IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 31, 2017, accepted May 12, 2017, date of publication May 25, 2017, date of current version August 29, 2017.

*Digital Object Identifier 10.1109/ACCESS.2017.2705428*

# On Losses, Pauses, Jumps, and the Wideband E-Model

**MUHAMMAD ADIL RAJA[1], ANNA JAGODZINSKA[2], AND VINCENT BARRIAC[2]**
[1]Namal College, Mianwali 42250, Pakistan
[2] Orange Labs, 22300 Lannion, France

Corresponding author: Muhammad Adil Raja (adil.raja@namal.edu.pk)

**ABSTRACT** There is an increasing interest in upgrading the E-Model, a parametric tool for speech quality estimation, to the wideband and super-wideband contexts. The main motivation behind this has been to quantify the quality gain lent by various new codecs and communication situations. There have been numerous such contributions, and all of them have been more or less successful. This paper reports on an extension of the E-Model to the mixed narrowband/wideband (NB/WB) context. More specifically, we take a novel approach toward deriving effective equipment impairment factors ($I_{e,WB,eff}$) by considering additional impairments related to the underlying communications network. These additional impairments are the pause and jump temporal discontinuities along with network-related loss and pure codec-related impairments. While the effect of loss is a thoroughly studied topic and has been integrated into the E-Model, pauses and jumps have been given little attention. Pauses and jumps manifest themselves as temporal dilation and contraction, respectively, in the resulting speech signal that is presented to the listener and are normally caused by jitter and jitter buffer interaction. In this paper, we initially present a four-state Markov model to characterize, and also emulate, loss, pause, and jump impairments. Then, we present alternative models for computing effective equipment impairment models. A large number of test stimuli were generated using different NB and WB codecs. WB-PESQ was used to evaluate the stimuli. Genetic programming was employed to derive equipment impairment factors. The proposed models have a high correlation with WB-PESQ. We claim that the models proposed by us outperform the existing E-Model by a factor of approximately 29% while using WB-PESQ as a reference model. The models also outperform the E-Model against results from auditory tests. It is also shown that the models outperform the results of multiple linear regressions.

**INDEX TERMS** Loss, pause, jump, GP, WB-PESQ.

## I. INTRODUCTION

Telecommunications technologies are evolving at a rapid pace. The old Public Switched Telephone Network (PSTN), which normally operates in the narrowband (NB) range (300 – 3400 Hz), is being replaced with wireless and voice over IP (VoIP) systems. A good feature of VoIP, among others, is that it allows the transition to wideband (WB) telephony (50–7000 Hz) by a simple change of codecs. A nice thing about WB telephony is that it offers more natural sounding speech as opposed to NB. The advent of WB telephony gives rise to the need for having appropriate speech quality assessment tools. Assessment tools allow the transmission planners and network service providers to assess the quality of service offered by their networks.

VoIP quality is affected by various factors which include packet loss, jitter, delay and impairments related to codecs. Numerous algorithms and tools have been developed to evaluate the quality of VoIP. The most prominent among these is ITU-T Recommendation G.107 [1], commonly known as the E-Model. The E-Model is an instrumental model that was initially designed for transmission planning purposes. It ensues from an impairment factor principle that assumes that degradations induced by various sources have a cumulative effect on speech quality and that they may accordingly be transformed to a *transmission rating scale (R scale)*. The E-Model was originally intended for NB speech quality estimation. In the recent past, Möller *et al.* [2] upstaged it to the mixed NB/WB context by using subjective *listening only* tests [3] for a mixture of various NB and WB codecs recommended by ITU-T. Their main emphasis was on deriving *effective equipment impairment factors* ($I_{e,WB,eff}$), in a mixed NB/WB context, that represent the degradation in the *listening quality*

due to both codec and loss related distortions. They suggested a quality advantage of 29% relative to the pure NB context. Raja *et al.* [4] presented a methodology for deriving $I_{e,WB,eff}$ for a mixed NB/WB context. Their approach was based on ITU-T P.834 [5] in which they employed WB-PESQ [6] as a reference instrumental model and GP to derive the functional form of $I_{e,WB,eff}$. More recently, Wältermann *et al.* [7], extended the E-Model to the *super wideband (SWB)* scale, suggesting a quality advantage of 39% relative to WB and 79% relative to NB.

In this paper, we take a novel perspective towards deriving the $I_{e,WB,eff}$ by incorporating additional known, but rarely addressed, jump and pause impairments. We use WB-PESQ as a reference instrumental model and we employ genetic programming (GP) to derive the functional form of $I_{e,WB,eff}$.

Rest of the paper is organized as follows. Section II gives an introduction to the E-Model. Section III elaborates on loss, pause and jump temporal discontinuities in the light of [8]. Section IV discusses the VoIP network simulators that we prepared and used in our work to learn the behavior of a typical VoIP network, along with a jitter buffer, and its effect on the consequent loss/pause/jump impaired packet stream. Section V describes our methodology. Section VI describes the various VoIP simulations carried out in this work to generate test stimuli. Section VII gives a brief introduction to Genetic Programming (GP). Section VIII discusses various GP experiments performed in this research. Section IX highlights the important results and presents various models. Conclusions and future goals are given in section X.

## II. THE E-MODEL

The E-Model is a computational model used to predict the combined effect of various impairments on speech quality for a conversational scenario. It is defined by ITU-T G.107 [1]. It was initially designed for NB handset telephony, however, its adaptation to WB and SWB scenarios is in fast progress [2], [4], [7]. The output of the model is a *rating factor*, $R$. While computing $R$ it is assumed that factors affecting speech quality are additive in nature [9]. $R$ is computed according to equation (1):

$$R = R_0 - I_s - I_d - I_{e,eff} + A \tag{1}$$

where $R$ is called the transmission rating factor and it ranges from 0 (poor quality) to 100 (optimum quality) for the NB case. $R_0$ is the basic signal to noise ratio which, for the NB case, defaults to 93.2. $I_s$ represents all the impairments which occur simultaneously with the voice including, for instance, overall loudness rating and non-optimum sidetone. $I_d$ marks the effect of delay related impairments such as echo and too long end-to-end delay that may affect the call quality in a conversational sense. $I_{e,eff}$ depicts the impairments due to low bit-rate codecs in the presence of packet loss. Finally, $A$ is the advantage factor that compensates for the above impairment factors when there are other advantages of access to the user depending on the nature of the underlying network. Thus, for instance, $A$ may be assigned a value of 0 for a wired network and 20 for a multi-hop satellite connection. $A$ is seldom used in reality. In the case where values of one or more of these factors may not be determined, default values are used from [1].

$R$ and *Mean Opinion Score (MOS)* are inter-convertible using corresponding transformations given in [1]. These transformations are referred to by (2).

$$R \Longleftrightarrow MOS \tag{2}$$

where $MOS$ varies between 1 (bad) to 4.5 (excellent), and it is a measure of how humans perceive speech quality. It must be mentioned that the relationship presented in equation 2 is presented in a shortened form for symbolic purposes. The exact relationship is a rather long equation that has been skipped here for the purpose of brevity. The exact relationship can be found in [1].

The above formulations hold for the case of NB codecs. Möller *et al.* [2] proposed an extension of the R scale of 29% from the NB case ($R_{NB}$) to the mixed NB/WB case ($R_{NB/WB}$) based on subjective tests performed in [10]. This extension is given by equation (3).

$$R_{NB/WB} = 1.29 \times R_{NB} \tag{3}$$

$R_{NB}$ can be calculated via (2). This extension is now an integral part of the E-Model (see Appendix II of ITU-T G.107 [11] and [12]), where the new default value for $R_0$ for the NB/WB case is 129. Following this, $I_{e,WB}$ (i.e. impairments solely due to NB/WB codecs) can be calculated according to equation (4).

$$I_{e,WB} = 129 - R_{codec} \tag{4}$$

where $R_{codec}$ may be calculated from (3) and 129 corresponds to the value of $R$ for the direct channel for the mixed NB/WB context. The direct channel in this context is represented by a 16-bit linear PCM with $f_s$=16 kHz (assuming that other impairments related to echo or delay are not present).

As suggested earlier this extension of the R-scale to the *NB/WB* case is based on auditory tests. Although a similar extension relative to an instrumental model, such as WB-PESQ [6], is available in the literature [13], we considered it incumbent upon us to attempt to derive this extension ourselves. Raja *et al.* [4] made a futile attempt to derive such an extension. Instead, the methodology discussed in ITU-T Recommendation P.834.1 [14] gives an outline on as to how results of instrumental models can be converted to $R_{WB}$. It makes use of (2) and (3).

## III. ON LOSSES, PAUSES, AND JUMPS

As discussed earlier, a number of attempts at extending the E-Model [1] to the mixed *NB/WB* scenario have been made. In most cases, $I_{e,WB,eff}$ have been derived in listening only contexts that predict the degradation in quality due to codec related distortions and packet loss related metrics. A much less studied topic is the effect of jitter on speech quality. In particular, a suitable formulation of $I_{e,WB,eff}$ in a mixed

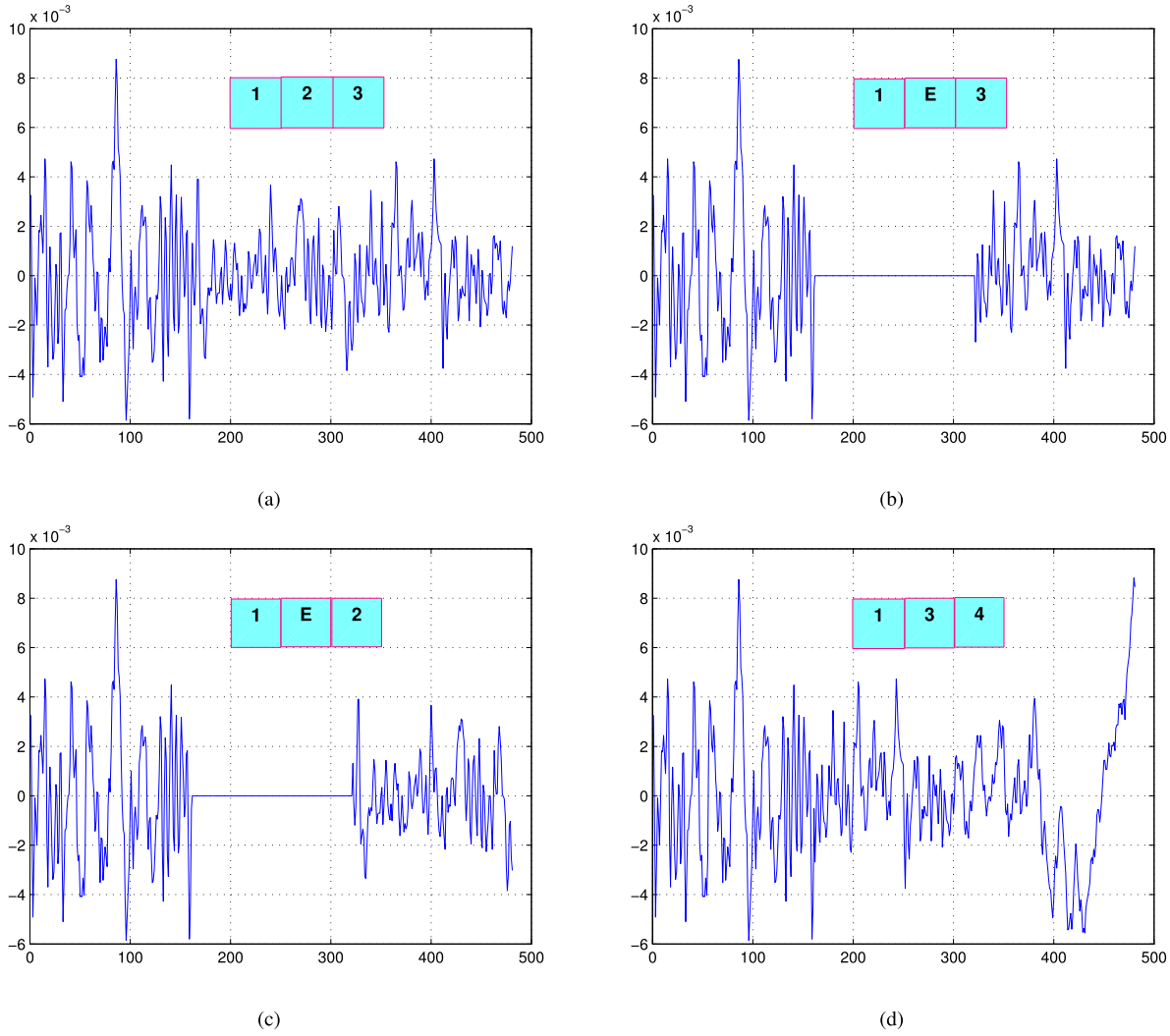



**FIGURE 1.** Waveform corresponding to three frames: (a) Without Impairment. (b) With a Loss, (c) With a Pause and (d) With a Jump. The numbered blocks show that the $i^{th}$ frame was received. E depicts an empty slot.

*NB*/*WB* is missing in the literature for the case of jitter. Normally it has been assumed that excessive jitter leads to packet loss, and thus, the effect of packet loss is taken into account only[1] [15].

Packet loss can be characterized by a signal outage for a certain period and some speech lost for that period (see Fig. 1b). Packet loss normally happens due to link failures and congestion at the intermediate nodes of the network. Network jitter results in two additional temporal discontinuities that may happen due to the inability of the jitter buffer to completely remove jitter. These discontinuities are referred to as pauses and jumps in the existing literature [8]. A pause discontinuity is characterized by a signal outage for a certain period of time. Speech is not lost as a result of a pause. (see Fig. 1c). Pauses happen in the periods when network jitter varies to acquire a large value to an extent that the jitter buffer becomes empty, and consequently does not have anything to play to the decoder for a certain period of time. In other words, pauses happen during the period when the jitter buffer under runs. A jump discontinuity is characterized by some speech loss but no signal outage (see Fig. 1d). Jumps happen in periods when the network jitter varies to acquire a very small value to an extent that packets arrive in the jitter buffer immediately one after the other and the packets have to be dropped from the jitter buffer due to a lack of a holding capacity therein. In other words, jumps happen in periods when the jitter buffer is overrun by the arriving packets. Jumps can alternatively also happen when the network jitter is low and one or more packets are lost in the network.

To the best of our knowledge, the terminology concerning *pauses* and *jumps* has only been used (and introduced) by Voran [8]. These rather less studied notions have, nonetheless, been brought to attention in the existing literature. For instance, in [16] it has been shown that a sequence of *negative*

[1]Excessive jitter also leads to extra delay. However, since the object of this work is not to address the effect of delay, this matter shall not be discussed further in this paper.

*jitters* (clustering) can result in downstream nodal congestion and consecutive packet loss (leading to jumps). Similarly, a sequence of *positive jitters* (dispersion) can result in consecutive packets experiencing excessive delays (leading to pauses). To this end, it is also apparent that, like packet loss, pauses and jumps are also bursty in nature.

In [8] Voran studied the effect of these three impairments (i.e. losses, pauses and jumps) on speech quality using the ITU-T G.723.1 codec [17]. He used subjective tests [3] to derive a conclusion that the overall effect of each of these sources of impairments is not different from the other.

Notwithstanding this, the scope of current research is to derive a formulation for $I_{e,WB,eff}$ that takes into account the individual and/or combined effect of each of these impairments. To this end, the goal would be to figure out the effect of each of these impairments on $I_{e,WB,eff}$ and also to find out a working formula for the latter which is a function of the significant impairments.

## IV. VoIP SIMULATION

The aim of this study is to develop a model for $I_{e,WB,eff}$ as a function of losses, pauses and jumps. This requires a thorough insight into the behavior of a VoIP network and the behavior of the jitter buffer. It has to be known as to how lossy and jitter-trodden VoIP packet stream interacts with the jitter buffer. Consequently, the knowledge of this can be used to understand as to how the speech frames get played to the user. In other words, knowledge of this can be used to understand the pattern of losses, pauses and jumps and their distribution.

In order to achieve this, a VoIP simulation environment was developed as a UDP client/server application. In this simulation software, the sender prepares packets with a payload corresponding to a 20 ms G.711 [18] frame and sends it to the receiver. The packets are also assigned sequence numbers. While the packets are being sent they are subjected to a jitter model. Internet packet end-to-end delay and packet-to-packet delay variation (jitter) are self-similar phenomena. This means that delay patterns are invariant of the time-scales at which they are observed. Packet jitter is normally modeled using a heavy-tailed statistical distribution. Most notable among these are Pareto, Weibull and exponential distributions. Although Pareto distribution has been more widely used in the past to model the behavior of Internet packet delay [19], the focus of some recent research has been on using Weibull distribution to model jitter in VoIP [15]. In this work Weibull distribution has been chosen to model jitter. Weibull distribution can be characterized using a location parameter, a shape parameter and a scale parameter. The location parameter determines the minimum value of a random variable. While modeling the Internet delay it can be used to set the minimum value of the end-to-end delay.

A value of zero was used for the location parameter in this work since the goal was to model the end-to-end delay variation (jitter) only. The shape and scale parameters, as the names suggest, determine the shape and spread of a statistical distribution. Values of 2.0 and 24 were chosen for shape and scale parameters to give a mean jitter value of, approximately, 20 ms assuming that a typical VoIP frame is of this duration.

It is important to briefly reflect on end-to-end delay characteristics and jitter buffer implementations of common VoIP applications. Wu *et al.* [20] comment on this subject quite comprehensively. According to their findings average jitter of various VoIP applications can reach up to 250 ms. Playout buffer sizes for commonly used applications such as Skype, Google Talk and MSN Messenger have been reported. For peak delay jitter values, reaching up to 250 ms, delay buffer sizes ranging up to 800 ms have been employed by various applications. They derive optimal buffer sizes as a function of delay jitter. For delay jitters up to 200 ms, the optimal size for jitter buffer reaches up to 800 ms, as proposed by them. However, interactivity decreases as the sizes of jitter buffers increase. According to their results, sizes for adaptive jitter buffers typically range between 100–400 ms. Similar results are reported in [21]–[23]. In our work, we have employed a static, fixed-sized, jitter buffer. Its details are given later in this section.

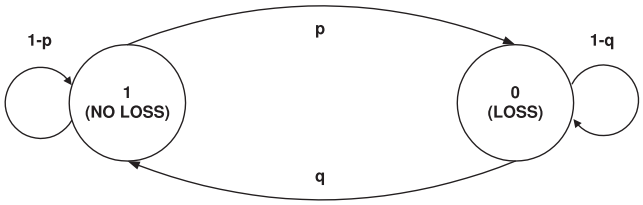



**FIGURE 2.** Two State Gilbert Model.

After passing through the jitter model the packets pass through a 2-state loss model (aka the Gilbert model) [24], which inflicts losses of a user specified intensity on the frames. The loss intensity can be specified in terms of a *target mean loss rate (mlr)*, henceforth abbreviated as *mlr_target* and a *target conditional loss probability (clp)*, henceforth abbreviated as the *clp_target*. The latter variable is used to model burstiness in packet loss. The two state Gilbert model can be seen in Fig. 2. It is worth mentioning here as to what the various probabilities mean and how they can relate to *mlr*. In this model, $p$ is the probability of transitioning from the no-loss state to the loss state, and $q$ is for the converse. $1 - q$ is the *clp* and is related to burstiness of the loss. Equations (5) and (6) show how *mlr* and *mbl_loss* (mean burst length of the loss) are related to $p$ and $q$. State transition probabilities of the model can be learned from a network trace analysis, or conversely, a suitable network trace can be generated using this model by providing values for these state probabilities using desired values of *mlr* and *mbl*. It was important to discuss this model here because latter in this section a 4-state loss, pause and jump model is derived using this.

$$mlr = \frac{p}{p+q} \tag{5}$$

$$mbl\_loss = \frac{1}{q} \tag{6}$$

The receiver waits for the packets to arrive and places them in a jitter buffer. A very simple, static, jitter buffer is implemented that is assumed to play out the packets to the decoder after every 20 ms. The buffer starts playing out frames as soon as it has a frame in it i.e. it does not wait for itself to get full (even beyond a certain limit). The jitter buffer can hold five frames of 20 ms each and, thus, has a size of 100 ms. An important thing that the jitter buffer does is to decipher a sequence of losses, pauses and jumps from the packets as they arrive.

**TABLE 1. VoIP simulation results - observed values for various network traffic parameters (%).**

| Variable | Simulation 1 | Simulation 2 |
|---|---|---|
| mlr | 0.09 | 0.024 |
| mpr | 0.11 | 0.014 |
| mjr | 0.13 | 0.104 |
| mbl_loss | 2.23 | 1.89 |
| mbl_pause | 1.71 | 1.1 |
| mbl_jump | 1.387 | 1.15 |

Two simulations were done with various values of *mlr_target* and *clp_target*. To be more precise, *mlr_target* was varied between [0, 0.002, 0.004, . . . , 0.1] and [0, 0.005, 0.01, . . . , 0.4] respectively for simulation 1 and simulation 2. This resulted in *mean* values of 0.017 and 0.02 respectively for *mlr_target* for both simulations. *clp_target* was varied between [0, 0.1, . . . , 0.8]. A Weibull distributed packet jitter was induced with a shape parameter equal to 1 in the first simulation and equal to 2 in the second simulation. A total of approximately 1,000,000 packets were generated. Various statistics related to these simulations can be seen in Table 1. It can be seen that the observed values for *mean jump rate (mjr)* and *mean pause rate (mpr)* are rather high for the first simulation (0.13 and 0.11 respectively). However, for the second simulation, the observed value of *mpr* was significantly less, 0.014. Whereas, there was only a marginal decrease in the observed value of *mjr*. Similarly, the overall observed *mlr* was 0.09 and 0.02 respectively. These values of *mlr* correspond to the average over the entire lengths of respective simulations.

The table also shows values for three additional variables, namely, *mbl_loss*, *mbl_pause* and *mbl_jump*, representing *mean burst lengths* of *losses, pauses* and *jumps* respectively. It is shown that the *mean burst lengths* vary between 1 – 2 for both simulations. It is hard to predict if these values correlate well with what may actually happen in a real VoIP network as these ensue from a simulation study. Moreover, the existing literature on Internet traffic analysis does not have quite valuable statistics related to pauses and jumps.

As stated earlier, a 4-state no-loss, loss, pause and jump model can be learned from these simulations. This can be done by extending the simple 2-state Gilbert loss model, as discussed in [24]. The model with its various state transition probabilities is shown in Fig. 3. The model gives us an idea on as to what sort of state transitions can occur in a

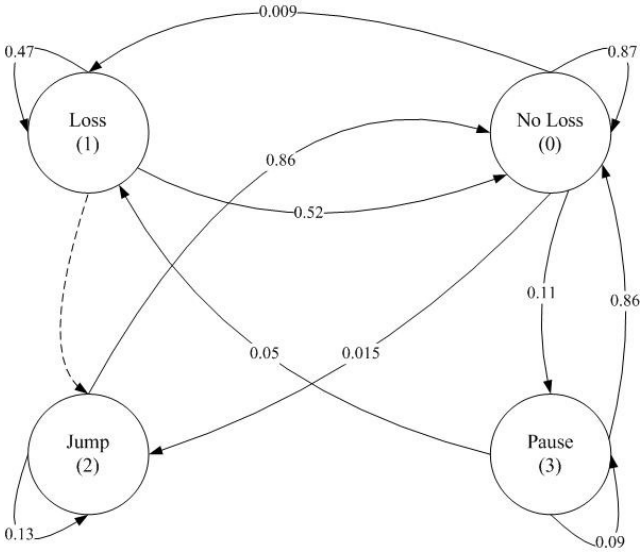



**FIGURE 3. 4-State Loss Pause and Jump State Model.**

pattern of losses, pauses and jumps, and also what could be the intensity of each state transition. For instance, it tells us that there is a significantly higher probability of staying in the no-loss state when the system is already in that state as opposed to moving in any of the impairment related states. In other words, it suggests that there is significantly higher probability of receiving a packet successfully in the $N^{th}$ time slot, as opposed to loosing it as a loss or a jump or having it delayed as a pause, given the packet in state $(N-1)^{th}$ time slot was also successfully received as well. The model also suggests the presence of somewhat logically less obvious edges. For instance, in the model there is an edge going from the pause to the loss state, however with a low transition probability of 0.05. This, however, suggests that a loss event can happen immediately after the pause event. Similarly, an absence of an edge going from the loss to the pause state suggests the lack of a possibility of occurrence of such an event. This is also intuitive because after $N$ loss events the system is normally supposed to restore to the no-loss state.

In this 4-state model as well, just as in the case of the 2-state Gilbert loss model [24], the probability of looping in the loss state is termed as *clp*. On similar lines, we can call the probability of looping in the pause state as the *conditional pause probability (cpp)* and the probability of looping in the jump state as the *conditional jump probability (cjp)*.

Computing *mbl_loss* as the reciprocal of $(1 - clp)$ as in equation (7) is equivalent to computing it as the reciprocal of $q$ as in equation (6). However, the former formulation is more feasible for the 4-state model as there are more than one edges coming out of the loss state in this model, as opposed to only one edge coming out of the loss state in the 2-state Gilbert loss model. Similar formulations can be formed for computing *mbl_jump* and *mbl_pause*, these are given by

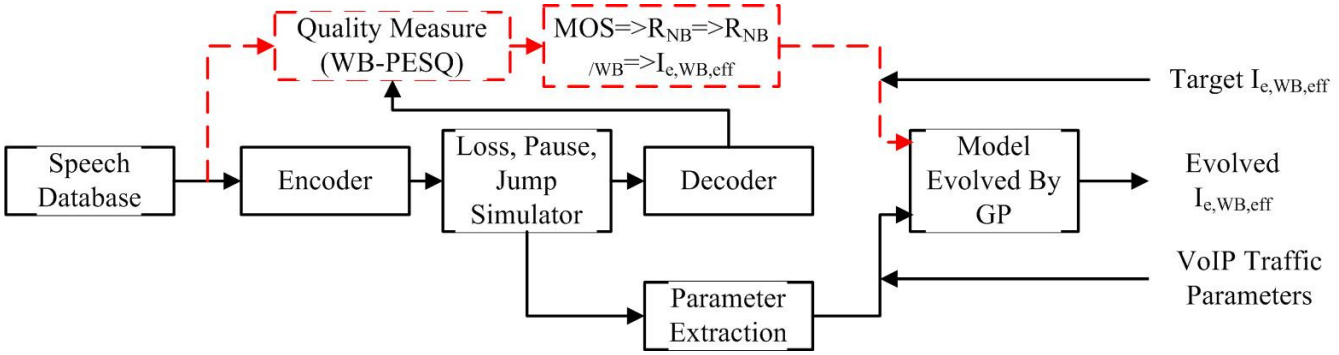



**FIGURE 4.** Simulation system for derivation of $I_{e,WB,eff}$.

equations (8) and (9) respectively.

$$mbl\_loss = \frac{1}{1 - clp} \tag{7}$$

$$mbl\_jump = \frac{1}{1 - cjp} \tag{8}$$

$$mbl\_pause = \frac{1}{1 - cpp} \tag{9}$$

According to [24] *mbl_loss* can be computed from network trace analysis by applying equations (10) and (7).

$$clp = \left(\sum_{i=2}^{m-1} b_i \times (i-1)\right) / \left(\sum_{i=1}^{m-1} b_i \times i\right) \tag{10}$$

where, $b_i$ is the number of loss bursts having length i. *mbl_pause* and *mbl_jump* can also be computed in a similar fashion with the only difference being that $b_i$s in those cases would represent respective number of burst lengths of pauses and losses respectively in equation (10) and instead of applying equation (7), equations (9) and/or (8) would have to be applied respectively.

Even though the 4-state model has been learned as a result of various VoIP simulations, it can be used for the converse too i.e. given a set of target network impairment rates, a pattern of *zeros', ones', twos'* and *threes'* (e.g. ...01212301 ...) can be generated (representing a sequence of no-losses, losses, jumps and pauses respectively) having more or less same level of impairments specified a priori. This is the crucial advantage of the 4-state model. The pseudo-code for the 4-state model can be seen in Algorithm 1. Reiterated, the algorithm works by providing values of *target_mlr*, *target_mjr*, *target_mpr*, *target_clp*, *target_cjp* and *target_cpp*. These values are then converted to various corresponding state transition probabilities. Once such a pattern has been generated, the values of the above parameters can be recomputed from it. These can be called as the observed values.

It is worth mentioning here the way in which the observed values for the aforementioned variables can be computed. Given a pattern of length $N$, containing 0s, 1s, 2s and 3s (no-losses, losses, jumps and pauses respectively), the observed values of *mlr* can be calculated according to equation (11). This gives a precise calculation of *mlr*, as it computes the fraction of packets lost as a function of total number of packets sent, which is equal to the sum of the number of packets received and the number of packets lost both as losses and jumps. The number of pauses are removed from the size of the pattern because they do not correspond to packets; they represent empty time slots in which the jitter buffer did not have anything to play to the decoder. Similarly, *mjr* and *mpr* can be computed precisely, and in a similar fashion, according to equations (12) and (13) respectively. *mpr*, in this case, represents the fraction of pauses as the total number of packets sent.

$$mlr = \frac{\sum losses}{pattern\_length - num\_pauses} \tag{11}$$

$$mjr = \frac{\sum jumps}{pattern\_length - num\_pauses} \tag{12}$$

$$mpr = \frac{\sum pauses}{pattern\_length - num\_pauses} \tag{13}$$

Algorithm 1 represents the pseudo-code for the 4-state model. It is worth describing a few abbreviations that are used in this model. n2l, n2p, n2j imply transition from no-loss to loss, pause and jump states respectively. Similarly, abbreviations j2n, p2n and l2n imply the converse, respectively. Meanings of other similar abbreviations can be inferred accordingly for various other state transitions.

## V. METHODOLOGY

Our methodology ensues from the work reported in [4]. The schematic in Fig. 4 depicts a conceptual diagram of our approach for deriving $I_{e,WB,eff}$ for VoIP. An initial requirement is to have a database consisting of clean speech signals. These signals are subjected to degradations typical of VoIP traffic; coding distortions and temporal discontinuities i.e. losses, pauses and jumps. Degraded speech stimuli are obtained in this way that are representative of typical VoIP streams. In the process of doing so the values of various VoIP traffic parameters, such as *mlr*, *mjr* etc., are calculated. The decoded speech stimuli are evaluated using a viable instrumental model that may report its results in

**Algorithm 1** The 4-State Model

```
state ← 0
for i = 1 : pattern_length() do
  trans_prob ← Random.number()
  if state = 0 then
    if trans_prob < n2l then
      state ← 1
    else if trans_prob ≥ n2l and trans_prob < (n2l + n2j) then
      state ← 2
    else if trans_prob ≥ (n2l + n2j) and trans_prob < (n2l + n2j + n2p) then
      state ← 3
    else
      state ← 0
    end if
  else if state = 1 then
    if trans_prob < l2n then
      state ← 0
    else if trans_prob ≥ l2n and trans_prob < (l2n+l2j) then
      state ← 2
    else
      state ← 1
    end if
  else if state = 2 then
    if trans_prob < j2n then
      state ← 0
    else
      state ← 2
    end if
  else if state = 3 then
    if trans_prob < p2n then
      state ← 0
    else if trans_prob ≥ p2n and trans_prob < (p2n + p2l) then
      state ← 1
    else
      state ← 3
    end if
  end if
  pattern ← pattern.append(state)
end for
```

terms of human assessment of speech quality i.e. MOS-LQO. Moreover, the model should be able to evaluate both *NB* and *WB* coded speech. An example of such a model is WB-PESQ, which has been used as a reference system in this research. It is worth mentioning that WB-PESQ has numerous well-known limitations that have been reported elsewhere in the literature [25, p. 105], [26], [27].

The resulting MOS-LQO is converted to $I_{e,WB,eff}$ using equations (2), (3) and eventually (4). We call this the *target* $I_{e,WB,eff}$. The process is repeated for a large number of speech stimuli with varying degrees of network distortion conditions. Once the target $I_{e,WB,eff}$ for all the speech signals have been computed and the values of corresponding VoIP network traffic parameters gathered, GP based evolution is performed to derive a suitable mapping. More specifically, the VoIP network traffic parameters serve as the input domain variables during evolution and the corresponding $I_{e,WB,eff}$ values form the *target* output values.

A linear interpolation between the $I_{e,WB}$ obtained by the instrumental model (WB-PESQ) and subjective tests may be performed as suggested by [5, p. 9] to adjust the target $I_{e,WB,eff}$. Previously an interpolation was performed between $I_{e,WB}$ values for 20 (14 WB and 6 NB) codecs using WB-PESQ, and from subjective tests reported in [28]. The interpolation results in stable values for $I_{e,WB,eff}$. Alternatively, one may use the *slope* and *intercept* terms resulting from an interpolation already performed and reported in [14] for a number of reference conditions. We performed an interpolation according to the scheme reported therein and the resulting slope and intercept values are reported in Table 2 in the row labeled *computed*. In this research, we leveraged from the precomputed slope and intercept values reported in [14] and are given in the row labeled *reported* of Table 2.

**TABLE 2.** Slope and intercept values for ITU-T P.834.1 [14] Interpolation.

| No. | Slope (a) | Intercept (b) |
|---|---|---|
| Computed | 0.9915 | 24.7996 |
| Reported in [14] | 0.8720 | 19.9487 |

A few words about alternative reference models are in order. At the time when this research work was commenced (i.e. between February, 2010 and February, 2011)ITU-T was designing a new model with the name of P.OLQA [29] that aimed to replace PESQ and WB-PESQ. P.OLQA was available in Orange Labs at the time when this work was being done. It was considered in meetings in France Telecom to employ P.OLQA for this work. However, P.OLQA was undergoing rigorous experimental validation at that time and could not be considered as authentic. Using P.OLQA while it was in such a phase at that time could naturally lead to unverifiable measurement artefacts in the quality estimation tests. Particularly, P.OLQA had not been benchmarked at that stage. Thus, it was decided to use WB-PESQ instead as it was a recommended standard of ITU-T for objective speech quality estimation. Moreover, the ability of P.OLQA to take into account pause and jump impairments is not known. Some results employing P.OLQA have appeared in recent literature [23], [30]. However, the efficiency of P.OLQA against distortions such as pauses and jumps have not appeared so far in any study. Notwithstanding this, we argue that the methodology presented by us is quite general and can easily incorporate P.OLQA or any other future model for objective estimation of speech quality. Any research work that would employ a newer or different reference model such as P.OLQA would, nonetheless, require the experimenter or researcher to be in a research facility such as Orange labs.

## VI. PREPARATION OF THE TEST MATERIAL

This section lists how the 4-state model can be used to generate stimuli that can be used either in auditory tests, or may be evaluated by an instrumental model such as WB-PESQ, the results of which may subsequently be used to evolve a formulation for $I_{e,WB,eff}$. To this end, this section is a continuation of section IV in the sense that a VoIP simulation completes once a network impaired speech signal has been generated. This can be accomplished in various ways. However, the best way to do this accurately is to use the codecs directly.

An alternate method could be to initiate a VoIP call over the Internet and to capture the VoIP packets using a packet sniffer at the receiver's side. The packet sniffer may also be provisioned with a jitter buffer, so as to match the capturing process closely with a typical VoIP client. However, in such a scheme one may face the problem of controlling the behavior of the Internet, which is not a trivial proposition. Another alternative could be to set up VoIP sessions between endpoints in a fully controlled environment inside a laboratory, thus allowing to accurately control the behavior of the Internet. However, we believe that such a scheme would be time-consuming (for instance, due to overheads associated with call setup and control) especially if a large number of test stimuli need to be generated. The former scheme is simple to implement as well. For instance, in this work, a loss, pause and jump pattern of an arbitrarily large length can be supplied to a decoder as a command line argument. The decoder can store this pattern in an array. While decoding the frames of the input speech file or the bit stream, the decoder can make decisions based on the contents of the array. Thus, at any stage of the decoding process if the decoder sees a "0" (no-loss) in the corresponding entry of the array, it faithfully decodes the frame. Similarly, if a "1" (loss) is found, the decoder takes the corresponding action for frame erasure either by playing its packet loss concealment (PLC) algorithm or by doing silence insertion. For a "2" (jump), the decoder does nothing i.e. neither it decodes the frame nor it plays the PLC algorithm and moves to the next frame. Similarly, for a 030 (pause) it plays the PLC algorithm before decoding the packet. In this manner the decoder faithfully generates a speech signal degraded by the pattern that was supplied to it by the user, resulting in an accurate simulation of the set of network impairments under test. Benefits of adopting a simulation-based approach for problem-solving is advocated favorably elsewhere in the technical literature [31].

In order to derive a formulation for $I_{e,WB,eff}$, it is important to prepare speech stimuli impaired with conditions that are representative of what may actually happen in a real VoIP network. To this end, the first thing that needs to be figured out are the various parameters. These parameters would also serve as independent variables for any formulation of the $I_{e,WB,eff}$ that may be derived. Similarly, the second thing to take into account is the maximum permissible values these parameters may acquire as well as the granularity of the values of these variables.

In this work, it has been decided to use a total of 10 such parameters. These are listed in Table 3. The first six parameters have been discussed earlier in this document and are self-explanatory. Rest of these can be described as follows:

**TABLE 3. Various network traffic parameters.**

| No. | Variable |
|---|---|
| 1 | mlr |
| 2 | mbl_loss |
| 3 | mpr |
| 4 | mbl_pause |
| 5 | mjr |
| 6 | mbl_jump |
| 7 | mir |
| 8 | mbl_impairment |
| 9 | $I_{e,WB}$ |
| 10 | $grad$ |

*mir* stands for *mean impairment rate* and is the sum of *mlr*, *mpr* and *mjr*.

Computation of *mbl_impairment* is a subtle matter and is assumed to be the sum of *mbl_loss*, *mbl_pause* and *mbl_jump* in this work. This, however, actually corresponds to the burst length of overall impairments, as opposed to the bursts of losses, or of pauses, or of jumps only, or to the sum of them. This parameter should actually be computed in the same manner as the individual *mbl_loss*, *mbl_pause* or *mbl_jump* are computed. To this end, the first pattern in Table 4, should be considered to have two bursts of impairments and *mbl_impairment* should be computed accordingly. The problem with this way of computing is that for the same values of *mbl_loss*, *mbl_pause* and *mbl_jump* in multiple patterns *mbl_impairment* may normally have different values. This can be seen in the second pattern of Table 4 where an insertion of a *zero* (bold faced) in the original pattern has altogether changed the value of *mbl_impairment* while the *mean burst lengths* of the individual impairments remain unchanged. Thus, this can obfuscate the process of generating multiple patterns, and subsequently multiple speech stimuli, with the same set of conditions; a requirement which normally needs to be fulfilled specially if an instrumental model is used to evaluate stimuli.[2] Thus, the way we compute *mbl_impairment* is less a matter of accuracy and more a matter of convenience.

**TABLE 4. Example loss, pause and jump pattern.**

| No. | pattern |
|---|---|
| 1 | ...0 0 1 1 1 2 2 3 1 2 0 1 2 3 2 1 3 0 0 ... |
| 2 | ...0 0 1 1 1 2 2 3 1 **0** 2 0 1 2 3 2 1 3 0 0 ... |

$I_{e,WB}$ corresponds to the *equipment impairment factor* and is a measure of purely codec related distortions.

*grad* corresponds to the gradient of $I_{e,WB,eff}$ for the overall *mir* ranging between 0–0.12. It was computed according to

[2]To be considered valid, it is normally required to aggregate the evaluations of an instrumental model for multiple stimuli for the same set of conditions [6].

**TABLE 5.** Values for $I_{e,WB}$–$R$, $I_{e,WB}$–$C$ and $grad$.

| No. | $I_{e,WB}$–$R$ | $I_{e,WB}$–$C$ | $grad$ |
|---|---|---|---|
| ITU-T G.711 | 31.9117 | 23.1735 | 3.4917 |
| ITU-T G.729 | 50.9637 | 39.9297 | 2.3949 |
| ITU-T G.722 | 17.7681 | 10.7343 | 5.6482 |

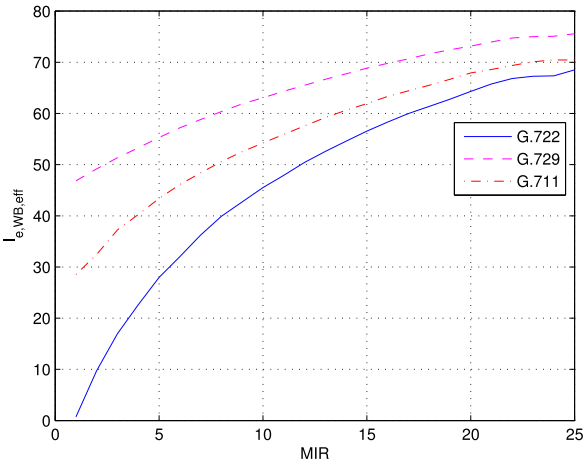



**FIGURE 5.** $I_{e,WB,eff}$ vs $mir$ for ITU-T G.711, G.729 and G.722.

equation (14). *grad* is a measure of degradation of quality as a function of the intensity of the combined temporal discontinuities and can be seen for the chosen codecs in Fig. 5.

Table 5 lists values for stable $I_{e,WB}$ and *grad* for various codecs. The first column lists values of $I_{e,WB}$–$R$ resulting from the slope and intercept terms *reported* in [14]. Whereas the second column lists the values of $I_{e,WB}$–$C$ resulting from the slope and intercept terms that were *computed* manually by following the procedure laid down in [14] and as reported in the second row of Table 2. In this work the former set of values has been utilized to remain coherent with [14]. The third column lists the values of *grad* computed for the codecs under consideration.

$$grad = \frac{I_{e,WB,eff}(mlr = 0.12) - I_{e,WB,eff}(mlr = 0.0)}{0.12 \times 100} \quad (14)$$

As stated earlier, a simulation based approach was pursued in this research, where distortions typical of a VoIP network were induced on a large number of clean speech samples during the process of decoding the corresponding coded bitstreams. Clean speech samples from experiments 1-A, 1-D and 1-O from ITU-T P-series supplement 23 were used. The NB codecs include: ITU-T G.711 (64 kbps) [18], ITU-T G.729 CS-ACELP (8 kbps) [32]. WB codec ITU-T G.722 (64 kbps) [33] was used in this research. All the codecs are enabled with their respective PLC algorithms specified in [32]–[34].

Each of *mlr*, *mpr* and *mjr* were simulated for *nine* values; [0, 0.005, 0.01, …, 0.04]. For each of the values of the variables stated above, their *target burst lengths* (i.e. *mbl_loss*, *mbl_pause* and *mbl_jump* respectively) are listed in Table 6. Thus, for each type of discontinuity, a total of *twenty-four*

**TABLE 6.** Various temporal discontinuity rates and the respective burst lengths (%).

| Temporal Discontinuity Rate | Burst Length |
|---|---|
| 0 | 0 |
| 0.005 | 1, 2 |
| 0.01 | 1, 2, 4 |
| 0.015 | 1, 2, 3 |
| 0.02 | 1, 2, 4 |
| 0.025 | 1, 2, 5 |
| 0.03 | 1, 2, 3 |
| 0.035 | 1, 2, 7 |
| 0.04 | 1, 4, 8 |

conditions were simulated with a fixed *payload size* of 20ms. For all the three temporal discontinuities, having twenty-four distortion conditions each, and three codecs this resulted in $24 \times 24 \times 24 \times 3 = 41,472$ network conditions. All of these conditions were applied separately to stimuli for each of the *three* codecs. Similarly, each condition was applied separately to stimuli from three different languages and, for each language, of two male and two female speakers by pseudo-randomly generating, possibly different, loss patterns each time. This was done to negate the effect of packet loss locations as in [15] by eventually averaging the *MOS* of the stimuli subjected to same distortion conditions. This was also aimed at removing the bias WB-PESQ may have towards language and gender [25, p. 105] [26]. This resulted in a total of $24 \times 24 \times 24 \times 3 \times 3 \times 4 = 497,664$ speech stimuli. The stimuli were latter evaluated using WB-PESQ. After aggregating the MOS-LQO for the stimuli subjected to similar network conditions, we were left with a total of $497,664/12 = 41,472$ input/output patterns.

A point worth mentioning is that, since the clean speech samples are coded at 16 kHz sampling rate, they were low-pass filtered and down-sampled using [35] before they were encoded in the case of NB codecs. Subsequently, the corresponding decoded speech samples were up-sampled before evaluation by WB-PESQ using [35].

## VII. INTRODUCTION TO GP

GP is a machine learning technique that coarsely emulates Darwinian evolution. The solution space in GP is composed of all the possible computer programs, or mathematical expressions, that might potentially solve a user specified problem. The computer programs are composed of *functions* and *terminals* that are relevant to a particular problem domain. The functions may be arithmetic operations and logical expressions. The terminals may be external inputs to the programs such as constants and input domain variables. A finite set of such randomly generated programs forms an initial population. The ability of the members of such a population to effectively solve a problem is enhanced by leveraging from genetic operators of biological evolution, namely crossover and mutation. Thus, for instance, two individual programs are randomly chosen from the underlying population and a new offspring is formed by applying

the aforementioned genetic operators. Normally, this mating process is repeated till the size of the offspring population becomes equal to that of the parent population. At this stage, the offspring are evaluated in terms of their *fitness* in solving the problem of interest. Consequently, the fitter individuals are retained and the worse are littered. This evolutionary process is repeated until a certain user-specified criterion is met. For instance, this could be to stop when an individual with desired fitness is achieved or when a certain number of generations have elapsed. To this end, GP differs from the traditional optimization approaches that aim to tune the coefficients of an already defined mathematical model that represents the solution of the problem at hand. Numerous solution representations for GP trees exist, albeit abstract syntax trees are by far the most common choices.

Given a problem setting, GP can potentially search for a globally optimal solution as opposed to getting stuck in local minima as in various other optimization algorithms used in machine learning. This is attributed to an evolutionary process driven by stochastic changes in the *genomes* of a population's individuals.

In addition to searching for a suitable structure of the desired solution, or a mathematical expression, GP is also known for tuning its coefficients. Such a mechanism is generally implicit in the evolutionary process whereby GP tries to fit number(s) from a set of user-specified constants (i.e. from the terminals set) into the genomes of individual programs. GP may also synthesize useful constants from various functions alone. Various combinations of functions and system variables may result in sufficing constants. A detailed account of this may be found in [36]. In certain implementations, GP-based evolution may be augmented with a dedicated optimization algorithm for tuning the coefficients of individual models. A number of meta-heuristic and numerical algorithms have been used in the past to achieve this objective [37]–[40].

GP is also known to chisel off redundant data attributes to a few significant ones, as in [41]. To this end, GP also performs a non-linear parameter significance analysis during program evolution by retaining the highly significant parameters in favorable programs while rejecting the less significant ones.

## VIII. EXPERIMENTAL SETUP

We performed three GP experiments to evolve models for $I_{e,WB,eff}$ using the input/output data patterns. The accumulation of data patterns has already been discussed in section VI. As previously in [4] and [42], we used GPLab[3] for evolution. The common parameters for all the experiments are listed in Table 7.

In all experiments, we chose scaled root mean squared error ($RMSE_s$) as the preferred fitness criterion. $MSE_s$ is given by equation (15).

$$MSE_s(y,t) = 1/n \sum_{i}^{n} (t_i - (a + by_i))^2 \tag{15}$$

where $y$ is a function of the input parameters (a mathematical expression), $y_i$ represents the value produced by a GP individual and $t_i$ represents the target value which is the corresponding $MOS$. $a$ and $b$ adjust the slope and y-intercept of the evolved expression to minimize the squared error. They are computed according to equations (16) and (17):

$$b = \frac{cov(t,y)}{var(y)} \tag{16}$$

$$a = \bar{t} - b\bar{y}, \quad b = \frac{cov(t,y)}{var(y)} \tag{17}$$

where $\bar{t}$ and $\bar{y}$ represent the mean values of the corresponding entities whereas *var* and *cov* mean the variance and covariance respectively. This approach is known as *linear scaling* and is found to be very beneficial for the symbolic regression tasks with GP [40]. Instead of using *protected* functions, any inputs were admissible to all the functions.

For the input values outside the domain of the functions *log, sqrt, division* and *pow*, *NaN* (undefined) values are generated. This results in the individual concerned being assigned the worst possible fitness.

Tournament selection with Lexicographic Parsimony Pressure (LPP) [43] was used in both experiments. In this selection strategy a group of G (G $\geq$ 2) individuals is picked randomly from the current population. The individual with the highest fitness in the group is selected as a parent. In the case of a tie between two or more individuals, their expression sizes are compared and the smaller individual is picked. Furthermore, the selection criterion was based on the notion that population diversity can be enhanced if mating takes place between two, fitness-wise, dissimilar individuals, as suggested by Gustafson *et al.* [44]. This selection scheme has been shown to perform better in the symbolic regression domain and, hence, it was employed in this research. This simple addition to the selection criterion only requires one to

**TABLE 7. Common GP parameters among all experiments.**

| Parameter | Value |
|---|---|
| Initial Population Size | 300 |
| Initial Tree Depth | 6 |
| Selection | LPP |
| Tournament Size | 2 |
| Genetic Operators | Crossover and Subtree Mutation |
| Operators Probability Type | Adaptive |
| Initial Operator probabilities | 0.5 each |
| Survival | Half Elitism |
| Function Set | plus, minus, multiply, divide,sin, cos, $log_2$, $log_{10}$, $log_e$, sqrt, power, if |
| Terminal Set | Random real-valued numbers between 0.0 and 1.0. Integers (2–10) and Network traffic parameters from Table III. |
| No. of Runs | 50 |

[3]GPLab is a Matlab toolbox for GP developed by Sara Silva and can be found at: http://gplab.sourceforge.net/

**TABLE 8.** Statistical analysis of the gp experiments and derived models. (a) *RMSE* Statistics for Best Individuals of 50 Runs for Experiments 1, 2 & 3. (b) Results of Mann-Whitney-Wilcoxon Significance Test. (c) Performance Statistics of the Proposed Models.

| Stats | Experiment 1 | | | Experiment 2 | | | Experiment 3 | | |
|---|---|---|---|---|---|---|---|---|---|
| | $RMSE_{tr}$ | $RMSE_{te}$ | $Size$ | $RMSE_{tr}$ | $RMSE_{te}$ | $Size$ | $RMSE_{tr}$ | $RMSE_{te}$ | $Size$ |
| Mean | 5.5482 | 98.9506 | 26.2800 | 5.5501 | 18.9480 | 29.34 | 5.3435 | 13.9066 | 28.18 |
| Std. Dev. | 0.3514 | 152.5866 | 11.4661 | 0.3850 | 70.2753 | 12.3612 | 0.4612 | 47.1959 | 9.7304 |
| Max. | 5.9748 | 500 | 68 | 6.0084 | 494.5177 | 73 | 5.8988 | 333.6037 | 59 |
| Min. | 4.5409 | 4.8182 | 11 | 4.4108 | 4.3708 | 6 | 4.4021 | 4.3808 | 14 |

(a)

| | Experiment 1 | | | Experiment 2 | | | Experiment 3 | | |
|---|---|---|---|---|---|---|---|---|---|
| | $RMSE_{tr}$ | $RMSE_{te}$ | $Size$ | $RMSE_{tr}$ | $RMSE_{te}$ | $Size$ | $RMSE_{tr}$ | $RMSE_{te}$ | $Size$ |
| **Experiment 1** | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| **Experiment 2** | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| **Experiment 3** | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |

(b)

| Model | Training | | | Test | | |
|---|---|---|---|---|---|---|
| | $RMSE_{sMOS}$ | $RMSE_s$ $I_{e,WB,eff}$ | $\rho$ $I_{e,WB,eff}$ | $RMSE_s$ $MOS$ | $RMSE_s$ $I_{e,WB,eff}$ | $\rho$ $I_{e,WB,eff}$ |
| Equation (19) | 0.1763 | 4.8185 | 0.8840 | 0.1759 | 4.8182 | 0.8805 |
| Equation (20) | 0.1602 | 4.4108 | 0.9038 | 0.1596 | 4.3708 | 0.9028 |
| Equation (22) | 0.1619 | 4.4021 | 0.9042 | 0.1611 | 4.3808 | 0.9023 |
| Equation (23) | 0.1692 | 4.6026 | 0.8948 | 0.1679 | 4.5460 | 0.8944 |
| Equation (24) | 0.1764 | 4.8644 | 0.8816 | 0.1948 | 5.4231 | 0.8781 |

(c)

ensure that mating does not take place between individuals of equal fitness.

The replacement criterion was based on retaining the best half of both children and parents in the new population while discarding the rest. This replacement strategy is termed as *half elitism* in GPLab [45].

It is typical to conduct several independent runs of GP. In this case, all experiments entailed 50 independent runs each spanning 50 generations.

The only difference between the first two experiments was that in experiment 1 a maximum tree depth of 17 was allowed whereas for experiment 2 it was allowed to be 10. A reduced tree depth in the second experiment was allowed to see if more parsimonious individuals could be obtained having fitness comparable to the individuals of the first experiment.

The difference between experiment 2 and experiment 3 was that in the latter only four network traffic parameters were allowed, namely, $I_{e,WB}$, $grad$, $mir$ and $mbl\_impairment$. These parameters were chosen as a result of a parameter significance analysis performed in section IX (see Fig. 8). A maximum tree depth of 10 was allowed in experiment 3. A reduced number of parameters was chosen to see if GP could find better models in a smaller corresponding search space.

## IX. RESULTS AND ANALYSIS

Of 41,472 input/output patterns reported in section VI 70% were used for training and 30% for testing the evolved models. Statistics pertaining to $RMSE_s$ (scaled root mean squared error) of training and test data of both GP experiments are listed in Table 8. The table also lists various statistics related to the tree sizes of GP individuals, in terms of the number of

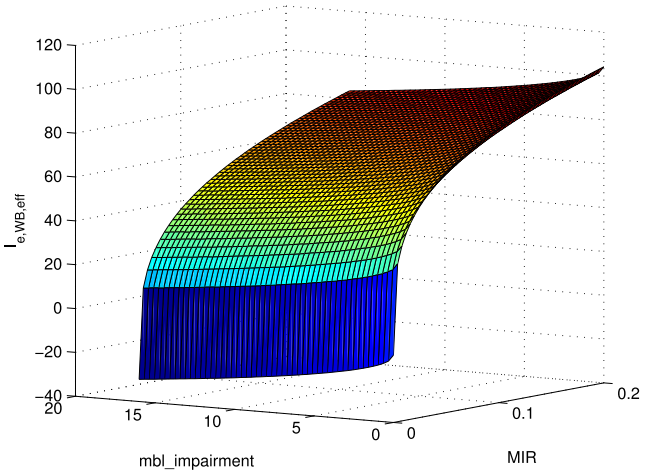



**FIGURE 6.** Response of equation (19). as a function of *mir* and *mbl_impairment*.

nodes. The results of the three experiments in the final generations were also treated to a Mann-Whitney-Wilcoxon test to assay the significance of differences in various respects. The significance analysis is reported in Table 8 where a value of '1' confirms a significant difference, at a 5% confidence level, whereas a '0' implies otherwise. It was found that the overall results of the first two experiments are not significantly different from each other in terms of fitness over training data and tree sizes, but vary in terms of fitness over test data. However, the results of experiment 3 differ from the first two experiments in terms of fitness over the training data, whereas in terms of test data and tree size the models are different only in comparison to experiments 1 and 2 respectively. A close analysis of the statistics reported in Table 8 reveals that the

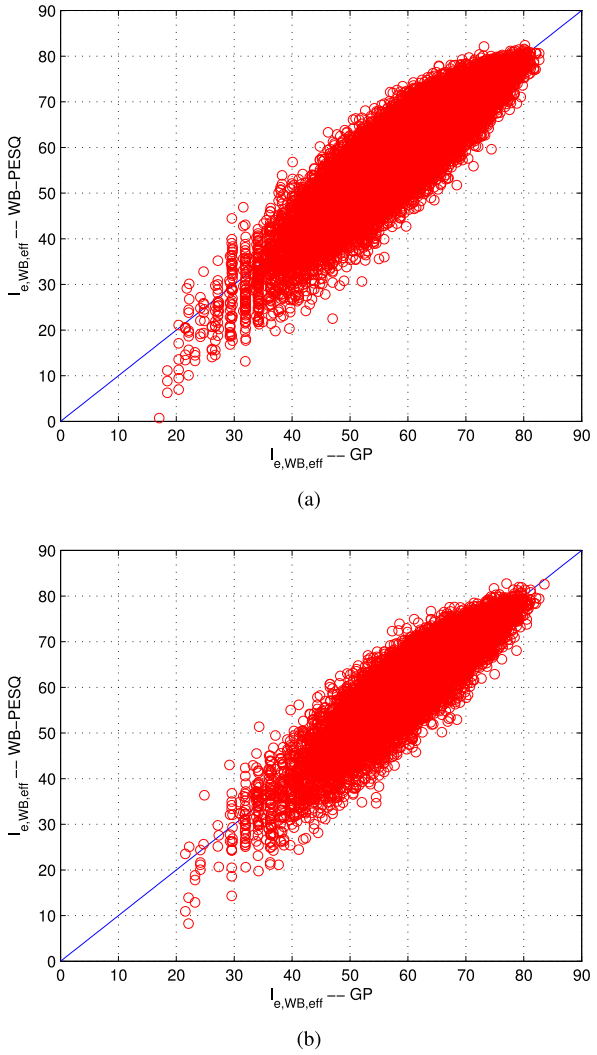



**FIGURE 7.** $I_{e,WB,eff}$ **predicted by equation (22) vs target** $I_{e,WB,eff}$ **for: (a) training data (b) test data.**

models produced by experiment 3 are generally superior to the first two experiments.

It is also worth mentioning that in Table 8, the difference between *RMSE* over training data as compared to test data is rather huge for all the three experiments. For instance, it can be seen that experiment 1 evolved a model that had $RMSE_{te} = 500$ (the model with the worst fitness over the test data). On the other hand, experiment 1 produced a model with $RMSE_{te} = 5.97$ (the model with worst fitness over the training data). We believe that this rather huge difference in results is due to the presence of outliers. In Table 8 $RMSE_{tr}$ and $RMSE_{tr}$ stand for the *scaled root mean squared error* over the *training* and *test* data respectively.

In this paper we propose four models. Three of these correspond to the best performing models of experiment 1, 2 & 3 with respect to the test data and are represented by equations (19), (20) and (22) respectively. The smallest overall models belonged to experiments 2 and 3. Nonetheless, the smallest individual of experiment 3 had a better fitness as compared to the smallest individuals of experiment 1 & 2 and it is represented by equation (23). The $RMSE_s$ and Pearson's product moment correlation coefficient ($\rho$), corresponding to $I_{e,WB,eff}$ for these models are compared with each other in Table 8. The values of $RMSE_s$ corresponding to *MOS-LQO* are also listed as another comparison. These were computed by converting the target values of $I_{e,WB,eff}$ and those obtained by the models under consideration to the *MOS* scale. This may be done by obtaining the values of $R$ corresponding to $I_{e,WB,eff}$ from equation (4). The result can then be transformed to the original $R$ scale for the NB-only context using equation (18); the inverse of equation (3). The resulting values of $R$ can be converted to the *MOS* scale using transformation (2). The significance of all of the models can be judged by observing that the values of $RMSE_s$ on the *MOS* scale in all cases range between 0.1– 0.16, exhibiting rather smaller values. Equation (22) represents the best overall model and is a function of three parameters only i.e. *grad*, *mir* and *mbl_impairment*.

Another criterion for choosing the model can be the size of the model. This can be a crucial factor as smaller models have lighter computational footprints and can be amenable for implementation on terminals or home gateways. As discussed earlier we tried to address this constraint in our experiments by using LPP as suggested by Luke [43]. Similarly, we also restricted the maximum tree depth in experiments 2 and 3 to *ten* in the hope of finding smaller models. To this end, if model *brevity* is a concern then equations (19) and (23) may be preferred as they are smaller in size.

$$R_{NB} = \frac{R_{NB/WB}}{1.29} \tag{18}$$

Equation (20) has the best statistics among all. Fig. 7 shows the scatter plots of equation (20) versus WB-PESQ, where it can be seen that the data points produced by both are firmly glued to the 45 degrees reference line.

$$I_{e,WB,eff} = \left\{ mir \times cos(I_{e,WB}) + \frac{\sqrt{mbl\_imp}}{I_{e,WB}} - mir^{1/4} - mir \right\} \times (-163.87) - 9.35 \tag{19}$$

$$I_{e,WB,eff} = \left\{ \sqrt{\frac{\sqrt{log_{10}(grad)}}{mir + 9}} + \left[ (sin(mir) + mir)^{\frac{log_{10}(grad)}{4}} - \sqrt{\frac{\sqrt{mpr}}{mbl\_loss+9}} \right]^{E} \right\} \times (-0.0933) + 87.1174 \tag{20}$$

where $E$ (the exponent) in equation (20) is given by equation (21).

$$E = \sqrt{log_{10}(grad)} - \sqrt{\frac{log_{10}(grad)}{mbl\_jump + 9}} \tag{21}$$

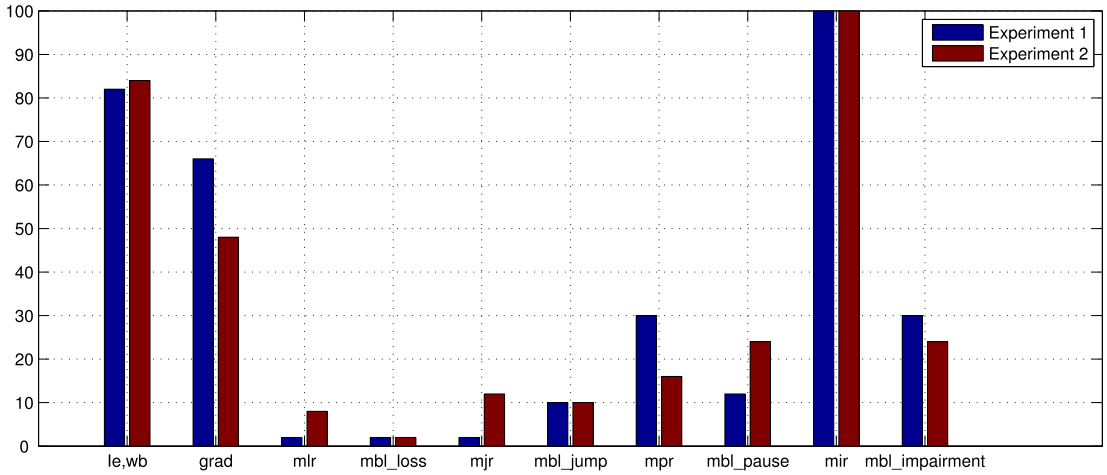



**FIGURE 8.** Percentage of the best individuals employing various input parameters in acceptable runs of each of the two experiments.

**TABLE 9.** Comparison between the prediction accuracies of the E-Model and the proposed model.

| Codec (kbps) | E-Model | | | | Equation (22) | |
|---|---|---|---|---|---|---|
| | Bpl Computed | RMSE Bpl Reported | RMSE train | RMSE test | RMSE train | test |
| G.711 (64) | 22.39 | 25.1 [47, pp–2] | 6.7971 | 6.7003 | 4.6748 | 4.5626 |
| G.729 (8) | 30.50 | 19.0 [47, pp–2] | 4.0824 | 3.8701 | 3.0513 | 3.1362 |
| G.722 (64) | 19.8053 | 7.1 [48, pp–2] | 8.1087 | 8.1510 | 5.6865 | 5.6093 |
| Average | – | – | 6.3294 | 6.2405 | 4.4709 | 4.4360 |
| % PG | – | – | – | – | 29.36 | 28.92 |

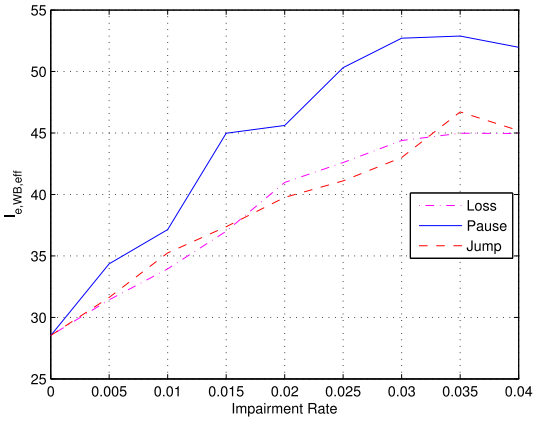
**TABLE 10.** Target network impairment conditions for the auditory tests.

| Condition | mlr | mbl_loss | mjr | mbl_jump | mpr | mbl_pause | mir | mbl_impairment |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0.03 | 1.0 | 0 | 0 | 0 | 0 | 0.03 | 1.0 |
| 3 | 0.03 | 4.0 | 0 | 0 | 0 | 0 | 0.03 | 4.0 |
| 4 | 0 | 0 | 0.03 | 1.0 | 0 | 0 | 0.03 | 1.0 |
| 5 | 0 | 0 | 0.03 | 4.0 | 0 | 0 | 0.03 | 4.0 |
| 6 | 0 | 0 | 0 | 0 | 0.03 | 1.0 | 0.03 | 1.0 |
| 7 | 0 | 0 | 0 | 0 | 0.03 | 4.0 | 0.03 | 4.0 |
| 8 | 0.06 | 1.0 | 0 | 0 | 0 | 0 | 0.06 | 1.0 |
| 9 | 0.06 | 4.0 | 0 | 0 | 0 | 0 | 0.06 | 4.0 |
| 10 | 0 | 0 | 0.06 | 1.0 | 0 | 0 | 0.06 | 1.0 |
| 11 | 0 | 0 | 0.06 | 4.0 | 0 | 0 | 0.06 | 4.0 |
| 12 | 0 | 0 | 0 | 0 | 0.06 | 1.0 | 0.06 | 1.0 |
| 13 | 0 | 0 | 0 | 0 | 0.06 | 4.0 | 0.06 | 4.0 |
| 14 | 0.09 | 1.0 | 0 | 0 | 0 | 0 | 0.09 | 1.0 |
| 15 | 0.09 | 4.0 | 0 | 0 | 0 | 0 | 0.09 | 4.0 |
| 16 | 0 | 0 | 0.09 | 1.0 | 0 | 0 | 0.09 | 1.0 |
| 17 | 0 | 0 | 0.09 | 4.0 | 0 | 0 | 0.09 | 4.0 |
| 18 | 0 | 0 | 0 | 0 | 0.09 | 1.0 | 0.09 | 1.0 |
| 19 | 0 | 0 | 0 | 0 | 0.09 | 4.0 | 0.09 | 4.0 |
| 20 | 0.04 | 4.0 | 0.04 | 4.0 | 0.04 | 4.0 | 0.12 | 12.0 |

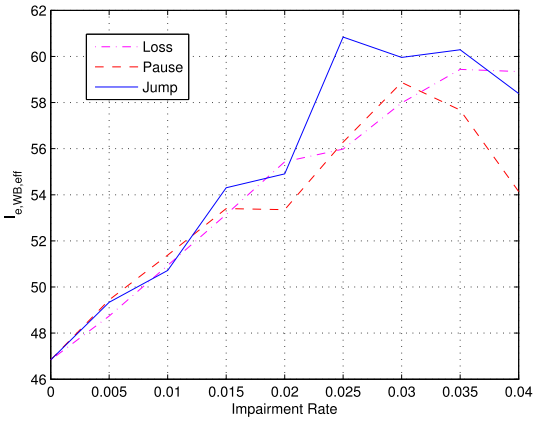
$$I_{e,WB,eff} = \left\{ \frac{log_{10}\left(\frac{0.54}{grad} + 3 \times mir\right) + \frac{log_{10}\left(\frac{0.74}{grad} + 2 \times mir\right)}{3}}{7 \times log_{10}\left(\frac{0.54}{grad} + 2 \times mir + 6.56 - \sqrt{mbl\_imp}\right) + mir} \right\} \times (270.37) + 102.40 \quad (22)$$

$$I_{e,WB,eff} = (sin(grad \times mir))^{\frac{\sqrt{40 \times mbl\_imp}}{I_{e,WB}}} \times 107.43 - 5.94 \quad (23)$$

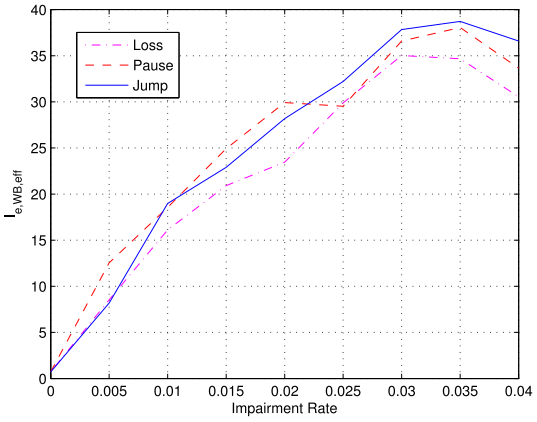
Fig. 6 displays the response of equation (19) as a function of *mir* and *mbl_impairment* suggesting monotonicity of this equation.




(a)




(b)




(c)

**FIGURE 9.** Variation in $I_{e,WB,eff}$ as a function of losses, pauses and jumps for: (a) ITU-T G.711 (b) ITU-T G.729 (c) ITU-T G.722. $I_{e,WB,eff}$ was derived using WB-PESQ.

Fig. 7a and 7b show the scatter plots for the output of equation (22) for the training and test data respectively. It can be seen that the points are considerably glued together around the 45° reference line.

A significance analysis of the various VoIP traffic parameters, in terms of their appearance in the best individuals

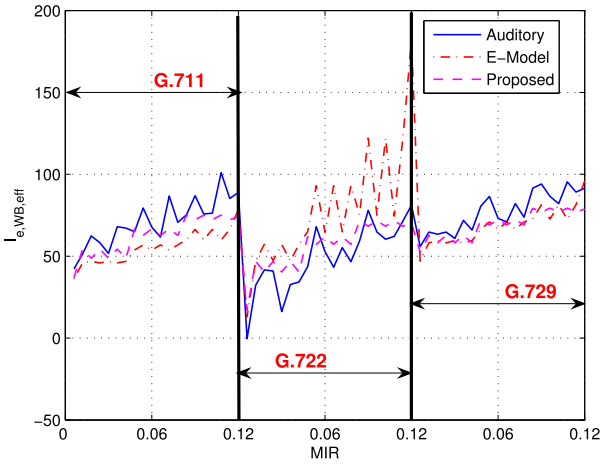



**FIGURE 10.** $I_{e,WB,eff}$ vs mir derived from auditory tests, E-Model and equation (22) are plotted for G.711, G.722 and G.729.

of 50 runs of each of the first two experiments, was done. The results are graphed in Fig. 8. According to this *mir* had the highest utility, and appeared in 100% of the individuals. The second most highly utilized parameter was $I_{e,WB}$ which appeared in more than 80% of the individuals of both experiments. The third most sought-after parameter was *grad*, appearing in up to 50–70% of the best individuals of both experiments. Rest of the parameters, namely, *mlr, mbl_loss, mjr, mbl_jump, mpr and mbl_impairment* only had utility in up to 30% of the best individuals of both experiments with *mbl_impairment* winning out. It is quite interesting to note that *mlr*, which is the main source of impairment in VoIP networks, had only a marginal representation in the models of both experiments. Same can be seen for *mjr*.

Fig. 9a, 9b and 9c show the variability of $I_{e,WB,eff}$ as a function of losses, pauses and jumps for ITU-T G.711, ITU-T G.729 and ITU-T G.722 respectively. It is interesting to note that WB-PESQ is not oblivious of the presence of pauses and jumps in the degraded stimuli. Rather in the case of ITU-T G.711 (Fig. 9a), it judges speech quality more severely for the case of pauses as compared to losses and jumps. This observation contrasts with the generally held notion that the WB-PESQ cannot estimate the effect of pauses and jumps [8]. It is also interesting to note that the individual effect of each of these impairments on speech quality is more or less the same. This observation is coherent with the results reported in [8].

### A. COMPARISON WITH MULTIPLE LINEAR REGRESSION

A multiple linear regression was performed to compare the performance enhancement lent by the models proposed by GP. To this end, we only included $I_{e,WB}$, *grad*, *mir* and *mbl_impairment* in the regression analysis as they appeared as the most significant parameters of the GP experiments (see Fig. 8). The resulting model is shown by equation (24). Performance results of this model are compared by those produced by GP and are shown in Table 8. It can be seen that

**TABLE 11.** Results of the auditory tests. (a) Comparison Between the Results of Auditory Tests and WB-PESQ. (b) Comparison between the Prediction Accuracies of the E-Model and the Proposed Model Against Data From Auditory Tests.

| $RMSE$ $MOS$ | $\rho$ $MOS$ |
|---|---|
| 0.4475 | 0.8399 |

(a)

| Codec (kbps) | E-Model | | | | Equation (22) | |
|---|---|---|---|---|---|---|
| | Bpl | $I_{e,WB}$ | $RMSE$ | $\rho$ | $RMSE$ | $\rho$ |
| G.711 (64) | 25.1 | 36 | 18.2549 | 0.7827 | 11.8532 | 0.8182 |
| G.729 (8) | 19.0 | 47 | 34.9341 | 0.8249 | 9.4212 | 0.9309 |
| G.722 (64) | 7.1 | 13 | 46.6618 | 0.8124 | 11.8840 | 0.8966 |
| Overall | – | – | 24.8994 | 0.3852 | 11.1129 | 0.8758 |

(b)

all the models produced by GP have a superior performance than equation (24).

$$I_{e,WB,eff} = 0.35 \times I_{e,WB} - 0.006 \times grad + 383.62 \times mir - 1.18 \times mbl\_imp + 34.65 \quad (24)$$

### B. COMPARISON WITH E-MODEL

Finally, a comparison of equation (22) was done with the E-Model's formulation of the $I_{e,WB,eff}$ which is represented by equation (25). Out of the four models proposed we chose equation (22) for the mere reason that it represents a smaller model with performance comparable to the best overall model (represented by equation (19)).

$$I_{e,WB,eff} = I_{e,WB} + (129 - I_{e,WB}) \times \frac{Ppl}{\frac{Ppl}{BurstR} + Bpl} \quad (25)$$

where *Ppl* refers to the *percentage of packet loss*. *Bpl* is the *packet loss robustness factor* and is a codec specific entity. 129 refers to $R_{NB/WB}$ for the direct channel. *BurstR* is the so-called Burst Ratio and is defined according to [1] by equation (26).

$$BurstR = (1 - Ppl/100) \times \frac{1}{q} \quad (26)$$

where $q$ refers to the transition probability between the *loss* and the *found* state according to the two-state Markov model shown in Fig. 2. It is also worth reiterating that according to this model $\frac{1}{q}$ refers to *mbl_loss* (see equation (6)). Thus, equation (26) can alternatively be represented as:

$$BurstR = (1 - Ppl/100) \times mbl\_loss \quad (27)$$

A direct comparison of the models proposed in this work is not possible with the E-Model. The reason is that the models proposed here are functions of *mir* and *mbl_impairment* as opposed to *Ppl* and *mbl_loss* (or *BurstR* corresponding to loss). To this end, to perform a fair comparison between the E-Model and the models proposed in this work we slightly tweak equations (25) and (26) respectively as follows.

$$I_{e,WB,eff} = I_{e,WB} + (129 - I_{e,WB}) \times \frac{Pir}{\frac{Pir}{BurstR} + Bpl} \quad (28)$$

where *Ppl (percentage packet loss)* has been replaced by a new term *Pir (percentage impairment rate)* and is equal to $mir \times 100$. It corresponds to all three impairments (i.e. losses, pauses and jumps) as opposed to losses only. Equation (27) can be altered to redefine *BurstR* as follows (29).

$$BurstR = (1 - mir) \times mbl\_impairment \quad (29)$$

where $mir = Pir/100$ has been chosen for the sake of convenience.

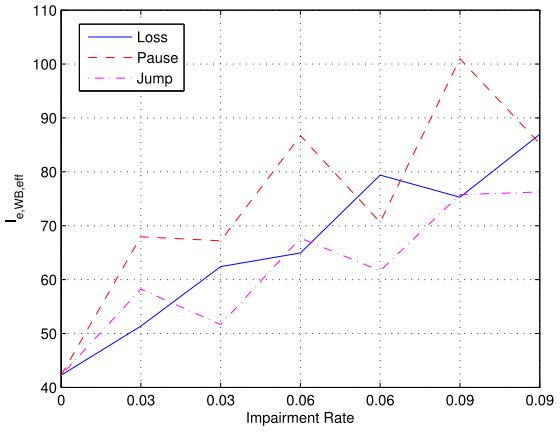
*Bpl* values for equation (25) were computed separately for each of the codecs over the training data by employing a simple generalized version of the *simulated annealing* algorithm [46]. The performance was analyzed using the test data. The results are reported in Table 9 for each codec. Percentage *Prediction Gain (PG)* of 28.92 % was observed for unseen data in an *RMSE* sense. This is calculated according to equation (30). Table 9 also compares the computed values of E-Model against those reported in the standards (see columns 2 & 3).

$$\%PG = \frac{RMSE_e - RMSE_p}{RMSE_e} \times 100 \quad (30)$$

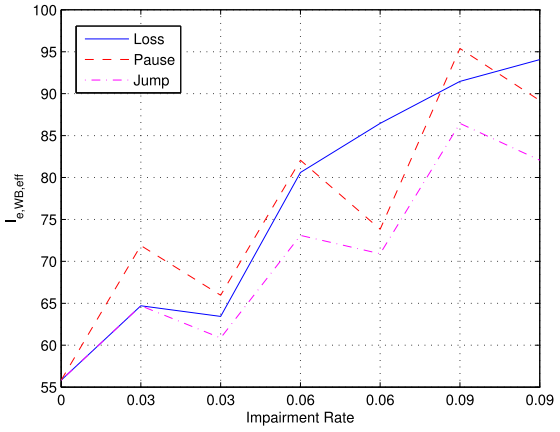
where, $RMSE_e$ and $RMSE_p$ represent the *RMSE* of equations (25) and (22) respectively.

### C. PERFORMANCE EVALUATION AGAINST DATA FROM AUDITORY TESTS

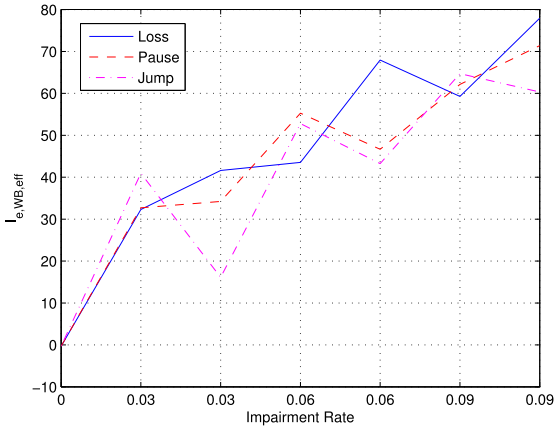
In order to further validate the results, a comparison was made between the results of the proposed model and the traditional E-Model formulation for $I_{e,WB,eff}$. To this end, an auditory test was performed in which a subset of network impairment conditions was inflicted on utterances from four French speakers (2 male and 2 female). The conditions are listed in Table 10. These conditions were supplied as target values to the 4-state model to obtain *four* sets of such conditions corresponding to each of the four speakers. This resulted in a total of 80 conditions. These conditions were applied to the stimuli for each of the three codecs to obtain a total of 240 speech stimuli (i.e. $80 \times 3 = 240$). Each of the stimuli were evaluated on an ACR scale by 24 subjects and the results were aggregated. Results were further aggregated over stimuli of all the speakers on the basis of similarity of target network conditions that had been inflicted on them, resulting in a total of 60 MOS values (i.e. $240/4 = 60$). Further, all the 240 stimuli were evaluated using WB-PESQ to compare it against the auditory tests. The results are reported in Table 11. The table shows that there is a high correlation between the results of the auditory data and WB-PESQ.




(a)




(b)




(c)

**FIGURE 11.** Variation in $I_{e,WB,eff}$ as a function of losses, pauses and jumps for: (a) ITU-T G.711 (b) ITU-T G.729 (c) ITU-T G.722. $I_{e,WB,eff}$ was derived using auditory tests.

The *MOS* values obtained by auditory tests were converted to target $I_{e,WB,eff}$ using equations (2), (3) and (4). The corresponding network conditions were used as input variables to obtain $I_{e,WB,eff}$ from the traditional E-Model formulation (equation (25)) and the proposed model (equation (22)) and compared against the $I_{e,WB,eff}$ produced by E-Model.

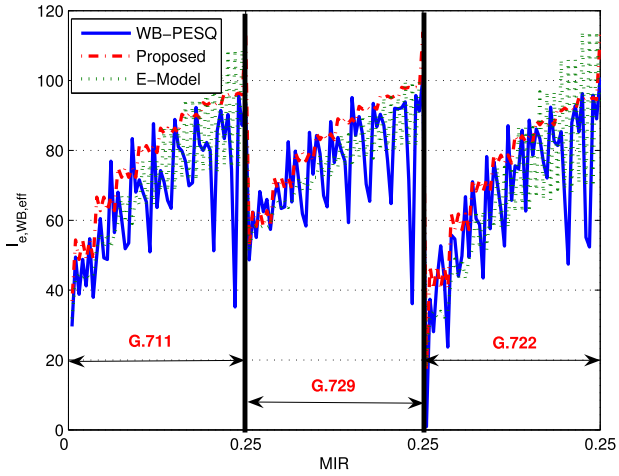



**FIGURE 12.** $I_{e,WB,eff}$ vs mir derived from WB-PESQ, E-Model and equation (22) are plotted for G.711, G.722 and G.729.

The comparison is shown in Table 11. It can be seen that the proposed model correlates well with the data from the auditory tests as opposed to the E-Model. However, performance of both models in terms of *RMSE* is not commendable (i.e. $RMSE = 11.11$ for equation (22)). Table 11 also reports values for *Bpl* that were taken from relevant standards (see [47, p. 2] for ITU-T G.711 and G.729 and [48, p. 2] for ITU-T G.722).

Fig. 10 plots $I_{e,WB,eff}$ as a function of *mir* for each of the three codecs derived from auditory tests, E-Model and equation (22).

Fig. 11 shows $I_{e,WB,eff}$ as a function of losses, pauses and jumps for each of the three codecs.

### D. ON EXTRAPOLATION ABILITY OF THE MODEL

The proposed models have been trained and tested on data where the *mean impairment rate* is less than or equal to 0.12 ($mir \leq 0.12$). In order to further validate the extrapolation ability of the model for *mir* exceeding beyond 0.12 another test was performed. The test was designed according to what has been reported in section IX-C with the only difference that the maximum *mir* was extended to 0.25. The conditions can be seen in Table 12. Again, these conditions were supplied as target values to the 4-state model to obtain *four* sets of such conditions corresponding to each of the four speakers. This resulted in a total of 200 conditions. These conditions were applied to the stimuli for each of the three codecs to obtain a total of 600 speech stimuli (i.e. $200 \times 3 = 600$). Eventually, all the stimuli were evaluated using WB-PESQ to obtain *MOS* scores.

The resulting *MOS* values obtained by WB-PESQ were converted to target $I_{e,WB,eff}$ using equations (2), (3) and (4). The corresponding network conditions were used as input variables to obtain $I_{e,WB,eff}$ from the traditional E-Model formulation (equation (25)) and the proposed model (equation (22)). The comparison is shown in Table 13. It can be seen that the proposed model has performed better than

**TABLE 12.** Target network impairment conditions for the extrapolation tests.

| Condition | mlr | mbl_loss | mjr | mbl_jump | mpr | mbl_pause | mir | mbl_impairment |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0.03 | 1.0 | 0 | 0 | 0 | 0 | 0.03 | 1.0 |
| 3 | 0.03 | 4.0 | 0 | 0 | 0 | 0 | 0.03 | 4.0 |
| 4 | 0 | 0 | 0.03 | 1.0 | 0 | 0 | 0.03 | 1.0 |
| 5 | 0 | 0 | 0.03 | 4.0 | 0 | 0 | 0.03 | 4.0 |
| 6 | 0 | 0 | 0 | 0 | 0.03 | 1.0 | 0.03 | 1.0 |
| 7 | 0 | 0 | 0 | 0 | 0.03 | 4.0 | 0.03 | 4.0 |
| 8 | 0.06 | 1.0 | 0 | 0 | 0 | 0 | 0.06 | 1.0 |
| 9 | 0.06 | 4.0 | 0 | 0 | 0 | 0 | 0.06 | 4.0 |
| 10 | 0 | 0 | 0.06 | 1.0 | 0 | 0 | 0.06 | 1.0 |
| 11 | 0 | 0 | 0.06 | 4.0 | 0 | 0 | 0.06 | 4.0 |
| 12 | 0 | 0 | 0 | 0 | 0.06 | 1.0 | 0.06 | 1.0 |
| 13 | 0 | 0 | 0 | 0 | 0.06 | 4.0 | 0.06 | 4.0 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 44 | 0.24 | 1.0 | 0 | 0 | 0 | 0 | 0.24 | 1.0 |
| 45 | 0.24 | 4.0 | 0 | 0 | 0 | 0 | 0.24 | 4.0 |
| 46 | 0 | 0 | 0.24 | 1.0 | 0 | 0 | 0.24 | 1.0 |
| 47 | 0 | 0 | 0.24 | 4.0 | 0 | 0 | 0.24 | 4.0 |
| 48 | 0 | 0 | 0 | 0 | 0.24 | 1.0 | 0.24 | 1.0 |
| 49 | 0 | 0 | 0 | 0 | 0.24 | 4.0 | 0.24 | 4.0 |
| 50 | 0.25 | 1.0 | 0.25 | 1.0 | 0.25 | 1.0 | 0.25 | 1.0 |

**TABLE 13.** Target network impairment conditions for the extrapolation tests.

| | E-Model | | Equation (22) | | Equation (22) after rescaling | |
|---|---|---|---|---|---|---|
| Codec (kbps) | $RMSE$ | $\rho$ | $RMSE$ | $\rho$ | $RMSE$ | $\rho$ |
| G.711 (64) | 18.0565 | 0.5331 | 16.4561 | 0.7549 | 11.4129 | 0.7549 |
| G.729 (8) | 14.1021 | 0.4382 | 14.3783 | 0.6135 | 11.5468 | 0.6135 |
| G.722 (64) | 20.1098 | 0.6107 | 14.7778 | 0.8313 | 11.7225 | 0.8313 |
| Overall | 17.5661 | 0.5691 | 15.2307 | 0.7659 | 11.5614 | 0.7659 |

the E-Model both in terms of the RMSE and the *Pearson's correlation coefficient* ($\rho$). The proposed model was further *rescaled* to better fit the target data. It was found that the proposed model showed a reduction in RMSE.

Fig. 12 plots $I_{e,WB,eff}$ as a function of *mir* for each of the three codecs derived from WB-PESQ, E-Model and equation (22).

## X. CONCLUSION

In this paper, we have proposed a novel methodology for determining NB/WB equipment impairment factors $I_{e,WB,eff}$, for a mixed NB/WB context. We have used GP to perform symbolic regression so as to generate simple formulae for $I_{e,WB,eff}$. GP is advantageous in the sense that the resulting formulae do not result from human bias, but as a direct consequence of program evolution. Moreover, parameter optimization is done in parallel with evolution for every model using linear scaling. The desired models are applicable for network distortion conditions under observation.

Another novelty of our work is that we have taken into account additional sources of impairments. These impairments are jumps and pauses. WB-PESQ has been used as a reference model as opposed to auditory tests. This is suitable for ease of repeatability. An interesting observation of our work is that WB-PESQ judges the effect of pauses and jumps almost equally well as it judges the effect of packet loss.

We have also proposed a 4-state loss, pause, jump Markov model to characterize the nature of VoIP traffic that may be affected both by the IP network and the jitter buffer. To this end, we propose it as a useful tool for simulating VoIP packet traces.

A comparison of one of the proposed models (equation (22)) has been done with the E-Model. It is found that the proposed model outperforms the existing formulation of E-Model by a margin of approximately 30% in terms of the prediction accuracy. Even though we have used WB-PESQ in this research, the proposed method is independent of it and requires a generic instrumental model of this kind. The methodology may also be augmented with auditory tests, which is one of our future goals too. Our models are already being used within Orange Labs for transmission planning and speech quality estimation.

We have also computed new values for *Bpl* for the case of WB-PESQ which are significantly different than those already reported in the standards (see Table 9). Although work has been done in the past to derive these values [13], we considered it important to derive new results that may be used for cross verification. It is worth mentioning, however, that the values reported in the standards are for auditory tests.

In future, we would like to extend our work to super-wideband contexts. We would also like to leverage from more recent developments in speech quality estimation, such

as P.OLQA. Inclusion of more network distortion conditions as well as a variety of new codecs shall also be our objective. We also aim to refine our machine learning based methodology by augmenting our algorithms with more recent developments in the field.

## REFERENCES


[1] *The E-Model, a Computational Model for Use in Transmission Planning*, ITU-T Recommendation G.107, Int. Telecommun. Union, Geneva, Switzerland, 2005.

[2] S. Möller, A. Raake, N. Kitawaki, A. Takahashi, and M. Wältermann, "Impairment factor framework for wide-band speech codecs," *IEEE Trans. Audio, Speech, Language Process.*, vol. 16, no. 6, pp. 1969–1976, Nov. 2006.

[3] *Methods for Subjective Determination of Transmission Quality*, ITU-T Recommendation P.800, Int. Telecommun. Union, Geneva, Switzerland, 1996.

[4] A. Raja, R. M. A. Azad, C. Flanagan, and C. Ryan, "A methodology for deriving VoIP equipment impairment factors for a mixed NB/WB context," *IEEE Trans. Multimedia*, vol. 10, no. 6, pp. 1046–1058, Oct. 2008.

[5] *Methodology for the Derivation of Equipment Impairment Factors From Instrumental Models*, ITU-T Recommendation P.834, Int. Telecommun. Union, Geneva, Switzerland, 2002.

[6] *Wideband Extension to Recommendation P.862 for the Assessment of wideband Telephone Networks and Speech Codecs*, ITU-T Recommendation P.862.2, Int. Telecommun. Union, Geneva, Switzerland, 2005.

[7] M. Wältermann, I. Tucker, A. Raake, and S. Möller, "Extension of the e-model towards super-wideband speech transmission," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Dallas, TX, USA, Mar. 2010, pp. 4654–4657.

[8] S. Voran, "Perception of temporal discontinuity impairments in coded speech-a proposal for objective estimators and some subjective test results," in *Proc. 2nd Int. Conf. Meas. Speech Audio Quality Netw.*, 2003, pp. 17–18.

[9] CCITT, *OPINE (Overall Performance Index Model for Network Evaluation)* (CCITT Blue Book, P11, Supplement No. 23), vol. 5. Geneva, Switzerland: ITU, 1989, pp. 281–299 and 318–324.

[10] V. Barriac, J. Y. Sout, and C. Lockwood, "Discussion on unified objective methodologies for the comparison of voice quality of narrowband and wideband scenarios," in *Proc. Workshop Wideband Speech Quality Terminals Netw. Assessment Predict.*, 2004. [Online]. Available: https://scholar.google.com.pk/scholar?q=Discussion+on+unified+objective+methodologies+for+the+comparison+of+voice+quality+of+narrowband+and+wideband+scenarios&btnG=&hl=en&as_sdt=0%2C5

[11] *The E-model: A Computational Model for Use in Transmission Planning. Amendment 1: New Appendix II—Provisional Impairment Factor Framework for Wideband Speech Transmission*, ITU-T Recommendation G.107 Amendment I, Int. Telecommun. Union, Geneva, Switzerland, 2005.

[12] *Wideband E-Model*, ITU-T Recommendation G.107.1, Int. Telecommun. Union, Geneva, Switzerland, Dec. 2012.

[13] S. Möller, N. Côté, V. Gautier-Turbin, N. Kitawaki, and A. Takahashi, "Instrumental estimation of E-model parameters for wideband speech codecs," *EURASIP J. Audio, Speech, Music Process.*, vol. 2010, no. 1, p. 782731, 2010.

[14] *Extension of the Methodology for the Derivation of Equipment Impairment Factors From Instrumental Models for Wideband Speech Codecs*, ITU-T Recommendation P.834.1, Int. Telecommun. Union, Geneva, Switzerland, Apr. 2009.

[15] L. Sun and E. C. Ifeachor, "Voice quality prediction models and their application in VoIP networks," *IEEE Trans. Multimedia*, vol. 8, no. 4, pp. 809–820, Aug. 2006.

[16] L. Zheng, L. Zhang, and D. Xu, "Characteristics of network delay and delay jitter and its effect on voice over ip (VoIP)," in *Proc. IEEE ICC*, vol. 1. Jun. 2001, pp. 123–126.

[17] *Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 and 6.3 kbit/s*, ITU-T Recommendation G.723.1, Int. Telecommun. Union, Geneva, Switzerland, Mar. 1996.

[18] *Pulse Code Modulation (PCM) of Voice Frequencies*, ITU-T Recommendation G.711, Int. Telecommun. Union, Geneva, Switzerland, Nov. 1988,

[19] J. Gordon, "Pareto process as a model of self-similar packet traffic," in *Proc. Global Telecommun. Conf.*, Red Bank, NJ, USA, Nov. 1995, pp. 2232–2236.

[20] C.-C. Wu, K.-T. Chen, C.-Y. Huang, and C.-L. Lei, "An empirical evaluation of VoIP playout buffer dimensioning in Skype, Google talk, and MSN messenger," in *Proc. 18th Int. Workshop Netw. Oper. Syst. Support Digit. Audio Video*, 2009, pp. 97–102.

[21] Z. Qiao, R. K. Venkatasubramanian, L. Sun, and E. C. Ifeachor, "A new buffer algorithm for speech quality improvement in VoIP systems," *Wireless Pers. Commun.*, vol. 45, no. 2, pp. 189–207, 2008.

[22] C. Hoene, H. Karl, and A. Wolisz, "A perceptual quality model intended for adaptive voip applications," *Int. J. Commun. Syst.*, vol. 19, no. 3, pp. 299–316, 2006.

[23] P. Počta, H. Melvin, and A. Hines, "An analysis of the impact of playout delay adjustments introduced by VoIP jitter buffers on listening speech quality," *Acta Acustica United Acustica*, vol. 101, no. 3, pp. 616–631, 2015.

[24] W. Jiang and H. Schulzrinne, "Modeling of packet loss and delay and their effect on real-time multimedia service quality," in *Proc. NOSSDAV*, Jun. 2000. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.22.8263&rep=rep1&type=pdf

[25] A. Raake, *Speech Quality of VoIP Assessment and Prediction*. Hoboken, NJ, USA: Wiley, 2006.

[26] C. Morioka, A. Kurashima, and A. Takahashi, "Proposal on objective speech quality assessment for wideband telephony," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Mar. 2004, pp. 49–52.

[27] A. Hines, P. Pocta, and H. Melvin, "Detailed comparative analysis of pesq and visqol behaviour in the context of playout delay adjustments introduced by voip jitter buffer algorithms," in *Proc. 5th Int. Workshop Quality Multimedia Exper. (QoMEX)*, Mar. 2013, pp. 18–23.

[28] *New Appendix IV—Provisional Planning Values for the Wideband Equipment Impairment Factor Ie,wb*, ITU-T Recommendation G.113, Int. Telecommun. Union, Geneva, Switzerland, Jun. 2006.

[29] *Perceptual Objective Listening Quality Assessment*, ITU-T Recommendation P.863, Int. Telecommun. Union, Geneva, Switzerland, Sep. 2014.

[30] A. Hines, J. Skoglund, A. Kokaram, and N. Harte, "Robustness of speech quality metrics to background noise and network degradations: Comparing ViSQOL, PESQ and POLQA," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May. 2013, pp. 3697–3701.

[31] M. A. Raja, S. Ali, and A. Mahmood, "Simulators as drivers of cutting edge research," in *Proc. 7th Int. Conf. Intell. Syst. Modelling Simulation (ISMS)*, Jan. 2016, pp. 114–119.

[32] *Coding of Speech at 8 kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP)*, ITU-T Recommendation G.72, Int. Telecommun. Union, Geneva, Switzerland, Jan. 2007.

[33] *7 kHz Audio Coding Within 64 kbit/s, Appendix III: A High-Quality Packet Loss Concealment Algorithm for G.722*, ITU-T Recommendation G.722, Int. Telecommun. Union, Geneva, Switzerland, Nov. 2006.

[34] *Pulse Code Modulation (PCM) of Voice Frequencies. Appendix 1: A High Quality Low-Complexity Algorithm for Packet Loss Concealment With G.711*, ITU-T Recommendation G.711, Int. Telecommun. Union, Geneva, Switzerland, Sep. 1999.

[35] *Software Tools for Speech and Audio Coding Standardization*, ITU-T Recommendation G.191, Int. Telecommun. Union, Geneva, Switzerland, Sep. 2005.

[36] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992.

[37] L. M. Howard and D. J. D'Angelo, "The GA–P: A genetic algorithm and genetic programming hybrid," *IEEE Expert*, vol. 10, no. 3, pp. 11–15, Jun. 1995.

[38] A. Topchy and W. F. Punch, "Faster genetic programming based on local gradient search of numeric leaf values," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, pp. 155–162. [Online]. Available: http://www.cs.bham.ac.uk/ wbl/biblio/gecco2001/d01.pdf

[39] E. M. Mugambi, A. Hunter, G. Oatley, and L. Kennedy, "Polynomial-fuzzy decision tree structures for classifying medical data," *Knowl.-Based Syst.*, vol. 17, nos. 2–4, pp. 81–87, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/B6V0P-4C4VYG9-2/2/8ee7c8541e99bf3c8c22922dad2ebfbf

[40] M. Keijzer, "Scaled symbolic regression," *Genet. Programm. Evolvable Mach.*, vol. 5, no. 3, pp. 259–269, Sep. 2004.

[41] W. B. Langdon and B. F. Buxton, "Genetic programming for mining DNA chip data from cancer patients," *Genet. Programm. Evolvable Mach.*, vol. 5, no. 3, pp. 251–257, Sep. 2004.

[42] A. Raja, R. M. A. Azad, C. Flanagan, and C. Ryan, "Real-time, non-intrusive evaluation of VoIP," in *Proc. 10th Eur. Conf. Genet. Programm.*, Apr. 2007, pp. 217–228.

[43] S. Luke and L. Panait, "Lexicographic parsimony pressure," in *Proc. Genetic Evol. Comput. Conf. (GECCO)*, New York, NY, USA, 2002, pp. 829–836.
[44] S. Gustafson, E. K. Burke, and N. Krasnogor, "On improving genetic programming for symbolic regression," in *Proc. IEEE Congr. Evol. Comput.*, vol. 1. Sep. 2005, pp. 912–919.
[45] *S. Silva, GPLAB A Genetic Programming Toolbox for MATLAB*, Univ. Coimbra, Portugal, Apr. 2007.
[46] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
[47] *Ammendment 2: Revised Appendix I—Provisional Planning Values for the Equipment Impairment Factor ie and Packet-Loss Robustness Factor Bpl*, ITU-T Recommendation G.113, Int. Telecommun. Union, Geneva, Switzerland, Jan. 2007.
[48] *Amendment 1: Revised Appendix IV—Provisional Planning Values for the Wideband Equipment Impairment Factor and the Wideband Packet Loss Robustness Factor*, ITU-T Recommendation G.113, Int. Telecommun. Union, Geneva, Switzerland, Mar. 2007.

**MUHAMMAD ADIL RAJA** received the B.Eng. degree in metallurgical engineering and materials science from the University of Engineering and Technology at Lahore, Lahore, Pakistan, in 2000, the M.S. degree in computer sciences from Lahore University of Management Sciences, Pakistan, and the Ph.D. degree in speech quality estimation from the University of Limerick, Ireland, in 2008. He was a Postdoctoral Researcher with Orange Labs, France, in 2010. He is currently an Associate Professor with the Department of Computer Science, Namal College, Mianwali, Pakistan. His research interests include the theory and applications of machine learning.

**ANNA JAGODZINSKA** is currently with Orange Labs, France. She is responsible for completing and carrying out projects from study and design to deployment and end-user experience feedback. She is also responsible for the coordination of internal teams and outsourcing. She is a peoples' manager and is involved in goal fixing, performance appraisal, skill set development and motivation. Her research interests include the design, development, and deployment of IT solutions for broadband service activation and after-sales.

**VINCENT BARRIAC** is currently the Vice Chairman of the ITU-T Study Group 12 and a Rapporteur for Q.15/12. He is also a Research and Development Engineer with Orange Labs, France. He is an Expert in voice quality assessment tools and methods. He has a profound involvement in the entire Orange Group, with outlets present in several European and African countries. As an expert, he serves as a Technical Support and an Adviser for operational entities about QoS supervision strategies and voice quality enhancements for both mobile and fixed services. His most recent activities concern quality assessment of telephony and video telephony services over 4G radio networks. His 20 year involvement in ITU-T has included contributing to the development of several standards in his area, such as P.561, P.564, P.862, P.863, and, more recently, as an Editor of G.1028.