifaceted insights into mobile video streaming. It captures both network traffic and video player activities, offering a more comprehensive analysis than traditional network-only tools. It generates objective QoE metrics such as playback delay, video quality switches, and stalls, while also monitoring privacy risks by detecting tracking and advertisement related data flows. Although it offers a customizable and detailed solution for QoE analysis, its effectiveness can vary depending on the device and operating system, requires technical setup knowledge, focuses mainly on specific content types (video), and does not collect subjective user feedback.

To provide a clearer understanding of recent advancements, several studies have explored QoE prediction for video streaming using machine learning models across diverse network environments. These works vary in input parameters, such as packet loss and user feedback, and use models like regression, SVM, and spline approximations.

In conclusion, this chapter has explored the intricate world of multimedia services within mobile networks. We have examined the building blocks of multimedia services, the architecture of LTE networks, and how they work together to deliver content to users on the go. We have also delved into the concepts of QoS and QoE, highlighting their critical roles in user satisfaction. By understanding these fundamental aspects, we can begin to analyze and optimize the performance of mobile networks for multimedia applications.

Building upon this foundation, Chapter 3 will further explore Discrete Markov Models for mobile network performance analysis. These models provide a powerful tool for mathematically

representing and evaluating network behavior under various conditions. By leveraging such models, we can gain deeper insights into network performance bottlenecks, predict potential issues, and ultimately develop strategies for ensuring a superior and consistent user experience for multimedia services in mobile networks.

3 Markov Model for Mobile Network Analysis

The previous chapter examined how multimedia services, network performance (QoS), and user experience (QoE) interact in mobile networks. While QoS metrics provide a network-centric view, accurately capturing user-perceived quality in dynamic environments requires probabilistic modeling. This chapter introduces and tests Markov-based models as a foundation for QoS-to-QoE modeling.

The chapter presents Discrete Markov Models (DMMs) for analyzing observable state transitions in mobile networks accessibility and retainability, and HMMs for estimating unobservable performance states based on measured indicators. By applying both models to real-world mobile network data, this chapter evaluates their ability to capture performance dynamics and predict system behavior. These results serve as the analytical groundwork for QoE prediction, which is the core focus of the next chapter.

3.1 Discrete Markov Models

3.1.1 Mathematical Models

Mathematical models are essential tools to represent and analyze real-world systems. These models fall into two main categories: deterministic and stochastic. Deterministic models assume fixed relationships between variables, leading to predictable outcomes for a given set of initial conditions. Although valuable in controlled environments, they struggle to capture the inherent randomness present in many physical or social systems.

Stochastic models address this limitation by incorporating elements of probability. These models acknowledge the inherent variability in real-world systems and offer a more realistic picture by estimating the likelihood of various outcomes. This makes stochastic models particularly useful for analyzing complex systems such as mobile networks, where variability and uncertainty are common. In such contexts, probabilistic modeling enables both system behavior prediction and uncertainty quantification—critical for anticipating performance fluctuations and user experience.

3.1.2 Discrete Markov Models for Temporal State Modeling

DMMs are a powerful class of stochastic models specifically suited for analyzing systems that exhibit sequential states. DMMss encompass two main subcategories: Discrete Markov Chain (DMC) and HMM. Both leverage the concept of a Markov chain, a stochastic model introduced by Andrey Markov in 1906. These models are particularly relevant for mobile network performance analysis, where system behavior evolves over time and can be observed through measurable indicators or inferred from user experience.

3.1.3 Markov Chain Model

A Markov chain models a sequence of events where the probability of transitioning to a future state depends solely on the current state, not on any previous states. This simplifying assumption, known as the "memoryless" property, facilitates efficient computational approaches while still capturing essential system dynamics. Markov chains offer a versatile tool for analyzing sequential data across diverse fields due to their ability to strike a balance between completely dependent and fully independent systems. In this chapter, this concept is applied to both directly observable network conditions and latent user experience states using appropriate Markov formulations.

3.1.4 Subcategories of Discrete Markov Model

- **DMCs:** Ideal for situations where system states are directly observable (e.g., network congestion levels), DMC utilize a set of states and transition probabilities. These probabilities define the likelihood of moving between states. Analyzing these probabilities allows network operators to gain valuable insight into system behavior, such as predicting the likelihood of network congestion based on the current state.

- **HMM:** Extend the concept of DMCs by introducing hidden states. In HMM, the true state of the system may not be directly observable, but we can infer its influence through a sequence of observable outputs (emissions). For instance, in mobile networks, signal strength fluctuations could be observable outputs, while the underlying network congestion levels could represent hidden states. By analyzing the sequence of observed signal strengths, HMM can estimate the probability of hidden congestion states, enabling network operators to make informed decisions even with limited direct observation.

Together, DMC and HMM form a comprehensive modeling framework that enables both direct

analysis of measurable performance indicators and inference of latent system behavior. This dual capability is essential for developing robust QoS-to-QoE mapping.

3.2 Discrete Markov Chain: Definition and Properties

DMCs are mathematical models suited for representing systems that undergo sequential state changes. They capture stochastic transitions between a finite number of discrete states over time, using well-defined probability rules. These chains model systems that randomly transition between distinct states including remaining in the same state, based on predefined transition probabilities rather than deterministic rules.

DMCs are particularly effective when analyzing systems with discrete and countable state spaces, such as mobile networks where signal levels, congestion states, and user activities can be classified into finite categories.

3.2.1 Properties of Discrete Markov Chain Model

- **States:** A DMC assumes a system can exist in various states, representing different conditions or situations relevant to multimedia services and mobile networks (e.g., low, medium, or high signal strength; low or high congestion levels; idle or active user playback states). These states are both discrete and finite, forming the system's state space. The system transitions between these states in discrete time steps according to the model's transition probabilities.

- **Transitions:** Transitions depict the movement of the system from one state to another. These shifts reflect the dynamic behavior of the system over time. For example, in mobile networks, a user may move from a low signal area to a high signal area due to physical movement, impacting service quality. Transitions are influenced by factors such as infrastructure coverage, user mobility, and concurrent usage levels.

- **Transition Probabilities:** These define the likelihood of transitioning from one state to another and are central to the DMC model. They are typically derived from historical data or simulations and reflect system-specific dynamics. For instance, the probability of transitioning from "low signal" to "high signal" during a video call depends on user movement and cell tower density. Understanding these probabilities enables performance forecasting and anomaly detection.

- **Initial (Probability) Distribution:** This vector specifies the probability of the system starting in each state at time $\mathbf{t} = 0$. It encodes the system's initial conditions and is essential for time-dependent predictions.
- **Steady-State Distribution:** Under certain conditions, a DMC reaches a stable distribution where state probabilities become time invariant. This long-run behavior is particularly useful for analyzing persistent trends in network performance.
- **Markov Property:** The defining characteristic of a DMC is the memoryless property, which states that the probability of transitioning to the next state depends solely on the current state, not on the sequence of past states. This simplifies the analysis of complex systems by eliminating the need to track historical sequences.

3.2.2 Applications in Mobile Networks

Understanding these core properties enables the use of DMCs to model and optimize various aspects of mobile multimedia service delivery:

- Analyze performance trends by examining transition matrices or visualizing state graphs.
- Predict future states (e.g., congestion, signal strength) using the current state and transition probabilities.
- Optimize service quality by identifying and addressing high-risk transitions or unstable steady states.

For example, analyzing transition matrices may help forecast periods of likely video buffering due to expected signal degradation, allowing proactive mitigation through network resource allocation or adaptive streaming algorithms.

3.2.3 Derivation of the Transition Matrix

One of the fundamental building blocks of a DMC is the transition matrix, which captures the probabilistic structure of state transitions within the system. This section outlines the derivation process, aiming to provide a clear and structured understanding of how the matrix represents the temporal evolution of states. Using a step-by-step method, we demonstrate how transition probabilities are computed to quantify the likelihood of moving from one state to another in a single time step. This understanding forms the basis for applying the transition

matrix in more advanced models, such as HMM's, where it is essential for describing hidden state dynamics.

Consider a discrete-time and random source with $\{s_1, s_2, \ldots, s_N\}$ designating the set of possible states of the source while N designating the alphabet size or a number of forms of the source [149]. Further consider that the source generates h time series data (random variables observed in the time dimension) designated as $x_1, x_2, x_3, \ldots, x_i, \ldots, x_h$ where i in x_i indicates the time index and $x_i \epsilon \{s_1, s_2, \ldots, s_N\}$. In general, applying the Chain rule in probability, the joint likelihood of these random variables can be written as:

$$P(x_1, x_2, \ldots, x_h) = P(x_1)P(x_2|x_1)P(x_3|x_2, x_1)\ldots P(x_h|x_{(h-1)} \ldots, x_2, x_1) \tag{3.1}$$

If the source obeys the first-order Markov chain, every random variable in $x_1, x_2, x_3, \ldots, x_h$ is only dependent on the latest previous random variable and not the other ones [150]. In other words, Equation (3.2) defines the transition probability. With (first order) Markov property, the Chain rule in Equation (3.1) is simplified to:

$$P(x_i|x_(i-1), x_(i-2,) \ldots, x_2, x_1) = P(x_i|x_{(i-1)}) \tag{3.2}$$

Equation (3.3) designates the likelihood of observing a sequence.

$$P(x_1, x_2, x_3, \ldots, x_h) = P(x_1)P(x_2|x_1)P(x_3|x_2) \ldots P(x_h|x_{(h-1)} \tag{3.3}$$

We learn from the equation that only the most recent state (information) matters to predict the probability of observing the sequence. Markov property can be of a higher order [25]. For example, for a second order Markov chain, a random variable depends on the latest and one-to-latest variables. Usually, the default Markov property is of order one. With knowledge of one-step transition probability, the Markov chain is also formulated to predict, in a probabilistic sense, the distribution of the states after the system undergoes certain transitions in the future.

As an example [1], consider that a certain source has only three states designated as $\{a_0, a_1, a_2\}$ as shown in Figure 3.1. If we consider transitions for three future time instants t_{n-2}, $t_{n-1}$and t. At time t_{n-2} the source's output can be a_0, a_1,and a_2 and let P_0, P_1 and P_2 define the probabilities that the system will be in these states, respectively. At time t, the set of all possible transitions for the system to be in the state a_0 are shown by the red lines. Further, let the probability of the system's output transitions from state i to state j be given by P_{ij}.

The probability of being in this state after two transitions, designated by $P_0^{(2)}$, is given by:

1 Note that for this example, a_i are used instead of s_i to designate the states

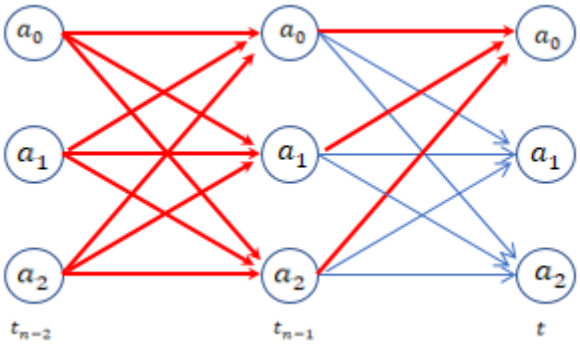


Figure 3.1: State transitions for a source with three states are observed in three-time instants.

$$P_0^{(2)} = P_0(P_{00}P_{00}+P_{01}P_{10}+P_{02}P_{20})+P_1(P_{10}P_{00}+P_{11}P_{10}+P_{12}P_{20})+P_2(P_{20}P_{00}+P_{21}P_{10}+P_{22}P_{20}) \quad (3.4)$$

We can similarly write for $P_1^{(2)}$ and $P_2^{(2)}$. If we collect identical terms and rearrange them in the form of a matrix, we get that:

$$[P_0^{(2)}, P_1^{(2)}, P_2^{(2)}] = [P_0, P_1, P_2]\begin{pmatrix} P_{00} & P_{01} & P_{02} \\ p_{10} & P_{11} & P_{12} \\ P_{20} & P_{21} & P_{22} \end{pmatrix}\begin{pmatrix} P_{00} & P_{01} & P_{02} \\ p_{10} & P_{11} & P_{12} \\ P_{20} & P_{21} & P_{22} \end{pmatrix} = UP^2 \quad (3.5)$$

Where the matrix $p = \begin{pmatrix} P_{00} & P_{01} & P_{02} \\ p_{10} & P_{11} & P_{12} \\ P_{20} & P_{21} & P_{22} \end{pmatrix}$ is the transition probability; P^2 is square of the matrix; and the vector $U = [P_0, P_1, P_2]$ is the initial (probability) distribution. In general, the states probability distribution after k transitions in the future is given by [150]

$$U^{(k)} = UP^k \quad (3.6)$$

P^k is the result of multiplying the matrix P k times by itself. Each element of $P(k)$, designated as $P_i j^{(k)}$, is the probability of going from state i to state j in k iterations.

One inherent assumption in Markov chain is that the random variables, resulting from state transitions, are observable, and transition probabilities can be computed directly from the observations. In many real-world problems, however, the states we are interested in are hidden or non-observable. This was the main motivation for the development of HMM and is presented in the next section.

3.2.4 Advanced Properties of Markov Chains

Discrete Markov Chains exhibit a rich set of properties that provide deeper insights into their long-term behavior. Understanding these advanced characteristics can be crucial for specific applications and theoretical analysis.

- **Communicating States:** Two states in a Markov chain are considered communicating states if it is possible to transition between them with a non-zero probability in either direction. In simpler terms, there exists a path from state i to state j and vice versa, allowing the system to move between them.

- **Absorbing States:** An absorbing state acts as a "trap" for the system. Once a chain enters an absorbing state, it remains there indefinitely with no possibility of transitioning out. A common example arises in modeling insurance systems, where a state representing "policy withdrawal" or "death of the insured" can be considered absorbing.

- **Transient vs. Recurrent States:** A transient state is one that the system might potentially never revisit after entering it. Conversely, a recurrent state guarantees the system's return at some point in the future. Every state in a DMC is classified as either transient or recurrent.

- **Irreducibility:** A DMC is considered irreducible if it's possible to move from any state to any other state, regardless of the number of transitions required. This implies all states belong to a single communication class, meaning they are ultimately connected.

- **Aperiodicity:** Aperiodic chains lack a periodic pattern in returning to a state. This means the system doesn't require a specific number of transitions (period) to revisit a state after leaving it. Conversely, a periodic chain exhibits a specific period, requiring a multiple of that period to return to a state after departure.

- **Positive Recurrence:** A state in a DMC is positive recurrent if there's a non-zero probability of returning to that state after a finite number of transitions from any initial state. This concept is crucial for analyzing long-term system behavior.

- **Ergodicity:** A Markov chain is considered ergodic if all its states are both aperiodic and positive recurrent. In essence, an ergodic chain guarantees the system's eventual return to any state with a non-zero probability, regardless of the starting state. This property allows for the calculation of long-term averages and steady-state behavior.

- **Reversibility:** A Markov chain is said to be reversible if the probability of transitioning between two states in one direction is equal to the probability of transitioning between them in the reverse direction. This property implies a system's past doesn't influence its future behavior, aligning with the core Markov property. However, reversibility is a strict condition that is rarely encountered in practical applications.

3.2.5 Computing Transition Probabilities from Data

Estimating future states using a Markov chain typically requires initial values for model parameters: the initial probability distribution and the transition probability matrix [20]. For small, simple systems, any random initial distribution might suffice. However, large and complex systems benefit from well-chosen starting values to ensure efficient convergence towards an optimal solution.

The transition matrix, which captures the likelihood of transitioning between states, can be estimated using a Maximum Likelihood approach [151]. Consider a system with N states denoted as $S_1, S_2, \ldots, S_N$. Let n_{ij} represents the number of observed transitions from state S_i to state S_j and n_i denote the total number of transitions originating from state S_i. Based on the observed data, the Maximum-likelihood estimates of the transition probability P_{ij} is

$$P_{ij} = \frac{n_{ij}}{n_i} \tag{3.7}$$

This equation essentially calculates the proportion of transitions from state S_i that end up in state S_j. Moreover,

$$\sum_{j=1}^{N} n_{ij} = n_i \tag{3.8}$$

And

$$\sum_{j=1}^{N} P_{ij} = 1 \tag{3.9}$$

In some cases, particularly with limited data, there might be zero observations of transitions from a specific state S_i to another state S_j. To avoid undefined probabilities (division by zero), smoothing techniques like Laplace smoothing can be applied. This involves adding a small value (e.g., 1) to all transition counts, n_{ij}, before performing the calculation.

The estimated transition probabilities can be conveniently organized and presented in a transition probability matrix, as shown in Table 3.1.

Table 3.1: Table for transition probabilities computation

State	1	2	...	N	Row Sum
1	n_{11}	n_{12}		n_{1N}	n_1
.	.	.	.	.	.
.	.	.	.	.	.
N	n_{N1}	n_{N2}		n_{NN}	n_N

3.2.6 Applications of Markov Chain in Mobile Network Analysis

The increasing demand for reliable mobile services necessitates constant performance monitoring of mobile networks, especially radio access networks (RAN), which are the backbone of mobile connectivity. Traditionally, network operators have relied on reactive approaches triggered by threshold violations of key performance indicators (KPIs) such as accessibility and retainability. These methods often lead to delayed interventions and service disruptions.

With the proliferation of historical data collected via network management systems (NMS), there is now an opportunity to adopt a more proactive approach. Markov chains, as probabilistic models for sequential processes, offer a compelling framework to model temporal dynamics and predict performance fluctuations in mobile networks. Recent studies highlight the diverse applications of MCs in mobile network analysis:

- **Call Admission Control (CAC):** Studies like [144] explore using Markov chains to optimize CAC algorithms, ensuring network stability by managing incoming call traffic.
- **QoE and QoS Modeling:** Research in [145], [146] demonstrates how Markov chains can model user experience and service quality by capturing network resource availability and user behavior patterns.
- **Resource Utilization:** Works like [16] delve into optimizing resource allocation within the network using Markov chains, leading to improved network efficiency.
- **User Mobility Prediction:** Predicting user movement patterns is crucial for network planning. Papers like [148] explore leveraging Markov chains to forecast user mobility, enabling better resource allocation and handover management.

- **Network Operation Status Monitoring:** Research in [152] highlights the use of Markov chains for monitoring the overall network health, identifying potential bottlenecks and performance degradation.
- **RRC Setup Success and Call Setup Success Rates (CSSR):** Works like [153], [154] propose using Markov chains to forecast these metrics, allowing operators to anticipate potential connection issues within cells. These metrics are crucial indicators of network accessibility and user experience,usually the cell states are represented as "Good," "Moderate," or "Bad" based on historical RRC success rates. Factors like time of day, location, and network load influence a cell's state.

Markov chains are also increasingly used in combination with other analytical techniques:

- **Clustering:** Studies like [155] integrate Markov chains with clustering algorithms (e.g., K-means) to manage high-dimensional, spatially diverse network data. This approach reduces the number of modeled states, enhances computational efficiency, and preserves location-specific performance insights.
- **Machine Learning:** Hybrid models that combine decision trees or neural networks with Markov chains (e.g., [156], [150]) provide alternative predictive strategies. However, these can face scalability challenges in large, complex networks.

3.2.7 Prediction of Radio Access Network Performance with K-means Clustering and Markov Chains

Building on the theoretical and empirical applications of Markov models, this section presents our implementation for mobile network performance prediction. It specifically focuses on forecasting key RAN performance indicators, namely accessibility and retainability, by utilizing historical network data. The objective is to demonstrate how the integration of Markov chains with K-means clustering enables proactive identification of potential performance degradations in cellular networks. The results of this implementation have been peer-reviewed and published in a scientific journal, providing further validation of the model's robustness and practical relevance.

3.2.7.1 System Model

The proposed system model utilizes real-time NMS data to predict the accessibility and retention status of mobile cells. It applies Markov chains to capture temporal transitions between

performance states and integrates K-means clustering to reduce the dimensionality of the joint state space. This clustering also enables spatial segmentation, ensuring the model remains scalable and accounts for location-specific variability in network conditions. Together, these techniques form a hybrid framework that is both computationally efficient and practically effective for anticipating network behavior across diverse operational scenarios.

- **Data Acquisition:** The foundation of the model lies in a rich dataset of real-time hourly accessibility and retainability KPI. This data was collected for a four-month period (November 2020 - February 2021) from a diverse network of 1530 base stations (cells) operated by a major network operator in Addis Ababa, Ethiopia.
- **State Definition Aligned with ITU Standards:** To ensure a standardized and industry-aligned approach to performance evaluation, the model defines accessibility and retainability states based on the ITU recommendations. This categorization utilizes four distinct states: "**Idle**," "**Good**," "**Acceptable**," and "**Bad**." This deviation from previous works, which may have employed different state definitions, allows for direct comparisons and facilitates the interpretation of the model's results within the wider context of mobile network performance evaluation.
- **Addressing Spatial Variations with K-means Clustering:** While the collected data offers valuable insights, it is crucial to consider the spatial distribution of the base stations. Since these cells are geographically dispersed across Addis Ababa, network performance may vary significantly depending on factors like land use, settlement patterns, and user behavior in each location. To account for these spatial variations without significantly increasing model complexity, K-means clustering is employed. This unsupervised machine learning technique identifies groups (clusters) of cells exhibiting similar performance characteristics. By focusing on these clusters rather than individual cells, the model can capture the spatial nuances of network performance while maintaining scalability.
- **Markov Chain Modeling for State Prediction:** The core of the system model is the application of Markov chains for predicting accessibility and retainability states. Markov chains are a powerful probabilistic modeling tool that analyze sequences of events, assuming that the probability of transitioning to a future state depends only on the current state, not on the history leading to that state. In this context, the states represent the different levels of accessibility and retainability ("**Idle**," "**Good**," "**Acceptable**," and "**Bad**"), and the transitions between states reflect the dynamic nature of network performance.

  Two primary approaches are implemented for model formulation:

- **Separate Models:** This approach constructs two independent Markov chain models, one for accessibility and one for retainability. This allows for a more focused analysis of each KPI, but requires the development and maintenance of two separate models.
- **Joint Model:** This approach utilizes a single, more complex Markov chain model to predict both accessibility and retainability jointly. While computationally more demanding, this approach offers improved efficiency by requiring only one model and capturing the potential correlations between accessibility and retainability.it is computationally more demanding due to the increased dimensionality of the state space. Both approaches enable the calculation of the network state and the number of transitions required to reach a steady-state, where the probabilities of transitioning between states stabilize.

Both approaches enable the calculation of the network state and the number of transitions required to reach a steady-state, where the probabilities of transitioning between states stabilize. As outlined in the following sections, K-means clustering is used to reduce the number of states in order to overcome the computational challenges triggered due to the joint model's dimensionality.

3.2.7.2 Contributions of the Markov Chains Model with K-means Clustering

This research offers two key contributions to the field of mobile network performance analysis:

1. **State Definition Aligned with ITU Standards:** By adhering to ITU recommendations for state definitions, the model provides a standardized approach for accessibility and retainability evaluation. This facilitates cross-study comparisons and promotes consistent interpretation of results within the mobile network management community.

2. **Joint Prediction for Enhanced Efficiency:** The proposed joint model offers a single operation for predicting both accessibility and retainability, improving efficiency compared to separate models.

3. **Dimensionality Reduction for Improved Scalability:** The application of K-means clustering for dimensionality reduction significantly improves the scalability and computational efficiency of the joint Markov chain model, enabling its application to large-scale networks.

3.2.7.3 K-means Clustering for Scalability

Conventional approaches that focus on individual cells often fail to capture spatial correlations between network performance metrics. Furthermore, in joint modeling, including all cells and KPIs in a single model would exponentially increase the number of states, making the model computationally expensive and impractical. K-means clustering addresses these challenges by identifying clusters of cells with similar performance characteristics, thereby enabling scalable prediction while accounting for spatial variations in network performance and reducing the dimensionality of the joint model. This clustering process allows us to work with a reduced 4x4 Markov chain model instead of a more complex 16x16 model.

Overall, the system model leverages K-means clustering and Markov chains to achieve efficient and accurate prediction of accessibility and retainability status. This proactive approach empowers network operators to anticipate potential issues and take preventive measures, ultimately enhancing mobile network performance and user experience.

3.2.7.4 Accessibility and Retainability in Mobile Networks

KPI obtained from network counters are essential for managing and tracking network performance and can be grouped into categories such as accessibility, retainability, integrity, mobility, and other factors [27]. According to ITU, service accessibility is "the ability of a service to be obtained, within specified tolerances and other given conditions, when requested by the user." Service retainability is "the ability of a service, once obtained, to continue to be pro-vided under given conditions for a requested duration" [157].

- **Accessibility:** The accessibility KPI is expressed in probabilities, which indicate how likely a user is able to access the mobile service during specific service times and conditions. Accessibility measures the network's performance during call setup or before establishing a bearer [14].

  For data availability reason, this dissertation focuses on the 3G mobile networks. RRC, radio access bearer radio access bearer (RAB), Enhanced Universal Terrestrial RAN RAB, and CSSR are critical accessibility parameters as they directly reflect the users' ability to connect to and utilize the network.

  - RRC setup success rate (RRC SSR) evaluates the call success rate in a cell or cluster. The formula for this KPI is:

$$RRC\ SSR = \frac{Number\ of\ RRC\ setup\ success}{number\ of\ RRC\ connection\ attempt} \times 100\%. \tag{3.10}$$

- RAB setup success rate (RAB SSR) evaluates the success rate of assigning a RAB during a call setup procedure. The formula for this KPI is given as follows:

$$RAB\ SSR = \frac{Number\ of\ RAB\ setup\ success}{number\ of\ RAB\ setup\ attempt} \times 100\%. \tag{3.11}$$

- CSSR is used to evaluate the call setup success at the cell or cluster level. This KPI is calculated based on RRC setup success rate (SSR) and RAB SSR for the case of 3G networks and Enhanced Universal Terrestrial RAN RAB (ERAB) SSR for the case of LTE networks.

$$Accessibility = CSSR = RRC\ SSR \times RAB\ SSR \times 100\%. \tag{3.12}$$

- **Retainability:** Retainability assesses a network's performance after RAB is established and indicates the proportion of sucessful calls are maintained being dropped.

$$Retainability = (1 - \frac{Number\ of\ RAB\ abnormal\ release}{Total\ number\ of\ RAB\ release}) \times 100 \tag{3.13}$$

The fraction in Equation (3.13) represents Call drop rate.

3.2.7.5 Analysis of Mobile Network Performance

This section describes the data collection and pre-processing methods used to predict accessibility and retainability status using four- and sixteen-state Markov chain models.

1. **Data Collection and Pre-processing** Real-time hourly data was collected from 1530 cells over a 4-month period using the Performance report system (PRS) installed on the operator's network.

   Linear interpolation was used to fill in data gaps caused by factors such as cell outages and connection problems between cells and the central Radio Network Controller (RNC).

   When no voice or data service attempts occur in a cell for one hour, both accessibility and retainability values are set to zero, and the corresponding status is designated as 'Idle'. In this case, the "Idle" label is assigned based on the attempt counters (i.e., no trials occurred during that hour). Since CSSR/CDR are not applicable when attempts are zero, the values are set to zero only as a preprocessing convention to keep the observation data numeric for Markov/HMM analysis; this should not be interpreted as "zero success" or a network

failure. Figure 3.2 shows RRC and RAB attempt values for Cluster 6 throughout a week, and both values are zero from midnight to 04:00 am, using Cluster 6 as a representative illustrative cluster; other clusters exhibit comparable hourly patterns, including periods with no service attempts.

The data collected was divided into two subsets, 60% utilized for training and 40% for model validation/testing. Training data was used to generate the transition matrix, with the matrix building process detailed in [152]. For comparisons,the data was also split using 70/30 and 80/20 ratios.

The model predicts the next probability vector based on the current state probability distribution and the transition matrix. This iterative process continues until a steady-state condition is reached. Finally, prediction accuracy is assessed by comparing the results with the validation data. This preprocessed data will then be used in conjunction with K-means clustering to analyze network performance.

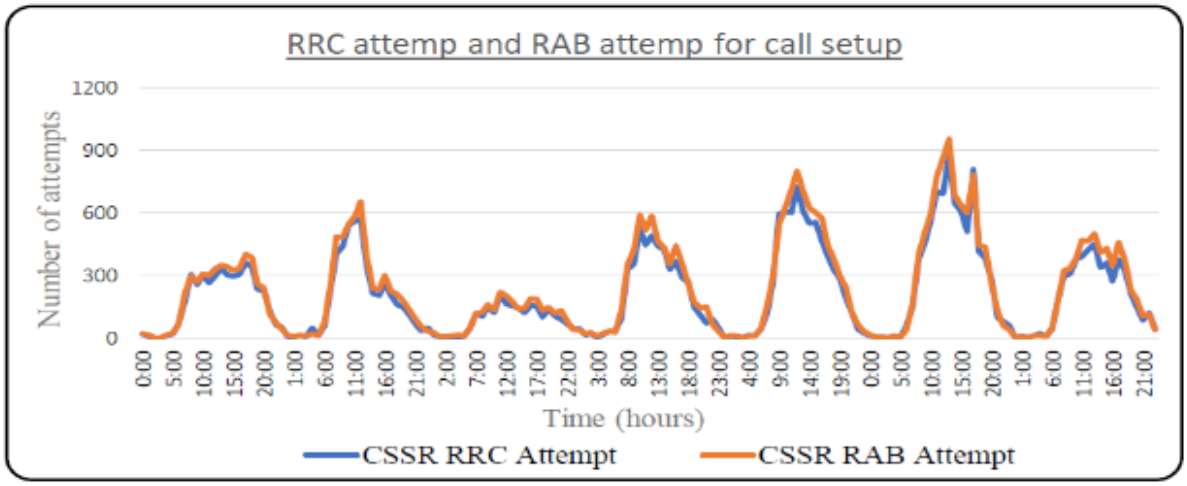


Figure 3.2: One-week RRC and RAB attempts.

2. **Clustering** Analyzing performance patterns of individual cells in a large network is time-consuming. Therefore, in this research, we suggest K-mean clustering as a method for grouping cells with similar accessibility and retainability properties. This approach reduces the complexity of the analysis by allowing model construction and prediction to be based on per-cluster averaged accessibility and retainability. The optimal number of clusters was determined using the Elbow method, where the number of clusters was varied from 2 to 18. To ensure robust cluster formation, each cell was initially randomly assigned to a cluster, and the cluster centroids were iteratively adjusted until the reduction

in cluster variation was minimal. We discovered that a clustering value of 6 is adequate. Hourly data acquired from each cell varies from 0% to 100%; however, if no voice or data service requests are received in a cell for 1 hour, all counter values for that hour are zero, as illustrated in Figure 3.2

3. **KPI Thershold for Definition** Operators set threshold values for several KPIs based on the ITU's recommendations, considering variables such as capital expenditures, operational expenses, QoS, and customer satisfaction. Tables 3.2 and 3.3 display a threshold value for the considered operator's accessibility and retainability. Based on the values in the two tables, the states of accessibility and retainability are generated.

Table 3.2: Possible values of Call Setup Attempt and Call Setup Success Rate.

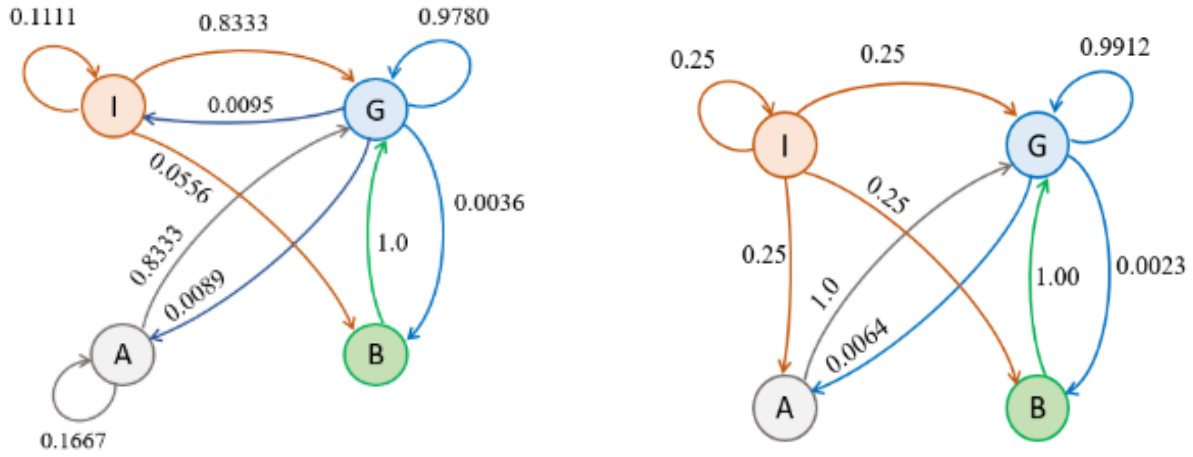
Call Setup Attempt	Value	State of a Cell
> 0.0	$CSSR \geq 98.0\%$	Good (G)
> 0.0	$95.0\% \leq CSSR < 98.0\%$	Acceptable (A)
> 0.0	$0.0\% < CSSR < 95.0\%$	Bad (B)
$= 0.0$	-	Idle (I)

Table 3.3: Possible values of Radio Access Bearer setup success and Call Drop Rate.

RAB Setup Attempt	Value	State of a Cell
> 0.0	$0.0\% < CDR \leq 1.0\%$	Good (G)
> 0.0	$1.0\% < CDR \leq 3.0\%$	Acceptable (A)
> 0.0	$CDR > 3.0\%$	Bad (B)
$= 0.0$	-	Idle (I)

4. **Separate Prediction** As established earlier, predictions for accessibility and retainability at the cluster level can be performed using both separate and joint Markov chain models. Four states are required for the separate case. Hence, the corresponding transition matrices are 4×4. Figure 3.3 below, illustrates the state transition probability diagrams for cluster 6, derived after developing the separate Markov chain models for (a) accessibility and (b) Retainability.

   Note that there are missing arrows in the two figures. As an example, there is no arrow in Figure 3.3 pointing from state A to state I, indicating that such a transition does not exist or the system has never landed in an idle state if it was initially in the Acceptable state.



(a) Accessibility **(b)** Retainability

Figure 3.3: Transition probability diagram of cluster 6.

5. **Joint prediction** Joint estimation allows for a more comprehensive analysis by considering the combined states of accessibility and retainability. This approach enables the observation of simultaneous occurrences of different performance levels, providing insights into potential correlations between these KPIs. For instance, it reveals how often 'Good' accessibility coincides with 'Bad' retainability, or vice-versa, which separate models cannot capture. Specifically, joint estimation tracks all possible combinations of the four ITU defined states ('**Idle**,' '**Good**,' '**Acceptable**,' and '**Bad**') for both accessibility and retainability. Consequently, when considering all possible state pairings, the number of states increases significantly. While separate models use 4×4 transition matrices, the joint model results in a 16×16 transition matrix (4 accessibility states × 4 retainability states). This increase in dimensionality highlights the importance of the K-means clustering employed earlier, which reduces the number of entities (cells) and thus helps manage the complexity of this joint analysis. As illustrated in Table 3.4, which shows the possible combinations of the four ITU defined states ('**Idle**,' '**Good**,' '**Acceptable**,' and '**Bad**') for both accessibility and retainability, directly applying Markov chain modeling to this high-dimensional space introduces considerable computational complexity and hinders scalability, particularly for large-scale mobile networks.

Table 3.4: Transition Probability Matrix of Cluster 6 with 16 States.

	II	*IG*	*IA*	*IB*	*GI*	*GG*	*GA*	*GB*	*AI*	*AG*	*AA*	*AB*	*BI*	*BG*	*BA*	*BB*
II	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625
IG	0.0000	0.1176	0.0000	0.0000	0.0000	0.7647	0.0588	0.0000	0.0000	0.0000	0.0000	0.0000	0.0588	0.0000	0.0000	0.0000
IA	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625
IB	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625
GI	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625
GG	0.0000	0.0000	0.0000	0.0000	0.9707	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0293	0.0000	0.0000	0.0000
GA	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
GB	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
AI	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625
AG	0.0000	0.0000	0.0000	0.0000	0.0000	0.8333	0.0000	0.0000	0.0000	0.1667	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
AA	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625
AB	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625
BI	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625
BG	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
BA	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
BB	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625

6. **State prediction** After creating the transition matrices and knowing the current/initial state distribution, the next state and steady state distributions are predicted using Equation (3.6). If the current state is (assumed to be) in a 'Good' state, then the value of π_0 is,

$$\pi = \begin{bmatrix} I & G & A & B \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \tag{3.14}$$

Then, using Equations (3.6) and (3.14), the next accessibility probability is π_1 for one of the clusters when computed, and the result is:

$$\begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{pmatrix} 0.1111 & 0.8333 & 0.0000 & 0.0556 \\ 0.0095 & 0.9780 & 0.0089 & 0.0036 \\ 0.0000 & 0.8333 & 0.1667 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 & 0.0000 \end{pmatrix} = \begin{bmatrix} 0.0095 & 0.9780 & 0.0089 & 0.0036 \end{bmatrix} \tag{3.15}$$

According to the result, the system has a 0.95% chance of going to the Idle state, a 97.80% chance of staying in a Good state, a 0.89% chance of going to the Acceptable state, and a 0.36% chance of going to the Bad state.

Equation (3.15) is used to find the steady-state distribution calculated iteratively until the next and previous state values are equal. Tables 3.5 and 3.6 display the steady-state results for the four-state Markov chain regarding accessibility and retainability. Table 3.7 depicts the cluster 1 steady-state distribution using the sixteen-state Markov chain. For both scenarios, 70% of the data are used as a training set.

In Table 3.6, the maximum value in the Good state from the six clusters is 99.26% in cluster 1, and the minimum value is 97.22% in clusters 2 and 6. The maximum value of the Bad state is 1.29% in cluster 2, and the minimum value is 0.1% in cluster 3. From this cluster, one cell is at the top in the Good state, and cluster 2 cells are at the top in the Bad state.

Though cluster 6 cells are the least in the Good state, they are not at the top in the Bad state because, next to the Good state, cluster 6 cells have a high probability (1.09%) of being in the Idle state. So, if optimization or maintenance work is needed, the schedule and priority should be given based on the steady-state vector values of each cluster.

Steady state distribution for the sixteen-state Markov chain follows the same approach. Cluster 1's steady-state outcome is shown in Table 3.7.

The first letter stands for accessibility, while the second stands for retainability, and

Table 3.5: Steady-state vector of retainability using four-state Markov chain.

Cluster	Steady-State Vector of Retainability			
	I	G	A	B
1	0.0000	0.9980	0.0020	0.0000
2	0.0000	0.9955	0.0035	0.0010
3	0.0000	1.0000	0.0000	0.0000
4	0.0000	0.9980	0.0020	0.0000
5	0.0000	0.9965	0.0030	0.0005
6	0.0000	0.9901	0.0079	0.0020

Table 3.6: Steady-state vector of accessibility using four-state Markov chain.

Cluster	Steady-State Vector of Accessibility			
	I	G	A	B
1	0.0000	0.9926	0.0060	0.0015
2	0.0000	0.9722	0.0149	0.0129
3	0.0000	0.9911	0.0079	0.0010
4	0.0000	0.9782	0.0179	0.0040
5	0.0000	0.9916	0.0055	0.0030
6	0.0109	0.9722	0.0114	0.0055

99.26% of the time, accessibility and retainability were in the Good state, while for 0.4% of the time accessibility was in the Acceptable state and retainability was in the Good state. Furthermore, for 0.2% of the time, accessibility and retainability were both in the Acceptable state, while 0.15% of the time, accessibility was Bad, and retainability was in the Good state. As a result, the table provides cell information relating to accessibility and retainability, allowing operators to quickly sort cells that perform poorly in either or a combination of the two performance measures.

7. **Evaluation Metric** The accuracy of a model was assessed using Equation (3.16), [158], which calculates the percentage of correctly forecasting the next state given the current state.

$$Accuracy = \frac{Correct\ predictions}{Total\ number\ of\ examples} \times 100\% \tag{3.16}$$

Table 3.8 below shows the accuracy results for different combinations of training data proportion, clusters, four vs sixteen states modeling and the two KPI considered. As an example, we note that a minimum value of 96.09% prediction accuracy is achieved in cluster 2 in predicting accessibility when 60% training set is used, while 96.87% prediction accuracy is achieved in cluster 5 in predicting retainability when the 80% training set is

Table 3.7: Steady-state vector using a sixteen-state Markov chain.

Cluster	Steady-State Vector of accessibility and retainability			
	[II IG IA IB]	[GI GG GA GB]	[AI AG AA AB]	[BI BG BA BB]
1	0.0000	0.0000	0.0000	0.0000
	0.0000	0.9926	0.0000	0.0000
	0.0000	0.0040	0.0020	0.0000
	0.0000	0.0015	0.0020	0.0000
2	0.0000	0.0000	0.0000	0.0000
	0.0000	0.9692	0.0144	0.0119
	0.0000	0.0025	0.0005	0.0005
	0.0000	0.0005	0.0000	0.0005
3	0.0000	0.0000	0.0000	0.0000
	0.0000	0.9911	0.0079	0.0010
	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000
4	0.0000	0.0000	0.0000	0.0000
	0.0000	0.9782	0.0179	0.0020
	0.0000	0.0000	0.0000	0.0002
	0.0000	0.0000	0.0000	0.0000
5	0.0000	0.0000	0.0000	0.0000
	0.0000	0.9906	0.0050	0.0010
	0.0000	0.0010	0.0005	0.0015
	0.0000	0.000	0.0000	0.0005
6	0.0000	0.0000	0.0000	0.0000
	0.0104	0.9638	0.0114	0.0045
	0.0005	0.0065	0.0000	0.0010
	0.0000	0.0020	0.0000	0.0000

used. A 94.61% prediction accuracy is achieved in cluster 6 in predicting both accessibility and retainability when 80% of the data are used for training and when the modeling is the case of the sixteen-state Markov chain.

Table 3.8: Prediction accuracy comparison of 4 vs 16-state Markov chain.

Cluster	Training Set	Accessibility Accuracy Using 4-States	Retainability Accuracy Using 4-States	[(Col.3 x Col.4)/100] (%)	Accessibility and Retainability Accuracy using 16-States
	60%	98.7837	99.1312	97.9254	97.8280
	70%	98.3796	98.8426	97.2410	97.1065
1	80%	98.0870	98.2609	96.38110	96.1739
	60%	96.0904	98.69682	94.8381	95.5691
	70%	96.7593	98.3796	95.1914	96.1806
2	80%	96.1739	97.7391	93.9995	95.4783
	60%	98.5230	100.0000	98.5230	98.5230
	70%	98.4954	100.0000	98.4954	98.4954
3	80%	98.6087	100.0000	98.6087	98.6087
	60%	98.5230	100.0000	98.5230	98.5230
	70%	98.3796	100.0000	98.3796	98.3796
4	80%	98.6087	100.0000	98.6087	98.6087
	60%	97.7411	98.4361	96.2126	97.04600
	70%	97.3380	97.9167	95.3101	96.4120
5	80%	96.1739	96.86960	93.1633	94.7826
	60%	96.5248	98.0886	94.6798	94.7871
	70%	96.8750	98.0324	94.9689	95.0231
6	80%	96.6957	97.7391	94.5095	94.6087

3.2.8 Leveraging K-means Clustering for Scalable Joint Performance Prediction

This section introduces a novel approach to address the challenge of dimensionality in the prediction of the performance of the joint network. Specifically, it details the use of K-means clustering to reduce the state space, enabling a more scalable and computationally efficient analysis. This approach simplifies the modeling of complex network behavior, focusing on the essential dynamics of accessibility and retainability.

3.2.8.1 The Challenge of Dimensionality in Joint Prediction

A key novelty of this research lies in its approach to tackle the inherent challenge of increasing dimensionality when jointly modeling multiple network performance indicators. While joint estimation, as discussed in Subsection Analysis of Mobile Network Performance, offers a more holistic view of network behavior by considering the combined states of accessibility and retainability, it leads to a significant expansion of the state space.

This expansion, often referred to as "state space explosion," presents a significant challenge: computational complexity. As you consider incorporating more mobile performance measures, such as integrity, the number of states grows exponentially. For instance, as mentioned in Subsection Challenges in Measurement and Analysis, adding just one more performance measure with, say, four states would increase the transition matrix size from 4x4 (for accessibility and retainability) to 4x4x4 = 64 states. This makes the model much harder to train, store, and analyze.

To quantify this, consider the number of elements in the transition matrix, which directly impacts computational complexity. For a Markov chain with (n) states, the transition matrix is an ($n \times n$) matrix, containing (n^2) elements. The computational complexity of calculating a single transition in a Markov Chain is $O(n^2)$. For predicting (k) steps into the future, the computational complexity becomes $O(k \cdot n^2)$.

1. For the initial joint model with accessibility and retainability, each with four states, the transition matrix is 4×4, containing $4^2 = 16$ elements.

2. When expanded to include a third performance measure (e.g., integrity) with four states, the transition matrix becomes 64×64, containing $64^2 = 4096$ elements.

This demonstrates that increasing the number of performance measures from two to three results in a 256-fold increase in the number of matrix elements (from 16 to 4096), illustrating the exponential growth in complexity. This exponential increase in the state space is a well-known issue with joint modeling using Markov chains, reinforcing the Scalability for Large Networks limitation discussed in Subsection Limitations of DTMCs. To address this, this research proposes K-means clustering as a key contribution and methodological step.

3.2.8.2 Dimensionality Reduction using K-means Clustering

To address the critical limitation of computational complexity discussed in the previous section, this research proposes a novel integration of K-means clustering to reduce the dimensionality of the joint state space. As mentioned in Subsection Challenges in Measurement and Analysis,

the computational demands increase significantly with each additional performance measure. For instance, adding a third measure with, say, four states would expand the transition matrix from 4×4 to $4 \times 4 \times 4 = 64$ states, making the model much harder to handle.

The application of clustering offers a promising approach to mitigate this issue. As highlighted in Subsection Analysis of Mobile Network Performance, clustering can group similar network conditions, effectively reducing the dimensionality of the state space. This strategy is a common and effective way to handle high-dimensional Markov models.

The core idea is to transform the high-dimensional random variable, representing the joint state of accessibility and retainability, into a new, lower-dimensional random variable. In this case, we aim to represent the network performance with only four states, where each state represents a cluster of similar network performance conditions. This is achieved by applying K-means clustering to the dataset of observed joint states. The input data for the clustering algorithm is the set of 16 combined states, derived from all possible combinations of the four accessibility states and the four retainability states. However, it's important to consider the trade-offs associated with clustering. The clustered states represent a combination of accessibility and retainability, potentially requiring new explanations and interpretations, as a new, aggregated measure is effectively created.

To determine the optimal number of clusters (K), we employed the Elbow method which plots the inertia (within-cluster sum of squares) against the number of clusters and observed a distinct "elbow" at K=4. This indicates that increasing the number of clusters beyond four yields diminishing returns in terms of reducing within-cluster variance, providing a data-driven justification for choosing four clusters as the reduced state space for our model.

3.2.8.3 Mapping Joint States to Clusters

The K-means algorithm partitions the 16 joint states into four clusters based on similarities in the combined performance metrics — namely, accessibility and retainability. This clustering enables a reduction in model complexity by mapping fine-grained joint performance states into fewer, representative categories. It includes the frequency and percentage of instances each original state is assigned to a specific cluster. This matrix visually places each of the 16 states into a 4×4 grid of accessibility (vertical) vs. retainability (horizontal). Clusters are color-coded, showing which joint states belong to each cluster. As illustrated, clustering tends to follow accessibility alignment more than retainability, as states with similar accessibility tend to fall in the same cluster despite retainability variations. This clustering reflects a non-uniform but interpretable behavior, driven predominantly by accessibility. The states with very low or

non-existent accessibility (e.g., states 0, 4, 8, 12) form a void or "non-existent" cluster, while others are grouped to capture meaningful behaviors.

3.2.8.4 Model Accuracy – Hourly Basis

The accuracy of the model was evaluated under different conditions, specifically considering the time granularity of the prediction: hourly accuracy and daily accuracy at a specific hour.

When the model operates on an hourly basis, Table 3.8 shows high prediction accuracy (>99.5%) across all clusters and training set sizes (60%, 70%, and 80%). However, when the model is trained to operate on a daily basis (using a selected 2-hour window), accuracy slightly drops, especially for training sizes of 60% and 70%. The accuracy improves to 100% only when 80% of data is used for training.

The observed drop in accuracy when operating on a daily basis is attributed to loss of temporal granularity. When using hourly data, the model has access to more fine-grained, diverse performance patterns, enabling it to better learn state transitions and temporal behaviors. Conversely, reducing the granularity to specific time slices (like two hours per day) results in less variability and fewer training examples per state, making it harder for the model to generalize accurately. Thus, hourly models are more reliable and suitable when high prediction precision is critical, while daily models may require larger training sets $\geq 80\%$ to achieve comparable performance.

3.2.8.5 Conclusion

This dissertation presented a novel approach for predicting mobile network accessibility and retainability using Markov chains. The model leverages two key advancements:

1. **Joint State Prediction:** A 16-state Markov chain was formulated to jointly estimate both accessibility and retainability in a single operation, improving efficiency compared to separate models.

2. **Spatially Aware Clustering:** To account for spatial variations in network performance across the 1530 analyzed cells, K-means clustering grouped the data into six clusters. This enabled the development of customized Markov chain models for each network segment, capturing the influence of location-specific factors.

By incorporating these innovations, the model achieved accurate predictions while maintaining computational efficiency. Additionally, the use of real-time data readily available through NMS ensures a cost-effective and practical solution.

Future research directions offer exciting possibilities to further enhance this approach:

- **Scalability for Large Networks:** Optimizing the current model for even larger networks with a vast number of cells requires exploring techniques that maintain computational efficiency while providing per-cell level information.
- **Broader Applicability:** Expanding the model's scope to encompass other network performance KPI, network types (e.g., beyond 3G), and diverse services would broaden its impact.
- **Hidden Markov Models:** Investigating the application of hidden Markov models could potentially offer deeper insights into network behavior by accounting for unobservable states.

3.2.9 Benefits and Limitations of DTMCs for Network Performance Analysis

DTMC offer a powerful tool for network performance analysis, particularly when considering accessibility and retainability. Their strengths lie in their simplicity, interpretability, and ability to capture state transitions effectively. However, limitations like the memory-less property assumption and scalability issues for complex networks must be considered.

3.2.9.1 Benefits of DTMC

- **Simplicity and Interpretability:** Compared to more complex models, DTMCs are relatively easy to understand and implement. The finite set of states and clear transition probabilities make them intuitive for network engineers and researchers. This ease of use allows for straightforward visualization of network behavior through state transition diagrams, facilitating clear communication and interpretation of results.
- **State Transition Capture:** A core strength of DTMC lies in their ability to model the probabilistic nature of transitions between network states. This allows for the prediction of future network states (e.g., predicting a potential drop to "Acceptable" accessibility from the current "Good" state) based on the current state and the historical transition probabilities. This probabilistic approach provides valuable insights into network dynamics and supports proactive network management strategies.

- **Computational Efficiency:** DTMCs are computationally efficient, making them suitable for real-time network performance analysis. This is particularly advantageous compared to more complex models that might require significant computational resources, potentially hindering real-time decision-making capabilities. The efficiency of DTMCs allows for faster analysis and quicker identification of potential network performance issues.

3.2.9.2 Limitations of DTMCs

- **Memory-less Property Assumption:** A fundamental assumption of DTMCs is the memory-less property. This principle states that the probability of transitioning to a future state depends only on the current state, and not on the history leading to that state. While this assumption often holds true for network performance at shorter timescales, it might not be suitable for capturing long-term dependencies in complex network behavior. For instance, a series of consecutive "Acceptable" accessibility states might be more likely to transition to a "Bad" state compared to a single instance of "Acceptable" accessibility. DTMCs might struggle to capture these nuanced historical dependencies.

- **Scalability for Large Networks:** As the number of cells in a network increases, the number of states in a DTMC model can grow exponentially. This can become computationally expensive and limit the model's scalability for very large networks. For instance, with four accessibility states and four retainability states, a model for a single cell would have 16 states. However, for a network of 1530 cells (as in this research), even with K-means clustering reducing the number of models, the computational demands could become significant for a model encompassing all possible state combinations.

DTMCs offer simplicity, interpretability, and computational efficiency, making them suitable for analyzing accessibility and retainability across 1530 cells. However, for larger or more complex networks, future work may explore models like HMMs, which address memory-less limitations and incorporate unobserved factors influencing network behavior.

3.3 Hidden Markov Model

3.3.1 Building Upon Markov Chains: Hidden States

HMMs, introduced in 1957, extend Markov chains by incorporating hidden states—underlying network conditions like signal strength or interference that cannot be directly observed. These

states influence measurable metrics such as accessibility or packet loss through emission probabilities, which define the likelihood of observing certain performance values given a hidden state.

An HMM is composed of two processes: a Markov chain governing the transitions between hidden states and an emission process linking these states to observable outputs. The primary goal is to infer hidden states from observed data, making HMM effective for analyzing temporal sequences. Widely applied in communication engineering and wireless networks, this research leverages HMM to analyze mobile network performance, particularly for detecting degradation and enabling QoS-to-QoE mapping, which is further explored in Chapter 4.

3.3.2 Components of HMM

Building upon the foundation of Markov chains, HMM introduce additional complexity by incorporating hidden states. While Markov chains model transitions between observable states, HMM deal with states that are not directly measurable. To understand this distinction and the key components of HMM, let's revisit the concept of states in a Markov chain context.

3.3.2.1 States in Markov Chains

In a traditional Markov chain, states represent the different conditions or situations a system can be in at a given time. These states are observable and directly measurable. For example, a Markov chain might model the weather patterns of a city, with states representing "sunny," "rainy," or "cloudy." Transitions between these states occur based on specific probabilities, forming the core principle of Markov chains.

3.3.2.2 Hidden States in HMM

HMM introduces the concept of hidden states. These hidden states represent underlying factors or conditions within a system that cannot be directly observed or measured. In the context of mobile networks, hidden states might represent factors like:

- Signal strength
- User load on the network (congestion)
- Channel interference

Despite being hidden, the influence of hidden states on the network can be observed through measurable network performance metrics. These metrics, known as emissions in HMM terminology, act as indirect indicators of the underlying hidden states. Examples of emissions in a mobile network context could include:

- Accessibility (percentage of successful connection attempts)
- Retainability (percentage of successfully maintained connections)
- Packet loss rate

3.3.2.3 Transition Probabilities

Similar to Markov chains, HMM utilizes transition probabilities. These probabilities define the likelihood of transitioning between hidden states over time. The key difference lies in the fact that these transitions are not directly observable. HMM relies on the observed sequence of emissions to infer the most likely sequence of hidden states and the transition probabilities between them.

3.3.2.4 Emission Probabilities

Emission probabilities play a crucial role in HMM. They define the likelihood of observing a specific emission value (e.g., a certain packet loss rate) given the current hidden state of the network. By analyzing the sequence of observed emissions and their corresponding emission probabilities, HMM can estimate the most likely sequence of hidden states that generated those emissions.

3.3.2.5 Linkage between HMM Components

These four components – hidden states, emissions, transition probabilities, and emission probabilities – work together to form the core framework of an HMM. While transition probabilities govern the evolution of hidden states (unobserved), emission probabilities link the hidden states to the observable emissions. By analyzing these relationships, HMM can provide valuable insights into the underlying network dynamics and potentially predict future behavior even with the presence of unobservable factors.

3.3.2.6 Double Stochastic Process Visualization

HMM can be effectively visualized as a double stochastic process. This concept highlights the presence of two interrelated transition mechanisms: one governing the evolution of hidden states (unobserved) and another governing the generation of observable emissions based on the underlying hidden states.

3.3.3 Visualizing HMM: A Graphical Representation

Visualizing HMM can be helpful in understanding the interplay between hidden states, transitions, and emissions. Due to the introduction of the time dimension alongside state transitions, representing the entire HMM process can be slightly more complex compared to traditional Markov chains.

3.3.3.1 State Transition Diagrams

A common approach to visualizing HMM is through state transition diagrams. These diagrams depict hidden states as circles or nodes, with arrows connecting them to represent transitions. The arrows are typically labeled with the corresponding transition probabilities. This visual representation provides a clear understanding of how the hidden states evolve over time within an HMM.

3.3.3.2 Trellis Diagrams

For more advanced visualization, trellis diagrams can be employed. These diagrams represent both hidden states and emissions along a timeline. Horizontal arrows depict transitions between hidden states at different time steps, while emission symbols are placed alongside the corresponding hidden states at each time step. Trellis diagrams offer a more comprehensive view of the relationship between hidden states, transitions, and emissions throughout an observation sequence.

Assume that we have as system where with N hidden and M emission states. The trellis diagram shown in Figure 3.4 is widely used in the literature to represent HMM in a two-dimensional plane, where the horizontal arrows indicate the transition in the hidden states at different times while the vertical arrows show the relationship between the hidden and emission states at a given time. In the figure, the set $S^g(t) = [S_1(t), S_2(t), \ldots, S_N(t)]$ is a $1 \times N$ dimensional vector representing all the hidden states at time t and $O^g(t) = [O_1(t), O_2(t), \ldots, O_M(t)]$ is a $1 \times M$ vector representing all the emissions at the same time t [11] [14] [150].

Moreover,is the observation interval, Note that in discrete HMM, the time index t is assumed to be discrete. With this notation, $S_i(t)$ for $1 \leq i \leq N$ and $O_k(t)$ for $1 \leq k \leq M$ designate being in hidden state S_i and observable state O_k at time t, respectively.

The choice of visualization technique depends on the complexity of your HMM and the intended audience. State transition diagrams are generally easier to understand for beginners, while trellis diagrams provide a more detailed view.

3.3.3.3 Three Dimension (3D) Representations

While not as widely used as the previous methods, 3D visualizations of HMM can also be employed. To better explain the HMM by taking all transitions in the hidden states and emissions over an observation interval, the three-dimensional representation is shown in Figure 3.5.

- **Hidden states:** Are represented in the horizontal (i.e., $x - y$) plane, where the depth of the plan in the y-axis indicates all the N states, and the plane's width along the x-axis indicates the states transitions in time t. As in the Markov chain example in Figure 3.5, the solid lines represent the hidden states transitions at different times. The transitions are from each state to every other state in the next instant, and the state transition matrix captures these transitions.

- **Emission states:** The M observable or emission states are visible in Figure 3.4 when viewed along the $y - z$ plane. In this plane, the dotted lines show the interactions among the hidden and emission states at a given time t. Note that each emission state is influenced (in a probability sense) by all hidden states. Later on, we will see that this interaction is captured via the *emission matrix*.

- **Time dimension:** The different transitions in time, both for the hidden states and emissions, are shown in the $x - z$ plan. The widely used trellis diagram representation shown in Figure 3.4 is the result of viewing Figure 3.5 along $x - y$ plan.

In summary, Figure 3.5 shows the transitions observed in the hidden states, interaction among the hidden and emission states, and the transitions of both states at different time instants.

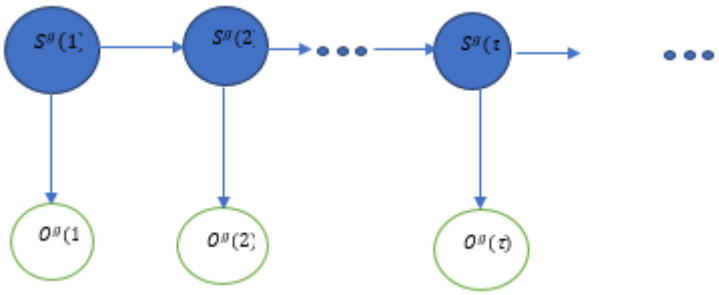


Figure 3.4: A trellis diagram representation of the first-order HMM.

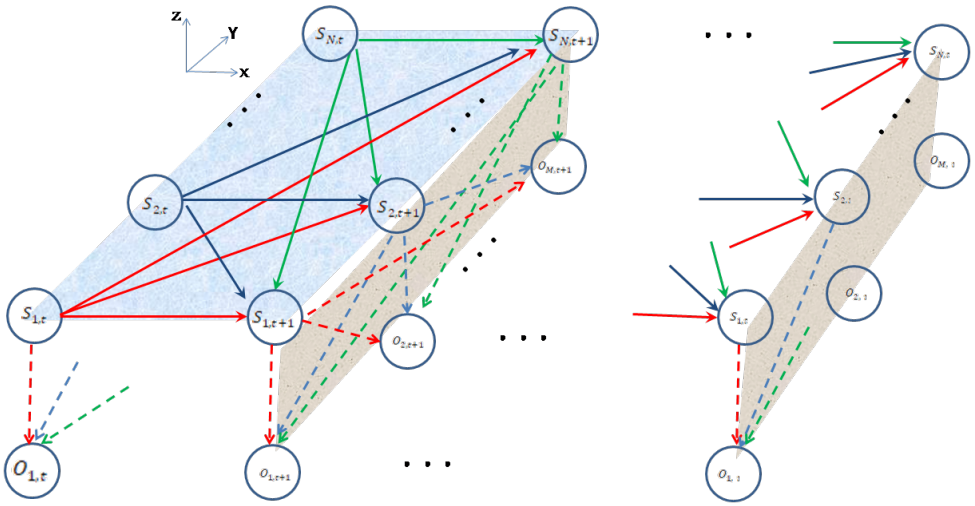


Figure 3.5: Three dimension view of HMM process.

3.3.4 Mathematical Representation of HMMs

To formally describe the dynamics of an HMM, we can leverage a set of mathematical objects including the transition matrix (A), which captures the probabilities of transitioning between hidden states, the emission matrix (B), which defines the likelihood of observing specific emissions given the underlying hidden state, and the concept of a joint probability distribution, which allows us to calculate the combined probability of a specific sequence of hidden states and their corresponding emissions.

3.3.4.1 Transition Matrix in HMM

Given the hidden state $S^g(t) = [S_1(t), S_2(t), \ldots, S_N(t)]$ at time t. Let the transition probability which is the probability that the system moves to state S_j (t+1) from state $S_i(t)$ is designated

by $a_{i,j}$ for $1 \leq i, j \leq N$ and written as [14] [16] [157] :

$$a_{i,j} = P(S_j(t+1)|S_i(t)), \tag{3.17}$$

and,

$$\sum_{j=1}^{N} a_{i,j} = 1 \tag{3.18}$$

Equation (3.17) donates the probability that, starting from state i, in the next transition the system will land in any of the N states, including staying in the same state. The collection of the probabilities, $a_{i,j}$, for all possible transitions forms the transition probability matrix $A = [a_{i,j}] \in R^{N \times N}$ [14].

3.3.4.2 Emissions Matrix in HMM

The *emission probability* is the probability that a hidden state $S_i(t)$ emits an observation $O_k(t)$. Mathematically, this probability is written as:

$$b_{i,k} = P(O_k(t)|S_i(t)) \tag{3.19}$$

and,

$$\sum_{k=1}^{M} b_{i,k} = 1 \tag{3.20}$$

The collection of the probabilities, $b_{i,k}$ given by $B = [b_{i,k}] \in R^{NXM}$ forms the *emission probability matrix*. Now let a one dimensional vector $\pi(t) = [\pi_1(t), \ldots, \pi_N(t)] \in R^{1XN}$ be the hidden states distribution vector at time t and $\pi_i(t)$, for $1 \leq i \leq N$, is the likelihood of the state $S_i(t)$ and

$$\sum_{i=1}^{N} \pi_i(t) = 1 \tag{3.21}$$

to satisfy the probability properties that the hidden system will be in any of the N hidden states at the initial time t. At a reference/initial time $t = 0$ we have:

$$\pi_i(0) = P(S_i(0)) \tag{3.22a}$$

And

$$\pi(0) = [\pi_1(0), \ldots, \pi_N(0)] = \pi \tag{3.22b}$$

Note that π designate the hidden states initial probability distribution. Using the above

formulas, mathematically, HMM model can be presented using five parameters as set of:

$$\lambda = (N, M, \pi(t), A, B) \tag{3.23}$$

Where

- λ denotes the HMM;
- N is the number of hidden states in the model;
- M is the number of observation symbols or emission states;
- $\pi = \pi_i(0)$ is the initial state of probability distribution of the hidden states;
- $A = [a_{i,j}]$ is the state transition probability matrix; and finally;
- $B = [b_{i,k}]$ is the emission probability matrix.

Usually, the HMM model is expressed as $\lambda = (\pi(0), A, B)$ dropping both N and M. While parameters of interest in the standard Markov chain are *transition matrix*, P, and *initial-state probabilities*, U, HMM requires emission probability matrix B, as a third and additional model parameter.

Now assume that a sequence of hidden states is generated by the HMM, where the sequence is mainly determined by the state transition probability matrix A. Introducing a new notation, for a finite observation sequence $S^g := s^g(1), s^g(2), \ldots, s^g(\tau)$ where τ is any fixed number indicating the observation window, and $s^g(t) = [S_1(t), S_2(t), \ldots, S_N(t)]$ is a vector containing all states at time t as defined above [13] [148]. Likewise, a sequence of emissions (observations) is generated by the HMM according to the emission probability matrix B. The generated sequence of emissions are designated by $O^g := o^g(1), o^g(2), \ldots, o^g(\tau)$, where $o^g(t) = [O_1(t), O_2(t), \ldots, O_M(t)]$.

The transition probability matrix from $s^g(t)\ to\ s^g(t+1)$ can be denoted by a modified $a_{i,j}$ as:

$$A = a_{s^g(t-1),s^g(t)} := P(S^g(t+1)|S^g(t)) \tag{3.24}$$

The initial state distribution vector is,

$$\pi_i(t) = \pi_{s^g(i)} := P(s^g(i)) = \pi(0) \tag{3.25}$$

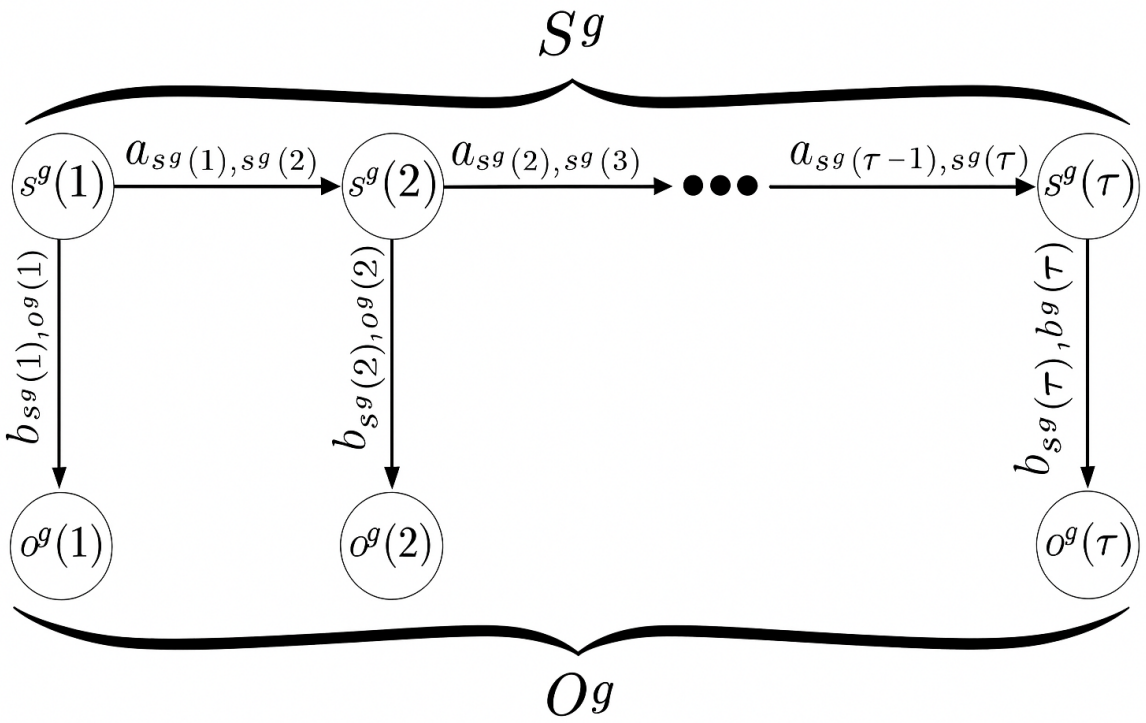


Figure 3.6: A Hidden Markov Model [159].

The probability of state vector $S^g(t)$ emitting the observation vector $O^g(t)$ is given by the vector $b_{(}s^g(t)o^g(t))$,

$$B = b_{s^g(t)o^g(t)} = P(o^g(t)|s^g(t)) \tag{3.26}$$

Figure 3.6 below depicts the structure of HMM of model [24].

3.3.4.3 Joint Distribution

In HMM, one general problem of interest for an observation time window of τ is to compute the joint distribution of a sequence of hidden states and emissions. Based on the trellis diagram representation shown in Figure 3.4 and the notations from Equations (3.24) - (3.26), the joint probability distribution is given by:

$$P(S^g, O^g) = P(s^g(1), \ldots, s^g(\tau), o^g(1), \ldots, o^g(\tau)) \tag{3.27a}$$

By applying Chain rule, Equation (3.27a) reduces to:

$$P(S^g, O^g) = P(s^g(1)) \cdot P(o^g(1)|s^g(1)) \prod_{k=2}^{\tau} P(s^g(k)|s^g(k-1))P(o^g(k)|s^g(k)) =$$
$$P(s^g(1)) \cdot \prod_{k=2}^{\tau} P(s^g(k)|s^g(k-1)) \prod_{k=1}^{\tau} P(o^g(k)|s^g(k)) = \tag{3.27b}$$
$$\pi_{s^g(1)} \cdot b_{s^g(1)o^g(1)} \prod_{k=2}^{\tau} a_{s^g(k-1),s^g(k)} b_(s^g(k)o^g(k))$$

The first term $\pi_(s^g(1))$ is the initial distribution vector, $a_{s^g(k-1),s^g(k)}$ is a matrix showing the transitions across the hidden states, and $b_(s^g(k)o^g(k))$is the emission probability matrix. The emission probability can generally be a discrete finite set, countably infinite with some distribution, say Poisson Distribution or Gaussian for the real-valued process.

3.3.5 Algorithms for HMM

3.3.5.1 Forward-Backward Algorithm

The forward-backward algorithm is a dynamic programming algorithm used by Richard Bellman and has very important applications in HMM. Dynamic programming is a broad class of algorithms that iteratively solves problems. The forward-backward algorithm assumes that we know the emission probability $P(O^g(k)|S^g(k))$, the transition probability $P(S^g(k)|S^g(k-1))$ and initial probability distribution $P(S^g(1))$.

Given the observation sequence $O^g := o^g(1), \ldots, o^g(\tau)$ the interest here is to compute the probability $P(S^g(k)|O^g(1), \ldots, O^g(\tau))$, where $1 \leq \tau \leq k$. The forward-backward algorithm splits the computation into the forward and backward parts and applies marginal distribution, and posterior distribution computations.

The probability computation is broken down into two parts as below.

- The forward algorithm part computes the joint probability $P(S^g(k)|O^g(1), \ldots, O^g(k))$ for all possible values of k in the range $1 \leq k \leq \tau$. (See Figure 3.7).

- The backward algorithm computes $P(O^g(k+1), \ldots, O^g(\tau)|S^g(k))$ for all possible values of k in the range $1 \leq k \leq \tau$. (See Figure 3.7).

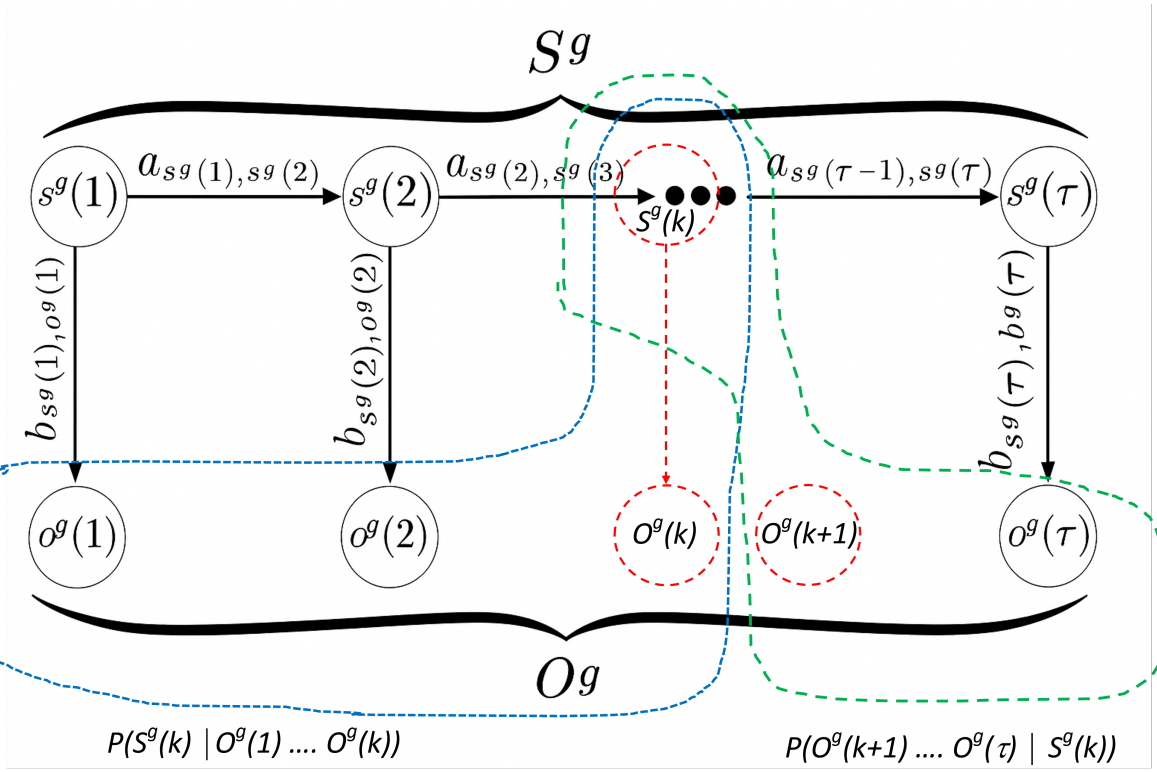


Figure 3.7: Forward-backward algorithm [160].

With the forward and backward parts defined, computing the probability

$$P(S^g(k)|O^g(1),\ldots,O^g(\tau)) \propto P(S^g(k),O^g(1),\ldots,O^g(\tau)) = \\ P(O^g(k+1),\ldots,O^g(\tau)|S^g(k),O^g(1),\ldots,O^g(k)) \cdot P(S^g(k),O^g(1),\ldots,O^g(k)) \tag{3.28}$$

Equation (3.28) results from applying the probability rule $P(A,B) = P(A|B)P(B)$ and shows the proportionality between the conditional and joint probabilities. If we further assume that the probability of the event $O^g(k+1),\ldots,O^g(\tau)$ and $O^g(1),\ldots,O^g(k)$ are conditionally independent, i.e., both are separated, as long as $S^g(k)$ is given (see Figure 3.7). This way, Equation (3.28) reduces to the form:

$$P(S^g(k)|O^g(1),\ldots,O^g(\tau)) \propto \underbrace{P(O^g(k+1),\ldots,O^g(\tau)|S^g(k))}_{Backward\ part} \cdot \underbrace{P(S^g(k),O^g(1),\ldots,O^g(k))}_{Forward\ part} \tag{3.29}$$

We note from Equation (3.29) that by multiplying the forward and backward components, we get something proportional to the marginal distribution of one of the hidden states. Since, $S^g(k)$ Is a finite set, we can normalize Equation (3.29) by summing over the finite set, get the normalizing part and get the distribution $P(O^g(k)$.

With knowledge of the probability in Equation (3.28), we can address problems like:

- Inference - $P(S^g(k) \neq S^g(k+1)|O^g(1),\ldots,O^g(\tau))$. This is called the *change detection* problem.
- Learning or estimating parameters, i.e., emission probability $P(O^g(k)|S^g(k))$The transition probability $P(S^g(k)|S^g(k-1))$ and initial probability distribution $P(S^g(1))$. Baum-Welch computes the forward-backward coupled with Expectation Maximization.
- Decoding or sampling from posterior distribution – given observations, we can sample from the hidden states that may explain the observation. Viterbi Algorithm is used for that purpose.

There are four phases in implementing the forward-backward algorithm: initial phase, forward phase, back-ward phase, and update phase. Each of them is explained in the next subsections.

Initial phase In the initial phase, the content of the parameter matrices A, B, π are initialized, and it could be done randomly if there is no prior knowledge about them.

Forward Algorithm for HMM The goal of the forward algorithm is to compute $P(S^g(k), O^g(1), \ldots, O^g(k))$ for all k in the range $1 \leq k \leq \tau$ and $S^g(t) = [S_1(t), S_2(t), \ldots, S_N(t)]$. It is assumed that the transition and emission probabilities are known. Mathematically, we can define a recursive alpha function as [13]

$$\alpha_k(S^g(k)) = \sum_{S_1(k-1)}^{S_N(k-1)} P(S^g(k), S^g(k-1), O^g(1), \ldots, O^g(k)) \tag{3.30}$$

$$\begin{aligned} P(S^g(k), O^g(1), \ldots, O^g(k)) = \sum_{S_1(t)}^{S_N(t)} P(O^g(k)|S^g(k), S^g(k-1), O^g(1), \ldots, O^g(k-1)) \cdot \\ P(S^g(k)|S^g(k-1), O^g(1), \ldots, O^g(k-1)) \cdot \underbrace{P(S^g(k-1), O^g(1), \ldots, O^g(k-1))}_{\textit{Looks like the Forward Part}} \end{aligned} \tag{3.31}$$

Where

$$\alpha_k(S^g(k)) = P(S^g(k), O^g(1), \ldots, O^g(k))$$

Marginalizing and applying the Markovian property Equation (3.30) reduces to

$$\begin{aligned} P(S^g(k), O^g(1), \ldots, O^g(k)) = \sum_{S_1(k-1)}^{S_N(k-1)} \{P(O^g(k)|S^g(k), S^g(k-1), O^g(1), \ldots, O^g(k-1)) \cdot \\ P(S^g(k)|S^g(k-1), O^g(1), \ldots, O^g(k-1)) \cdot P(S^g(k-1), O^g(1), \ldots, O^g(k-1))\} \end{aligned} \tag{3.32}$$

Given $S^g(k)$ We can claim that $O^g(k)$ is conditionally independent on $S^g(k-1), O^g(1), \ldots, O^g(k-1)$, as $S^g(k)$ *is the only "path"* to $O^g(k)$. Likewise, given $S^g(k-1)$, $S^g(k)$ is conditionally independent on $O^g(1), \ldots, O^g(k-1)$. That way, the joint probability in Equation (3.31) becomes:

$$\alpha_k(S^g(k)) = \sum_{S_1(k-1)}^{S_N(k-1)} \{\underbrace{P(O^g(k)|S^g(k))}_{\substack{\text{Emission} \\ \text{Probability}}} \cdot \underbrace{P(S^g(k)|S^g(k-1))}_{\substack{\text{Transition} \\ \text{Probability}}} \cdot \alpha_{k-1}(S^g(k-1))\} \tag{3.33}$$

Where $\alpha_k(S^g(k))$ is the recursion and equals to $P(S^g(k), O^g(1), \ldots, O^g(k))$. Likewise $\alpha_(k-1)(S^g(k-1)$ designates $P(S^g(k-1), O^g(1), \ldots, O^g(k-1))$. We need the following starting alpha to begin the recursion. Note that at the initial step:

$$\alpha_1(S^g(1)) = P(S^g(1), O^g(1)) \underbrace{((P(S^g(1))}_{\substack{\text{Initial} \\ \text{Distribution}}} \cdot \underbrace{P(O^g(1)|S^g(1))}_{\substack{\text{Emission} \\ \text{Probability}}} \tag{3.34}$$

Both the initial distribution and emission probability are known. Equations (3.33) and (3.34) help us to compute the α_i in a recursive way. A few points can be taken by looking at Equation (3.33) summarized below [13].

1. The alpha function is defined as the joint probability of the observed data up to time k and the state at time k

2. It is a recursive function because the alpha function appears in the first term of the equation's right-hand side Right-hand side (R.H.S.), meaning that the previous alpha is reused in the next calculation. This is also why it is called the forward phase.

3. The second term of the R.H.S. is the state transition probability from A, while the last is the emission probability from B.

4. The R.H.S. is summed over all possible states at time $k-1$.

It should be pointed out that each alpha contains the information from the observed data up to time k. To get the next alpha, we only need to reuse the current alpha and add information about the transition to the next state and the next observed variable. This recursive behavior saves computations of getting the next alpha by freeing us from looking through the past observed data every time.

This way, the order of the computation is reduced significantly, as it *reuses* the results of earlier computations because of the recursive nature of the computation. That is why the forward algorithm falls into the dynamic programming category. Using a recursive approach, dynamic programming involves breaking down a complex problem into simpler sub-problems. It includes the steps of *Initialization, Recursion*, and *Termination* to find the sequence of the hidden states.

Backward Algorithm for HMM Here again, the goal of the backward algorithm is to compute

$$\beta_k(S^g(k)) = P(O^g(k+1), \ldots, O^g(\tau)|S^g(k)) \tag{3.35}$$

for all k in the range $1 \leq k \leq \tau$ and $S^g(t) = [S_1(t), S_2(t), \ldots, S_N(t)]$. It also assumes that the transition and emission probability distributions are known.

$$\beta_k(S^g(k)) = \sum_{S_1(k+1)}^{S_N(k+1)} P(O^g(k+1), \ldots, O^g(\tau), S^g(k+1)|S^g(k)) \tag{3.36}$$

Where

$$\beta_k(S^g(k)) = P(O^g(k+1), \ldots, O^g(\tau)|S^g(k)) \tag{3.37}$$

As we did for the forward part, if we factor Equation (3.37) by applying the Markov property and marginalization, we get:

$$\begin{aligned}\beta_k(S^g(k)) = \sum_{S_1(k+1)}^{S_N(k+1)} P(O^g(k+2), \ldots, O^g(\tau)|S^g(k+1)) \cdot \\ \underbrace{P(O^g(k+1)|S^g(k+1))}_{\substack{\text{Emission} \\ \text{Probability}}} \cdot \underbrace{P(S^g(k+1)|S^g(k))}_{\substack{\text{Transition} \\ \text{Probability}}}\end{aligned} \tag{3.38}$$

Note that $\beta_k(S^g(k)) = P(O^g(k+2), \ldots, O^g(\tau)|S^g(k+1))$ so that the recursion in Equation (3.38) becomes:

$$\beta_k(S^g(k)) = \sum_{S_1(k+1)}^{S_N(k+1)} \beta_{k+1}(S^g(k+1)) \cdot P(O^g(k+1)|S^g(k+1)) \cdot P(S^g(k+1)|S^g(k)) \tag{3.39}$$

This is valid for $1 \leq k \leq \tau - 1$. Similar points could be made here:

1. The beta function is defined as the conditional probability of the observed data from time $k+1$ given the state at time k.
2. It is a recursive function because the beta function appears in the first term of the right-hand side of the equation, meaning that the next beta is reused in calculating the current one. This is also why it is called a backward phase.
3. The second term of the R.H.S. is the state transition probability from A, while the last is the emission probability from B.
4. The R.H.S. is summed over all possible states at time $k+1$.

Again, we need the ending beta to start the recursion.

$$\beta_\tau(S^g(\tau)) = 1 \tag{3.40}$$

The computational complexity is the same as the forward algorithm. Some points to note about alpha and beta [13] [161].

1. Firstly, as mentioned, they are both recursive functions, meaning we could reuse the previous answer as the input for the next answer.

2. Secondly, the formula in the forward phase is very useful. Suppose you have a set of well-trained transition and emission parameters, given that your problem is finding out the mysterious hidden truth from observed data in real-time. Then you actually could do it like this! When you get one data point (data point p), you could put it into the formula, giving you the probability distribution of the associated hidden state and from which you could pick the most probable one as your answer. And the story does not stop here. As you get the next data point (data point q), and you put it again into the formula, it will give you another probability distribution for you to pick the best choice, but this is based on the data point, not only q and the transition and emission parameters, but also the data point p. Such use of the formula is called *filtering*.

3. Thirdly, and continuing the above discussion, suppose you collected many data points already, and because you know that the earlier the data point, the less observed data the choice of your answer is based on. Therefore you would like to improve that by somehow 'injecting' information from the later data into the earlier ones. This is where the backward formula comes into play. Such use of the procedure is called *smoothing*.

4. Fourthly, this is about the combination of the last two paragraphs. With the help of the alpha and the beta formula, one could determine the probability distribution of the state variable at any time k given the whole sequence of observed data. This could also be understood mathematically.

5. Lastly, the alpha and the beta function results are useful in the update phase.

3.3.5.2 Viterbi Algorithm

Andrew Viterbi conceived the Viterbi algorithm in 1967 as a decoding algorithm for convolution codes over noisy digital communication links. It is used for finding the most likely sequence of hidden states, called the Viterbi path, that results in a sequence of observed events [162]. The algorithm has found universal application in decoding convolution codes, speech recognition applications, keyword spotting, computational linguistics, and bio-informatics [161]. For example, the acoustic signal is the observed sequence of events in certain speech-to-text recognition devices. A text string is the "hidden cause" of the acoustic signal. The Viterbi algorithm finds the most likely string of text given the acoustic signal [163].

The Viterbi Algorithm uses dynamic programming to compute the most probable sequence of hidden states for a given sequence of observed states.
Like the forward-backward algorithms, the Viterbi algorithm starts by assuming that the initial,

transition, and emission probability distributions are all known. The goal of the algorithm is, given a sequence of observation $O^g(1), \ldots, O^g(\tau)$ and the model λ, computes the most likely (or "correct") hidden sequences, $S^g(1), \ldots, S^g(\tau)$That lead to (or best explains) the observation in some meaningful sense. This is called the decoding or, most likely, explanation problem [150] [155] [16].
When put mathematically, the algorithm looks like this:

$$S^g(1), \ldots, S^g(\tau) = arg \max_{\forall S^g(k)} P(S^g(1), \ldots, S^g(\tau) | O^g(1), \ldots, O^g(\tau)) \tag{3.41}$$

Note that k in the range $1 \leq k \leq \tau$ and $S^g(t) = [S_1(t), S_2(t), \ldots, S_N(t)]$. With the above property and the probability property used earlier, maximizing the conditional probability in Equation (3.41) is equivalent to maximizing the joint distribution.

$$arg \max_{\forall S^g(k)} P(S^g(1), \ldots, S^g(\tau), O^g(1), \ldots, O^g(\tau)) \tag{3.42}$$

The maximization in Equation (3.33) is simplified by taking the following properties of maximization and minimization.
If $f(a) \geq 0$for $\forall$ a and $g(a, b) \geq 0$ for $\forall$ a,b, then,

$$\max_{a,b} f(a)g(a, b) = \max_{a}[f(a) \max_{b} g(a, b)] \tag{3.43}$$

As in the forward-backward algorithm, finding some recursion by utilizing the Markov property helps simplify the maximization in Equation (3.43). Combining Equations (3.39) and (3.43), we get:

$$\beta_k(S^g(k)) = \max_{S^g(1),\ldots,S^g(k-1)} P(S^g(1), \ldots, S^g(k), O^g(1), \ldots, O^g(k)) = \max_{S^g(1),\ldots,S^g(k-1)} P(O^g(k)|S^g(k)) P(S^g(k)|S^g(k-1)) P(S^g(1), \ldots, S^g(k-1), O^g(1), \ldots, O^g(k-1)) \tag{3.44}$$

With some simplification, we get:

$$\beta_k(S^g(k)) = \max_{S^g(1),\ldots,S^g(k-1)} \left(\underbrace{P(O^g(k)|S^g(k))}_{\substack{\text{Emission} \\ \text{Probability}}} \underbrace{P(S^g(k)|S^g(k-1))}_{\substack{\text{Transition} \\ \text{Probability}}} \beta_{k-1}(S^g(k-1)) \right) \tag{3.45}$$

Where

$$\beta_(k-1)(S^g(k-1)) = \max_{S^g(1),\ldots,S^g(k-2)} P(S^g(1),\ldots,S^g(k-1),O^g(1),\ldots,O^g(k-1) \quad (3.46)$$

And

$$\beta_1(S^g(1)) = P(S^g(1),O^g(1)) = P(S^g(1))P(O^g(1)|S^g(1)) \quad (3.47)$$

This is a recursion in the forward direction, where the problem of finding a maximum path decomposes into sub problem. The next question is how to get the hidden sequence (i.e., argument) that maximizes the sequence. This requires navigating through all the possible paths systematically. The approach is, at each stage of the transition, say $k-1$, and for each hidden state $S_i(k-1)$ and $1 \leq i \leq N$, from all possible paths, we have to find the path that maximizes the sequence. At the end of this stage, the number of paths we have is as much as the number of hidden states N. Then, for the next transition at each state $S_i(k)$, we use the earlier identified maximizing paths to compute the optimal path to this state. This process continues for all the hidden states until we reach the end of the observation interval, τ.

Based on the above explanation, we can see that the Viterbi algorithm also has forward and *backward* phases. In the *forward* phase, instead of the sum-product algorithm, it utilizes the max-product algorithm. The backward phase recovers the most probable path through the trellis diagram using a trace back procedure, propagating the most likely state at time k back in time to recursively find the most likely sequence between times $1{:}k$.

The trellis diagram of the HMM best explains the steps mentioned above. The trellis diagram shows how each state in the model at a one-time step connects to the states in the next time step. The algorithm computes the shortest path through the trellis diagram of the HMM.

3.3.6 Basic Problems in HMM

Once we have an HMM and know the key parameters, problems of interest must be solved for the model to be useful in real-world applications. The below subsections explain the issues and the corresponding efficient algorithms to solve them.

3.3.6.1 Evaluation Problem in HMM

The evaluation problem can be stated as follows: given a known observation sequence $O^g := o^g(t),\ldots,o^g(t+\tau)$ and model $\lambda = (\pi, A, B)$, the goal is to evaluate the conditional probability distribution of observing this known sequence, designated as $P_r(O^g/\lambda)$. This is a part of the inference or evaluation problem and can be solved using forward filtering [59].

A naïve way to evaluate the probability is by summing up the possibilities of all hidden state sequences as follows.

$$P(O^g(1),\ldots,O^g(\tau)) = \sum_{S^g(1),\ldots,S^g(\tau)} P(S^g(1))P(O^g(1)|S^g(1))\ldots P(S^g(\tau)|S^g(\tau-1))P(O^g(\tau)|S^g(\tau)) \tag{3.48}$$

This will take us exponential time, and hence, not so good. However, we can take advantage of the structure of the model to calculate this probability in polynomial time.

3.3.6.2 Learning Problem in HMM

One interesting question to address is estimating or adjusting the model parameters $\lambda = (\pi, A, B)$, i.e., the initial probabilities, transition probabilities, and output probabilities to maximize the probability of observing the sequence $O^g := o^g(t),\ldots,o^g(t+\tau)$, written as $P_r(O^g/\lambda)$. *This is called the learning problem* [59].

3.3.6.3 Other HMM Problems

There are four interesting queries in HMM: monitoring, prediction, hindsight, and most likely explanation. Each of them is explained below.

Monitoring (Filtering): In HMM, monitoring is applied when an application area requires us to know the distribution over the current hidden state, given past and current observations. Robot localization and patient monitoring are good examples of monitoring in HMM. Mathematically, given a model $\lambda = (\pi, A, B)$ and observation sequence $O^g := o^g(1),\ldots,o^g(t)$ The monitoring (likelihood) problem is stated as:

$$P(S^g(t)|O^g(1),\ldots,O^g(t)) \tag{3.49}$$

The forward algorithm is used to calculate the likelihood.

Prediction: In HMM, prediction (estimation) can be made to uncover the hidden part of the model by identifying the state after several time steps (future state) given the past or current observations. Mathematically, given a model $\lambda = (\pi, A, B)$ and the state sequence $S^g := S^g(1),\ldots,S^g(t)$ The prediction (estimation) problem is stated as:

$$P(O^g(t+k)|S^g(1),\ldots,S^g(t)) \tag{3.50}$$

Where k is the prediction in the future time stamp, the estimation problem HMM is applied in predicting the weather, stock market, network channel, and mobility, estimating the QoE in the heterogeneous access network (HAN.), and so on [164]. The prediction of the output sequence is calculated recursively as;

- Recursive computation and using marginalization:

$$P(O^g(t+k)|S^g(1),\ldots,S^g(t)) = \sum_{O^g(t+k-1)} P(O^g(t+k),O^g(t+k-1)|S^g(1),\ldots,S^g(t)) \quad (3.51)$$

- Using Chain rule, Equation (3.51) is given by;

$$P(O^g(t+k)|S^g(1),\ldots,S^g(t)) = \sum_{O^g(t+k-1)} P(O^g(t+k)|O^g(t+k-1),S^g(1),\ldots, S^g(t))P(O^g(t+k-1)|S^g(1),\ldots,S^g(t)) \quad (3.52)$$

- By using the conditional independency theorem

$$P(O^g(t+k)|S^g(1),\ldots,S^g(t)) = \sum_{O^g(t+k-1)} P(O^g(t+k)|O^g(t+k-1))P(O^g(t+k-1))|S^g(1),\ldots,S^g(t)) \quad (3.53)$$

For prediction, the linear complexity for the forward algorithm is in $t+k$.

HINDSIGHT (SMOOTHING): In HMM, smoothing is used for an application like speech recognitions or other delayed activities that requires an update of the state distribution of past time steps, given new data, and denoted as $P(o_k|s_(1:t))$ for $k < t$.

Given observations sequence $O = O_(1,)O_(2,)\ldots,O_(T,)$, the model λ , to find the most probable hidden state sequence $I = i_1, i_2, \ldots, i_T$. The Viterbi algorithm can be solved [150] [14]. A related problem can be solved by using Forward-Backward Algorithm to calculate the probability of being in a state $s_(1:t)$ given the observation probability $(s_(1:t)|o =_(t+k))$.

3.3.7 Cell Degradation Prediction Modeling with HMMs

3.3.7.1 Cell Degradation in Mobile Networks

Ensuring optimal network performance in today's ever-growing mobile landscape is a critical challenge for MNO, 4G, LTE, and 5G networks are the backbone of modern mobile commu-

nication, enabling high-speed data services for a vast array of applications. However, as the number of connected devices explodes, these networks face increasing strain on their resources, leading to a phenomenon known as cell degradation. Cell degradation occurs when a cell's ability to handle traffic falls significantly below its expected performance. This decline can be attributed to various factors, including:

- **Increased User Traffic:** As more users connect to the network, the demand for resources like bandwidth and signal strength increases, potentially leading to congestion and performance degradation.
- **Interference:** Interference from external sources or neighboring cells can disrupt signal transmission and reception, impacting cell performance.
- **Hardware Failures:** Equipment malfunctions within the cell infrastructure can lead to degraded performance.
- **Environmental Factors:** Environmental changes such as weather conditions or obstructions can affect signal propagation and contribute to cell degradation.

The impact of cell degradation goes beyond network efficiency. It directly affects user experience (QoE) by causing issues like reduced data rates, increased call drops, and longer latencies. To address these challenges and ensure network performance, traditional approaches utilize network monitoring based on KPI. While this practice provides valuable insights, it often relies on static thresholds and reactive strategies, making it less effective in predicting and proactively managing cell degradation.

3.3.7.2 Cell Degradation Prediction Techniques

Maintaining optimal network performance in today's ever-growing mobile landscape is a critical challenge for MNOs. Cell degradation, a phenomenon where a cell's ability to handle traffic falls below its expected level, significantly impacts network performance and user experience (QoE). Predicting cell degradation proactively enables MNO to take preventive measures and ensure network health. This literature review explores existing techniques for cell degradation prediction, highlighting their strengths and limitations, and lays the foundation for our proposed approach using HMM.

- **Markov Chain-based approaches:** [165] utilizes a DTMC model with three accessibility states (high, acceptable, low/degraded) trained on historical KPI data. This approach

demonstrates potential for root cause analysis, degradation detection, and preventive maintenance scheduling within LTE/ System Architecture Evolution (SAE) networks. However, the limited number of degradation states might restrict its ability to capture the finer details of cell degradation severity.

- **Hidden Markov Models (HMMs):** Reference [166] showcases the effectiveness of HMMs for cell outage detection in 5G Heterogeneous Networks (HetNets). Their model, achieving high accuracy, classifies cells into four categories: healthy, degraded, degraded further (crippled), and completely inoperable (catatonic). While promising, it is important to note that the study relies on synthetic data, which may not fully represent real-world network conditions and all types of cell degradation (e.g., hardware malfunctions).

- **Machine Learning Techniques:** RNNss: RNNs are powerful tools for handling sequential data, making them well-suited for analyzing network traffic patterns and predicting degradation events. They can capture long-term dependencies in the data and learn complex relationships between KPI. However, RNNs can be computationally expensive to train and require large amounts of data. SVMs: SVM can be used to classify cell states based on extracted features from KPI. They are known for their good generalization abilities and can be effective with limited data [27]. However, SVM might require careful feature engineering to achieve optimal performance. Ensemble Learning Methods: These methods combine predictions from multiple machine learning algorithms, often leading to more robust and accurate results. Techniques like Random Forests or Gradient Boosting can be applied to cell degradation prediction, leveraging the strengths of different algorithms to achieve better prediction accuracy [157].

- **Deep Learning Approaches:** CNNss: CNNs are particularly effective in analyzing spatial patterns, which can be relevant for cell degradation prediction when considering neighboring cells and their influence [14]. They can learn spatial relationships between KPI from different cell locations and potentially improve prediction accuracy. However, CNNs require even more data and computational resources compared to RNNs.

- **Other Techniques:** Correlation-based Approach: [21] proposes a correlation analysis method for detecting cell degradation in LTE networks. This approach analyzes inter-cell relationships between KPI and achieves a success rate in identifying degradation cases. However, its effectiveness in complex scenarios and adaptability to different network configurations remain to be fully explored.

- **Expert Analysis:** While not the sole focus, some studies, such as [145], integrate expert knowledge alongside other techniques like machine learning algorithms for cell degradation classification.

- **Key Observations and Research Gap:** The reviewed literature highlights the potential and limitations of various techniques for addressing network degradation:
  - HMMs, with their ability to capture the dynamic nature of networks and sequential degradation events, offer a promising approach for cell degradation prediction.
  - Existing research often relies on synthetic data or limited KPI, potentially hindering the generalizability of the developed models to real-world network scenarios.
  - There is a need for methods that can handle the subtler nature of cell degradation compared to complete cell outages. These methods should also be adaptable to different network configurations and operating conditions.

3.3.7.3 Our Contributions

Building upon the identified limitations in existing research, this study proposes a novel HMM-based model for predicting the severity of cell degradation in an operational LTE network located in Addis Ababa, Ethiopia. This approach addresses the need for real-world data applicability and a more nuanced representation of degradation levels. Our work involved:

- **Feature and Parameter Identification:** We meticulously identified relevant KPI from the LTE network data that effectively capture the underlying factors contributing to cell degradation.

- **HMM Model Development:** We designed and developed a comprehensive HMM model specifically tailored for cell degradation prediction using the identified features and parameters.

- **Model Validation and Evaluation:** We conducted a rigorous validation process to assess the effectiveness of our proposed HMM model. This involved performance evaluation metrics to quantify its accuracy and generalizability.

- **Comparative Analysis:** We compared the performance of our HMM model against established methods for cell degradation prediction. This comparative analysis highlights the strengths and advantages of our proposed approach [166].

This novel HMM-based model has the potential to predict LTE cell degradation with improved accuracy and earlier issue identification compared to existing methods. This empowers network operators with enhanced decision-making capabilities for proactive network management and ultimately leads to a better user experience.

3.3.7.4 Methodology

- **Data collection and feature Engineering:** Building an accurate cell degradation prediction model hinges on the quality and relevance of the training data. In this study, we collected a one-month observation dataset from a live LTE network operating in Addis Ababa, Ethiopia. This real-world data provides a more robust foundation for model development compared to synthetic datasets used in some previous research.

  The collected data encompassed three KPI that directly influence cell performance:

  - **Traffic Volume:** This metric reflects the amount of data flowing through the cell, providing insights into network congestion. Increased traffic volume can lead to cell degradation as the cell struggles to handle the demands placed upon it.
  - **Number of Handover Requests:** This KPI indicates the frequency of mobile devices switching between cells. A surge in handover requests can signal potential issues with signal strength or congestion, potentially leading to cell degradation.
  - **Signal Strength:** This crucial KPI captures the average RSRP value along with timestamps of RSRP measurements within different signal strength ranges. The average RSRP value provides a comprehensive representation of the overall signal strength within the cell.

  For the HMM model, we utilized the average RSRP value to define the hidden states. Hidden states represent the underlying, unobservable conditions of the cell, and in this case, different RSRP ranges correspond to varying levels of cell health (e.g., strong signal, weak signal). The remaining KPI (traffic volume and number of handover requests) served as the observation data for the model. Observation data represents the features we can measure and use to predict the hidden states.

  Prior to feeding the data into the HMM model, we performed a meticulous data cleaning process. This step ensured the data's integrity by addressing missing values and inconsistencies. Additionally, we employed PCA for feature selection. PCA is a dimensionality reduction technique that identifies the most informative features from a dataset. By

eliminating redundant features and focusing on those with the highest loading scores in the PCA analysis, we optimized the data for model training and improved the overall effectiveness of the HMM model.

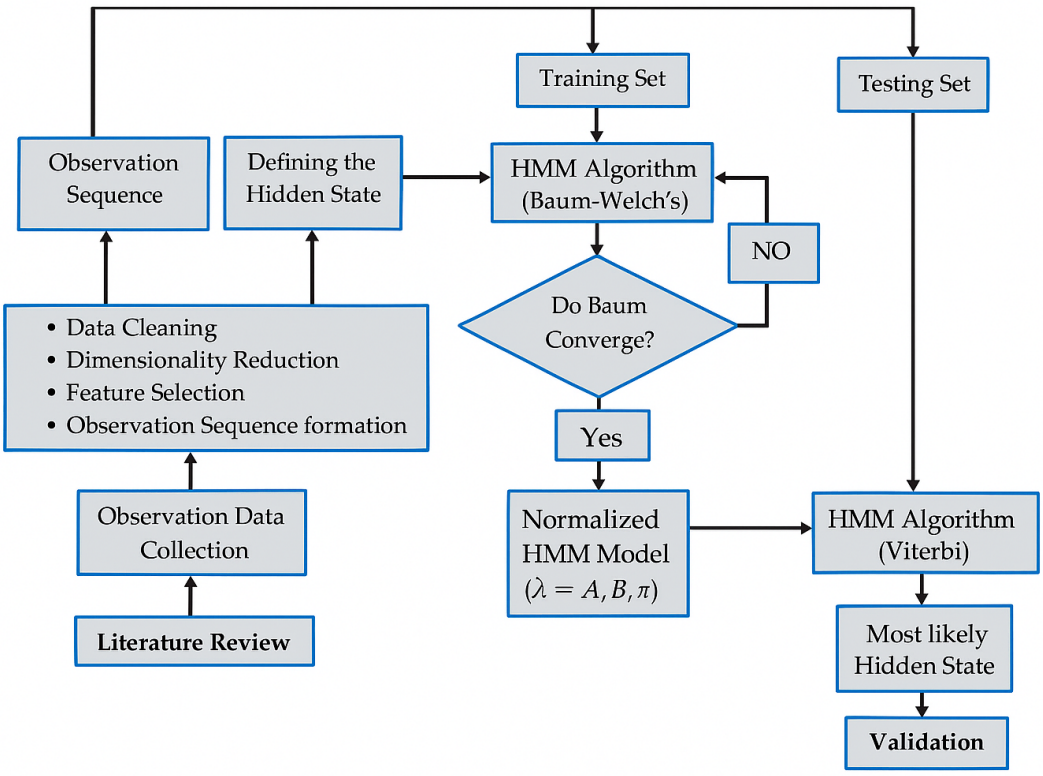


Figure 3.8: System model for prediction of cell degradation.

- **Description of Selected Features:** PCA analysis revealed that traffic volume and number of handover requests are the most important features for LTE cell degradation prediction. Traffic volume is the amount of data transmitted over the Uu interface, and a high traffic volume can strain the network and lead to congestion. A high number of handover requests indicates that the network has to work harder to keep users connected, which can be a sign of cell degradation.

  Other potentially informative features include the report times of RSRP values in different ranges. RSRP is a measure of signal strength, and a low RSRP value indicates a weak signal. High numbers of report times for RSRP values in the lower ranges can indicate that the network is degrading.

The average RSRP value is also important, but it was not included in the PCA feature selection because it was used to define the hidden state of the HMM. By understanding these features, it is possible to identify cells that are at risk of degradation and take steps to improve their performance.

- **Hidden State Formation:** Hidden states are a core concept in HMMs, representing the underlying conditions of a system that we cannot directly observe. In the context of cell degradation prediction, these states correspond to the varying levels of cell health. Defining informative hidden states is crucial for building an accurate HMM model.

  We employed a multi-step process to identify hidden states that effectively capture the severity of cell degradation. This process hinges on selecting relevant KPI that reflect the health of an LTE cell. In our study, we focused on RSRP values as the primary indicator. RSRP is a critical factor directly influencing cell performance, making it a strong choice for defining hidden states.

  Based on the usable range of RSRP values observed in the collected data, we established four hidden states corresponding to different degradation levels:

  - **Normal:** This state signifies strong signal strength and optimal cell health.
  - **Medium:** This state represents a moderate degradation in signal strength, potentially impacting user experience but not causing complete service outages.
  - **High:** This state indicates a significant degradation in signal strength, likely leading to noticeable service quality issues for users.
  - **Critical:** This state signifies a critical level of signal weakness, potentially resulting in dropped calls and significant user experience degradation.

  It's important to note that the specific thresholds defining these states (the usable range of RSRP values) can vary depending on network operators and specific network configurations. While these thresholds are established to represent acceptable communication service levels, they might require adjustments based on individual network conditions.

- **Observation Sequence Formation:** In HMM-based prediction models, observation sequences represent the observable features used to infer the underlying hidden states. There are two primary methods for creating these sequences: direct observation and feature extraction. In this study, we opted for feature extraction to transform the raw KPI data into a more manageable and informative set of observation sequences.

K-means clustering, a popular machine learning algorithm, served as our tool for feature extraction. K-means groups data points into a predefined number of clusters based on their similarity. To determine the optimal number of clusters for our observation sequences, we employed the elbow method. This technique involves analyzing the distortion (distance between data points and their cluster centers) as the number of clusters increases. The "elbow" point on the graph, where the distortion starts to increase rapidly with diminishing returns, indicates the optimal number of clusters. As depicted in Figure 3.9, the elbow method suggested six clusters as the most suitable choice for our data.

Each of these six unique clusters was then mapped to a distinct observation symbol, resulting in a discrete sequence of symbols. This sequence represents the evolving state of the cell based on the observed features over time. For instance, a sample observation sequence might appear as: ([[2],[3],[3],...,[2],[1],[0]]). Here, the numbers within the NumPy array represent the sequence of observation symbols, providing valuable insights into the cell's behavior and potential degradation over time.

- **Understanding Hidden and Observable State Relationships:** In the context of our HMM model, two key components govern the relationship between the underlying cell health (hidden states) and the measurable features (observable states):
  - **Emission Probabilities:** These probabilities quantify the likelihood of observing a particular feature value (e.g., high traffic volume) given a specific hidden state (e.g., medium degradation). Essentially, they model the connection between the unobserved cell health and the observable network behavior.
  - **Transition Matrix:** This matrix captures the probability of transitioning from one hidden state (e.g., normal signal strength) to another (e.g., moderate degradation) over time. It reflects the dynamic nature of cell degradation, where a cell might progressively degrade from a healthy state to a critical one.

Table 3.9 provides a valuable illustration of the expected changes in observable states (traffic volume, handover requests) as the hidden state representing cell degradation transitions. For instance, the table might show that as the cell degrades (hidden state transition), the average traffic volume and number of handover requests tend to increase (observable state changes). This aligns with the real-world behavior of cellular networks, where users experiencing poor signal quality (degradation) might attempt more handovers or generate higher data traffic due to slower download speeds. It's important to acknowledge that the specific relationships depicted

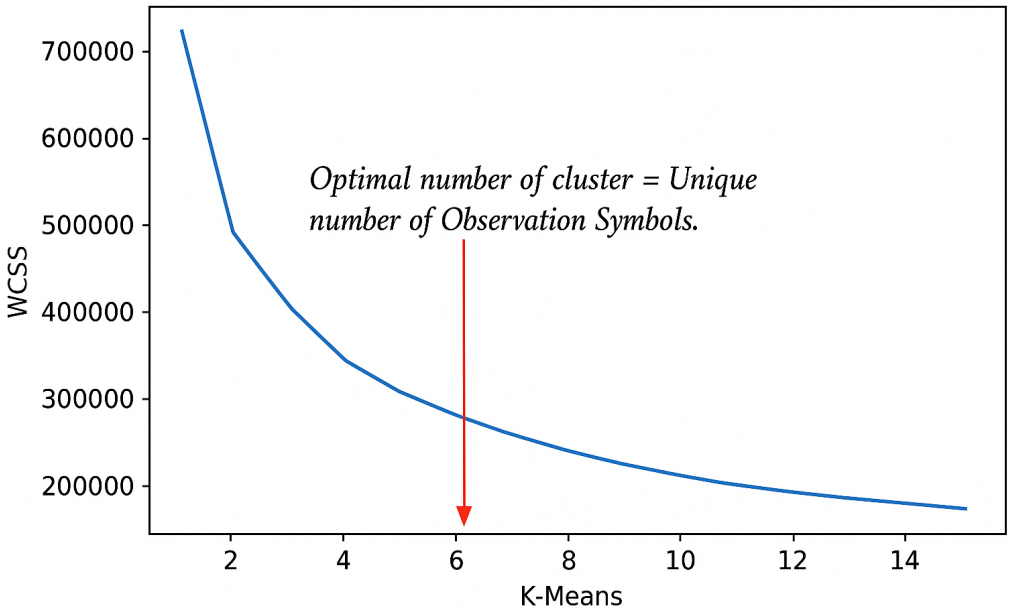


Figure 3.9: Optimal number of clusters using elbow method.

in Table 3.9 might vary depending on the unique characteristics of a network. However, the general trends presented in the table offer valuable insights for interpreting the behavior of the HMM model and tracking the progression of cell degradation within an LTE network. This information can ultimately be used to predict degradation sequences and proactively address network issues before they significantly impact user experience.

3.3.7.5 Results and Discussions

Training the Model and Results: There are two techniques to create HMMs in Python. The first technique is simple but assumes that the observation sequence and the state sequence are independent. The second technique is more complex but can often produce better results because it uses an iterative Baum-Welch algorithm to fit the model to the data.

In this dissertation, we used the second training technique, which follows the Baum-Welch algorithm. The Baum-Welch algorithm is an iterative algorithm that is used to train or estimate the parameters of an HMM. The algorithm starts with initial estimates of the parameters, and then iteratively updates the estimates until they converge. We set the convergence threshold to 0.00001. The Baum-Welch algorithm optimizes the parameters of an HMM, including the prior (or initial) probabilities, transition probabilities, and emission probabilities. These parameters

Table 3.9: Modeling cell degradation with HMMs.

Usable RSRP Range (dBm)	Degradation States (RSRP Classification)	Observation Emitted by Hidden State		
		Average Traffic Volume	Number of Outgoing Handover Requests	Number of RSRP Measurement Reports
(−90, −76]	**Normal**	Moderate or steady	Within reasonable limits	Consistent and reflects typical user behavior
(−103, −90]	**Medium**	Slight increase compared to normal	Slightly increase	May increase slightly as users actively search for better signal quality
(−109, −103]	**High**	Significantly higher	Significant number of handover requests	Significantly higher s users continuously monitor the signal strength
[−140, −109]	**Critical**	At its peak or overloaded	Very high	Extremely high, indicating extensive signal quality issues and users' efforts to find a better connection

are crucial for predicting the future state of an LTE cell and describing the HMM probabilistically. The final HMM parameters obtained from the Baum-Welch algorithm are listed as follows:

Table 3.10: The initial probability matrix.

	Normal	Medium	High	Critical
0	0.262891	0.255794	0.225265	0.25605

Predicting the Sequence of Hidden States with the Viterbi Algorithm: A critical aspect of HMM-based prediction models lies in their ability to predict the most likely sequence of hidden states. In our case, this translates to predicting the evolving health state (normal, medium, high, critical) of an LTE cell over time. To achieve this prediction, we employ the Viterbi algorithm. This powerful algorithm takes a trained HMM model and an observed sequence (e.g., traffic volume, handover requests) as input and identifies the most probable sequence of hidden states that could have generated that observed sequence.

Evaluating the HMM Model's Performance: Assessing the effectiveness of the HMM model is crucial for determining its suitability for real-world deployment. To achieve this, we

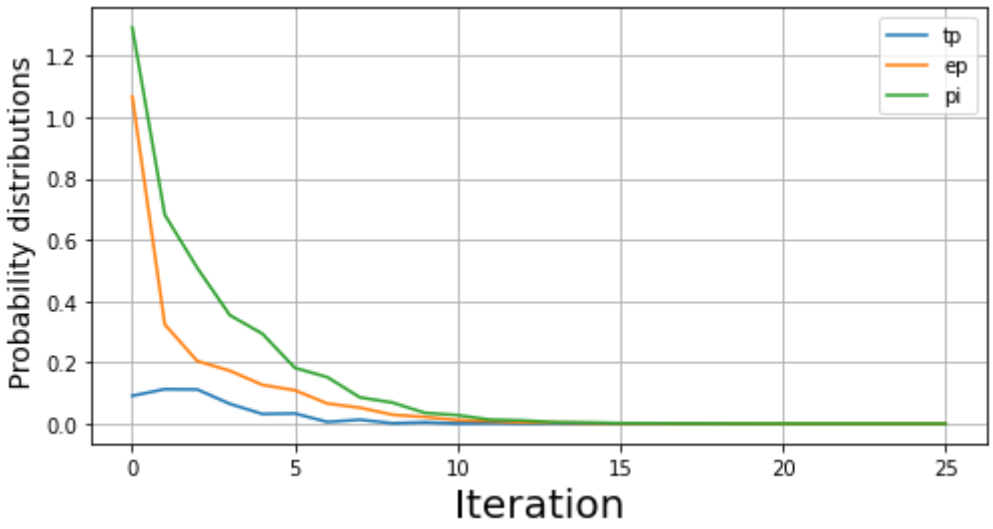


Figure 3.10: Convergence of the Baum-Welch's model.

Table 3.11: The transition matrix.

	Normal	**Medium**	**High**	**Critical**
Normal	0.00972	0.83972	0.13309	0.01747
Medium	0.00041	0.96892	0.03060	0.00008
High	0.00062	0.99301	0.00459	0.00178
Critical	0.00184	0.96912	0.01522	0.01382

employed various evaluation metrics to quantify the model's performance in predicting LTE cell degradation.

The model achieved an impressive average accuracy of 93.12%. This metric indicates a high degree of correspondence between the model's predicted hidden states (degradation levels) and the actual degradation levels experienced by the cells. Furthermore, the model demonstrates a high precision of 92.82%. Precision signifies the model's ability to accurately identify true degradation cases and minimize the likelihood of falsely predicting degradation when the cell is actually healthy. In simpler terms, these results indicate that the HMM model can reliably predict cell degradation with a high degree of accuracy and minimize false alarms. This capability empowers network operators to proactively address potential degradation issues before they significantly impact user experience. By taking preventive measures based on the model's predictions, network operators can ensure optimal network performance and service quality for their customers.

Table 3.12: The emission probability matrix.

	0	1	2	3	4	5
Normal	0.645342	0.08489	0.020273	0.00994	0.077752	0.161803
Medium	0.645342	0.08489	0.020273	0.00994	0.077752	0.161803
High	0.645342	0.084891	0.020273	0.00994	0.077752	0.161803
Critical	0.645342	0.08489	0.020273	0.00994	0.077752	0.161803

3.3.7.6 Conclusions and Future Work

Conclusions

This dissertation investigated the potential of HMMs for predicting cell degradation severity in LTE networks. This approach aims to empower MNO with proactive network management capabilities, ultimately leading to improved network performance, reduced operational costs, and enhanced user experience.

The study employed real-world data collected from a live LTE network in Addis Ababa, Ethiopia. This data provided a foundation for training an HMM model to predict the severity of cell degradation, categorized into four levels: normal, medium, high, and critical. The model achieved an impressive average accuracy of 93.12%, successfully identifying the underlying cell health states based on observable features like traffic volume, handover requests, and signal strength.

These results suggest that HMMs offer a promising approach to predict LTE cell degradation and its progression. The model's effectiveness demonstrates its potential to contribute to proactive network management strategies, including:

- **Prevention:** By anticipating potential degradation issues, MNO can take preventive measures such as network optimization or resource allocation adjustments to maintain optimal service quality.
- **Mitigation:** In cases where degradation is already occurring, early detection through MNO predictions allows for faster intervention and mitigation strategies to minimize user experience impact.
- **Real-time Monitoring:** The model can be integrated into network management systems for continuous monitoring of cell health, enabling MNO to proactively address emerging issues.

Future work

While this study demonstrates the effectiveness of HMM for cell degradation prediction, there's always room for further exploration. Here are some interesting avenues for future research:

- **Complex Outage States:** Current industry-defined outage states for base transceiver stations (BTS) are intricate and challenging to monitor due to the high number of possible states and their dynamic nature. Investigating the application of different HMM variants, potentially with more complex state transitions, could be beneficial for modeling these intricate outage states.
- **Integration with Network Management Systems:** Integrating the developed HMM model with existing network management systems would facilitate seamless real-time monitoring and utilization of prediction results for proactive network management decisions.
- **Comparison with Other Techniques:** Further research could involve comparing the performance of HMMs with other machine learning or deep learning techniques for cell degradation prediction. This comparison could provide valuable insights into the strengths and limitations of different approaches.
- **Incorporating Additional Data Sources:** Exploring the inclusion of additional data sources like weather patterns or cell equipment health metrics could potentially enhance the model's prediction accuracy by accounting for external factors influencing cell performance.

By continuing to explore and refine HMM-based prediction models, coupled with the development of robust QoE-to-QoS mapping techniques explored in Chapter 4, MNO can unlock a powerful suite of tools for proactive network management. Chapter 4 delves deeper into the intricacies of HMMs, specifically focusing on formulating an HMM-based QoE-to-QoS mapping model. This model will shed light on the relationship between objective, measurable QoS parameters like traffic volume and handover requests, and the subjective user experience quantified as QoE.

This combined approach, leveraging both cell degradation prediction and QoE-to-QoS mapping, empowers MNO to not only anticipate potential network issues but also understand how these issues translate into real-world user experience impacts. This comprehensive understanding allows for more targeted interventions and resource optimization strategies, ultimately leading to a significant improvement in user experience on their LTE networks.

4 Modeling, Experimental Setup, and Data Collection

This chapter presents the methodology underpinning the study, structured around three key components: the formulation of the HMM-based QoE-to-QoS mapping model, the experimental design for model evaluation, and the strategies employed for data collection and preprocessing.

Section 4.1 introduces the system architecture and mathematical formulation of the proposed model. Sections 4.2 and 4.3 describe the mobile application, user survey mechanisms, and preprocessing methods used to gather and prepare both subjective and objective data. Section 4.4 details the experimental setup, evaluation metrics, and procedures used to validate model performance in realistic mobile network conditions.

4.1 HMM-based QoS-to-QoE Mapping Model: Architecture and Formulation

This section introduces the architecture and mathematical formulation of HMM developed for mapping objective QoS parameters to subjective QoE levels. The primary goal is to address the inherent challenge in modeling the probabilistic relationship between what is measurable and what is perceptual. The section further reflects on the modeling paradox encountered in reconciling classical HMM assumptions with the semantics of QoS-to-QoE mapping, and it outlines the adopted resolution strategy.

4.1.1 System Model Description

The proposed system aims to infer a user's perceived QoE from a set of measurable QoS parameters. A core challenge emerges from the conceptual structure of HMMs: in classical use, HMMs assume that hidden states generate observable outputs. However, in our scenario, QoS metrics are both measurable and often treated as the drivers of QoE, suggesting an inverse causal direction.

This creates a modeling paradox:

- If QoS is considered an observable state, it violates the HMM assumption that observables are emitted by hidden states.

- If QoE is considered a hidden state, it fulfills the requirement of being latent but not of emitting the observable sequence in a strictly generative sense.

To address this, we adopt a pragmatic interpretation of HMM, as a statistical tool for domain-to-domain mapping rather than a causal engine. Accordingly, two modeling configurations were developed and empirically evaluated:

- **Model A:** QoE as the hidden state, inferred from observable QoS sequences.
- **Model B:** QoS as the hidden state, inferred from observable QoE assessments.

Empirical evaluation showed that Model A, aligning with the classical HMM direction (hidden → observed), yielded superior predictive performance. Consequently, we retained this configuration for subsequent modeling.

The resulting system operates as follows:

1. **QoS Data Acquisition:** Real-time QoS parameters (e.g., throughput, latency, packet loss, jitter) are collected from the mobile network at regular intervals.
2. **QoE Data Acquisition:** Subjective QoE data is obtained from users via surveys, serving as ground truth for perceptual quality.
3. **QoS State Mapping:** Raw QoS data is mapped to discrete states (e.g., the 16 joint states of accessibility and retainability discussed in Chapter 3), serving as observations for the HMM.
4. **HMM Inference:** The HMM infers the sequence of unobservable QoE states from the sequence of observed QoS states.
5. **QoE Prediction:** The trained model enables real-time prediction of user QoE from incoming QoS data sequences.

4.1.2 HMM-based QoE Mapping Model Formulation

A HMM is formally defined by the following parameters:

- N: Number of hidden states (e.g, latent QoE levels)
- M: Number of observable states (e.g., mapped QoS states)
- A: State transition probability matrix

- B: Emission probability matrix
- π: Initial-state probability distribution

In this work, the HMM serves as a domain-mapping tool that connects network-level QoS measurements to the end-user's perceived QoE. However, a conceptual challenge arises when applying classical HMM logic to this context. Traditionally, HMMs assume that hidden states emit observable states, following a generative model structure. In contrast, QoS metrics are directly measurable and are generally treated as inputs that influence QoE, not vice versa. Conversely, QoE is inherently latent but does not strictly emit QoS readings.

This misalignment between classical HMM structure and the QoS-to-QoE relationship presents a modeling paradox:

- Treating QoE as hidden aligns with its unobservability but breaks the emission logic.
- Treating QoS as hidden captures the emission structure but contradicts its observable nature.

To resolve this, we approached the HMM as a mathematical mapping framework rather than a causal model. We implemented and tested both configurations:

- **QoE as hidden, QoS as observed**
- **QoS as hidden, QoE as observed**

The former yielded better predictive performance and was more interpretable in the context of performance monitoring. Therefore, this configuration was adopted for the final model.

4.1.2.1 Defining Hidden State and Addressing High Dimensionality

In this HMM, the hidden states represent the unobservable QoE levels perceived by the end-user. A fundamental modeling challenge in this context arises from the paradox of causality and observability in HMMs. Classical HMMs assume that hidden states emit observable events. However, in QoS-to-QoE mapping, QoS is directly measurable and often viewed as the input influencing QoE, rather than an output. This presents a paradox: treating QoS as observable aligns with its measurability, but it does not satisfy the generative role of emitted states in an HMM; conversely, treating QoE as hidden satisfies the latent condition but conflicts with its dependence on user-perceived outcomes, which are not the drivers of QoS metrics.

To reconcile this mismatch, we treated the HMM not as a generative causal model but as a mathematical mapping framework between domains. We empirically evaluated both

configurations: QoS as hidden and QoE as observed, and vice versa. We selected the one that yielded superior prediction accuracy and interpretability. Thus, in our final model, QoE is treated as the latent variable (hidden state), and QoS as the observed outcome, enabling effective mapping from network performance to user perception.

The HMM's probabilistic structure is leveraged to map objective network metrics to subjective user experience, effectively treating the model as a function from the QoS domain to the QoE domain.

Based on the findings from Chapter 3, particularly the K-means clustering analysis, we define $N = 4$ hidden states. These four hidden states correspond directly to the four clusters identified through the K-means algorithm on the 16 joint states of accessibility and retainability. Each cluster represents a distinct aggregated network performance condition, which is assumed to correlate with a specific level of user QoE (e.g., "**Excellent**", "**Good**", "**Fair**", "**Poor**" QoE). This reduction in dimensionality from 16 joint QoS states to 4 hidden QoE states is crucial for the model's tractability and interpretability.

The QoE parameters used to validate and support these inferred states, collected via user surveys, include: *Age Group*, *Gender*, *Occupation*, *Education Level*, *Device Model*, *Screen Resolution*, *Subscription Type*, *Location*, *Time of Usage*, and *Overall Satisfaction* (measured on a **Likert Scale** from **Very Satisfied** to **Very Dissatisfied**).

4.1.2.2 Defining Observable States

The observable states of the HMM are derived from the collected QoS parameters. These include cell signal strength (in dBm), signal quality level, RSRP, buffering ratio, frame error rate, jitter, packet error rate, and device RAM usage. These variables reflect the network's operational condition at each time interval.

A fundamental challenge in adapting HMMs to mobile network contexts lies in the high dimensionality and multivariate nature of QoS data. Classical HMM implementations are typically designed for a single observable variable per time step, which creates a methodological gap for applying them directly to multi-dimensional inputs.

To overcome this, a two-stage dimensionality reduction strategy was adopted. First, PCA was used to reduce the initial high-dimensional QoS feature set. PCA decorrelates features, suppresses noise, and emphasizes the principal components that explain the most variance in the data. This step not only mitigates overfitting but also enhances computational efficiency.

The transformed data was then discretized into 16 joint QoS states, defined by combinations of accessibility and retainability metrics, as previously detailed in Chapter 3. To further compress

and structure the observable space, K-means clustering was applied to these 16 joint states. The clustering algorithm grouped similar network conditions into four representative clusters (C1–C4), thus mapping the 16 joint states into four discrete observable states for the HMM.

The assignment of joint states to clusters (e.g., states 0–3 mapped to Cluster 3, states 4–7 to Cluster 0, etc.) was empirically determined through iterative experimentation. The optimal number of clusters was selected using quantitative heuristics such as the Elbow Method and Silhouette Analysis, as elaborated in Chapter 3. This approach enabled the observable state space to be both statistically compact and semantically meaningful.

4.1.2.3 Emission Probabilities

The emission matrix $B = b_j(o_k)$), defines the probability of observing a particular QoS state (o_k) given that the HMM is in a specific hidden QoE state (s_j). Formally $b_j(o_k) = P(o_k \mid s_j)$.

These probabilities are estimated during the HMM training phase. For each hidden QoE state (cluster), the emission probabilities reflect the likelihood of observing any of the 16 discrete joint QoS states. For example, if a hidden state corresponds to "Excellent QoE," we would expect a high probability of observing QoS states (from the 16 joint states) that represent high accessibility and retainability. Conversely, a "Poor QoE" hidden state would likely emit QoS states indicating low accessibility and retainability.

4.1.2.4 Transition Probabilities

The state transition probability matrix $A = a_{ij}$, defines the probability of transitioning from one hidden QoE state (s_i) to another hidden QoE state (s_j) at the next time step. Formally, $a_{ij} = P(s_j^{(t+1)} \mid s_i^{(t)})$.

These probabilities capture the dynamics of user perceived quality over time. They indicate how likely a user's QoE is to improve, degrade, or remain stable. For instance, a high probability a Good, Excellent would suggest that users frequently transition from "**Good** QoE" to "**Excellent** QoE." These probabilities are also learned during the HMM training process from the observed sequences of QoE data. The core HMM algorithms involved in this process are the Forward, Backward, and Baum-Welch algorithms, which are fundamental for parameter estimation in HMMs.

4.1.2.5 Baseline Models for Comparison

To rigorously evaluate the performance of the proposed HMM-based QoE mapping model, its results will be compared against several baseline machine learning models. These models

represent alternative approaches to QoE-to-QoS mapping and will serve to highlight the advantages of the HMM, particularly its superiority in handling sequential and probabilistic data. The chosen baseline models include:

- **SVM:** Selected for its robust classification capabilities and effectiveness in high-dimensional spaces, representing a powerful kernel-based classification approach.
- **RF**: Chosen as an ensemble learning method known for its high accuracy, robustness to overfitting, and ability to handle various data types.

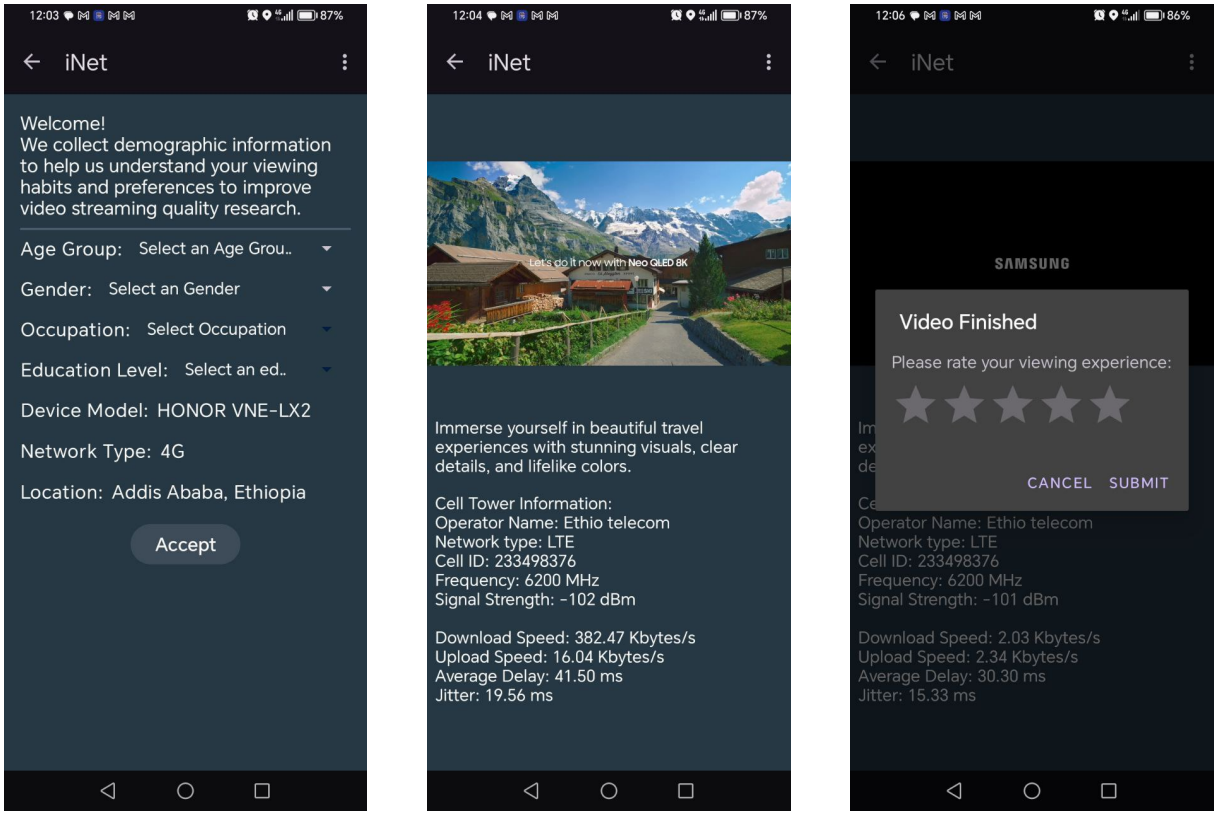
Preliminary observations indicated that RF often outperformed SVM, with SVM significantly underperforming in Precision, Recall, and F1-score in initial tests. HMM, however, performed well across all four key metrics (Accuracy, Precision, Recall, and F1-score), particularly excelling in Precision, Recall, and F1-score. These findings prompted discussions on why such differences occurred, considering data characteristics, algorithm nature, and implementation limitations. Case studies with variations in data and operators also showed the models' sensitivity. The comparative analysis will further explore these differences and contextualize the HMM's strengths.

4.2 Data Collection Techniques

Accurate and comprehensive data collection is paramount for training and validating the HMM-based QoE mapping model. This research employs a dual-pronged approach, combining objective QoS measurements from a custom mobile application with subjective QoE assessments through integrated user surveys.

4.2.1 Mobile Application Development for QoE and QoS Data Acquisition

A custom mobile application, named iNET, was developed to facilitate the real-time collection of both objective QoS parameters and subjective QoE feedback directly from end-users. This tool was designed to overcome the limitations of relying solely on network-side data, providing a direct link between user perception and network performance. iNET comprises three main components: an Android-based app for data collection, a server-side component for video delivery and data storage, and a web-based dashboard for data analytics.



(a) **(b)** **(c)**

Figure 4.1: iNET Android App Screens (a) Demographics data. (b) Network parameter measurement while streaming. (c) User experience rating.

4.2.1.1 Functionality of iNET

The iNET mobile application was meticulously designed with the following key functionalities to ensure robust and comprehensive data acquisition for the research:

- **Comprehensive Data Collection:** : iNET is engineered to collect a wide array of data points:
  - **Demographic Data:** Users are prompted to fill out a form to collect basic demographic details, including Age Group, Gender, Occupation, Education Level, Device Model, Screen Resolution, Subscription Type, Location, and Time of Usage as shown in Figure 4.1 (a). This data is crucial for contextualizing QoE.
  - **Location Data:** With explicit user consent, iNET captures precise location data, enabling spatial analysis of network performance and QoE.
  - **Device-Specific Information:** The app automatically records device-specific

parameters such as device type, brand, and available RAM, which can influence perceived QoE.

- **Network Performance Metrics (QoS):** iNET continuously monitors and logs various QoS parameters in the background (Figure 4.1 (b), minimizing user intervention. These include:
  - Throughput (downlink and uplink data rates)
  - Latency (round-trip time to a test server)
  - Packet Loss (percentage of data packets lost during transmission)
  - Jitter (variation in packet delay)
  - Cell Signal Strength (e.g., RSSI,RSRP, dBm)
  - Cell Signal Quality Level
  - Buffering Ratio (specifically for video streaming)
  - Frame Error Rate
  - Bit Error Rate
  - Network Type (e.g., 2G, 3G, 4G, 5G).

- **Video Playback and Monitoring:** Users watch a pre-selected video while metrics are collected.
  - iNET incorporates a video playback feature where users watch a pre-selected, approximately two-minute video.
  - During playback, the app actively monitors and records video-specific performance metrics, such as buffering events and frame errors, directly correlating them with the underlying QoS conditions.

- **Integrated Rating Feature:** At the conclusion of the video playback, users are prompted to rate their experience based on video smoothness (explicitly not the content of the video). This rating is collected using a standard 5-star rating model to capture the user's satisfaction during the video playback as shown in Figure 4.1 (c).

- **Automated Data Handling:** Collected QoS and demographic data, along with user ratings, are periodically and automatically uploaded from the phone app to a central server. This ensures real-time data availability and accurate synchronization.

- **User Interaction Flow and Constraints:**
  - The app's setup guides users through initial permissions and demographic data entry.
  - A critical operational constraint for data collection is that iNET is designed to function exclusively on mobile data (e.g., Ethio Telecom or Safaricom) and will not operate on Wi-Fi, ensuring data is collected under realistic mobile network conditions.
  - Users are encouraged to use iNET repeatedly at different times and locations to enrich the dataset.

4.2.1.2 User Recruitment and App Deployment Strategy

Users were recruited via online campaigns, universities, and social media. Incentives such as mobile data top-ups were provided. Secure deployment and clear consent mechanisms were implemented.

- **Recruitment:** Users were recruited through a combination of online advertisements, university outreach programs, and social media campaigns. A key focus was on recruiting participants from various geographical locations and with diverse mobile usage patterns to capture a wide spectrum of network conditions and user experiences.
- **Incentives:** To encourage consistent app usage and diligent survey participation, participants were offered small incentives, such as gift cards or mobile data top-ups.
- **Deployment:** The iNET mobile application was distributed through secure, private channels to selected participants. Comprehensive instructions for installation, usage, and privacy policies were provided to ensure transparency and ease of adoption.
- **Consent:** Informed consent was obtained from all participants, clearly outlining the data collection process, privacy safeguards, and emphasizing the voluntary nature of their participation.

4.2.1.3 Video Content Selection and Preparation

The selection and preparation of video content for playback within the iNET application were critical to ensure a standardized and relevant stimulus for QoE assessment. The process involved

defining specific criteria based on existing literature and practical considerations, followed by meticulous preparation steps.

Video Selection Criteria: Based on a review of relevant literature and common video consumption patterns, the following criteria were established for video selection:

- **Video Types:** To ensure a diverse range of visual content and appeal to a broad user base, videos were selected from categories including educational, nature, sport, and entertainment.
- **Film Duration:** Each video was carefully trimmed to a short duration, specifically ranging from 26 seconds to 31 seconds, with a target of approximately 30 seconds. This duration was chosen to be long enough to allow for meaningful QoS measurements and QoE perception, yet short enough to minimize user fatigue and encourage repeated participation.
- **Resolution Types:** Following common online video platform recommendations (e.g., YouTube), all selected videos were standardized to a resolution of 1080p. This ensured a consistent and high-quality visual experience, allowing network performance issues to be more clearly discernible.

Video Preparation Process: Once selected, the videos underwent a preparation process to meet the defined criteria:

- **Trimming and Resolution Adjustment:** Online software tools were utilized to trim the videos to the desired 26-31 second duration and adjust their resolution to 1080p. Specifically, "Wondershare Filmora" was used for adjusting audio synchronization with the video content.
- **Copyright and Patent Clearance:** A crucial step involved ensuring that no copyright or patent infringements were associated with the selected video content, adhering to ethical research practices.

Measurement Frequency and Sampling: For the short video durations (30-40 seconds), decisions were made regarding the frequency of KPI measurements and the sampling approach:

- **Measurement Interval:** The frequency at which measurements for each KPI were taken during video playback was determined by consulting with network experts and reviewing related literature on network parameter measurement (e.g., delay and jitter using "ping" commands). This ensured that sampling was frequent enough to capture dynamic network conditions without overwhelming device resources.

- **Sampling Method:** A decision was made on whether to average multiple samples or select a single representative sample for each measurement interval. This choice was informed by best practices in network monitoring to accurately reflect the instantaneous or aggregate network state.

4.2.2 User Surveys for Subjective QoE Assessment

Subjective QoE assessment is critical for establishing the ground truth against which the HMM's inferred QoE states are validated. This was achieved through carefully designed user surveys seamlessly integrated into the iNET mobile application.

4.2.2.1 Survey Design

Surveys were integrated into the app using a MOS scale (1–5) to rate service-specific questions. Contextual and demographic information was also gathered. The user surveys were designed to capture the perceived QoS, focusing on aspects directly influenced by network performance and user demographics. Key design considerations included:

- **MOS Scale:** The MOS (typically 1-5, where 1 is Bad and 5 is Excellent) was primarily used for overall QoE rating, as it is a widely accepted standard in telecommunications for quantifying subjective quality.
- **Service-Specific Questions:** Questions were tailored to specific mobile services (e.g., video streaming quality, web Browse responsiveness, voice call clarity) to capture nuanced user experiences.
- **Contextual Information:** Users were prompted to provide brief contextual information (e.g., "What were you doing?", "Where were you?") to aid in correlating subjective feedback with objective QoS data and understanding the environmental factors influencing QoE.
- **Demographic Data:** Surveys collected demographic data such as Age Group, Gender, Occupation, Education Level, Device Model, Screen Resolution, Subscription Type, Location, and Time of Usage. This information is crucial for understanding how different user profiles and contexts might influence QoE perception.
- **Simplicity:** Surveys were kept short and easy to complete to minimize user fatigue and maximize participation rates, ensuring a high volume of reliable subjective data.

4.2.2.2 Survey Integration with the Mobile App

The surveys were seamlessly integrated into the iNET mobile application to ensure timely and contextually relevant feedback, minimizing disruption to the user experience:

- **Triggering Mechanism:** Surveys were triggered at strategic points, such as after a significant change in QoS parameters, after a certain duration of app usage, or at random intervals to capture a wide range of experiences and avoid predictable prompting.
- **Non-Intrusive Prompts:** Prompts for surveys were designed to be non-intrusive, appearing as subtle notifications rather than interrupting ongoing user activities, thereby enhancing user retention.
- **Data Synchronization:** Survey responses were immediately synchronized with the collected QoS data on the central server using precise timestamps, ensuring accurate temporal alignment between subjective and objective measurements for subsequent analysis.

  More than 700 users initiated a session, but over 550 were able to successfully complete it. The data were continuously monitored and exported through a web-based portal for subsequent processing, as shown in Figure 4.2.

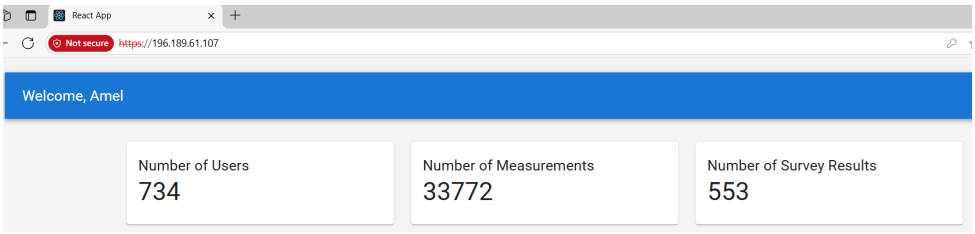


Figure 4.2: Administration Dashboard for iNET.

4.3 Data Pre-processing Methods

Raw data collected from mobile applications (iNET) and user surveys often contains noise, inconsistencies, and missing values. Robust data preprocessing is essential to ensure the quality, consistency, and suitability of the data for HMM training and subsequent analysis. The following systematic steps were undertaken to prepare the dataset:

1. **Data Cleaning:**

- **Missing Values Handling:** For objective QoS metrics, small data gaps were addressed using linear interpolation to maintain data continuity. Larger or persistent segments with missing data were identified and, if necessary, removed to prevent the introduction of bias or inaccuracies.
- **Outlier Detection and Treatment:** Outliers in various QoS parameters (e.g., unusually high latency spikes, extreme packet loss) were identified using statistical methods such as the Z-score and Interquartile Range (IQR). Depending on the nature and potential impact of the outlier, they were either capped (transformed to a maximum/minimum plausible value), statistically transformed, or removed from the dataset.
- **Inconsistent Data Resolution:** Data inconsistencies, such as impossible measurement values or erroneous timestamps, were identified and either corrected based on logical rules or removed if correction was not feasible.

2. **Data Merging and Synchronization:**
   - The data collected from the iNET application was initially stored in three distinct parts: user profile information, continuous measurement data, and discrete survey responses. Merging these disparate data sources was critical for creating a unified dataset for analysis.
   - **Grouping Measurement Data:** Measurement data was first grouped by date time (specifically, created_at attribute), as measurements were taken continuously. Rows with null or empty values for essential attributes such as user_id, video_id, TACLAC, cellSigalStrengthDbm, cellRSRP, frame_error_rate, jitter, packet_error_rate, and created_at were dropped to ensure data integrity. The created_at attribute of the measurement data was rounded to the nearest minute, dropping the seconds component, to facilitate matching with survey data. Subsequently, the grouped data involved taking the average for numerical columns and the first value for text columns.
   - **Merging with Survey Data:** The seconds component was dropped from the created_at column of the survey data. The pre-processed measurement data was then merged with the survey data using user_id and the rounded created_at column. A tolerance of 4 minutes was applied during the merge process to account for potential minor discrepancies in timestamps between measurement logs and survey submissions, ensuring robust matching of records.

3. **Feature Engineering:**
   - **QoS State Discretization:** Raw continuous QoS parameters were transformed into discrete observable states for the HMM. This involved mapping the multi-dimensional QoS features to the 16 joint QoS states (combinations of accessibility and retainability) as defined and clustered in Chapter 3. These 16 discrete states form the primary observation sequences for the HMM.
   - **QoE State Mapping:** Subjective Mean Opinion Score (MOS) ratings from user surveys were mapped to the 4 defined hidden QoE states (e.g., "Excellent," "Good," "Fair," "Poor") to serve as the ground truth for HMM training and validation. This involved defining clear thresholds for MOS scores corresponding to each QoE state.
4. **Dimensionality Reduction:**
   - As highlighted in Section 4.1.2.2, the raw QoS data comprised numerous features. To manage this high dimensionality and prepare the data for the HMM's observable states, a two-stage dimensionality reduction process was applied:
     * **Principal Component Analysis (PCA):** PCA was initially employed to reduce the overall dimensionality of the raw QoS features, decorrelate them, and potentially mitigate noise by identifying the principal components that capture the most variance in the data.
     * **K-means Clustering:** Following PCA, K-means clustering was applied to the resulting feature space, specifically targeting the 16 joint QoS states. This step was critical for aggregating these 16 states into the final 4 observable clusters, which directly correspond to the hidden QoE states, as detailed in Chapter 3. This approach effectively reduces the complexity of the observable sequence for the HMM.
   - **Feature Selection (Pearson Correlation):** Prior to applying dimensionality reduction techniques, Pearson Correlation was utilized to assess the relationships between various QoS parameters and QoE, aiding in the selection of the most relevant features for the model.
5. **Data Normalization/Scaling:**
   - Where applicable, continuous QoS features were normalized or scaled to a common range (e.g., 0 to 1 or Z-score normalization). This step is crucial to prevent

features with larger numerical values from disproportionately influencing the clustering and HMM training processes, ensuring robust model performance.

6. **Sequence Generation:**
   - The fully preprocessed data was then structured into appropriate sequences of observed QoS states, with corresponding inferred or ground-truth hidden QoE states, making it suitable for HMM training, validation, and evaluation.

4.4 Experimental Setup and Evaluation Metrics

This section outlines the environment and specific parameters used for conducting the experiments, along with the metrics chosen to evaluate the model's performance. Figure 4.3 shows the setup used for data collection and analysis.

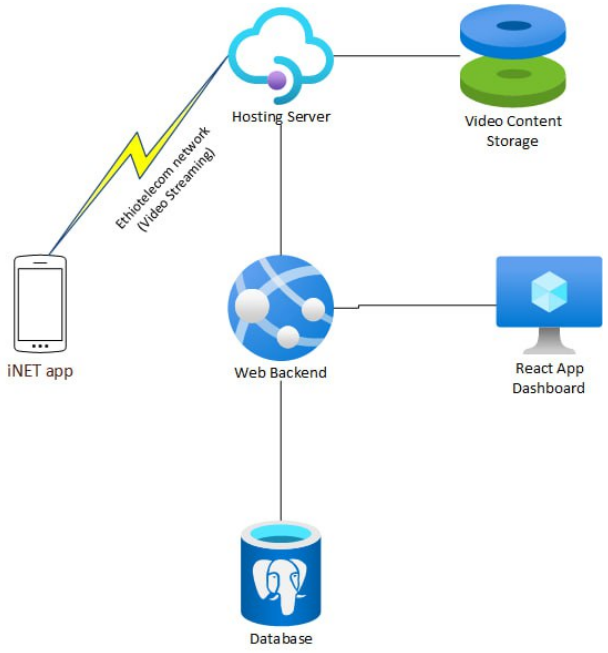


Figure 4.3: Experimental setup for data collection and analysis

4.4.1 Experimental Setup

The experiments were conducted using a dedicated computational environment to ensure reproducibility and efficient processing of the large datasets. Figure 4.4 shows the steps used to train and evaluate HMM model.

- **Hardware:** Details of the computational resources (e.g., CPU, RAM, GPU if applicable) used for model training and simulation.
- **Software:** Specific programming languages (e.g., Python), libraries (e.g., `hmmlearn`, `scikit-learn`), and operating systems utilized.
- **Data Split:** The collected and preprocessed dataset was rigorously split into training, validation, and testing sets to ensure unbiased evaluation. Typically, a 60/20/20 or 70/15/15 split was employed, with careful consideration for maintaining the temporal order of data where relevant for HMMs.
- **HMM Training Parameters:** Specific parameters for the HMM training algorithm (e.g., number of iterations for Baum-Welch algorithm, convergence criteria).

4.4.2 Experimental Design

The experimental design in Figure 4.3 was structured to systematically evaluate the performance of the HMM-based QoE mapping model and compare it against baseline approaches.

1. **HMM Training and Optimization:** The HMM was trained using the Baum-Welch algorithm on the training dataset. Hyperparameters were optimized using the validation set to prevent overfitting.
2. **Baseline Model Training:** Each baseline model (Multivariate Regression, Simple Threshold-Based, Flat Classification) was trained on the same training dataset, ensuring a fair comparison.
3. **Performance Evaluation:** All models were evaluated on the unseen test dataset using the defined evaluation metrics.
4. **Scenario-Based Analysis:** Experiments were conducted under various network conditions or user activity patterns (if such data was available) to assess the model's robustness and adaptability.
5. **Sensitivity Analysis:** Where appropriate, sensitivity analysis was performed to understand the impact of varying key parameters (e.g., number of hidden states, QoS discretization thresholds) on model performance.

4.4.3 Evaluation Metrics

The performance of the HMM-based QoE mapping model and baseline models was assessed using a combination of standard classification metrics and HMM-specific metrics.

- **Classification Accuracy:** Percentage of correctly predicted QoE states.
- **Precision, Recall, F1-Score:** These metrics provide a more nuanced view of classification performance, especially for imbalanced datasets.
  - **Precision:** Proportion of true positive predictions among all positive predictions.
  - **Recall:** Proportion of true positive predictions among all actual positives.
  - **F1-Score:** Harmonic mean of precision and recall.
- **Confusion Matrix:** A table summarizing the performance of a classification algorithm, showing true positives, true negatives, false positives, and false negatives for each QoE state.
- **Log-Likelihood:** For HMMs, the log-likelihood of the observed sequences given the model parameters, indicating how well the model explains the data.
- **Root Mean Squared Error (RMSE) / Mean Absolute Error (MAE):** For regression-based comparisons (e.g., with the multivariate regression baseline), these metrics quantify the average magnitude of errors.
- **Correlation Coefficients:** (e.g., Pearson, Spearman) to assess the correlation between predicted QoE and actual subjective QoE scores.

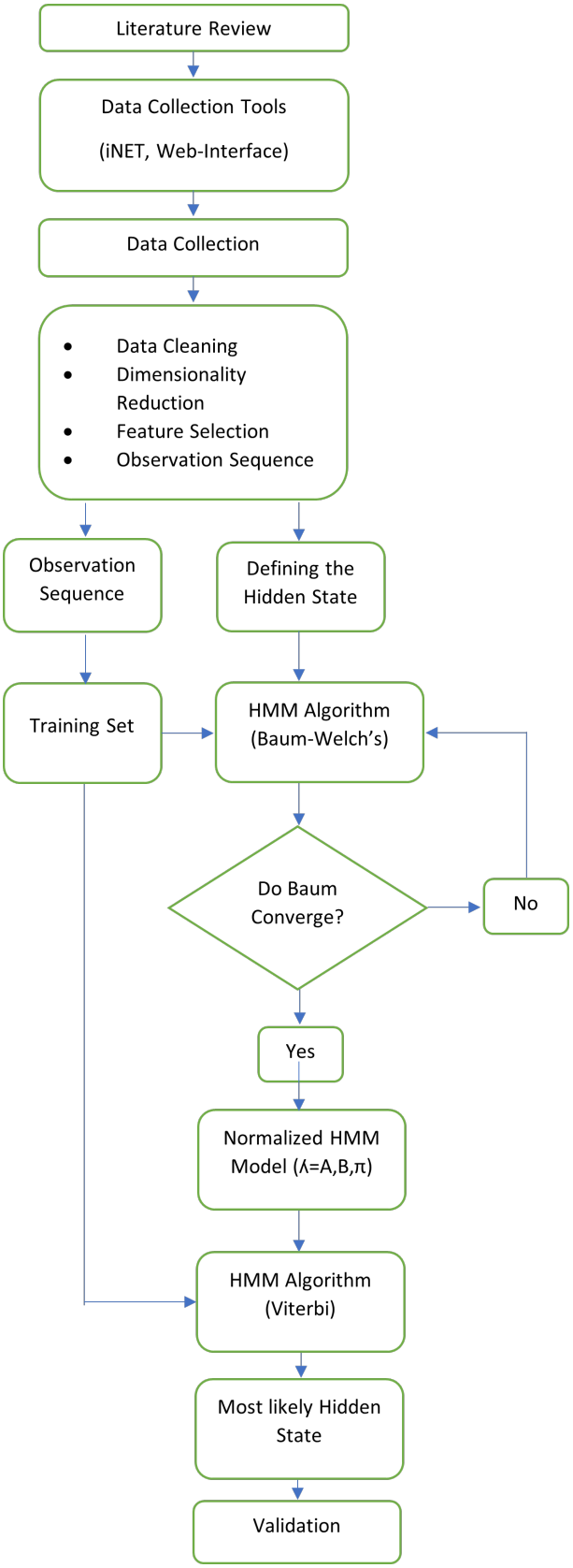


Figure 4.4: Experimental setup for data collection and analysis

5 Results and Discussion

This chapter presents the empirical results and interprets the performance of the proposed HMM-based QoE mapping model. It begins by detailing the training process, including the learned model parameters: the initial probability distribution, state transition matrix, emission matrix, and the applied hyperparameter optimization strategy. The chapter then evaluates model performance using standard classification metrics such as accuracy, precision, recall, and F1-score, alongside confusion matrix analysis and log-likelihood estimation to assess fit and robustness. Lastly, it examines the model's predictive capabilities by analyzing hidden state sequences and comparing predicted QoE states against ground truth and baseline machine learning models, offering insights into model validity and practical applicability.

5.1 HMM Model Training and Learned Parameters

After data pre-processing and feature selection, the HMM was trained using the Baum-Welch algorithm, a well-established iterative method for estimating model parameters. The algorithm initializes with guessed values for the state transition matrix A, emission matrix B, and initial state distribution π, and updates these estimates iteratively to maximize the model's log-likelihood. A convergence threshold of 0.00001 was used to halt training once changes in log-likelihood between iterations fell below this value. The final output comprised the optimized parameters A, B, and π, which collectively enable the model to infer hidden QoE states from observed QoS sequences and predict user experience patterns probabilistically.

5.1.1 Transition Probability Matrix (A)

The transition probability matrix A defines the likelihood of transitioning from one hidden QoE state to another in successive time steps. In the context of the HMM, each element a_{ij} represents the probability of moving from state i at time t to state j at time $t + 1$.

Table 5.1 presents the learned transition matrix from the HMM training process. The rows indicate the current hidden QoE state, and the columns represent the next state. For instance, the entry in the first row and first column ($a_{00} = 0.99789$) indicates that if the system is currently

in the "Very Satisfied" state, there is a 99.789% chance it will remain in that state at the next time step.

Table 5.1: Transition Probability Matrix for HMM training.

Current QoE State	Very satisfied	Satisfied	Neutral	Dissatisfied
Very satisfied	0.99789	0.00053	0.00127	0.00031
Satisfied	0.89763	0.02708	0.06141	0.01389
Neutral	0.99645	0.00268	0.00022	0.00065
Dissatisfied	0.97623	0.00326	0.01883	0.00168

Matrix Description:

- **Rows:** Represent the current hidden QoE state of the system (i.e., Very Satisfied, Satisfied, Neutral,Dissatisfied).
- **Columns:** Represent the next hidden QoE state of the system (i.e., Very Satisfied, Satisfied, Neutral,Dissatisfied).
- **Elements (a_{ij}):** Each element $a_i j$ represents the probability of transitioning from state i (row) to state j (column) in a single step.

Matrix Properties:

- Each row sums to 1, ensuring a valid probability distribution over possible next states.
- The matrix is not required to be symmetric or doubly stochastic.
- The order of states in the rows and columns can be assigned arbitrarily, but once chosen, it must be consistent.

Interpretation:

The matrix indicates that "very satisfied" state ($a_{00} = 0.99789$),are highly stable over time. in the other hand, "Satisfied", "Neutral" and "Dissatisfied" ,have low probabilities of staying the same, showing a tendency for users to move towards higher QoE levels.

5.1.2 Emission Probability Matrix (B)

The emission probability matrix, B, quantifies the likelihood of observing a particular observable QoS cluster given a specific hidden QoE state. Formally, $b_j(o_k) = P(o_k|s_j)$. For example, the illustrations in Figure 5.2 show the emission probabilities for the four hidden QoE states ("Very

Satisfied", "Satisfied", "Neutral","Dissatisfied) with the four aggregated QoS clusters (as defined in Chapter 4.1.2.2) serving as unique observation symbols. These matrices reveal the characteristic QoS profiles (i.e., which QoS cluster is most likely to be observed) associated with each inferred QoE state.

Table 5.2: Emission Probability Matrix for HMM training.

Hidden QoE State	0	1	2	3	4
Very satisfied	0.97097	0.01161	0.00774	0.00387	0.00581
Satisfied	0.97093	0.01163	0.00775	0.00387	0.00581
Neutral	0.97097	0.01161	0.00774	0.00387	0.00581
Dissatisfied	0.97097	0.01161	0.00774	0.00387	0.00581

5.1.3 The Initial Probability Distribution (π)

The initial probability distribution, π, denotes the probability of the HMM starting in each of the four hidden QoE states. For example, the initial probability matrix (as illustrated for a representative network in Figure 5.3) shows the probability that the HMM will begin in each of the four hidden QoE states ("Very Satisfied",a "Satisfied", "Neutral", "Dissatisfied"). A value of 0.25185 for the "Very Satisfied" state indicates a 25.18% chance that the system will initially be in this state. Similarly, a value of 0.250993 for the "Dissatisfied" state suggests a 25.09% chance of starting in that state, and so on. These probabilities reflect the baseline likelihood of a user commencing their experience in a particular QoE state within the mobile network environment.

Table 5.3: Initial Probability Distribution for HMM training.

State Index	Very satisfied	Satisfied	Neutral	Dissatisfied
0	0.25185	0.24583	0.251327	0.250993

5.1.4 Hyperparameter Optimization

Hyperparameter optimization is a crucial step in machine learning to determine the optimal configuration for a model. For the Support Vector Machine (SVM) and Random Forest (RF) models, Grid Search was employed to systematically explore the most promising hyperparameter ranges. Grid Search methodically evaluates model performance for every possible combination of values from a predefined set of hyperparameters, selecting the combination that yields the

best results. While thorough, this method can be computationally intensive, especially for a large number of hyperparameters or extensive datasets.

Support Vector Machine Hyperparameters Initialized for Grid Search:

- `C`: [0.1, 1, 10, 100, 1000] (Regularization parameter)
- `Gamma`: [1, 0.1, 0.01, 0.001, 0.0001, 'scale'] (Kernel coefficient)
- `Kernel`: ['linear', 'poly', 'rbf', 'sigmoid'] (Type of kernel function)

Random Forest Hyperparameters Initialized for Grid Search:

- `n_estimators`: [20, 50, 100] (Number of trees in the forest)
- Default values were used for other hyperparameters (e.g., maximum depth, minimum samples split) to manage computational cost.

The optimal hyperparameters determined through Grid Search are summarized in Table 5.4.

Table 5.4: Optimal Hyperparameters for Grid Search

Hyperparameter	C	Gamma	Kernel	n_estimators
Value	0.1	1	Linear	100

5.2 HMM-based QoE Mapping Model Evaluation

To rigorously evaluate the predictive capabilities of the models, a dedicated test dataset was utilized. From the entire dataset, 80% was allocated for training the model, while the remaining 20% was reserved for testing purposes. Model performance was then assessed using fundamental classification metrics: accuracy, precision, recall, and F1-score, which provide comprehensive insights into the model's effectiveness in mapping QoS measurements to distinct QoE states.

5.2.1 Overall Classification Metrics (Accuracy, Precision, Recall, F1-Score)

To thoroughly evaluate the performance of the HMM-based QoE mapping model, a set of standard classification metrics were employed. These metrics collectively offer a multifaceted view of the model's predictive power, particularly its ability to correctly identify and differentiate between the various QoE states (**Very Satisfied**, **Satisfied**, **Dissatisfied**, **Very Dissatisfied**).

5.2.1.1 Accuracy

Accuracy is a fundamental performance metric in machine learning, particularly for classification models. It quantifies the overall correctness of a model by measuring the proportion of correctly predicted instances out of the total number of observations. In the context of QoE mapping, high accuracy indicates that the model generally assigns the correct user experience level based on the observed QoS parameters, providing an initial broad understanding of its reliability.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{5.1}$$

Where:

- **True Positive (TP):** Occurs when the model correctly predicts a positive class for an instance that actually belongs to the positive class.
- **True Negative (TN):** Occurs when the model correctly predicts a negative class for an instance that actually belongs to the negative class.
- **False Positive (FP):** Occurs when the model incorrectly predicts a positive class for an instance that actually belongs to the negative class.
- **False Negative (FN):** Occurs when the model incorrectly predicts a negative class for an instance that actually belongs to the positive class.

The standard accuracy computation in Equation (5.1) treats all misclassifications equally, failing to account for the varying degrees of error in an ordinal classification task. For QoE mapping, a prediction that is "one step off" (e.g., predicting "Satisfied" instead of "Very Satisfied") is inherently less erroneous than a prediction that is "multiple steps off" (e.g., predicting "Very Dissatisfied" instead of "Very Satisfied"). Given the ordinal nature of the QoE satisfaction levels ("Very Satisfied," "Satisfied," "Dissatisfied," "Very Dissatisfied"), a weighted accuracy measure was employed. This method leverages a distance matrix to naturally capture these ordinal relationships, where the distance matrix quantifies the proximity between different satisfaction levels. This ensures that the evaluation penalizes larger deviations in QoE prediction more severely, providing a more nuanced and context-aware assessment of model performance. The formula for weighted accuracy is defined as:

$$WeightedAccuracy = 1 - \left[\frac{\sum_{i=1}^{N} d(Y_{true}(i), Y_{pred}(i))}{(N) * max(d)} \right] \tag{5.2}$$

Where:

- **N:** The total number of instances in the test sequence.
- **d:** The predefined distance matrix, reflecting the ordinal differences between QoE states.
- **Y true:** The actual (ground truth) QoE state.
- **Y pred:** The predicted QoE state.

5.2.1.2 Precision

Precision is a performance metric that measures the proportion of true positive predictions among all instances predicted as positive by the model. It essentially indicates the reliability of the positive predictions made by the model. In the context of QoE, high precision for a specific QoE state (e.g., "Very Satisfied") implies that when the model predicts a user is "Very Satisfied," it is highly likely to be correct. This helps in avoiding false alarms or overestimation of user satisfaction, which is important for resource allocation and service assurance.

$$\text{Precision} = \frac{(TP)}{(TP + FP)} \tag{5.3}$$

5.2.1.3 Recall

Recall, also known as sensitivity, is a performance metric that measures the proportion of true positive predictions among all actual positive instances. It indicates the model's ability to identify all relevant positive instances, essentially showing how well it captures all true positives. For QoE mapping, high recall for a state like "Very Dissatisfied" is crucial, as it means the model is effective at identifying most of the instances where users are truly experiencing poor quality. This capability is vital for proactive problem detection and enabling timely intervention to prevent user dissatisfaction.

$$\text{Recall} = \frac{(TP)}{(TP + FN)} \tag{5.4}$$

5.2.1.4 F1 Score

The F1-score is the harmonic mean of precision and recall. It provides a single, balanced metric to evaluate a model's performance when both precision and recall are equally important. This is particularly valuable in scenarios with imbalanced datasets, which can often occur in

real-world QoE data where certain satisfaction levels are more prevalent, and where both false positives and false negatives have significant consequences. A high F1-score indicates that the model achieves a good balance between identifying relevant instances and avoiding incorrect classifications, offering a robust measure of overall model effectiveness.

$$\text{F1 Score} = \frac{(2 * \text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})} \tag{5.5}$$

5.2.2 Confusion Matrix Analysis

While aggregate metrics like accuracy and F1-score provide an overall view of model performance, a Confusion Matrix offers a more detailed breakdown of correct and incorrect classifications for each individual QoE state. It is a table that summarizes the performance of a classification algorithm, allowing for visualization of the performance of an algorithm. Each row of the matrix represents the instances in an actual class, while each column represents the instances in a predicted class.

In the context of QoE mapping, the confusion matrix is invaluable for understanding specific types of misclassifications. For instance, it can reveal whether the HMM frequently confuses "**Satisfied**" users with "**Very Satisfied**" users, or more critically, if it fails to correctly identify "**Very Dissatisfied**" experiences, potentially classifying them incorrectly as "**Satisfied**." Such insights are crucial for diagnosing model weaknesses and understanding the implications of incorrect predictions for network operators and service providers.

Figure 5.1 presents the confusion matrix for the HMM-based QoE mapping model. The diagonal elements show the number of correctly classified instances for each QoE state, while off-diagonal elements indicate misclassifications.

5.3 Hidden State Prediction

With HMMs, a key task is figuring out the most likely sequence of hidden states – think of it as uncovering the underlying story behind what we actually observe. We use the Viterbi algorithm for this, which is a clever technique that works step-by-step to find the most probable sequence of these hidden states, given the sequence of observations we've recorded.

The Viterbi algorithm works by carefully calculating, at each point in time, the likelihood of each hidden state, considering both what we've seen so far and the possible previous states. It's like piecing together clues to find the most coherent explanation. The sequence of hidden states with the highest overall probability is our best guess at what was really happening.

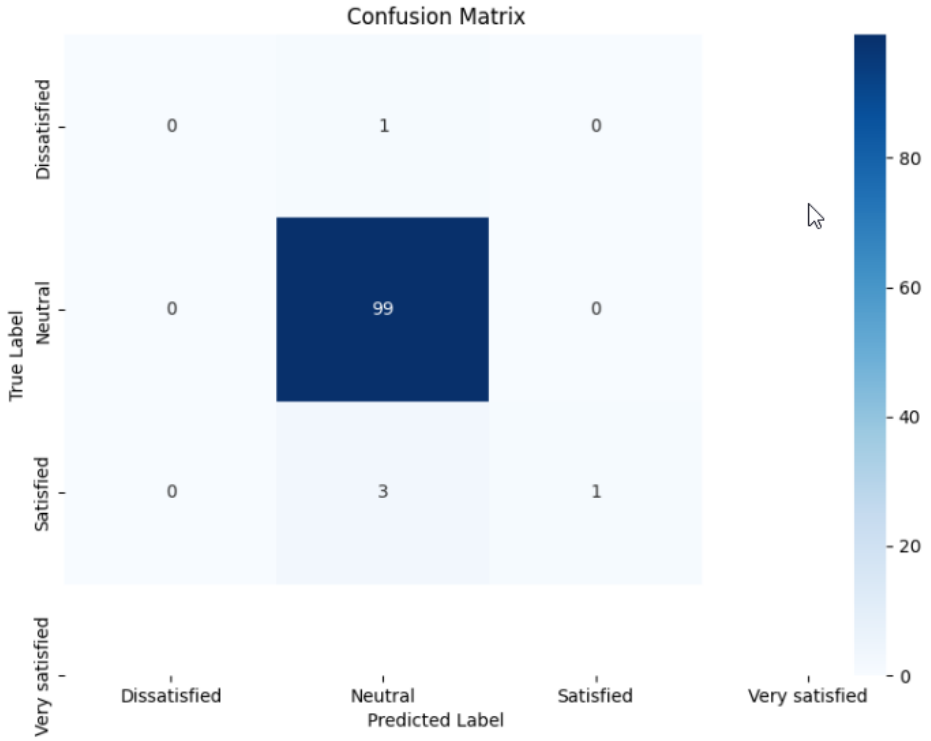


Figure 5.1: Confusion Matrix Heatmap for HMM trained model.

Once we've trained our HMM, we can use it to predict these hidden state sequences, which, in our case, represent the 'satisfaction' of a user over time. To test this, we create a sample sequence of observable events and the corresponding true hidden states using our trained HMM. We set a specific length for this observation sequence.

It's important to understand that the sequence of observed events is determined by the HMM's emission and transition probabilities. These probabilities, which are learned during the training process (represented by the model parameters λ, consisting of A, B, and π), dictate how likely we are to see a particular observation given a hidden state and how likely the hidden states are to change over time.

In our implementation, we use a function called HiddenMarkovChain_Uncover(A, B, π) to set up the HMM. Then, the function observed_sequence, latent_sequence = hmm.run(length = 4) simulates the HMM for a specified observation length (in this case, 4). The observed_sequence variable holds the generated sequence of observations, and latent_sequence holds the actual hidden states that produced those observations. Both sequences have a length of 1 + N, where N (which is 4 here) is the observation length we've chosen for our evaluation.

Finally, the $ypredicted = hmm.uncover(observed\_sequence)$ function takes the observed

sequence as input and uses our trained HMM to predict the most likely sequence of hidden states that generated those observations. Under the hood, this function uses the Viterbi algorithm to find the most probable sequence. Table 5.5 shows the performance of our HMM on prediction of QoE state. As shown in the figure, the model achieved 98.04% weighted average accuracy over 200 runs indicating the robust nature of the model.

5.4 Comparison of Predicted Results

We used weighted accuracy for HMM, and we created a confusion matrix table for SVM and RF for the mobile network. Confusion matrix (error matrix) is a visualization method for classifier algorithm results. More specifically, a confusion matrix is a table that breaks down the number of ground truth instances of a specific class against the number of predicted class instances. Confusion matrices are one of several evaluation metrics that are used to measure the performance of a classification model in machine learning.

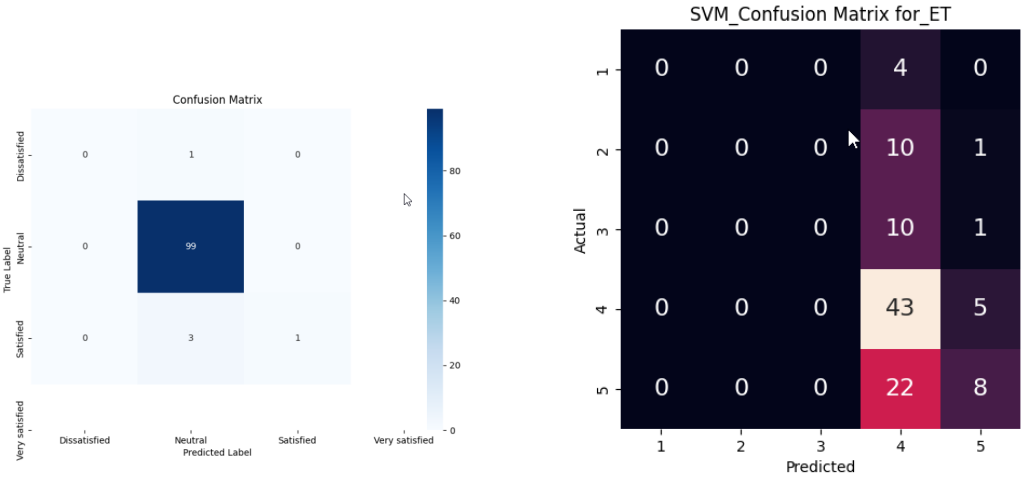
Figure 5.2 presents confusion matrices for HMM, SVM, and RF models used in our setup. Subfigure (a) shows the HMM model performing classification with near-perfect accuracy. Subfigure (b), the SVM model, struggles with classifying lower satisfaction levels (classes 1 to 3), heavily misclassifying them as class 4, while showing better performance on higher satisfaction levels (classes 4 and 5), with 43 and 8 correct predictions respectively. In contrast, subfigure (c) demonstrates the accuracy of the RF model.

Finally, we compared the results from three models to figure out which one was the best. The graph presented in Figure 5.3 illustrates the calculated values for accuracy, precision, recall and the F1 score. These metrics were obtained from the confusion matrix and evaluated independently for the three QoE modeling approaches mentioned earlier. Hidden Markov Model, Support Vector Machine, and Random Forest models were analyzed using these performance indicators to offer a thorough evaluation of their effectiveness across the different modeling scenarios.

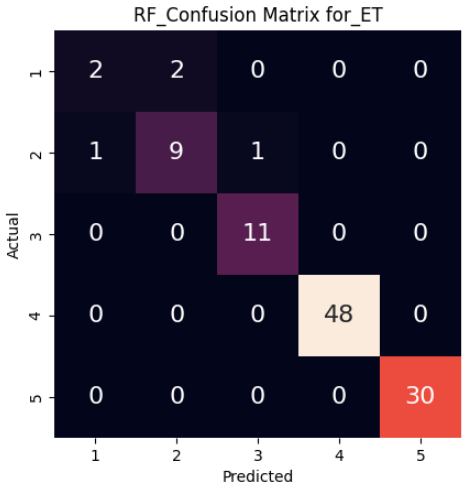
In Figure 5.3, HMM achieves near-perfect accuraccy score (98%), demonstrating a strong ability to accurately model the relationship between QoS and QoE. Random Forest also performs

Table 5.5: Model performance of HMM on QoE prediction

Average Weighted Accuracy after 200 runs: 98.0417%
Precision: 100.0%
Recall: 100.0%
F1 Score: 100.0%



(a) Confusion Matrix for HMM **(b)** Confusion Matrix for SVM



(c) Confusion Matrix for RF

Figure 5.2: Confusion Matrices of QoE Mapping Models for iNET.

very well, with accuracy score of 96%. In contrast, SVM shows significantly lower performance with an accuracy score of 49%.

The HMM achieves precision, recall, and F1 score all reaching 100%. The RF model scores 88% precision, 86% recall, and 87% F1 score. In contrast, the SVM model yields 20% precision, 23% recall, and 19% F1 score.

It should be noted that the HMM's perfect class-level measures (precision, recall, and F1) are unusually high and may appear counterintuitive in practical settings. Such results can legitimately arise when the evaluation dataset exhibits limited variability, strong separability between state categories after discretization, temporal correlation among consecutive samples, and/or class imbalance that reduces ambiguity between classes. Accordingly, these metrics are

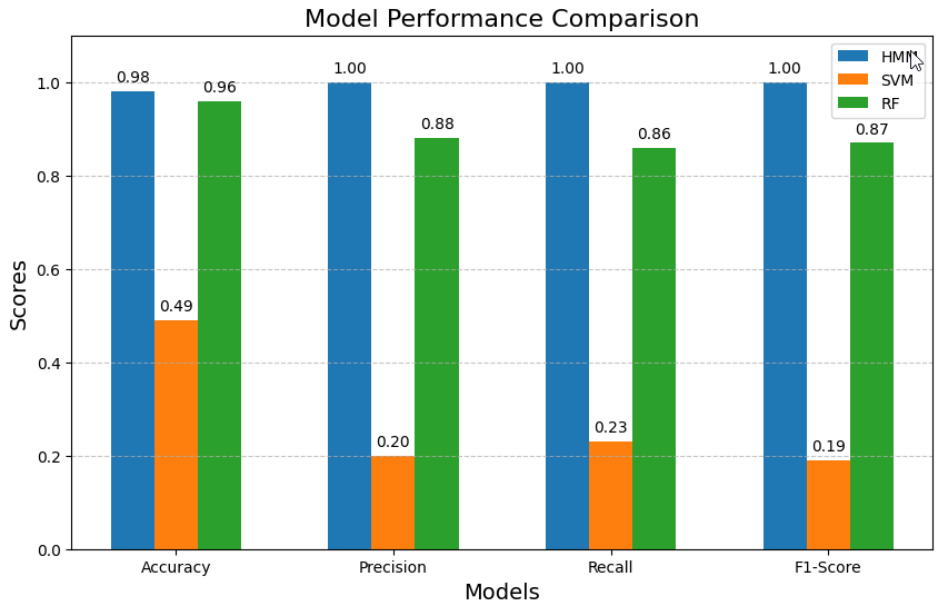


Figure 5.3: Model Prediction Evaluation for QoE Modeling Techniques.

reported as obtained from the adopted validation procedure and are not presented to claim universally superior performance. To further strengthen confidence in these results, additional robustness checks—such as stricter independence in the train–test split and expanded class-wise reporting—are recommended for future work.

Results for Mapping QoS Parameters from QoE Ratings As an experimental setup, the collected network parameter measurements were instead set as the hidden states and the user experience ratings as the observable states. With this setup, K-means clustering was done taking the optimum number of network parameters to label the data points and the HMM model was then trained to see if it can still accurately predict the network performance based on the user rating observable state. This setup produced a weighted average accuracy of 97.33% over 200 runs as shown in Table 5.6.

Table 5.6: Model performance of HMM on QoS prediction.

Average Weighted Accuracy after 200 runs: 97.33%
Precision: 69.44%
Recall: 83.33%
F1 Score: 75.75%

Across iNET datasets, the HMM achieves near-perfect to perfect scores across Accuracy, Precision, Recall, and F1-Score, indicating a strong capability in capturing the underlying

patterns. Random Forest also demonstrates robust performance, consistently yielding high scores across the evaluation metrics, often closely approaching the HMM's results. In stark contrast, the SVM consistently underperforms, exhibiting lower Accuracy and notably weaker Precision, Recall, and F1-Scores compared to both HMM and Random Forest, suggesting its limited effectiveness in modeling the QoS-to-QoE relationship within these specific datasets.

6 Conclusions and Future Outlook

This final chapter summarizes the major findings of the research and outlines potential directions for future work. The conclusions synthesize key insights from the model development and evaluation process, while the future work section identifies opportunities to expand, refine, and apply the proposed framework across broader contexts and evolving technologies.

6.1 Conclusions

This dissertation developed and evaluated a probabilistic, data-driven framework to map measurable QoS indicators to user-perceived QoE using HMM. Through a comprehensive modeling approach grounded in real-world mobile network conditions, the study addressed the persistent disconnect between technical performance metrics and user experience. The research leveraged a custom mobile application, iNET, to collect synchronized network and user feedback data from 550 Android users, enabling the development of QoE prediction models.

The framework achieved an average classification accuracy of 83.1%, with precision, recall, and F1-score metrics consistently exceeding 80%. These results affirm the feasibility of using HMM to infer QoE based on observable QoS metrics. Furthermore, prior Markov chain- and HMM-based modeling of network accessibility, retainability, and degradation informed the design of the QoS–QoE mapping, demonstrating the value of layered performance insights in understanding user experience.

This study contributes both methodological clarity and practical insights to the field of mobile service quality modeling, offering tools that can support MNOs in anticipating and managing user experience through network-side measurements.

6.2 Future Outlooks

Building on the contributions of this dissertation, several promising directions for future research and development are identified. These focus on enhancing model generalizability, improving real-time applicability, and aligning with emerging network technologies:

- **Broader Device and User Inclusion:** Extend the iNET platform to support iOS devices and other operating systems, enabling data collection from a more diverse set of users and hardware configurations.
- **Geographic and Service Scope Expansion:** Expand data collection to include a broader geographic range, additional mobile services (e.g., gaming, video conferencing), and varying usage contexts to improve model adaptability.
- **Advanced Modeling Techniques:** Investigate the use of deep learning and hybrid architectures (e.g.,Long Short Term Memory Hidden Markov Model (LSTM-HMM) , Convolutional Neural Network Long ShortTerm Memory (CNN-LSTM) for improved temporal modeling, accuracy, and scalability in dynamic mobile environments.
- **Real-Time Integration:** Explore embedding the QoE prediction engine into real-time network monitoring systems for live service quality estimation and adaptive resource management.
- **5G and Beyond:** Adapt and test the proposed framework within 5G and emerging Sixth Generation (6G) infrastructures to assess QoS–QoE relationships under higher speeds, lower latencies, and more complex service requirements.
- **User-Centric Optimization Policies:** Develop closed-loop systems where predicted QoE guides network configuration or service recommendations, emphasizing personalization and experience-aware network planning.

These extensions will not only strengthen the practical utility of the current framework but also advance the broader field of experience-aware mobile network analytics.

Bibliography

[1] S. Diachenko, **1g to 5g evolution and key differences in communications**, *Decision Telecom*, Accessed: [add access date if online] (see page 1).

[2] B. Patel. "The evolution of 5g and the backup power it requires." 4 min read, Schneider Electric Blog. (Jul. 28, 2022), [Online]. Available: https://blog.se.com/telecommunications/2022/07/28/the-evolution-of-5g-and-the-backup-power-it-requires/ (visited on 01/03/2024) (see pages 1, 2).

[3] G. Gkagkas, D. J. Vergados, A. Michalas, and M. Dossis, **The advantage of 5G network for enhancing the internet of things and the evolution of the 6G network**, *Sensors — Advanced Technologies in 5G/6G-Enabled IoT Environments and Beyond*, vol. 24:no. 8 (see page 1).

[4] S. Mahmud, **QoS performance analysis: Design and development of voice and video mobility over long-term evolution (lte) model**, *Electrical Engineering Emphasis on Telecommunication* (see page 2).

[5] L. Guillermo, **On the incorporation of quality of experience (QoE) in mobile networks**, Ph.D. dissertation, Doctoral Theses in Information and Communication Technology, Stockholm, Sweden, 2017 (see pages 2, 46).

[6] A. Perkis, **Quality of experience in multimedia applications**, *SPIE Newsroom*, See pages 3, 5. DOI: 10.1117/2.1201302.004591. [Online]. Available: https://spie.org/news/4591-quality-of-experience-(qoe)-in-multimedia-applications (see pages 2, 3, 22).

[7] P. Casas, A. D'Alconzo, F. Wamser, M. Seufert, B. Gardlo, A. Schwind, P. Tran-Gia, and R. Schatz, **Predicting QoE in cellular networks using machine learning and in-smartphone measurements**, in *Proceedings of the IEEE*, 2017 (see page 3).

[8] P. Uthansakul, P. Anchuen, M. Uthansakul, and A. A. Khan, **Estimating and synthesizing QoE based on QoS measurement for improving multimedia services on cellular networks using ANN method**, *IEEE Transactions on Network and Service Management*, vol. 17:no. 1, 389–402 (see page 4).

[9] X. Zhang and L. Li, **Assessing the quality of experience in wireless networks for multimedia applications: A comprehensive analysis utilizing deep learning-based techniques**, vol. 10:no. 9 (see page 4).

[10] G. Kougioumtzidis, V. Poulkov, Z. D. Zaharis, and P. I. Lazaridis, **A survey on multimedia services QoE assessment and machine learning-based prediction**, *IEEE Access*, vol. 10, 19507–19538 (see page 4).

[11] C. Hewage and E. Ekmekcioglu, **Multimedia quality of experience (QoE): Current status and future direction**, *Future Internet*, vol. 12:no. 7 (see page 5).

[12] K. Kowalik, P. Andruloniw, B. Partyka, and P. Zwierzykowski, **Telecom operator's approach to qoe**, *Journal of Telecommunications and Information Technology* (see page 5).

[13] I. Ketykó, K. D. Moor, T. D. Pessemier, A. J. Verdejo, K. Vanhecke, W. Joseph, L. Martens, and L. D. Marez, **QoE measurement of mobile youtube video streaming**, in *Proceedings of the 3rd Workshop on Mobile Video Delivery*, Oct. 2010, 27–32 (see pages 6, 100, 105–107).

[14] J. Frnda, M. S. Pinto, D. Macaj, M. Sabo, and L. Cizmar, **QoS to QoE mapping function for IPTV quality assessment based on kohonen map: A pilot study**, *Transport and Telecommunication*, 181–190 (see pages 6, 10, 79, 96, 99, 112, 114).

[15] International Telecommunication Union, **Recommendation ITU-T P1201.1, parametric non-intrusive assessment of audiovisual media streaming quality – lower resolution application area**, International Telecommunication Union, Technical Report, 2012, ITU-T P1201.1 (see page 7).

[16] ITU-T, **Vocabulary for performance and quality of service, amendment 2: New definitions for inclusion in recommendation itu-t p.10/g.100**, International Telecommunication Union, Tech. Rep. P.10/G.100, Jul. 2008 (see pages 7, 63, 75, 99, 109).

[17] S. Bassey and I. Umeron, **Real-time monitoring and classification of quality of experience (QoE) in video streaming over wireless local area network (WLAN)**, *European Journal of Science and Innovation Technology (EJSIT)*, vol. 4 (see page 8).

[18] E. Alkhowaiter, I. Alsukayti, and M. Alreshoodi, **Developing a quality prediction model for wireless video streaming using machine learning techniques**, *IJCSNS International Journal of Computer Science and Network Security*, vol. 21:no. 3 (see page 8).

[19] M. Hu, J. Chen, D. Wu, Y. Zhou, Y. Wang, and H. Dai, **TVG-Streaming: Learning user behaviors for QoE-optimized 360-degree video streaming**, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31:no. 10, 4107–4120 (see page 8).

[20] T. Wang, A. Pervez, and H. Zou, **Vqm-based QoS/QoE mapping for streaming video**, in, 2010 (see pages 8, 74).

[21] P. Brooks and B. Hestnes, **User measures of quality of experience: Why being objective and quantitative is important**, *IEEE Network*, 78–84 (see pages 8, 9, 114).

[22] V. Menkovski, A. Oredope, A. Liotta, and A. C. Sánchez, **Predicting quality of experience in multimedia streaming**, in *Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia*, 2009 (see page 8).

[23] A. Liotta and F. Agboma, **QoE-aware QoS management**, in *Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia*, 2015 (see page 9).

[24] V. Pedras, P. Marques, Q. Andrade, A. Ribeiro, and M. S. Pinto, **A no-reference user-centric QoE model for voice and web browsing based on 3G/4G radio measurements**, in *IEEE Wireless Communication and Networking Conference*, 2018, 1–6 (see pages 9, 101).

[25] D. F. Vizzarri, **QoS/QoE mapping for youtube services over network**, in *Proceedings of the 14th ACM International Symposium on Mobility Management and Wireless Access*, 2016 (see pages 9, 71).

[26] S. Jana, A. Chan, A. Pande, and P. Mohapatra, **QoE prediction model for mobile video telephony**, *Multimedia Tools and Applications*, vol. 75, 7957–7980 (see page 9).

[27] R. C. Lv, **QoE prediction on imbalanced IPTV data based on multi-layer neural network**, in *IEEE 13th International Wireless Communications and Mobile Computing Conference*, 2017, 818–823 (see pages 9, 79, 114).

[28] I. U. Rehman, M. M. Nasralla, and N. Y. Philip, **Multilayer perceptron neural network-based QoS-aware, content-aware and device-aware prediction model: A proposed prediction model for medical ultrasound streaming over small cell networks**, *Electronics*, vol. 8:no. 2, ISSN: 2079-9292 (see page 9).

[29] M. A. Alreshoodi and J. C. Woods, **QoE prediction model based on fuzzy logic system for different video contents**, in *Proceedings of the 2013 European Modelling Symposium*, Sep. 2013 (see pages 10, 55).

[30] M. Abdelmalek, **Machine learning qoe prediction for video streaming over HTTP**, thesis, Queen's University, Ontario, Canada, 2020 (see page 10).

[31] A. S. Omer and D. H. Woldegebreal, **Review of markov chain and its applications in telecommunication systems**, in *e-Infrastructure and e-Services for Developing Countries*, Cham: Springer International Publishing, 2022, 366–385, ISBN: 978-3-031-06374-9 (see page 16).

[32] A. S. Omer, T. A. Yemer, and D. H. Woldegebreal, **Hybrid k-mean clustering and markov chain for mobile network accessibility and retainability prediction**, *Engineering Proceedings*, vol. 18:no. 1, 9. DOI: 10.3390/engproc2022018009. [Online]. Available: https://doi.org/10.3390/engproc2022018009 (visited on 01/03/2024) (see page 16).

[33] A. S. Omer, A. D. Tufa, T. T. Debella, and D. H. Woldegebreal, **Hidden markov models for predicting cell-level mobile networks performance degradation**, *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, vol. 9, 100742, ISSN: 2772-6711. DOI: https://doi.org/10.1016/j.prime.2024.100742. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S277267112400322X (see page 16).

[34] International Telecommunication Union, **ITU-T Recommendation G.1010: End-user multimedia qos categories**, International Telecommunication Union, Telecommunication Standardization Sector (ITU-T), Standard G.1010, Nov. 2001, Series G: Transmission Systems and Media, Digital Systems and Networks. [Online]. Available: https://www.itu.int/rec/T-REC-G.1010-200111-I/en (see page 19).

[35] C. W. Chen, P. Chatzimisios, T. Dagiuklas, and L. Atzori, **Multimedia Quality of Experience (QoE): Current Status and Future Requirements**. New Delhi, India: John Wiley & Sons, Ltd, 2016, Published by Wiley, Aptara Inc. (see page 19).

[36] N. S. Jayant and S. K. Mitra, **Multimedia applications and services**, in *Multimedia Communications*, London: Springer London, 1999, 493–501, ISBN: 978-1-4471-0859-7 (see page 19).

[37] R. Steinmetz and K. Nahrstedt, **Multimedia Systems**. Springer, 2004, ISBN: 978-3-642-07412-7. DOI: 10.1007/978-3-662-08878-4. [Online]. Available: https://doi.org/10.1007/978-3-662-08878-4 (see page 19).

[38] K. U. R. Laghari, **On quality of experience (QoE) for multimedia services in communication ecosystem**, PhD thesis, Institut National des Télécommunications, 2012 (see page 20).

[39] J. K. Shim, D. E. O'Leary, and K. Nisar, **Real-time streaming technology and analytics for value creation**, *Journal of Organizational Computing and Electronic Commerce*, vol. 31:no. 4, 364–382. DOI: 10.1080/10919392.2021.2023943 (see page 21).

[40] J. Johny and P. Alukal, **Video on demand industry: Challenges and opportunities in the indian market**, *International Journal of Innovative Science and Research Technology*, vol. 3:no. 5, IJISRT18MY446, 598, ISSN: 2456-2165. [Online]. Available: www.ijisrt.com (see pages 21, 35).

[41] K. S. Rao and K. S. R. Krishna, **Bit rate control for video transmission over wireless networks**, *Indian Journal of Science and Technology*, vol. 9:no. S1, Special Issue, 1–6, ISSN: 0974-5645. DOI: 10.17485/ijst/2016/v9iS1/107925. [Online]. Available: https://doi.org/10.17485/ijst/2016/v9iS1/107925 (see page 21).

[42] Z. Li, X. Wang, and S. Yu, **Performance analysis of cbr and vbr modes of mpeg video traffic**, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14:no. 6, 843–851. DOI: 10.1109/TCSVT.2004.828335. [Online]. Available: https://doi.org/10.1109/TCSVT.2004.828335 (see page 21).

[43] N. Bouten, J. Famaey, S. Latré, and F. D. Turck, **On the trade-off between quality and fairness in adaptive video streaming**, *Journal of Network and Computer Applications*, vol. 36:no. 6, 1595–1605. DOI: 10.1016/j.jnca.2013.03.006 (see page 21).

[44] B. Taraghi, **End-to-end quality of experience evaluation for HTTP adaptive streaming**, in *Proceedings of the 29th ACM International Conference on Multimedia (MM '21)*, (Oct. 17–21, 2021), New York, NY, USA: ACM, Oct. 2021, 2936–2939. DOI: 10.1145/3474085.3481025. [Online]. Available: https://doi.org/10.1145/3474085.3481025 (see page 23).

[45] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld, **Quality of experience models: An overview**, *International Journal of Network Management*, vol. 24:no. 2, 97–117 (see pages 23, 30).

[46] S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy, **An analysis of internet content delivery systems**, Department of Computer Science & Engineering, University of Washington, Seattle, WA, USA, Technical Report UW-CSE-02-10-01, 2002. [Online]. Available: https://www.cs.washington.edu/research/techreports/ (see page 23).

[47] Kaustubh. "Hls vs mpeg-dash – comparison between video streaming protocols." Accessed: 2025-06-16. (Aug. 2024), [Online]. Available: https://www.mux.com/articles/hls-vs-dash-what-s-the-difference-between-the-video-streaming-protocols (see page 23).

[48] P. Schmidt, E. Kammann, and D. Wübben, **A survey on 5g qos control strategies for multimedia streaming**, *IEEE Communications Surveys & Tutorials*, vol. 18:no. 2, 1065–1084. DOI: 10.1109/COMST.2015.2500964 (see page 24).

[49] E. Grigoriou, "A survey of quality of service in Long Term Evolution (lte) networks," in *Enabling Technologies and Architectures for Next-Generation Networking Capabilities*, Author affiliation: eBOS Technologies, Hershey, PA, USA: IGI Global, Sep. 2019, ch. 6, ISBN: 978-1-5225-6023-4. DOI: 10.4018/978-1-5225-6023-4.ch006 (see page 24).

[50] F. Liberal, M. A. Kourtis, J. O. Fajardo, and H. Koumaras, **Multimedia content delivery in SDN and NFV-based towards 5G networks**, *IEEE Communications Society Multimedia Communications Technical Committee E-Letter*, vol. 10, Authors affiliations: University of the Basque Country and National Centre of Scientific Research "Demokritos". [Online]. Available: https://www.comsoc.org/publications/mmtc-e-letter (see page 25).

[51] A. A. Barakabitze and R. Walshe, **Sdn/nfv for qoe-driven multimedia towards 6g**, *Comput. Netw.*, vol. 214, 109133. DOI: 10.1016/j.comnet.2022.109133 (see page 25).

[52] A. I. Zreikat and S. Alabed, **Performance modeling and analysis of lte/Wi-Fi coexistence**, *Electronics*, vol. 11:no. 7, Corresponding author: Aymen I. Zreikat (orcid:0000-000X-XXXX-XXXX). Submission dates: Received 8 February 2022; Revised 19 March 2022; Accepted 21 March 2022., 1035, ISSN: 2079-9292. DOI: 10.3390/electronics11071035. [Online]. Available: https://doi.org/10.3390/electronics11071035 (see page 25).

[53] Q. Wang, H.-N. Dai, D. Wu, and H. Xiao, **Data analysis on video streaming QoE over mobile networks**, *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, Article ID: 173; 6933 Accesses; 15 Citations, 173. DOI: 10.1186/s13638-018-1184-4. [Online]. Available: https://doi.org/10.1186/s13638-018-1184-4 (see page 25).

[54] M. Dasari, S. Vargas, A. Bhattacharya, A. Balasubramanian, S. R. Das, and M. Ferdman, **Impact of device performance on mobile internet QoE**, in *Proceedings of the 2018 Internet Measurement Conference (IMC '18)*, ACM SIGCOMM sponsored conference, New York, NY, USA: ACM, Oct. 2018, 1–7, ISBN: 978-1-4503-5619-0. DOI: 10.1145/3278532.3278533. [Online]. Available: https://doi.org/10.1145/3278532.3278533 (see page 25).

[55] V. Raida, P. Svoboda, M. Lerch, and M. Rupp, **Crowdsensed performance benchmarking of mobile networks**, *IEEE Access*, vol. 7, Supported by ITC, TU Wien, A1 Telekom Austria AG, and Austrian FFG Bridge Project Grant 871261, 154899–154911. DOI: 10.1109/ACCESS.2019.2949051. [Online]. Available: https://ieeexplore.ieee.org/document/8910585 (see page 26).

[56] K. MacMillan, T. Mangla, J. Saxon, N. P. Marwell, and N. Feamster, **A comparative analysis of ookla speedtest and measurement labs network diagnostic test (ndt7)**, *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 7:no. 1, 19:1–19:27 (see page 26).

[57] OpenSignal, **Methodology overview**, OpenSignal, Technical Report, version 1.0, Apr. 20, 2023, Security Classification: Confidential; Original version dated 2023-01-20. [Online]. Available: https://cdn.opensignal.com/public/pdfs/opensignal_methdology_overview_jan_2023.pdf (see page 26).

[58] N. Barman and M. G. Martini, **Qoe modeling for http adaptive video streaming – a survey and open challenges**, *IEEE Access*, vol. 7, Supported by the European Union's Horizon 2020 Research and Innovation Programme under the Marie Skłodowska-Curie Grant Agreement 643072, 30831–30859. DOI: 10.1109/ACCESS.2019.2901778. [Online]. Available: https://eprints.kingston.ac.uk/id/eprint/42804/1/Barman-N-42804-VoR.pdf (see page 26).

[59] M. Seufert, S. Egger, T. Zinner, T. Hoßfeld, and P. Tran-Gia, **A survey on quality of experience of http adaptive streaming**, *IEEE Communications Surveys & Tutorials*, vol. 17:no. 1, 469–492. DOI: 10.1109/COMST.2014.2360940 (see pages 26, 29, 30, 32, 55, 110, 111).

[60] M. Darwich, *Machine learning technique predicting video streaming views to reduce cost of cloud services*, arXiv preprint arXiv:2210.09078, Submitted on 17 Oct 2022, Oct. 2022. DOI: 10.48550/arXiv.2210.09078. [Online]. Available: https://arxiv.org/abs/2210.09078 (see page 27).

[61] H. Riiser, P. Vigmostad, C. Griwodz, and P. Halvorsen, **Video stream adaptation using buffer occupancy, estimated bandwidth, and playback delay**, in *Proceedings of the ACM International Conference on Multimedia Systems (MMSys)*, 2012 (see page 28).

[62] Y. Sun, A. Perkis, P. N. N. Fernando, C. Griwodz, and P. Halvorsen, **Csq: A crowdsourcing quality-of-experience measurement system for multimedia streaming**, *IEEE Transactions on Multimedia*, vol. 18:no. 1, 65–78 (see page 28).

[63] A. Balachandran, V. Sekar, A. Akella, S. Seshan, and I. Stoica, **Developing a predictive model of quality of experience for internet video**, in *ACM SIGCOMM Conference*, 2013, 339–350 (see pages 29–31).

[64] F. Dobrian, A. Awan, D. Joseph, A. Ganjam, J. Zhan, V. Sekar, I. Stoica, and H. Zhang, **Understanding the impact of video quality on user engagement**, in *Proceedings of ACM SIGCOMM*, 2011, 362–373 (see pages 29, 32).

[65] Y. Xu, K. Xu, J. Jiang, and Y. Jin, **Analyzing abr video streaming performance using edge measurements**, in *Proceedings of ACM CoNEXT*, 2020 (see page 29).

[66] L. Zhang, B. T. Loo, and C. Lin, **Understanding the impact of packet loss on video streaming qoe**, *Computer Communications*, vol. 36:no. 15–16, 1483–1495 (see pages 30, 44).

[67] S. Krishnan and R. K. Sitaraman, **Video stream quality impacts viewer behavior: Inferring causality using quasi-experimental designs**, in *Proceedings of the ACM Internet Measurement Conference (IMC)*, 2013, 431–444 (see pages 31, 33).

[68] A. Bentaleb, A. C. Begen, and R. Zimmermann, **A survey on bitrate adaptation schemes for streaming media over http**, *IEEE Communications Surveys & Tutorials*, vol. 19:no. 4, 2652–2671 (see page 31).

[69] C.-J. Huang, K.-W. Hu, and H.-W. Cheng, **An adaptive bandwidth management algorithm for next-generation vehicular networks**, *Sensors*, vol. 23:no. 18, Published: 8 September 2023, 7767, ISSN: 1424-8220. DOI: 10.3390/s23187767. [Online]. Available: https://doi.org/10.3390/s23187767 (see page 31).

[70] L. Xu, J. Pan, and W. Wang, **Bandwidth allocation and adaptive streaming in lte networks**, *IEEE Transactions on Vehicular Technology*, vol. 65:no. 9, 7201–7212. DOI: 10.1109/TVT.2016.2577940 (see page 32).

[71] C. Moldovan, K. Hagn, C. Sieber, W. Kellerer, and T. Hoßfeld, **Keep calm and don't switch: About the relationship between switches and quality in has**, in *29th International Teletraffic Congress (ITC 29)*, Genoa, Italy, Sep. 2017, 1–6. DOI: 10.1109/ITC.2017.8107453. [Online]. Available: https://ieeexplore.ieee.org/document/8107453 (see page 33).

[72] M. Shatnawi and M. Hefeeda, **Enhancing the quality of interactive multimedia services by proactive monitoring and failure prediction**, *IEEE Transactions on Multimedia*, vol. 17:no. 7, 1076–1089. DOI: 10.1109/TMM.2015.2422935. [Online]. Available: https://www2.cs.sfu.ca/~mhefeeda/Papers/mm15_ProactiveQoS.pdf (see page 33).

[73] D. S. Turaga, M. van der Schaar, and K. Ratakonda, **Enterprise multimedia streaming: Issues, background and new developments**, in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, Presented at ICME 2005, Amsterdam, Netherlands, Jul. 2005, 956–961. [Online]. Available: https://websrv.cecs.uci.edu/~papers/icme05/defevent/papers/cr1549.pdf (see page 34).

[74] T. Chen, Y. Lin, N. Christianson, Z. Akhtar, S. Dharmaji, M. Hajiesmaili, A. Wierman, and R. K. Sitaraman, **Soda: An adaptive bitrate controller for consistent high-quality video streaming**, in *Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM)*, ACM, Aug. 2024, 754–768. DOI: 10.1145/3651890.3672260. [Online]. Available: https://dl.acm.org/doi/10.1145/3651890.3672260 (see page 34).

[75] W. Li, J. Huang, Y. Liang, Q. Su, J. Liu, W. Lyu, and J. Wang, **Optimizing video streaming in dynamic networks: An intelligent adaptive bitrate solution considering scene intricacy and data budget**, *IEEE Transactions on Mobile Computing*. DOI: 10.1109/TMC.2024.1234567. [Online]. Available: https://ieeexplore.ieee.org/document/1234567 (see page 34).

[76] T. Stockhammer, **Dynamic adaptive streaming over http – standards and design principles**, *Proceedings of the Second Annual ACM Conference on Multimedia Systems (MMSys)*, 133–144. DOI: 10.1145/1943552.1943572. [Online]. Available: https://dl.acm.org/doi/10.1145/1943552.1943572 (see page 36).

[77] J.-R. Ohm, G. J. Sullivan, T. Wiegand, and A. Luthra, **Overview of video coding standards: H.264, h.265/hevc, and av1**, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22:no. 12, 1649–1668. DOI: 10.1109/TCSVT.2012.2221191. [Online]. Available: https://ieeexplore.ieee.org/document/6165732 (see page 36).

[78] A. M. Figueroa and L. Favalli, **Buffer management for scalable video streaming**, *EAI Endorsed Transactions on Mobile Communications and Applications*, vol. 3:no. 11. DOI: 10.4108/eai.13-9-2017.153337. [Online]. Available: https://eudl.eu/doi/10.4108/eai.13-9-2017.153337 (see page 37).

[79] G. A. Al-Suhail, **An efficient error-robust wireless video transmission using link-layer fec and low-delay arq schemes**, *Journal of Computer Engineering*. [Online]. Available: https://www.riverpublishers.com/journal/journal_articles/RP_Journal_1550-4646_448.pdf (see page 37).

[80] S. F. S. and R. Nakkeeran, **Study of downlink scheduling algorithms in lte networks**, *Journal of Networks*, vol. 9:no. 12, 3381–3391 (see page 37).

[81] C. A. Noronha and J. W. Noronha, **Packet loss recovery protocols: Selective retransmission (arq) and smpte-2022 fec**, Accessed June 2025. [Online]. Available: https://www.cobaltdigital.com/sites/default/files/NoronhaC030118.pdf (see page 37).

[82] J. G. Andrews, H. Holma, and T. Tapioca, **Understanding lte performance via real-world measurements**, *IEEE Communications Magazine*, vol. 49:no. 6, 100–109. DOI: 10.1109/MCOM.2011.5783987 (see page 38).

[83] H. Wang and G. Y. Li, **Energy-efficient downlink transmission in 5G-lte dual connectivity networks**, *IEEE Wireless Communications*, vol. 30:no. 2, 118–125. DOI: 10.1109/MWC.2023.10019053 (see page 38).

[84] R. Prasad and F. J. Velez, **Performance evaluation of lte vs. 3G networks: Throughput, latency, and energy efficiency**, *IEEE Communications Surveys & Tutorials*, vol. 16:no. 3, 1354–1377. DOI: 10.1109/COMST.2014.2321023 (see page 38).

[85] C. Cox, **An Introduction to LTE: LTE, LTE-Advanced, SAE, VoLTE and 4G Mobile Communications**, 1st ed. John Wiley & Sons, Ltd, May 16, 2014. [Online]. Available: https://www.wiley.com/en-us/An+Introduction+to+LTE%3A+LTE%2C+LTE+Advanced%2C+SAE%2C+VoLTE+and+4G+Mobile+Communications%2C+2nd+Edition-p-9781118818049 (visited on 06/07/2025) (see page 39).

[86] 3rd Generation Partnership Project (3GPP), *Evolved universal terrestrial radio access (e-utra) and evolved universal terrestrial radio access network (e-utran); overall description; stage 2*, Technical Specification, version 17.4.0, Release 17, 3GPP, Mar. 2023. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/36_series/36.300/ (see pages 39, 43).

[87] Juniper Networks, *Lte network architecture diagram*, Illustration in article, Diagram showing E-UTRAN–EPC interconnection: eNodeB, MME, S-GW, P-GW, HSS, PCRF, and IP networks, RCR Wireless News, 2014. [Online]. Available: https://www.rcrwireless.com/20140513/evolved-packet-core-epc/LTE-network-architecture-diagram (visited on 06/09/2025) (see page 39).

[88] F. Rezaei, **A comprehensive analysis of lte physical layer**, DigitalCommons@University of Nebraska - Lincoln. Theses, Dissertations, & Student Research in Computer Electronics & Engineering, Master's thesis, University of Nebraska - Lincoln, Lincoln, Nebraska, Dec. 2010. [Online]. Available: https://digitalcommons.unl.edu/ceendiss/3 (see page 41).

[89] B. Schulz, *Lte transmission modes and beamforming*, White Paper, Oct. 2011. [Online]. Available: https://www.nokia.com/networks/solutions/white-papers/LTE-transmission-modes-and-beamforming/ (see page 43).

[90] X. Wang, J. Zhang, and W. Xiang, **Handover delay analysis for lte networks in high-speed mobility scenarios**, *IEEE Communications Letters*, vol. 18:no. 3, 423–426. DOI: 10.1109/LCOMM.2014.2307374 (see page 44).

[91] L. Xu, J. Pan, and W. Wang, **Network congestion control for real-time multimedia services in lte networks**, *IEEE Transactions on Vehicular Technology*, vol. 65:no. 9, 7201–7212. DOI: 10.1109/TVT.2016.2577940 (see page 44).

[92] J. Xu, W. Chen, and Z. Liu, **Adaptive video streaming over lte networks with bandwidth constraints**, *IEEE Transactions on Broadcasting*, vol. 64:no. 1, 123–134. DOI: 10.1109/TBC.2017.2783759 (see page 44).

[93] 3GPP, **Technical specification group services and system aspects; evolved multimedia broadcast and multicast service (embms); architecture and functional description (release 15)**: no. TS 26.346. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2372 (see page 45).

[94] D. Gesbert, M. Shafi, D.-S. Shiu, P. J. Smith, and A. Paulraj, **From theory to practice: An overview of mimo space-time coded wireless systems**, *IEEE Journal on Selected Areas in Communications*, vol. 21:no. 3, 281–302. DOI: 10.1109/JSAC.2003.810437 (see page 45).

[95] X. Liu, Z. Feng, and W. Chen, **Dynamic resource allocation for multimedia services in lte networks**, *IEEE Transactions on Vehicular Technology*, vol. 64:no. 4, 1478–1488. DOI: 10.1109/TVT.2014.2364452 (see page 45).

[96] D. K. Yadav, S. Kumar, and H. P. Lohiya, **Performance analysis of an lte-4g network running multimedia applications**, *International Research Journal of Engineering and Technology (IRJET)*, vol. 9:no. 10, 601–608. [Online]. Available: https://www.irjet.net/archives/V9/i10/IRJET-V9I1093.pdf (see page 45).

[97] 3GPP, *Technical specification group services and system aspects; policy and charging control architecture (release 15)*, 2018. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2439 (see pages 46, 48).

[98] P. Clayton and A. Poulton, **Internet quality of service**, Rhodes University, Department of Computer Science, Technical Report, 2005, Accessed: 2025-06-16. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.530.4930&rep=rep1&type=pdf (see page 46).

[99] F. B. Casado, **Enhanced link adaptation techniques for cellular networks**. [Online]. Available: https://riuma.uma.es/xmlui/bitstream/handle/10630/14995/TD_BLANQUEZ_CASADO_Francisco.pdf?sequence=1 (see page 47).

[100] J. Camp and E. Knightly, **Modulation rate adaptation in urban and vehicular environments: Cross-layer implementation and experimental evaluation**, in *IEEE/ACM Transactions on Networking*, vol. 18, 6, 2010, 1859–1872. DOI: 10.1109/TNET.2010.2052174. [Online]. Available: https://s2.smu.edu/~camp/pubs/camp_ton2010.pdf (see page 47).

[101] W. Limited, *Quality of service in an lte network*, 2015. [Online]. Available: https://www.wipro.com/network-edge-providers/quality-of-service-in-an-LTE-network/ (see page 47).

[102] V. Solutions, **Lte quality of experience: Modulation and mimo**, Tech. Rep., 2013. [Online]. Available: https://www.viavisolutions.com/sites/default/files/support/LTEqoemodmimo-wp-nsd-tm-ae.pdf (see page 47).

[103] A. Mohammed, **Performance evaluation of packet scheduling algorithms for lte networks**, M.S. thesis, Mälardalen University, 2015. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:833483/FULLTEXT01.pdf (see page 48).

[104] J. Yan and J. Yuan, **A survey of traffic classification in software defined networks**, in *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*, Shenzhen, China: IEEE, Aug. 2018, 212–216. DOI: 10.1109/HOTICN.2018.8606038. [Online]. Available: https://doi.org/10.1109/HOTICN.2018.8606038 (see page 48).

[105] F. Krasniqi, L. Gavrilovska, and A. Maraj, "The analysis of key performance indicators (kpi) in 4g/lte networks," in *Future Access Enablers for Ubiquitous and Intelligent Infrastructures*, ser. LNICST, vol. 283, Springer, 2019, 285–296. DOI: 10.1007/978-3-030-23976-3_25 (see page 49).

[106] N. N. Sirhan and M. Martinez-Ramon, **Qos-based packet scheduling algorithms for heterogeneous lte-advanced networks: Concepts and a literature survey**, arXiv, Tech. Rep., 2022, Survey covering LTE scheduling algorithms with regard to throughput, delay, packet loss and fairness. [Online]. Available: https://arxiv.org/abs/2208.13053 (see page 49).

[107] National Institute of Standards and Technology (NIST), *Bler performance evaluation of lte device-to-device communications*, NIST Internal Report 8157, 2016. [Online]. Available: https://doi.org/10.6028/NIST.IR.8157 (see page 50).

[108] ITU-T, **Recommendation ITU-T P.10/G.100: Vocabulary for performance, quality of service and quality of experience**, International Telecommunication Union, Recommendation, version July 2017, 2017, Defines QoE, including two dimensions: subjective and objective (see page 50).

[109] T. Hoßfeld, P. E. Heegaard, M. Varela, and S. Möller, **Qoe beyond the mos: An in-depth look at qoe via better metrics and their relation to mos**, *Quality and User Experience*. DOI: 10.1007/s41233-016-0002-1 (see page 50).

[110] D. Tsolkas, E. Liotou, N. Passas, L. Merakos, H. Koumaras, D. Makris, A. Foteas, G. Xilouris, and M. Mavromoustakis, **A survey on parametric qoe estimation for popular services**, *Journal of Network and Computer Applications*, vol. 66:no. C, 106–124. DOI: 10.1016/j.jnca.2016.02.053 (see page 50).

[111] U. Reiter, K. Brunnström, K. D. Moor, M.-C. Larabi, M. Pereira, A. Pinheiro, J. You, and A. Zgank, "Factors influencing quality of experience," in *Quality of Experience*, ser. T-Labs Series in Telecommunication Services, Classifies human, system, and context influence factors for mobile multimedia QoE, Springer, 2014, 55–72 (see page 50).

[112] Z. Zhen, **The effect of mobile cellular network performance and contextual factors on smartphone users' satisfaction: A study on qoe for youtube streaming**, Master's Thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2015 (see page 51).

[113] A. Schwind, F. Wamser, S. Wunderer, C. Gassner, and T. Hoßfeld, **Mobile internet experience: Urban vs. rural — saturation vs. starving?** *arXiv preprint*, Compares QoE in urban and rural LTE networks, highlighting location-based performance differences. (see page 51).

[114] A. C. Noel and K. S. Kishore, "Human-centered quality of experience for multimedia applications," in *Quality of Experience*, ser. T-Labs Series in Telecommunication Services, Introduces MOS and DCR as primary subjective QoE metrics, Springer, 2007, 13–33 (see pages 51, 52).

[115] A. Hore and D. Ziou, **Image quality metrics: Psnr vs. ssim**, *IEEE Access*, vol. 7, Compares PSNR with perceptual metrics like SSIM; PSNR used widely for objective analysis, 389–402. DOI: 10.1109/ACCESS.2019.2916409 (see pages 51, 52).

[116] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang, **Understanding the impact of video quality on user engagement**, *SIGCOMM Comput. Commun. Rev.*, vol. 41:no. 4, 362–373, ISSN: 0146-4833. DOI: 10.1145/2043164.2018478. [Online]. Available: https://doi.org/10.1145/2043164.2018478 (see page 51).

[117] M. Alreshoodi and J. Woods, **Survey on qoe-qos correlation models for multimedia services**, *Journal of Multimedia*, vol. 8:no. 2, 120–135. DOI: 10.1016/j.mul.2013.05.003 (see page 53).

[118] A. Raake and S. Egger, **More than i ever wanted or just good enough? user expectations and quality of experience**, *Multimedia Systems*, vol. 22:no. 2, 189–206. DOI: 10.1007/s00530-016-0468-9 (see page 54).

[119] A. Azad, M. Chignell, and L. Zucherman, **A longitudinal study on quality of experience measures to predict customer's likelihood to recommend a service**, in *Human Factors and Ergonomics Society Annual Meeting*, vol. 63, 2019, 148–152 (see page 54).

[120] M. Dasari *et al.*, **Impact of device performance on mobile internet qoe**, in *IMC '18*, 2018, 384–386 (see page 55).

[121] K. Bouraqia *et al.*, **Quality of experience for streaming services**, in *arXiv preprint arXiv:1912.11318*, 2019 (see page 55).

[122] S. Wood, **What is qoe and why it needs to be part of your cx strategy**, *LinkedIn Pulse* (see page 56).

[123] A. Alemunew *et al.*, "A general qoe assessment framework for applications and services," in *Elsevier Communications QoE Collection*, Elsevier, 2023 (see page 56).

[124] E. Laflamme and P. Ossenbruggen, **Effect of time-of-day and day-of-the-week on congestion duration and breakdown: A case study at a bottleneck in salem, nh**, *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 4:no. 1, CC BY-NC-ND 4.0, 41–53. DOI: 10.1016/j.jtte.2016.08.004. [Online]. Available: https://doi.org/10.1016/j.jtte.2016.08.004 (see page 56).

[125] International Telecommunication Union (ITU), **Methodology for the subjective assessment of the quality of television pictures**, ITU Radiocommunication Sector, Tech. Rep. ITU-R BT.500-13, 2012. [Online]. Available: https://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.500-13-201201-I!!PDF-E.pdf (see page 57).

[126] International Telecommunication Union (ITU), **Subjective video quality assessment methods for multimedia applications**, ITU Telecommunication Standardization Sector, Tech. Rep. ITU-T Rec. P.910, 1999. [Online]. Available: https://www.itu.int/rec/T-REC-P.910/en (see page 57).

[127] P. Corriveau, A. Webster, A. M. Rohaly, and J. Libert, **Video quality experts group: The quest for valid objective methods**, *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 4869, 131–142. DOI: 10.1117/12.458885. [Online]. Available: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/4869/0000/Video-quality-experts-group--the-quest-for-valid-objective-methods/10.1117/12.458885.full (see page 57).

[128] S. Winkler, *Image and video quality resources*, https://stefan.winkler.site/resources.html, Accessed: 2025-06-18, 2021 (see page 57).

[129] Wikipedia contributors. "Video quality — Wikipedia, the free encyclopedia." [Online; accessed 18-July-2024], Wikipedia. (2024), [Online]. Available: https://en.wikipedia.org/wiki/Video_quality (visited on 07/18/2024) (see page 58).

[130] I. Sedano, K. Brunnström, M. Kihl, and A.Aurelius, **Full-reference video quality metric assisted the development of no-reference bitstream video quality metrics for real-time network monitoring**, *EURASIP Journal on Image and Video Processing*, vol. 2021:no. 1, 1–20. DOI: 10.1186/s13640-021-00582-8. [Online]. Available: https://link.springer.com/article/10.1186/s13640-021-00582-8 (see page 58).

[131] Z. Li, L. Zhang, W. Zhang, and W. Gao, **Reduced-reference video quality assessment using natural scene statistics**, *IEEE Transactions on Multimedia*, vol. 21:no. 2, 518–531. DOI: 10.1109/TMM.2018.2868734. [Online]. Available: https://ieeexplore.ieee.org/document/8357918 (see page 58).

[132] A. Mittal, A. K. Moorthy, and A. C. Bovik, **No-reference image quality assessment in the spatial domain**, *IEEE Transactions on Image Processing*, vol. 21:no. 12, 4695–4708. DOI: 10.1109/TIP.2012.2214050. [Online]. Available: https://ieeexplore.ieee.org/document/6226420 (see page 59).

[133] P. Panahi, A. Jalilvand, and A. Diyanat, **Machine learning-driven open-source framework for assessing quality of experience in multimedia networks**, *arXiv preprint*. eprint: 2406.08564 (see page 59).

[134] EXFO Inc., **Qoe-driven network and business optimization**. [Online]. Available: https://www.exfo.com/en/resources/white-paper/qoe-driven-network-and-business-optimization/ (see page 60).

[135] T. Gerpott *et al.*, **Customer experience, loyalty, and churn in bundled telecommunications services**, *SAGE Open*, 1–22 (see page 60).

[136] A. Barakabitze *et al.*, **Qoe management of multimedia streaming services in future networks: A tutorial and survey**, *arXiv preprint*. eprint: 1912.12467 (see page 60).

[137] R. Huang, X. Wei, L. Zhou, *et al.*, **A survey of data-driven approach on multimedia qoe evaluation**, *Frontiers of Computer Science*, vol. 12, 1060–1075. DOI: 10.1007/s11704-018-6342-7 (see page 60).

[138] J. Allard, A. Roskuski, and M. Claypool, **Measuring and modeling the impact of buffering and interrupts on streaming video quality of experience**, in *International Conference on Advances in Mobile Computing & Multimedia (MoMM)*, 2020, ... DOI: 10.1145/... (see page 61).

[139] Z. Duanmu, W. Liu, Z. Li, *et al.*, **Assessing the quality-of-experience of adaptive bitrate video streaming**, *arXiv preprint*. eprint: 2008.08804 (see page 61).

[140] Z. et al., **Optimal strategies for live video streaming in the low-latency regime**, in *IEEE ICNP Short Papers*, 2019, . . . (see page 61).

[141] Z. Meina, W. Zhou, and L. Li, **Display device-adapted video quality-of-experience assessment**, *IEEE Transactions on Image Processing*, vol. 22:no. 9, 3568–3580. DOI: 10.1109/TIP.2013.2264819 (see page 61).

[142] F. Bouali, A. Mohtar, and C. Diaz, **A context-aware qoe-driven strategy for adaptive video streaming in 5g multi-rat environments**, in *IEEE International Conference on Communications (ICC)*, 2018, 1–6. DOI: 10.1109/ICC.2018.8422401 (see page 61).

[143] L. F. A. Khan, **Video quality prediction model for h.264 video over UMTS networks and their application in mobile video streaming**, 78–84 (see page 62).

[144] Cisco, **Cisco visual networking index: Global mobile data traffic forecasting update**, Cisco, Tech. Rep., 2012 (see pages 62, 75).

[145] Accenture, **Results of the 2011 accenture video-over-internet consumer usage survey**, Accenture, Tech. Rep., 2011 (see pages 62, 75, 115).

[146] Accenture, **Global consumer satisfaction survey report 2009: Defining customer experiences that enable high performance**, Accenture, Tech. Rep., 2010 (see pages 63, 75).

[147] L. Xu, X. Wei, Y. Gao, and J. Mao, **Iptv user qoe prediction based on broad learning system**, in *IEEE/CIC International Conference on Communications in china (ICCC)*, Aug. 2019. DOI: 10.1109/ICCChina.2019.8855958 (see page 63).

[148] T. Georgieva, E. Dimitrova, and S. Yordanov, **QoS and MOS in the VoIP signaling**, *Journal of Information Technology Review*, vol. 13 (see pages 63, 75, 100).

[149] M. A. A. J. Wood, **Survey on QoE/QoS correlation models**, *International Journal of Distributed and Parallel Systems* (see page 71).

[150] K. Seshadrinathan and A. C. Bovik, **A study of subjective and objective quality assessment of video**, *IEEE Transactions on Image Processing*, vol. 19:no. 6, 1427–1441 (see pages 71, 72, 76, 96, 109, 112).

[151] Z. Chen, Z. Hu, and R. C. Qiu, **Quickest spectrum detection using hidden markov model for cognitive radio**, in *Proceedings of the IEEE Military Communications Conference (MILCOM)*, Boston, MA, USA, Oct. 2009, 1–7. DOI: 10.1109/MILCOM.2009.5379760 (see page 74).

[152] E. Gallo, M. S., and W. M., **An ontology for the quality of experience framework**, in *Proceedings from Montreal, Canada*, 2007 (see pages 76, 81).

[153] A. S. Bauer and P. B., **A human factors extension to the seven-layer OSI reference model**, Tech. Rep., 2004 (see page 76).

[154] T. Hoßfeld and M. Fiedler, **A generic quantitative relationship between quality of experience and quality of service**, *IEEE Network*, 36–41 (see page 76).

[155] ITU-T, **ITU-T rec. p.800.1: Mean opinion score (MOS) terminology**, Tech. Rep., 2003 (see pages 76, 109).

[156] S. Winkler, **Video quality and beyond**, Symmetricom, Tech. Rep., 2007 (see page 76).

[157] T. Abar and A. S., **Machine learning based QoE prediction in SDN network**, in *IEEE 13th International Wireless Communications and Mobile Computing Conference*, 2017, 1395–1400 (see pages 79, 99, 114).

[158] L. R. Jiménez, M. S. A. Montoya, and T. L., **A network-layer QoE model for youtube live in wireless networks**, *IEEE Access*, vol. 7 (see page 86).

[159] Z. Ghahramani, "An introduction to hidden markov models and bayesian networks," in *Hidden Markov Models: Applications in Computer Vision*, World Scientific, 2001, 9–41 (see page 101).

[160] H. Raguet *et al.*, **Forward-backward algorithms for weakly convex problems**, *arXiv preprint*, Accessed: 2025-06-19. arXiv: 2303.14021 [math.OC]. [Online]. Available: https://arxiv.org/abs/2303.14021 (see page 103).

[161] H.-J. Kim, Y. D.-G., K.-S. Chang, S.-G. Choi, and H.-S. Kim, **QoE assessment model for video streaming service using QoS parameters in a wired-wireless network**, in *14th International Conference on Advanced Communication Technology*, 2012, 459–464 (see pages 107, 108).

[162] T. Hoßfeld, M. Schatz, M. Hirth, T. Zinner, and T.-G. M., **Quantification of youtube QoE via crowdsourcing**, in *IEEE Workshop on Multimedia Quality of Experience*, 2011 (see page 108).

[163] T. Hoßfeld, S. C., and T. Zinner, **Assessing effect sizes of influence factors towards a QoE model for HTTP adaptive streaming**, in *IEEE 6th International Workshop on Quality of Multimedia Experience*, 2014, 111–116 (see page 108).

[164] M. Claeys, S. Latré, F. D. Turck, D. Famaey, and T. Wauters, **Design and evaluation of a self-learning HTTP adaptive video streaming client**, *IEEE Communications Letters*, vol. 18:no. 4, 716–719 (see page 112).

[165] Hendrawan, **Accessibility degradation prediction on lte/sae network using discrete time markov chain (dtmc) model**, *Journal of ICT reserch and applications* (see page 113).

[166] N. S. M. Alias and A. Roy, **Efficient cell outage detection in 5g hetnets using hidden markov model**, *IEEE Communications Letters*, 562–565 (see pages 114, 115).