CAQoE: A Novel No-Reference Context-aware Speech Quality Prediction Metric

RAHUL KUMAR JAISWAL, Department of Information and Communication Technology, Faculty of Engineering and Science, University of Agder, Norway
RAJESH KUMAR DUBEY, Department of Electrical Engineering, School of Engineering and Technology, Central University of Haryana, India

The quality of speech degrades while communicating over Voice over Internet Protocol applications, for example, Google Meet, Microsoft Skype, and Apple FaceTime, due to different types of background noise present in the surroundings. It reduces human perceived Quality of Experience (QoE). Along this line, this article proposes a novel speech quality prediction metric that can meet human's desired QoE level. Our motivation is driven by the lack of evidence showing speech quality metrics that can distinguish different noise degradations before predicting the quality of speech. The quality of speech in noisy environments is improved by speech enhancement algorithms, and for measuring and monitoring the quality of speech, objective speech quality metrics are used. With the integration of these components, a novel no-reference context-aware QoE prediction metric (CAQoE) is proposed in this article, which initially identifies the context or noise type or degradation type of the input noisy speech signal and then predicts context-specific speech quality for that input speech signal. It will have of great importance in deciding the speech enhancement algorithms if the types of degradations causing poor speech quality are known along with the quality metric. Results demonstrate that the proposed CAQoE metric outperforms in different contexts as compared to the metric where contexts are not identified before predicting the quality of speech, even in the presence of limited size speech corpus having different contexts available from the NOIZEUS speech database.

CCS Concepts: • **Computing methodologies** → **Machine learning**; *Machine learning approaches*; Neural networks;

Additional Key Words and Phrases: Classifier, deep neural network, speech enhancement, no-reference, speech quality, quality of experience (QoE), voice activity detector, VoIP

ACM Reference format:
Rahul Kumar Jaiswal and Rajesh Kumar Dubey. 2023. CAQoE: A Novel No-Reference Context-aware Speech Quality Prediction Metric. *ACM Trans. Multimedia Comput. Commun. Appl.* 19, 1s, Article 35 (January 2023), 23 pages.
https://doi.org/10.1145/3529394

Authors' addresses: R. K. Jaiswal (corresponding author), Department of Information and Communication Technology, Faculty of Engineering and Science, University of Agder, Grimstad, Norway, 4879; email: rahul.jaiswal@uia.no; R. K. Dubey, Department of Electrical Engineering, School of Engineering and Technology, Central University of Haryana, Mahendragarh, India, 123031; email: rajesh.dubey@cuh.ac.in.

1551-6857/2023/01-ART35 $15.00
https://doi.org/10.1145/3529394

1 MOTIVATION AND RELATED WORK

In the era of exponentially increasing number of internet users and mobile devices, the uses of various **Voice over Internet Protocol (VoIP)** applications, for example, Google Meet, Microsoft Skype, and Apple FaceTime, are growing dramatically. For fulfilling the users expectations to meet better **quality of experience (QoE)** while using such VoIP applications, it is necessary to measure and monitor real-time speech quality predictions at different nodes of speech communication networks. Traditionally, **Absolute Category Rating (ACR)** [1] method is used to measure speech quality, where the speech materials are played and speech quality ratings are provided by the subjects. Although it is highly reliable and provides an accurate assessment of speech quality, one needs to arrange number of subjects who will have no objections in listening the speech material for speech quality ratings. Moreover, it consumes more time, and thus speech quality monitoring in real time using this method is impractical. Alternatively, one can utilize objective speech quality assessment metrics conveniently, which can use different computational algorithms for speech quality measurement. Measuring and monitoring real-time speech quality at different nodes of speech communication networks using objective metrics are less costly, faster, and practical.

The **International Telecommunication Union (ITU)** has standardized various objective speech quality assessment metrics. For example, Instrumental metrics, as shown in Figure 1, are used to estimate the average user judgement of the quality of a service [2]. Signal-based metric employs received (degraded) speech signals for speech quality estimation. Signal-based metrics are of two types: full-reference metric (also called "intrusive" or "double-ended" metric) and no-reference metric (also called "non-intrusive" or "single-ended" metric).

Intrusive (reference-based) metrics usually calculate the distance between spectral representations of the transmitted reference signal and the received degraded signal. For example, Perceptual Evaluation of Speech Quality [3], Perceptual Objective Listening Quality Assessment [4], and ViSQOL [5] and its improved version ViSQOL v3 [6] are some of the popular intrusive metrics. Since one cannot have the access to the original input reference signal in most of the speech processing applications, the intrusive metrics are not appropriate for real-time speech quality monitoring at different nodes of communication networks.

Non-intrusive (no-reference) speech quality metrics are preferred for real-time monitoring of speech quality and the scenarios where a reference speech signal is unavailable. These metrics only use the degraded speech signal to predict the speech quality and could be easily installed at the end point of VoIP channels or at any nodes of communication networks for monitoring the quality of speech. For the assessment of narrow-band speech signals, two no-reference speech quality metrics are standardized [2]: first, the ITU recommended P.563 [7] and, second, the American National Standard Institute standardized "Auditory non-intrusive quality estimation plus (ANIQUE+) [8]." ANIQUE+ is a perceptual model that simulates the functional roles of human auditory system (that is, based on human perception of speech or audio) and deploys improved modelling of quality estimation by a statistical learning paradigm [8]. An implementation of ANIQUE+ is only commercially available while P.563 is publicly available and currently in force. Moreover, both ANIQUE+ and P.563 metrics are not context-aware or context-sensitive quality metrics.

The examples of non-standardized speech quality prediction metrics include **Low Complexity Speech Quality Assessment (LCQA)** [9] and **Deep Neural Network– (DNN)** based speech quality prediction metrics [10–14]. The LCQA algorithm deploys low complexity (execution time) to monitor the quality of speech over a network. It estimates the quality of speech by mapping the global statistical features obtained from speech codecs using Gaussian Mixture Model [15]. For each frame, the global features of speech like mean, variance, skewness and kurtosis are calculated from the parameters of speech-coding [16]. Moreover, LCQA is restricted only to the parametric

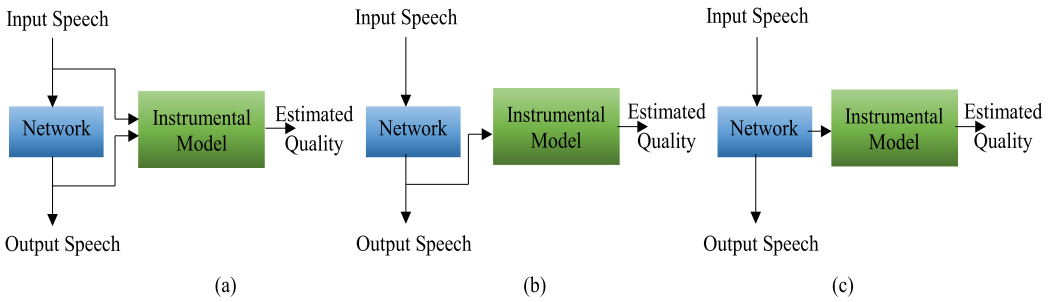

Fig. 1. Types of instrumental speech quality metrics: (a) Full-reference signal-based metric, (b) No-reference signal-based metric, and (c) Parametric metric.

representation of the input speech signal without its perceptual transform. The recent work on DNN-based speech quality prediction metrics include [10–14]. For example, Reference [12] predicts speech quality based on extracting Mel Frequency Cepstral Coefficents features and then training a DNN, and [13] predicts speech quality from the speech signal and then train a convolutional neural network. The author of Reference [14] deploys deep autoencoder and sub-band autoencoder features to train **artificial neural network (ANN)** on noisy speech samples for predicting quality of speech. Reported results indicate that these DNN-based metrics perform poorly in the scenarios of competing speaker type degradations. Also, these metrics directly predict the speech quality without identifying/classifying the context (noise class) of the input speech signal before predicting its speech quality.

Parametric metrics [17], for example, the E-Model [18], estimates the quality of speech using the network and the terminal parameters. Network delay and packet loss are the key components of network parameters. Jitter buffer overflow, coding distortions, jitter buffer delay, and echo cancellation are the key components of terminal parameters. Impairments of the received speech signal are predicted using these parameters, and then the rating factor is converted into **Mean Opinion Score (MOS)**.[1] Some recent study on speech quality monitoring using parametric metric includes wired, wireless, or atmospheric degradations in speech signals [19–23]. However, the E-model has limitations, that is, it cannot clearly represent non-linear relationship between perceptual characteristics of speech signal and network planning parameters due to the dynamic change in the speech signal characteristics. Moreover, the disadvantage of the parametric metric is that it does not involve the speech signal in the prediction of speech quality, and, therefore, it is not appropriate for predicting quality of speech based on the signal-noise characteristics [2]. For real-time measuring and monitoring of the quality of speech, a no-reference signal-based speech quality prediction metric is the most appropriate.

The motivation to propose a **Context-aware QoE Prediction Metric (CAQoE)** came to light while digging into the literature and knowing that there is no signal-based context-aware speech quality prediction metric. There are three main components in the proposed metric: (i) a context-classifier to classify the context of the speech signal (noise type), (ii) a **Voice Activity Detector (VAD)** to identify the voiced segments present in the noisy signals, and (iii) Context-specific **Speech Quality Estimation Model (CSQM)** to predict context-specific speech quality. To perfectly train the context-classifier and the CSQM, one needs to have a large size of noisy speech database. However, due to the availability of small size database of speech samples of different noise classes in the NOIZEUS speech corpus, we also have addressed the challenges of small size

[1]MOS describes the speech quality on a scale from 1 (bad) to 5 (excellent).

training samples database in building an accurate **machine learning (ML)** classifier for classifying the context of speech signal and also in training the CSQM for precisely estimating the quality of speech signal.

The rest of this article is organized as follows: The usage of machine learning and the concept of QoE in speech processing applications are presented in Section 2. Experimental dataset to evaluate speech quality is described in Section 3. Section 4 presents the detailed explanation of the proposed metric and its each associated component. The evaluation methodologies of each component and overall speech quality metric are described in Section 5. Section 6 presents the system setup to execute the program and the choices of different parameters and hyper-parameters. The results of each component and overall proposed metric are presented and discussed in Section 7. Section 8 presents the summary, which includes the key contributions, limitations of the proposed metric and the plans for the future work.

2 QUALITY OF EXPERIENCE AND MACHINE LEARNING

The degree of delight or annoyance of the user while using a particular application or service is defined as the QoE. It results from the fulfillment of the customer's expectations with respect to the utility and/or enjoyment of that application or service in light of their personality and current state [24]. There are various factors that influence the QoE while using VoIP applications for VoIP call, namely system, network, content, context of use, and user. The types of channel (mono or stereo), position of microphone, central processing unit overload, and so on, are the key factors of the service and system. Jitter, packet loss, and delay of the transmitted speech signal are the network factors. Content, that is, the characteristics of speech and voice, may be affected by processing and can influence the QoE. Location of using the service is the part of contextual factor, for example, in the noisy environments, such as a car, street, or train, compared to the noiseless and silence at home. Finally, the good quality is expected by the service users.

In view of this, a novel intelligent technique is necessary to design appropriate data-driven context-aware QoE metric that can perform real-time prediction of speech quality and can make its efficient utilization in speech quality monitoring. Using **artificial intelligence (AI)** and ML techniques, smart decision making AI-based algorithms can be introduced into the mobile devices to improve the QoE and the performance gains of the end-user. Moreover, the proposed speech quality prediction metric CAQoE can be easily deployed by the internet service providers for continuously measuring and monitoring the quality of service performance by detecting the impairments and potentially identifying the context (noise type). The potential root causes can be identified using this assistance, and then the QoE-aware management actions can be installed to react and maintain the end-user QoE levels [25].

3 THE SPEECH QUALITY EVALUATION DATASET

Different datasets have different applicabilities. For example, the ITU-T P.Supplement-23 database [26] contains the coded version of speech utterances used in the ITU-T 8 kbps codec characterization tests [27]. Experiment-1 examines the G.729 codec with coded speech samples, which are, thus, not useful for our experimentation. Experiment-2 investigates the effect of background noise for transmission quality but not having classified noise such as car, street, train noise, and so on. Also the method of assessment is comparison category rating not ACR, which is, thus, not suitable for this work. Experiment-3 investigates effects of channel degradations using coded speech samples and, thus, not useful for our experimentation. Therefore, the ITU-T P.Supplement-23 database is not suitable for testing our proposed metric due to the non-availability of different types/classes of degradations present in speech signal.

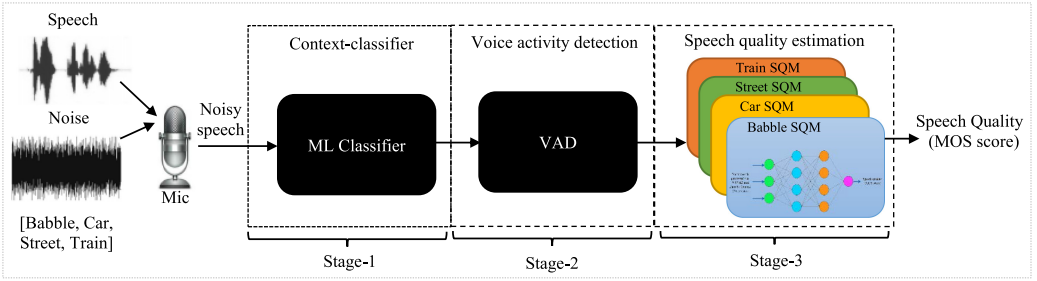

Fig. 2. Block diagram of the proposed context-aware speech quality prediction metric.

As different types of environmental noise degrades the speech, the performance of proposed context-aware speech quality prediction metric is investigated using noisy speech samples from a publicly available noisy dataset, that is, NOIZEUS [28]. Three male and three female speakers pronounced 30 phonetically balanced IEEE English sentences. Four real-world noises: babble, car, street, and train at two **signal-to-noise ratios (SNRs)**: 5 and 10 dB are used for degrading each sentences. Noises are taken from the AURORA database [29]. Each sentences are down-sampled from 25 kHz to 8 kHz, that is, narrow-band noisy speech samples. For down-sampling, the "resample" function resamples the input speech signal at p/q times the original sampling rate. It applies an FIR antialiasing low-pass filter to input speech signal and compensates for the delay introduced by the filter. The average duration of each utterance is 3 seconds. All the noisy speech samples are saved in .WAV format (16 bit PCM, mono).

4 PROPOSED CAQOE METRIC

The outline of the proposed context-aware speech quality prediction metric is shown in Figure 2. It is based on three main building blocks/stages: (i) a context/noise classifier for classifying the context of speech signal, that is, noise type; (ii) a VAD to separate the voiced and non-voiced part from the noisy speech signal; and (iii) context-specific speech quality models, which are trained using DNNs for each noise type for evaluating speech quality under that specific noise conditions. The hypothesis behind developing the proposed CAQoE metric is as follows: "Having the knowledge of the noise-type of the signal under test via classification, which can be routed to a speech quality assessment model that is trained and optimised for that particular noise degradation."

4.1 Context Classifier

The first component in the proposed metric is a context-classifier. Since different speech enhancement algorithms have different associated noise estimation techniques, resulting in varying performance in enhancing the noisy speech signals and degraded under different noise contexts, we apply this knowledge in building the context-classifier. With this knowledge, the metric processes each noisy speech signal with 12 standard speech enhancement algorithms [28, 31] as presented in Table 1. Each class of speech enhancement algorithm has different speech enhancement algorithms. For example, Adaptive filtering has WT, Wiener-as, AudSup; Spectral subtraction has RDC, MB; Statistical-model based has MMSE-SPU, logMMSE, logMMSE-ne, logMMSE-SPU, pMMSE; and Signal-subspace has KLT and pKLT [28, 31]. Including one original unprocessed noisy speech signal and 12 processed noisy speech signals, we obtain 13 variations of input signals. These 13 signals (12 processed + 1 unprocessed) are then passed to the P.563 metric [7] to obtain 13 different predictions of speech quality, called "MOS scores." These MOS scores are combined and then deployed as an input feature vector for training the ML classifiers. The flow diagram of the

Table 1. Classes of the Speech Enhancement Algorithms with Corresponding Noise Estimation Techniques [28, 31]

Class of SE Algorithms	No. of SE Algorithms	Noise Estimation Techniques
Adaptive filtering	3	Wiener filtering [32] and A-priori signal to noise estimate [33]
Spectral subtraction [34]	2	Adaptive gain averaging and reduced delay convolution [35]
Statistical model-based [36]	5	Minimum mean square error (MMSE) [37] & log-MMSE [38]
Signal-subspace [39]	2	Karhunen-Loeve transform [40]

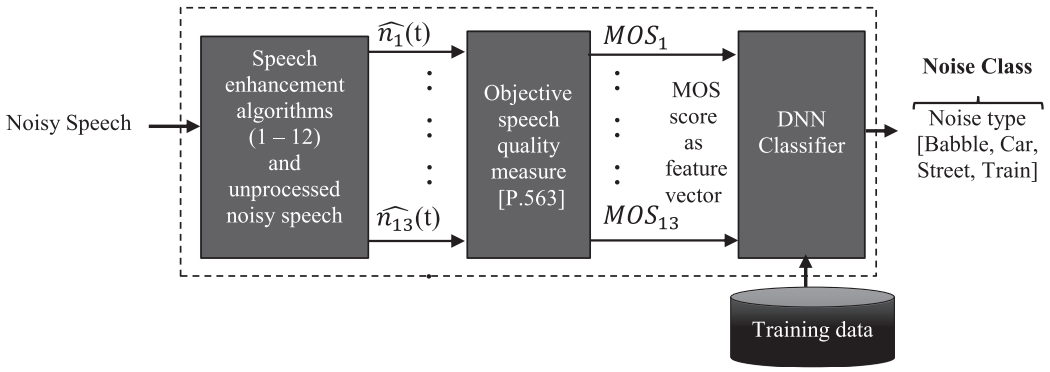

Fig. 3. Flow diagram of context-classifier showing feature extraction technique to train ML classifiers [30].

context-classifier demonstrating feature extraction technique is shown in Figure 3. The two sub components of the context-classifier and the used ML classifiers are discussed as follows:

4.1.1 Speech Enhancement. To enhance the degraded speech, **speech enhancement (SE)** algorithms are used. The selection of SE algorithms depends on the characteristics of the noise and the noise estimation techniques associated with it. The 12 standard SE algorithms as discussed in Section 4.1 consist of four classes of implementations. Each speech enhancement algorithm is developed by deploying a separate noise estimation algorithm, which are able to enhance the target speech with varying success as presented in Table 1.

4.1.2 No-Reference Speech Quality Metric (P.563). ITU standardized P.563 [7] metric was designed for assessing samples of active speech and is used for narrow-band speech signals. The P.563 metric is based on three principles [41], namely a physical model of vocal tract, a reconstruction of intermediate reference signal to assess unmasked distortions, and focusing on specific distortions, e.g., temporal clipping, robotization, and noise. Quality prediction by P.563 involves several steps: pre-processing, classification of dominant distortion class, and perceptual mapping. The degraded speech signal is pre-processed, which involves reverse filtering, speech level adjustment, identifying speech portions, and calculating speech and noise levels via a VAD [7]. Distortion classes include unnaturalness of speech, robotic voice, beeps, background noise, SNR, mutes, interruptions, and so on, extracted from the voiced parts. A dominant distortion class is then determined and mapped to a single mean opinion score referred to as a **Mean opinion score of objective**

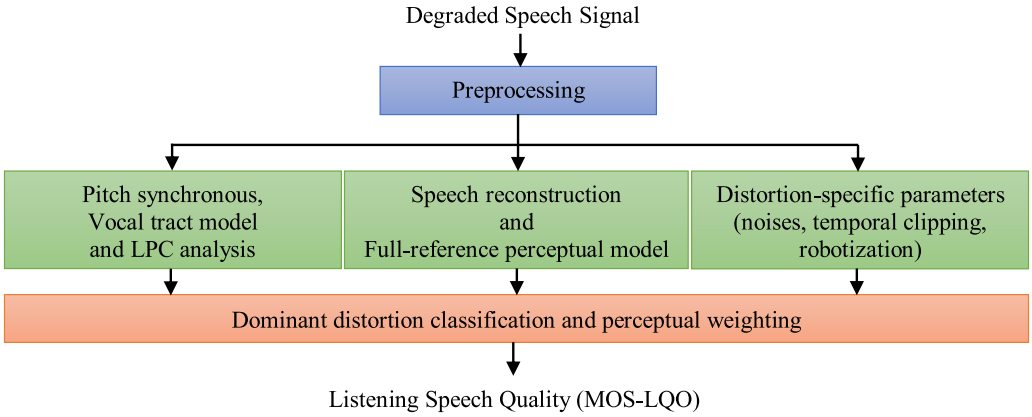

Fig. 4. Overall structure of P.563 metric [41].

Table 2. Requirements of Speech Signals in P.563 [7]

Sampling frequency	8,000 Hz
Amplitude resolution	16-bit linear PCM
Minimum active speech in file	3.0 seconds
Maximum signal length	20.0 seconds
Minimum speech activity ratio	25%
Maximum speech activity ratio	75%
Range of active speech level	−36.0 to −16.0 dBov

listening quality (MOS-LQO). Figure 4 shows the overall structure of the P.563 metric, and Table 2 presents the requirements of speech signals to be assessed in the P.563 metric.

We anticipate that the ML classifiers will be able to learn the relationship between the quality estimates of unprocessed and enhanced speech samples while correctly classifying the context of speech signal.

4.1.3 ML Classifiers Used. In our data-driven approach, a major challenge is to build a highly accurate and reliable classifier with fewer available speech samples that has the ability to distinguish the context (noise class) of the input signal based on the effective feature set. Therefore, we investigate seven different ML classifiers, which are described as follows.

(i) **XGBoost: (Extreme Gradient Boosting, known as XGBoost)** is an ensemble of classification and regression trees [42]. Gradient boosting[2] [43] is used to optimize the trees in XGBoost. Boosting is a sequential technique that works on the principle of an ensemble of weak learners to deliver improved accuracy. Let x being the input vector and w_q being the score of the corresponding leaf q, then tree output is as follows:

$$f(x) = w_q(x_i). \tag{1}$$

The ensemble of K trees lead to an output as [44]

$$y_i = \sum_{k=1}^{K} f_k(x_i). \tag{2}$$

[2]Gradient boosting is a boosting technique that minimises the loss function of the model by adding weak learners using gradient descent.

The objective function J at step t is minimised by the XGBoost algorithm as [44]

$$J(t) = \sum_{i=1}^{n} L\left(y_i, \hat{y_i}^{t-1} + f_t(x_i)\right) + \sum_{i=1}^{t} \Omega(f_i), \tag{3}$$

where the first part comprises the train loss function L (e.g., mean square error for regression and binary cross entropy for classification) between the actual class y and the predicted output $\hat{y}$ for n samples. The second part comprises the regularizer for controlling the complexity of the model and helping to avoid overfitting.

(ii) **Decision Tree (DT):** This is the combination of decision nodes and leaf nodes. Each decision node corresponds to a test X over a single attribute of the input data and has a number of branches, each of which handles an outcome of the test X. Each leaf node represents a class, that is, the result of decision for a case. Decision Tree is constructed based on the principle of divide and conquer [45, 46]. A set T of training data consists of k classes $(C_1, C_2, \ldots, C_k)$. If T only consists of one single class, then T will be a leaf. If T contains no case, then T is a leaf and the associated class with this leaf will be assigned with the major class of its parent node. If T contains more than one class, then a test based on some attribute a_i of the training data will be carried and T will be split into n subsets $(T_1, T_2, \ldots, T_n)$, where n is the number of outcomes of test over attribute a_i. The same process of constructing DT is recursively performed over each T_j; $1 \leq j \leq n$, until every subset belongs to a single class.

(iii) **Random Forest (RF):** It is the combination of several decision trees. Each tree gives a classification result and the forest chooses the class that has the highest votes [47]. The random forest classifier [48] uses the Gini index [49] as an attribute selection measure, which measures the impurity of an attribute with respect to the classes. For a given training set T, selecting one case at random and saying that it belongs to some class C_i, the Gini index can be written as follows:

$$\sum \sum_{j \neq i} (f(C_i, T)/|T|)(f(C_j, T)/|T|), \tag{4}$$

where $f(C_i, T)/|T|$ is the probability that the selected case belongs to the class C_i. For generating a random forest classifier, two parameters, namely; number of features used at each node to generate a tree and number of trees to be grown are needed.

(iv) **Logistic Regression (LR):** It predicts the presence or absence of a classifier outcome based on the values of set of predictor variables. It uses the logistic function to squeeze the output of a linear equation between 0 and 1 because for classification, the probabilities lie between 0 and 1. With $P(y = 1)$ being the probability of presence, and $\beta_0, \beta_1, \ldots, \beta_p$ being the coefficients of explanatory variables $x_0, x_1, \ldots, x_p$, LR model for p independent variables can be written as

$$P(y = 1) = \frac{1}{1 + \exp^{-(\beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p)}}. \tag{5}$$

(v) **K-nearest Neighbors (KNN):** In KNN, majority votes of class labels of neighbors are used to find the class of each test sample [47]. The most frequent class label in the first "k" training samples (nearest neighbors) are assigned as the class of the test sample, by computing and sorting the distances between the test sample and each training sample. For finding the nearest neighbors, Euclidean, Manhattan, and Minkowski distances are used [50].

(vi) **Support Vector Machine (SVM):** It works on the principle of structural risk minimization [51]. For classification, it finds an optimal separating hyperplane to maximize the margin between two classes of data. The margin is the sum of distances to the hyperplane from the closest points of two classes. Suppose that there are m instances of training data. Each

instance consists of an (x_i, y_i) pair where $x_i \in \mathbb{R}^N$ is a vector containing attributes of the ith instance, and $y_i \in \{+1, -1\}$ is the class label for the instance. The objective of the SVM is to find the optimal separating hyperplane $w.x + b = 0$ between the two classes of data. Quadratic programming optimization techniques are used to find optimal separating hyperplane.

(vii) **Naive Bayes (NB):** It is a probabilistic classifier that works on Bayes's theorem [52] with the assumption that features are independent given class. Despite this unrealistic assumption, the classifier is effective and is particularly suited when the dimensionality of inputs is high. Basically the NB classifier ignores the possible dependencies, namely correlations among the inputs and reduces a multivariate problem to a group of univariate problems.

4.2 Voice Activity Detection

The VAD is second component in the proposed metric. Based on the speech features, the VAD identifies the active voiced segments in the input noisy speech signal. It works as a pre-processing block to process the input noisy speech signal before sending it to the speech quality estimation metric. Our previous study [53] reflects that a **weighted spectral centroid (WS)** VAD performs outstanding in identifying and separating the voiced segments from the noisy speech signal.

The WS VAD extracts one of the speech feature, that is, spectral centroid from the noisy speech signal. Spectral centroid is barycenter of spectrum and its high value corresponds to the "brightness" of the sound. To design the WS VAD, one can divide the speech samples into overlapping frames of size 25 ms with a 10-ms shift and then compute short-time spectral centroid for each frame as [53]

$$C_i = \frac{\sum_{k=1}^{N}(k+1)X_i(k)}{\sum_{k=1}^{N}X_i(k)}, \tag{6}$$

where $X_i(k),\ k = 1, 2, \ldots, N$, is the Discrete Fourier Transform coefficients of the ith short-time frame of frame-length N. A smoothing filter, namely; median filter[3] [54] is applied throughout the feature sequence. Then, the histogram is computed and its local maxima are detected. The threshold T is computed dynamically using the weighted average between the first and second local maxima, given by $T = \frac{W_1M_1+W_2M_2}{W_1+W_2}$, where M_1 and M_2 are the first and second local maxima, respectively. W_1 and W_2 are the user-defined weights and are set as $W_1 = 5$ and $W_2 = 1$. For extracting voiced segments, a threshold is applied to each frame of signal.

4.3 Speech Quality Estimation

The third component in the proposed metric is the context-specific speech quality prediction metric, which is a collection of DNNs, each trained and optimised for a particular context (noise type). For building context-specific speech quality prediction metric, fully connected DNNs and lasso feature selections are used.

4.3.1 Fully connected DNN. A DNN as shown in Figure 5 is an ANNs, which consists of multiple number of hidden layers between the input and the output layer, each of which consisting of a linear operation followed by a pointwise non-linearity, also known as activation function. Consider a feed-forward DNN with L layers, labelled $l = 1, \ldots, L$ and each with a corresponding dimension q_l. The layer l is defined by the linear operation $\mathbf{W_l} \in \mathbb{R}^{q_{l-1} \times q_l}$ followed by a non-linear activation function $\sigma_l : R^{q_l} \rightarrow R^{q_l}$. Layer l receives input from the $l-1$ layer denoted as, $\mathbf{w_{l-1}} \in R^{q_{l-1}}$, the resulting output of the layer l, $w_l \in R^{q_l}$, is then computed as $\mathbf{w_l} := \sigma_l(W_l\mathbf{w_{l-1}})$, where $\sigma_l(\cdot)$ is the pointwise activation function. The final output of the DNN, w_L, is then related to the input w_0 by

[3]A median filter is a non-linear filtering technique used to remove noise from the signal [54].

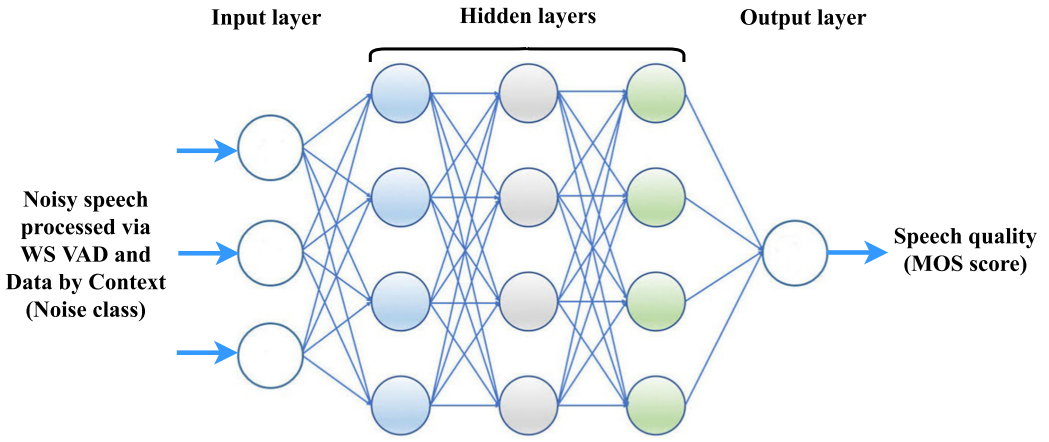

Fig. 5. A simple feed-forward DNN to design speech quality model (SQM) for each noise class.

propagating through the various layers of the DNN as $w_L = \sigma_L(W_L(\sigma_{L-1}(W_{L-1}(\ldots(\sigma_1(W_1 w_0)))))))$. The DNN learns the layerwise weights $w_1, w_2, \ldots, w_L$ [55].

The input to the designed DNN is the noisy speech signal processed via WS VAD (that is, silence separated speech signals) and the context-specific training data. It is important to note here that zero padding is used at the end of the processed speech samples to make the length of each processed speech sample same before feeding to the DNN. The output of the DNN is the subjective speech quality score (MOS-LQS). The hidden layer's activation function σ_l includes a **rectified linear unit (ReLU)**, defined as $\sigma_l(x) = 0$ for $x < 0$ and x for $x > 0$, and the output layer's activation function σ_o includes a linear activation function, defined as $\sigma_o(x) = cx$ where c is constant. The classifier, first, classifies the input noisy speech signal and then activates the corresponding trained DNN to evaluate that noisy speech signal, which provides a predicted speech quality (MOS score).

4.3.2 Lasso Feature Selection. To analyze and reduce the complexity of the data-driven models, high-dimensional feature selection technique, such as lasso [56, 57], is deployed, which separates relevant features from the irrelevant ones. The tuning parameters are adjusted properly to all aspects of the model using cross-validation. The combination of square error loss and $L1$ penalty on the regression coefficients are minimized by the lasso. With $\{x_n, y_n\}_{n=1}^{N}$ being the training data, b being the intercept, λ being the Lagrange multiplier, the loss function to be minimized, is given as [58]

$$\hat{\beta} \leftarrow \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{n=1}^{N} \left(y_n - b - \sum_{d=1}^{D} \beta_d x_{dn} \right)^2 \right\} + \lambda ||\beta||_1. \tag{7}$$

A competition between the regression coefficients is imposed by the $L1$ norm, leading to shrinking some of them to 0, and then producing a sparse model that can explain the data with less features. The λ is typically inferred from cross-validation. Fivefold cross-validation is used to perform this optimization.

5 EVALUATING THE SPEECH QUALITY METRIC

This section presents the evaluation methodology used for the proposed CAQoE metric and its components, which include context-classifier, WS VAD, and the context-specific speech quality estimation metric.

5.1 Evaluating the Context-classifier

The context-classifier takes 30 noisy speech samples of each noise type: babble, car, street and train at two SNRs: 5 and 10 dB from the NOIZEUS speech corpus. Therefore, each noise class/type has 60 (30×2) noisy speech samples, resulting in total 240 (30 samples $\times$ 4 noise types $\times$ 2 SNRs) noisy speech samples. These noisy speech samples are in the original form, that is, having both voiced and unvoiced segments. Each noisy speech sample is processed with 12 standard speech enhancement algorithms (see Table 1) and one sample remains present in the original unprocessed form. Further, these 12 processed samples along with the one original unprocessed sample, that is, $\hat{n_1}(t), \hat{n_2}(t), \ldots, \hat{n_{13}}(t)$, are injected to the P.563 metric (see Figure 3) to obtain 13 different objective speech quality prediction scores, that is, $\text{MOS}_1, \text{MOS}_2, \ldots \text{MOS}_{13}$. These 13 MOS scores are then combined and used as an input feature vector for training the ML classifiers for classifying the context of speech signal [30].

There are only 60 noisy speech samples in each noise class,[4] which is a very small amount of data (sample) for training any data-driven model, while training a ML classifier directly with these small amounts of data resulted in an classification accuracy of only 35%. Therefore, we switched our attention to perform one-vs.-all approach of multi-class classification, that is, binary classification with imbalanced datasets [59] for each noise class. Of four noise classes, we assigned the first noise class (e.g., Babble) as "class 1" and the remaining three noise classes as "class 0" and labelled it as "Babble." Similarly, we assigned the second noise class as "class 1" and the remaining three noise classes as "class 0" and labelled it as "Car." The same strategy is followed for the remaining noise classes. For balancing both the majority and minority noise classes, we reduced the size of the majority noise class (class 0) to be equal to the size of minority noise class (class 1) using an under-sampling technique [60, 61]. After balancing both noise classes, we divide the data of each noise class into a 80:20 ratio for training and testing the classifier.

F-score (FS), or test accuracy, and **geometric mean (G-mean)**, or balanced accuracy, are employed to measure the performance of ML classifiers. The FS is the weighted harmonic mean of **precision (PR)** and **recall (RC)**, given as [30]

$$PR = \frac{TP}{TP + FP}, \; RC = \frac{TP}{TP + FN}, \; FS = \frac{2(PR \times RC)}{(PR + RC)}. \tag{8}$$

G-mean [62] is the geometric average of the classification precision of the minority and the majority class. It evaluates the model's ability to correctly classify the minority and majority class and is given as [30]:

$$G - mean = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}}. \tag{9}$$

Here, TP, TN, FP, and FN are true positive, true negative, false positive, and false negative, respectively [30].

5.2 Evaluating the VAD

The VAD mask, which is the binary mask,[5] is obtained for noisy speech samples (framewise) using the WS VAD and compared to the **ground truth (GT)** mask[6] to obtain the TP, TN, FP, and FN, which measures the performance of a particular VAD using PR, RC, and FS as in our previous

[4] A noise class refers to noisy samples obtained by combining noisy samples at two SNRs 5 and 10 dB, e.g., Babble.

[5] Binary mask is binary decision taken by a VAD. If measured value exceeds a threshold, then VAD = 1, that is, voiced segment, else, VAD = 0, that is, noise/silence.

[6] The GT mask is the ideal binary mask, which is computed as silence, if the frame's sample value = 0; and voiced segments, otherwise; for the reference speech samples. This mask is representation of the perfect VAD.

study [53]. F-score measures accuracy and its maximum value is 1 (PR = RC = 1), which means that the voice activity decision of a certain VAD algorithm is equal to the reference transcription.

5.3 Evaluating the Speech Quality Metric

The success of DNN models in replicating different optimization-based speech quality solution relies heavily on the availability of sufficiently large learning datasets. However, getting large datasets to satisfy the data hungry nature of DNN models is not possible because most of the real data is publicly unavailable and/or costly to obtain. One alternative is to generate realistic data using advanced deep learning techniques, such as **Generative Adversarial Networks (GANs)**[7] [64, 65]. However, that is beyond the scope of this research. The original noisy speech samples are processed via the WS VAD to obtain the voiced segments of the speech samples as the voiced segments contribute in speech quality prediction only [53]. Further, the processed speech samples of each noise class via WS VAD and the training data of each noise class (see Figure 5) are fed as the input to the DNNs. The subjective prediction scores (MOS-LQS) of each noise class obtained from Dubey [66] are fed as the output to the DNNs. A collection of DNNs (see Figure 5) are trained and optimised for each noise class to obtain noise-specific speech quality prediction metric. The motivation for exploiting DNNs is primarily due to its universal approximation capability [67] and supplemented by the fact that trained DNN models are computationally very simple [68] to execute.

To extract most appropriate features for training the DNNs of each noise class, lasso feature selection technique is deployed. The layers of DNNs are densely connected. We experimented with various combinations of layers and weights to achieve best DNN models of each noise class that could be trained in the reasonable time. For the performance evaluation of the DNNs, the training and testing **mean square error (MSE)** of each noise class are calculated. MSE is the average of the square of the differences between the actual and the predicted quality scores. With n being the number of speech samples, $y_1, y_2, \ldots, y_n$ and $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n$ being the actual and the predicted quality scores, respectively, MSE is given as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2. \tag{10}$$

5.4 Evaluating the Overall Proposed CAQoE Metric

The performance evaluation of the proposed CAQoE metric is quantified in terms of the **Pearson correlation coefficients (PCC)**, **Spearman's correlation coefficients (SCC)**, and **root mean square error (RMSE)** between the objective predicted quality scores (MOS-LQO), obtained by the proposed speech quality metric for each noise class and their corresponding subjective listener quality scores (MOS-LQS). With n being the number of speech samples used in evaluation, $y_1, y_2, \ldots, y_n$ and $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n$ being the subjective and the predicted speech quality scores, respectively, μ_y and $\mu_{\hat{y}}$ being the mean of subjective and predicted speech quality scores, respectively, and d_i being the difference between ranks of the subjective and the predicted speech quality scores, these measures are given by Equations (11), (12), and (13), respectively, as

$$PCC = \frac{\sum_{i=1}^{n}(\hat{y}_i - \mu_{\hat{y}})(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \mu_{\hat{y}})^2 (y_i - \mu_y)^2}}. \tag{11}$$

[7]GANs is a new class of generative methods for data distribution learning where the objective is to learn a model that can generate samples close to the target distribution [63, 64].

$$SCC = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}. \tag{12}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}. \tag{13}$$

6 SYSTEM SETUP

This section presents the simulation platform used and different parameters and hyper-parameters selected for the classifiers and the DNN models trained to obtain the speech quality estimation metric. The VAD program is implemented in MATLAB 2018a. The standard objective speech quality metric (P.563) is programmed in "C" language. The ML classifiers and the DNN-based **speech quality metrics (SQMs)** are implemented in Python 3.7.3 with TensorFlow 2.2.0 on Windows 10 laptop having Intel Core i5 8th generation processor, Intel UHD Graphics 620, and 16 GB of memory.

6.1 Choice of Parameters and Hyper-Parameters

To obtain accurate context-classifier for classifying the context of speech signal and to build robust DNN-based SQM trained for each noise class separately, we experiment with different choices of parameters and hyper-parameters. We also perform grid search to obtain optimal parameters.

For the tested ML classifiers, namely XGBoost, Gradient Boosting, DT, RF, LR, KNN, SVM, and NB, the choices of parameters and hyper-parameters are outlined in Table 3.

To obtain robust DNN-based SQM for each noise class, we train DNNs with different number of hidden layers, each having a variable number of neurons. ReLU is used as the activation function to the hidden layers and linear to the output layer. The weights of DNN models are initialized randomly. MSE is used as the loss function and RMSprop and NADAM [69] optimizer for the stochastic optimization of DNNs. We select the proper learning rate of the optimizer and employ different sizes of mini-batch for each SQM. To minimize the overfitting, and to speed up the training, dropouts [70] of different values are used after each hidden layer. We standardize dataset by taking the mean and scaling to unit variance. The choices of parameters and hyper-parameters of DNN-based speech quality metrics for each noise class are presented in Table 4.

6.2 Training Stage

For training the DNN-based SQM model, we divide the data of each noise class into a 80:20 ratio, that is, we use 80% data/sample for the training and 20% data for testing of DNNs. We use the entire training data to optimize the weights of each DNN models. MSE between the predicted quality scores and the output of the network (subjective prediction scores) is used as the cost/loss function. We employ an efficient implementation of mini-batch stochastic gradient descent, called the RMSprop algorithm for optimization, which divides the gradient by a running average of its recent magnitude [71]. We also employ Nadam [69] optimizer, which is the same as Adam optimizer [72] with Nesterov momentum. The choice of optimizers used for each SQM is presented in Table 4.

6.3 Testing Stage

During testing, the test samples of each noise class are passed to the seven trained classifies for automatically (correctly) recognizing that test sample, that is, from the class it belongs to. After recognizing its class, the test sample is passed to the corresponding trained and optimized SQM to obtain the optimized objective speech quality predictions (MOS-LQO). Further, we also collect

Table 3. Parameters and Hyper-parameters of Classifiers for Each Noise Class

Babble noise class						
XGBoost	DT	RF	LR	KNN	SVM	NB
Estimators = 200	Split = entropy	Ests = 100	Regul = L2	Neighbors = 7	Penalty = 2	var. = 2
Depth = 6	criterion	Depth = 1	Penalty = 1	Weights = uni.	Kernel = rbf	smooth
Child weight = 0.0606	Depth = 1	Split = gini	Solver = liblin.	Algo = bf	Degree = 3	
Subsample = 0.181818	Splitter = best	criterion	Weight = bal.	Dis = Eucl.	Weight = bal.	
Learning rate = 1		Weight = bal.	Iters = 100	Leaf size = 30	Iters = 10	
Car noise class						
XGBoost	DT	RF	LR	KNN	SVM	NB
Estimators = 200	Split = entropy	Ests = 100	Regul = L2	Neighbors = 7	Penalty = 1	var. = 2
Depth = 6	criterion	Depth = 3	Penalty = 1	Weights = uni.	Kernel = rbf	smoot
Child weight = 1.0	Depth = 3	Split = entropy	Solver = lbfgs	Algo = auto	Degree = 3	
Subsample = 0.188889	Splitter = best	criterion	Weight = 1.0	Dis. = Eucl.	Weight = 1.0	
Learning rate = 0.5		Weight = 1.0	Iters = 100	Leaf size = 30	Iters = 10	
Street noise class						
Gradient Boosting	DT	RF	LR	KNN	SVM	NB
Estimators = 100	Split = entropy	Ests = 100	Regul = L2	Neighbors = 7	Penalty = 1	var. = 2
Val. frac. = 0.2	criterion	Depth = 1	Penalty = 1	Weights = uni.	Kernel = rbf	smooth
Iters. no change = 1000	Depth = 3	Split = gini	Solver = lbfgs	Algo. = auto	Degree = 3	
Tolerance = 0.01	Splitter = best	criterion	Weight = 1.0	Dis. = Eucl.	Weight = 1.0	
Learning rate = 0.2		Weight = 1.0	Iters = 100	Leaf size = 30	Iters = 50	
Train noise class						
Gradient Boosting	DT	RF	LR	KNN	SVM	NB
Estimators = 100	Split = entropy	Ests = 100	Regul = L2	Neighbors = 7	Penalty = 2	var. = 2
Val. frac. = 0.2	criterion	Depth = 1	Penalty = 1	Weights = uni.	Kernel = rbf	smooth
Iters. no change = 100	Depth = 3	Split = gini	Solver = lbfgs	Algo. = auto	Degree = 3	
Tolerance = 0.01	Splitter = best	criterion	Weight = 1.0	Dis. = Eucl.	Weight = 1.0	
Learning rate = 2		Weight = bal.	Iters = 100	Leaf size = 30	Iters = 10	

Table 4. Parameters and Hyper-parameters for Each DNN-based SQMs

Default to all SQMs		Babble SQM	Car SQM	Street SQM	Train SQM
Input layer neurons = 10	Neurons	13,3,10	13,3,10	13,2,10	13,3,10
Number of hidden layers = 3	Dropout	0.15,0.15,0.15	0.2,0.2,0.2	0.1,0.1,0.1	0.2,0.2,0.2
Hidden layers act. fun. = ReLU	Optimizer	Nadam	RMSprop	RMSprop	Nadam
Output layer neurons = 1	Learning rate	0.0008	0.01	0.01	0.0008
Output layer act. fun. = Linear	No. of epochs	1100	130	220	1900
Weight initialization = Random	batch size	5	10	10	10
Max. no. of iterations = 100,000 of LassoCV					

all test samples together and pass it to the seven trained classifiers for detecting its noise class automatically and then to obtain corresponding objective speech quality predictions (MOS-LQO).

7 RESULTS AND DISCUSSIONS

This section presents the numerical results, which fits into our proposed metric and offers solution to the gap discovered in the literature. It also discusses the results of each component to showcase the effectiveness of the proposed metric.

Table 5. F-score of Classifiers for Each Noise Class

Classifier Noise class	XGBoost	DT	RF	LR	KNN	SVM	NB
Babble	0.95	0.50	0.66	0.54	0.70	0.54	0.54
Car	0.91	0.62	0.62	0.62	0.70	0.62	0.54
Street	0.95	0.62	0.70	0.66	0.66	0.70	0.66
Train	0.91	0.54	0.66	0.75	0.62	0.66	0.54
Average	0.93	0.57	0.66	0.64	0.67	0.63	0.57

Table 6. G-Mean of Classifiers for Each Noise Class

Classifier Noise class	XGBoost	DT	RF	LR	KNN	SVM	NB
Babble	0.95	0.27	0.64	0.54	0.70	0.52	0.50
Car	0.91	0.50	0.62	0.61	0.67	0.55	0.45
Street	0.95	0.50	0.67	0.66	0.66	0.64	0.57
Train	0.91	0.39	0.66	0.74	0.61	0.66	0.39
Average	0.93	0.41	0.64	0.63	0.66	0.59	0.47

7.1 Context-classifier Response

The F-score and G-mean of each classifier for each noise class are presented in Tables 5 and 6, respectively. Both tables reflect that XGBoost[8] classifier is performing better and has an average test accuracy and balanced accuracy of 93% for each noise class, which is the highest among all the classifiers. Therefore, XGBoost classifier is used further to develop the complete proposed metric.

7.2 VAD Response

Figure 6 illustrations a test clean speech sample `sp12.wav` having "The drip of the rain made a pleasant sound," pronounced by a male speaker and the corresponding noisy speech `sp12_train_sn5.wav` degraded by train noise at 5 dB and its framewise VAD masks computed from different VAD algorithms, namely **Energy (E)** VAD, **Weighted energy (WE)**[9] VAD, and WS VAD, as compared to the GT mask. It can be seen that E and WE VAD are under-estimating the speech components, resulting in poor accuracy as compared to the GT VAD. However, WS VAD observes Precision of 1, Recall of 0.98, and F-score of 0.99, resulting in the best detection/separation of the speech and the non-speech segments in the input noisy speech signal.

The performance of each VAD algorithm is also measured on speech samples of each noise class of the NOIZEUS dataset at two SNRs: 5 and 10 dB to investigate their effectiveness. The Precision, Recall, and F-score are calculated for each noise degradation and the F-score is presented in Table 7. It can be noticed that the E and WE VAD are inaccurately identifying the voiced segments as compared to the GT VAD, resulting in poor accuracy. However, WS VAD performs outstanding in correctly identifying the speech components for each noise class/degradation type with its higher F-score, resulting in best accuracy among all other VADs. Hence, WS VAD is highly robust to pre-process the speech samples in our proposed metric. We integrate WS VAD further for developing the complete metric.

[8]XGBoost is used for babble and car and gradient boosting is used for street and train noise class.
[9]Energy and weighted energy VAD are developed by extracting the energy features of the speech samples [53].

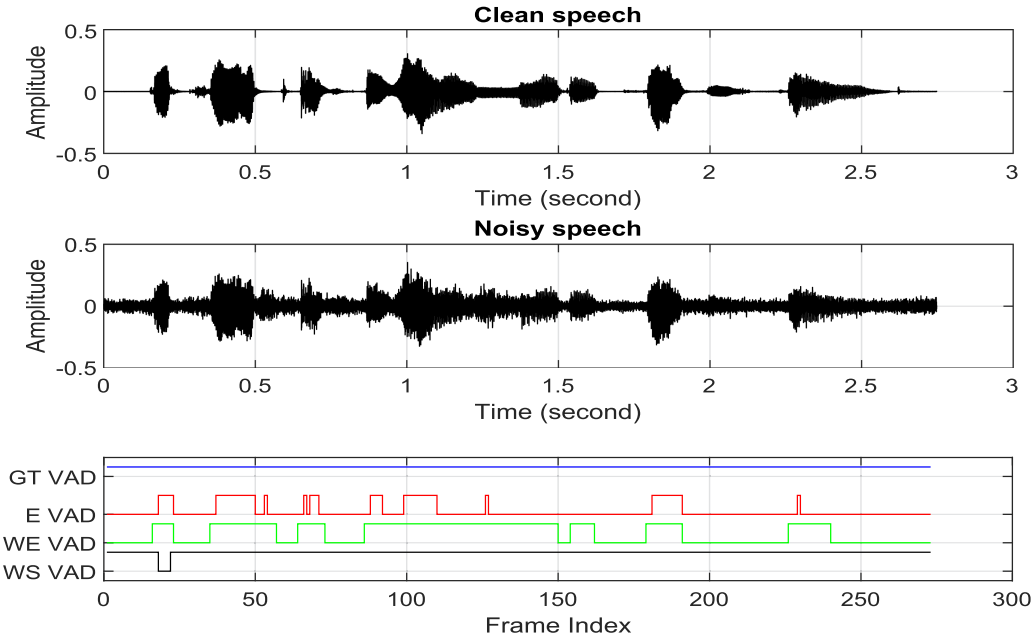

Fig. 6. Illustration of clean speech, noisy speech and framewise VAD masks compared to the GT mask for the test sample.

Table 7. F-score of VAD for Each Noise Class at Two SNRs

VAD	Babble		Car		Street		Train	
	5 dB	10 dB	5 dB	10 dB	5 dB	10 dB	5 dB	10 dB
GT	1	1	1	1	1	1	1	1
E	0.293	0.265	0.296	0.268	0.300	0.266	0.298	0.265
WE	0.571	0.500	0.530	0.503	0.585	0.517	0.565	0.510
WS	0.931	0.950	0.958	0.957	0.954	0.963	0.971	0.960

Table 8. Model Learning for Each SQM

MSE	Babble SQM	Car SQM	Street SQM	Train SQM
Train	0.016	0.104	0.108	0.032
Test	0.031	0.138	0.109	0.063

7.3 Speech Quality Metric Response

The training and testing errors obtained while training the DNN models of each speech quality metric are presented in Table 8. It can be observed that the accuracy, that is, the test MSE of each speech quality metric is comparable to its counter training part. All speech quality predictions of test samples can be estimated with a small error in the range of 0.03 to 0.13. However, the errors between the training and testing quality estimates are significantly different for individual SQMs. Thus, the results imply a better quality predictions for our SQMs.

Figure 7 shows the accuracy in terms of MSE for each SQM. It can be visualized that the individual model learning is converging toward the local minima, that is, with the increase in number of training epochs, there is the decrease in the training loss. The testing model follows and achieves it,

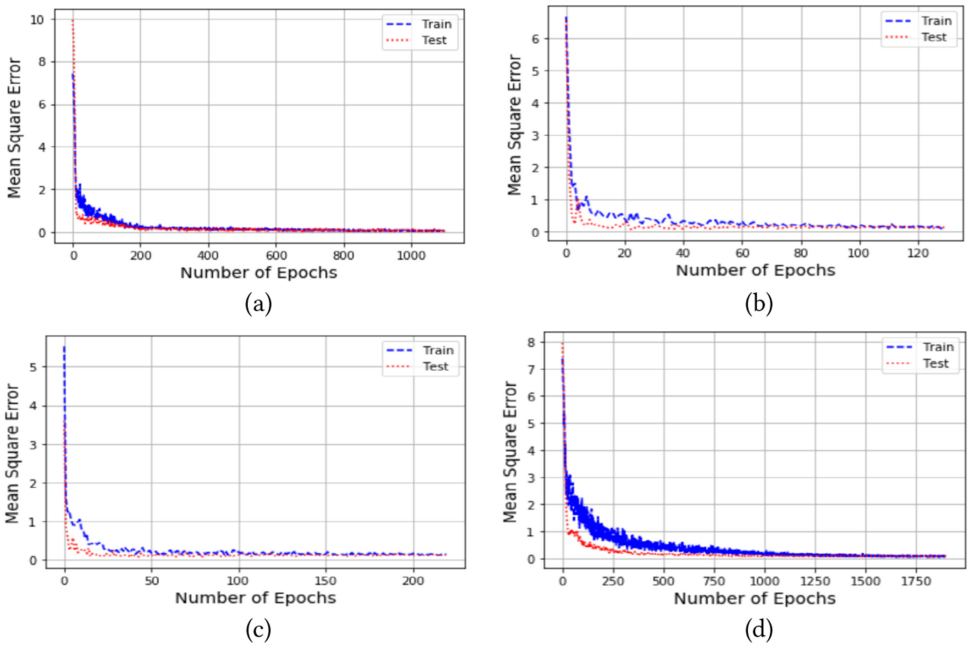

Fig. 7. Accuracy in terms of MSE for each speech quality metric (a) Babble SQM, (b) Car SQM, (c) Street SQM, and (d) Train SQM.

resulting in optimized accuracy. Therefore, these context-specific optimized speech quality models (SQMs) can be combined in our proposed metric.

7.4 Overall Proposed Metric Response

This section presents PCC, SCC, and RMSE between the objective quality scores (MOS-LQO) obtained for each noise class and their corresponding subjective listener quality scores (MOS-LQS). Three different scenarios are tested to investigate the effectiveness of our proposed metric.

7.4.1 Scenario A: Proposed Metric without Context-classifier. The context-classifier is removed here, that is, the context (noise class) of the input noisy speech samples are not classified. The input noisy speech samples are directly injected to the WS VAD and then the processed samples are passed through P.563 metric to obtain the objective quality scores (MOS-LQO). Table 9 presents PCC, SCC, and RMSE for each noise class in this scenario.

7.4.2 Scenario B: Proposed Metric without Context-classifier and VAD. The context-classifier and the VAD both are removed here, that is, neither the context (noise class) of the input noisy speech samples are classified nor the input noisy speech samples are processed by the VAD. The input noisy speech samples are directly passed through P.563 metric to obtain the objective quality scores (MOS-LQO). Table 10 presents PCC, SCC, and RMSE for each noise class in this scenario.

7.4.3 Scenario C: Proposed CAQoE Metric. This is our proposed CAQoE metric scenario where both context-classifier and VAD are present, that is, the context (noise class) of the input noisy speech samples are first classified and then processed with a VAD. Thereafter, it is passed through P.563 metric to obtain objective quality scores (MOS-LQO). Table 11 presents PCC, SCC, and RMSE for each noise class in this scenario.

Table 9. PCC, SCC, and RMSE for Each Noise Class with a Grouped Results for All Noise Classes without Context-classifier

Noise class →	Babble	Car	Street	Train	All
PCC	0.456	0.451	0.587	0.539	0.509
SCC	0.472	0.449	0.625	0.532	0.522
RMSE	0.733	0.844	0.833	0.772	0.797

Table 10. PCC, SCC, and RMSE for Each Noise Class with a Grouped Results for All Noise Classes without Context-classifier and VAD

Noise class →	Babble	Car	Street	Train	All
PCC	0.463	0.468	0.599	0.537	0.516
SCC	0.506	0.491	0.645	0.523	0.536
RMSE	0.677	0.858	0.842	0.738	0.782

Table 11. PCC, SCC, and RMSE for Each Noise Class with a Grouped Results for All Noise Classes in Proposed Metric

Noise class →	Babble	Car	Street	Train	All
PCC	0.995	0.803	0.943	0.927	0.833
SCC	0.974	0.785	0.799	0.983	0.884
RMSE	0.043	0.311	0.349	0.190	0.254

It can be observed from Tables 9, 10, and 11 that the PCC, SCC, and RMSE of the proposed metric (Scenario C) are better than the Scenario A and Scenario B for each noise class tested and for all noise classes together. The PCC and SCC are highest and the RMSE is lowest for our proposed metric. This reflects that our proposed metric is performing outstanding as compared to these two different baseline scenarios. The usages of both the context-classifier for classifying the context of speech signal (noise class) and the VAD for separating the voiced segments, are the key components of the proposed metric. In particular, it is the context sensitivity that is leading to the better results.

Further, it can be observed from Table 11 of our proposed CAQoE metric that the PCC and SCC of babble class are high and RMSE is less among all other noise classes. The train class also has high PCC and SCC. The car class is having around 80% PCC and SCC, with little high RMSE, which is in the acceptable range. The PCC of street class is also high. The overall proposed metric performs well, giving around 85% average accuracy in terms of correlations with all noise classes tested together. An improvement of around 40 % accuracy in terms of correlations can be observed as compared to both baseline scenarios.

To explore the impact of the proposed metric, the scatter plot for the objective speech quality predictions obtained from each speech quality model trained by each noise class (denoted by MOS-LQO) is compared to the subjective speech quality results (denoted by MOS-LQS) in Figure 8. A good correlation can be observed here for each noise class.

The overall simulation results of the proposed CAQoE metric show that the deep learning-based approach has a great advantage when the speech signals are degraded by various types of degradations. It demonstrates that DNNs have the great capability to remember and analyze the complicated characteristics of the speech signals. Therefore, we believe that our proposed CAQoE metric can be deployed by the internet service providers for measuring and monitoring real-time speech quality in the environments where the speech quality is degraded due to the presence of different

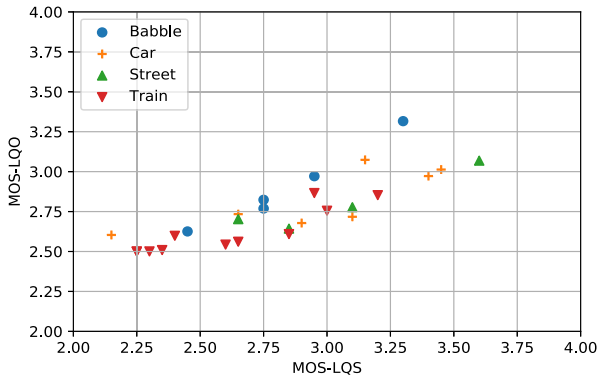

Fig. 8. Subjective vs. objective quality prediction of each noise class.

types of background noises and then, QoE-aware management actions can be installed to react and maintain the end-user QoE levels.

8 SUMMARY

In this article, we have proposed a framework for context (noise class) detection-based speech quality prediction metric (CAQoE) for measuring and monitoring real-time quality of speech in VoIP applications.

8.1 Key Contributions

The key contributions of the proposed work include the following:

- Presenting the speech quality monitoring problem using a context-aware speech quality metric from noisy speech samples.
- Using a three-step process for speech quality monitoring problem: (1) to obtain context-classifier; (2) to perform pre-processing using VAD; and (3) to develop context-specific speech quality metric. All three steps have separate relevance and can be integrated to develop context-aware speech quality monitoring metric.
- Developing a context-classifier using a novel feature extraction technique comprising speech enhancement algorithms and objective speech quality metric (P.563).
- Developing a VAD algorithm to detect the presence of active speech segments.
- Developing a group of DNNs that are trained and optimized for specific noise classes, that is, context sensitive.
- Classifying the context (noise class) of the input speech signal and then switching to a specific SQM can result in better speech quality prediction and it will allow the end-user to perceive a better QoE over VoIP applications.
- CAQoE experiments on NOIZEUS speech corpus containing different noise degradations demonstrating the advantages of the proposed technique.
- The effectiveness of the proposed metric is validated through the correlation and RMSE between the subjective and objective speech quality predictions, which is high and less, respectively, for each noise class.
- Analysis of the CAQoE metric reflects that detecting the context of input speech samples by the context-classifier and then passing through the corresponding SQM, results in better correlation with the subjective speech quality.

- The proposed three-stage CAQoE framework shows improved performance and a significant advantage over the simple speech quality metrics where the speech quality is predicted directly without identifying the context (noise class).

8.2 Limitations

Some limitations of the proposed work include the following:

- The proposed approach is empirical. The model is proposed and analyzed based on the performance metrics.
- The proposed model is investigated with less number of narrow-band speech signals available from the NOIZEUS corpus.

8.3 Future Works

The current work constitutes a preliminary step toward understanding the ability of the DNNs for this type of problems. Many interesting questions can be addressed in the future and some of them are listed below:

- Extending dataset further to produce even more better results.
- Developing dataset for different contexts (noise classes) and for noise coming from multiple sources. It could be used to validate the proposed VoIP monitoring metric.
- Performing subjective listening test to obtain subjective MOS score to compare the proposed metric performance.
- The DNN model should have a good generalization ability in real-world applications, such that when one deploys the model online and the conditions do not match completely with the noisy speech class data then it should still work efficiently. An initial experiment is performed in this article to demonstrate the generalization ability of the DNN model with respect to some parameters of the noisy speech data. More experiments, rigorous analysis and comprehensive discussions are the part of future considerations.

ACKNOWLEDGMENT

The author thanks Dr. Andrew Hines, University College Dublin, Ireland, for his valuable suggestions. The author is also thankful to the anonymous reviewers whose valuable suggestions helped in improving the quality of the manuscript.

REFERENCES

[1] ITU-T Rec. P.800: Methods for subjective determination of transmission quality. International Telecommunication Union, Geneva, 1996.

[2] Sebastian Möller, Wai Yip Chan, Nicolas Côté, Tiago H. Falk, Alexander Raake, and Marcel Wältermann. 2011. Speech quality estimation: Models and Trends. *IEEE Sign. Process. Mag.* 28, 6 (2011), 18–28.

[3] Antony W. Rix, John G. Beerends, Michael P. Hollier, and Andries P. Hekstra. 2001. Perceptual evaluation of speech quality (PESQ)- a new method for speech quality assessment of telephone networks and codecs. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 2. 749–752.

[4] ITU-T Rec. P.863: Perceptual objective listening quality assessment (POLQA). International Telecommunication Union, Geneva, 2011.

[5] Andrew Hines, Jan Skoglund, Anil C. Kokaram, and Naomi Harte. 2015. ViSQOL: An objective speech quality model. *EURASIP J. Aud. Speech Mus. Process.* 2015, 1 (2015), 1–18.

[6] Michael Chinen, Felicia S. C. Lim, Jan Skoglund, Nikita Gureev, Feargus O'Gorman, and Andrew Hines. 2020. ViSQOL v3: An open source production ready objective speech and audio metric. In *Proceedings of the 12th International Conference on Quality of Multimedia Experience (QoMEX'20)*. IEEE, 1–6.

[7] 2004. ITU-T Recommendation P.563: Single-ended Method for Objective Speech Quality Assessment in Narrow-band Telephony Applications.

[8] Doh Suk Kim and Ahmed Tarraf. 2007. ANIQUE+: A new American National Standard for non-intrusive estimation of narrow-band speech quality. *Bell Labs Techn. J.* 12, 1 (2007), 221–236.
[9] Stefan Bruhn, Volodya Grancharov, and Willem Bastiaan Kleijn. 2012. Low-complexity, Non-intrusive Speech Quality Assessment. (June 2012). US Patent 8,195,449.
[10] Jasper Ooster, Rainer Huber, and Bernd T. Meyer. 2018. Prediction of perceived speech quality using deep machine listening. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH'18)* (2018), 976–980.
[11] Szu-Wei Fu, Yu Tsao, Hsin-Te Hwang, and Hsin-Min Wang. 2018. Quality-Net: An end-to-end non-intrusive speech quality assessment model based on BLSTM. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH'18)*.
[12] Anderson R. Avila, Hannes Gamper, Chandan Reddy, Ross Cutler, Ivan Tashev, and Johannes Gehrke. 2019. Non-intrusive speech quality assessment using neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'19)*. 631–635.
[13] Andrew A. Catellier and Stephen D. Voran. 2020. Wawenets: A no-reference convolutional waveform-based approach to estimating narrowband and wideband speech quality. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'20)*. IEEE, 331–335.
[14] Meet H. Soni and Hemant A. Patil. 2021. Non-intrusive quality assessment of noise-suppressed speech using unsupervised deep features. *Speech Communication* 130 (2021), 27–44.
[15] Haojun Wu, Yong Wang, and Jiwu Huang. 2017. Identification of reconstructed speech. *ACM Trans. Multimedia Comput. Commun. Appl.* 13, 1 (2017), 1–20.
[16] Volodya Grancharov, David Yuheng Zhao, Jonas Lindblom, and W. Bastiaan Kleijn. 2006. Low-complexity, non-intrusive speech quality assessment. *IEEE Trans. Aud. Speech Lang. Process.* 14, 6 (2006), 1948–1956.
[17] Haemin Yang, Kyungguen Byun, Hong Goo Kang, and Youngsu Kwak. 2016. Parametric-based non-intrusive speech quality assessment by deep neural network. In *Proceedings of the IEEE International Conference on Digital Signal Processing (DSP'16)*. 99–103.
[18] Jan A. Bergstra and C. A. Middelburg. 2003. ITU-T recommendation G. 107: The E-Model, a computational model for use in transmission planning. International Telecommunication Union, Geneva, Switzerland.
[19] Demóstenes Z. Rodríguez, Dick Carrillo, Miguel A. Ramírez, Pedro H. J. Nardelli, and Sebastian Möller. 2021. Incorporating wireless communication parameters into the e-model algorithm. *IEEE/ACM Trans. Aud., Speech Lang. Process.* 29 (2021), 956–968.
[20] Demóstenes Zegarra Rodríguez and Sebastian Möller. 2019. Speech quality parametric model that considers wireless network characteristics. In *Proceedings of the 11th International Conference on Quality of Multimedia Experience (QoMEX'19)*. IEEE, 1–6.
[21] Emmanuel T. Affonso, Rodrigo D. Nunes, Renata L. Rosa, Gabriel F. Pivaro, and Demostenes Z. Rodriguez. 2018. Speech quality assessment in wireless VoIP communication using deep belief network. *IEEE Access* 6 (2018), 77022–77032.
[22] Demóstenes Z. Rodríguez, Renata L. Rosa, Franciscone L. Almeida, Gabriel Mittag, and Sebastian Möller. 2019. Speech quality assessment in wireless communications with MIMO systems using a parametric model. *IEEE Access* 7 (2019), 35719–35730.
[23] Rodrigo Dantas Nunes, Renata Lopes Rosa, and Demóstenes Zegarra Rodríguez. 2019. Performance improvement of a non-intrusive voice quality metric in lossy networks. *IET Commun.* 13, 20 (2019), 3401–3408.
[24] Kjell Brunnström, Sergio Ariel Beker, Katrien De Moor, Ann Dooms, Sebastien Egger, Marie Neige Garcia, Tobias Hossfeld, Satu Jumisko Pyykkö, Christian Keimel, Mohamed Chaker Larabi, et al. 2013. Qualinet white paper on definitions of quality of experience. HAL-00977812.
[25] H. Z. Jahromi, A. Hines, and D. T. Delanev. 2018. Towards application-aware networking: ML-based end-to-end application KPI/QoE metrics characterization in SDN. In *Proceedings of the 10th International Conference on Ubiquitous and Future Networks (ICUFN'18)*. 126–131.
[26] ITU-T Rec. Coded-Speech Database Series P, Supplement 23. International Telecommunication Union, Geneva, 1998.
[27] Rajesh Kumar Dubey and Arun Kumar. 2013. Non-intrusive speech quality assessment using several combinations of auditory features. *Int. J. Speech Technol.* 16, 1 (2013), 89–101.
[28] Yi Hu and Philipos C. Loizou. 2006. Subjective comparison of speech enhancement algorithms. In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing*, Vol. 1. 153–156.
[29] Hans Günter Hirsch and David Pearce. 2000. The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In *Proceedings of the Automatic Speech Recognition: Challenges for the New Millenium (ASR'00), ISCA Tutorial and Research Workshop (ITRW'00)*.
[30] Rahul Jaiswal and Andrew Hines. 2020. Towards a non-intrusive context-aware speech quality model. In *Proceedings of the 31st Irish Signals and Systems Conference (ISSC'20)*. 1–5.

[31] Yi Hu and Philipos C. Loizou. 2007. Evaluation of objective quality measures for speech enhancement. *IEEE Trans. Aud. Speech Lang. Process.* 16, 1 (2007), 229–238.
[32] Yi Hu and Philipos C. Loizou. 2004. Speech enhancement based on wavelet thresholding the multitaper spectrum. *IEEE Trans. Speech Aud. Process.* 12, 1 (2004), 59–67.
[33] Pascal Scalart et al. 1996. Speech enhancement based on a priori signal to noise estimation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 2. 629–632.
[34] Sunil Kamath and Philipos Loizou. 2002. A multi-band spectral subtraction method for enhancing speech corrupted by colored noise. In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing*, Vol. 4. 4160–4164.
[35] Harald Gustafsson, Sven E. Nordholm, and Ingvar Claesson. 2001. Spectral subtraction using reduced delay convolution and adaptive averaging. *IEEE Trans. Speech Aud. Process.* 9, 8 (2001), 799–807.
[36] Yariv Ephraim. 1992. Statistical-model-based speech enhancement systems. *Proc. IEEE* 80, 10 (1992), 1526–1555.
[37] Yariv Ephraim and David Malah. 1984. Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. *IEEE Trans. Acoust. Speech Sign. Process.* 32, 6 (1984), 1109–1121.
[38] Israel Cohen. 2002. Optimal speech enhancement under signal presence uncertainty using log-spectral amplitude estimator. *IEEE Sign. Process. Lett.* 9, 4 (2002), 113–116.
[39] Yi Hu and Philipos C Loizou. 2003. A generalized subspace approach for enhancing speech corrupted by colored noise. *IEEE Trans. Speech Aud. Process.* 11 (2003), 334–341.
[40] Udar Mittal and Nam Phamdo. 2000. Signal/noise KLT based approach for enhancing speech degraded by colored noise. *IEEE Trans. Speech Aud. Process.* 8, 2 (2000), 159–167.
[41] Ludovic Malfait, Jens Berger, and Martin Kastner. 2006. P.563-The ITU-T standard for single-ended speech quality assessment. *IEEE Trans. Aud. Speech Lang. Process.* 14, 6 (2006), 1924–1934.
[42] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 785–794.
[43] Jerome H. Friedman. 2001. Greedy function approximation: A gradient boosting machine. *Ann. Statist.* (2001), 1189–1232.
[44] Georgios N. Dimitrakopoulos, Aristidis G. Vrahatis, Vassilis Plagianakos, and Kyriakos Sgarbas. 2018. Pathway analysis using XGBoost classification in Biomedical data. In *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*. 1–6.
[45] J. Ross Quinlan. 1993. *C4.5: Programs for Machine Learning*. Elsevier.
[46] Leo Breiman, Jerome Friedman, Charles J. Stone, and Richard A. Olshen. 1984. *Classification and Regression Trees*. CRC Press.
[47] Esrafil Jedari, Zheng Wu, Rashid Rashidzadeh, and Mehrdad Saif. 2015. Wi-Fi based indoor location positioning employing random forest classifier. In *Proceedings of the IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN'15)*. 1–5.
[48] Qingyong Wang, Yun Zhou, Weiping Ding, Zhiguo Zhang, Khan Muhammad, and Zehong Cao. 2020. Random forest with self-paced bootstrap learning in lung cancer prognosis. *ACM Trans. Multimedia Comput. Commun. Appl.* 16, 1s (2020), 1–12.
[49] Leo Breiman. 2001. Random forests. *Mach. Learn.* 45, 1 (2001), 5–32.
[50] Xiaomei Liang, Xuerong Gou, and Yong Liu. 2012. Fingerprint-based location positoning using improved KNN. In *Proceedings of the 3rd IEEE International Conference on Network Infrastructure and Digital Content*. 57–61.
[51] Vladimir Vapnik. 2000. *The Nature of Statistical Learning Theory*. Springer.
[52] Ethem Alpaydin. 2020. *Introduction to Machine Learning, 4th Edition*. MIT Press.
[53] Rahul Jaiswal. Performance analysis of voice activity detector in presence of non-stationary noise. In *Proceedings of the11th International Conference on Robotics, Vision, Signal Processing and Power Applications*. Springer, 59–65.
[54] Leonidas Deligiannidis and Hamid R. Arabnia. 2014. *Emerging Trends in Image Processing, Computer Vision and Pattern Recognition*. Morgan Kaufmann.
[55] Mark Eisen, Clark Zhang, Luiz F. O. Chamon, Daniel D. Lee, and Alejandro Ribeiro. 2018. Online deep learning in wireless communication systems. In *Proceedings of the 52nd Asilomar Conference on Signals, Systems, and Computers (ACSSC'18)*. IEEE, 1289–1293.
[56] Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc.: Ser. B (Methodol.)* 58, 1 (1996), 267–288.
[57] Yun Li, Tao Li, and Huan Liu. 2017. Recent advances in feature selection and its applications. *Knowl. Inf. Syst.* 53, 3 (2017), 551–577.
[58] R. K. Jain, T. Damoulas, and C. E. Kontokosta. 2014. Towards data-driven energy consumption forecasting of multi-family residential buildings: Feature selection via the lasso. In *Computing in Civil and Building Engineering*. 1675–1682.

[59] Hongyi Zhang, Haoke Zhang, Sandeep Pirbhulal, Wanqing Wu, and Victor Hugo C. De Albuquerque. 2020. Active balancing mechanism for imbalanced medical data in deep learning-based classification models. *ACM Trans. Multimedia Comput. Commun. Appl.* 16, 1s (2020), 1–15.

[60] Chris Drummond and Robert C. Holte. 2003. C 4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In *Proceedings of the 20th International Conference on Machine Learning (ICML'03) Workshop on Learning from Imbalanced Data Sets*.

[61] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C. Prati, Bartosz Krawczyk, and Francisco Herrera. 2018. *Learning from Imbalanced Data Sets*. Vol. 11. Springer.

[62] Sara Belarouci and Mohammed Amine Chikh. 2017. Medical imbalanced data classification. *Adv. Sci. Technol. Eng. Syst. J.* 2, 3 (2017), 116–124.

[63] Hao Ye, Geoffrey Ye Li, Biing-Hwang Fred Juang, and Kathiravetpillai Sivanesan. 2018. Channel agnostic end-to-end learning based communication systems with conditional GAN. In *Proceedings of the IEEE Globecom Workshops*. 1–5.

[64] Masoumeh Zareapoor and Jie Yang. 2021. Equivariant adversarial network for image-to-image translation. *ACM Trans. Multimedia Comput. Commun. Appl.* 17, 2s (2021), 1–14.

[65] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative nets. In *Advances in Neural Information Processing Systems*. 2672–2680.

[66] Rajesh Kumar Dubey and Arun Kumar. 2015. Comparison of subjective and objective speech quality assessment for different degradation/noise conditions. In *Proceedings of the International Conference on Signal Processing and Communication*. IEEE, 261–266.

[67] Haoran Sun, Xiangyi Chen, Qingjiang Shi, Mingyi Hong, Xiao Fu, and Nikos D. Sidiropoulos. 2017. Learning to optimize: Training deep neural networks for wireless resource management. In *Proceedings of the 18th IEEE International Workshop on Signal Processing Advances in Wireless Communications*. 1–6.

[68] Hao Ye, Geoffrey Ye Li, and Biing-Hwang Juang. 2017. Power of deep learning for channel estimation and signal detection in OFDM systems. *IEEE Wireless Commun. Lett.* 7, 1 (2017), 114–117.

[69] Timothy Dozat. 2016. Incorporating Nesterov momentum into Adam. In *Proceedings of the 4th International Conference on Learning Representations (ICLR'16)*.

[70] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15, 1 (2014), 1929–1958.

[71] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. 2012. Neural networks for machine learning; lecture 6a overview of mini-batch gradient descent.

[72] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR'15)*.

Received 23 June 2021; revised 20 January 2022; accepted 28 March 2022