International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

P.565

(11/2021)

SERIES P: TELEPHONE TRANSMISSION QUALITY, TELEPHONE INSTALLATIONS, LOCAL LINE NETWORKS

Objective measuring apparatus

Framework for creation and performance testing of machine learning based models for the assessment of transmission network impact on speech quality for mobile packet-switched voice services

Recommendation ITU-T P.565

ITU-T P-SERIES RECOMMENDATIONS

TELEPHONE TRANSMISSION QUALITY, TELEPHONE INSTALLATIONS, LOCAL LINE NETWORKS

Vocabulary and effects of transmission parameters on customer opinion of transmission quality	P.10–P.19
Voice terminal characteristics	P.30–P.39
Reference systems	P.40–P.49
Objective measuring apparatus	P.50–P.59
Objective electro-acoustical measurements	P.60–P.69
Measurements related to speech loudness	P.70–P.79
Methods for objective and subjective assessment of speech quality	P.80–P.89
Voice terminal characteristics	P.300–P.399
Objective measuring apparatus	**P.500–P.599**
Measurements related to speech loudness	P.700–P.709
Methods for objective and subjective assessment of speech and video quality	P.800–P.899
Audiovisual quality in multimedia services	P.900–P.999
Transmission performance and QoS aspects of IP end-points	P.1000–P.1099
Communications involving vehicles	P.1100–P.1199
Models and tools for quality assessment of streamed media	P.1200–P.1299
Telemeeting assessment	P.1300–P.1399
Statistical analysis, evaluation and reporting guidelines of quality measurements	P.1400–P.1499
Methods for objective and subjective assessment of quality of services other than speech and video	P.1500–P.1599

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T P.565

Framework for creation and performance testing of machine learning based models for the assessment of transmission network impact on speech quality for mobile packet-switched voice services

Summary

Recommendation ITU-T P.565 provides the output of the framework which is a machine learning based speech quality prediction model that predicts the impact on speech quality from Internet protocol (IP) transport and underlying transport, as well as a standardized or pre-defined jitter buffer in the end client; thus, providing a network centric view on the speech quality service delivered on mobile packet switched networks. This is expressed in terms of a mean opinion score-listening quality objective (MOS-LQO) under the assumption of an otherwise clean transmission, without background noise, non-standard-conformant encoding on sending device, automatic gain control, voice enhancement devices, transcoding, bridging, frequency response, non-standard-conformant jitter-buffer (for IP multimedia systems (IMS) mobile calls) or decoding, clock drift or any other impairment not caused by the IP transport and underlying transport. The models according to this framework can use information on the temporal structure of the reference signal to identify the importance of individual sections of the bitstream with regard to speech quality. These models do not perform any perceptual analysis of the recorded speech signal.

The framework specifies three modules required for the development of these kinds of metrics: the databases generator module, the machine learning module, and the validation module for the trained model. In addition, the database content and the features used by the machine learning algorithm are described. The framework also provides a large set of test vectors, in the form of error (jitter and packet loss) patterns files for learning and validation. This Recommendation specifies the minimum required performance, as well as conditions and requirements for an independent additional validation for models developed based on the framework. This Recommendation also specifies implementation requirements.

The models developed based on the framework enable the assessment of transmission network impact on speech quality for mobile packet-switched voice services, and therefore benefit operators and regulators alike with a fast and easy speech quality trend monitoring/benchmarking and troubleshooting. In addition, if predictors according to this framework are used together with perceptual speech quality metrics such as ITU-T P.863, it is possible to identify if the source of problems resides inside or outside the transport network observed by the predictor according to this framework. Consequently, a more detailed analysis of the situation can be achieved and troubleshooting of less obvious degradations such as the ones occurring outside of the transport network (e.g., emerged from automatic gain control, voice enhancement devices, transcoding or analogue processing) is enabled.

This Recommendation includes electronic attachments containing detailed descriptions of generic jitter files and a reference speech sample (see Annex D).

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T P.565	2020-01-13	12	11.1002/1000/14152
2.0	ITU-T P.565	2021-11-29	12	11.1002/1000/14827

Keywords

AMR/AMR-WB, EVS, generic jitter files, HD voice, intrusive parametric, machine learning, ML, OTT, QoE, VoLTE.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

© ITU 2022

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Table of Contents

		Page
1	Scope	1
2	References	1
3	Definitions	2
4	Abbreviations and acronyms	2
5	Conventions	3
6	Applications for models developed based on the framework	4
7	High level overview of the framework	5
7.1	Framework architecture	5
7.2	Generic jitter files	6
7.3	Reference speech file	7
8	Learning and validation database generator for IMS mobile EVS use case	7
8.1	Simulate network block	7
8.2	EVS coding and decoding blocks: EVS codec and codec parameters	7
8.3	MOS grading block	7
8.4	EVS process jitter file block: processing of the jitter file	8
8.5	Learning and validation databases	8
9	Machine learning module for IMS mobile EVS use case	8
9.1	ML algorithm	9
9.2	ML features	9
10	Statistical evaluation module	10
11	Framework's inputs and outputs	11
12	Aspects related to the run-time of models developed based on the framework	12
12.1	Operation mode	12
12.2	Reference speech samples	12
12.3	Pre-processing at run time	12
12.4	The measurement procedure	12
13	Requirements for models developed based on the framework	13
13.1	Mandatory conditions and procedures	13
13.2	Minimum performance requirements	13
Annex A – Example method for ML overfitting/underfitting test		15
Annex B – Check list of requirements for a model developed based on the framework		18
Annex C – Conditions and requirements of an additional independent validation of a model developed based on the framework		20
C.1	Conditions and requirements of an independent validation	20
C.2	Validation procedure	21
Annex D – Electronic attachments		24
D.1	Reference speech samples (FB and SWB)	24

Page

D.2 Generic jitter files data bases 24

D.3 Data bases description 24

Appendix I – Procedure for feature extraction based on machine learning 25

I.1 Create statistical features 25

I.2 Create jitter buffer-based features 26

I.3 Codec based features (rate and channel aware) 28

I.4 Create reference speech-based features 28

Appendix II – Descriptions of generic jitter files creation 31

II.1 Generic jitter files for model development and final validation (source Infovista) 31

II.2 Unknown independent validation live data set description (source Rohde&Schwarz) 34

Appendix III – Justification of the minimum requirements based on performance results' analysis 36

III.1 IMS mobile EVS use case 36

III.2 Results on learning and validation data sets 36

III.3 Results on unknown validation live data 39

III.4 ML overfitting/underfitting test 40

Bibliography 42

Electronic attachments: Generic jitter files and a reference speech sample.

Recommendation ITU-T P.565

Framework for creation and performance testing of machine learning based models for the assessment of transmission network impact on speech quality for mobile packet-switched voice services

1 Scope

This Recommendation[1] specifies a framework in the form of constraints, performance criteria and methods for the development of intrusive parametric, machine learning (ML) based models for the assessment of transmission network impact on speech quality for mobile packet-switched voice services.

Models developed according to this Recommendation estimate the speech quality based on IP-bit-stream and a temporal distribution of speech energy as expected for the speech sample, where the quality prediction is applied. The models use the adaptiveness of the jitter buffer in the end client as well as Internet protocol (IP) transport and underlying transport behaviour of typical voice services such as high definition voice over Internet protocol (HD VoIP) and IP multimedia systems (IMS) mobile calls such as voice over LTE (VoLTE), voice over new radio (VoNR) using narrow band (NB), wide band (WB), super wideband (SWB) and full band (FB) voice, and over the top (OTT) (e.g., WhatsApp, Skype, Viber, WeChat, among others).

This Recommendation specifies techniques using machine learning to predict speech quality based on what it has learnt in the controlled and verified environment of the framework. Continuous learning based on real time adaptation of the ML algorithm's coefficients is not used. In addition, the Recommendation explains how the framework should be used and what are the requirements to be met in order for a ML based predictor to conform to this Recommendation. Test datasets are provided and required to be used in order to prove that models developed based on the framework meet the minimum required performance as defined by the framework. This Recommendation also specifies conditions and requirements for an independent additional validation of models developed based on the framework.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T G.1050] Recommendation ITU-T G.1050 (2016), *Network model for evaluating multimedia transmission performance over Internet Protocol.*

[ITU-T J.247] Recommendation ITU-T J.247 (2008), *Objective perceptual multimedia video quality measurement in the presence of a full reference.*

[ITU-T J.343] Recommendations ITU-T J.343 (2014), *Hybrid perceptual bitstream models for objective video quality measurements.*

[1] This Recommendation includes electronic attachments containing detailed descriptions of generic jitter files and a reference speech sample (see Annex F).

[ITU-T P.120x] Recommendations ITU-T P.1203 (2017) and ITU-T P.1203.1 to ITU-T P.1203.3 series, *Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport*.

[ITU-T P.564] Recommendation ITU-T P.564 (2007), *Conformance testing for voice over IP transmission quality assessment models.*

[ITU-T P.800] Recommendation ITU-T P.800 (1996), *Methods for subjective determination of transmission quality.*

[ITU-T P.800.1] Recommendation ITU-T P.800.1 (2016), *Mean opinion score (MOS) terminology.*

[ITU-T P.800.2] Recommendation ITU-T P.800.2 (2016), *Mean opinion score interpretation and reporting.*

[ITU-T P.863] Recommendation ITU-T P.863 (2018), *Perceptual objective listening quality prediction.*

[ITU-T P.863.1] Recommendation ITU-T P.863.1 (2019), *Application guide for Recommendation ITU-T P.863.*

[ITU-T P.1401] Recommendation ITU-T P.1401 (2020), *Methods, metrics and procedures for statistical evaluation, qualification and comparison of objective quality prediction models.*

[ETSI TR 102 506] ETSI TR 102 506, V1.4.1 (2011), *Speech and Multimedia Transmission Quality (STQ); Estimating Speech Quality per Call.*

[ETSI TS 126 448] ETSI TS 126 448, V15.0.0 (2018), *Universal Mobile Telecommunications System (UMTS); LTE; Codec for Enhanced Voice Services (EVS); Jitter Buffer Management.*

3 Definitions

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AMR	Adaptive Multi Rate
DCT	Discrete Cosine Transform
DRX	Discontinuous Reception
DTX	Discontinuous Transmission
EVS	Enhanced Voice Services
FB	Full Band
FE	Frame Erasure
FER	Frame Erasure Rate
HD	High Definition
IMS	IP Multimedia Systems
IP	Internet Protocol
LTE	Long Term Evolution

MAE	Management Application Entity
ML	Machine Learning
MOS-LQO	Mean Opinion Score-Listening Quality Objective
MSE	Mean Squared Error
MTSI	Multimedia Telephony Service for IP Multimedia Subsystem
NB	Narrow Band
OTT	Over The Top
PLC	Packet Loss Concealment
QoE	Quality of Experience
RLC	Radio Link Control
RMS	Root Mean Square
RTP	Real-time Transport Protocol
RMSE	Root Mean Square Error
SID	Silence Interruption Duration
SWB	Super Wideband
VBR	Variable Bit Rate
VoIP	Voice over Internet Protocol
VoLTE	Voice over LTE
VoNR	Voice over New Radio
WB	Wide Band

5 Conventions

Machine leaning

Within the context of this Recommendation, the term "machine learning" (ML) is used for a set of mathematical and statistical algorithms which use computational methods to "learn" information directly from data without relying on a predetermined equation as a model [b-Jiménez].

Machine learning features

Within the context of this Recommendation the term "machine learning features" (ML features) is used with its scientific meaning of inputs to the ML algorithm.

MOS scoring

[ITU-T P.800.2] stipulates that "detailed context description" of the source of the MOS values reported should be specified (cf. clause 14).

In order to limit complicated repetition throughout the present document and as such for the convenience of the reader of this Recommendation the following conventions are introduced:

(1) MOS(ml) denotes the predicted MOS-LQO value(s) generated by models compliant with the framework based on machine learning (ml).

(2) MOS-LQO (a.k.a MOS) denotes listening quality predicted by objective models, e.g., [ITU-T P.863].

NOTE 1 – The accuracy of MOS(ml) models is evaluated against MOS-LQO as predicted by [ITU-T P.863].

NOTE 2 – These conventions are void outside the context of this Recommendation – context in this case covers any related Recommendations specifying models which have proven compliance with the framework specified in this Recommendation, e.g., Recommendations ITU-T P.565.n {with n= 1, 2, 3, etc.}.

6 Applications for models developed based on the framework

The models developed according to this framework predict network centric speech quality without consideration of effects caused by background noise, automatic gain control, voice enhancement devices, frequency response, clock drift, transcoding, bridging or any other impairments not caused by the packet transport.

Models developed according to this Recommendation shall meet the following:

– Require active testing, meaning the transmission of a dedicated speech sample; therefore, the models are intrusive.

– Provide a network centric view and do not predict end to end speech quality.

– Focus on the network impact on the voice service's speech listening quality in case of transcoding-free transmission.

– Run-in-field-testing tools (e.g., drive test tools, device-based measurements) for the following applications:

  - Monitoring trends of network impact on speech quality.
  - Network troubleshooting.
  - Speech quality benchmarking of transport network impact on speech quality.
  - Network monitoring for regression testing.
  - Call quality based on aggregation of individual scores during the call as described in [ETSI TR 102 506].

Table 1 describes applications, conditions and test scenarios for which the framework is applicable.

Table 1 – Applications, conditions and test scenarios for models developed according to this framework

Category	Application
Service type	IMS mobile calls (e.g., VoLTE, VoNR), OTT VoIP-OTT VoIP calls
Codecs	EVS, AMRWB, Opus
Jitter Buffer management	EVS JBM ([ETSI TS 126 448]), PJSIP
Audio channel bandwidths	NB, WB, SWB, FB
Conditions	Packet based transmissions, encoding/decoding in the terminals.
Test scenarios	Network centric including the error concealment and EVS standardized jitter buffer behaviour or emulated OTT jitter buffer behaviour.
Test scope/application	Monitoring trends of transport network impact on speech quality, network benchmarking of transport network impact on speech quality. Network monitoring for regression testing. Apply an aggregation for call quality scoring as described in [ETSI TR 102 506].

Table 2 describes applications, conditions and test scenarios for which the framework is not applicable.

Table 2 – Applications, conditions and test scenarios not applicable for models developed according to this framework

Category	Application
Service type	CS, CSFB, 2G/3G-IMS mobile voice (e.g., VoLTE, VoNR) IRAT calls
Codecs	Proprietary OTT
Clients	Proprietary OTT
Conditions	Speech signal processing in the terminals such as: background noise, automatic gain control, voice enhancement devices, frequency response, clock drift, transcoding.
Test scenarios	End to end which include the device's speech path processing (e.g., frequency characteristic). Bridged networks. Proprietary error concealment and jitter buffer handling.
Test scope/application	Codec performance evaluation

7 High level overview of the framework

The framework is aimed at developing intrusive parametric speech quality predictors, which use both IP stream, codec, and client parameters as well as the reference speech sample(s). An intrusive parametric approach shall be considered due to the need for a sufficiently good enough accuracy against ground truth for speech quality.

[ITU-T P.863] is used as mean opinion score-listening quality objective (MOS-LQO) reference for the speech quality ground truth since the development of a machine learning based solution requires a large amount of data for learning and validation, something not feasible to achieve through subjective tests. An alternative for creating the ground truth is presented in [b-Mathwork].

The framework is based on machine learning algorithms which are used for developing an instance of the MOS-LQO predictor of the network impact. Therefore, its implementation in field testing tools is not designed to learn and adapt in real time, but only to predict based on what it has learnt in the controlled and verified environment of the framework.

The framework according to this Recommendation specifies the model realization using as an example, the IMS mobile telephony service with enhanced voice services (EVS) codec (use case 'IMS mobile EVS' for e.g., VoLTE-VoLTE or VoNR-VoNR calls as service type and EVS as codec).

7.1 Framework architecture

The framework is based on a three-module structure: learning and validation database generator, machine learning based model, and statistical validation module.

Figure 1 presents an example implementation of a three-module structure according to this framework. In this example, the learning and validation databases generator comprise of a simulator that uses the reference speech file, network error patterns (jitter and packet loss) called "generic jitter files", and codec/client specific information to create the learning and validation databases. The simulator uses codec/client specific information, and therefore it is use case specific. Similarly,

the machine learning module can use ML features as inputs which depend on the used codec/client and therefore it is use case specific. The statistical evaluation module is use case transparent since it evaluates the performance of the ML based predictor's model.

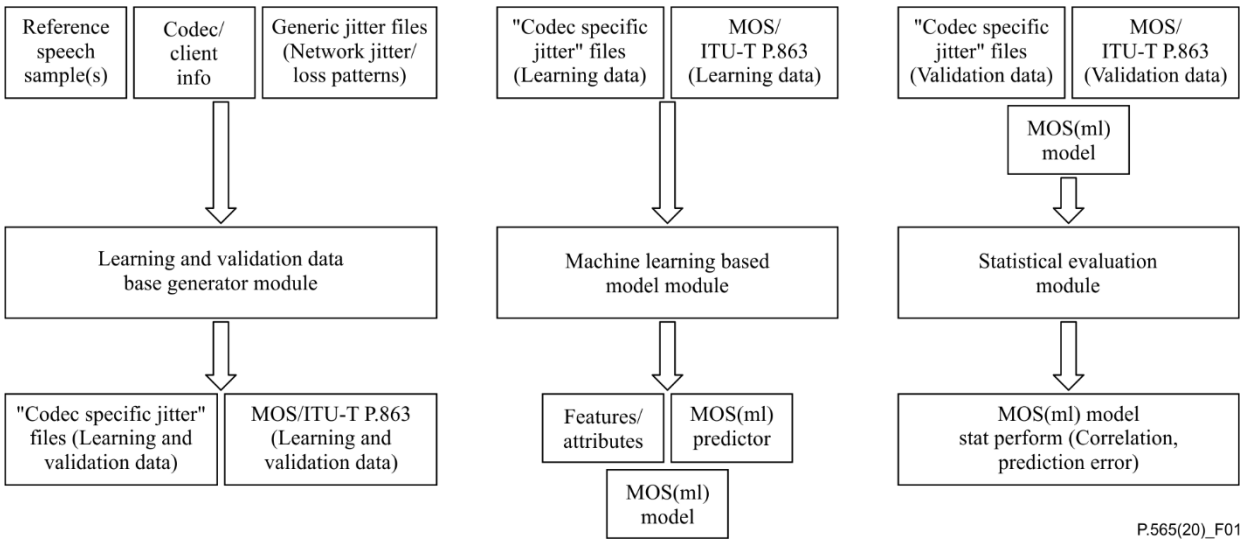


Figure 1 – Framework structure

7.2 Generic jitter files

The network error pattern files contain a combination of packet loss and jitter and are described by a "generic jitter file" in a column format suitable for machine learning input format. Therefore, the generic jitter file contains two columns with comma separated packet number and arrival time in milliseconds. The network error patterns (jitter and packet loss) are simulated based on the analysis and learning of error patterns from live networks in order to cover a large range and as real as possible network conditions.

Two mandatory generic jitter files sets are provided in Annex D of this Recommendation.

1) Generic jitter files for learning and validation (simulations) (GenericJitterFiles_TrainingAndValidation.7z) which cover a broad range of conditions as follows:
   - "Live (drive test) data modulated with simulations" to broaden the conditions' range (e.g., randomizing the live degradation position and amplitude).
   - "Gilbert burst packet loss and burst jitter up to 30%". This mimic error cases seen in live drive test data, for example during handover, where packets are buffered and suddenly released.
   - "Gilbert severe burst jitter to 70%". This database improves the learning and testing of large jitter which is the hardest case to predict. In addition, these conditions are typical to OTT voice services.
   - "Random packet loss and random jitter". This database handles reordering of packets.
   - "Manually designed packet loss". This database simulates a mobile device going in and out of coverage resulting in long and short consecutive packet losses.
2) Generic jitter files to be used for final validation (live recordings) (GenericJitterFiles_ValidationLiveData.7z). It is mandatory for this data set to be used only after the learning and validation phase is finalized, and a stable algorithm is in place; in this way this data set is "unknown" to the algorithm under test.
   - Four final validation generic jitter file sets based on live recorded data (drive tests).

Procedures and details on how the generic files have been created are presented in Appendix II.

7.3 Reference speech file

Models to be developed in accordance with this framework shall use a reference speech file which meets [ITU-T P.863.1] requirements and should be defined or referenced by the individual model descriptions.

8 Learning and validation database generator for IMS mobile EVS use case

This clause presents an example implementation of a simulator that can be used to build a database composed of codec specific jitter files and target MOS values for the IMS mobile enhanced voice services (EVS) case.

The learning and validation database generator consists of a simulator that uses as inputs: the reference speech sample(s) (clause 7.3), EVS codec parameters and the generic jitter files (clause 7.2) describing network error (jitter/loss) patterns. The error patterns are selected to cover the entire MOS range and this is to be verified for each use case (see Appendix III with MOS distributions). The output is the learning and validation database which contains a large set of "codec specific jitter files" (clause 8.4) with their corresponding target [ITU-T P.863] MOS values.

The simulator is shown in Figure 2 and all the simulator blocks are described in detail in clauses 8.1 to 8.4.

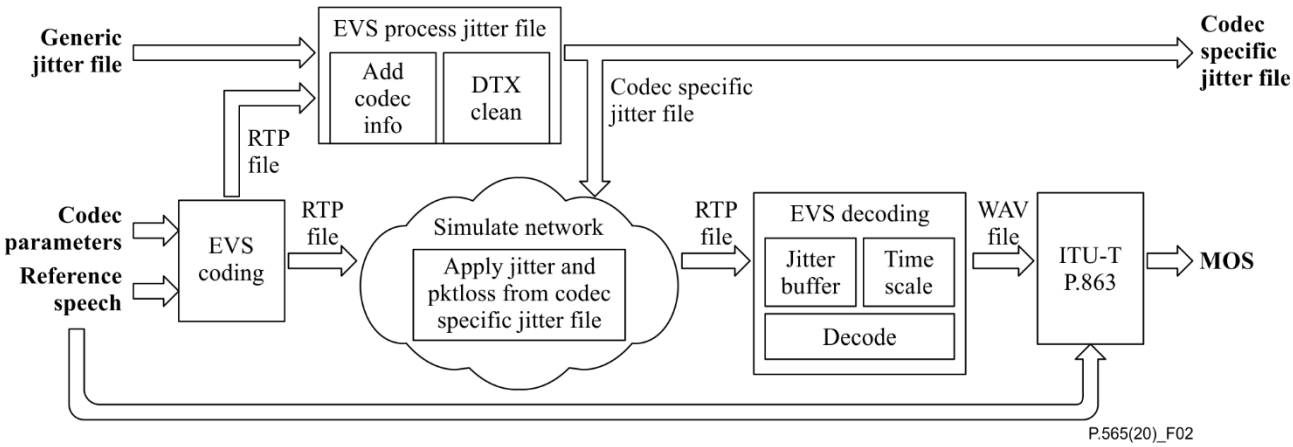


Figure 2 – Simulator for learning and validation databases generation

8.1 Simulate network block

The network simulation changes the arrival time of the EVS coded frames and removes coded frames when there is packet loss, both based on the codec specific jitter file (see clause 8.4).

8.2 EVS coding and decoding blocks: EVS codec and codec parameters

The simulator uses the standardized MTSI VoIP client implementing the EVS jitter buffer [ETSI TS 126 448] and the EVS codec and decoder. The standardized EVS VoIP client removes the variances which different VoIP client implementations can generate.

The reference speech file used as input to the simulator is coded with different settings for bandwidth, codec rate and channel aware mode (EVS coding). The coded packets are submitted to jitter and packet loss, by the simulate network block, as defined in the codec specific jitter file (see clause 8.4). Finally, the EVS decoding block combines jitter buffer, decoding and time scaling in order to produce a degraded speech file.

8.3 MOS grading block

The degraded speech file is the input to [ITU-T P.863] algorithm and the output is a MOS score.

8.4 EVS process jitter file block: processing of the jitter file

In order to keep the generic jitter files independent of codec and reference speech sample and to simplify the machine learning feature creation, it is required to perform two processing steps on the generic jitter files: discontinuous transmission (DTX) cleaning and adding codec information. The output of the processing steps is a codec specific jitter file which besides the sequence number and arrival time contains new columns with codec speech payload size and channel aware mode. The two processing steps are described in detail in clauses 8.4.1 and 8.4.2. The output of this processing is a new jitter file called "codec specific jitter file".

8.4.1 DTX cleaning

DTX cleaning pre-processing takes care of the fact that distortions during DTX periods (occurring during silence) do not impact the perceived speech quality. Consequently, the pre-processing operation creates a new jitter file with no packet loss and no jitter during DTX periods, which greatly simplifies the feature creation.

In the case of the EVS codec, there is a DTX packet during silence every 8*20 ms which is a network dependent parameter. The payload of a DTX packet is less than an ordinary speech packet. The DTX-cleaning pre-process generates packets every 20 ms and the arrival time continues from the moment the DTX period started. Any reference speech sample starts with silence, consequently a DTX period. This is a special case which requires a different way to calculate the arrival time. It uses as reference point, the first speech packet in the reference speech sample and recalculates the arrival time backwards from this packet. The result from the DTX-cleaning pre-process is a new jitter file called "codec specific jitter file" and a new coded speech file with no added degradations during DTX.

8.4.2 Add–on codec information

During the second pre-processing step the codec information is added to the codec specific jitter file, and it consists of speech payload size and channel aware mode indication. Packet payload size indirectly gives information on both codec rates if DTX was used. This information is given for every packet since they can change on packet level.

To summarize, the pre-processing handles codec specific operations like DTX, codec rate, and channel aware mode, and consequently keeps the generic jitter files, both codec and reference speech sample independent. Therefore, these files can be used in a generic manner in the framework.

8.5 Learning and validation databases

The learning and the validation databases are the output of the simulator (Figure 2) and contain:

– The codec specific jitter files.

– The wave files corresponding to the codec specific jitter files.

– [ITU-T P.863]/MOS scores of the wave files.

These databases are used as inputs to the machine learning module.

9 Machine learning module for IMS mobile EVS use case

The ML module, as shown in Figure 3, uses the learning and validation databases as inputs for selecting the features used as inputs to the ML algorithm, as well as for predicting [ITU-T P.863].

The outputs of the machine learning module are:

– The ML based speech predictor according to this framework.

– The ML algorithm.

– The features used as inputs to the ML algorithm.

Within the context of this framework the ML algorithm uses a 50/50 split for learning and validation in order to ensure a controlled and structured data content, and a good and even coverage of all distortions cases covering the entire quality range, from poor to very good quality.

NOTE – Other splits, such as 80/20 split randomly selects the data for learning, and there is a risk that a special case can end up only in the validation set and excluded from the learning process, or conversely a special case to be only in the learning set and therefore never validated.

9.1 ML algorithm

This Recommendation does not specify an ML algorithm, nor values for its parameters, in order to enable the inherited adaptability nature of the ML algorithms and to exploit it towards the best solution. Regardless of the algorithm, the parameters' selection must be performed to avoid overfitting/underfitting towards a specific database. The proof of potential under- or overfitting is obligatory, an appropriate example of how to perform the ML algorithm's parameters selection is presented in Annex A.

NOTE – ML decision trees category (e.g., Tree Bagger) with the two main parameters Number of trees = 100 and Leaf size = 5 show to best fit within the context of the framework, in terms of correlation performance and algorithm processing time length. Other algorithms such as support vector regressor and neural networks prove to work similarly.

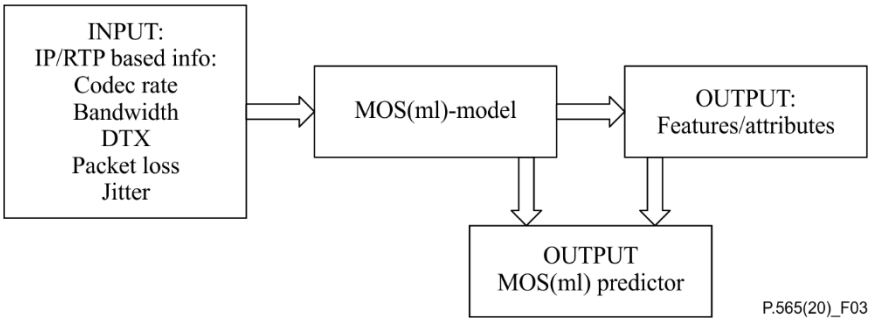


Figure 3 – Machine learning based model architecture

9.2 ML features

The content of this clause is considered as an example of realization of a model for the IMS mobile EVS use case. Other individual model realizations may use different feature sets but should also follow the principles of this clause.

9.2.1 ML features creation

The features' creation process is based on two main sources:

– The information from the real-time transport protocol (RTP) stream generated by a simulated jitter buffer implementation.

– Statistical metrics built from the RTP stream.

In order to improve the correlation towards [ITU-T P.863] MOS scores, reference speech-based features are introduced. These features shall use weightings based on the location of the feature in the reference speech sample. The position is described either by a feature giving the position of a packet loss dip or by weighting the number of frame erasures. The root mean square (RMS) of each 20 ms speech frame in the reference speech sample is used as weighting function. It should be noted that information on the reference speech is used, and not on the received degraded speech. Regardless of the technique, the weighting does not involve any explicit perceptual modelling.

The process refers to EVS case, but other codecs can require other and/or additional weightings techniques (e.g., bit rate). Details on the ML features' creation are presented in Appendix I.

Figure 4 shows the ML features along with their significance/importance for EVS use case.

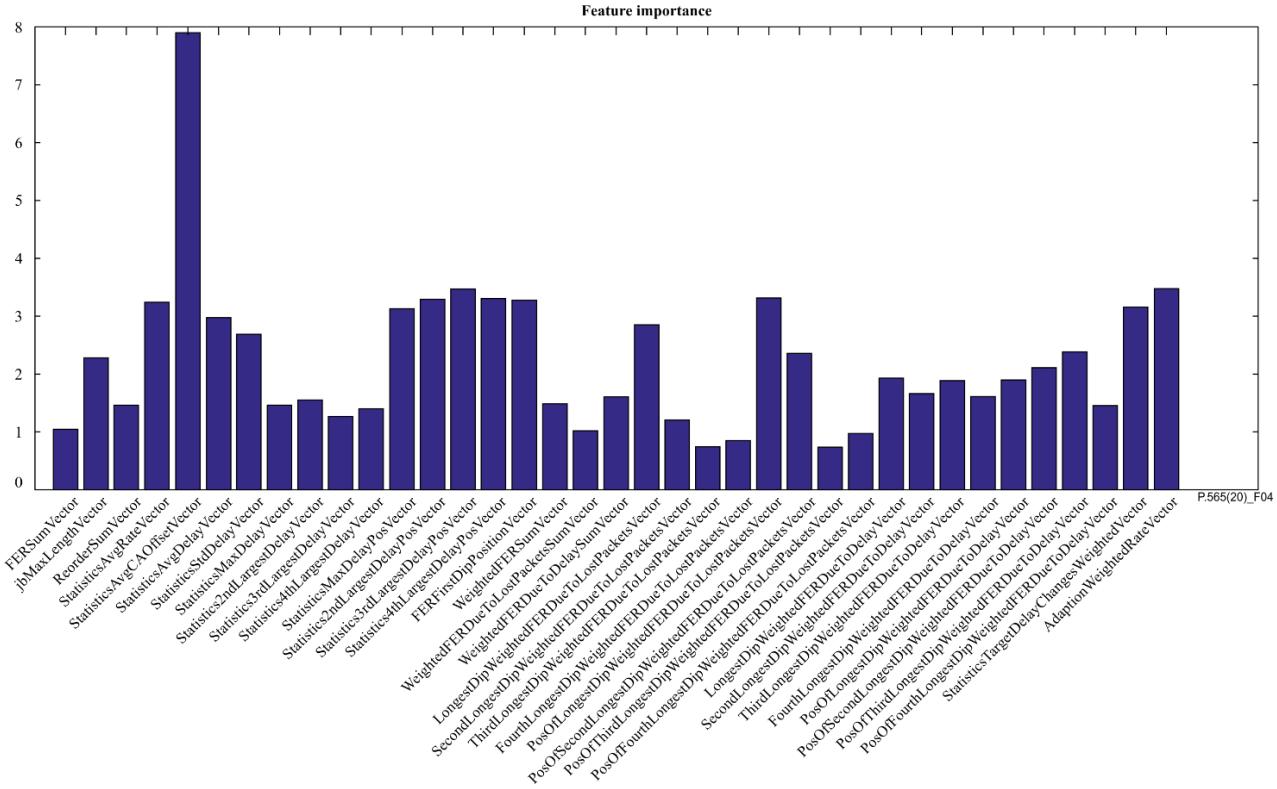


Figure 4 – Machine learning features and their significance for EVS use case

9.2.2 ML features selection

There are a number of ways to check the importance of the features and select the most meaningful ones. One possibility is to add and remove features in a structured manner, finding out which features improve correlation towards [ITU-T P.863]. However, this procedure is not only tedious, but it is also to a large extent hard to control due to the complex inter-dependencies between the features. Therefore, machine learning algorithms shall be used for features selection.

10 Statistical evaluation module

The statistical evaluation module is transparent to the use case and executes two tasks:

– It performs the statistical evaluation of the ML based predictor according to this framework using [ITU-T P.863] generated scores as target reference values.

– It verifies that the ML technique does not show overfitting and/or underfitting on the used databases.

Models developed based on the framework are in conformance with the requirements of this Recommendation only if the performance is evaluated as described in this clause. In addition, it is mandatory to use the provided "unknown" data set only for the final validation of the model.

The following evaluation steps shall be applied:

(i) Calculation of the correlation coefficient and prediction error (e.g., RMSE) statistics as main performance metrics, see [ITU-T P.1401]; required to show that models developed based on the framework meet ITU standards' requirements.

The results should be presented for validation simulated databases, and the unknown live database (Annex D). Unknown data set must be used only for the final validation.

(ii) Analysis of the absolute error (MAE) distribution; it is a required proof of meeting ITU standards requirements.

The results should be presented for validation simulated databases, and the unknown live database (Annex D). Unknown data set must be used only for the final validation.

(iii) Calculation of the difference between the scores of the ML based predictor according to this framework and the scores of the [ITU-T P.863] for the best encoding quality (no errors) as well as for poor network quality (high FER values). The evaluation is required since it is crucial for speech quality prediction solutions implemented in field testing tools to accurately evaluate both the best and the worst quality.

(iv) Run the overfitting/underfitting test, as described by the procedure presented in Annex A. The test is required in order to ensure that the ML technique has been applied properly and the results are not biased towards a specific dataset (overfitting) and the algorithm's parameters have been optimized towards the best performance (or minimal error or underfitting free).

The overfitting/underfitting independency is also indicated by the performance results on the unknown live data set, as follows. In case the distortions are similar in the validation simulated and live datasets, it is inferred that no overfitting/underfitting is present if the models (developed based on the framework) show performance results on the validation live data statistically similar with the results obtained on the validation simulated datasets.

The performance results of the models developed based on the framework need to meet the minimum performance requirements described in clause 13.2 within statistical significance bounds. Note that the performance results are determined based on the comparison with [ITU-T P.863] generated scores, thus errors of [ITU-T P.863] and the trained model will accumulate when performing a comparison of results from subjective listening tests.

Annex III presents performance results as described in this clause for the IMS mobile (VoLTE) EVS case.

11 Framework's inputs and outputs

The following are the inputs to the framework, as depicted in Figure 1:

– Reference speech sample, which is a sentence pair with a male and a female talker and with characteristics that meet [ITU-T P.863.1].

– Generic jitter files (jitter and packet loss patterns).

– Codec and client information (e.g., EVS).

The outputs of the framework are:

– The model of the ML based predictor according to the framework.

– The performance results of the model.

The inputs of the model developed based on the framework are:

– The selected ML features are calculated based on the IP bit stream, codec and client information. The list of features can vary depending on the targeted use case; each codec involves a specific error concealment scheme that it is expected to require and/or benefit from additional features. See Annexes A and B for other use cases.

– The temporal structure of the reference speech sample, which is a sentence pair with a male and a female talker and with characteristics that meet [ITU-T P.863.1] requirements.

NOTE – Using the ML features as inputs that cover information based on both the IP stream as well as the reference speech sample brings the parametric concept to another level of complexity and performance, respectively "intrusive parametric".

The output of the model developed based on the framework is:

– The ML based speech quality predictor with values on the 1-5 MOS scale, where three decimals are used for statistical performance evaluation, and representing scores per individual speech sample. The ML based predictor's scores should adhere to the accuracy of the minimum performance requirements described in clause 13.2. The evaluated accuracy is related to predicting [ITU-T P.863] MOS scores and not to the prediction of subjective test results.

12 Aspects related to the run-time of models developed based on the framework

The framework's scope is to create a speech quality predictor which runs in field testing tools. Therefore, this clause specifies aspects related to its run-time in order to ensure the best performance of the predictor.

12.1 Operation mode

The speech quality predictor developed based on the framework operates with knowledge of the end point or embedded in the device (user equipment). In addition, it requires active testing using female-male sentence pair as reference speech sample.

12.2 Reference speech samples

The machine learning based predictor according to this framework is trained using a reference file. As described, the algorithm uses ML features based on the reference speech sample. Therefore, during run time, when the algorithm is deployed, the same reference file must be injected to the network under test. The measurement methodology and reference file requirements follow the [ITU-T P.863]/[ITU-T P.863.1] specifications.

12.3 Pre-processing at run time

The pre-processing phase shall be performed in the same way as during simulations. Codec information is retrieved through deep packet inspection. In addition, during run time the pre-processing needs to synchronize the reference speech and the IP/RTP stream to ensure that the position-based features reflect the reference speech. This shall be performed by correlating the pattern of silence interruption duration (SID) and speech frames with the reference speech sample(s). In the case of variable bit rate codecs (e.g., OTT) which are not using SID frames, the payload size shall be correlated with the reference speech samples.

Regardless of the use case, no recorded speech is needed.

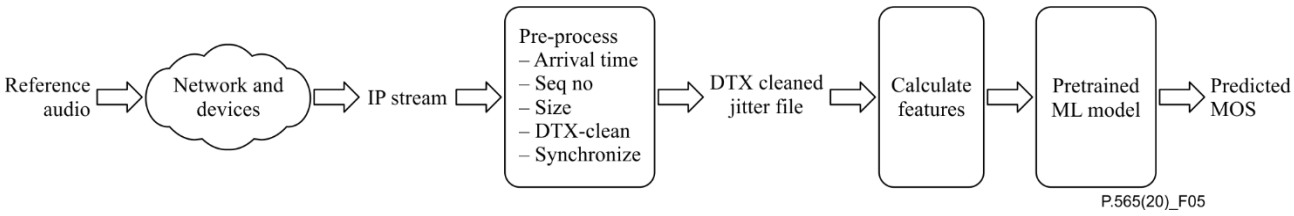


Figure 5 – Run time view of the model developed based on the framework (ML based predictor)

Figure 5 shows the implementation scheme of models developed based on the framework.

12.4 The measurement procedure

Models developed based on the framework use the same measurement set-up as for [ITU-T P.863], but, as mentioned above, without the need for recording the degraded speech sample.

In addition, comparing results achieved with both types of models, [ITU-T P.863] and ML based, enables the possibility for troubleshooting less obvious degradations such as the ones occurring outside of the transport network, e.g., emerged from automatic gain control, voice enhancement devices, transcoding, bridging or analogue speech processing.

13 Requirements for models developed based on the framework

13.1 Mandatory conditions and procedures

This Recommendation offers the user a framework based on which a ML based speech quality predictor is developed. The ML predictor refers to a network centric view and shall not be used when the evaluation of the end to end quality including the device's speech path processing (e.g., frequency response) is required.

The following steps are mandatory for developing an ML based speech quality predictor using the framework in this Recommendation.

– Select the use cases the model is aimed for, as e.g., use case as in clause 8 (IMS mobile EVS use case) to set-up the framework.

– Use Appendix I as an example for guidance for ML algorithm's features creation and selection.

– Use an ML algorithm known not to be easily prone to overfitting, such as from "decision tree" category (e.g., Tree Bagger). The Recommendation is not limited to a specific ML in order to maximize the benefit of the inherited nature of adaptiveness characteristic of ML techniques. See clause 9 and Annex A for details.

– Use the generic jitter files (clause 7.2) to run the framework's simulator.

– Learning and validation of the model:

  • Use the generic jitter files of the Recommendation for learning (see clause 7.2, "Generic jitter files") with a 50/50 split and/or learning datasets outside the Recommendation; use Appendix II for guidance on how these are created.

  • Use the generic jitter files of the Recommendation for validation (see clause 7.2, "Generic jitter files") with a 50/50 split; use Appendix II for guidance on how these are created.

  • Use the generic jitter files of the Recommendation for validation (unknown live dataset, see clause 7.2, "Generic jitter files").

13.2 Minimum performance requirements

A model shall prove that it has been developed based on the framework described herein as follows:

– It shall be proven that all the steps in clause 13.1 have been followed.

– It shall be proven that the performance evaluation meets clause 10 requirements (statistical evaluation module) and the results are presented individually for the following datasets:

  • Learning and validation simulated data (see clause 7.2 "Generic jitter files").

  • Unknown validation live data (see clause 7.2 "Generic jitter files"). These results must be provided only as the final validation; the unknown validation live dataset shall not be part of the training process of the model.

– It shall be proven that the average performance results across all data sets are within the range of the minimum required values as defined in Annex B.

  These values are specified based on long time ITU-T experience with requirements for various types of quality of experience (QoE) models. In addition, Appendix III provides

justification of these minimum required performance values based on the results that an ML predictor in accordance with this framework can achieve for IMS mobile EVS use case.

A model is accepted as being in conformance with this Recommendation if the requirements' check list in Annex B are met. Annex C specifies the mandatory conditions and requirements of an additional independent validation, within the statistical significance bounds. In both cases, the objectives should be met for all modes and bit rates as well as for the most relevant modes in a certain use case, typically the codec and bitrate combinations actually used in the real-field (e.g., EVS 24.4, EVS 13.2, AMR-WB 23.85, AMR-WB 12.65 for the VoLTE use case). The relevant bitrates and modes are considered equally in the aggregated statistical analysis. The selected modes for validation must be reported for the individual model realizations. For information, the individual performance for all other modes and bitrates should be reported too if applicable.

Annex A

Example method for ML overfitting/underfitting test

(This annex forms an integral part of this Recommendation.)

One of the challenges of machine learning (ML) resides with their sensitivity to overfitting or underfitting to the training/learning dataset which can generate misleading results of the applied ML algorithm.

It shall be noted that overfitting/underfitting to the learning (or training) data sets is a problem that can occur even for non-ML speech quality predictors. ML based predictors such as decision trees used in the context of this Recommendation have the benefit of offering mathematical methods to verify overfitting/underfitting. Therefore, once the ML is selected based on its performance against the minimum required performance of the predictor, the overfitting and underfitting test shall be performed.

The ML overfitting/underfitting test is described in this annex.

The optimization process used for machine learning models' development is the minimization of the learning and validation errors. The learning error is defined as the average loss over the learning points [b-Huang, et al]:

$$\frac{1}{N}\sum_{i=1}^{N} L(y_i - f_w(x_i))^2 \qquad \text{(A-1)}$$

Where N is the total number of observations in the learning set, L is the "loss function" (e.g., mean squared error (MSE)) over each one of the learning observations, f_w is the "fit function" (estimated values) using the learning data. The validation error is defined as the average loss over an independent test sample.

The trend of learning and validation error's curves provide insights on two major metrics, bias (a.k.a overfitting) and variance (a.k.a underfitting or accuracy) [b-Friedman, et al], which indicate how well the machine learning model is performing under evaluation. The bias and variance metrics are typically validated by observing the trend of learning and validation errors while tuning each of the model's parameters ("hyperparameters") and repeatedly testing them against the loss function (MSE). Generally, a large difference between the learning and validation error indicate high variance, while a high learning error indicates high bias.

The trade-off between the two metrics, bias and variance, provides an indication of a reoccurring machine learning problem – overfitting and underfitting [b-Friedman, et al].

Scenario for the use cases presented in this Recommendation

In this annex, learning curves [b-van Rijn et al] are used as technique for testing ML model's under/over-fitting.

The underfitting phenomenon is avoided by the fact that the hyper-parameters of the selected decision tree algorithm have been optimized towards the minimum required performance for the task at hand, respectively speech quality prediction. This is proved with the performance results as described in clause 10 and Appendix III.

The overfitting phenomenon is evaluated by determining the learning curves using multiple learning data set sizes and test each of them against a fixed validation data set size, while observing the changes of the errors, for both the learning and the validation datasets.

The total size of the datasets used for the presented case is of 128661 samples. For the purpose of the learning curves test, the learning data set sizes are selected as:

$$\text{Train_size} = 128661 * s \qquad \text{(A-2)}$$

where s = ([0.00001, 0.01, 0.10, 0.33, 0.55, 0.78, 1.]). For each of the learning size a five-fold cross validation is performed; random samples for each of the learning size are taken five times, prediction is determined, and the results are averaged. Figure A.1 shows the results. In the first run, with a train set size of 1, the model perfectly fits the learning data, thus a learning error of 0. However, a model with only one learning set performs poorly on unknown validation sets, thus a high validation error which shows an overfitted model. Further, as the learning set size increases, the validation error gradually decreases, reaching arguably acceptable MSE value of just below 0.4. At the same time the learning error is kept at minimum, which indicates low bias (overfitting) and discards the underfit problem [b-Friedman, et al], as expected. In addition, the variance between the MSEs for learning and validation data sets is low, indicating that overfitting is not likely.

The trend of the validation error suggests that overall accuracy can be improved by increasing the number of samples, but only very slightly.

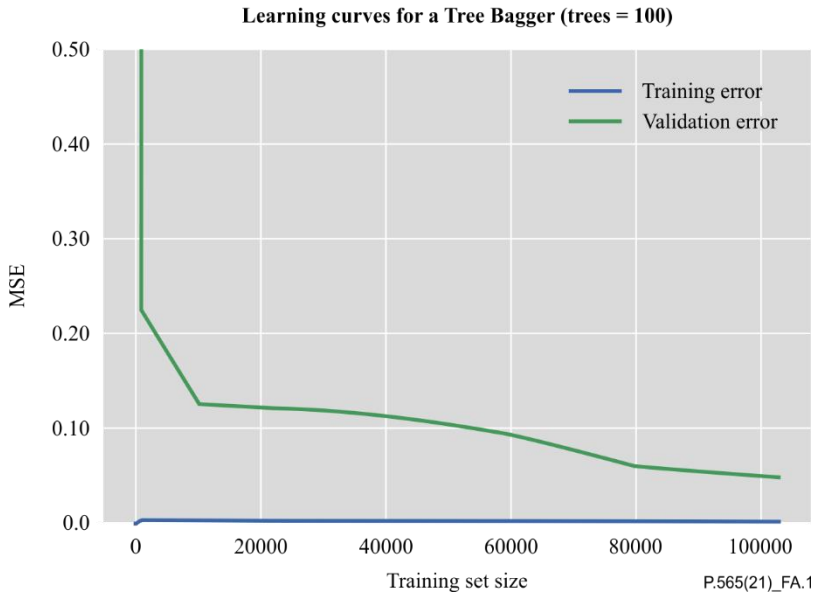


Figure A.1 – Learning curves

General scenario

In a more general case, with N as the size of the training-validation dataset, the learning curve test can have two outcomes, overfitting and/or underfitting, as shown in Figure A.2 a, b.

In the case of the Tree Bagger algorithm, the MSE is calculated for different sizes of the learning set and for the validation set. Figure A.2 a, b shows the two possible outcomes.

In the first case (Figure A.2 a), the error on the learning databases (blue curve) is high, regardless of the learning size, which means that the model is too complex and cannot learn and adapt/tune. At the same time the validation dataset (green curve) shows the same high error respectively, similar to the one for the test set. This means that the algorithm is underfitted. New learning points will not help, since this kind of results indicate that the algorithm itself has not been properly selected for the use case in discussion. The underfitting scenario can be avoided selecting the suitable ML algorithm to the task at hand as well as to optimize each of its hyper-parameters towards the minimum required performance defined by the speech quality prediction.

In the second case (Figure A.2 b), the error for the learning set is low, while the error for the validation set remains far apart from it, regardless of the learning set. This means that the algorithm is overfitted. New learning points can help to improve this outcome and consequently ensure to avoid overfitting.

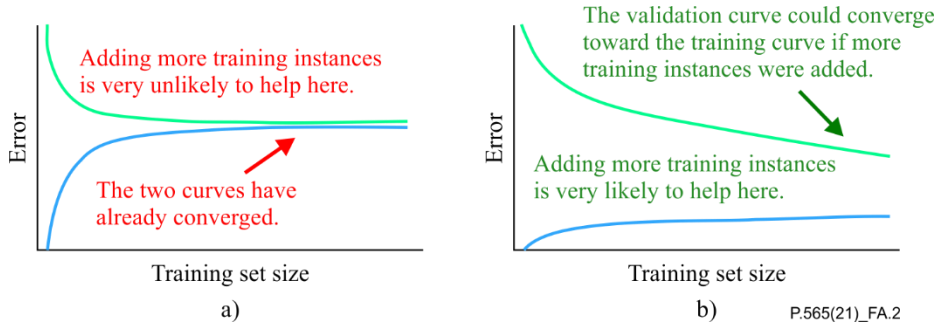


Figure A.2

The results of the under/overfitting test for the IMS EVS use case presented in this Recommendation are presented in Appendix III.

Annex B

Check list of requirements for a model developed based on the framework

(This annex forms an integral part of this Recommendation.)

A model developed based on the framework described in this Recommendation must follow the requirements presented in Table B.1. The results should be provided and presented in a contribution to an ITU meeting. After reviewing the presented results, the model should be accepted as being in accordance with the Recommendation unless an additional independent validation is required by a participant during the meeting. See Annex C for the procedure in this case.

The acceptance of a model will be recorded in the meeting notes. The model developing party shall be allowed to draft a new Recommendation to the series which specifies the validated model.

The objectives have to be proven for

Set A: All available bit rates (aggregated) declared by the model developing party for each of the defined use cases, as well as for

Set B: The most relevant bitrates and modes (aggregated) of the defined uses cases (real-field), where this is a subset of all rates as above.

These most relevant bitrates in Set B are for the example of 'IMS mobile VoLTE EVS' the codec settings EVS 24.4 kbit/s SWB and EVS 13.2 kbit/s SWB, for 'IMS mobile VoLTE AMR-WB' the codec settings AMR-WB 23.85 kbit/s and AMR-WB 12.65 kbit/s.

For acceptance of the model, the objective criteria must be met for the aggregated evaluation of all bitrates (Set A) for the defined use cases and for the defined relevant bitrates and modes (Set B) in each use case.

The performance for all possible bitrates and modes should be reported per bitrate for information purposes only.

For IMS mobile use cases other than VoLTE (e.g., VoNR), there might be other relevant bitrates and modes and they must be declared by the model developer and justified by real-field measurements and agreed by the experts in ITU SG12 before the validation process starts.

Table B.1

Steps	Description/requirements
Develop simulator	As described in the text of the Recommendation (clause 8.1).
Create the speech sample database	Run the jitter files (Annex D) through the simulator and create wav files for each jitter file.
ML features selection	An example can be found in Appendix I.
ML algorithm selection	The decision should be made based on its performance towards meeting the requirements from this Recommendation.
Validate ML based predictor	Use provided validation simulated dataset (Annex D). It is mandatory to use for validation only the validation simulated dataset.
Run over/underfitting test and provide learning curves	Learning curves should show the errors for the learning and validation data set converge.

Table B.1

Steps	Description/requirements
Provide performance results on validation simulated data set: correlation, RMSE, MAE and AE distribution – For all bit rates belonging to a use case (Set A) – For selected bit rates corresponding to the real field application of each use case (Set B)	Min. req. performance as follows: Correlation coefficient > 0.95 RMSE < 0.35MOS MAE < 0.3MOS AE distribution • More than 65% of values with errors less than 0.25MOS • More than 85% of values with errors less than 0.5MOS • More than 95% of values with errors less than 1MOS
Final validation of the ML based predictor, MOS(ml)	Use validation unknown live data set. It is mandatory to use the unknown data set only for the final validation. Unknown live data set should not by any means be part of the training.
Provide performance results on validation unknown live data set: correlation, RMSE, MAE and AE distribution. – For all bit rates belonging to a use case (Set A) – For selected bit rates corresponding to the real field application of each use case (Set B)	Min. req. performance as follows: Correlation coefficient > 0.95 RMSE < 0.35MOS MAE < 0.3MOS AE distribution • More than 65% of values with errors less than 0.25MOS • More than 85% of values with errors less than 0.5MOS • More than 95% of values with errors less than 1MOS

Annex C

Conditions and requirements of an additional independent validation of a model developed based on the framework

(This annex forms an integral part of this Recommendation.)

This annex specifies an independent validation procedure of a model developed based on this Recommendation. This independent validation is in addition to the validation described in Annex B and shall take place if a request to perform an independent validation is received.

The validation procedure can be extended by a requested pre-validation to prove the reconstruction and decoding process as well as the creation of the reference values to discover potential problems in the training process. The pre-validation should not last longer than three months. Potential problems discovered in the pre-validation should be resolved by the model developing party before the validation process can be started.

C.1 Conditions and requirements of an independent validation

1) An independent validation should be requested during the same ITU-T meeting when a model developed based on this Recommendation is presented in a contribution to the meeting and it is claimed and proved that it passes the check list requirements described in Annex B.

   The performance requirements as defined in Annex B must be validated for an aggregation of all bitrates of the defined use cases (Set A) as well as for the aggregation of the most relevant bitrates and modes of the defined uses cases (Set B). These are for the example of 'IMS mobile VoLTE EVS' the codec settings EVS 24.4 kbit/s SWB and EVS 13.2 kbit/s SWB, for IMS mobile VoLTE AMR-WB' the codec settings AMR-WB 23.85 kbit/s and AMR-WB 12.65 kbit/s.

   For acceptance the model, the performance requirements as in Annex B must be met for both aggregations in Set A and Set B. The performance for all possible bitrates and modes should be reported by bitrate for information purposes only.

   For IMS mobile EVS use cases other than VoLTE (e.g., VoNR), there might be other relevant bitrates and modes and they must be declared by the model developer and justified by real-field measurements and agreed by the experts in ITU SG12 before the validation process starts.

   If no request for independent validation is made, then the model shall be accepted as being in accordance with this Recommendation without explicit validation on an additional unknown data set.

   The acceptance of a model will be recorded in the meeting notes, and the model developing party shall be allowed to draft a new Recommendation to the series which specifies the validated model.

2) The independent validation party shall perform the validation in good faith and submit the results within 6 months from the time of the independent validation request, otherwise the model is accepted as being in accordance with this Recommendation. During the validation processes the independent party and the model developing party shall have a close and in good faith collaboration and exchange of information and keep interested parties informed of the progress.

3) The independent validation shall use a completely new unknown dataset, which has not been part of the model development (learning/training) and/or validation and which is therefore different from the unknown data set provided by this Recommendation. It is

required that the new unknown data set meets the Recommendation's requirements, and its scope.

4) The new unknown data set shall be created by the independent validation party requesting the validation. The model developing party shall support, in good faith, the process of the data creation (see "Validation procedure" in clause C.2).

5) Based on the results of the independent validation on new additional unknown data and whether the performance criteria are met within statistical significance, the model shall be accepted as being in accordance with this Recommendation.

6) After the completion of the individual validation process, the independent validation party shall donate the unknown created database to be added to this Recommendation for further improving the learning process of ML based solutions for speech quality prediction.

7) The independent validation party and the model developing party can agree to modify these rules and periods of time in case it is needed and in good faith to all involved parties. However,

   a) any change of rules or timeline that might be needed for various reasons shall be justified and interested parties be informed about it;

   b) a time window of up to 9 months for the entire validation process shall not be exceeded;

   c) any dispute that might occur between the two parties due to various reasons shall be made public.

C.2 Validation procedure

1) Within 15 days from the date of request, the model developing party should:

   a) Provide all the 'auxiliary' components required for creating jitter files, reconstructing wave files and others to the independent validation party, so that the validation party is able to process and verify all intermediate steps and build a database with samples that are well-distributed in the MOS range (as required in point 2).

   b) Provide the model's executable together with all 'auxiliary' components to ITU TSB for storage and potential clarification of differences in results between the developing and the validating party.

2) Within 60 days from the date of receiving the executable, the independent party shall send to the model developing party the newly unknown created generic jitter files data set (use Appendix II as guidance) which shall meet the following:

   – cover conditions only as described in Table 1 and within the scope of this Recommendation;

   – contain a min. of 500 files. This count is in alignment with the size of unknown data set provided by this Recommendation, and therefore large enough for a fair validation while keeping the validation running time within reasonable limits;

   – cover real live network conditions;

   – evenly cover degradation conditions which can generate MOS scores on the entire MOS range. The number of jitter-files leading to perfectly reconstructed wave files must not exceed 10% of the total amount of conditions.

   – The selection of jitter files to construct a database with evenly distributed qualities by the validating party is preferably done by using the wave-file reconstruction process and its components provided by the model developing party. It should be ensured that the ML model is not available at that time to the validating party and the selection of the test conditions is purely made on quality predictions on the wave files. In case, the wave file reconstruction components are not available to the validating party,

alternatively quality prediction obtained on the wave files decoded by the recording device can be used.

During this process, the model developing party should support, in good faith, the independent validation party. If timing cannot be kept, then the reason shall be justified and interested parties be informed.

3) After receiving the generic jitter files dataset, the model developing party shall run the DTX cleaning and the simulator chain (as described in the Recommendation) in order to create the wave files correspondent to the generic jitter files for set A and set B. The reference speech file provided by the model developing party should be used.

The model developing party shall test if the wave files indeed cover the whole MOS range by running [ITU-T P.863]. Any found anomaly should be discussed and solved in good faith with the independent validation party. If unacceptable files are detected and agreed on by the two parties, then those need to be replaced in order to keep the min 500 generic jitter files count. The whole process shall not exceed 30 days and interested parties be informed about its progress.

4) The model developing party shall send the codec specific jitter files and correspondent created wave files to the independent validation party who should analyse these.

This can be performed on randomly selected files from the entire independent validation set. The findings shall show that both the resulting codec specific jitter file and the resulting WAV file has the same packet loss pattern as the generic jitter file. The same shall be found for jitter when it is large enough to trigger late loss.

Any found anomaly should be discussed and solved in good faith by the two parties. If unacceptable files are detected and agreed on by the two parties, then those need to be replaced in order to keep the min of 500 generic jitter file count. The whole process shall not exceed 60 days and interested parties be informed about its progress.

5) In the case in which both parties feel in good faith that additional checking is required, then the two parties shall meet in person or via video conference and run a supervised random check as follows. The independent validation party selects a set of generic jitter files and the model developing party generates the corresponding wav file by running DTX cleaning and the simulator chain on the spot under the supervision of the independent party. The number of files to be checked should be agreed on by the two parties in good faith and shall be kept reasonable in terms of required processing time for the DTX cleaning and simulation chain.

Any found anomaly shall be discussed and solved in good faith by the two parties. If unacceptable files are detected and agreed on by the two parties, then those need to be replaced in order to keep the min of 500 file count. The whole process shall not exceed 15 days and interested parties be informed about its progress.

6) The independent validation party shall run the model on the codec specific jitter files and [ITU-T P.863] on the correspondent wave files, run the analysis and share the results with the model developing party. Any found miscalculations shall be discussed and solved in good faith by the two parties. The whole process should not exceed 30 days.

The performance criteria shall be the same as in Annex D on unknown data set.

1) The independent party shall submit the results in a contribution to an ITU-T meeting. The model shall be accepted as being in accordance with this Recommendation with explicit validation on an additional unknown data set.

The acceptance of a model will be recorded in the meeting notes, and the model developing party shall be allowed to draft a new Recommendation to the series which specifies the validated model.

2) In the case in which the timing of the independent validation process is at risk of being elapsed, then the independent validation party shall present the results in an interim meeting and the discussion results will be recorded in the interim meeting report, and in the subsequent ITU-T meeting to be published in the final report of the ITU-T meeting.

Annex D

Electronic attachments

(This annex forms an integral part of this Recommendation.)

This annex contains the reference speech samples and the generic jitter files data bases to be used for the development and validation of the model. Infovista and Rohde&Schwarz kindly contributed to build these data bases.

D.1 Reference speech samples (FB and SWB)

– UsEnglish_FB_48k.wav

– UsEnglish_SWB_48k.wav

D.2 Generic jitter files data bases

Generic jitter files to be used in the model development

– for learning and validation (GenericJitterFiles_InfovistaTrainingAndValidation.7z)

Generic jitter files to be used in the model validation

– unknown data set for final validation (GenericJitterFiles_InfovistaValidationLiveData.7z)

– unknown data set for independent validation (GenericJitterFiles_Rohde&SchwarzIndependentValidationLiveData.7z)

D.3 Data bases description

The description of the data bases for model's development and final and independent validation is presented in Appendix II.

Appendix I

Procedure for feature extraction based on machine learning

(This appendix does not form an integral part of this Recommendation.)

An example on how the features can be developed is presented in this appendix.

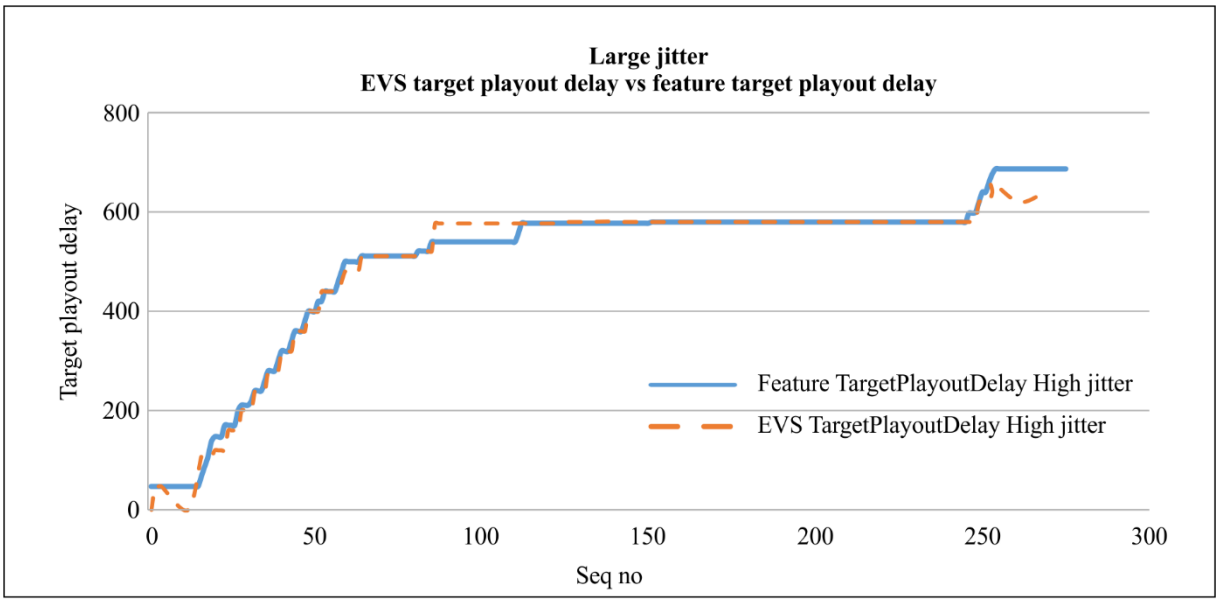
I.1 Create statistical features

A list of the statistical features is presented in Table I.1.

Table I.1 – Statistical features

Average delay
Standard deviation of delay
Maximum delay
Second largest delay
Third largest delay
Fourth largest delay
Target delay changes

These features are self-explained except "Target delay changes". Target delay mimics the statistical calculations performed by the EVS VoIP client. Simulations and tests are required to compare the target playout delay calculated by the feature and the target playout delay from the real EVS VoIP client. One example for very high jitter is shown in Figure I.1. The resemblance between the feature and EVS behaviour is very good.



P.565(20)_FI.1

Figure I.1 – Calculated target playout delay feature and real EVS client target playout

The target delay increases rapidly during parts of the speech with high distortions, see Figure I.2. This means the total amount of target delay change (in %) is a good feature. Correlation tests on data with severe jitter also indicate this, where the correlation increases from 0.84 to 0.88 by adding the target delay changes feature.

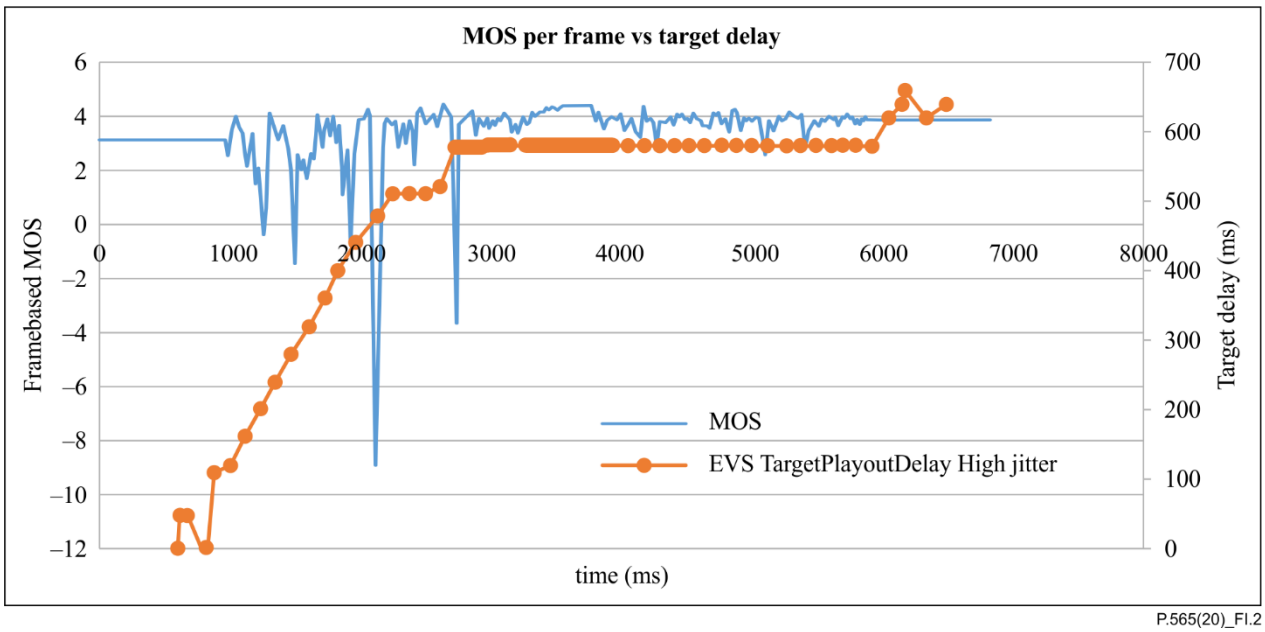


Figure I.2 – MOS and target delay during one speech reference of 5.5 s

I.2 Create jitter buffer-based features

A simulated jitter buffer is a good feature to handle time delay variations (jitter). An overview of the EVS jitter buffer management is shown in Figure I.3.

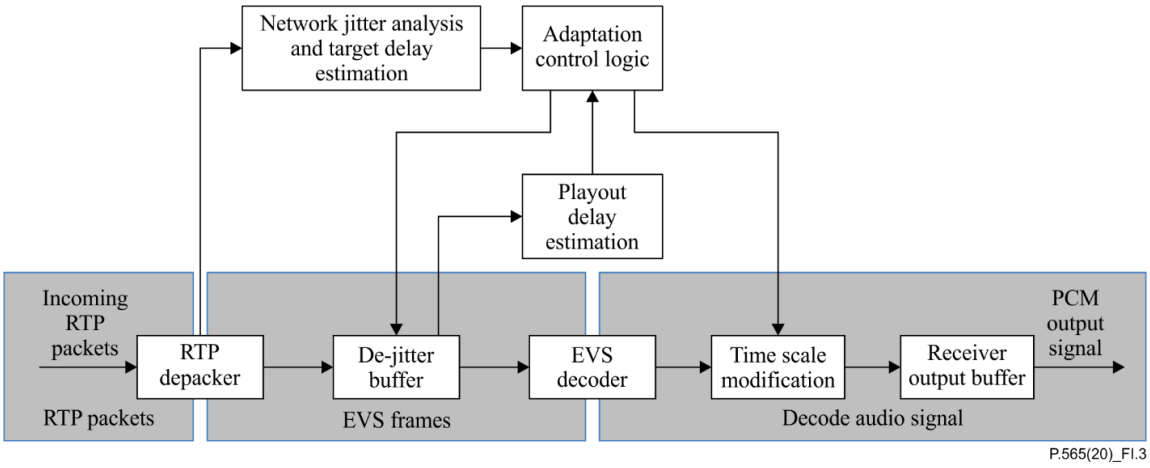


Figure I.3 – Modules of the EVS jitter buffer management solution from [ETSI TS 126 448]

Extensive tests and analysis are required to find the most important characteristics of the EVS codec and how it impacts speech. This is performed both by logging information from the codec and visualizing how different delays in the IP-stream affect the decoded speech. If a 20 ms speech frame cannot be played out from the jitter buffer when it should, it is called a frame erasure (FE). Frame erasure rate (FER) is the percentage of lost frames at the play out instance. Some important study cases for how the EVS handles network errors are as follows:

1) Network jitter during silence or close to silence. The packet loss concealment (PLC) generates silent or close to silent frames and increases the overall delay. All speech content of the reference speech sample is still in the recording, but later than it should be. Adding silence to already silent parts has no effect on the perceptual quality and [ITU-T P.863] MOS score reflects this. Figure I.4 illustrates how delays at silent positions have no effect on the MOS score. The feature following from this is FER due to delay weighted by RMS of the speech at the last played packet before the buffer became empty.

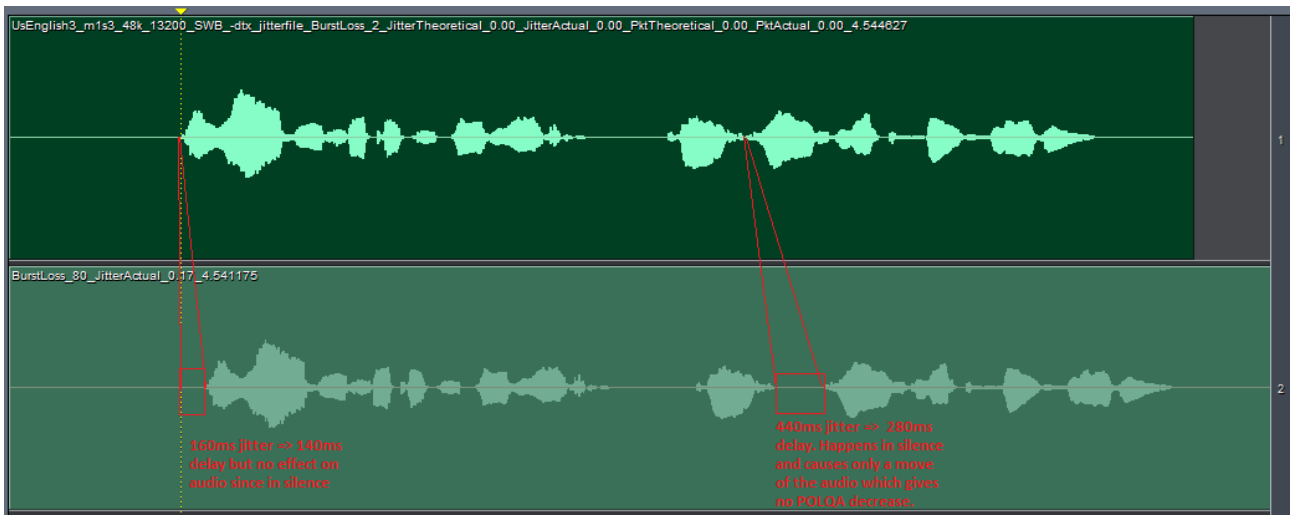


Figure I.4 – Ilustration of delays at silent positions

Figure I.4 shows that two long jitter delays in silent parts have no effect on MOS. Both of the examples shown have the same MOS of 4.5.

2) Network jitter during the speech. The PLC generates estimated speech and increases the overall delay. All speech content of the reference sample is still in the recording but with some degraded speech inserted. The PLC is based on the last packet before the delay and converges to the background noise (see clause 5.3.1 of [b-3GPP TS 26.448]). The convergence speed depends on the speech but about 3-4 packages are usually needed to reach the background noise. This is verified using a couple of speech examples. The first correct frame after the delay will be modified to avoid discontinuities (see clause 5.3.3.3 of [b-3GPP TS 26.448]). Correlation tests show no improvement by using a limit of 3-4 for the frame erasure so the feature described in case 1 can also handle this case.

3) Too late (out of order) or lost packets. The PLC generates estimated speech for the lost packets up to 3-4 packets. The degradation will become worse if more consecutive packets are lost, and this will be handled by the machine learning. The feature becomes: FER due to lost packets weighted by the RMS of the speech at the lost packet.

4) Delay in general. The jitter buffer prefers to keep delays, introduced by jitter, instead of removing packets to decrease the overall delay. This behaviour has been built into the simulated jitter buffer and is reflected by a feature called *Jitter Buffer max length*.

5) Actual play out delay differs from target playout delay. When the target delay differs from actual delay the jitter buffer can perform either frame-based adaption or signal based adaption (time scaling). To capture this, a feature called *Adaption* is introduced. It is the difference between target playout delay and estimated playout delay when larger than 60 ms.

6) Out of order packets. This case is handled by a feature called *ReorderSum* which counts the number of reorders in the simulated jitter buffer.

Table I.2 shows a list of the jitter buffer based features.

Table I.2 – Jitter buffer based features

Adaption rate
FER
FER due to delay
FER due to lost packets
FER dips 1..4
FER dip positions
Jitter buffer max length
ReorderSum

I.3 Codec based features (rate and channel aware)

The DTX cleaning pre-process adds two columns in the cleaned jitter files with codec data for each packet:

1) Speech payload size
2) Channel aware mode

Speech payload size

The speech payload size is used as a measure of the codec rate. The feature is calculated as the average of the payload size, with DTX packets excluded, and thereby it can handle the variable rate of the EVS 5.9 Kbit/s and might also handle rate adaption which could be enabled in the future.

To check if the ML algorithm can predict all rates in one model or if separate models for each rate are needed the test below is performed. On the database for severe jitter, simulations of rates 9.6, 13.2, 16.4 and 24.4 are run with learning performed both on all rates and on each separate rate. The test shows almost the same correlation for both so one model for all rates can be used.

Channel aware mode

The channel aware mode improves the error resilience especially on channels with burst errors by piggybacking important speech on later packets. In this way even if a packet is lost the important part of its speech can be recreated from a later packet. The results (Appendix III) show that the machine learning based model predicts MOS/[ITU-T P.863] within minimum requirements as defined in clause 13.2.

I.4 Create reference speech-based features

Information from the speech references is used to create some of the features that are used by the ML algorithm. The information is speech based and it can be, but is not limited to, the location of a FE dip, speech energy per 20 ms frame or the encoded payload size per 20 ms frame.

This kind of information can be weighted towards optimizing the ML based predictor for minimum RMSE and the highest correlation coefficient.

I.4.1 Types of reference speech-based features

Three types of reference speech-based features are suggested using the above-mentioned information.

1) The location (packet number or time in ms since the start of the reference) of the start of a delay or a FE dip (explained in clause I.2).

2) Versions of jitter buffer features weighted with speech energy, for example weighting the FE dips, based on the root mean square (RMS) of 20 ms frames in the reference speech. The reason for the FE, delay or loss, decides which speech frames should be used for weighting, explained in clause I.2.

3) Versions of jitter buffer features weighted with codec payload size, for example weighting the FE dips based on the payload size of the 20 ms frames from encoded reference speech. The reason for the FE, delay or loss, decides which speech frames should be used for weighting, explained in clause I.2.

I.4.2 Features' weighting function calculation

Different weighting functions and scaling of the weights can be used for each of the reference speech- based features.

The following techniques proved to work well providing good performance of the ML based predictor (see results in Appendix III).

Speech energy weighting

For the speech energy weighting an RMS with a constant scaling of 3 and a limitation to zero for values below 0.3, proved to provide a good compromise between implementation effort and result.

$$weights_{audioEnergy}(index_{frame}) = \begin{cases} 0, \ \ 0 \leq RMS(index_{frame}) < 0.1 \\ 3 * RMS(index_{frame}), \ \ 0.1 \leq RMS(index_{frame}) \end{cases} \quad \text{(I.1)}$$

Payload weighting

There are two scenarios: fixed and variable codec bit rates. Fixed bit rates characterize AMR WB and all EVS codec rates except 5.90 rate. Variable bit rates characterize EVS 5.90 rate and OTT variable bit rate (VBR) codec.

Based on tests of various combinations of weights towards optimized RMSE and correlation coefficient the weighting functions are calculated as follows:

1) EVS and AMR WB fixed bit rate (all rates except EVS 5.90)

- Payload size corresponding to DTX => 0 as weighting value
- Payload size corresponding to Speech => 1 as weighting value

Equation I.2 describes the weighting technique in this case. The weights and the scaling function are used to decide on the payload size.

$$weights_{codec}(index_{frame}) = \begin{cases} 0, \ \ 0 \leq currentPacketSize_{referenceAudio}(index_{frame}) \leq 48 \\ 1, \ \ 48 < currentPacketSize_{referenceAudio}(index_{frame}) \leq 488 \end{cases} \quad \text{(I.2)}$$

2) OTT variable bit rate and EVS 5.90 rate

In this case, the optimal weights have been determined as:

$$weights_{codec}(index_{frame}) = \begin{cases} 0, \ \ 0 \leq CodecPacketSizeRatio(index_{frame}) < 0.5692 \\ \log_2\left(1 + CodecPacketSizeRatio(index_{frame})\right), \ \ 0.5692 \leq CodecPacketSizeRatio(index_{frame}) \leq 1.0 \end{cases} \quad \text{(I.3)}$$

Where CodecRatio is scaling the weights based on the equation I.4:

$$CodecPacketSizeRatio(index_{frame}) = \frac{currentPacketSize_{referenceAudio}(index_{frame})}{maxPacketSize_{referenceAudio}}, 0 \leq index_{frame} \leq NoOfPacketsInReferenceAudio \quad \text{(I.4)}$$

The list of the weighted features is presented in Table I.3.

Table I.3 – Weighted features based on speech payload

Weighted adaption rate
Weighted FER
Weighted FER due to delay
Weighted FER due to lost packets
Weighted FER dips 1..4

Appendix II

Descriptions of generic jitter files creation

(This appendix does not form an integral part of this Recommendation.)

This appendix describes procedures and how the generic jitter files have been created.

II.1 Generic jitter files for model development and final validation (source Infovista)

II.1.1 Learning and validation generic jitter files

The learning and validation generic jitter files cover the following simulated databases:

– "Live (drive test) data modulated with simulations" to broaden the conditions' range (e.g., randomizing the live degradation position and amplitude).

– "Gilbert burst packet loss and burst jitter up to 30%". This mimics error cases seen in live drive test data, for example during handover, where packets are buffered and suddenly released.

– "Gilbert severe burst jitter to 70%". This database improves the learning and testing of large jitter which is the hardest case to predict. In addition, these are typical for OTT voice services.

– "Random packet loss and random jitter". This database handles reordering of packets.

– "Manually designed packet loss". This database simulates a mobile device going in and out of coverage resulting in long and short consecutive packet losses.

II.1.1.2 Live (drive test) data modulated with simulations

The drive route for the live test was carefully planned to give varying results. Measurements were performed on two of the most popular brands of smartphone devices on three of the large tier 1 operators. All operators used DRX (discontinuous reception) which gives a "background" jitter with arrival times of 0 ms or 40 ms on each second packet.

A base jitter file which can be modulated is created from the parts of the RTP stream resulting in decreased MOS. The cut-out parts are synchronized with the speech reference. DTX at the start, middle and end are replaced with normal background jitter (0 or 40 ms). This result represents one base jitter file. The base jitter file is run through a script to modulate the error patterns. First, the jitter and packet loss are shifted to 20 different positions to simulate the errors occurring at any time during the speech. Second, for every of the 20 positions, the positive jitter (local delay increase), is also amplified by 10%, 25%, 50% and 100%, see Figure II.1. Every base jitter file thus results in 100 generic jitter files.

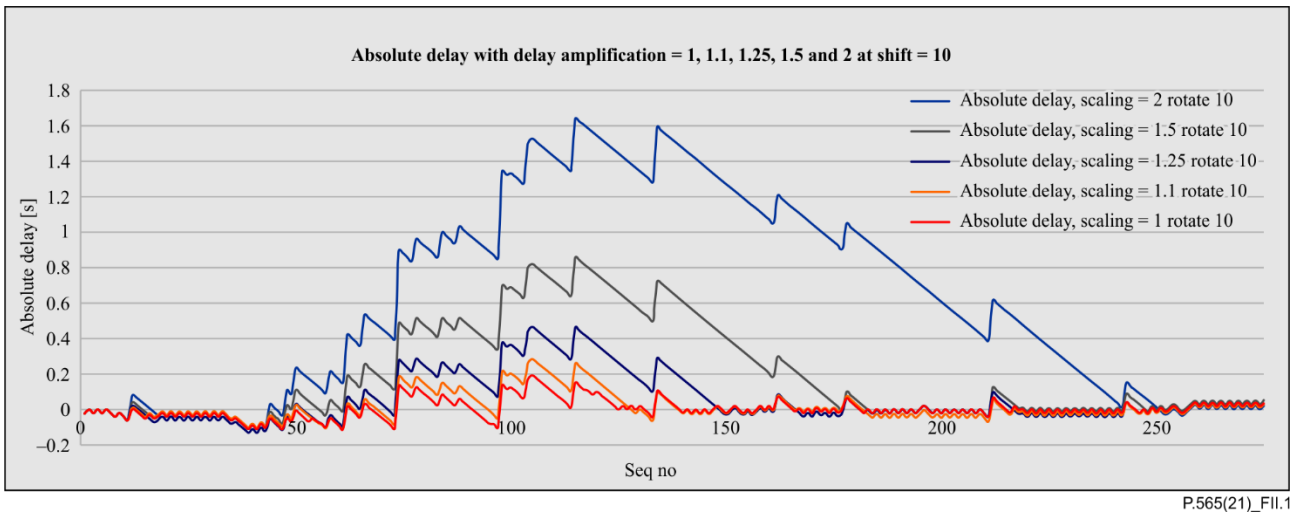


Figure II.1 – Delay amplifications for a live base Jitter file

The positive jitter amplification is performed when the time difference is larger than 55 ms, based on the jitter buffer's initial size of 60 ms, to avoid amplifying the jitter background from DRX and other insignificant jitter delays. To avoid the absolute delay to increase beyond reasonable values, amplified positive jitter must be followed by amplified negative jitter (local delay decrease). The negative jitter is performed where a negative jitter in the original RTP stream exists. Negative jitter is detected when the time difference is almost zero for two packets in a row. The negative jitter is applied as long as there is no new local delay increase. The sum of the delay increase and decrease is managed by a delay budget where the aim is to get the budget to zero. This results in the amplified jitter keeping a similar shape to the original delay curve and implies that the scaling is a good simulation of real jitter, see Figure II.1.

II.1.1.2 Gilbert burst packet loss and burst jitter

A Markov chain according to the Gilbert model is implemented to generate burst packet loss. The model is extended to jitter by calculating the delay peaks from the number of consecutive losses generated by the Gilbert model. When a peak occurs, it will be followed by packets arriving at the same time until the total delay goes to zero or a new burst loss occurs, using the same method as described in delay amplifying of the live data.

When there is jitter, the EVS jitter buffer (and generally most other buffer implementations) will prefer to increase the total delay and thereby keep the late loss down. This means the jitter buffer makes the probability of late loss larger in the beginning of the speech than at the end when the buffer has had time to increase. To make the FE due to late loss more uniformly distributed over the speech, the delay generated from the Gilbert simulation is extended with a probability for the start of the simulation. The start is rectangularly distributed between packet number 1 and 175 so that there are always at least 100 packets for the Gilbert produced delays to be simulated over. Figure II.2 shows the frame erasure due to late loss to be evenly distributed over the speech.

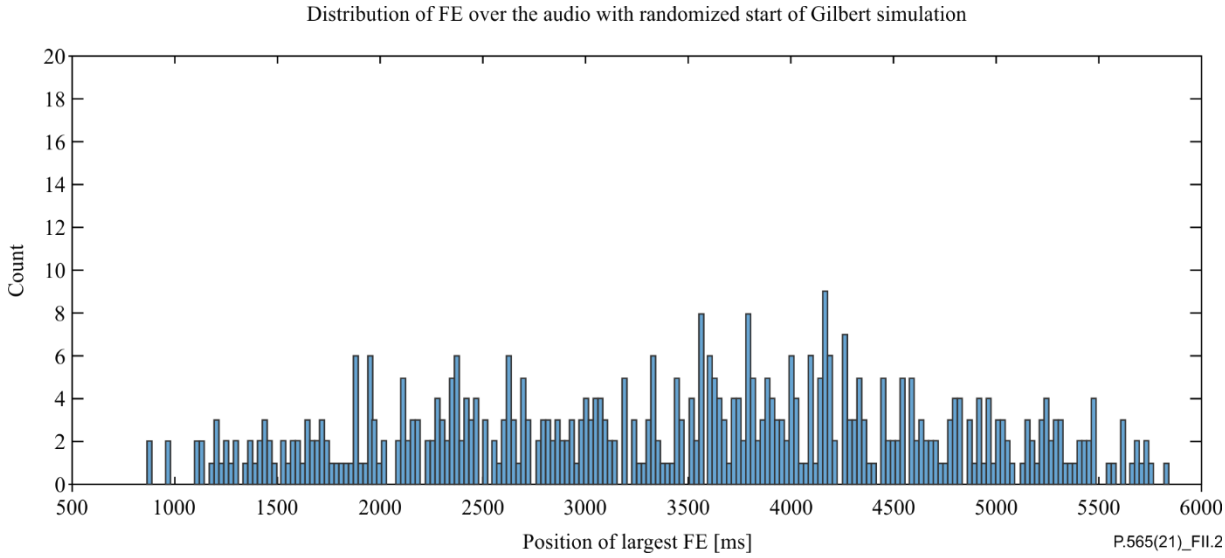


Figure II.2 – Illustration of frame erasure (FE) due to late loss evenly distributed over the speech

Applying the generic jitter files generated by the Gilbert burst packet loss and burst jitter model on the EVS simulator shows that using losses of up to 30% for both packet loss and jitter gives a good distribution of the MOS values.

II.1.1.3 Gilbert severe burst jitter

Extensive research shows that machine learning has the most difficulties with severe jitter. A Gilbert simulated database with jitter up to 70% is created to improve the learning and verify the algorithm's prediction power in these circumstances.

Severe jitter values are common for OTT voice services, and thereby important to be represented in the generic jitter files databases.

II.1.1.4 Random packet loss and random jitter

Random jitter both positive and negative combined with random packet loss has been simulated. The random distribution is uniform and packet loss is simulated up to 50% and jitter is simulated up to delays of ±200 ms. This type of simulation will cause packets to arrive out of sequence order.

II.1.1.5 Manually designed test cases

An important packet loss scenario is when the device goes in and out of coverage. This will lead to long consecutive packet losses. Manual test cases for consecutive loss of 1, 2, 3, 4, 5, 6, 8, 16, 32, 64 and 128 packets are designed. Also, combinations of two consecutive packet losses are designed. For every test case, the loss positions are shifted to 20 different positions.

II.1.2 Unknown validation live data sets description

The live validation generic jitter files have been created from four live data sets and their content is described below:

– Drive test in suburban and rural areas using Sony device G8141 and EVS 24.4. 40 MOS values exist between MOS 1 and 2 so the corresponding jitter files were selected. 40 MOS values from the other MOS groups (2-3, 3-4, 4-4.8) were then randomly selected and their corresponding generic jitter file added to the database. Total 153 jitter files.

– Drive test in suburban and rural areas using Samsung device 935F and EVS 24.4. 11 MOS values exist between MOS 1 and 2 so the corresponding jitter files were selected. 11 MOS values from the other MOS groups (2-3, 3-4, 4-4.8) were then randomly selected and their corresponding generic jitter files added to the database. Total 41 jitter files.
– Drive test in urban and suburban areas using Samsung device 950U and EVS 9.60. Drive test aimed to capture a large number of handovers. All MOS with handover in the speech was selected. Total 52 generic jitter files.
– Damper test using Sony device XP8131 and EVS 9.60. A damper in the "radio shielded room" was used to get poor quality. To get the worst values without losing the call the damper was completely turned down one to two seconds during speech. All 270 jitter files were used.

This database consists of 514 generic jitter files.

II.2 Unknown independent validation live data set description (source Rohde&Schwarz)

For the purpose of building the independent validation database, the RTP information of 20 000 audio samples in IMS VoLTE (EVS, AMRWB) and OTT/WhatsApp calls has been logged in during drive testing scenarios. The time frame for data collection was September-October 2020, the data was collected in three different networks of two different countries (Switzerland and Italy) also ensuring different infrastructure vendors. The dedicated audio sample to be used for transmission is the one provided in this Recommendation (Annex D). All presented scores are based on a single speech file in American English as defined by this Recommendation and can show deviations to other speech files such as in ITU-T P.501 Annex D or average scores such as in [ITU-T P.863.1].

To avoid any bias on the data due to devices, five different smartphone models have been used, all devices with native VoLTE support.

Phone model	Chipset type
Samsung Galaxy S9	Exynos
Samsung Galaxy S10	Exynos
Samsung Galaxy S20+	Exynos
Samsung A90	Qualcomm
Sony XZ2	Qualcomm

Each phone was used as sending and receiving side when doing the test calls. There have been also 'mixed' setups – calls between different devices – so that all combinations between chipsets were covered. VoLTE-VoLTE calls between a Samsung A90 (AMR-WB in VoLTE) and an EVS-capable phone used AMR-WB at both sides, but not EVS AMR-WB IO.

It must be noted that the Samsung A90 was selected intentionally, as it supports VoLTE but in the considered version it does not support EVS. This ensured RTP capturing using AMR-WB in VoLTE.

A validation database has been targeted that is balanced and covers all quality ranges equally as defined in this Recommendation (Annex B). From the total of around 20 000 collected samples, the vast majority was free of RTP loss and jitter and resulted in perfectly decoded voice samples.

Such a big number of initially collected samples was needed because it is very hard to capture bad VoLTE conditions with scores in the range of 1.0-2.0 MOS, without dropping into older technologies as 3G or 2G. Actually, all the bad samples collected in the dedicated drive tests were included in the validation database. Then, according to the available number of samples in the range of 1.0-2.0 MOS, a similar number of samples were randomly selected in the upper quality ranges (2.0-3.0 MOS, 3.0-4.0 MOS and 4.0-5.0 MOS). These scores needed for the quality-balanced

selection were obtained 'live' on the recorded wave signals by the R&S equipment during the data collection.

Finally, 532 VoLTE and 364 OTT/WhatsApp samples (RTP jitter files) have been selected, so that each set of samples presented a flat distribution over the MOS range as required in this Recommendation. Thus, a total of 836 recorded samples compose the R&S Validation Database.

The two histograms of Figure II.3 show the distribution of the ITU-T P.863 Edt. 3 FB scores for the VoLTE and the OTT/WhatsApp data that compose the independent validation database.

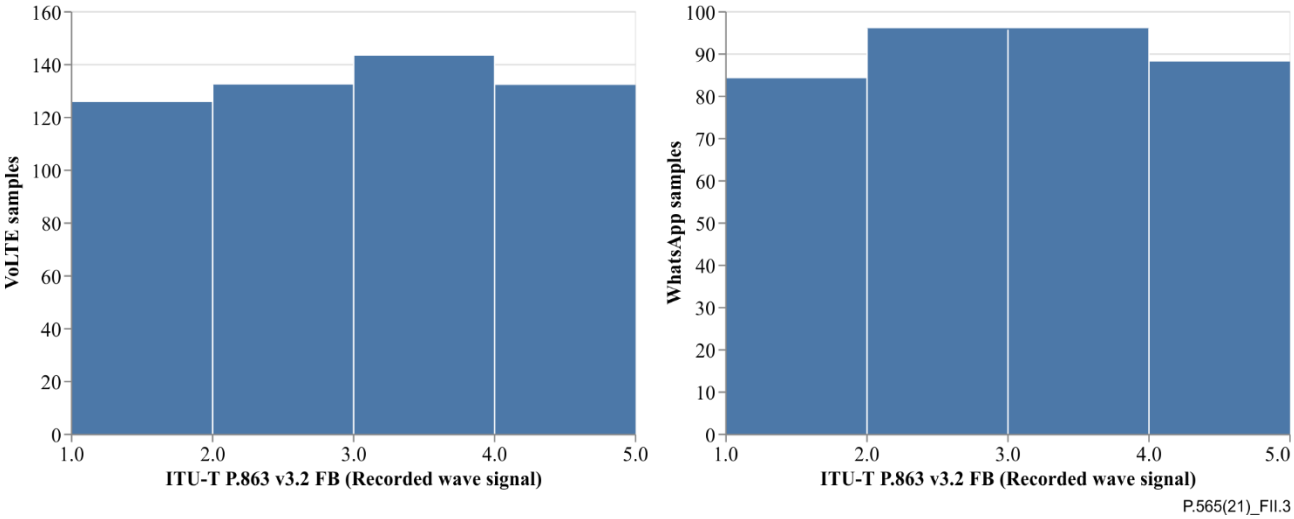


Figure II.3 – Histogram of the ITU-T P.863 Edt. 3 FB live scores on the R&S validation database for the VoLTE samples (left) and the OTT/WhatsApp samples (right)

The RTP streams corresponding to the selected audio samples were then cleaned and processed by R&S in order to create the generic jitter files according to the process described in this Recommendation.

Appendix III

Justification of the minimum requirements based on performance results' analysis

(This appendix does not form an integral part of this Recommendation.)

III.1 IMS mobile EVS use case

Generally, poor speech quality conditions are the hardest cases to predict by machine learning based algorithm, and therefore requiring more learning. Thus, the learning and validation generated databases shall contain a significant amount of severe network conditions. This is seen in the distribution of MOS/[ITU-T P.863] values for all used databases (Figure III.1); a total of 128908 jitter/MOS [ITU-T P.863] files have been simulated for each reference.

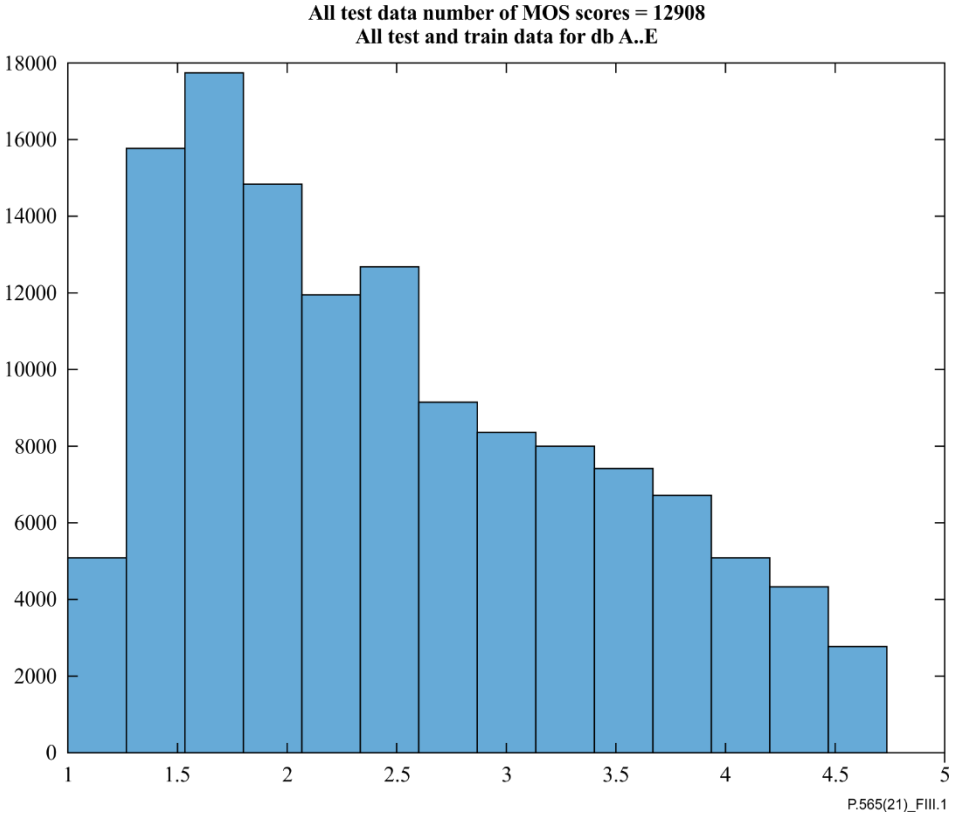


Figure III.1 – Distribution of all test MOS values

The performance results of an ML predictor developed according to the framework for the enhanced voice services (EVS) case are presented below. It should be noted that the performance results are calculated against [ITU-T P.863] scores as target reference values.

III.2 Results on learning and validation data sets

The total number of learning and validation data points for this example is 128908.

(i) [ITU-T P.1401] performance statistics

The results across all test data bases created based on the generic jitter files (clause 7.2; Annex D) and using reference speech – based features show a correlation coefficient of 96% and an RMSE of 0.25 MOS as well as a mean absolute value of 0.18 MOS (Figure III.2). These values are well within the range of ITU-T expected performance of speech/video QoE prediction models (e.g., [ITU-T P.863], [ITU-T P.120x], [b-ITU-T J.343.x], [ITU-T J.247] and [b-ITU-T J.246]).

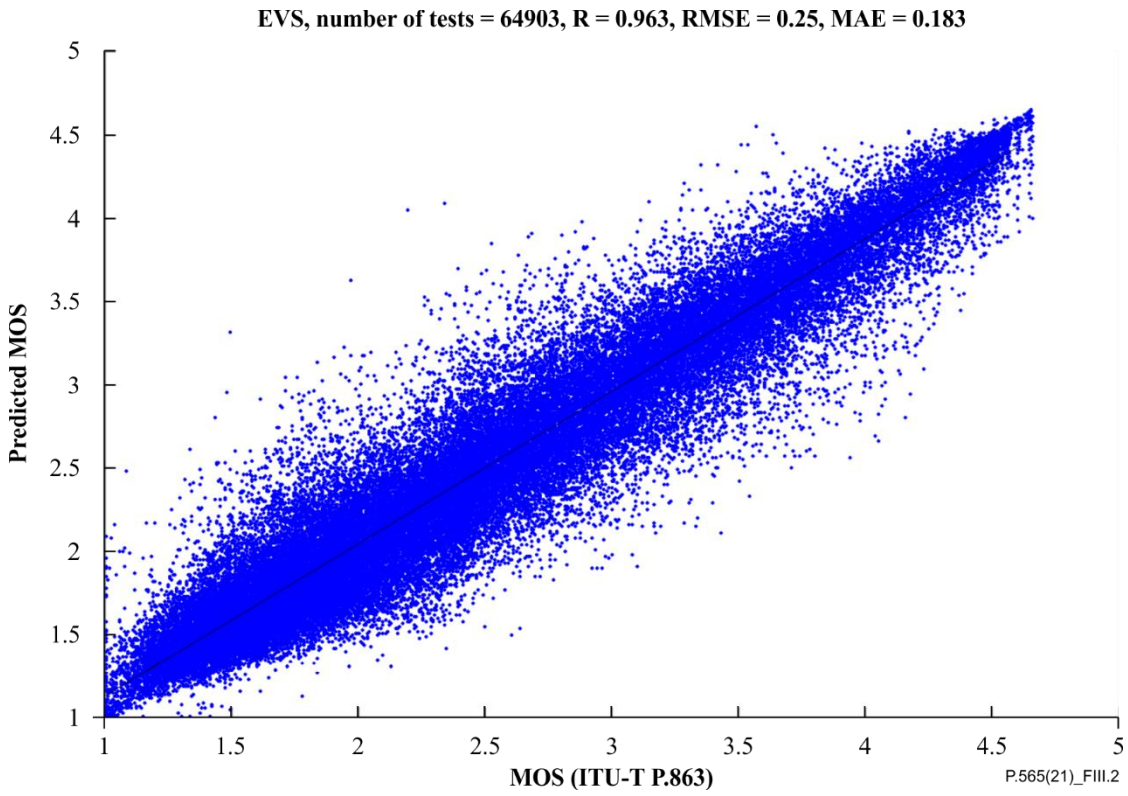


Figure III.2 – Performance of ITU-T P.1401 statistics results using reference speech-based features for validation simulated database

(ii) AE distribution

The AE distribution for the presented use case shows that 75.64% points satisfy an error less than 0.25 MOS, 94.82% points less than 0.5 MOS and 99.75% points less than 1 MOS (Figure III.3).

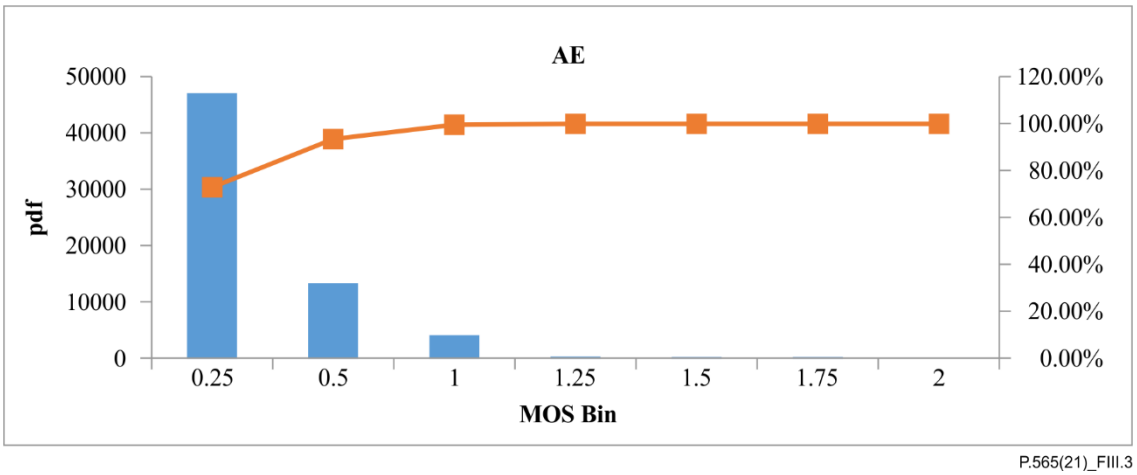


Figure III.3 – Distribution of AE for the validation simulated data set

(iii) Results for clean coding/decoding quality

This is tested on all databases with all rates up to 24.4 (see Table III.1) for a US English reference speech sample (clause 7.3, Annex D). Table III.1 shows that the differences for clean coding/decoding bit rates are zero at the 2nd decimal.

Table III.1 – Clean coding/decoding MOS and predicted MOS for EVS rates

Codec	Bit rates	Clean MOS (ITU-T P.863) (EVS FB/SWB/AMR-IO)	Clean predicted MOS	Difference
EVS FB / SWB/ AMR-IO (incl. CA mode) rates	5.90	na / 3.835 /na	3.835	0.00
	6.60	na/ na / 3.739	3.739	0.00
	7.20	na / 3.97 / na	3.97	0.00
	8.00	na/ 4.053/ na	4.053	0.00
	8.85	na/ na / 4.191	4.191	0.00
	9.60	na / 4.428 / na	4.428	0.00
	12.65	na / na / 4.413	4.413	0.00
	13.20	na / 4.423 / na	4.423	0.00
	16.4	4.639 /4.606 /na	4.639/4.606	0.00
	23.85	na/ na / 4.467	4.467	0.00
	24.40	4.657 / 4.65 / na	4.657/4.65	0.00

(iv) Results for poor network quality (high FER values).

The trend for medium to bad frame erasure (high FER values) is tested. The results for both AMR IO (interoperability) and EVS rates are presented in Figure III.4. It can be seen that MOS and predicted MOS are very close for FER < 10% and almost the same for FER > 10%. The same results are achieved for the channel aware mode (Figure III.5). This shows the good discrimination power of the predicted MOS (MOS(ml) according to this framework) in highly degraded conditions.

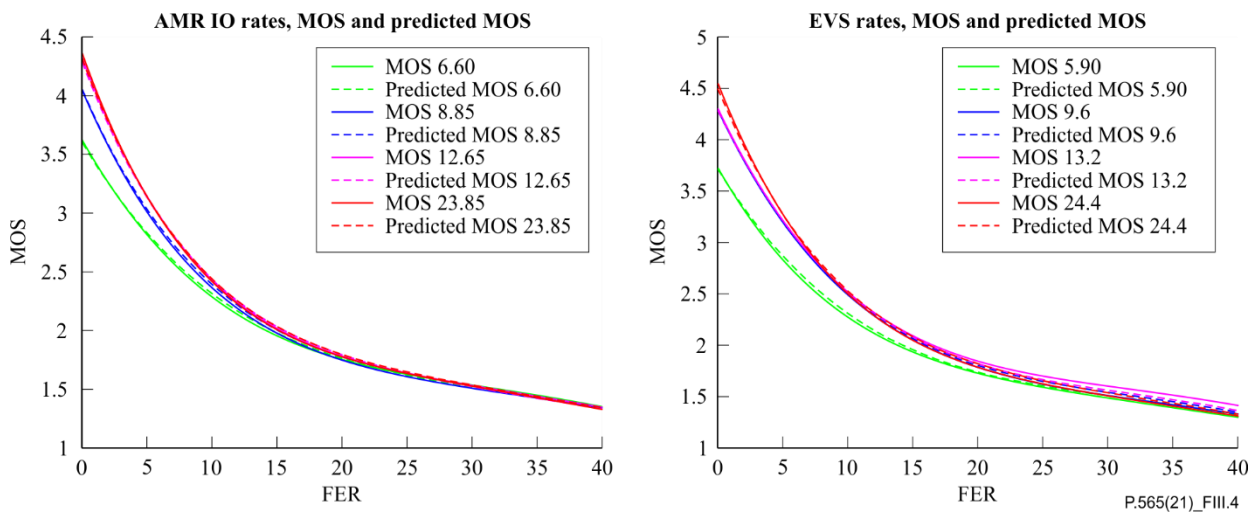


Figure III.4 – MOS/[ITU-T P.863] vs MOS from predicted MOS (MOS(ml), ML based predictor according to the framework) on low to high FER

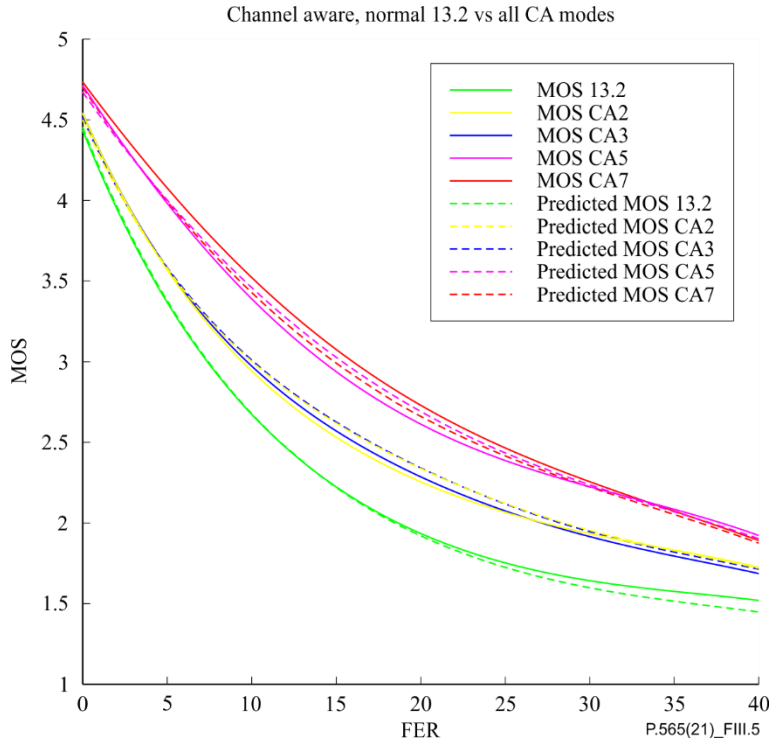


Figure III.5 – MOS/[ITU-T P.863] vs MOS from predicted MOS (MOS(ml), ML based predictor according to the framework) on all CA modes

III.3 Results on unknown validation live data

Live datasets contain more complex conditions and it shall be expected that performance values show weaker values than for the simulated dataset. However, differences shall not be statistically significant.

(i) [ITU-T P.1401] performance statistics

Figure III.6 shows results of a correlation coefficient (0.977), a RMSE (0.254), and a MAE (0.164) that are within the same performance values as seen on the validation simulated dataset above.

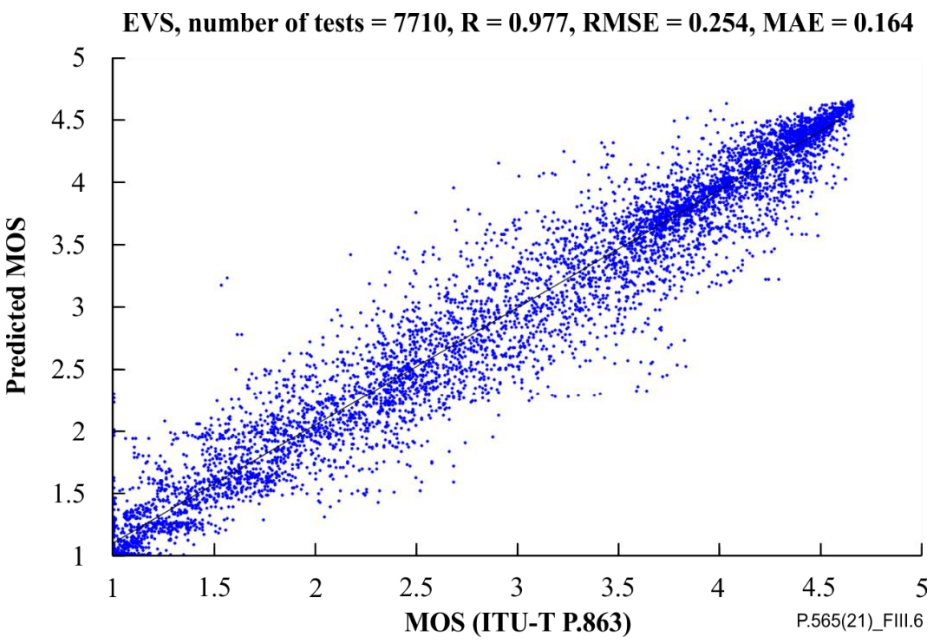


Figure III.6 – ITU-T P.1401 performance statistics for validation live data set

(ii) AE distributions

It can be seen (Figure III.7) that the ML based predictor (MOS (ml)) shows statistically the same performance with [ITU-T P.863], for the validation live dataset as it achieved for the validation simulated data set. This is reflected in the MAE distribution, with 76.04% points below 0.25 MOS, 93.02% below 0.5 MOS and 99.45% below 1 MOS. Although a slight spread in error (MOS bins between 1-1.75 MOS) is noticed, overall the compared speech quality solutions (MOS (ml) and ITU-T P.863 MOS) show statistically the same performance on the validation live data set, as in the case of the validation simulated data set above.

Therefore, the results on the validation live data set show that ML based predictor exhibits consistent performance on both validation simulated and unknown live data sets. Thus, it can be inferred that the used ML algorithm has not been overfitted/underfitted on the test/learning database.

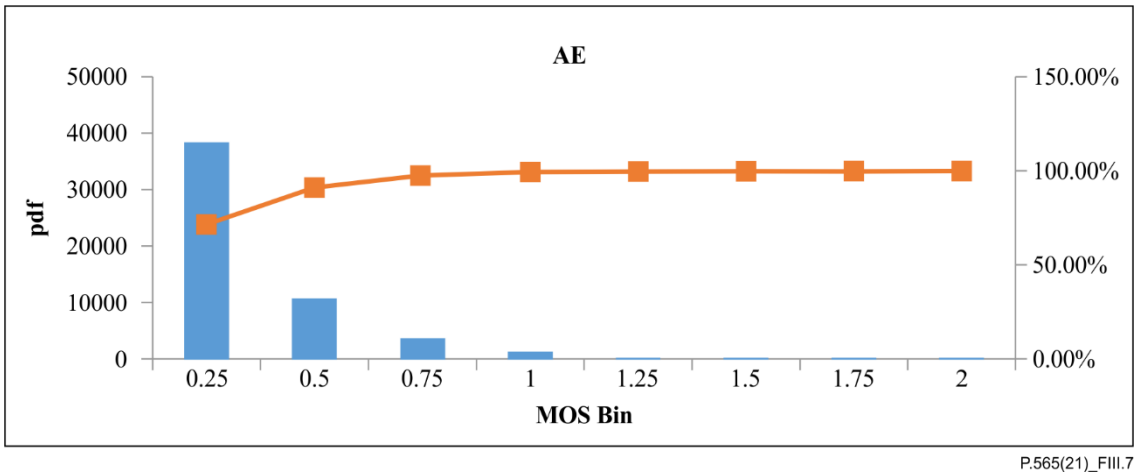


Figure III.7 – Distribution of AE for validation live data set

(iii) Results for the best quality are the same as above since the unknown live datasets uses the EVS codec as well.

(iv) Results for poor network quality (high FER values) are the same as above since the unknown live databases uses the same EVS codec.

III.4 ML overfitting/underfitting test

ML based solutions are sensitive to overfitting/underfitting scenarios which can be caused by the possibility that the training/learning data sets did not embed the entropy well enough when describing various network conditions. There are two techniques to verify if the developed ML based solution is free of the overfitting/underfitting effect.

First is the test on an unknown live data set as described in clause 10 (paragraph iv). The results presented above show that it can be inferred that the ML based speech quality predictor is under/overfitting free.

Second is a robust ML based method called the overfitting/underfitting test which determines the learning curve. Details on this method are presented in Annex A.

The ML overfitting/underfitting test method is applied for the EVS use case in order to ensure that the achieved performance values are reliable.

A split 80/20 split (131156/32790 samples) is used in order to identify the worst case, large learning set and smaller validation set. In addition, the learning curve is plotted with additional five-fold cross-validation, meaning that the validation data has been randomly selected five times from all the

data points. This ensures a good entropy of the validation data set and verifies the unbiased and no bound to the learning set. The results are presented in Figure III.8.

Figure III.8 shows that the used ML algorithm has a very good MSE on the learning set and at the same time the MSE for the validation data set converges fast to the one for the learning database, for a learning set size larger than about 70000 samples. Therefore, the learning curve shows that the error (MSE) is acceptable with the 65000 samples learning set used for the presented use case. In addition, the trend of the validation error does not deviate a lot and steadily decreases which proves that having larger data sets cannot decrease the error significantly.

Also, comparing the results with the ones for the two scenarios, overfitting and underfitting, it can be concluded that the used ML algorithm for speech quality prediction has been neither overfitted nor underfitted. Both are also proved by the results achieved on the unknown live data set, as shown above in this clause.

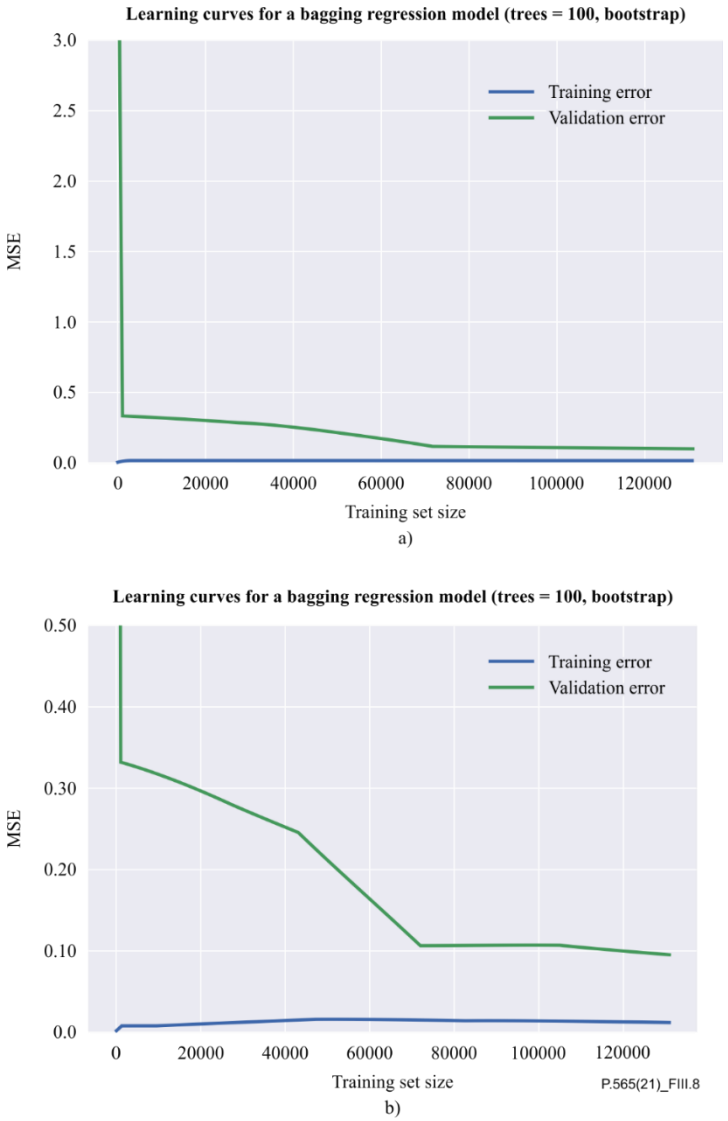


Figure III.8 – Overfitting/underfitting results (based on technique described in Annex A); (a) on large MSE scale; (b) zoomed in on up to 0.5MSE scale

Bibliography

[b-ITU-T J.246] Recommendation ITU-T J.246 (2008), *Perceptual visual quality measurement techniques for multimedia services over digital cable television networks in the presence of a reduced bandwidth reference*.

[b-ITU-T J.343.x] Recommendations ITU-T J.343 and ITU-T J.343.1 to ITU-T J.343.6 series.

[b-3GPP TS 26.448] 3GPP TS 26.448 (2019), 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; *Codec for Enhanced Voice Services (EVS); Jitter Buffer Management*.

[b-Friedman, et al] Jerome Friedman, Trevor Hastie, and Robert Tibshirani (2001), *The elements of statistical learning*. Vol. 1. 10. Springer series in statistics New York, NY, USA.

[b-Huang, et al] Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2006), *Extreme learning machine: theory and applications*. Neurocomputing, 70(1-3), 489-501.

[b-Jiménez] Rafael Zequeira Jiménez, Anna Llagostera, Babak Naderi, Sebastian Möller, Jens Berger. *Modeling Worker Performance Based on Intra-rater Reliability in Crowdsourcing A Case Study of Speech Quality Assessment*.

[b-Mathwork] Mathwork, *Introducing Machine Learning*. eBook.

[b-van Rijn et al] Jan N van Rijn et al. (2015), *Fast algorithm selection using learning curves*. International symposium on intelligent data analysis. Springer, pp. 298–309.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	**Telephone transmission quality, telephone installations, local line networks**
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2022