Data-Driven Quality of Experience Prediction Model and Measurement of Multimedia Streaming Services using Machine Learning

By

Farhan Ahmed

2019-NUST-MS-CSE-317765

Supervisor

Dr. Muhammad Tariq Saeed

School of Interdisciplinary Engineering and Science (SINES)

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

April 2, 2023

Data-Driven Quality of Experience Prediction Model and Measurement of Multimedia Streaming Services using Machine Learning

By

Farhan Ahmed

2019-NUST-MS-CSE-317765

Supervisor

Dr. Muhammad Tariq Saeed

A thesis submitted in conformity with the requirements for the degree of *Master of Science* in Computational Sciences and Engineering

School of Interdisciplinary Engineering and Science (SINES)

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

April 2, 2023

Declaration

I certify that this research work titled *"Data-Driven Quality of Experience Prediction Model and Measurement of Multimedia Streaming Services using Machine Learning"* and the work presented in it are my own and has been generated by me as a result of my own original research. This work has not been presented elsewhere for assessment.

I confirm that:

1. This work was done wholly or mainly while in candidature for a Master of Science degree at NUST
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at NUST or any other institution, this has been clearly stated
3. Where I have consulted the published work of others, this is always clearly attributed
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work
5. I have acknowledged all main sources of help
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself

---

Farhan Ahmed,

2019-NUST-MS-CSE-317765

Copyright Notice

This thesis is dedicated to *my beloved parents*

Abstract

The Internet traffic today is mostly multi-media traffic due to the exceptionally expanding interest in Over The Top (OTT) services like Facebook, YouTube, Netflix etc. It has proven to be extremely difficult for Internet service providers (ISPs) to meet their customers' needs in terms of Quality of Experience (QoE) because of the proliferation of networking data, especially video streaming. Therefore, QoE modelling and measurement of multimedia services is an open research area for the research community. Due to ever-increasing user demand for multimedia services, ISPs and OTT providers require innovative solutions for QoE prediction of HTTP Adaptive Video-streaming (HAS) applications as most of the video services over the internet are HAS-based. Therefore, the QoE prediction model will lead towards identifying the root causes for QoE impairments and understanding the impact of different Key Quality Indicators (KQIs). The primary objective of this study is to propose supervised-learning-based QoE prediction ensemble Voting Regression (VR) and Stacking Regression (SR) models based on machine-learning algorithms such as Random Forests (RFs), Support Vector Regression (SVRs), Stochastic Gradient Descent (SGD) and Multilayer Perceptron (NN) models considering appropriate QoE influencing factors. We utilize Waterloo Streaming Quality-of-Experience Database for more accurate prediction of QoE over the multimedia video streaming services in this study. This work has a multi-fold contribution: First, the data set was optimized using four feature selection techniques based on machine learning also including Principal Component Analysis (PCA) to investigate the impact of different KQIs and retain the most appropriate ones in the feature-engineering stage. Secondly, making k-fold validations and hyper-parameter tuning of standalone ML models was adopted to check the accuracy of each model over the given data set in the model optimization and training stage. Thirdly, upon these hyper-parameter-tuned base ML models, ensemble VR and SR models were created. In the final stage,

different ML models were evaluated based on learning curves, execution times, training times and performance metrics for comparative analysis among various features obtained from different feature selection techniques and then analyzed the algorithm which suits best for estimated QoE prediction. The results show significantly higher scores of R^2 i.e 0.852367 and a significant improvement of 4.64% in terms of R^2 was observed as compared to previous studies. Finally, the lower values of MAE, MSE and RMSE i.e 0.085513, 0.220756 and 0.469846 were obtained respectively, while showing the highest PLCC value of 0.92539 and SRCC value of 0.875782 while predicting QoE.

Acknowledgments

I would like to thank my supervisor, Dr Muhammad Tariq Saeed, for his continuous guidance and efforts towards teaching me how to conduct my scientific research independently. From the beginning of the development of the study project to the completion of my thesis, I have gained a lot of knowledge and been inspired by every conversation we have had. His support to develop my data analysis, machine learning and analytical skills with current trends is highly appreciated. I had a great time working with him on this voyage. I am grateful for his encouragement, constructive ideas, and ability to expand my thinking by asking critical questions.

I would also like to thank Dr Zamir Hussain and Dr Absaar Ul Jabbar for their guidance towards this journey.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

With the massive growth of networking data especially in video streaming over the past few years, it has posed great challenges for Internet Service Providers (ISPs) to meet the users' demands, especially in terms of Quality of Experience (QoE) [2]. As per the most recent Cisco Visual Networking Index (VNI) Forecast, 82% of all IP traffic will be video by 2023 [3]. Delivering high video quality to the users is a major concern of the ISPs to fulfil their customer's needs and also to increase their revenue potential nowadays [4]. For this purpose, ISPs need an efficient approach for accurate prediction and measurement of QoE for video-streaming services (e.g., YouTube, Facebook and Netflix), over the internet.

1.2 QoE in HTTP Adaptive Video-Streaming (HAS)

The International Telecommunication Union (ITU-T) has established a standard definition, according to which QoE is defined as, "The degree of delight or annoyance of the user of an application or service"[5]. The QoE is a multidisciplinary concept that depends on multiple influencing factors such as application, business, context, network, system and the users [6].

To reduce video playing interruptions and increase bandwidth consumption, most of the video streaming services use **H**TTP **A**daptive **V**ideo-**S**treaming (**HAS**), which is a video-streaming technique that adjusts the video to the current network conditions. It

allows service providers to optimise resource usage and Quality of Experience (QoE) by including information from various network layers to provide and adjust video in the highest possible quality based on client adaptation algorithm [7].

1.3 Issues in HTTP Adaptive Video-Streaming (HAS)

Multiple applications and network Key Quality Indicators (KQIs) e.g, video content, stalling ratio, stalling duration, bit rate etc. affect the QoE for streaming videos using HTTP Adaptive Streaming (HAS) [8]. Moreover, end-to-end encryption is used by Over The Top (OTT) providers which means the ISPs do not have direct access to the end-users devices, making it more difficult and challenging for ISPs to accurately predict QoE and meet the QoE requirements of video streaming applications for end-users[9].

The QoE prediction of HAS is a complex, multivariate non-linear problem involving multiple factors such as end-users devices, network Key Performance Indicators (KPIs) such as bandwidth, throughput, end-to-end delay, etc., application-related KQIs such as stalling events, video resolutions, frame rates, video quality layer switch, etc. Therefore, both ISPs and OTTs need a novel solution to meet user-perceived quality which can be achieved by accurate prediction of QoE using Machine-Learning (ML) approaches[10]. Although the studies in [10–12] offer QoE prediction for encrypted video streaming services, it is still challenging to accurately forecast QoE from encrypted video streaming data. The ML-based QoE prediction models for HAS found in the literature are limited due to: the accuracy of predictive models, limited availability of data set with ground truth, near-real-time requirement to be deployed in real networks/service management and monitoring, complex multi-variate non-linear nature of quality perceived by the users.

1.4 Research Objectives

The aim of this research is to propose an optimized ensemble ML-based QoE prediction model for HAS video streaming. The objectives of this work are as follows:

- Understanding the impact of different KQIs on users' QoE.
- Investigation and comparative analysis of QoE prediction of different ML models.

- Proposing an optimized ensemble ML-based QoE prediction model for HAS video streaming services.

1.5 Research Contribution

This work has the following contributions:

1. We performed Principal Component Analysis (PCA) and feature selection techniques to study the impact of dimensionality reduction and select important QoE features of HAS.
2. We provide the learning curves, execution and training times of ML models over the techniques applied in this literature and the comparative analysis is done on the basis of performance of different standalone ML models and ensemble models based on performance metrics.
3. We propose an optimal ensemble-based ML model for QoE prediction of HAS video streaming.

1.6 Structure of Thesis

This thesis is organised as follows: Chapter 2 investigates state-of-the-art works proposed in the literature. Chapter 3 explains the proposed methodology in which we have provided the details on creating our ensemble ML models and discussed novel approaches for QoE prediction of HAS. Chapter 4 covers details of the experimental results for this study and provides a detailed comparative analysis of the techniques applied in this study. While Chapter 5 provides a detailed discussion on comparative analysis done for this study. Finally, Chapter 6 concludes and provides future work for this study.

1.7 Summary

In this chapter, we have given a brief overview of QoE, its definition in International Telecommunication Union (ITU-T) [5] and its importance in video-streaming services mentioned in Cisco Visual Networking Index (VNI) Forecast [3]. Further, we have

described various issues in arising for QoE prediction of HAS depending upon various factors. Based on these issues we have further discussed our strategy for a more accurate QoE prediction of HAS.

Chapter 2

Literature Review

Full Reference (FR), Reduced Reference (RR), and No Reference (NR) are the three categories in which objective video quality metrics are categorised based on the material they use to accomplish the assessment [13]. The best performance in terms of accuracy to human perception has been found with FR metrics, which do a frame-to-frame comparison between the original and received (impaired) data. However, these metrics demand access to the source data and are computationally intensive. As a result, they are unsuitable for real-time evaluation and are better suited to benchmarking. RR and NR metrics, on the other hand, evaluate simply the material that has been received and the network conditions. As a result, in terms of timeliness and computing efficiency, they are the best option [14]. NR video quality metrics have been classified into encoding parameters, playstats, network-related bitrate parameters, users' devices and content type. Our work is also based upon NR metrics considering supervised learning models. Table 2.1 shows parameters related to these categories

This chapter provides details of works related to QoE prediction of video streaming. Using decision trees, Staehlens *et al.* [15] examined and modelled impaired visibility in HD H.264/AVC encoded video sequences. They claim in their work that it is possible to forecast the visibility of various impairments using only a few parameters (a total of 39 parameters were retrieved from the bitstream total parameters). They generated a Mean Opinion Score (MOS) based on solely Decision Trees (DT) for binary classification on their data set. Because of the increased complexity and time required for decision tree training, it is inadequate for applying regression and predicting the exact score of QoE.

Table 2.1: Parametric Features for HAS Video Streaming

Encoding	**Playstat**	**Bit Stream**	**Network KPIs**	**Users device**	**Content type**
Encoding Codec e.g. H.264 H.265 VP9	Initial loading time	Bitrates	Bandwidth	Resolution e.g. HDTV, smartphone	Animation
Frame Rate	Number of Stalling events	Average time on highest video quality layer	End-to-End Latency	Operating system (Window, Linux, Android, iPhone)	Sport
Video Resolution	Average stalling duration	Switching between video quality layers	Throughput	Hardware capabilities (CPU, GPU, RAM, Battery, Hardisk)	News
Bitrates	Stalling ratio	Client adaptation Algorithm			

Narwaria *et al.* [16] compared the performance of Support Vector Regression against eight different visual quality predictors on two video databases. They also calculated execution time for different metrics. The results reveal that prediction accuracy has improved significantly. However, due to higher computational complexity and overfitting, both of which affect the evaluation performance of SVR, their study is constrained in terms of the model employed.

A linear regression model based on bitstream and network characteristics was proposed by Khan *et al.* [17]. They showed that movies subjected to a simulated (NS2) UMTS situation had very good correlation values. This is the first time we've seen a method that uses simulated impaired movies rather than synthetic solutions. However, they only consider video spatial quality (the visual quality of a video frame) and disregard temporal artifacts like stalls. Konuk *et al.* [18] used linear regression methods to extract independent characteristics from spatial and temporal parameters derived from video packet losses, bit rate, and spatio-temporal complexity. On the LIVE Video quality database [40], they report correlations greater than 0.8 for spatio-temporal complexity. However, their work does not take into account other encoding formats. Moreover, it does not offer a comprehensive general-purpose solution: for different sorts of distortions,

Table 2.2: Comparative Analysis of state-of-the-art w.r.t consider parametric features

Parameteric no-reference Regressive Model	**Considered Parametric Features**					
	Encoding	**Playstat**	**Bit Stream**	**Network KPIs**	**Users device**	**Content type**
Staehlens *et al.* [15]	X		X	X	X	X
Narwaria *et al.* [16]	X				X	X
Khan *et al.* [17]	X		X	X	X	X
Konuk *et al.* [18]	X		X	X		X
Staelens *et al.* [19]	X		X	X	X	X
Zhu *et al.* [20]	X		X	X	X	X
Sogaard *et al.* [21]	X		X		X	X
Shahid *et al.* [22]	X		X			X
Pandremmenou *et al.* [23]	X		X	X		X
Huang *et al.* [24]	X		X	X	X	X
Torres Vega *et al.* [25]	X		X	X	X	X
Shalala *et al.* [26]	X		X	X	X	X
Duc *et al.* [27]	X	X	X	X	X	X
Liu *et al.* [28]	X		X	X	X	X
Qian *et al.* [29]	X	X		X		X
Ahmad *et al.* [10]	X	X	X	X	X	X
Zhou *et al.* [30]	X		X	X	X	X
Taha *et al.* [31]	X		X	X	X	X
Kang *et al.* [32]	X		X	X	X	X
Danish *et al.* [33]	X		X		X	X
Youssef *et al.* [34]	X		X	X	X	X
Minovski *et al.* [35]	X		X	X	X	X
Tao *et al.* [36]	X	X	X		X	X
Zhang *et al.* [37]	X	X	X	X	X	X
Laiche *et al.* [38]	X	X	X	X	X	X
Youssef *et al.* [39]	X		X		X	X
Our Work	X	X	X	X	X	X

four alternative evaluation functions are utilised.

Staelens *et al.* [19] described a method for estimating NR video quality that leverages a symbolic regression framework trained on a wide collection of codec parameters. While their approach has a strong correlation with subjective assessments, it is only suitable for H.264 compressed streams, limiting its generality.

To forecast the quality of a video sequence, Zhu *et al.* [20] proposed using neural networks and features extracted from the study of Discrete Cosine Transform (DCT) coefficients of each decoded frame. In compressed movies from four separate well-known datasets, their method produced good correlation findings. However, due to its complexity, the method is unsuited for real-time deployments. Using features collected from specific codecs (MPEG or H.264/AVC), the analysis of DCT coefficients, and the calculation of the quantization level employed in the I-frames to gauge the quality of videos warped by the compression process, similar methods were given in [21]. They correlate better with subjective studies than several state-of-the-art metrics (FR, RR, and NR), making this a very promising solution for H.264/AVC compressed streams. However, the approach is not suitable for modelling real video artifacts because it was created to assess compressed video sequences. However, this conclusion is based on only one video content being tested in each fold and without predicting subjective scores.

Shahid *et al.* [22] suggested a model for estimating quality that included several bitstream-layer features with an Artificial Neural Network. Moreover, they put their method to the test on compressed videos and compared it to PSNR. For measuring the correctness of bitstream parameters to full reference metrics and subjective assessments in videos affected by compression and synthetic impairments, Pandremmenou *et al.* [23] used the Least Absolute Shrinkage and Selection Operator (LASSO) regression technique. NR), making it a suitable alternative for only compressed H.264/AVC streams.

The NR metric proposed by Huang *et al.* [24] focuses on the impairments resulting from compressing videos into HEVC. They proposed using pixel-level characteristics to build Elastic Nets [41] to measure the perceived degradation when using HEVC compression. They tested their network on the SJTU videoset after training it on the LIVE [40]. To subjective research, they got a 90 percent spearman correlation. They demonstrate a viable method for detecting video degradation caused by HEVC videos. Though learning a specific codec can help such algorithms work well, combining them into a

single general-purpose model or extending them to new codecs is difficult.

In order to construct a representative feature set, the authors in Torres Vega *et al.* [25] extracted eight NR video features (which occur at the bit-stream and pixel level) and combine them with the nominal bit-rate and anticipated level of packet loss. This feature set is then fed into regression-based predictive algorithms, which determine the quality of experience. They tested 9 models (ranging from linear regression to support vector machines) in a network-impaired video set and compared the performance of their NR predictive technique to VQM [42]. With ensemble regression trees, they were able to achieve near-perfect accuracy. However, they focused mostly on transmission and compression video artifacts.

Similarly, Shalala *et al.* [26] worked on QoE prediction of HAS video streaming considering user profile bitrate, FPS, resolution and device application. They considered six different models i.e LR, linear discriminant analysis, KNN, DT, Gaussian naive Bayes and SVM. They also considered three different feature selection techniques i.e Recursive Feature Elimination (RFE), Univariate Feature selection and Model-based Feature selection using Decision Trees (DT) with a prediction accuracy of about 73.37 to 87.63% based on SL techniques. Their work is limited, however, because it is based on a feature selection technique that ignored the temporal artifacts like stalls etc. which are important parameters considered in ITU-T P.1203 [1].

Duc *et al.* [27] considered Bidirectional LSTM: is a unified end-to-end prediction approach that uses the MOS measure to assess QoE and is built on deep learning (DL) as a combination of CNN and LSTM, which focuses on both forward and backward dependencies in accordance with bitrate changes and rebuffering information, can capture the memory-related temporal impacts of QoE for continuously predicting QoE in HTTP adaptive streaming. Similarly, in the extension of their work Liu *et al.* [28] proposed a deep learning approach by combining CNN and LSTM for QoE prediction of HAS. The assessment metric considered was the Mean Opinion Score (MOS) with a prediction accuracy of 88.74%. Because the CNN architecture was created to combine various sorts of multimedia input and forecast QoS/QoE values. As a result, their next work focuses on a deep reinforcement learning-based framework for bitrate change based on viewer interest. They evaluated the perceived video quality at a specific time; it simply reflects the quality assessment locally within a specific time period, without taking into account

the cumulative effects of previous occurrences. As a result, their work is highly prone to spatio-temporal impairments.

In Qian *et al.* [29] used SVM to predict MOS with a 91.3 PLCC, taking into account colour information (CI), frames per second (FPS), encoding bitrate (EBR), resolution, initial buffering delay, and rebuffering time ratio (RTR), among other variables. However, SVMs based models take longer to train and are difficult to adapt to different conditions, they also discovered that an SVM-based model with low computing complexity can be utilised to estimate QoE in real time for HTTP video streaming services. Also Ahmad *et al.* [10] used supervised learning techniques for QoE prediction of HAS considering seven different ML models i.e, SVR, RF, GB, SGD, NN, KNN and DT and also provided their comparative analysis. They computed MOS with an accuracy of 80.6%. They also discussed the importance of learning curves and the execution time of ML approaches used in their work. Their comparative study's experiments are based on a Waterloo QoE dataset [43] of short video sequences (average duration of 13 seconds), which may limit the comparison to simply brief video clips. As a result, more effort is needed in the future in order to build a large video sequence dataset for the development and comparison of ML-based QoE prediction models according to their studies. Besides state-of-the-art methods Zhou *et al.* [30] worked on deep feature representations using off-the-shelf DCNN models based on spatio-temporal human visual perception for quality of video-streaming were used, and the findings were encouraging. Their future work considers the investigation of adaptive video streaming quality assessment where deep neural networks were used to build immersive 3D/stereoscopic video streaming QoE assessment techniques. For more accurate adaptive video streaming QoE assessments, Taha *et al.* [31] created the LASSO regression method to predict a correlation between network parameters, video quality, and end-user QoE device capacity. While predicting the Degradation Mean Opinion Score (DMOS), their Mean Squared Error (MSE) was minimised using the LASSO regression model, reaching 0.0036, which is closer to zero. Their future advice considers to plan using the proposed method on a more sophisticated system, such as a mobile video streaming service, and study utilising a deep learning model to train to acquire data on the parameters to measure QoE, as well as employing other devices and video codecs.

For H.264/AVC video streaming services over wireless networks Kang *et al.* [32] proposed a no-reference, content-based QoE estimation approach using Radial Basis Func-

tion Network (RBFN), a type of artificial neural network (ANN) with subjective and objective metrics predicting MOS with 0.89 Pearson's Linear Correlation Coefficient (PLCC) and 0.28 Root Mean Square Error (RMSE) claiming that the model's computational complexity is low. The application layer, network layer, and user equipment content features and parameters are utilised. However, the playstats buffer metrics for HAS video streaming services, on the other hand, are not taken into account.

The studies in [33–37] use prediction models for mobile video streaming to assess the influence of cross-layer Influencing Factors (IFs), such as QoS components from the application and physical layers. Danish *et al.* [33] suggested a cross-layer prediction model based on Random Neural Networks (RNN) for estimating the perceptual quality of mobile video in no reference mode. The model takes advantage of crucial video quality characteristics. The simulation results reveal a high level of predictability, with (R^2) correlation of 0.90 and a root mean squared error of 0.39. However, additional QoS parameters are not included in their study. Also, Youssef *et al.* [34] investigated the efficiency of incremental learning to predict QoE of mobile video streaming considering MOS using the incremental multiclass SVM approach (multiclass-iSVM) with a classification accuracy of 89% with execution time of 60 milli-seconds (ms). Their future work focuses on verifying these results on larger datasets. However, their solutions rely on single learners who have a limited understanding of the QoE data. With an accuracy of 85% R^2, Minovski *et al.* [35] used the RF regression model to measure video QoE for mobile networks and predicted video QoE related to MOS. Their future work is to figure out how service providers might use these estimates to improve their service quality. Their model, however, performed poorly, with a regression error (RMSE) of more than 10%, which is clearly unsatisfactory for QoE prediction at any time. Similarly, Tao *et al.* [36] investigated the relationship between network parameters and subjective QoE scores for mobile video transmission using a deep neural network approach for QoE prediction on a large-scale QoE dataset, which has around 80000 pieces of data regarding four types of subjective scores and 89 network metrics. They also found that their proposed approach has a 0.8686 RMSE and a 0.7609 MAE. However, they don't examine the link between prediction accuracy and dimensionality reduction. Zhang *et al.* [37] also developed a deep learning approach called DeepQoE which uses DL as a combination of word embedding, 3D convolutional neural network (C3D) and representation learning and predicted QoE with a classification accuracy of 90.94% for mobile video streaming

on VideoSet dataset. However, their study does not account for the impact of changing bitrates during playback.

The works in [38, 39] describe the link between social context elements, user engagement characteristics, and QoE, as well as evaluated the influence of QoS and Quality of Application (QoA) factors, to predict YouTube video streaming QoE. Laiche *et al.* [38] predicted QoE for YouTube video streaming using 3 SL models i.e, KNN, DT and RF. They predicted MOS with a maximum PLCC value of 0.864 for DT. Similarly, In extension to their previous work in [34] Youssef *et al.* [39] suggested model uses boosting support vector regression (BSVR) to investigate the efficacy of integrating several learners rather than a single learner for improving QoE prediction performance. They Boosting Support Vector Regression (BSVR) based QoE ensembling model using 10 SVRs with RBF kernel with RMSE value of 0.475 which provides the least prediction error. However, these studies don't look at the cost-effective analysis of prediction accuracy as compared to our studies.

2.1 Research Gap

After the literature review, we found that previous researchers were applying single learning algorithms to predict QoE. Also, they did not cover important parametric playstat features like initial loading time, stalling events etc. while ignoring the spatio-temporal artifacts which are important aspects of HAS video-streaming services. Most of them generated their own data set over the video-streaming on the same device and did not accompany the various client-side adaptation algorithms while working on only a few key quality factors to predict QoE which did not give us the whole picture to measure QoE in real-time scenario.

Another important missing aspect of measuring QoE is model's execution/training times to predict QoE in a timely fashion to avoid users' grief which was missing while predicting QoE in real-life scenarios in previous studies. Moreover, the models in the previous literature are limited due to the accuracy of the predictive models, limited availability of the dataset with ground truth and the models proposed in the literature can not meet the near-real-time requirement to be deployed in real networks while measuring QoE.

2.2 Summary

In this chapter, we have provided the details of previous works for measuring and predicting QoE using various models and techniques. We further concluded that supervised learning algorithms produce better results. Depending upon the models' performance, we observed that ensemble models provide more accurate results for QoE prediction as compared to single learners.

Chapter 3

Proposed Methodology

Our goal is to develop a powerful QoE estimation model that takes into account multiple influencing factors and investigate the complex relationship between an application, network parameters, and user perception in the context of HAS video streaming services based on ensemble learning methods for QoE modeling. To improve generalizability/robustness over a single estimator, ensemble ML techniques combine the predictions of multiple base models created with a given learning algorithm in order to create a single best prediction model [44]. The proposed methodology in Fig. 3.1 represents the detailed pipeline of our QoE model which consists of five different stages. The stages include data collection/readiness, feature engineering, model optimization and training, model testing/prediction and model evaluation based on QoE performance metric(s) stages. These stages further consist of several steps.

3.1 Data Collection/Readiness

In the data collection/readiness stage, the data set is collected from the world's largest QoE publicly available database, as well as ground truth provided by Duanmu *et al.* [43] in order to construct a pipeline for our video streaming QoE estimation model. The sample size in this database is 450 with each sample representing the QoE feature vector, including the key influencing factors of QoE HAS video streaming with a subjective score of ground truth labels (reality), with a video streaming session of 13 seconds on average, objective QoE metrics, and subjective test scores. The video streaming sessions are generated within the database by six main client-side adaptive algorithms performing

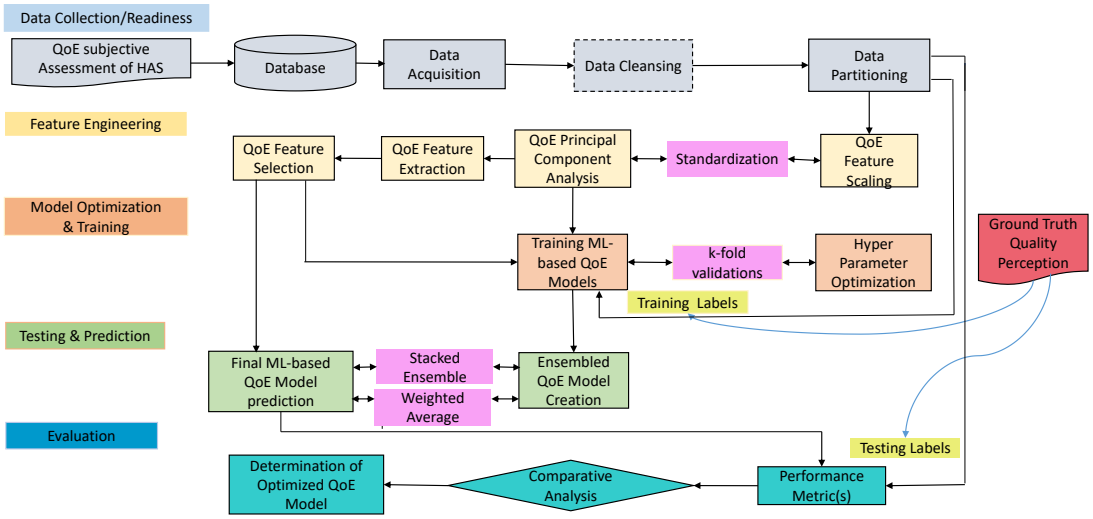


Figure 3.1: Proposed Methodology

under 13 different network bandwidth conditions as evaluated by 34 subjects under realistic conditions. The database is based on a thorough evaluation of objective QoE models. Regarding correlation with human perception, 15 QoE algorithms from four categories are evaluated: signal fidelity-based, network QoS-based, application QoS-based, and hybrid QoE models. There were no missing values, so data cleansing was unnecessary.

3.2 Feature Engineering

The features we employed in our experiments from Waterloo QoE video streaming database [43], are based on the QoE parameters in ITU-T P.1203 [1], which is a parametric bit-stream-based quality assessment/prediction model for HAS. The ITU-T P.1203 standard offers four modes of operation that take into account several QoE KQIs for HAS such as video coding quality, stalling duration, the total number of stalling events, and so on. Table 3.1 lists all of the extracted features indicated by QoE/KQIs. In the feature engineering stage, feature scaling is done for all the input variables in our dataset so that all the variables take on a comparable range of values. We have employed standardization, which is also known as z-score normalization for feature scaling for more accurate results. Standardization scales each input element of the variable separately

Table 3.1: QoE Features of HAS video streaming according to ITU-T P.1203 Standards [1]

Type of features (QoE/KQIs)	***Description***
Initial video loading time	Before starting playback, the video streaming service takes some time to load video segments into the client buffer.
Total number of stalling events (except initial video loading)	During HAS video streaming, player buffer starvation occurs, causing video playback to be interrupted.
Total stalling duration	The total length in seconds of all stalling events that occur during video playback.
Stalling frequency	A metric for how frequently video playback is stalled.
Stalling ratio	The user's perception of quality degradation based on the total duration of stalling events across the duration of video playback.
Time of the last stalling from playback end	The human perception's recency influence on QoE degradation, calculated by using the time stamp difference between the video playback length and the last stalling event's occurrence time.
Playback bitrates	The video representation's visual quality. The average playback bitrate is thought to be strongly linked to user-perceived QoE.
Video quality layer	Due to the adaptive nature of HAS video streaming, the median of the video quality layers being played during video playback is taken into account.
Visual Quality Index	To calculate the visual quality index over the course of a video session using the ITU-T ACR scale (1–5) as a function of playback bitrates, device resolutions, and video encoding resolutions.
Content	The content type of video e.g, animation, sports, news etc.
Frame rate	The number of frames delivered each second.
Bitstream Switching	The bit rate of encoded video after switching between different video quality layers (240p,360p,480p,540p,720p) depending upon bandwidth.

by calculating the mean μ and standard deviation σ of that variable and subtracting the mean μ from each element of that variable and then dividing it by the standard deviation σ to shift the distribution of a variable to a normal range. This is shown by equation (3.2.1), where x is the value of a particular feature, μ points towards the mean and σ points towards the standard deviation of the particular feature [45].

$$z - score = \frac{(x - \mu)}{\sigma} \tag{3.2.1}$$

This is done for all the input variables involved in our dataset to normalize each input QoE feature vector. Moreover, we have also performed Principal Component Analysis

(PCA), to observe the impact of dimensionality reduction of those features on training and execution time of our QoE models [46]. Similarly, for feature selection, to optimise the features set, removing redundant features and incorporating appropriate ones, also to improve the model's computational efficiency and reduce generalisation error in our models, we have used 4 different types of feature selection algorithms [47] from the Python Scikit-Learn module [48].

Univariate Feature Selection

Univariate feature selection uses univariate statistical tests to evaluate each feature independently to determine the strength of the feature's relationship with the response variable. These results are evaluated based on F-score and p-values [48].

Recursive Feature Elimination

The concept behind recursive feature elimination is to continually build a model, select the best or worst performing features, set them aside, and then repeat the process with the remaining features. This procedure is carried out up until all dataset features have been used. Then, features are sorted in order of when they were dropped. In our research, this has been done using Gradient Boosting Regressor (GB) with 5-fold cross-validations to find the optimal QoE features from data set [48, 49].

Select From Model

For this feature selection technique, we have employed Random Forest (RF) as a meta-estimator to find the optimal numbers of features from our data set. Random Forest algorithm creates decision trees on different samples and averages them in the case of regression [50]. In this case, it offers a feature scoring system, based on that, which is used to analyse and optimise our data set [48].

Sequential Feature Selection

Sequential Feature Selection (SFS) is a greedy algorithm that takes multiple features from a set of features and then evaluates them for model iteration, reducing and enhancing the number of features in order for the model to achieve optimal performance

and results. For this purpose we have employed Random Forest regressor (RF) and employed Forward-SFS to select the most appropriate features for our models [48].

3.3 Model Optimization and Training

Lets suppose we have set of n training samples

$X = \{(x_1, y_1), ..., (x_n, y_n)\}$.

Where $x_i = \{QoE\_KQIs\} \epsilon R^m$, $i = 1, ..., n$ are the input QoE key quality indicators, $y_i = MOS \epsilon R$ is the target label which represents the Mean Opinion Score (MOS) and m is the number of input QoE feature vectors. The regression function which is common for all the models is defined as:

$$f(x) = \sum_{i=1}^{n} \phi(x_i) w + b \tag{3.3.1}$$

Where w is the weight vector, ϕ is the non-linear mapping and b is the bias of the regression function

In this stage, we have considered the following four well-known supervised ML algorithms as our base models: Random Forest (RF), Support Vector Regression (SVR), Stochastic Gradient Descent (SGD) and Multi-Layer Perceptron (MLP) based Neural Network. Hyper-parameter optimization of base models is performed in the training stage with the 5-fold cross-validations using grid search CV algorithm [51]. The data set is divided into $k = 5$ subsets using this technique. The model is then trained using $k - 1$ of the subsets and tested using the last one. The testing is repeated $k = 5$ times for all subset combinations to get the best hyper-parameters for each model. Our four trained ML-based QoE prediction models are tested on a testing subset and the predictions made by these base models are then fed to our ensemble models discussed in Chapter 3.4 to get final predictions. The details about our base models are described in this chapter:

Random Forest (RF)

Random Forest (RF) is an ensemble machine learning technique, that combines several base models (Decision Trees) trained on random samples obtained as subsets of original data. In the case of regression, the output of the base models is averaged on different samples to predict the final output. The pros of Random Forest are based on the

idea that even though a decision tree's predictions might not be accurate, utilising a combination of them will increase prediction accuracy and it also covers the problem of overfitting [50].

Support Vector Regressor (SVR)

Since Support Vector Machines (SVM) and logistic regression are related to each other, they are frequently and widely utilised for classification issues in machine learning [52]. However, we have used its regression model in our study. SVR minimises the norm of the squared weight vector $\|w\|^2$ to a quadratic problem by reducing the issue of locating the ideal hyperplane, which is a best-fit line used to predict continuous output for regression that has a maximum number of points within a decision boundary in the training samples [53]. The main advantage of SVM is its low computational cost.

Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent algorithm updates the parameters w and b of the regression model by minimizing the cost function for every single observation instead of updating the entire data set. The parameters w and b are updated on each iteration simultaneously until convergence to the global minimum [54].

Multilayer Perceptron Regressor (NN)

In our work, we have also employed a deep learning technique called Multilayer Perceptron Regressor which is a feed-forward artificial neural network model, also called Deep Neural Networks (DNNs) to measure QoE based on human perception. An MLP Neural Network (NN) typically consists of an input layer, a number of hidden layers, and an output layer. It propagates an input through hidden layers using weights, biases, and activation functions to produce an output [55].

Voting Regressor (VR)

A voting ensemble is a machine learning ensemble technique that optimizes the performance of the system by using many models rather than just one model. By combining the outputs of various techniques, this strategy can be used to address classification and

regression problems. The estimators of all models are averaged to obtain a final estimate for regression problems, for which the ensembles are referred to as voting regressors (VRs) [56]. We have used weighted average (WA) by further adjusting the weights of our base models to overcome the issue of average voting (AV); in which all of the base models in the ensemble are accepted as equally effective in spite of their performance. A weight coefficient is assigned to each ensemble member in a weighted average. The weight can either be a floating-point number between 0 and 1, in which case the sum is equal to 1, or an integer starting at 1, which indicates the number of votes given to the associated ensemble member [57].

As discussed earlier RF, SVR, SGD and NN hyper-parameter tuned ML algorithms were chosen as base learners to form the ensemble VR to estimate the QoE. These ML approaches were chosen on the basis of their outstanding performance in the previous literature. Voting regression and the base learners used in the ensemble learning employed in this study improved the performance and outperformed conventional techniques. $\hat{y}_{RF}$, $\hat{y}_{SVR}$, $\hat{y}_{SGD}$ and $\hat{y}_{NN}$ denote the predictions of each single base learner in Figure 3.2. Instead of using average voting in which all the models are accepted as equally effective regardless of their performance, the ranking method, a type of weighted voting, was utilised to modify the weights. The procedure uses a ranking to show how many votes each ensemble received in the weighted average. For instance, if there are four ensemble learners, the best model receives four votes, the second-best receives three, the third-best receives 2 votes, and the worst receives one vote based on their performance. The votes based on performance are represented by w_1, w_2, w_3 and w_4. The final step was to get and evaluate the suggested model's performance in predicting QoE to that of standalone ML techniques. An illustration approach for finding the optimal weights of the base regressors by the ranking method can be found in Algorithm 1.

Stacking CV Regressor (SR)

Stacking is another technique that outperforms VR for predicting an output. In general, stacking is a method for creating a new model by combining the predictions from multiple base models. The base models are the ones from which the new model is constructed [58]. It is constructed on two levels, also called layers, namely Level-0 and Level-1 which is illustrated in Fig 3.3. In Level-0, we used RF, SVR, SGD and NN as our base models.

Algorithm 1 Find the optimal weights of the base regressors

Input: The labels predicted by the base regressors $\hat{y}_{RF}$, $\hat{y}_{SVR}$, $\hat{y}_{SGD}$ and $\hat{y}_{NN}$

Output: The optimal values of the weights W = $\{w_1, w_2, w_3, w_4\}$

max_$rsquare \leftarrow 0$

for $i \in \{1, \ldots, 4\}$ **do**
  for $j \in \{1, \ldots, 4\}$ **do**
    for $k \in \{1, \ldots, 4\}$ **do**
      for $l \in \{1, \ldots, 4\}$ **do**
        Combine the labels of the base regressors
        using these weights
        Calculate the R^2 using actual labels
        if rsq > max_rsquare **then**
          max_$rsquare \leftarrow$rsq
          weight_vector $\leftarrow \{w_1, w_2, w_3, w_4\}$
        end if
      end for
    end for
  end for
end for
return weight_vector

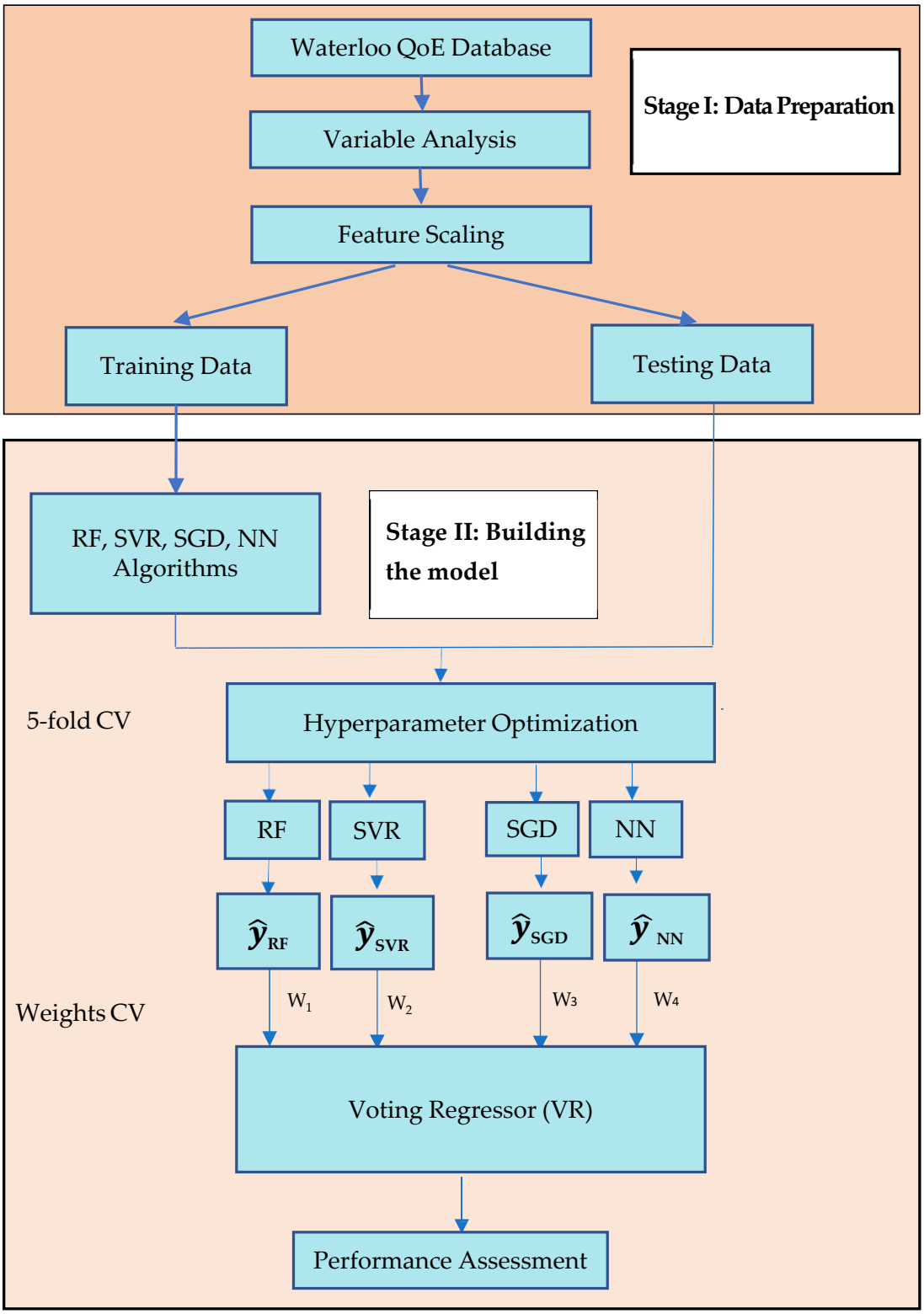


Figure 3.2: Voting Regressor block diagram.

There are two limitations to the traditional stacking procedure. The first limitation of stacking is that the final prediction is equally influenced by each of the base models.

Therefore cross-validation is necessary for each of the base estimators. The second is, the second-level regressor's inputs are prepared using the same training set as the first-level regressors, which could result in overfitting.

To address these issues we have used StackingCVRegressor which provides stacking with cross-validation for base models along with meta-model, also known as a final model, which combines these base models that were learned in parallel to provide the final prediction [59]. For this scenario, we used the Gradient Boosting algorithm (GB) as a meta-model in Level-1 which we will further discuss in detail in this chapter [49]. Secondly, to overcome the issue of overfitting, the StackingCVRegressor, however, makes use of out-of-fold predictions. The dataset is divided into k folds, and in k subsequent rounds, $k-1$ folds are used to fit the first level regressor where $k = 5$. The last 1 subset that was not used for model fitting in each iteration receives the first-level regressors in each round. The predictions made by the first level regressors RF, SVR, SGD and NN be $\hat{y}_{RF}$, $\hat{y}_{SVR}$, $\hat{y}_{SGD}$ and $\hat{y}_{NN}$ respectively. The second-level regressor, that is, GB receives the stacked predictions as input data after they have been made. The first-level regressors are fitted to the full dataset for the best predictions following the training of the StackingCVRegressor. The final prediction produced by SR after training on the predictions of base model is denoted by $\hat{y}_{SR}$.

The reason behind choosing GB as a meta-estimator is because of its prediction speed and accuracy. GB algorithm combines individual decision trees by generating simpler (weak) decision trees sequentially and using the boosting approach to predict the error left behind by the prior model. Moreover, it also helps to minimize the bias error of the base models, which in case of regression is MSE, thus improving the performance in terms of accuracy for this study.

Let's suppose for Level-0 regressors we have a set of n training samples

$X = \{(x_1, y_1), ..., (x_n, y_n)\}$.

Where $x_i = \{QoE\_KQIs\} \; \epsilon \; R^m$, $i = 1, ..., n$ are the input QoE key quality indicators, $y_i = MOS \; \epsilon \; R$ is the target label which represents the Mean Opinion Score (MOS) and m is the number of input QoE feature vectors.

Let's suppose the new data set that is formed after the predictions made by base models be:

$P = \{(p_1, y_1), ..., (p_n, y_n)\}$.

Where $p_i = \{QoE\_Predictions\} \epsilon R^d$, $i = 1, ..., n$ are the input QoE predictions by base models. $y_i = MOS \epsilon R$ is the target label which represents the Mean Opinion Score (MOS) and d is the number of input QoE prediction vectors.

Let $\psi(y, F(b))$ be the loss function. The five steps of the GB algorithm are as follows [60]:

1. Using β as an initial constant value, the following is how it is calculated:

$$F_0(p) = argmin_\beta \sum_{i=1}^{n} \psi(y_i, \beta). \tag{3.3.2}$$

2. The gradient loss function is as follows for $k = 1, 2, \ldots, K$ iterations:

$$y_i{}^* = \frac{\partial \psi(y_i, F(p_i))}{\partial \psi F(p_i)}_{F(p)=F_{k-1}(p)}, i = \{1, 2 \ldots, n\}. \tag{3.3.3}$$

3. The initial model $h(p_i; \theta_k)$ is created by fitting the predicted data in the manner described below; and the parameter θ_k is calculated using the method of least squares in the following manner:

$$\theta_k = argmin_{\theta,\beta} \sum_{i=1}^{n} [y_i{}^* - \beta h(p_i; \theta)]^2 \tag{3.3.4}$$

4. The loss function is minimised to obtain the new model weight:

$$\gamma_{k=argmin_\gamma} \sum_{i=1}^{n} \psi(y_i, F_{k-1}(p) + \gamma h(p_i; \theta_k)) \tag{3.3.5}$$

5. The optimized model is obtained as:

$$F_k(p) = F_{k-1}(p) + \gamma_k h(p_i;_k). \tag{3.3.6}$$

This loop continues to run until a predefined number of iterations are reached or convergence conditions are met. Thus, the computational complexity of SR is reduced due to reduction in the dimensionality of the original data set, fitting to the predictions of the base models and accuracy is improved in terms of R^2. Also, the issue of overfitting is resolved.

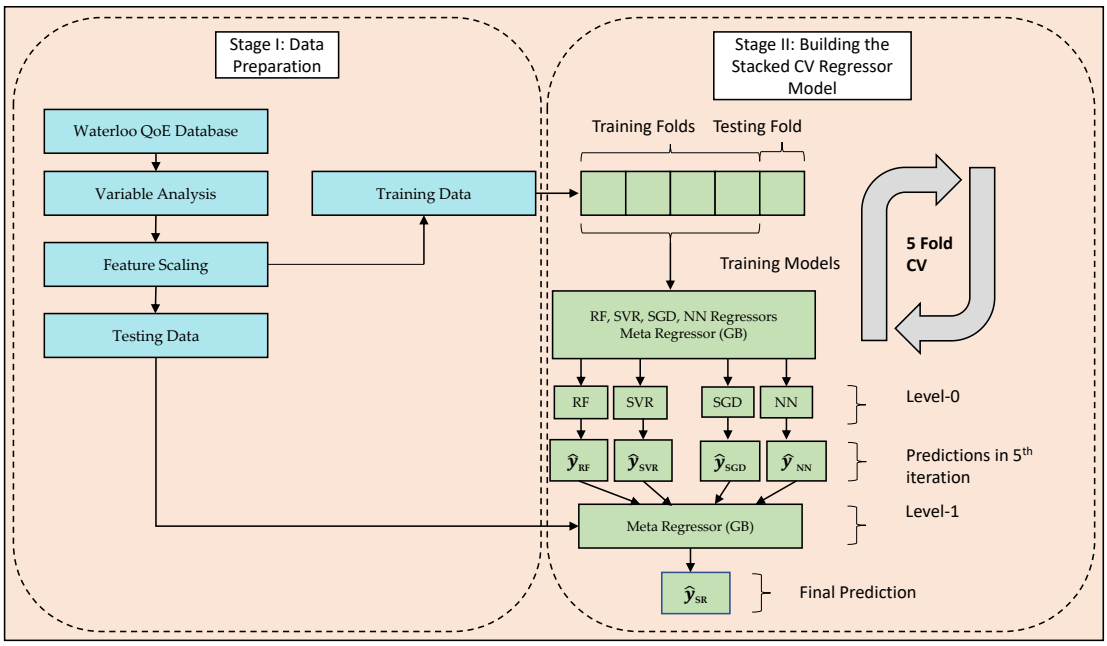


Figure 3.3: Stacking CV Regressor block diagram.

3.4 Model Testing and Prediction

In this stage, we have employed two well know ensemble learning techniques for more accurate predictions: Voting Regressor (VR) and Stacked CV Regressor (SR). In order to get the final QoE prediction, the predictions made by our optimized base models discussed in Chapter 3.3 are fed to our ensemble models to get the final predictions.

3.5 Model Evaluation based on QoE performance metric(s)

In this research, we have employed regression models described in Chapter 3.3 because the subjective test score i.e, MOS is a continuous value with a range of 1 to 5 [1]. In this context, we have considered five different performance indicators for our supervised ML models including Coefficient of Determination (R^2) as an objective criterion for measuring accuracy for each model along with Mean Squared Error (MSE), Root Mean Squared Error ($RMSE$), Mean Absolute Error (MAE), Pearson's Linear Correlation Coefficient ($PLCC$) and Spearman's Rank Correlation Coefficient ($SRCC$) performance measures whose details are briefly described in this chapter.

Coefficient of Determination (R^2)

The coefficient of determination, frequently abbreviated as R^2, assesses how well the model fits the data. It specifically explains the part of the dependent variable's variance that the independent variable can explain for, as shown in Equation (3.5.1). The dependent variable's variation is more accurately explained by the independent variables when the R^2 values are higher [61].

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{3.5.1}$$

Where y_i are the actual values, $\hat{y}_i$ are the predicted values and n are the total number of samples considered.

Mean Square Error (MSE)

The sum of squared errors divided by the total number of predicted values is known as the Mean Squared Error (MSE). This gives larger errors more weight. This is especially helpful in situations when a larger weight for larger errors is desired. Equation (3.5.2) is used to measure it [62].

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.5.2}$$

Root Mean Square Error ($RMSE$)

Root mean squared error ($RMSE$), which scales (MSE) values to be close to the ranges of observed values, is nothing more than the square root of (MSE) [62]. It is calculated using Equation (3.5.3).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.5.3}$$

Mean Absolute Error (MAE)

Mean Absolute Error (MAE) is another metric used for model evaluation for regression purposes. The average of each individual prediction error's absolute value over all the test set occurrences is the mean absolute error of a model with regard to that test set. Each prediction error represents the difference between the instance's true value and the predicted value. Equation (3.5.4) is used to measure it [62].

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3.5.4}$$

Pearson's Linear Correlation Coefficient ($PLCC$)

The strength of a linear correlation between two variables is judged by Pearson's Linear Correlation Coefficient ($PLCC$). It tries to fit a line through the data of two variables and shows how far away from this line of best fit all these data points are. Equation (3.5.5) is used to measure it [63].

$$PLCC = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{3.5.5}$$

Where x_i are the values of the x-variable in samples, $\bar{x}$ is the mean of the values of the x-variable and n are the total number of samples considered.

Spearman's Rank Correlation Coefficient ($SRCC$)

The strength and direction (positive or negative) of a relationship between two variables can be described using Spearman's Rank Coefficient of Correlation ($SRCC$), a non-parametric measure of rank correlation. Equation (3.5.6 is used to measure it [64].

$$SRCC = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2 - 1)} \tag{3.5.6}$$

Where d_i^2 is the difference between the two ranks or dimensions of each observation in samples, and n is the total number of samples considered.

Finally, we have compared and discussed in detail the performance of our base models and ensemble models based on these performance metrics separately in Chapter 4.4 of this literature.

3.6 Summary

In this chapter, we provided a detailed description of our methodology and discussed its various stages for QoE prediction of HAS. We discussed the features in our data set according to ITU-T P.1203 standards [1] and optimization techniques for the data set. We further provided the details of the models we used in this research and performance

metrics on the basis of which the performance of the models will be observed and compared to each other combining with various feature selection techniques.

Chapter 4

Experiments and Results

This chapter provides the details of experiments done for this study. Here the first experiment provides the details about QoE optimization using all the features in our data set [43]. The second experiment provides a detailed analysis of the dimensionality reduction using PCA. In the third experiment, for further optimization of QoE, we have provided the details of experiments for various QoE feature selection techniques discussed in Chapter 3, to optimize our data set for various ML algorithms. The results obtained in the form of heat maps, joint plots and residual plots are further evaluated on the basis of performance metrics discussed previously in Chapter 3 of this literature. Further, we have provided the comparative analysis of ML models by providing the training and testing times of various ML models used for QoE optimization and also provided their learning curves in each experiment. For further evaluation, we have provided a comparison of ML models and the impact of these techniques on the performance of those models on the basis of performance metrics. In Chapter 6 we have provided conclusion and future works for this study.

As mentioned earlier that an ensemble voting regression and stacking cross-validation regression that utilized RF, SVR, SGD and NN to estimate QoE was developed in this work and shown in Fig 3.2 and Fig 3.3 respectively. Combining ML algorithms to provide the capability to effectively predict MOS was the critical factor in the success of the proposed models, which is why they are superior as compared to standalone models. All analyses were performed on a desktop computer with the following configuration: using Python version 3.9.10 and Jupyter Notebook. 16.0 GB RAM, Intel(R) Core(TM) i7CPU processor based on x64 architecture, and a 64-bit operating system. There is a

5-fold cross-validation of all the results.

4.1 All QoE Features

Experiments done for QoE prediction for all the features are discussed and results are displayed in this study without using any dimensionality reduction technique in our data set. First of all, the dataset is imported using Pandas Python library [65].

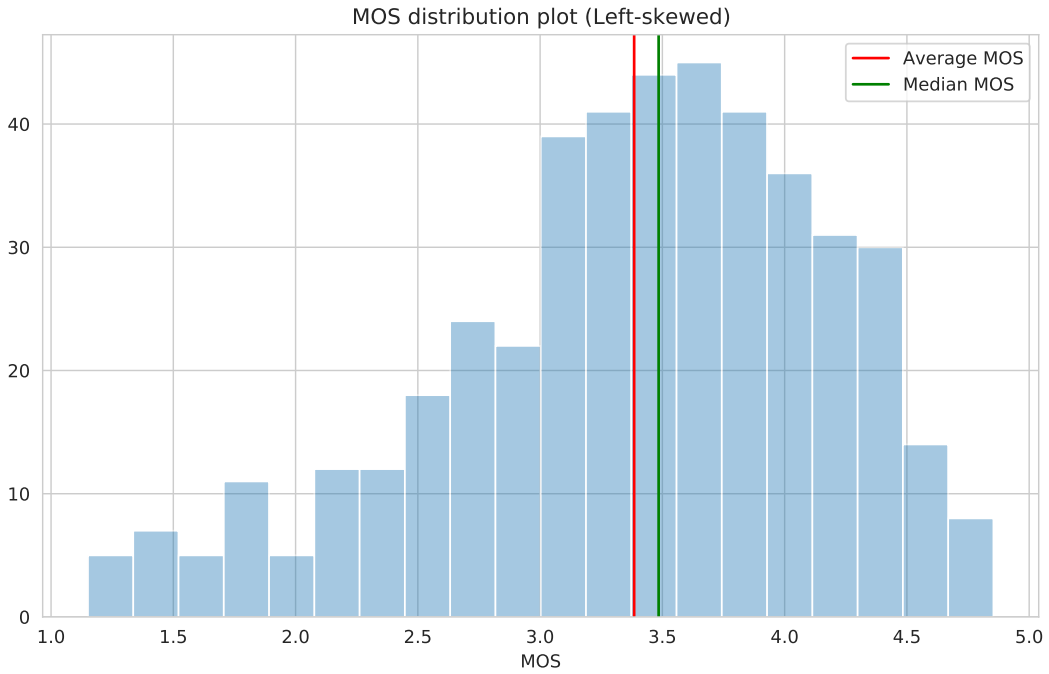


Figure 4.1: Distribution Plot showing the distribution of MOS on the scale of (1-5)

For the purpose of data visualization, we refer to Fig 4.1, which shows the distribution of subjective MOS for both training and testing data sets which is our target label to predict QoE. The histogram's distribution is left-skewed, where the red line represents the average (mean) at 3.38 and the green line represents the median, both lying towards the left side of the peak value which is 4.85.

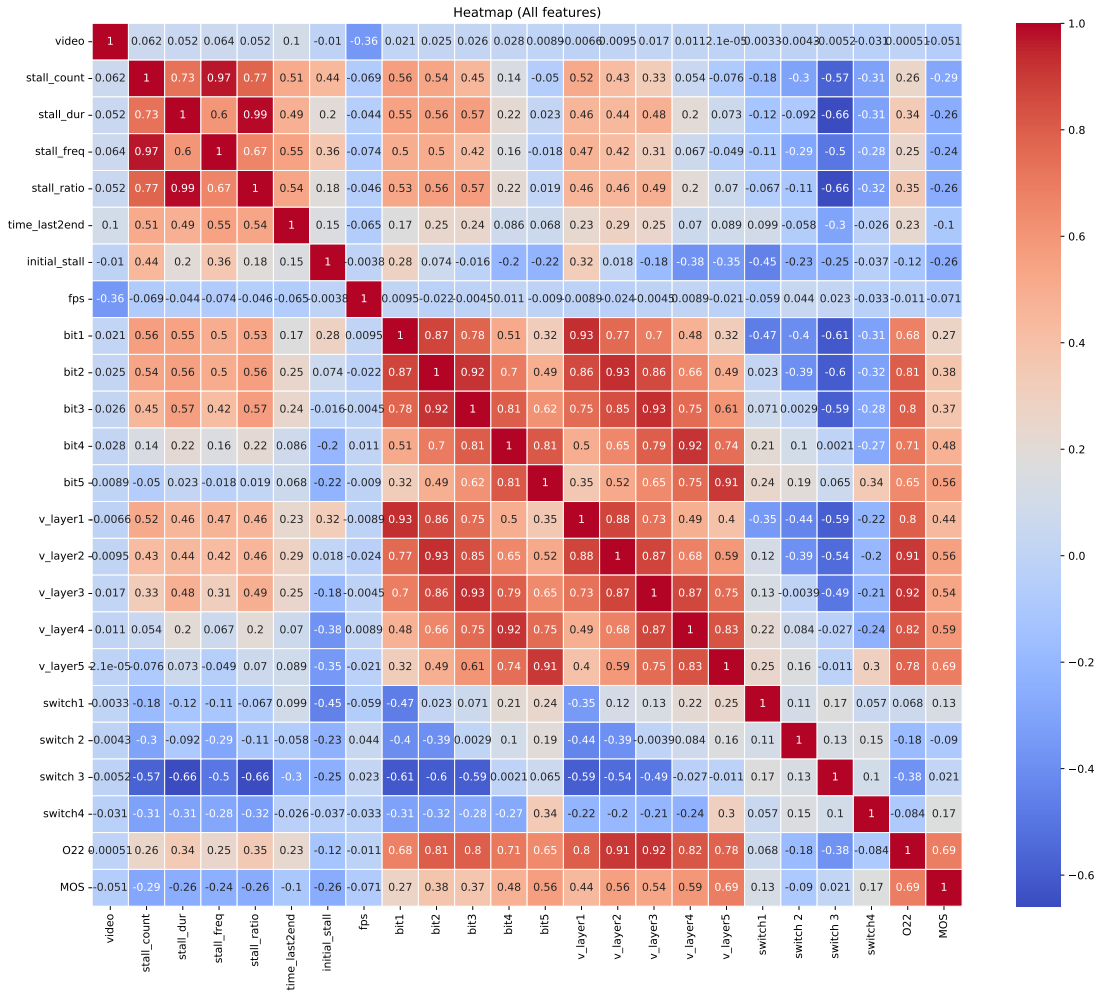


Figure 4.2: QoE features of HAS with correlation existing between various features and MOS using all features

For data exploration, Fig 4.2 shows the heatmap indicating a correlation between all QoE features of HAS including MOS. The colours closer to red represent a strong positive correlation and the colours closer to blue represent a strong negative correlation while the colours closer to grey represent less correlation among the features and so on. From there, it can be seen that video content is strongly negatively correlated with frames/sec; frames/sec itself is negatively correlated with MOS and is also less negatively correlated with all the features. Although changing video content for HAS is an important feature according to ITU-T recommendation [1]. Stalling count and stalling frequency are strongly positively correlated with each other and strongly negatively correlated to switching rate where the media adaptation algorithm switches video playback between a known set of media quality levels, that is layer 3 (480p). Stalling frequency

is strongly positively correlated with stalling count whereas the stalling ratio is strongly positively correlated with stalling duration. Stalling ratio is strongly negatively correlated with bitrate switching at layer 3. Time of the last stalling event is positively correlated with playstat features, that is, the number of stalling events, stalling duration, stalling frequency, stalling ratio etc. while being negatively correlated with MOS. Bitrate at layers 1, 2 and 3 are positively correlated with playstat feature while being strongly positively correlated with video quality layers and also negatively correlated with bitrate switching at layer 3. Bitrates at layer 4 and layer 5 are strongly negatively correlated with stalling duration and stalling ratio while being positively correlated with MOS. Video quality layers 1, 2, 3, 4 and 5 are strongly positively correlated with bitrates at layers 1, 2, 3, 4 and 5 respectively while showing positive correlation with 022 i.e, that is, video quality index, which is a function of bitrates and video quality layers [66]. Bitrate switching at layer 3 is strongly negatively correlated with playstat features which indicates that most of the stalling events occur at layer 3 while being negatively correlated with bitrates at layers 1, 2, 3 and video quality layers 1, 2, 3 respectively. 022 being a function of bitrates and video quality layers is positively correlated with these features and also strongly positively correlated with MOS as indicated by the heatmap shown in Fig 4.2. Fig A.1 represents heatmap using 8 principal components from 0 to 7 showing correlation between various features and principal components themselves; with the first principal component, that is, 0 having greater significance after feature scaling is performed. Similarly, Fig A.2, Fig A.3, Fig A.4 and Fig A.5 represent heatmaps of retained features after performing various feature selection techniques.

For all QoE features, after feature scaling is performed, datasets were divided into training (75% of samples) and testing (25% of samples) sets for the purposes of training and evaluating. Samples were considered at 0 random state, which is common for all feature selection techniques and PCA. Using training data, the individual machine learning algorithms RF, SVR, SGD, and NN were fitted. Each model has been trained utilising a grid search method and five-fold cross-validation considering R^2 as an objective criterion for hyperparameter optimization. In addition to the ML models' default parameters listed in the Scikit-Learn documentation [48], the hyper-parameter optimization employing all QoE features results in the selection of the following parameters:

- For RF, the number of trees (n-estimators) in the forest was 500 and the maximum

features of each tree (max_features) was 'log2'

- SVR has a coefficient of regularization (C) equal to 1 with RBF kernel and type is epsilon which is equal to 0.1
- SGD has maximum epochs(max_iter) equal to 1000 with tolerance limit equal to $1e-3$
- For NN, the learning rate is 'constant' by default while the activation function is rectified linear unit (relu) and maximum iterations are 100 for 'lbfgs' solver for weights optimization with 20 hidden layers.

By combining these four ML algorithms, an ensemble voting regressor (VR) was created using weighted averages based on the results of the individual ML methods. The trained ensemble VR was then fitted. As discussed previously, ($\hat{y}_{RF}$, $\hat{y}_{SVR}$, $\hat{y}_{SGD}$, $\hat{y}_{NN}$) denote the predictions of each single ML algorithm.

- For all features, weights assigned to each prediction results in $(w_1 = 3, w_2 = 1, w_3 = 3, w_4 = 1)$ for VR including all features, respectively.
- For PCA, weights assigned to each prediction results in $(w_1 = 4, w_2 = 2, w_3 = 2, w_4 = 1)$ for VR, respectively.
- For univariate feature selection, weights assigned to each prediction results in $(w_1 = 3, w_2 = 2, w_3 = 3, w_4 = 1)$ for VR including all features, respectively.
- For univariate feature selection technique, weights assigned to each prediction results in $(w_1 = 3, w_2 = 2, w_3 = 3, w_4 = 1)$ for VR including all features, respectively.
- For univariate feature selection, weights assigned to each prediction results in $(w_1 = 3, w_2 = 2, w_3 = 3, w_4 = 1)$ for VR including all features, respectively.
- For recursive feature elimination technique, weights assigned to each prediction results in $(w_1 = 4, w_2 = 1, w_3 = 4, w_4 = 1)$ for VR including all features, respectively.
- For select from model technique, weights assigned to each prediction results in $(w_1 = 4, w_2 = 2, w_3 = 2, w_4 = 1)$ for VR including all features, respectively.

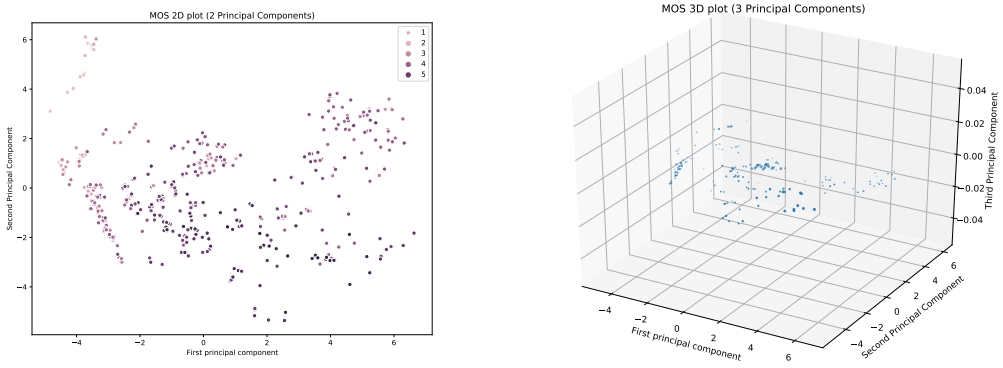
- For sequential feature selection technique, weights assigned to each prediction results in $(w_1 = 3, w_2 = 2, w_3 = 3, w_4 = 1)$ for VR including all features, respectively.

SR has meta_regressor equals to GB while the predictions ($\hat{y}_{RF}$, $\hat{y}_{SVR}$, $\hat{y}_{SGD}$, $\hat{y}_{NN}$) with 5-fold cross-validations for each of the individual base models (RF, SVR, SGD, NN), respectively, are fed to GB. GB utilizes these predictions as training data set while giving parameter use_features_in_secondary equals to 'True', enabling it to train on both base models' predictions and corresponding features data sets, and parameter store_train_meta_features equals to 'True' for storing predictions data set features in the form of array for training purposes. The above parameters are common for all the experiments done for this study.

In section 4.4 we have provided the comparison of results for this experiment.

4.2 Dimensionality Reduction using PCA

In this study we investigated the impact of dimensionality reduction of QoE features using PCA.



(a) MOS 2D plot

(b) MOS 3D plot

Figure 4.3: MOS visualization using Principal Components

In Fig 4.3a and Fig 4.3b MOS can be visualized in 2D and 3D using 2 and 3 principal components, respectively. For this technique, we divided the scaled dataset into 8 principal components.

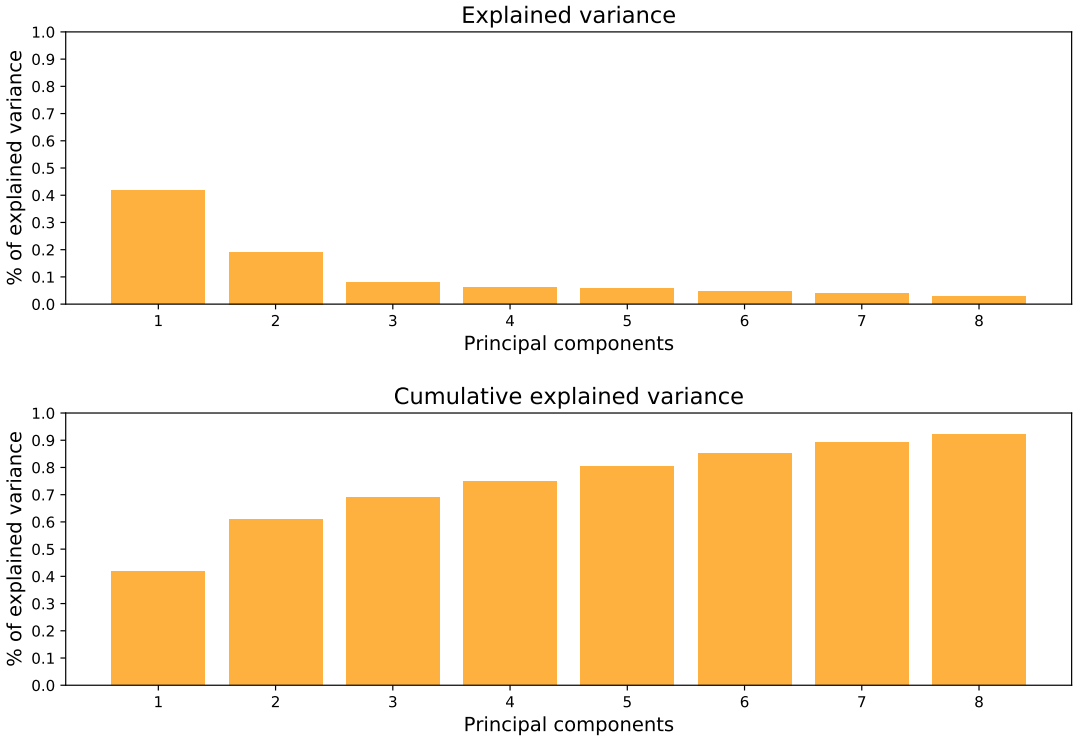


Figure 4.4: Bar charts showing the variance covered by each component and their cumulative variance respectively.

Fig 4.4 shows the explained variance score with the first principal component explaining maximum variance of about 0.42 % approximately, the second principal component explaining 0.19% variance of the dataset and so on. The graph below represents cumulative explained variance by adding the explained variance of each principal component to another, starting from the first principal component and so on. The datasets were divided into training (75% of samples) and testing (25% of samples) sets for the purposes of training and evaluating. Using training data, the individual machine learning algorithms RF, SVR, SGD, and NN were fitted. Each model has been trained utilising a grid search method and five-fold cross-validation considering R^2 as an objective criterion for hyperparameter optimization. In addition to the ML models' default parameters listed in the Scikit-Learn documentation [48], the hyper-parameter optimization employing all QoE features results in the selection of the following parameters using 8 principal components:

- For RF, the number of trees (n-estimators) in the forest was 500 and the maximum features of each tree (max_features) was 'log2'

- SVR has coefficient of regularization (C) equal to 1 with RBF kernel and type is epsilon which is equal to 0.1
- SGD has maximum epochs(max_iter) equal to 1000 with tolerance limit equal to $1e-3$
- For NN, the learning rate is 'constant' by default while the activation function is rectified linear unit (relu) and maximum iterations are 100 for 'lbfgs' solver for weights optimization with 20 hidden layers.

In section 4.4 we have provided the comparison of results for this experiment.

4.3 Feature Selection Techniques

Univariate Feature Selection (UFS)

In this study, we will discuss the experiments done for univariate feature selection technique. For this technique, after feature scaling is performed and 17 features are obtained after selection by setting a parameter percentile equal to 80. The features obtained are shown in Fig A.2. After feature selection is performed, the datasets were divided into training (75% of samples) and testing (25% of samples) sets for the purposes of training and evaluating with 0 random state. Using training data, the individual machine learning algorithms RF, SVR, SGD, and NN were fitted. Each model has been trained utilising a grid search method and five-fold cross-validation considering R^2 as an objective criterion for hyperparameter optimization. In addition to the ML models' default parameters listed in the Scikit-Learn documentation [48], the hyper-parameter optimization employing all QoE features results in the selection of the following parameters:

- For RF, the number of trees (n-estimators) in the forest was 500 and the maximum features of each tree (max_features) was 'log2'
- SVR has a coefficient of regularization (C) equal to 1 with RBF kernel and type is epsilon which is equal to 0.1
- SGD has maximum epochs(max_iter) equal to 1000 with tolerance limit equal to $1e-3$

- For NN, the learning rate is 'constant' by default while the activation function is rectified linear unit (relu) and maximum iterations are 100 for 'lbfgs' solver for weights optimization with 20 hidden layers.

In section 4.4 we have provided the comparison of results for this experiment.

Recursive Feature Elimination (RFE)

In this study, we will discuss the experiments done for recursive feature elimination technique. For this technique, after feature scaling is performed, GB is used with 5-fold cross-validation considering R^2 as an objective criterion for feature selection, each tree's shrinkage coefficient (learning rate) was 0.1, the number of trees (n_estimators) was 100, loss function (estimator_loss) is 'ls' which refers to least squares error to be optimized, with number of features to eliminate recursively after giving parameter 'step' equals to 1 to eliminate one feature with least importance recursively after each iteration while giving 17 features to select. The features obtained are shown in Fig A.3. After feature selection is performed, the datasets were divided into training (75% of samples) and testing (25% of samples) sets for the purposes of training and evaluating with 0 random state. Using training data, the individual machine learning algorithms RF, SVR, SGD, and NN were fitted. Each model has been trained utilising a grid search method and five-fold cross-validation considering R^2 as an objective criterion for hyperparameter optimization. In addition to the ML models' default parameters listed in the Scikit-Learn documentation [48], the hyper-parameter optimization employing all QoE features results in the selection of the following parameters:

- For RF, the number of trees (n-estimators) in the forest was 300 and the maximum features of each tree (max_features) was 'log2'
- SVR has a coefficient of regularization (C) equal to 1 with RBF kernel and type is epsilon which is equal to 0.1
- SGD has maximum epochs(max_iter) equal to 1000 with tolerance limit equal to $1e-3$
- For NN, the learning rate is 'constant' by default while the activation function is rectified linear unit (relu) and maximum iterations are 500 for 'lbfgs' solver for weights optimization with 20 hidden layers.

In section 4.4 we have provided the comparison of results for this experiment.

Select From Model (SFM)

In this study, we will discuss the experiments done for select from model technique using RF. For this technique, after feature scaling is performed, while utilizing RF for feature selection purposes with parameter max_features equals to 17 to select a maximum of 17 features. The features obtained are shown in Fig A.4. After feature selection is performed, the datasets were divided into training (75% of samples) and testing (25% of samples) sets for the purposes of training and evaluating with 0 random state. Using training data, the individual machine learning algorithms RF, SVR, SGD, and NN were fitted. Each model has been trained utilising a grid search method and five-fold cross-validation considering R^2 as an objective criterion for hyperparameter optimization.In addition to the ML models' default parameters listed in the Scikit-Learn documentation [48], the hyper-parameter optimization employing all QoE features results in the selection of the following parameters:

- For RF, the number of trees (n-estimators) in the forest was 500 and the maximum features of each tree (max_features) was 'sqrt'
- SVR has a coefficient of regularization (C) equal to 1 with RBF kernel and type is epsilon which is equal to 0.1
- SGD has maximum epochs(max_iter) equal to 1000 with tolerance limit equal to $1e-3$
- For NN, the learning rate is 'constant' by default while the activation function is rectified linear unit (relu) and maximum iterations are 500 for 'lbfgs' solver for weights optimization with 20 hidden layers.

In section 4.4 we have provided the comparison of results for this experiment.

Sequential Feature Selection (SFS)

In this study, we will discuss the experiments done for the sequential feature selection technique. For this technique, after feature scaling is performed, RF for feature selection purposes with parameter max_features equal to 17 to select a maximum of 17 features

is used along with 5 fold cross-validation to select best features considering R^2 as an objective criterion for the model. The features with best cross-validation score are selected. The features obtained are shown in Fig A.5. After feature selection is performed, the datasets were divided into training (75% of samples) and testing (25% of samples) sets for the purposes of training and evaluating with 0 random state. Using training data, the individual machine learning algorithms RF, SVR, SGD, and NN were fitted. Each model has been trained utilising a grid search method and five-fold cross-validation considering R^2 as an objective criterion for hyperparameter optimization.In addition to the ML models' default parameters listed in the Scikit-Learn documentation [48], the hyper-parameter optimization employing all QoE features results in the selection of the following parameters:

- For RF, the number of trees (n-estimators) in the forest was 300 and the maximum features of each tree (max_features) was 'log2'
- SVR has a coefficient of regularization (C) equal to 1 with RBF kernel and type is epsilon which is equal to 0.2
- SGD has maximum epochs(max_iter) equal to 1000 with tolerance limit equal to $1e-3$
- For NN, the learning rate is 'constant' by default while the activation function is rectified linear unit (relu) and maximum iterations are 100 for 'lbfgs' solver for weights optimization with 20 hidden layers.

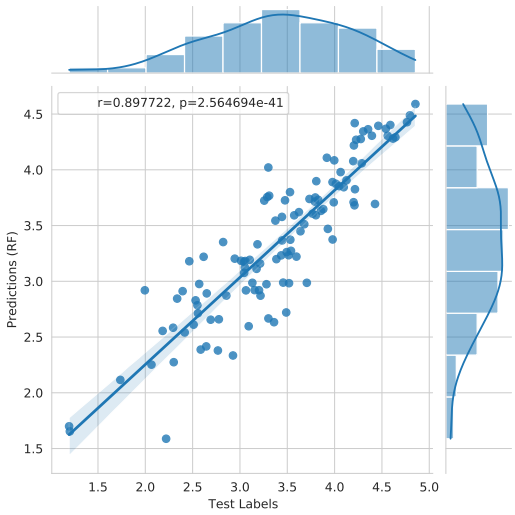
In section 4.4 we have provided the comparison of results for this experiment.

4.4 Comparative Analysis and Discussion

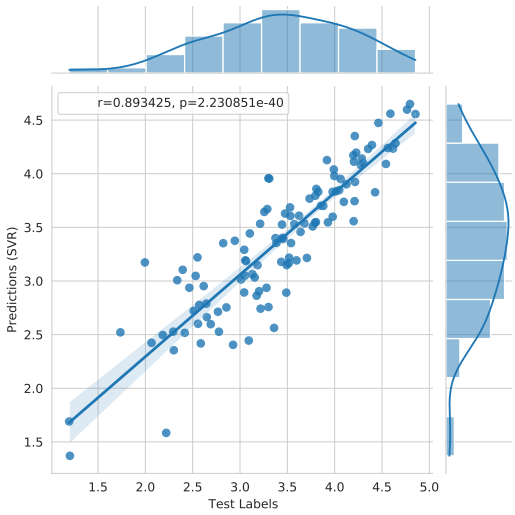
Our ML-based QoE prediction models are compared in this chapter. Further, we have provided the comparative analysis of the single ML models with ensemble ML models in each technique used in this literature and also compared it with the previous literature.

Joint Plots

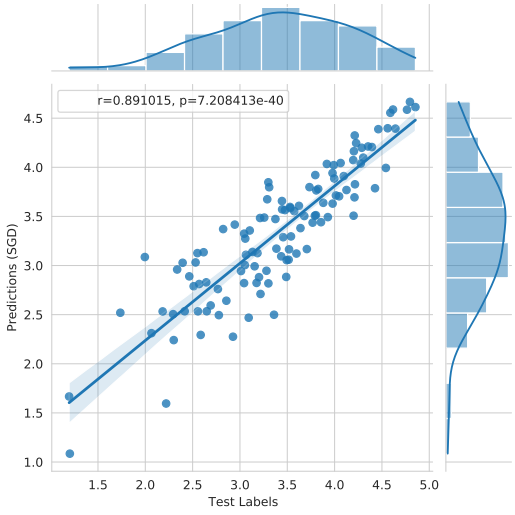
In this study, we will discuss the comparison of joint plots referred to Fig 4.5, Fig 4.6, Fig 4.7, Fig 4.8, Fig 4.9 and Fig 4.10 represents the joint plots of ML models.



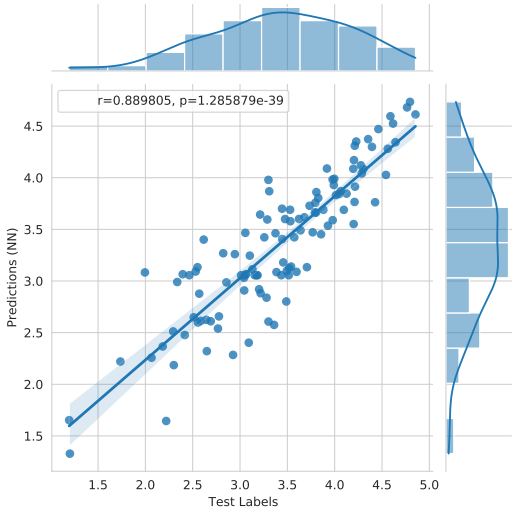
(a) Joint plot RF



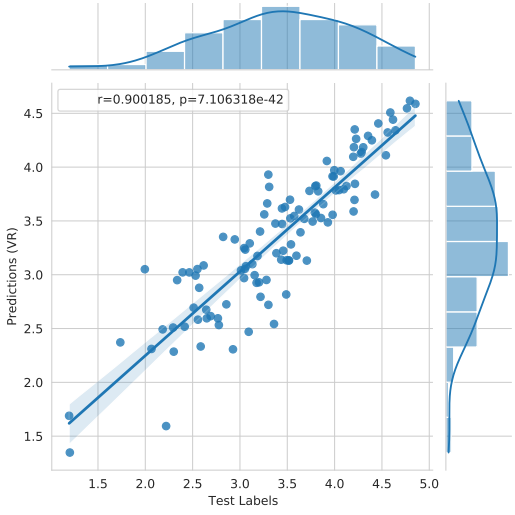
(b) Joint plot SVR



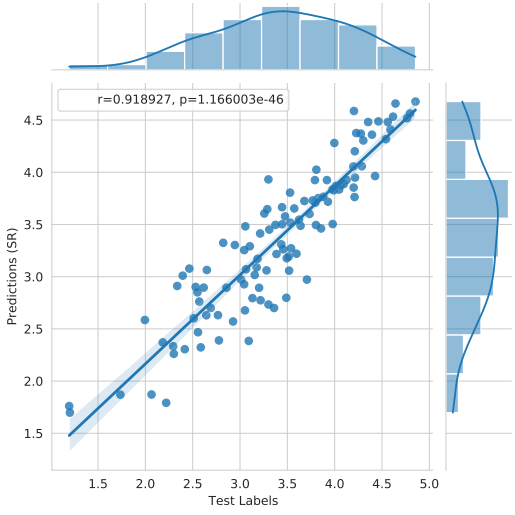
(c) Joint plot SGD



(d) Joint plot NN

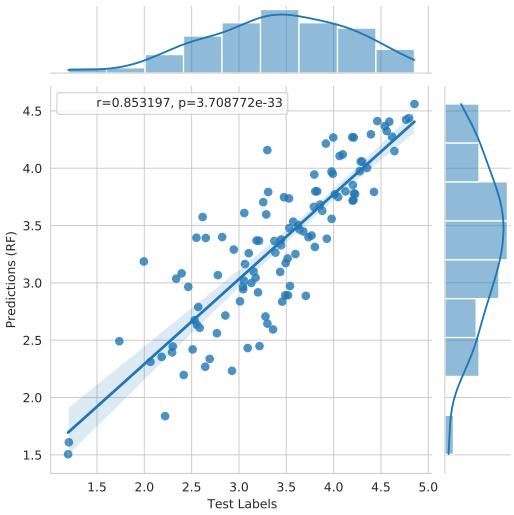


(e) Joint plot VR

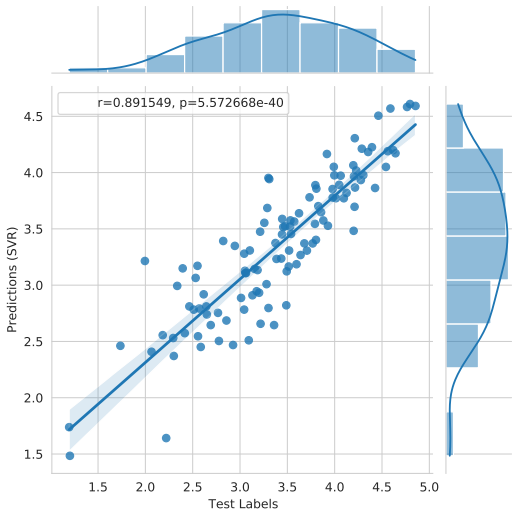


(f) Joint plot SR

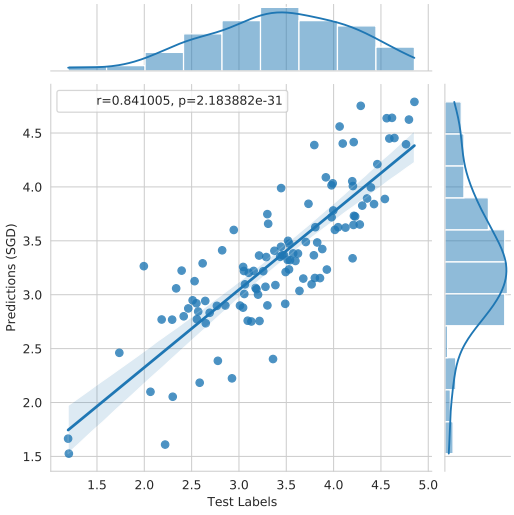
Figure 4.5: Joint plots of Supervised-Learning models applied on all QoE features



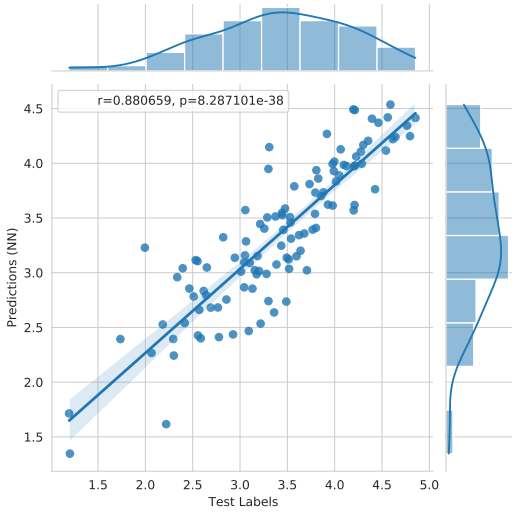
(a) Joint plot RF



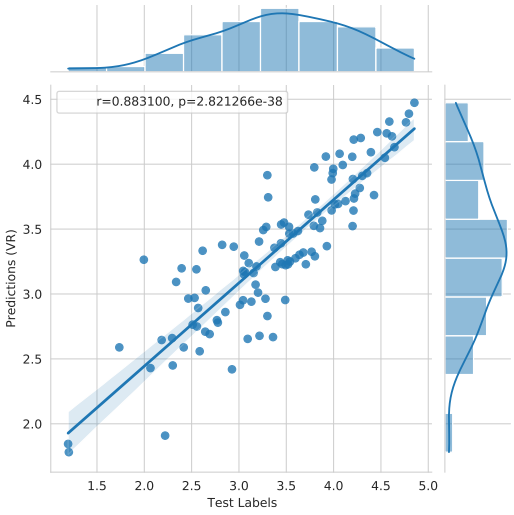
(b) Joint plot SVR



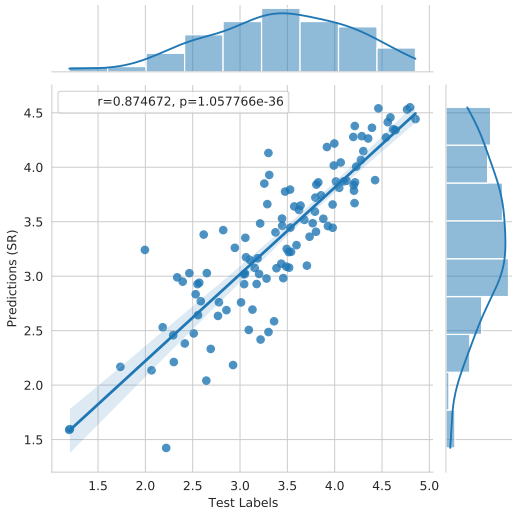
(c) Joint plot SGD



(d) Joint plot NN

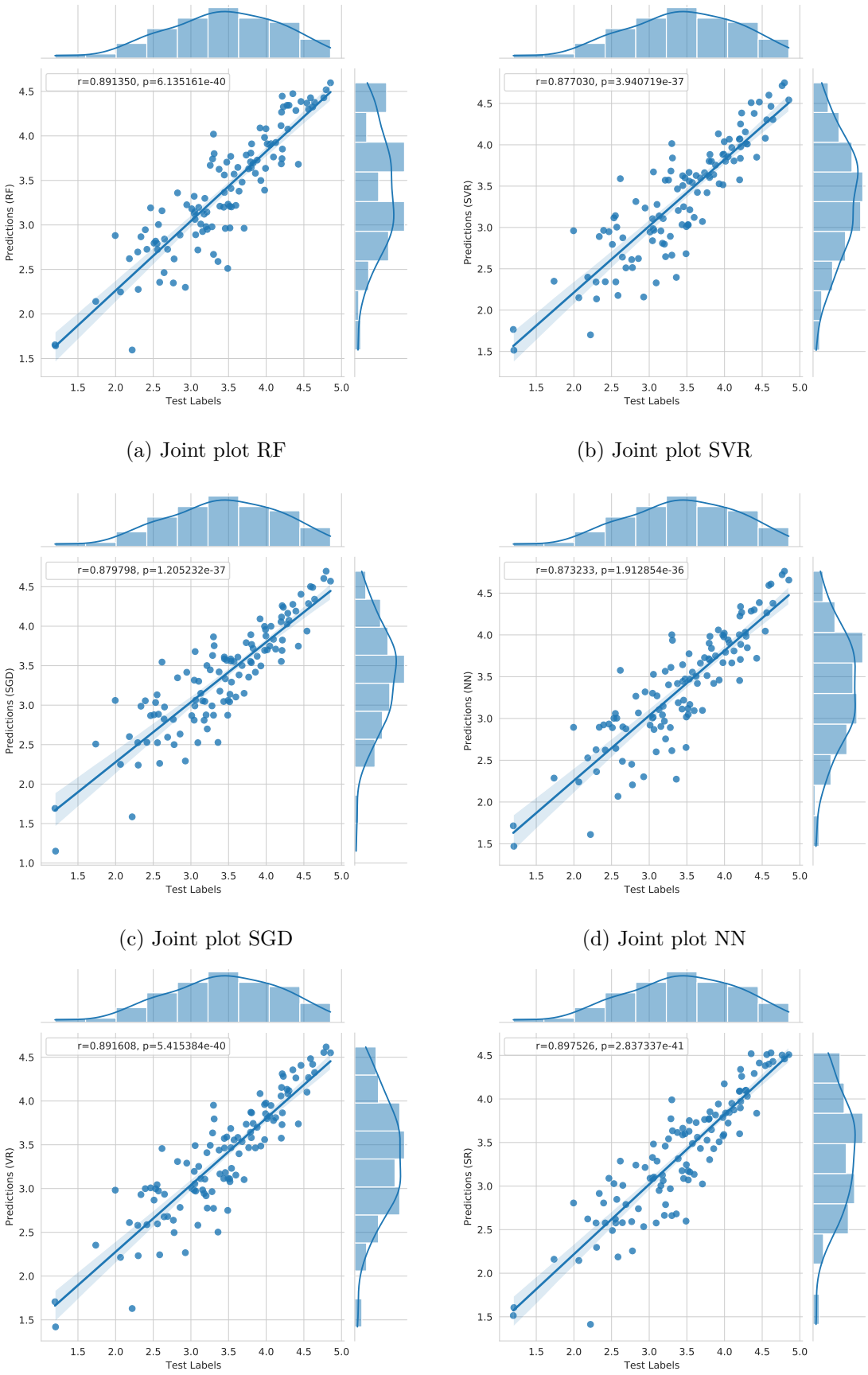


(e) Joint plot VR



(f) Joint plot SR

Figure 4.6: Joint plots of Supervised-Learning models applied on QoE Principal Components



(a) Joint plot RF

(b) Joint plot SVR

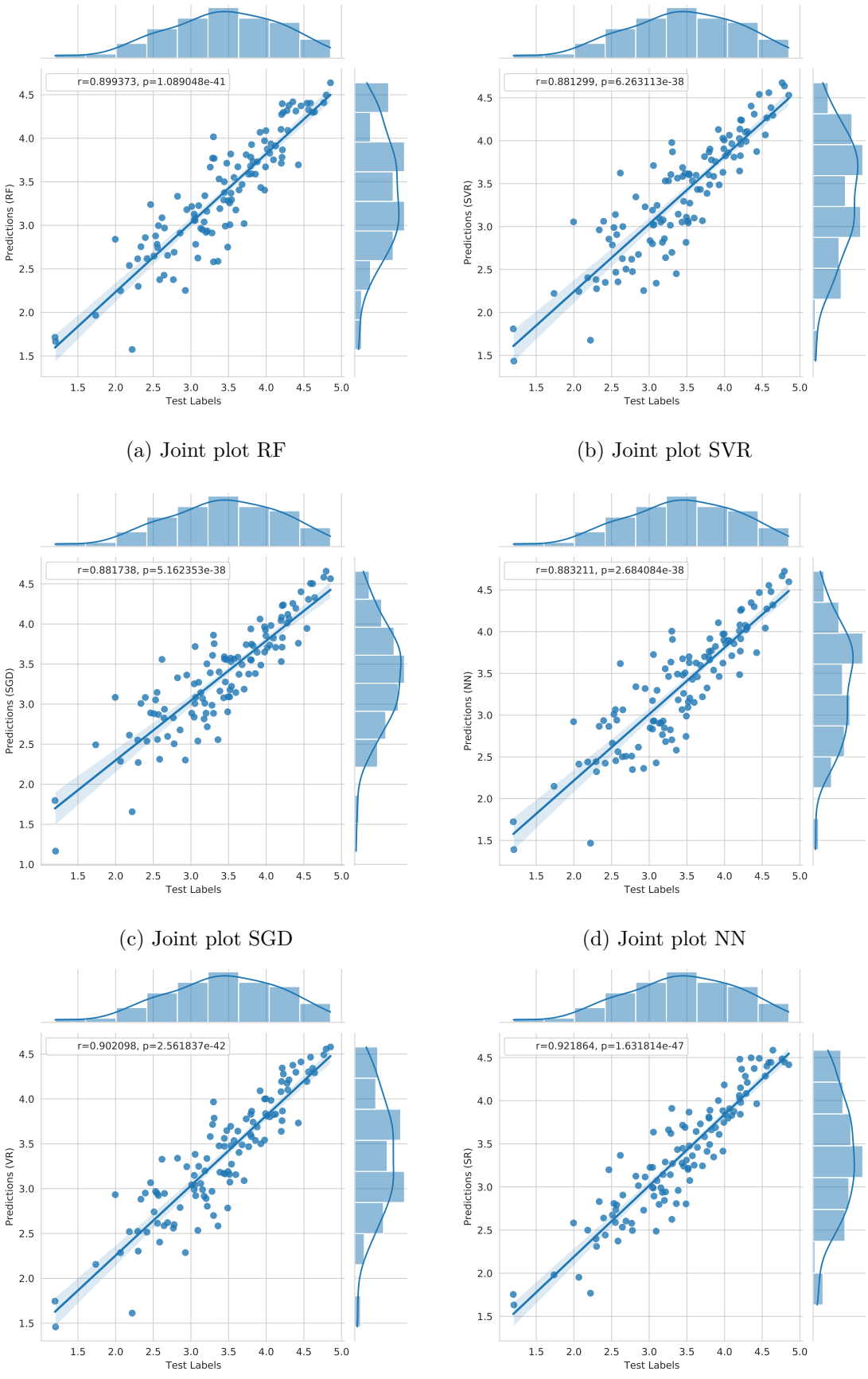
(c) Joint plot SGD

(d) Joint plot NN

(e) Joint plot VR

(f) Joint plot SR

Figure 4.7: Joint plots of Supervised-Learning models applied on all QoE features (Univariate Feature Selection)



(a) Joint plot RF

(b) Joint plot SVR

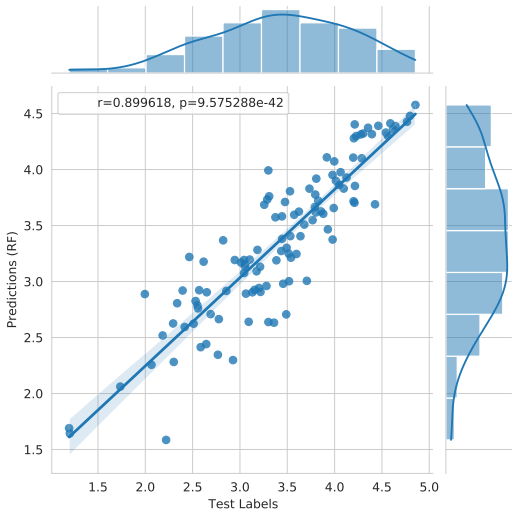
(c) Joint plot SGD

(d) Joint plot NN

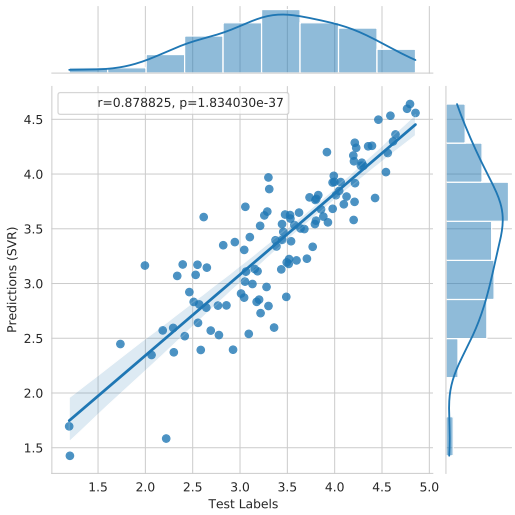
(e) Joint plot VR

(f) Joint plot SR

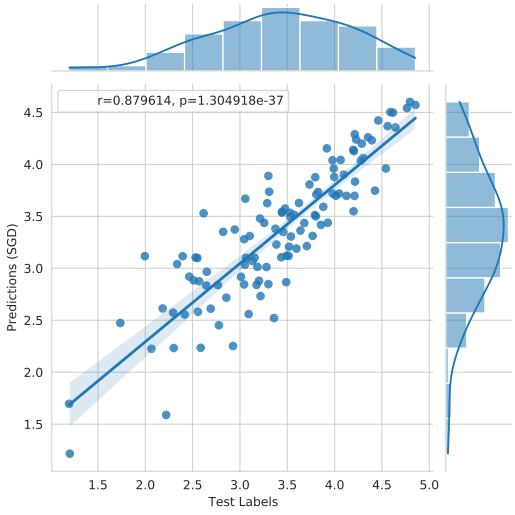
Figure 4.8: Joint plots of Supervised-Learning models applied on all QoE features (Recursive Feature Elimination)



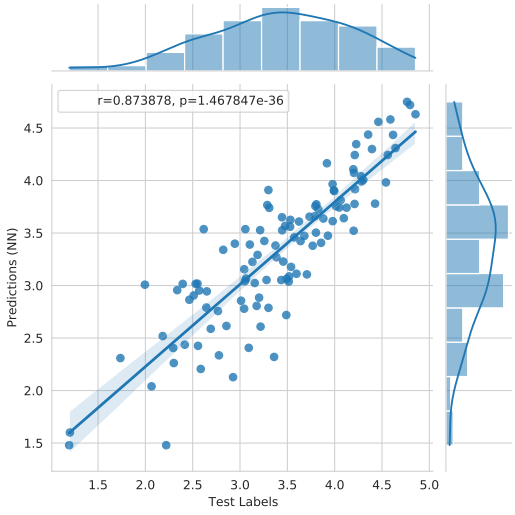
(a) Joint plot RF



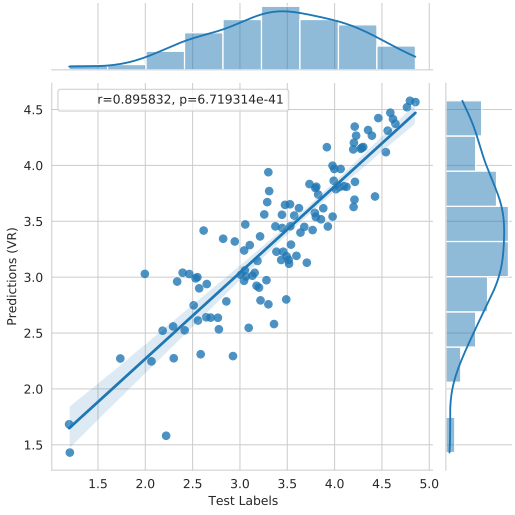
(b) Joint plot SVR



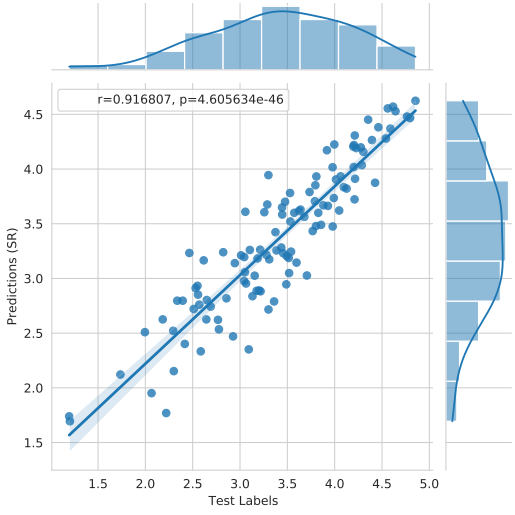
(c) Joint plot SGD



(d) Joint plot NN

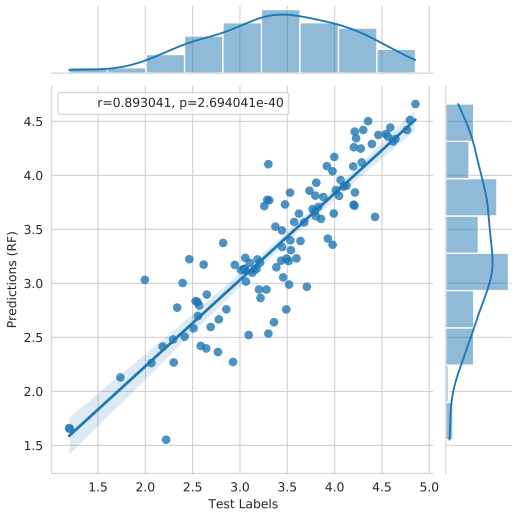


(e) Joint plot VR

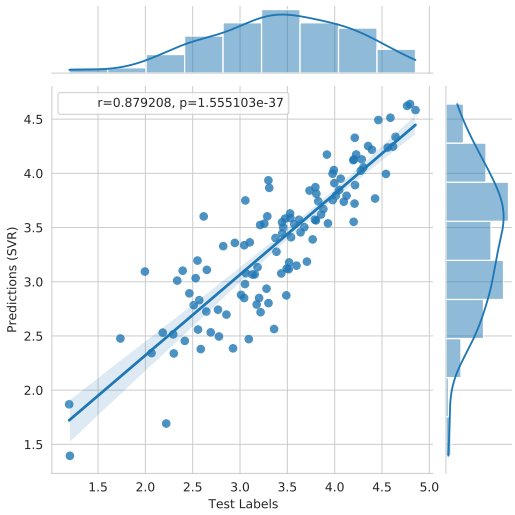


(f) Joint plot SR

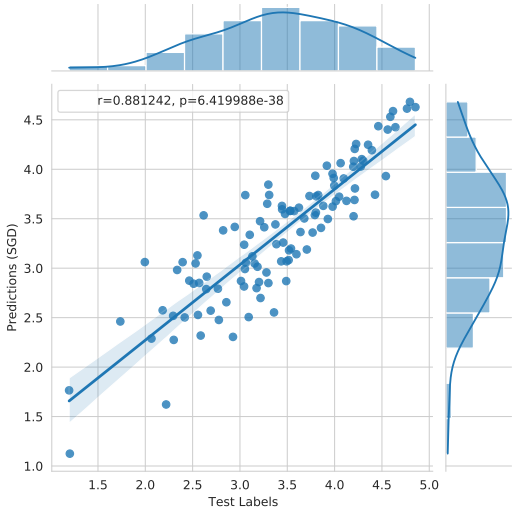
Figure 4.9: Joint plots of Supervised-Learning models applied on all QoE features (Select From Model)



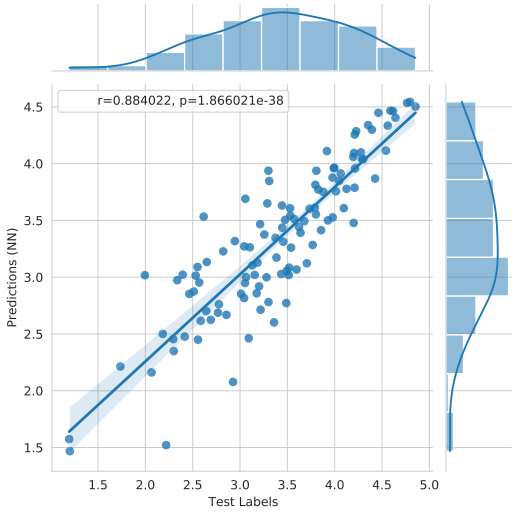
(a) Joint plot RF



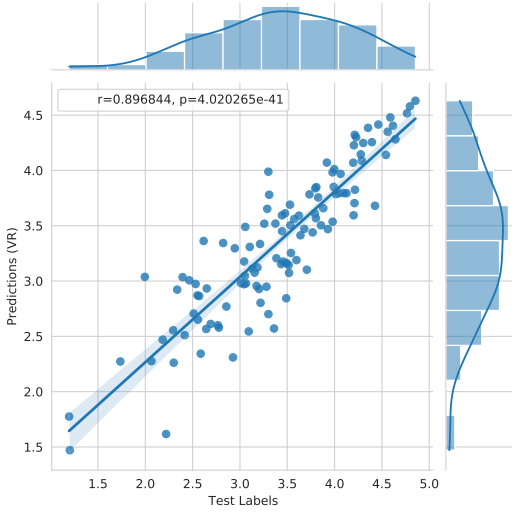
(b) Joint plot SVR



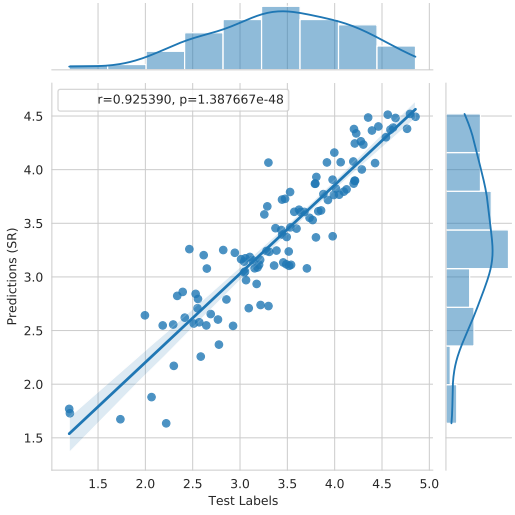
(c) Joint plot SGD



(d) Joint plot NN



(e) Joint plot VR



(f) Joint plot SR

Figure 4.10: Joint plots of Supervised-Learning models applied on all QoE features (Sequential Feature Selection)

In the joint plots shown above, scatterplots are displayed together with two histograms. We observe in the scatterplots, the predicted MOS and subjected MOS (Test Labels) appear to have a positive correlation because they both rise in value as one variable's values do. Because the graph's points are dispersed for the lower values of MOS, the strength of the association appears to be lower for low subjective scores as the Waterloo video-streaming database has fewer samples for low subjective scores. For higher values of subjective scores (MOS), the graph's points are gathered closely or even merged together for higher values of MOS, the strength of the association appears to be stronger for higher subjective scores as the database has more samples for high subjective scores. Both of the marginal histograms are left-skewed because the majority of data are centred on the right side of the distribution while the left side is longer. The graph contains outliers in both the scatterplot and the histogram, which are defined as data points that are significantly different from the other data values. Regression lines or "lines of best fit" illustrate the relationship between a dependent variable and one or more independent variables graphically. The line is drawn in the graphs such that it is as close as feasible to each data point. We can predict the dependent variable for a range of independent variable values by computing the regression line using mathematical equations. Here we have considered values PLCC and p-values for each graph. The regression line on a scatter plot can be used to spot outliers. The data points that deviate most from the regression line are the outliers. The scatterplots shown above have very few outliers. The points are scattered for the worst-performing model and less scattered or even merged for better-performing models.

Residual Plots

In this study, we will discuss the comparison of residual plots refers to Fig A.6, Fig A.6, Fig A.6, Fig A.6, Fig A.6 and Fig A.6 represents the residual plots of ML models used for prediction of QoE using various techniques.

Residual plots show the difference between the actual values of MOS (Test Labels) and predicted MOS. As shown in the above graphs, it can be seen as the histograms of the residuals.

Learning Curves

In this study, we will discuss the comparison of learning curves refers to Fig 4.11, Fig 4.12, Fig 4.13, Fig 4.14, Fig 4.15 and Fig 4.16 for further analysis of the performance of ML models used for prediction of QoE using various techniques. The learning curve describes the model's behaviour in terms of convergence based on the minimization of loss function. Here we have considered MSE as an objective loss function which is to be minimized over the increasing number of training samples with a step size of 10% which is common for all the models with 10-fold cross-validation for training data.

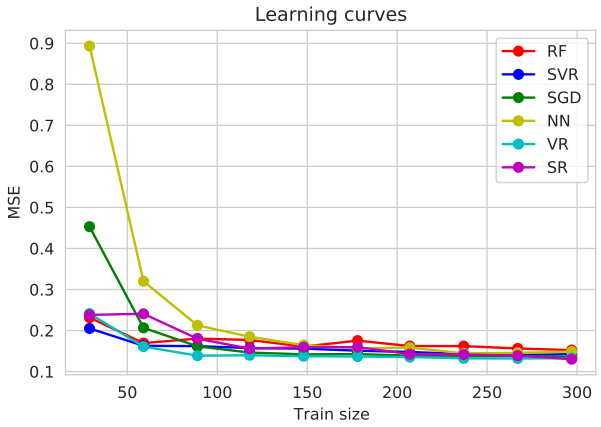


Figure 4.11: Learning curves of ML models (All Features): MSE vs training size.

Learning curves of ML models for all QoE features are shown in Fig 4.11. NN and SGD models show higher values of MSE at the beginning which decrease exponentially with the increase in the training size while RF and SVR show minimum MSE even for smaller training sizes which keeps on decreasing for further increasing the training sample sizes. Similarly, VR and SR models have shown minimum MSE for small training sizes which keep on decreasing further with increasing training step size until convergence is achieved after 240 training samples. So, given the little training data size, ML models like RF, SVR, VR and SR will be a decent option for predicting the QoE of HAS video streaming.

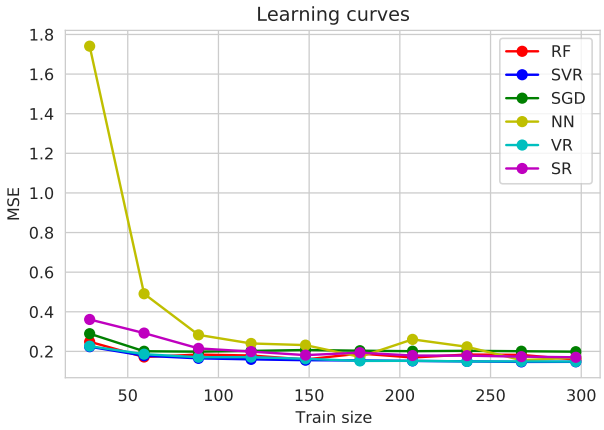


Figure 4.12: Learning curves of ML models (Principal Components): MSE vs training size.

For PCA using 8 principle components, learning curves of ML models are shown in Fig 4.12. NN shows relatively higher values of MSE at the beginning and converges to lower values exponentially while increasing the training sample size. While, RF, SGD, and SR show minimum MSE even for smaller training sizes which keeps on decreasing for further increasing the training sample sizes. Similarly, VR and SVR models performed approximately equal to each other and have shown relatively minimum MSE as compared to other models for small training sizes which keep on decreasing further with increasing training step size until convergence is achieved after 270 training samples.

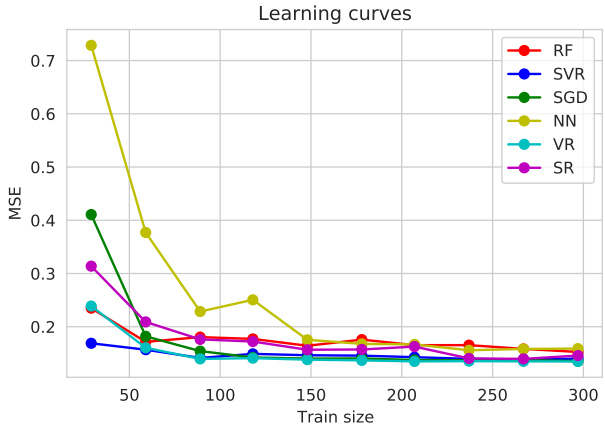


Figure 4.13: Learning curves of ML models (Univariate Feature Selection): MSE vs training size.

For univariate feature selection using 17 features, learning curves of ML models are

shown in Fig 4.13. NN shows relatively higher values of MSE at the beginning and converges to lower values exponentially while increasing the training sample sizes. While, SGD, as compared to NN, has lower MSE values at the beginning and converges to minimum MSE values by further increasing the training sample sizes. Similarly, SVR, VR and SR models have shown relatively minimum MSE as compared to other models for small training size which keep on decreasing further with increasing training step size until convergence is achieved after 270 training samples.

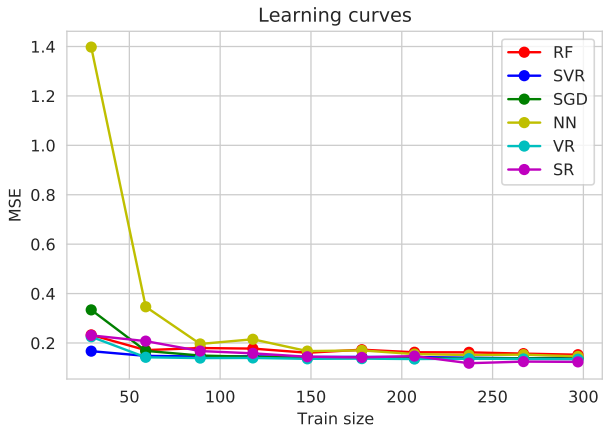


Figure 4.14: Learning curves of ML models (Recursive Feature Elimination): MSE vs training size.

For recursive feature elimination, learning curves of ML models are shown in Fig 4.14. NN shows relatively higher values of MSE at the beginning and converges to lower values exponentially while increasing the training sample sizes. While, RF, SVR, SGD, VR and SR show minimum MSE even for smaller training sizes which keeps on decreasing for further increasing the training samples size. Similarly, VR and SVR models performed approximately equal to each other and have shown relatively minimum MSE as compared to other models for small training sizes which keep on decreasing further with increasing training step size until convergence is achieved after 270 training samples.

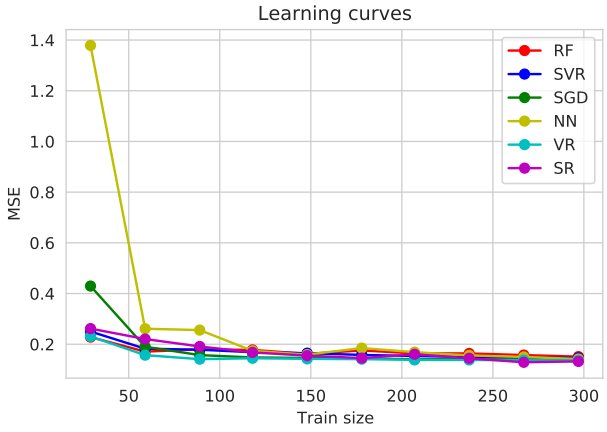


Figure 4.15: Learning curves of ML models (Select From Model): MSE vs training size.

For select from model using 17 features, learning curves of ML models are shown in Fig 4.15. NN shows relatively higher values of MSE at the beginning and converges to lower values exponentially while increasing the training sample sizes. While SGD has comparatively larger MSE at the beginning as compared to RF, SVR, VR and SR models which keeps on decreasing for further increasing the training sample sizes. Whereas, RF, SVR, VR and SR models performed better and have shown relatively minimum MSE as compared to other models for small training sizes which keep on decreasing further with increasing training step size until convergence is achieved after 210 training samples.

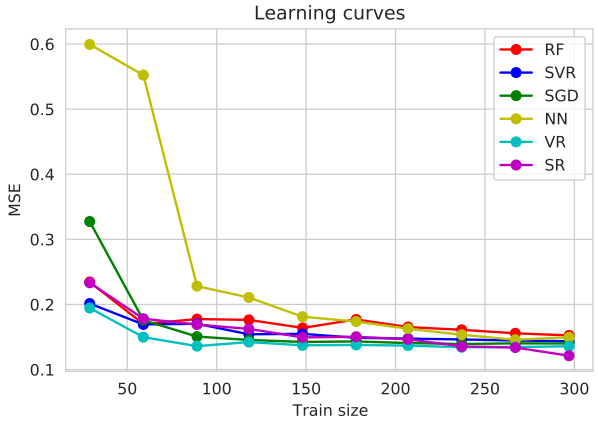


Figure 4.16: Learning curves of ML models (Sequential Feature Selection): MSE vs training size.

For sequential feature selection using 17 features, learning curves of ML models are shown in Fig 4.16. NN shows relatively higher values of MSE at the beginning and

converges to lower values exponentially while increasing the training sample sizes. While SGD has comparatively larger MSE at the beginning as compared to RF, SVR, VR and SR models which keeps on decreasing for further increasing the training samples size. MSE for RF, SVR, VR and SR is smaller for smaller training sample sizes where SR has shown significantly lower MSE values as compared to other models until convergence is achieved after 240 training samples.

Execution/Testing Time vs Training Time

In this study, we will discuss the comparison of execution and training time where Fig 4.17 and Fig 4.18, Fig 4.19 and Fig 4.20, Fig 4.21 and Fig 4.22, Fig 4.23 and Fig 4.24, Fig 4.25 and Fig 4.26, Fig 4.27 and Fig 4.28represents the execution and training times of ML models used for prediction of QoE using various techniques. For real-time applications, execution time is still crucial, particularly the solution must be scalable and cost-effective to run when it comes to real-time QoE prediction.

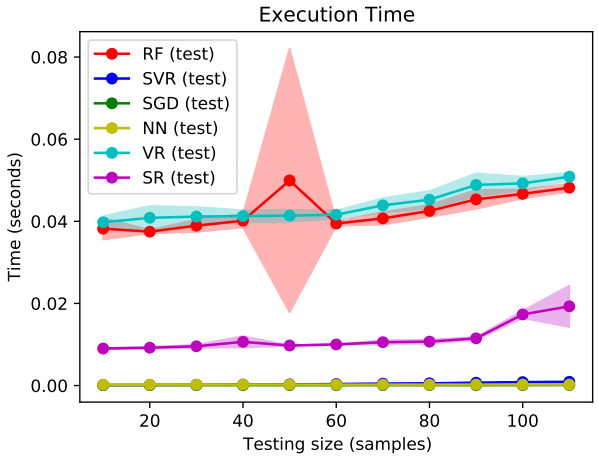


Figure 4.17: Execution (testing) time - All QoE Features.

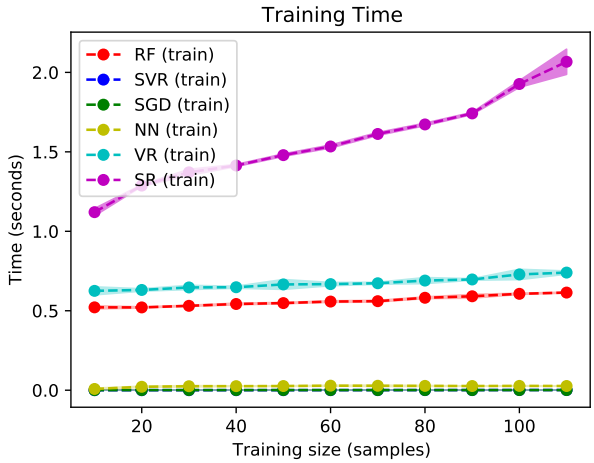


Figure 4.18: Training time - All QoE Features.

Fig 4.17 and 4.18 compare ML models based on the execution time for both the testing and training phases respectively, for all QoE features. For the computation of both training and testing times the tests are repeated 10 times, and 4.17, it is shown how long the ML models typically take to run. In this case, the execution time of SVR, SGD and NN is lowest while VR and RF have the highest execution times with increasing testing data set size. Whereas SR has slightly less execution time as compared to other ensemble models which increases with increment in testing samples. VR and RF models are computationally as costly as other ML models due to longer execution times. However, SR model due to its scalability and accuracy has the advantage of being less computationally expensive in the testing phase for unseen complex data as compared to other ensemble models. For the training time of the ML models in Fig 4.18, SR has maximum training time which keeps on increasing for increasing number of training samples while NN, SVR and SGD models take minimum time for increment in training samples. RF and VR take slightly more time as compared to SGD and NN and less time as compared to SR for model training. However, SR model's computational cost is greater due to its complexity as compared to other models which is a cost-accuracy tradeoff for training purposes.

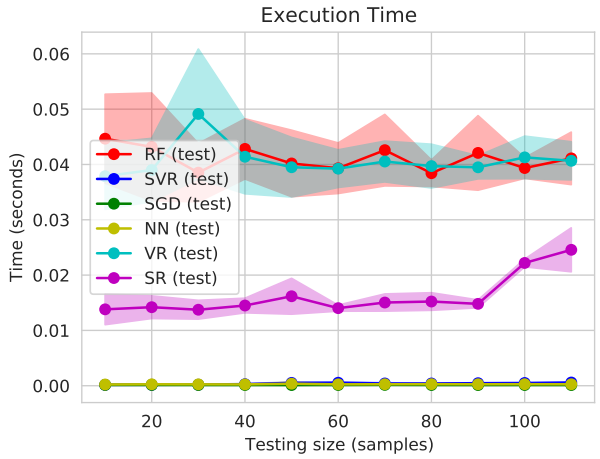


Figure 4.19: Execution (testing) time - Principal Component Analysis.

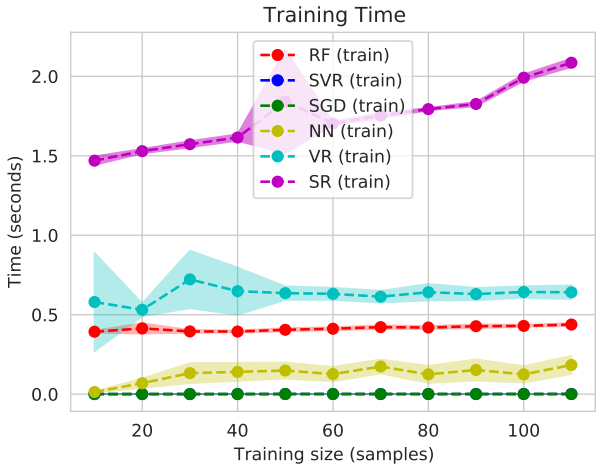


Figure 4.20: Training time - Principal Component Analysis.

Fig 4.19 and Fig 4.20 show the execution/testing and training times of ML models for PCA, respectively. In this case, the execution time of SVR, SGD and NN is the lowest while VR and RF have the highest execution times with increasing testing data set size. Whereas SR has slightly less execution time as compared to other ensemble models which increases with increment in testing samples. VR and RF models are computationally as costly as other ML models due to longer execution times. However, SR model due to its scalability and accuracy has the advantage of being less computationally expensive in testing phase for unseen complex data as compared to other ensemble models. For the training time of the ML models in Fig 4.20, SR has maximum training time which keeps on increasing for the increasing number of training samples while NN, SVR and SGD

models take minimum time for increment in training samples. RF and VR take slightly more time as compared to SGD and NN and less time as compared to SR for model training. However, SR model's computational cost is greater due to its complexity as compared to other models which is a cost-accuracy tradeoff for training purposes.

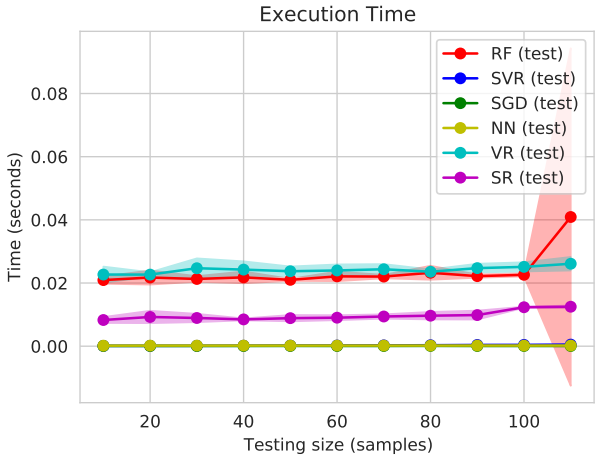


Figure 4.21: Execution (testing) time - Univariate Feature Selection.

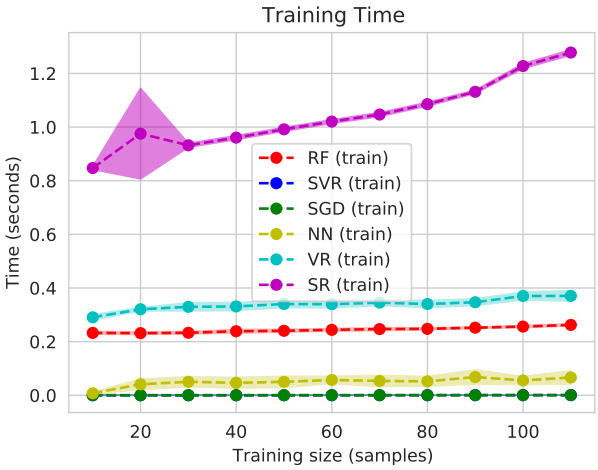


Figure 4.22: Training time - Univariate Feature Selection.

Fig 4.21 and Fig 4.22 show execution/testing and training times of ML models for univariate feature selection, respectively. In this case, the execution time of SVR, SGD and NN is lowest while VR and RF have the highest execution times with RF requiring slightly more time as compared to VR with increasing testing data set size. Whereas SR has slightly less execution time as compared to other ensemble models which increases with increment in testing samples. VR and RF models are computationally as costly

as other ML models due to longer execution times. However, SR model due to its scalability and accuracy has the advantage of being less computationally expensive in the testing phase for unseen complex data as compared to other ensemble models. For the training time of the ML models in Fig 4.22, SR has maximum training time which keeps on increasing for increasing number of training samples while NN, SVR and SGD models take minimum time for increment in training samples. RF and VR take slightly more time as compared to SGD and NN and less time as compared to SR for model training. However, SR model's computational cost is greater due to its complexity as compared to other models which is a cost-accuracy tradeoff for training purposes.

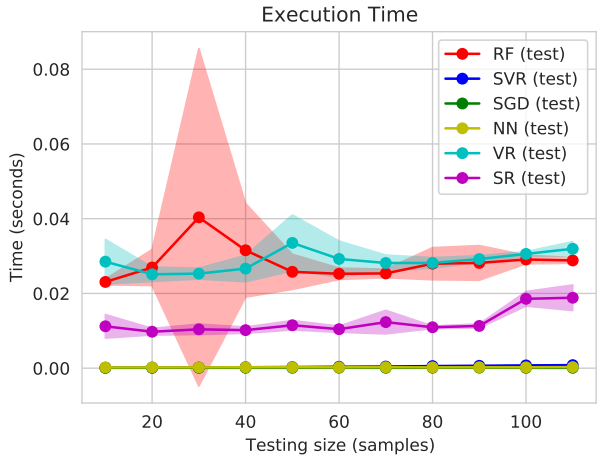


Figure 4.23: Execution (testing) time - Recursive Feature Elimination.

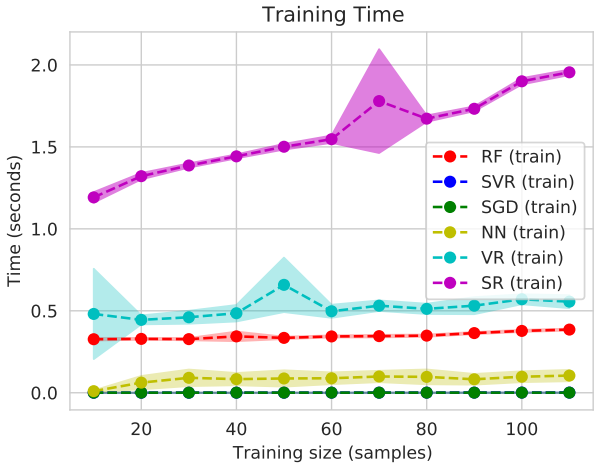


Figure 4.24: Training time - Recursive Feature Elimination.

Fig 4.23 and Fig 4.24 show execution/testing and training times of ML models for

recursive feature elimination, respectively. In this case, the execution time of SVR, SGD and NN is lowest while VR and RF have the highest execution times with increasing testing data set size. Whereas SR has slightly less execution time as compared to other ensemble models which increases with increment in testing samples. VR and RF models are comparatively computationally expensive to other ML models because of higher execution times. However, SR model due to its scalability and accuracy has the advantage of being less computationally expensive in testing phase for unseen complex data as compared to other ensemble models. For the training time of the ML models in Fig 4.24, SR has maximum training time which keeps on increasing for increasing number of training samples while NN, SVR and SGD models take minimum time for increment in training samples. RF and VR take slightly more time as compared to SGD and NN and less time as compared to SR for model training. However, SR model's computational cost is greater due to its complexity as compared to other models which is a cost-accuracy tradeoff for training purposes.

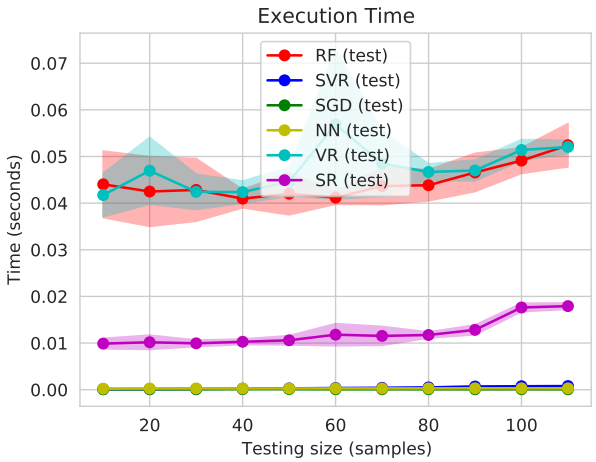


Figure 4.25: Execution (testing) time - Select From Model.

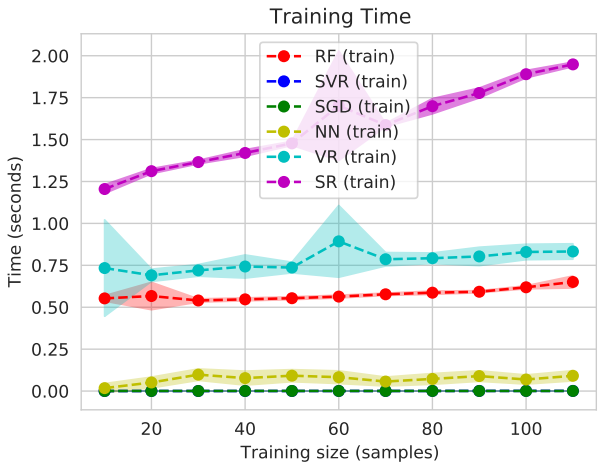


Figure 4.26: Training time - Select From Model.

Fig 4.25 and Fig 4.26 show execution/testing and training times of ML models for select from model, respectively. In this case, the execution time of SVR, SGD and NN is lowest while VR and RF have the highest execution times with increasing testing data set size. Whereas SR has slightly less execution time as compared to other ensemble models which increases with increment in testing samples. VR and RF models are comparatively computationally expensive to other ML models because of higher execution times. However, SR model due to its scalability and accuracy has the advantage of being less computationally expensive in the testing phase for unseen complex data as compared to other ensemble models. For the training time of the ML models in Fig 4.26, SR has maximum training time which keeps on increasing for increasing number of training samples while NN, SVR and SGD models take minimum time for increment in training samples. RF and VR take slightly more time as compared to SGD and NN and less time as compared to SR for model training. However, SR model's computational cost is greater due to its complexity as compared to other models which is a cost-accuracy tradeoff for training purposes.

Figure 4.27: Execution (testing) time - Sequential Feature Selection.

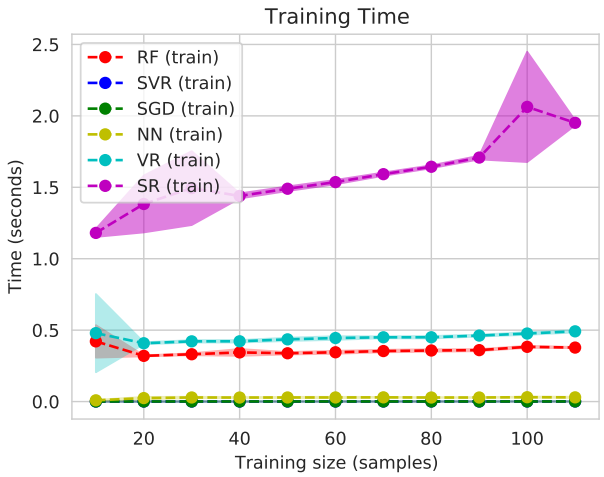


Figure 4.28: Training time - Sequential Feature Selection.

Fig 4.27 and Fig 4.28 show execution/testing and training times of ML models for sequential feature selection, respectively. In this case, the execution time of SVR, SGD and NN is lowest while VR and RF have highest execution times with increasing testing data set size. Whereas SR has slightly less execution time as compared to other ensemble models which increases with increment in testing samples.For the training time of the ML models in Fig 4.28, SR has maximum training time which keeps on increasing for increasing number of training samples while NN, SVR and SGD models take minimum time for increment in training samples. RF and VR take slightly more time as compared to SGD and NN and less time as compared to SR for model training. However, SR model's computational cost is greater due to its complexity as compared to other models

which is a cost-accuracy tradeoff for training purposes.

Performance Metric(s)

In this study, we will discuss the comparison of performance metric(s) refers to Table 4.1, Table 4.2, Table 4.3, Table 4.4, Table 4.5 and Table 4.6 represent the comparison of ML models used for prediction of QoE using various techniques on the basis of performance metric(s).

Table 4.1: Comparison of the supervised learning models (all features).

Metric	RF	SVR	SGD	NN	VR	SR
R^2	0.799837	0.794165	0.786429	0.786028	0.802913	**0.841003**
MAE	0.11594	0.119225	0.123706	0.123938	0.114158	**0.092095**
MSE	0.272041	0.266696	0.279948	0.269443	0.2651	**0.24256**
RMSE	0.521575	0.516426	0.529101	0.519078	0.51487	**0.49250**
PLCC	0.897722	0.893425	0.891015	0.889805	0.900185	**0.918927**
SRCC	0.89398	0.894121	0.889006	0.882877	0.898313	**0.882877**

Table 4.1 presents an overview of a comparison of machine learning methods based on performance measures for all QoE features. In terms of MSE, RMSE and MAE, SR predicts QoE with the minimum MSE, RMSE and MAE values respectively while NN shows the highest MAE value whereas SGD shows the highest MSE and RMSE values. MAE of RF is less as compared to SVR and SGD while MSE and RMSE values are greater as compared to SVR and less as compared to SGD. We noticed that our ensemble models, VR and SR, performed better as compared to standalone models (RF, SVR, SGD and NN models etc.). Similarly, in the case of R^2, SR model outperformed standalone models with a significantly highest score as compared to other models while VR model scored the highest score after SR model. However, in the case of standalone models, RF also being an ensemble of bagged decision trees also better comparably in terms of R^2. The primary cause is that QoE prediction is a challenging non-linear and complicated problem, and our complex machine learning models based on ensemble methods outperformed other models. Moreover, the QoE predicted by the SR algorithm has the highest PLCC and SRCC scores among all taken-into-account ML methods, whereas NN gives the lowest scores for PLCC and SRCC comparably. The

PLCC and SRCC scores depicted by RF, SVR and SGD models are also high for QoE prediction of HAS.

Table 4.2: Comparison of the supervised learning models (principal component analysis).

Metric	**RF**	**SVR**	**SGD**	**NN**	**VR**	**SR**
R^2	0.718058	**0.786145**	0.698732	0.768799	0.748197	0.756119
MAE	0.163308	**0.12387**	0.174502	0.133917	0.145851	0.141262
MSE	0.323927	**0.278463**	0.342297	0.289424	0.309571	0.297128
RMSE	0.569145	**0.527695**	0.585061	0.537981	0.556391	0.545094
PLCC	0.853197	**0.891549**	0.841005	0.880659	0.8831	0.874672
SRCC	0.846497	**0.8921**	0.842588	0.879441	0.88172	0.879441

Table 4.2 presents an overview of a comparison of machine learning methods based on performance measures for principal components. In this case, SVR model combined with principal components outperformed other ML models with considerably lowest values in terms of MAE, MSE and RMSE while SGD showed the highest MAE, MSE and RMSE values, respectively. However, NN also performed better in conjunction with principal components with lower values of MAE, MSE and RMSE after SVR. VR and SR models as compared to RF and SGD models performed better with lower values of MAE, MSE and RMSE, respectively. In the same manner, it can also be noticed that while predicting QoE, R^2 scores of SVR are significantly greater as compared to other ML models with NN predicting the second largest scores of R^2. However, SGD model showed a minimum score in terms of R^2. VR and SR models performed better as compared to RF and SGD models with higher scores of R^2. While predicting QoE, PLCC and SRCC values of SVR are also significantly greater as compared to other ML models with NN having the second highest values of PLCC and SRCC, respectively. Similarly, VR and SR algorithms show high values of PLCC and SRCC as compared to RF and SGD algorithms in QoE prediction of HAS. Here we noticed that our ensemble models did not perform better in conjunction with principal components. We further assume 8 principal components to observe their significance based on SVR model performance.

Model	R2	MSE	Training time (microseconds)
scaled data	0.7941646915445836	0.11922507230269555	536530
7 Principal Components	0.7821613460108972	0.12617771687017498	512819
6 Principal Components	0.7821613460108972	0.12617771687017498	460920
5 Principal Components	0.7821613460108972	0.12617771687017498	447221
4 Principal Components	0.7821613460108972	0.12617771687017498	502105
3 Principal Components	0.711570772886726	0.16706558128844587	454322
2 Principal Components	0.5651885993446668	0.2518538780842694	464547
1 Principal Components	0.3070102629887972	0.4013973701605348	474876
All Principal Components	0.7821613460108972	0.12617771687017498	474181

Figure 4.29: Principal component analysis using SVR.

Fig 4.29 shows the R^2 and MSE criterion is used to check SVR model performance by accumulating various principal components one by one at a time and observing their training time. Here we observe that the model is converged after 4 principal components. So it would not be necessary to combine all the principal components which will give the same results. However, by combining 5 principal components we observe the training time is minimum. Thereby, using only five principal components we would predict QoE more accurately instead of nine principal components. Further, in this context, we notice that SVR predicts more accurately on all scaled features as compared to PCA with greater R^2 and minimum MSE values. Therefore, in case of the above findings, we can say that PCA does not work well for QoE prediction of HAS.

Table 4.3: Comparison of the supervised learning models (univariate feature selection).

Metric	**RF**	**SVR**	**SGD**	**NN**	**VR**	**SR**
R^2	0.79058	0.760581	0.767288	0.756542	0.787851	**0.798917**
MAE	0.121302	0.138678	0.134792	0.141017	0.122882	**0.116472**
MSE	0.279837	0.29124	0.293485	0.294923	0.278551	**0.275831**
RMSE	0.528996	0.53966	0.541742	0.543068	0.527779	**0.525196**
PLCC	0.89135	0.87703	0.879798	0.873233	0.891608	**0.897526**
SRCC	0.886045	0.875075	0.874534	0.872455	0.889123	**0.872455**

Table 4.3 presents an overview of a comparison of machine learning methods based on performance measures using univariate feature selection technique. In terms of MSE, RMSE and MAE, SR predicts QoE with the minimum MSE, RMSE and MAE values respectively while NN shows the highest MSE, RMSE and MAE values. MAE of RF is less as compared to SVR, SGD and VR while MSE and RMSE values of RF are less as

compared to SVR and SGD but greater than that of VR model. Similarly, in case of R^2, SR model outperformed standalone models with significantly highest score as compared to other models while RF model scored highest R^2 score after SR model. However, in case of standalone models, RF also being an ensemble of bagged decision trees also better comparably in terms of R^2. Moreover, the QoE predicted by the SR algorithm has the highest PLCC scores among all taken-into-account ML methods while SRCC scores of SR and NN are the same and are less as compared to RF, SVR, SGD and VR algorithms respectively whereas both SR and NN give the lowest scores for SRCC comparably. VR scored highest PLCC values for QoE prediction after SR algorithm. The PLCC and SRCC scores depicted by SVR and SGD models are also high for QoE prediction of HAS.

Table 4.4: Comparison of the supervised learning models (recursive feature elimination).

Metric	**RF**	**SVR**	**SGD**	**NN**	**VR**	**SR**
R^2	0.802934	0.771016	0.769729	0.770756	0.806548	**0.843597**
MAE	0.114146	0.132633	0.133379	0.132784	0.112052	**0.090593**
MSE	0.271839	0.287215	0.290941	0.291571	0.267207	**0.244069**
RMSE	0.521381	0.535924	0.539389	0.539973	0.516920	**0.494033**
PLCC	0.899373	0.881299	0.881738	0.883211	0.902098	**0.921864**
SRCC	0.89398	0.878834	0.877113	0.876713	0.898762	**0.876713**

Table 4.4 presents an overview of a comparison of machine learning methods based on performance measures using recursive feature elimination technique. In terms of MSE, RMSE and MAE, SR predicts QoE with the minimum MSE, RMSE and MAE values respectively while SGD shows the highest MAE values whereas NN shows the highest MSE and RMSE values. MAE of RF is less as compared to SVR and SGD but greater than MAE of VR while MSE and RMSE values are less as compared to SVR and SGD but greater as compared to VR. We noticed that our ensemble models, VR and SR, performed better as compared to standalone models (RF, SVR, SGD and NN models etc.) in this case. Similarly, in case of R^2, SR model outperformed standalone models with the significantly highest score as compared to other models while VR model depicted the highest R^2 score after SR model. However, in case of standalone models, RF also performed better comparably in terms of R^2. Moreover, the QoE predicted

by the SR algorithm has the highest PLCC scores among all taken-into-account ML methods, while SRCC scores of SR and NN are the same and are less as compared to RF, SVR, SGD and VR algorithms respectively whereas both SR and NN give the lowest scores for SRCC comparably. VR scored highest PLCC values for QoE prediction after SR algorithm. The PLCC and SRCC scores depicted by RF, SVR and SGD models are also high for QoE prediction of HAS.

Table 4.5: Comparison of the supervised learning models (select from model).

Metric	**RF**	**SVR**	**SGD**	**NN**	**VR**	**SR**
R^2	0.803886	0.769724	0.768008	0.752673	0.796213	**0.835644**
MAE	0.113595	0.133382	0.134376	0.143258	0.118038	**0.095199**
MSE	0.271052	0.285102	0.289761	0.301012	0.272054	**0.24944**
RMSE	0.520626	0.533949	0.538294	0.548645	0.521587	**0.499439**
PLCC	0.899618	0.878825	0.879614	0.873878	0.895832	**0.916807**
SRCC	0.896251	0.877628	0.876347	0.877472	0.892059	**0.877472**

Table 4.5 presents an overview of a comparison of machine learning methods based on performance measures using select from model technique. In terms of MSE, RMSE and MAE, SR predicts QoE with the minimum MSE, RMSE and MAE values respectively while NN shows the highest MAE, MSE and RMSE values. MAE, MSE and RMSE values of RF are less as compared to SVR, SGD and VR. Similarly, in case of R^2, SR model outperformed standalone models with significantly highest score as compared to other models while RF model scored the highest R^2 score after SR model. However, RF also being an ensemble of bagged decision trees performed better than other standalone models and VR comparably in terms of R^2. The VR model predicted less R^2 score as compared to RF because the features were selected based on RF feature importance criterion where RF algorithm gives more importance to the features with high cardinality as trees are biased towards those features while giving lower importance to correlated ones and neglecting them at once. Also, VR model assigns weights to other models' predictions and takes a weighted average of their predictions. In this case, if one model does not perform well, then its prediction would affect the prediction of VR model in a slight manner. However, our ensemble SR model still outperformed other ML models when combined with this technique because its prone to overfitting. Moreover, the

QoE predicted by the SR algorithm has the highest PLCC scores among all taken-into-account ML methods while NN depicts lower values in terms of PLCC. The VR model scored a slightly lower value of PLCC as compared to RF while scoring a greater value of PLCC as compared to SVR, SGD and NN. Moreover, RF scored higher values of SRCC while SGD scored the lowest values. The SRCC scores of SR and NN are the same while are less as compared to RF, SVR and VR algorithms respectively.

Table 4.6: Comparison of the supervised learning models (sequential feature selection).

Metric	**RF**	**SVR**	**SGD**	**NN**	**VR**	**SR**
R^2	0.793077	0.769265	0.769563	0.772611	0.797436	**0.852367**
MAE	0.119855	0.133648	0.133475	0.131709	0.11733	**0.085513**
MSE	0.270159	0.287553	0.289872	0.286551	0.271276	**0.220756**
RMSE	0.519768	0.536239	0.538397	0.535304	0.520841	**0.469846**
PLCC	0.893041	0.879208	0.881242	0.884022	0.896844	**0.92539**
SRCC	0.887825	0.878543	0.875116	0.875782	0.894629	**0.875782**

Table 4.6 presents an overview of a comparison of machine learning methods based on performance measures using sequential feature selection technique. In terms of MSE, RMSE and MAE, SR predicts QoE with the minimum MSE, RMSE and MAE values respectively while SVR shows the highest MAE value whereas SGD shows the highest MSE and RMSE values. MAE of RF is less as compared to SVR, SGD and NN while greater as compared to VR. Our VR model has a slightly greater value of MSE and RMSE as compared to RF for this technique. Whereas MSE and RMSE values of VR are less as compared to SVR, SGD and NN models. Similarly, in case of R^2, SR model outperformed standalone models with significantly highest score as compared to other models while VR model scored the highest R^2 score after SR model. However, in case of standalone models, RF also being an ensemble of bagged decision trees also scored better comparably in terms of R^2. The primary cause is that QoE prediction is a challenging non-linear and complicated problem, and our complex machine learning models based on ensemble methods outperformed other models. Moreover, the QoE predicted by the SR algorithm has the highest PLCC scores among all taken-into-account ML methods, whereas SVR gives the lowest scores for PLCC comparably. The PLCC scores depicted by VR model are greater as compared to RF, SVR, SGD and NN models and are also

high for QoE prediction of HAS. Moreover, VR scored higher values of SRCC while SGD scored the lowest values. The SRCC scores of SR and NN are the same while are less as compared to RF, SVR and VR algorithms respectively.

Further, we noted that except for univariate feature selection, all other techniques included video content which is an important parameter mentioned in ITU-T P.1203 standard [1] while all the feature selection techniques ignored the frames per second parameter which is considered to be less important in our study. However, playstats features such as stalling duration, stalling frequency and also bitrates at layer 3 (480p), layer 4 (540p) and layer 5 (720p) are commonly retained in all feature selection techniques. Similarly, video quality layers 2, 5 and O22 are commonly considered which shows the importance of these parameters which must be included while predicting QoE.

The proposed ensemble VR model outperformed standalone models i.e, RF, SVR, SGD and NN models in terms of R^2 for all the feature selection techniques except PCA and select from model. However, SR algorithm outperformed VR and standalone algorithms in all the techniques except PCA in terms of R^2, MAE, MSE, RMSE and PLCC metrics. We also noted that SRCC scores of SR and NN models are the same in all the scenarios. Moreover, another interesting observation we found that in sequential feature selection technique the SR model has shown significantly higher scores of R^2 i.e 0.852367 and significantly lower values of MAE, MSE, RMSE i.e 0.085513, 0.220756, 0.469846 respectively while showing highest PLCC value of 0.92539 as compared to other ML models. Comparisons with prior studies are not possible because there haven't been any studies that used the VR and SR approach. However, the proposed approach performed more effectively and outperformed other machine learning methods in the literature for QoE prediction of HAS [10, 19, 27, 29, 32, 34, 35, 38, 39] with less number of features. As previously mentioned, it is difficult to evaluate and predict QoE because the variety of factors might have an impact. The study's findings suggested that the proposed model for calculating QoE performed well even though it used minimal actual data and just provided the most important input variables. Additionally, the VR and SR algorithms did not call for any realistically challenging assumptions, unlike statistical models.

4.5 Summary

In this chapter, we provided the details of the experiments done for this research. For data visualization, we provided a distribution plot of MOS which is our target variable for predicting QoE. Also provided a detailed principal component analysis. Heatmaps, joint plots, residual plots, learning curves of various ML model and their training and execution times were plotted in each technique to evaluate the results based on performance metrics. We also compared the importance of the features retained in various feature selection techniques based on heatmaps. Further the results obtained from those techniques were compared accordingly.

Chapter 5

Discussion

In Chapter 4 we provided a detailed comparative analysis based on learning curves, execution/training times and performance metrics. In this chapter, we provide further provide discussion based on our experiments while predicting QoE. In the distribution plot in Fig 4.1, we observed that the data is not normally distributed but it is biased towards higher MOS values. As discussed earlier, the red line represents the average (mean) and the green line represents the median, both of which are located to the left of the peak value, which is 4.85. The histogram's distribution is negatively skewed. This is because the number of samples obtained for higher values of MOS is greater as compared to lower values, so there will be a bias-variance trade-off in our models. Where bias is the difference between our model's average forecast and the correct value that we are attempting to predict and variance is the variability of model prediction for a specific data point or value. A model with a large bias ignores the training data and oversimplifies the model. This trade-off is further depicted by the joint plots of our models. It can be seen in joint plots refer to Fig 4.5 for all QoE features, the predicted MOS and the subjected MOS (Test Labels) appear to have a positive connection in the scatterplots since their values both increase as one variable's values do. The strength of the correlation appears to be reduced for low subjective scores as the Waterloo video-streaming database includes fewer samples for low subjective scores. We also observed outliers in our data set for lower MOS values because of fewer samples of lower subjective scores. Higher levels of subjective scores (MOS) cause the graph's points to cluster closer or even merge altogether. Since the dataset contains more samples for higher subjective scores, the correlation seems to be stronger for higher subjective scores. It can also be

seen that the models which performed better i.e, RF, VR and SR models, the data points are closer to the regression line as compared to the models which performed worst i.e SGD and NN models. Moreover the PLCC values of RF, VR and SR models shown in these graphs are also greater which depicts that our ensemble models performed better for QoE prediction as compared to other models. However, PLCC values are significantly higher for SR model which outperformed other models. In Fig 4.6, SVR model in conjunction with PCA outperformed other models and shows higher PLCC values among other models. The data points are also closely merged about the regression line in case of SVR model. But our ensemble models i.e RF, VR and SR did not show higher PLCC values and data points are scattered around the regression line. So PCA does not work with these models. In univariate feature selection technique refer to Fig 4.7, PLCC values of SR model are greater as compared to other models and gives us best-fit regression line between the predicted MOS and the subjected MOS (Test Labels). In recursive feature elimination technique refer to Fig 4.8, we have also observed greater PLCC values of SR model as compared to other models while predicting QoE. In select from model technique refer to Fig 4.9, the SR model outperformed other models and shows higher values of PLCC as compared to other models. Simlarly for sequential feature selection technique refer to Fig 4.10 our SR model significantly outperformed other models and PLCC values are also greater as compared to models and other techniques after performing sequential feature selection technique on our data set which indicates that the features obtained after sequential feature selection technique were more significant as compared to features selected after performing other techniques and gives us best performance while predicting QoE. As mentioned earlier, our goal is to make an optimized QoE model that will minimize the prediction errors (bias and variance) and achieves a good trade-off while still meeting the requirements of both goals. We have tried to overcome the issue of bias-variance trade-off and computational cost by proposing cost-efficient ensemble QoE models in the reference architecture for the ML-based QoE forecasting. Experiments are conducted for the comparison analysis taking into account the main QoE influencing parameters for video streaming. However, in the joint plots, we can say that our state-of-the-art QoE models are overfitted for higher values of MOS and underfitted for lower values of MOS to some extent depending upon the dataset that we have employed in this research. As seen from these graphs, we notice that the points are closely merged for sequential feature selection technique employed with Stacking

Regressor (SR) model which shows our ensemble SR model outperforms other models used in this research. Further, we observed that the residual errors of our ML model in residual plots are normally distributed along the mean which validates that all our regression models were the correct choice for the QoE prediction of HAS.

We further diagnosed the performance of ML models based on learning curves over the training samples considering MSE as an objective criterion. The model's current state may be assessed at each stage of the training process when a machine learning model is being trained. To evaluate how effectively the model is learning, we assessed each model on the training dataset with a step size of 10% which is common for all the models. After comparing the learning curves of ML models for various techniques, we noticed that our models' learning curves were smooth and did not show overfitting. Although all the models converge to the minimum value of objective loss function MSE for increasing training sample sizes, however, when comparing the learning curves of our models we found that RF, SVR, VR and SR models perform comparatively better as compared to NN and SGD model with SR model showing relatively less value for MSE until convergence in all the techniques mentioned in this literature However for sequential feature selection technique, SR model showed less value for MSE thus outperforming other feature selection techniques. Thus, SR model will be a good choice for QoE prediction of HAS video streaming.

For the real-time monitoring of QoE, our models' execution times are considered an important aspect and it is necessary to predict QoE more accurately from devices installed near the end-user, which do not necessarily belong to the ISP. So it's necessary that our models respond in a timely fashion for QoE prediction so that the users don't suffer from waiting times namely stalling caused by rebuffering events also considering the initial delay, or the interval before playback, as well. While comparing the execution times of our ML models, we observed that VR and RF models are computationally costly as compared to other ML models because of higher execution times. However, SR model due to its scalability and accuracy has the advantage of being less computationally expensive in the testing phase for unseen complex data as compared to other standalone and ensemble models. Further regarding our ML models' training times, as mentioned earlier, SR model is trained on the predictions of other base models on each training sample, therefore, it takes more training time with increments in train size. Whereas, VR model takes less training time as compared to SR while it takes more training time

as compared to other models because it assigns weights to other models' predictions and then predicts depending upon other models' performance. From the graphs, we can say that for the sequential feature selection technique, execution times for all the models, especially SR model reduced as compared to all other ensemble techniques applied in this literature. So combining the features retained after applying SFS with SR model gives less computational cost as compared to other ensemble methods with an advantage of more accurate real-time QoE predictions.

Further, we noted that except for univariate feature selection, all other techniques included video content which is an important parameter mentioned in ITU-T P.1203 standard [1] while all the feature selection techniques ignored the frames per second parameter which is considered to be less important in our study. Instead of distinguishing between low and high resolution, our study predicts the six most popular video resolution classes: 144p, 240p, 360p, 480p and 720p. By utilising regression algorithms, it also offers a continuous assessment of the typical video bitrate. The playstats features such as stalling duration, stalling frequency and also bitrates at layer 3 (480p), layer 4 (540p) and layer 5 (720p) are commonly retained in all feature selection techniques. Similarly, video quality layers 2, 5 and O22 are commonly considered which shows the importance of these parameters which must be included while predicting QoE.

As mentioned previously, for our supervised ML models, we have taken into account five different performance measures, including the Coefficient of Determination (R^2) as an objective criterion for measuring accuracy for each model, Mean Squared Error (MSE), Root Mean Squared Error (RM SE), Mean Absolute Error (MAE), Pearson's Linear Correlation Coefficient (PLCC), and Spearman's Rank Correlation Coefficient (SRCC). Thus as far as the performance of ML models is concerned, we observed that our proposed ensemble VR model outperformed standalone models i.e, RF, SVR, SGD and NN models in terms of R^2 for all the feature selection techniques except PCA and select from model. However, SR algorithm outperformed VR and standalone algorithms in all the techniques except PCA in terms of R^2, MAE, MSE, RMSE and PLCC metrics. We also noted that the SRCC scores of SR and NN models are the same in all the scenarios. Moreover, another interesting observation we found that in the sequential feature selection technique the SR model has shown significantly higher scores of R^2 i.e 0.852367 and significantly lower values of MAE, MSE, RMSE i.e 0.085513, 0.220756, 0.469846 respectively while showing highest PLCC value of 0.92539 as compared to other

ML models. Comparisons with prior studies are not possible because there haven't been any studies that used the VR and SR approach. However, the proposed approach performed more effectively and outperformed other machine learning methods in the literature for QoE prediction of HAS [10, 19, 27, 29, 32, 34, 35, 38, 39] with less number of features. As previously mentioned, it is difficult to evaluate and predict QoE because a variety of factors might have an impact. The study's findings suggested that the proposed model for calculating QoE performed well even though it used minimal actual data and just provided the most important input variables. Additionally, the VR and SR algorithms did not call for any realistically challenging assumptions, unlike other statistical models.

5.1 Summary

In this chapter, we discussed the results obtained in detail from our comparative study and compared them with previous literature to propose our optimized ensemble ML model for QoE prediction of HAS.

Chapter 6

Conclusion and Future Work

A more accurate estimation of QoE is necessary from a business perspective for both OTTs and ISPs to enhance users' experience, particularly in video streaming to attract more customers and meet their requirements in terms of QoE. Most video streaming services employ HTTP Adaptive Streaming (HAS). We have outlined the specifics of prior studies that used a variety of models and methodologies to assess and predict QoE. We also came to the conclusion that algorithms for supervised learning yield superior outcomes. Moreover, we found that ensemble models, when compared to single learners, produce results for QoE prediction that are more accurate depending on the performance of the models. We described our process in great depth and covered all of its steps for QoE prediction of HAS. We discussed the features in our data set according to ITU-T P.1203 standards and optimization methods for the data set. In this research, we provided a comparative study of various ML models (RF, SVR, SGD, NN models etc.) and our proposed ensemble VR model was then constructed using these hyper-parameter tuned ML algorithms as base estimators as illustrated in Fig 3.2, which employed the ranking approach to apply weights based on how well each individual ML model performed as well as ensemble SR model which trained on the predictions made by standalone models as illustrated in Fig 3.3, utilizing GB as meta-estimator along with feature selection techniques and also included PCA. Moreover, the data set optimization was done using four feature selection techniques and PCA. Based on learning curves, training/testing execution times, R^2, mean absolute error, mean square error, PLCC and SRCC, the comparative study has offered performance comparison of the machine learning models. RF predicted more accurately as compared to other standalone models

while the overall prediction accuracy of SR model is comparatively greater as compared to other models. Also, we see that SR model is prone to overfitting while predicting QoE. We employed GB as our meta-estimator in SR model which minimizes the bias error of the previous models predictions and provides an accurate measure of QoE. However in PCA, SVR model performed better but this approach does not work for more accurate prediction of QoE. However, the training time of SVR model is lowest as compared to other models but its R^2 value is comparatively lower in other scenarios except PCA. Due to reduction of redundant features in our data set, we have observed the enhancement in the accuracy of all the models in SFS technique instead of utilizing all QoE features. RFE technique also provided more accuracy in terms of R^2 as compared to all QoE features but the features employed in our SFS technique were more reliable as compared to other features from various feature selection techniques and give us a more accurate prediction of target label i.e, MOS. We presented a distribution plot of MOS, which is our objective criteria for forecasting QoE, in order to visualise the data. Detailed principal component analysis was also given. To assess the outcomes based on performance measures, heatmaps, joint plots, residual plots, learning curves of several ML models, and their training and execution times were plotted in each approach. Additionally, we contrasted the weighted value of the characteristics kept in different feature selection methods based on heatmaps. We noticed that the strength of the correlation appears to be smaller for low subjective scores since the graph's points are spread for lower MOS values and the Waterloo video-streaming database includes fewer samples for low subjective scores in joint plots. We also reviewed the limitations of various feature selection strategies for QoE optimization as well as the learning curves, training times, and execution times of various ML models. When comparing the learning curves of ML models for different techniques, we find that while all models converge to the minimum value of the objective loss function MSE for larger training sample sizes, RF, SVR, VR, and SR models perform comparably better than NN and SGD models, with SR showing relatively less value for MSE for sequential feature selection (SFS) until convergence, outperforming other ML models. Moreover, we noticed that SR model takes maximum training time and less execution time as compared to other ensemble methods i.e, RF and VR models. Based on its performance, we finally recommended the best ML model for QoE prediction and concluded that the features retained by SFS technique were most appropriate as compared to features retained after performing other feature selec-

tion techniques because SR has shown significantly higher values of R^2 and PLCC, and lower values of MAE, MSE and RMSE, respectively. The experiments carried out for the comparative study rely on a dataset of short video sequences (average duration: 13 seconds), obtaining less number of samples for lower MOS values, which can restrict the comparative analysis to just brief video sequences and models will be undertrained for samples containing the lower values of MOS. As a result, additional effort will be needed to build long video sequence data sets and collect more samples of video sequences with lower values of MOS from users' perspectives for the development of ML-based QoE prediction models and their comparison. In order to accurately predict QoE for video streaming, data must be carefully gathered and taken into account at each stage of the procedure shown in Fig. 3.1. These factors have been emphasised, and experiments have shown that even when supervised, data-driven models are given identical data and features, their robustness and accuracy can differ greatly. Although various algorithms may be able to offer accuracy that is comparable, other factors like training and execution times will affect the adoption of models in real-time QoE prediction.

6.1 Summary

In this chapter, we concluded our work for QoE prediction of HAS. We discussed each model's pros and cons for QoE prediction of HAS. We also discussed the limitations and future works related to our research.

References

[1] ITUTP Recommendation. 1203.3,"parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport-quality integration module,". *International Telecommunication Union*, 2017.

[2] Alcardo Alex Barakabitze, Nabajeet Barman, Arslan Ahmad, Saman Zadtootaghaj, Lingfen Sun, Maria G Martini, and Luigi Atzori. Qoe management of multimedia streaming services in future networks: a tutorial and survey. *IEEE Communications Surveys & Tutorials*, 22(1):526–565, 2019.

[3] Cisco. Cisco Visual Networking Index: Forecast and Methodology, 2018-2023 - White paper, 2017. URL `https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf`.

[4] Arslan Ahmad, Alessandro Floris, and Luigi Atzori. QoE-centric service delivery: A collaborative approach among OTTs and ISPs. *Computer Networks*, 110:168–179, 2016.

[5] Definition of quality of experience (qoe). `https://www.itu.int/md/T01-SG12-040324-D-0197/en`, 2004.

[6] Patrick Le Callet, Sebastian Möller, Andrew Perkis, et al. Qualinet White Paper on Definitions of Quality of Experience (2012). In *European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003)*, Lausanne, Switzerland, Version 1.2, March 2013.

[7] Michael Seufert, Sebastian Egger, Martin Slanina, Thomas Zinner, Tobias Hoßfeld, and Phuoc Tran-Gia. A survey on quality of experience of http adaptive stream-

ing. *IEEE Communications Surveys Tutorials*, 17(1):469–492, 2015. doi: 10.1109/COMST.2014.2360940.

[8] Sarah Wassermann, Michael Seufert, Pedro Casas, Li Gang, and Kuang Li. Vicrypt to the rescue: Real-time, machine-learning-driven video-qoe monitoring for encrypted streaming traffic. *IEEE Transactions on Network and Service Management*, 17(4):2007–2023, 2020.

[9] Michael Seufert, Pedro Casas, Nikolas Wehner, Li Gang, and Kuang Li. Stream-based machine learning for real-time qoe analysis of encrypted video streaming traffic. In *2019 22nd Conference on innovation in clouds, internet and networks and workshops (ICIN)*, pages 76–81. IEEE, 2019.

[10] Arslan Ahmad, Atif Bin Mansoor, Alcardo Alex Barakabitze, Andrew Hines, Luigi Atzori, and Ray Walshe. Supervised-learning-based qoe prediction of video streaming in future networks: A tutorial with comparative study. *IEEE Communications Magazine*, 59(11):88–94, 2021.

[11] Maria Torres Vega, Cristian Perra, Filip De Turck, and Antonio Liotta. A review of predictive quality of experience management in video streaming services. *IEEE Transactions on Broadcasting*, 64(2):432–445, 2018.

[12] Özge Celenk, Thomas Bauschert, and Marcus Eckert. Machine learning based kpi monitoring of video streaming traffic for qoe estimation. *ACM SIGMETRICS Performance Evaluation Review*, 48(4):33–36, 2021.

[13] Shyamprasad Chikkerur, Vijay Sundaram, Martin Reisslein, and Lina J Karam. Objective video quality assessment methods: A classification, review, and performance comparison. *IEEE transactions on broadcasting*, 57(2):165–182, 2011.

[14] Maria Torres Vega, Vittorio Sguazzo, Decebal Constantin Mocanu, and Antonio Liotta. An experimental survey of no-reference video quality assessment methods. *International Journal of Pervasive Computing and Communications*, 2016.

[15] Nicolas Staelens, Glenn Van Wallendael, Karel Crombecq, Nick Vercammen, Jan De Cock, Brecht Vermeulen, Rik Van de Walle, Tom Dhaene, and Piet Demeester. No-reference bitstream-based visual quality impairment detection for high definition

h.264/avc encoded video sequences. *IEEE Transactions on Broadcasting*, 58(2): 187–199, 2012. doi: 10.1109/TBC.2012.2189334.

[16] Manish Narwaria and Weisi Lin. Svd-based quality metric for image and video using machine learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2):347–364, 2012. doi: 10.1109/TSMCB.2011.2163391.

[17] Asiya Khan, Lingfen Sun, and Emmanuel Ifeachor. Qoe prediction model and its application in video quality adaptation over umts networks. *IEEE Transactions on Multimedia*, 14(2):431–442, 2012. doi: 10.1109/TMM.2011.2176324.

[18] Baris Konuk, Emin Zerman, Gokce Nur, and Gozde Bozdagi Akar. A spatiotemporal no-reference video quality assessment model. In *2013 IEEE International Conference on Image Processing*, pages 54–58, 2013. doi: 10.1109/ICIP.2013.6738012.

[19] Nicolas Staelens, Dirk Deschrijver, Ekaterina Vladislavleva, Brecht Vermeulen, Tom Dhaene, and Piet Demeester. Constructing a no-reference h.264/avc bitstream-based video quality metric using genetic programming-based symbolic regression. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(8):1322–1333, 2013. doi: 10.1109/TCSVT.2013.2243052.

[20] Kongfeng Zhu, Chengqing Li, Vijayan Asari, and Dietmar Saupe. No-reference video quality assessment based on artifact measurement and statistical analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(4):533–546, 2015. doi: 10.1109/TCSVT.2014.2363737.

[21] Jacob Søgaard, Søren Forchhammer, and Jari Korhonen. No-reference video quality assessment using codec analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(10):1637–1650, 2015. doi: 10.1109/TCSVT.2015.2397207.

[22] Muhammad Shahid, Joanna Panasiuk, Glenn Van Wallendael, Marcus Barkowsky, and Benny Lövström. Predicting full-reference video quality measures using hevc bitstream-based no-reference features. In *2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX)*, pages 1–2, 2015. doi: 10.1109/QoMEX.2015.7148118.

[23] K. Pandremmenou, M. Shahid, L. P. Kondi, and B. Lövström. A no-reference bitstream-based perceptual model for video quality estimation of videos affected by

coding artifacts and packet losses. In Bernice E. Rogowitz, Thrasyvoulos N. Pappas, and Huib de Ridder, editors, *Human Vision and Electronic Imaging XX*, volume 9394, pages 486 – 497. International Society for Optics and Photonics, SPIE, 2015. doi: 10.1117/12.2077709. URL https://doi.org/10.1117/12.2077709.

[24] Xin Huang, Jacob Søgaard, and Søren Forchhammer. No-reference pixel based video quality assessment for hevc decoded video. *Journal of Visual Communication and Image Representation*, 43:173–184, 2017. ISSN 1047-3203. doi: https://doi.org/10.1016/j.jvcir.2017.01.002. URL https://www.sciencedirect.com/science/article/pii/S1047320317300020.

[25] Maria Torres Vega, Decebal Constantin Mocanu, Stavros Stavrou, and Antonio Liotta. Predictive no-reference assessment of video quality. *Signal Processing: Image Communication*, 52:20–32, 2017. ISSN 0923-5965. doi: https://doi.org/10.1016/j.image.2016.12.001. URL https://www.sciencedirect.com/science/article/pii/S092359651630176X.

[26] Raffael Shalala, Ran Dubin, Ofer Hadar, and Amit Dvir. Video qoe prediction based on user profile. In *2018 International Conference on Computing, Networking and Communications (ICNC)*, pages 588–592. IEEE, 2018.

[27] Tho Nguyen Duc, Chanh Minh Tran, Phan Xuan Tan, and Eiji Kamioka. Bidirectional lstm for continuously predicting qoe in http adaptive streaming. In *Proceedings of the 2019 2nd International Conference on Information Science and Systems*, pages 156–160, 2019.

[28] Lu Liu, Han Hu, Yong Luo, and Yonggang Wen. When wireless video streaming meets ai: a deep learning approach. *IEEE Wireless Communications*, 27(2):127–133, 2019.

[29] Liyan Qian, Huifang Chen, and Lei Xie. Svm-based qoe estimation model for video streaming service over wireless networks. In *2015 International Conference on Wireless Communications & Signal Processing (WCSP)*, pages 1–6. IEEE, 2015.

[30] Wei Zhou, Xiongkuo Min, Hong Li, and Qiuping Jiang. A brief survey on adaptive video streaming quality assessment. *arXiv preprint arXiv:2202.12987*, 2022.

[31] Miran Taha, Alejandro Canovas, Jaime Lloret, and Aree Ali. A qoe adaptive management system for high definition video streaming over wireless networks. *Telecommunication Systems*, 77(1):63–81, 2021.

[32] Yaqian Kang, Huifang Chen, and Lei Xie. An artificial-neural-network-based qoe estimation model for video streaming over wireless networks. In *2013 IEEE/CIC International Conference on Communications in China (ICCC)*, pages 264–269. IEEE, 2013.

[33] Emad Danish, Mohammed Alreshoodi, Anil Fernando, Bander Alzahrani, and Sami Alharthi. Cross-layer qoe prediction for mobile video based on random neural networks. In *2016 IEEE International Conference on Consumer Electronics (ICCE)*, pages 227–228. IEEE, 2016.

[34] Yosr Ben Youssef, Mariem Afif, Riadh Ksantini, and Sami Tabbane. A novel online qoe prediction model based on multiclass incremental support vector machine. In *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, pages 334–341. IEEE, 2018.

[35] Dimitar Minovski, Christer Åhlund, Karan Mitra, and Per Johansson. Analysis and estimation of video qoe in wireless cellular networks using machine learning. In *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6. IEEE, 2019.

[36] Xiaoming Tao, Yiping Duan, Mai Xu, Zhishen Meng, and Jianhua Lu. Learning qoe of mobile video transmission with deep neural network: A data-driven approach. *IEEE Journal on Selected Areas in Communications*, 37(6):1337–1348, 2019. doi: 10.1109/JSAC.2019.2904359.

[37] Huaizheng Zhang, Linsen Dong, Guanyu Gao, Han Hu, Yonggang Wen, and Kyle Guan. Deepqoe: A multimodal learning framework for video quality of experience (qoe) prediction. *IEEE Transactions on Multimedia*, 22(12):3210–3223, 2020.

[38] Fatima Laiche, Asma Ben Letaifa, Imene Elloumi, and Taoufik Aguili. When machine learning algorithms meet user engagement parameters to predict video qoe. *Wireless Personal Communications*, 116(3):2723–2741, 2021.

[39] Yosr Ben Youssef, Mariem Afif, Riadh Ksantini, and Sami Tabbane. A novel qoe model based on boosting support vector regression. In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE, 2018.

[40] Kalpana Seshadrinathan, Rajiv Soundararajan, Alan Conrad Bovik, and Lawrence K Cormack. Study of subjective and objective quality assessment of video. *IEEE transactions on Image Processing*, 19(6):1427–1441, 2010.

[41] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67 (2):301–320, 2005.

[42] Margaret H Pinson and Stephen Wolf. A new standardized method for objectively measuring video quality. *IEEE Transactions on broadcasting*, 50(3):312–322, 2004.

[43] Zhengfang Duanmu, Abdul Rehman, and Zhou Wang. A quality-of-experience database for adaptive video streaming. *IEEE Transactions on Broadcasting*, 64(2): 474–487, 2018. doi: 10.1109/TBC.2018.2822870.

[44] Python scikit-learn ensemble methods,", . URL `https://scikit-learn.org/stable/modules/ensemble.html`.

[45] T Jayalakshmi and A Santhakumaran. Statistical normalization and back propagation for classification. *International Journal of Computer Theory and Engineering*, 3(1):1793–8201, 2011.

[46] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[47] Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

[48] Python scikit-learn feature selection library,", . URL `https://scikit-learn.org/stable/modules/feature_selection.html`.

[49] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[50] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[51] Python scikit-learn library,", . URL https://scikit-learn.org/stable/modules/grid_search.html.

[52] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[53] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

[54] Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.

[55] Geoffrey E Hinton. Connectionist learning procedures. In *Machine learning*, pages 555–610. Elsevier, 1990.

[56] Kun An and Jiang Meng. Voting-averaged combination method for regressor ensemble. In *International Conference on Intelligent Computing*, pages 540–546. Springer, 2010.

[57] How to develop a weighted average ensemble with python. URL https://machinelearningmastery.com/weighted-average-ensemble-with-python/#:~%7B%7D:text=Weighted%20average%20or%20weighted%20sum%20ensemble%20is%20an%20ensemble%20machine,related%20to%20the%20voting%20ensemble.

[58] Leo Breiman. Stacked regressions. *Machine learning*, 24(1):49–64, 1996.

[59] Stackingcvregressor: stacking with cross-validation for regression. URL http://www.rasbt.github.io/mlxtend/user_guide/regressor/StackingCVRegressor/.

[60] Yiheng Li and Weidong Chen. A comparative performance assessment of ensemble learning for credit scoring. *Mathematics*, 8(10):1756, 2020.

[61] Scott Menard. Coefficients of determination for multiple logistic regression analysis. *The American Statistician*, 54(1):17–24, 2000.

[62] Cort J Willmott and Kenji Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, 30(1):79–82, 2005.

[63] Philip Sedgwick. Pearson's correlation coefficient. *Bmj*, 345, 2012.

[64] Philip Sedgwick. Spearman's rank correlation coefficient. *Bmj*, 349, 2014.

[65] Pandas python library. URL `https://pandas.pydata.org/`.

[66] ITU-T Recommendation. 1203.1,"parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport – video quality estimation module,". *International Telecommunication Union*, 2019.

Appendix A

Appendix

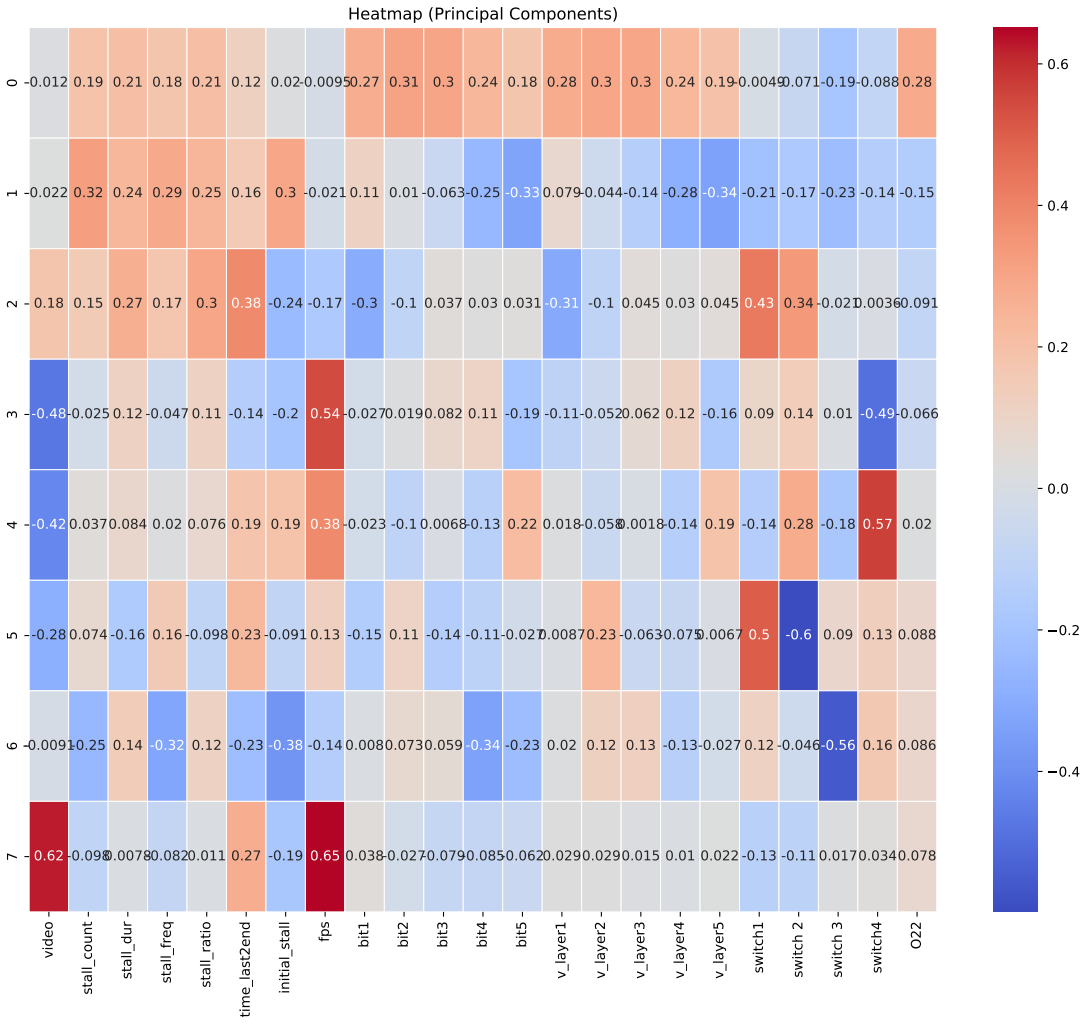


Figure A.1: Correlation between principal components and all QoE features

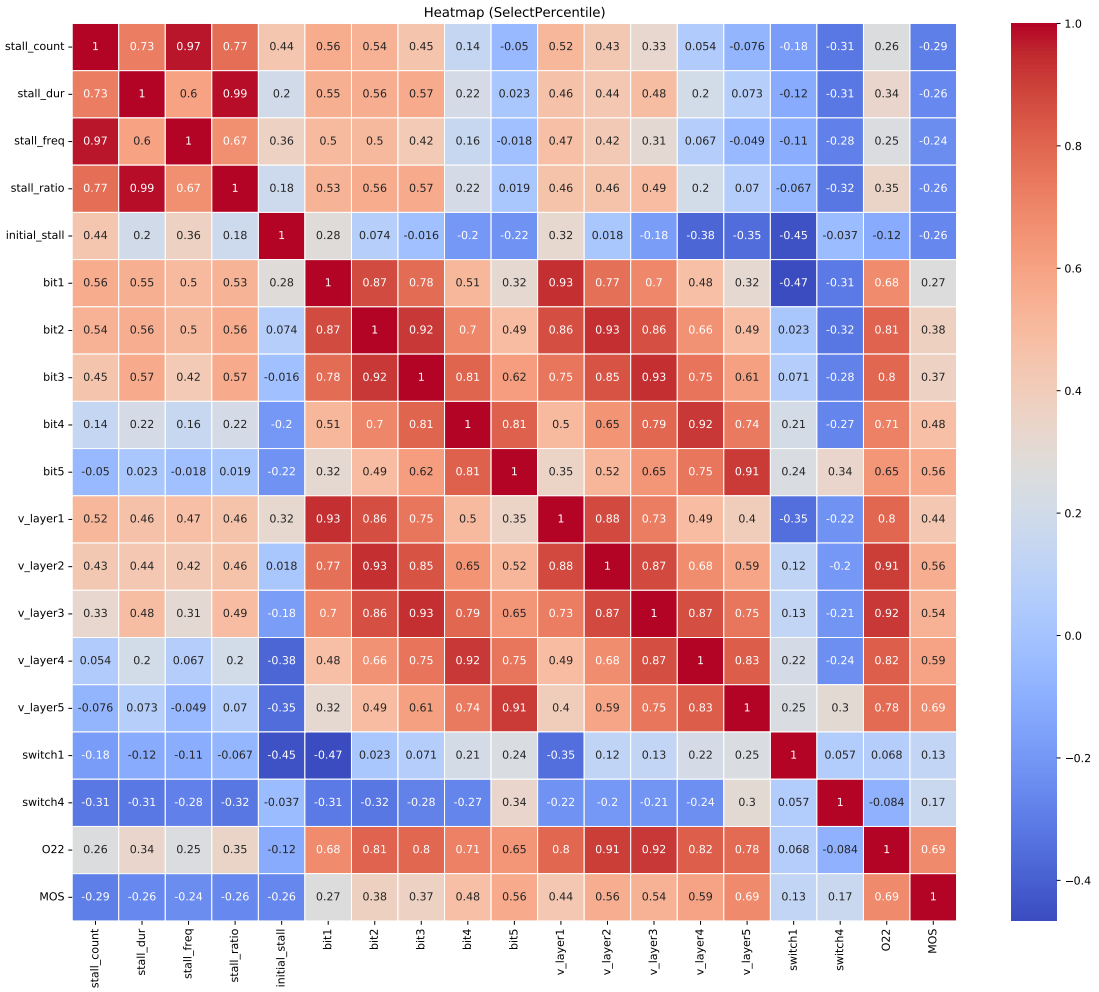


Figure A.2: QoE features of HAS with correlation existing between various features and MOS (Univariate Feature Selection)

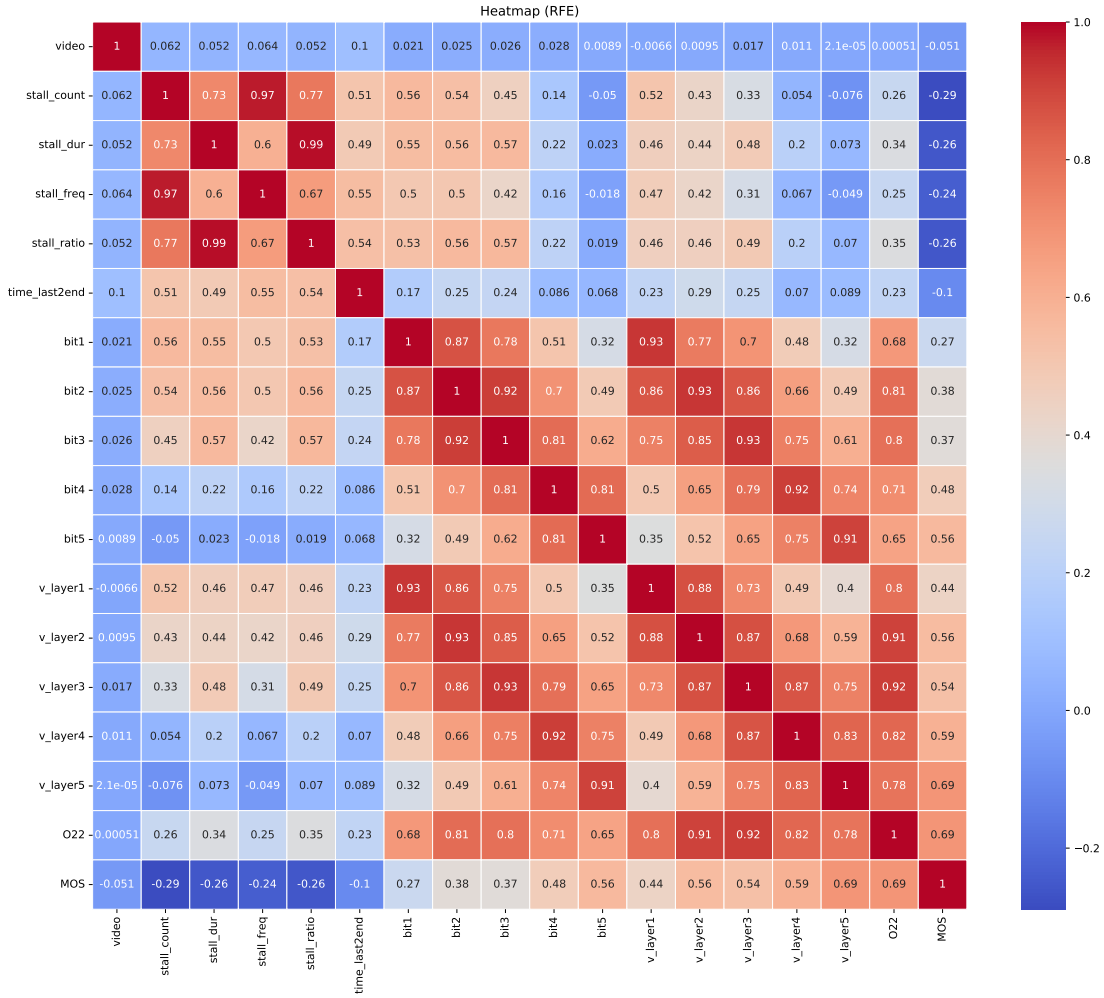


Figure A.3: QoE features of HAS with correlation existing between various features and MOS (Recursive Feature Elimination)

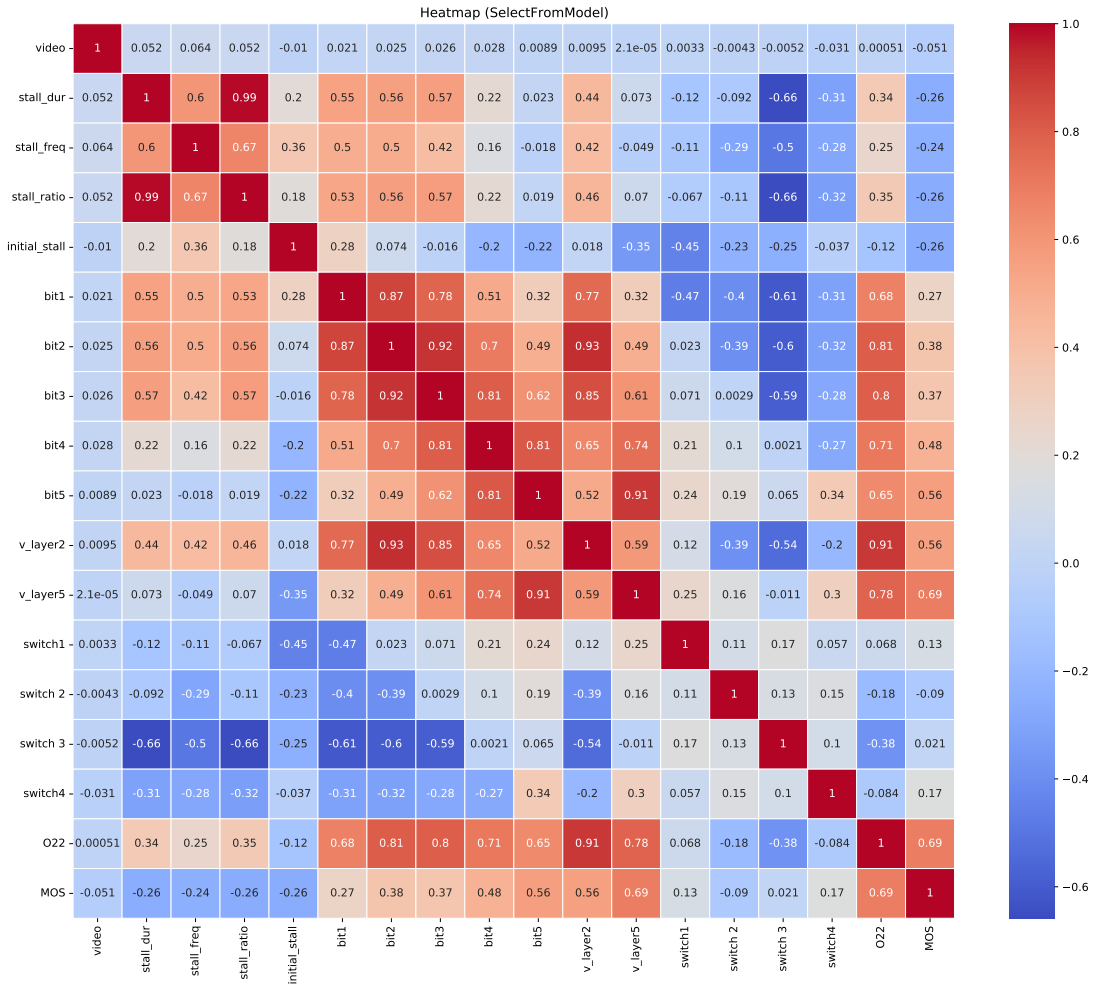


Figure A.4: QoE features of HAS with correlation existing between various features and MOS (Select From Model)

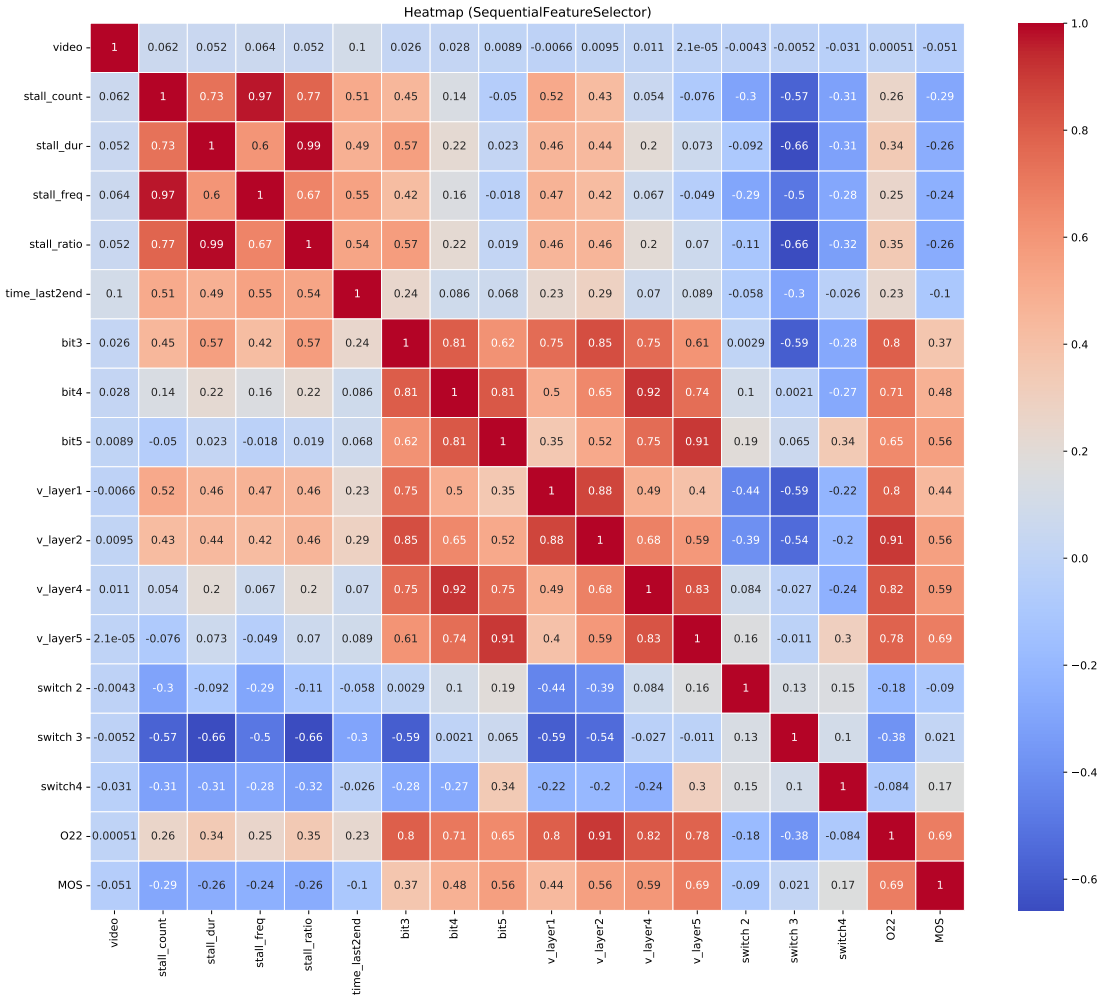
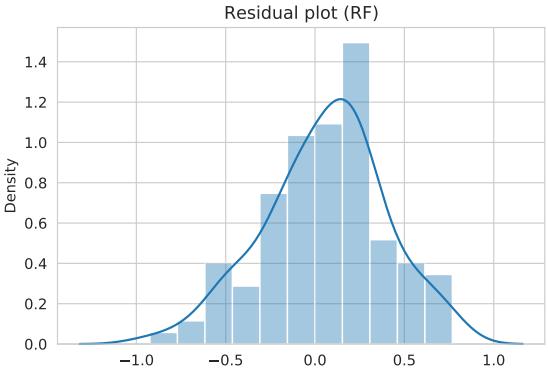
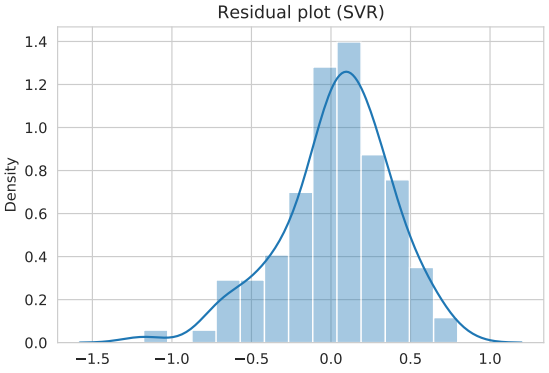


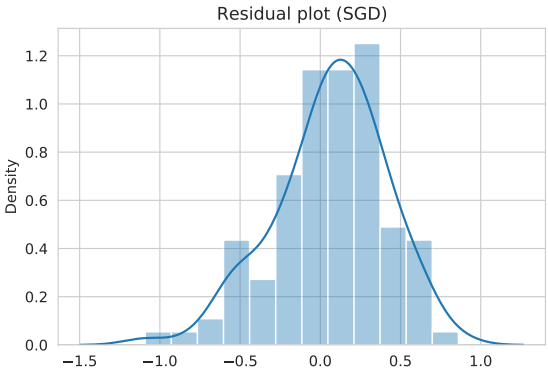
Figure A.5: Heat map showing correlation between QoE features and MOS (Select From Model)



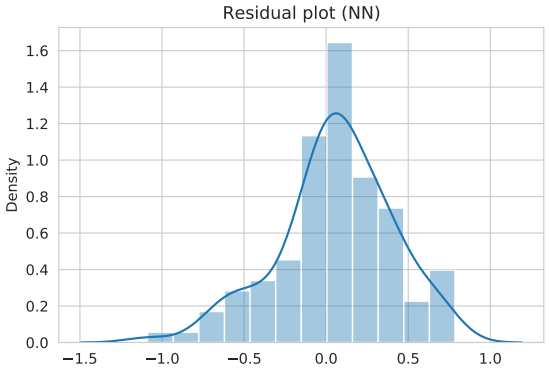
(a) Residual Plot RF



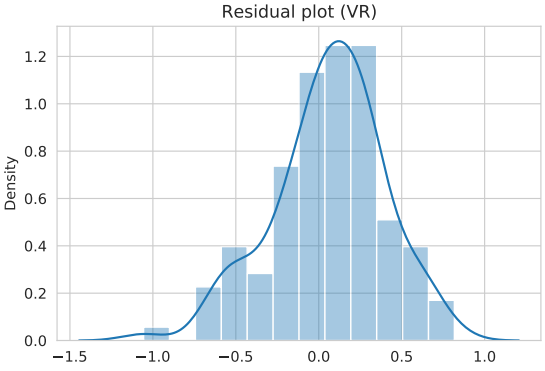
(b) Residual Plot SVR



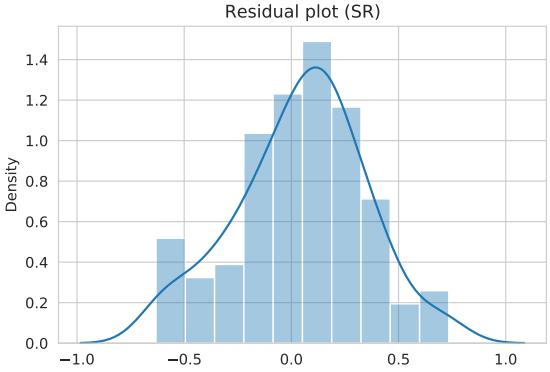
(c) Residual Plot SGD



(d) Residual Plot NN

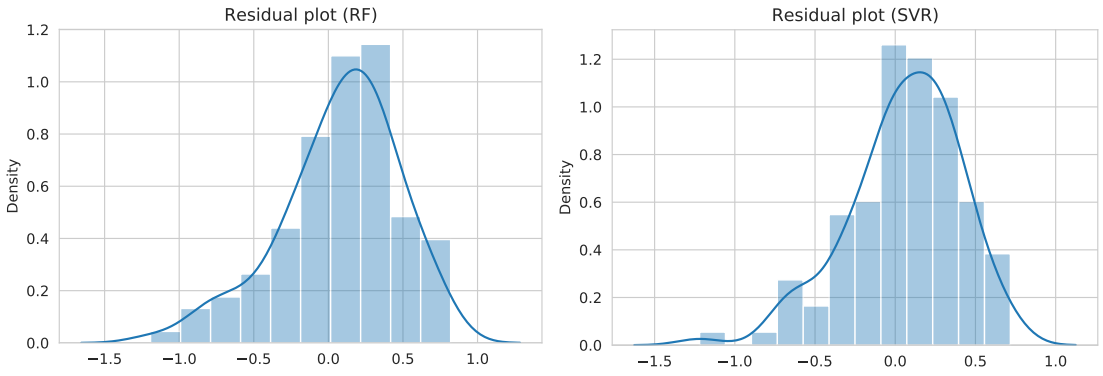


(e) Residual Plot VR



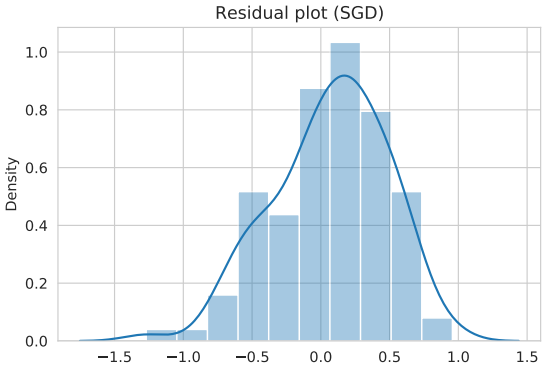
(f) Residual Plot SR

Figure A.6: Residual plots of Supervised-Learning models applied on all QoE features

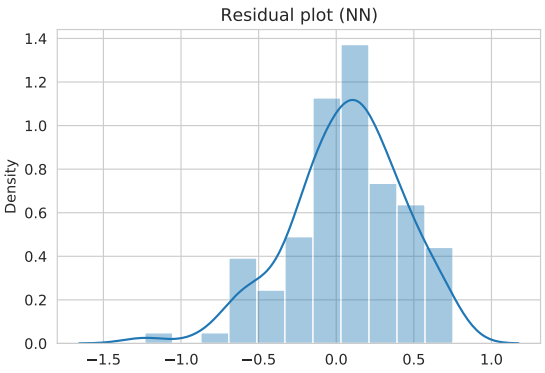


(a) Residual Plot RF

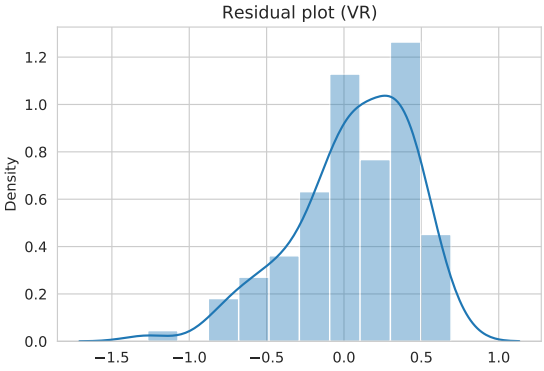
(b) Residual Plot SVR



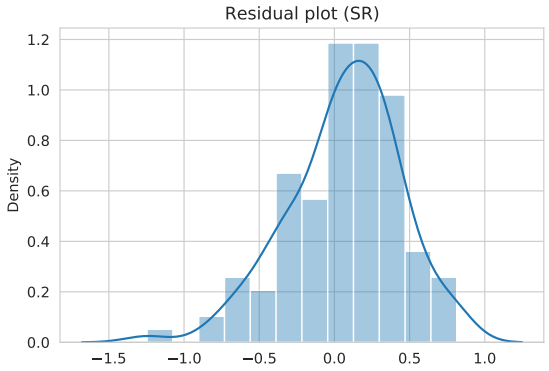
(c) Residual Plot SGD



(d) Residual Plot NN

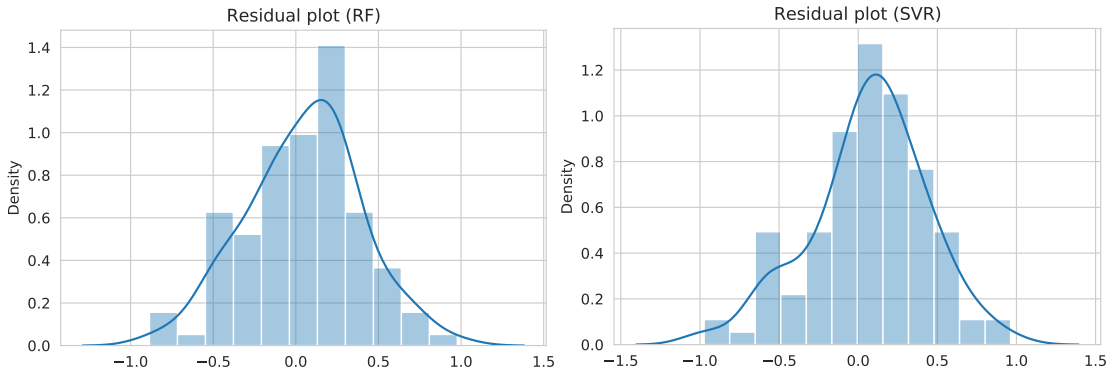


(e) Residual Plot VR



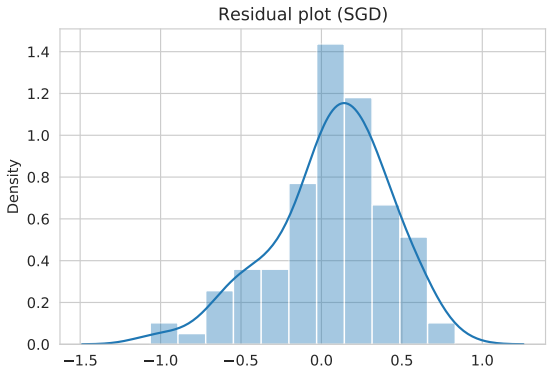
(f) Residual Plot SR

Figure A.7: Residual plots of Supervised-Learning models applied on QoE Principal Components

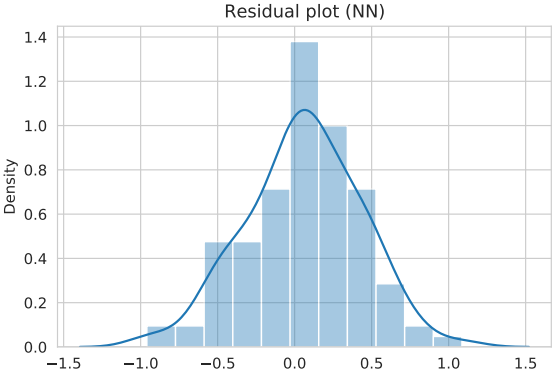


(a) Residual Plot RF

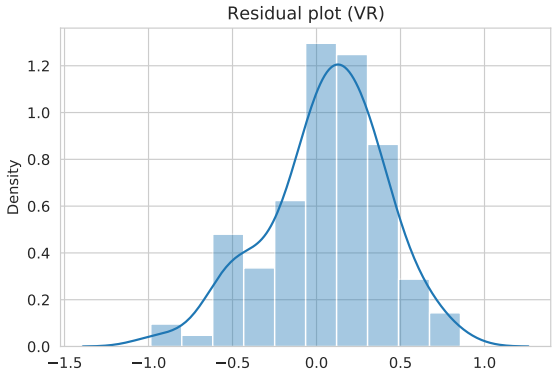
(b) Residual Plot SVR



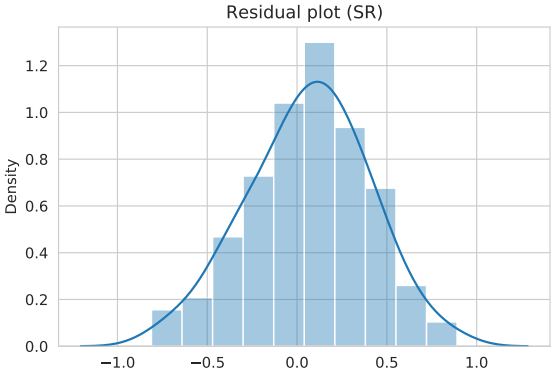
(c) Residual Plot SGD



(d) Residual Plot NN

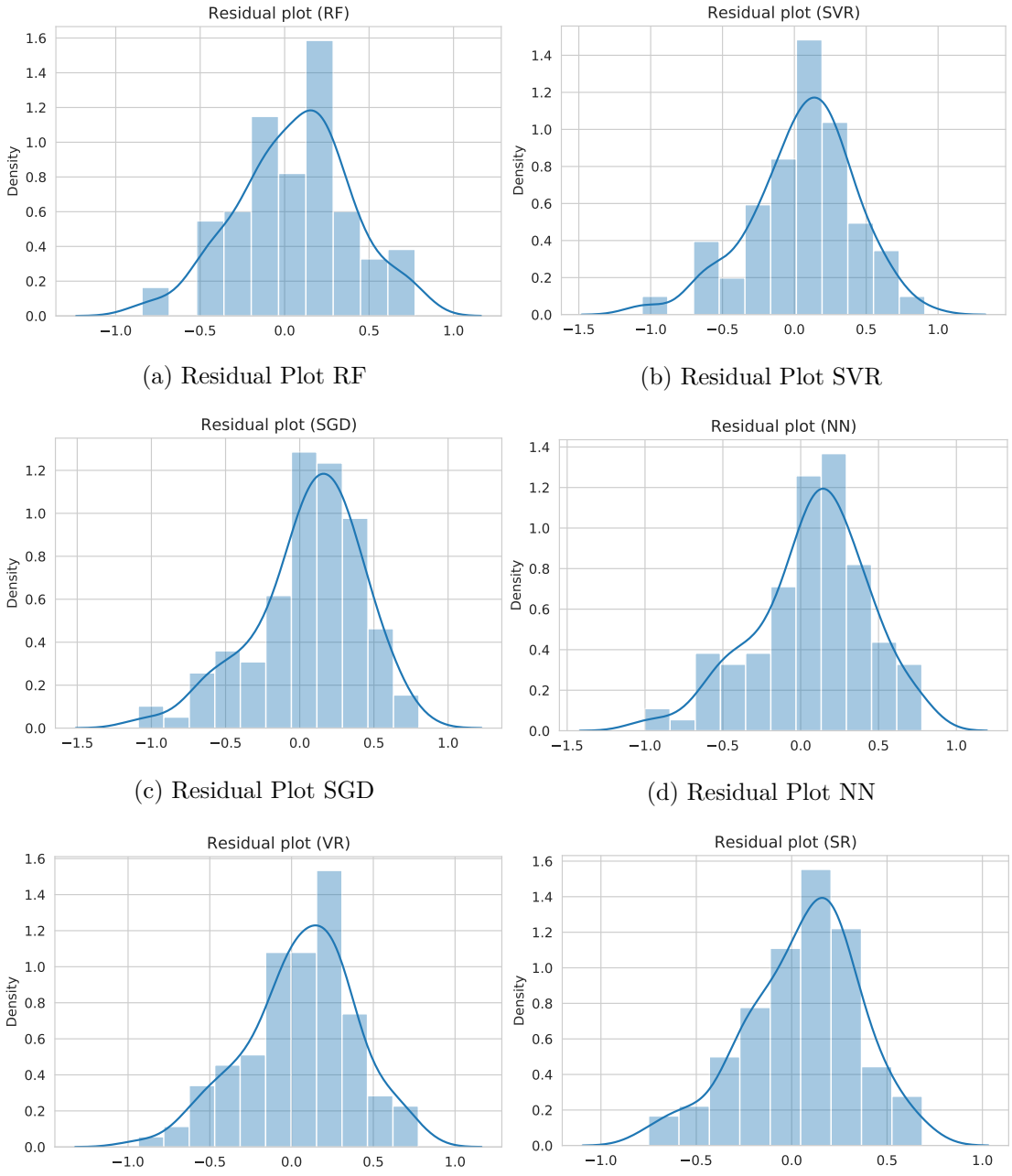


(e) Residual Plot VR



(f) Residual Plot SR

Figure A.8: Residual plots of Supervised-Learning models applied on QoE features (Univariate Feature Selection)



(a) Residual Plot RF

(b) Residual Plot SVR

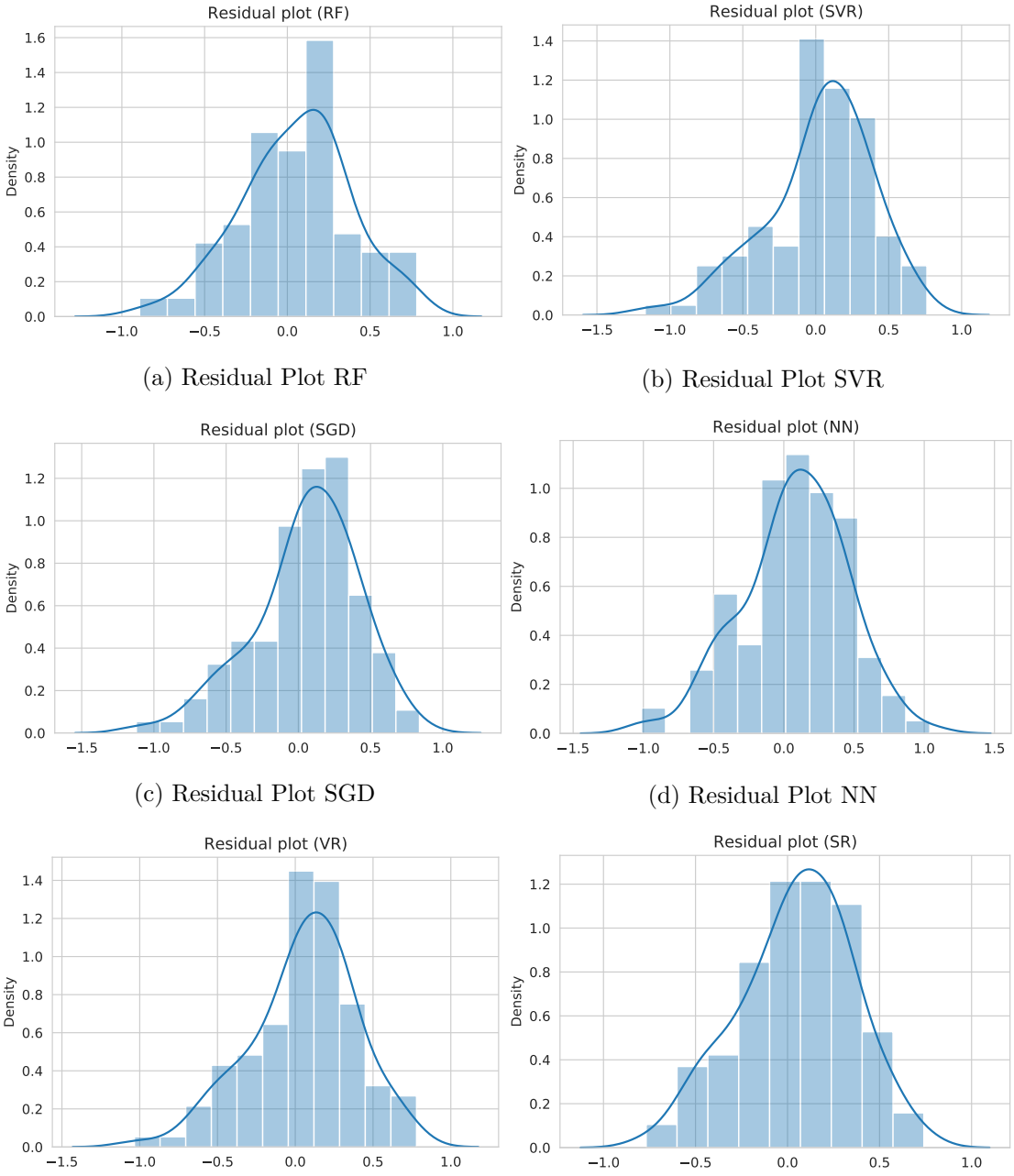
(c) Residual Plot SGD

(d) Residual Plot NN

(e) Residual Plot VR

(f) Residual Plot SR

Figure A.9: Residual plots of Supervised-Learning models applied on QoE features (Recursive Feature Elimination)



(a) Residual Plot RF

(b) Residual Plot SVR

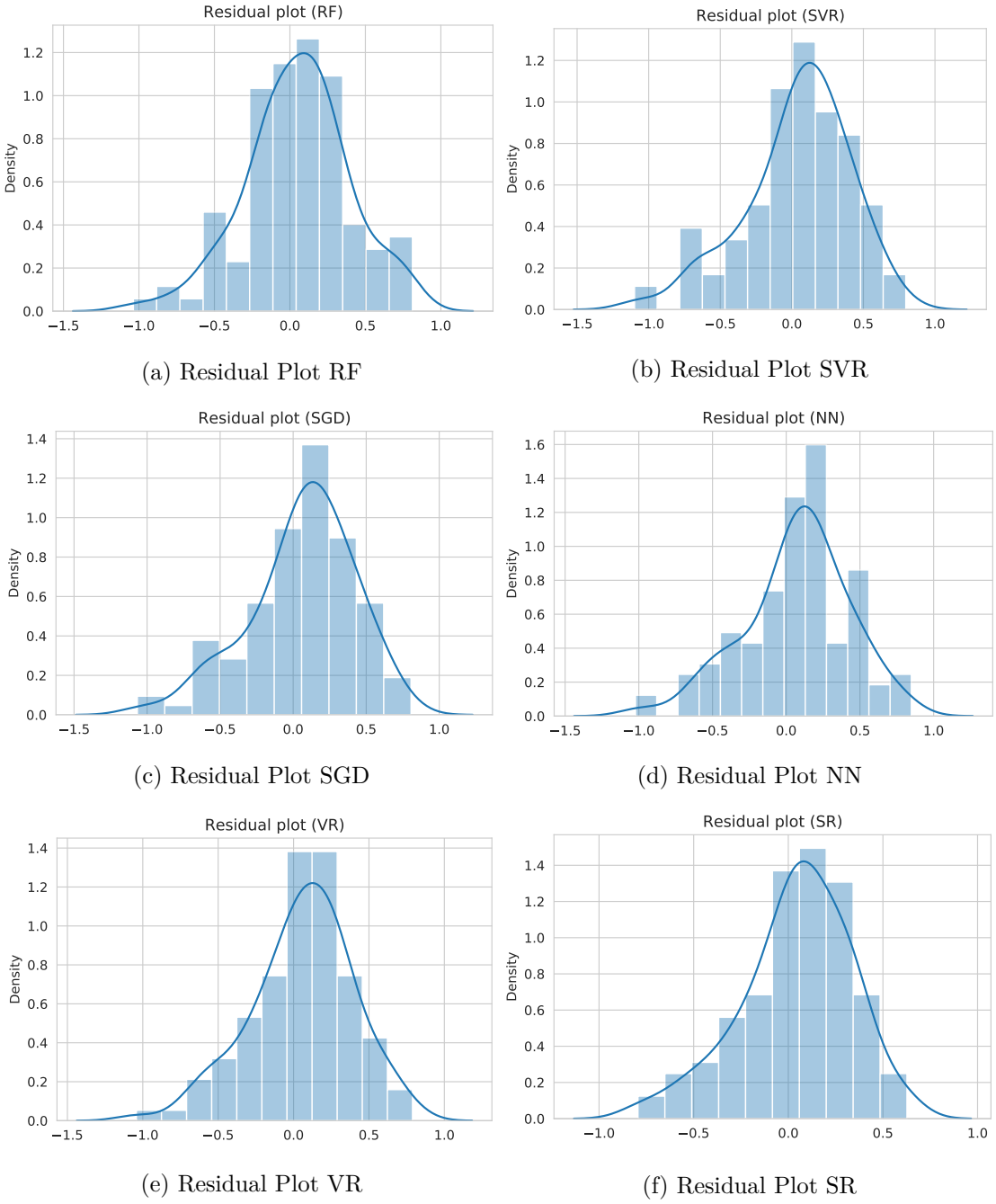
(c) Residual Plot SGD

(d) Residual Plot NN

(e) Residual Plot VR

(f) Residual Plot SR

Figure A.10: Residual plots of Supervised-Learning models applied on QoE features (Select From Model)



(a) Residual Plot RF

(b) Residual Plot SVR

(c) Residual Plot SGD

(d) Residual Plot NN

(e) Residual Plot VR

(f) Residual Plot SR

Figure A.11: Residual plots of Supervised-Learning models applied on QoE features (Sequential Feature Selection)