Intelligent Speech Features Mining for Robust Synthesis System Evaluation

Moses E. Ekpenyong(✉), Udoinyang G. Inyang, and Victor E. Ekong

Department of Computer Science, University of Uyo, P.M.B. 1017, Uyo 520003, Nigeria
{mosesekpenyong,udoinyanginyang,victoreekong}@uniuyo.edu.ng,
mosesekpenyong@gmail.com

Abstract. Speech synthesis evaluation involves the analytical description of useful features, sufficient to assess the performance of a speech synthesis system. Its primary focus is to determine the degree of semblance of synthetic voice to a natural or human voice. The task of evaluation is usually driven by two methods: the ***subjective*** and ***objective*** methods, which have indeed become a regular standard for evaluating voice quality, but are mostly challenged by high speech variability as well as human discernment errors. Machine learning (ML) techniques have proven to be successful in the determination and enhancement of speech quality. Hence, this contribution utilizes both supervised and unsupervised ML tools to recognize and classify speech quality classes. Data were collected from a listening test (experiment) and the speech quality assessed by domain experts for naturalness, intelligibility, comprehensibility, as well as, tone, vowel and consonant correctness. During the pre-processing stage, a Principal Component Analysis (PCA) identified 4 principal components (intelligibility, naturalness, comprehensibility and tone) – accounting for 76.79% variability in the dataset. An unsupervised visualization using self organizing map (SOM), then discovered five distinct target clusters with high densities of instances, and showed modest correlation between significant input factors. A Pattern recognition using deep neural network (DNN), produced a confusion matrix with an overall performance accuracy of 93.1%, thus signifying an excellent classification system.

Keywords: Deep neural network · Dimension reduction · Machine learning
Pattern recognition · Speech quality evaluation

1 Introduction

Speech synthesis has remained a challenging task due to high variability in speech signals. This problem stems from the fact that speakers may have dissimilar accents, dialects, or pronunciations, and converse in different styles – at different rates, and in variable emotional states. Also, the presence of environmental noise, reverberation, microphone type and recording devices could introduce further variability. Synthesized speech can be evaluated using different methods and at several levels. All methods give some information on speech quality, but the correctness of the evaluated data can be brought to question. Perhaps the most suitable way to test a speech synthesizer is to select several methods to assess each feature separately. For instance using segmental,

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 3–18, 2018.
https://doi.org/10.1007/978-3-319-93782-3_1

sentence level, prosody, and overall tests together provides lots of useful information, but this on the other hand is labour intensive and time-consuming.

Perceptual Evaluation of Speech Quality (PESQ)[1] represents a family of standards comprising a test methodology for automated assessment of speech quality – as experienced by a user of a telephony system. It is an objective method developed to model subjective tests commonly used in telecommunications (e.g. ITU-T P.800) for the assessment of voice quality, perceived by humans. Although PESQ utilizes true voice samples as test signals, and has been proved to be the most reliable measure for assessing speech quality, it is computationally demanding and requires access to the whole utterance. In some applications, this might not be acceptable. Ideally, the objective measure should predict the quality of speech independent of the type of distortions introduced by the system – be it a network, a speech coder or a speech enhancement algorithm. Hence, many systems are now optimized for speech and can respond in an unpredictable way to signal degradation.

1.1 Speech Synthesis Framework

The Hidden Markov Model (HMM) speech synthesis [1, 2] has provided a flexible framework for synthesizing speech. Although HMM has tremendously improved flexibility by enabling: (i) varying speaking styles and modified speaker/voice characteristics; and (ii) small memory footprint and robustness; it however suffers from speech quality degradation [3], compared to the unit selection approach [4, 5]. Factors responsible for the degraded quality include: over-simplified vocoder techniques; acoustic modelling inaccuracy; and over-smoothing of the generated speech parameters [2].

Recently, Deep Neural Networks (DNNs) have achieved great improvements in both quality and accuracy in speech synthesis [3], as their trainings converge faster and outperform other approaches such as Gaussian Mixture Models (GMMs) and HMMs – if their initial parameter values are pre-trained instead of randomly initialized [6]. The pre-training methods use unsupervised techniques [7]. DNN simulates human speech production as a layered hierarchical structure that transforms linguistic texts into speech utterances. The Neural Network (NN) is trained to map the input phonetic transcriptions of the training text into sequences of acoustic feature vectors, sufficient to yield predefined speech waveforms when processed by a signal generation module. The training data may correspond to written transcription of speech carried in a predefined speech waveform. Before training a DNN, a set of utterances is transcribed phonetically with sequences of phonetic-context descriptors – each containing a set of phonetic speech units – indicating contexts of the respective speech units and their durations. The trained NN may then map the sequence of phonetic-context descriptors to predicted feature vectors, which are finally transformed into synthesized speech by the signal generation module. DNN-based speech synthesizers are therefore (more) likely to overcome the limitations of HMMs, given its high accuracy level and intelligent prediction models. Deep learning is at the intersection of neural networks, graphical modelling, optimization, pattern recognition, and signal processing. Deep learning methods have grown

[1] http://www.itu.int/rec/T-REC-P.862/en.

increasingly richer, encompassing those of neural networks, hierarchical probabilistic models, and a variety of unsupervised and supervised feature learning algorithms [8]. The three main reasons for the wide applications of deep learning in recent times are the drastically increased chip processing abilities, the reduced cost of computing hardware, and the recent advances in machine learning and signal/information processing research. These advances have enabled the deep learning methods to effectively exploit complex, composite non-linear functions, to learn distributed and hierarchical feature representations, and to make effective use of both labelled and unlabeled data. Three major categories of deep learning networks include [8]:

(i) Unsupervised or generative deep learning networks, which captures high-order correlation of the observed or visible data for pattern analysis, or synthesis purposes when no information about target class labels is available. Subclasses of generative or unsupervised deep networks includes, the energy-based deep models;
(ii) Deep networks for supervised learning, which are intended to directly provide discriminative power for pattern classification purposes, often by characterizing the posterior distributions of classes conditioned on the visible data. Target label data are always available in direct or indirect forms for such supervised learning. They are also called discriminative deep networks;
(iii) Hybrid deep networks, where the goal is discrimination which is assisted, often in a significant way, with the outcomes of generative or unsupervised deep networks. The goal can also be accomplished when discriminative criteria for supervised learning are used to estimate the parameters in any of the deep generative or unsupervised deep networks in category 1 above.

In [9], DNN was used as an intelligent tool for robust evaluation of speech synthesis systems. Data for the training were measures obtained from listening experiments and an objective evaluation of the various listening tests (to establish the speech quality range) by experts. The evaluation measures or criteria considered *naturalness*, *intelligibility*, *comprehensibility*, *tone correctness*, *vowel correctness* and *consonant correctness*. They observed that a 10-layer DNN gave better performance compared to 3- and 5-layer DNNs.

In this paper the work done in Ekpenyong, Inyang and Ekong [9] is extended, by studying the mappings and patterns presented by expert knowledge for efficient knowledge synthesis and discovery. To achieve this, we propose a framework that integrates a self organizing map (SOM) – an unsupervised technique useful for visualizing patterns inherent in the evaluation measures obtained in the listening test experiment [9]; and a DNN-Pattern recognizer – a supervised machine learning approach – for efficient classification of the simulated target classes. Early speech synthesizers were evaluated primarily for intelligibility and naturalness. But in [9], comprehensibility, tone correctness, vowel correctness, and consonant correctness were added, to examine their relevance in the overall speech quality prediction.

This paper proceeds as follows: Sect. 2 discusses speech synthesis evaluation and the approaches involved; Sect. 3 presents the proposed methodology; Sect. 4 discusses the results obtained; and, Sect. 5 concludes the paper with a pointer to future work.

2 Speech Synthesis Evaluation

Speech synthesis evaluation involves the analytic description of system performance in terms of defined factors, and focuses on whether or not the voice is accepted as quality – by attempting to discover how closely it is to the human voice [10–12]. Synthesized speech may be evaluated with regards to the following measures: intelligibility, naturalness, and appropriateness for used application [13, 14], but most applications discriminate these measures based on the target users. For instance, in reading systems for the blind, intelligibility with high speech rate is often the most preferred measure than naturalness. Conversely, prosodic features and naturalness are essential for multimedia applications or electronic mail readers. The evaluation procedure is usually done by subjective listening tests with response to a set of syllables, words, sentences, or with other questions. The test material is mainly focused on consonants, because they are more problematic to synthesize than vowels. Precise and reliable assessment of speech quality is becoming vital, to meet with users' satisfaction of the deployed speech processing system. Individual users have varying internal standards of what constitutes "excellent" or "poor" speech quality, which results in wide variability in rating scores among listeners. Several methods for evaluating synthetic speech have evolved over the last decades. The most commonly used methods for evaluating intelligibility include: *segmental evaluation methods* (where the diagnostic and modified rhyme tests are the most famous – [15, 16]; *sentence level tests* (where the Haskins sentences and semantically unpredictable sentences (SUS) are the notable ones [17]; and c*omprehension tests* [18, 19]. To evaluate the overall speech quality, the following methods are often preferred: *mean opinion score (MOS)* Goldstein [15]; categorical estimation (CE) [20]; *pairwise comparison (PC)* [20]; and, *magnitude and ratio estimation* [21]. Two approaches are outstanding in the evaluation of synthetic voices: the subjective and objective approaches. These approaches are discussed in the following sub sections.

2.1 Subjective Approach

The most reliable method of speech synthesis evaluation relies on measurement of the perceptual performance of human listeners (often referred to as ***subjective*** test of speech quality), yet there are strong demands for ***objective*** evaluation and is widely driven by the desire to determine the quality of speech technology systems, without requiring significant number of human listeners [22]. Although subjective techniques to evaluating synthetic speech make use of human experts, the expertise required may vary from technique to technique – ranging from untrained native speakers of the language to trained phoneticians. However, subjective votes are heavily influenced by numerous factors such as the preferences of individual subjects and the experiment context. Robust procedures using untrained listeners would of course be extremely useful since besides the extra cost of sourcing for trained phoneticians, there is also the danger of experts being influenced by theoretical biases, particularly in such controversial domain such as prosody

2.2 Objective Approach

Measuring whether or not an utterance from a synthetic voice acoustically matches the same utterance in human speech is beneficial for a number of reasons. First, by their very nature, objective measures offer a clear measurement of how a voice is performing and can be used to diagnose problem areas requiring development. Also, compared to the time and cost involved in user testing, objective measures can be an efficient form of evaluation. However, findings from such evaluations may not always match up with listener perceptions. Clark and Dusterhoff [23] have shown that some objective measures may be oversensitive compared to the human ear. The opposite may also be true, in that synthetic samples may be perfect in acoustic terms, but may be perceived as unnatural by listeners. Morton [12] had explained this based on the processing required to listen to speech. He suggested that human speakers naturally vary their speech, with the aim of being understood, and that this affects the way human speech is processed by listeners. Since a synthesizer does not have this capability, the processing needs of the listener are not accounted for. And this may cause the listener to find the synthetic speech unnatural or difficult to understand. Objective/acoustic measures can be used in evaluation of different aspects of a voice, such as prosodic features or intonation. Examples of acoustic measures which may be of interest include fundamental frequency (F0), segmental duration and intensity. A common approach is to use statistical methods (Root Mean Squared Error: RMSE) to model the expected performance of the voice against actual performance, measuring the accuracy of the synthetic voice against a natural voice. Whilst objective measures have been described as more efficient than user testing, of course specialist knowledge is required in order to run these tests. Objective measures can be a useful means of evaluation for synthetic voices, and can be a very efficient form of testing. However, such techniques require specialist knowledge.

3 System Design

3.1 Data Collection and Representation

The data collection and initial pre-processing methodology is conceptualized in Fig. 1. After a HMM-based synthesis experiment, the data collection spanned two stages. First, a listening test was performed on a set of synthesized voices. Second, domain experts' assessments were used for validating listeners' responses. Speech quality (SQ) measures considered in this experiment were *naturalness, intelligibility, comprehensibility, tone correctness, vowel correctness and consonant correctness.* The mean opinion score (MOS) test was used to determine the naturalness of the synthesized voices, while the Modified Rhyme Test (MRT) was the method for evaluating intelligibility of the voices, both of which were in the range (1–5). In order to extract results for tone, vowel and consonant correctness, the MRT results were extrapolated through a comparison of the listeners' responses with a list of correct utterances. The resultant values are elements of the set {0, 1}, corresponding to true (correct) or false (wrong). Semantically Unpredictable Sentences (SUSs) – i.e., sentences that are syntactically correct, but semantically anomalous, were used

to rank the comprehensibility of the synthetic voices in the range (1–4). To achieve this, listeners were made to comprehend a set of SUSs – by listening to the sentences (at least three times), before typing what they've heard. Although SUSs seals the listeners from contextual cues that contribute to perceived intelligibility, it however maintains a reliable evaluation that keeps test results undistorted.

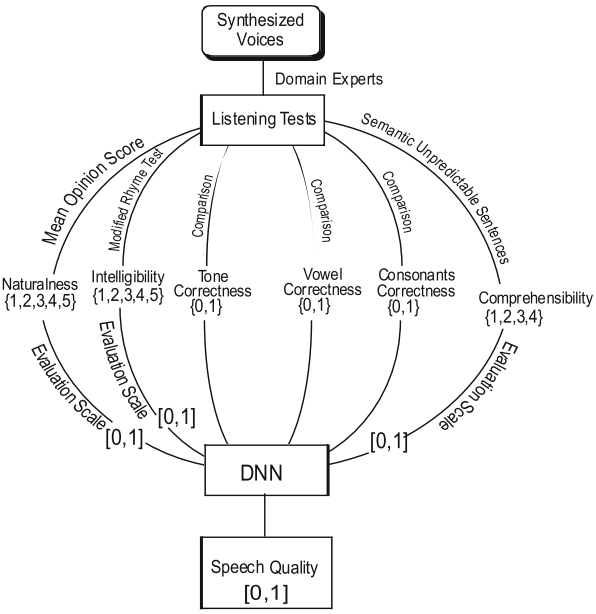

Fig. 1. Data collection and pre-processing work-flow

The coding of the dataset attributes for the experiment is as summarized in Table 1.

Table 1. Input parameter description and coding

S/No.	Input parameter	Modality	Code
1.	Naturalness	Mean opinion score	5-Excellent, 4-V. good, 3-Good, 2-Fair, 1-Poor
2.	Intelligibility	Modified rhyme test	5-Excellent, 4-V. good, 3-Good, 2-Fair, 1-Poor
3.	Tone correctness	Comparison	1-Correct, 2-Wrong
4.	Vowel correctness	Comparison	1-Correct, 2-Wrong
5.	Consonant correctness	Comparison	1-Correct, 2-Wrong
6.	Comprehensibility	SUS	4-High, 3-Medium, 2-Low, 1-No comprehension
7.	Voice quality	DNN	1-Very high, 2-High, 3-Medium, 4-Low, 5-V.low

The results obtained for naturalness, intelligibility and comprehensibility were further transformed to yield results in the range (0–1). Our dataset comprised 563 data points, out of which, 393 (70%) records was randomly selected for training the DNN-Pattern recognizer; while 15% (85) each was selected as test and validation datasets. A section of the sample dataset is shown in Table 2. The target classes were obtained through expert evaluation, and include: 1-Poor, 2-Fair, 3-Good, 4-V.Good, 5-Excellent

Table 2. Dataset for speech synthesis evaluation

S/no.	Input class							Target class				
	Nat	Intell	Comp	T	V	C	SQ	C1	C2	C3	C4	C5
1.	0.8757	0.8838	0.7505	1	1	1	5	0	0	0	0	1
2.	0.3588	0.6361	0.4129	1	1	1	4	0	0	0	1	0
3.	0.6485	0.8068	0.8187	0	1	1	5	0	0	0	0	1
4.	0.5144	0.5778	0.6667	1	1	1	3	0	0	0	0	1
5.	0.6137	0.7873	0.7713	0	1	1	5	0	0	0	0	1
6.	0.7447	0.6529	0.8115	1	1	1	4	0	0	0	0	1
7.	0.3728	0.5286	0.7491	0	1	1	3	0	0	0	1	0
8.	0.7218	0.7492	0.6195	1	0	1	4	0	0	0	0	1
9.	0.4642	0.5316	0.8104	1	1	1	3	0	0	0	0	1
10.	0.6730	0.7231	0.8259	1	1	0	4	0	0	0	0	1
11.	0.5858	0.5360	0.4565	1	1	1	3	0	0	0	0	1
12.	0.2476	0.5487	0.0292	1	1	1	3	0	0	0	1	0
13.	0.5759	0.5639	0.4016	0	1	1	3	0	0	0	1	0
14.	0.6020	0.5421	0.4277	0	1	1	3	0	0	0	1	0
15.	0.5723	0.4715	0.7010	1	1	1	3	0	0	0	0	1
16.	0.5452	0.7074	0.5017	1	1	1	4	0	0	0	0	1
17.	0.4354	0.7020	0.4864	1	1	1	4	0	0	0	1	0
18.	0.5682	0.5563	0.3072	1	1	1	3	0	0	0	0	1
19.	0.5795	0.3081	0.0457	0	1	1	2	0	0	1	0	0
20.	0.5151	0.6520	0.4131	1	1	0	4	0	0	0	1	0

3.2 SOM Methodology

SOM is used to divide the input data points into one of several groups. Training data are provided to the SOM. During training phase, the SOM groups the input data into clusters, where data with the most similar properties are clustered together. The training process in SOM is unsupervised and draws its strengths from the pioneering works of Kohonen [24]. SOM forms clusters by itself on the basis of training data, and then places any future data into similar clusters. As a data item or pattern is input to the processing (output) layer, the output neuron is considered the winner when it contains the weights most similar to the input. The similarity is then computed by comparing typically the Euclidean distance between the set of weights from each neuron. The shortest or least distance wins. A linear activation function only is employed with no bias. As usual, training of SOM starts by using random values for weights as is the case for any network,

provided that no valid "guess" (estimate) for starting values is unknown, which could speed up training. Unlike other neural networks types, the training of SOM normally involves a fixed number of iterations.

3.3 DNN-Pattern Recognition Model

A DNN is a neural network or multilayer perceptron with two or more hidden layers – whose weights are fully connected and initialized using either a supervised or unsupervised pre-training approach [8]. DNN is able to represent high dimensional and correlated features efficiently and can compactly model highly complex mapping functions, which contributes to improving generalization – as weights are trained from all the training data. The training can however be optimized by back-propagating derivatives of the mean square error (MSE) cost functions that measure the discrepancy between the target and actual outputs. Zen, Senior and Schuster [3] however argued that this approach may require large computations than building decision trees, since at prediction stage DNNs require matrix multiplication at each layer, compared to decision trees which only require traversing trees from the root to terminal nodes using a subset of their input features. Hence, the restricted Boltzman machine (RBM) pre-training has been recently explored to minimize this bottleneck to achieve a fast initial reduction in training error. Optimizing DNN training can be achieved using the following steps [7, 25]:

(i) pre-training each layer, exclusively, using a greedy algorithm
(ii) applying unsupervised learning at every layer in a way that preserves information from the input and disentangling any factor of variation
(iii) fine-tuning the entire network, subject to the ultimate criterion of interest

Figure 2 shows an architecture of the proposed DNN, with L layers (L > 3). Each layer has connection weights, a bias vector ($\mathbf{b_L}$), and an output vector ($\mathbf{O_L}$). The number of neurons in each layer is denoted by m, where, $m^1, m^2, \ldots, m^L$, are the number of neurons in Hidden Layers 1, 2, …, L, respectively. The input to the system, x_i are the various criteria for evaluation. In this paper, we have identified the following factors:

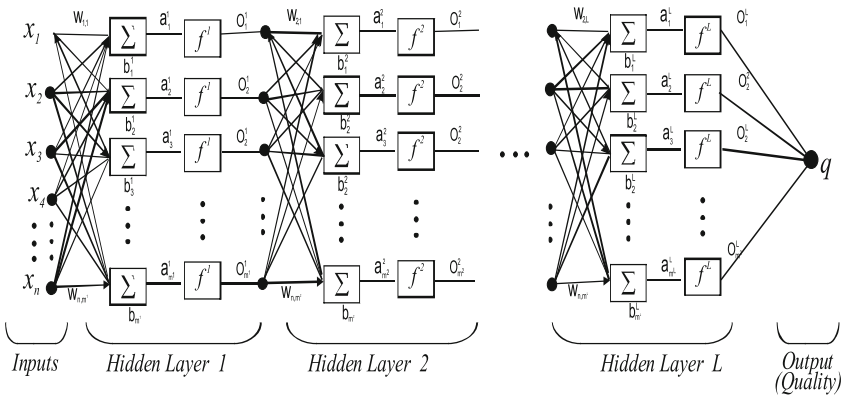

Fig. 2. System architecture of the proposed DNN

naturalness, intelligibility, comprehensibility, tone correctness, vowel correctness and consonant correctness, for use in the prototype evaluation. In each of the layers, the input vector elements enter the DNN through weights, (w_{i,m_n^L}), which represents the weight of the link between the *ith* input neuron and the nth neuron of the *Lth* hidden layer.

The matrix of weights (W) for each layer is obtained in Eq. (1):

$$W = \begin{bmatrix} w_{1,m_1^L} & w_{1,m_i^L} & w_{1,m_3^L} & \cdots & w_{1,m_n^L} \\ w_{2,m_1^L} & w_{2,m_2^L} & w_{2,m_3^L} & \cdots & w_{2,m_n^L} \\ w_{3,m_1^L} & w_{3,m_2^L} & w_{3,m_3^L} & \cdots & w_{3,m_n^L} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ w_{n,m_1^L} & w_{n,m_2^L} & w_{n,m_3^L} & \cdots & w_{n,m_n^L} \end{bmatrix} \tag{1}$$

The inputs to each layer of the DNN are denoted as, a, with a^1, a^2,…, a^L, representing inputs to layers 1, 2, …, L, respectively. Therefore the inputs to the DNN are as follows:

$$\begin{aligned} a^1 &= \sum_{i=1}^{n} f^1(w_{1,m^1} x_i + b_{m^1}^1) \\ a^2 &= \sum_{i=1}^{n} f^2(w_{2,m^2} a^1 + b_{m^2}^2) \\ a^L &= \sum_{i=1}^{n} f^L(w_{L,m^L} a^{L-1} + b_{m^L}^L) \end{aligned} \tag{2}$$

where, i = 1, 2…, n are the number of input variables (length of the input vector), *m*, is the number of neurons in *Lth* layer (L > 3), while f^L, is the transfer function of the *Lth* Layer. The output of a proceeding layer is the input of the immediate succeeding layer.

Next, a resilient back-propagation pattern recognition algorithm (RPROP), with sigmoid hidden and softmax output neurons (*patternnet*), was used to classify the input vectors. RPROP stands for *resilient propagation* and is an efficient learning scheme that performs a direct adaptation of the weight step based on local gradient information. One of the main advantages of PROP lies in the fact that for many problems no choice of parameters is needed at all to obtain optimal or at least nearly optimal convergence times. The pseudo-code in Fig. 3 shows the kernels of the RPROP adaptation and learning process. The min(max) operator is exposed to deliver the min(max) of two numbers, and the sign operator returns +1, if argument is positive, −1, if negative, and 0, otherwise.

The steps used to implement the DNN-Pattern recognizer include: (i) Initialize the hidden layer size; (ii) Create the pattern recognition network; (iii) set up the division of data for training, validation, and testing; (iv) train the network; (v) test the network; (vi) view the network and display performance plots.

4 Results

4.1 Feature Dimension Reduction and Loading

It is important to assess the contributions of each of the extracted features, to speech quality. The assessment is to enhance the prediction and evaluation of the speech quality. One major objective of data mining is classification learning, and this is achieved by representing relevant data with the smallest number of feature dimensions – such that its characteristics are not lost while reducing the processing complexities in the data [27]. In addition, the accuracy and reliability of a classification or evaluation results will degrade if highly uncorrelated features of interest are used. The process is also necessary to prepare the dataset for supervised learning. The Principal Component Analysis (PCA) is one of the notable and widely used techniques for dimension reduction. In this paper, feature dimensionality assessment was performed to obtain a set of degrees of freedom with which a large proportion of the variability of a speech quality could be explained. Specifically, the speech corpus datasets of size 6×563 was mapped to the given k-principal component framework and transformed into dataset of size $563 \times k$, where k is the number of extracted features with eigenvalues of at least one (1).

The PCA analysis was implemented in Matlab 2015a, and the result is presented in Table 3. Results reveal that naturalness is the most important principal component accounting for 23.92% of the variability of the target feature; followed by intelligibility (20.33%); comprehensibility (19.03%); and tone (13.51%). The first 4 Principal Components are effective for classification since they account for 76.79% of the total variation associated with all 6 original features. This suggests that most of the variability in the

$$
\begin{aligned}
&for\ all\ weight\ and\ biases\ \{\\
&\quad if\left(\frac{\partial E}{\partial w_{ij}}(t-1) * \frac{\partial E}{\partial w_{ij}}(t) > 0\right) then\ \{\\
&\qquad \Delta_{ij}(t) = \min(\Delta_{ij}(t-1) * \eta+, \Delta_{max})\\
&\qquad \Delta w_{ij}(t) = -sign\left(\frac{\partial E}{\partial w_{ij}}(t)\right) * \Delta_{ij}(t)\\
&\qquad w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)\ \}\\
&\qquad else\ if\left(\frac{\partial E}{\partial w_{ij}}(t-1) * \frac{\partial E}{\partial w_{ij}}(t) < 0\right) then\ \{\\
&\qquad\quad \Delta_{ij}(t) = \max(\Delta_{ij}(t-1) * \eta-,\ \Delta_{min})\\
&\qquad\quad w_{ij}(t+1) = w_{ij}(t) - \Delta w_{ij}(t-1)\\
&\qquad\quad \frac{\partial E}{\partial w_{ij}}(t) = 0\ \}\\
&\qquad else\ if\left(\frac{\partial E}{\partial w_{ij}}(t-1) * \frac{\partial E}{\partial w_{ij}}(t) = 0\right) then\ \{\\
&\qquad\quad \Delta w_{ij}(t) = -sign\left(\frac{\partial E}{\partial w_{ij}}(t)\right) * \Delta_{ij}(t)\\
&\qquad\quad w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)\ \}\}
\end{aligned}
$$

Fig. 3. The RPROP algorithm Source [26]

dataset can be obtained with 66.67% of the original dimension in the dataset. The other SQ features – vowel and consonants, together account for 23.21% of the variance, each with eigen value less than 1. In term of communality loadings, the principal components show higher loadings than the non-principal components, and imply that they are suitable for classification of the target feature. Correlation matrix of the principal feature components are given in Table 4. The positive correlations between these features confirm the effectiveness of classification with these features.

Table 3. Principal components and variability proportion

Feature	Eigen value	Proportion (%)	Cumulative (%)	Communality estimates	
				Prior	Final
Naturalness	1.315403	23.92	23.92	0.06384	0.15768
Intelligibility	1.099593	20.33	44.25	0.04210	0.11628
Comprehensibility	1.021843	19.03	63.28	0.04253	0.11656
Tone	1.000558	13.51	76.79	0.01252	0.05564
Consonant	0.859490	12.32	89.11	0.00855	0.04477
Vowel	0.773114	10.89	100.00	0.00197	0.01982

Table 4. Correlation matrix of the principal feature components

	Naturalness	Comprehensibility	Intelligibility	Tone
Naturalness	**1.0000**	0.1955	0.2334	0.0400
Comprehensibility	0.1955	**1.0000**	0.1006	0.0022
Intelligibility	0.2334	0.1006	**1.0000**	0.0529
Tone	0.0400	0.0022	0.0529	**1.0000**

4.2 SOM Visualization

Our SOM implemented the batch learning. The learning rate was fixed to 1, and the neighbourhood function was 1, for all neighbourhood nodes, and 0, elsewhere. This implies that after each epoch, a node's position is averaged over the data points for which the node or one of its neighbours was the winner (the nearest node). The size of the neighbourhood decreases linearly from the specified initial value to 1, in a specified number of epochs (called the ordering phase). Afterwards the neighbourhood size was fixed to 1, for the remaining training epochs (called the tuning phase).

A calibration of the cluster map is shown in Fig. 4. The calibration algorithm determines the degree of convergence of the input classes required to setting a basis for the neuron probabilities for the discovery of the target clusters, and the average variance of clustering provides a measure of the distance between the SOM neuron cluster centers. The calibration considers that each data point is valid with some probability, and could belong to any of the clusters of the SOM. The calibration step connects the clustering and classification steps in a highly logical manner and the procedure is performed for all the neurons regardless of the size and topology of the network. As seen in Fig. 4, the

resultant calibration confirms the presence of five pattern classes, in the prediction of the target features. The calibration shows the probability density of feature instances of the dataset in the discovered SQ clusters.

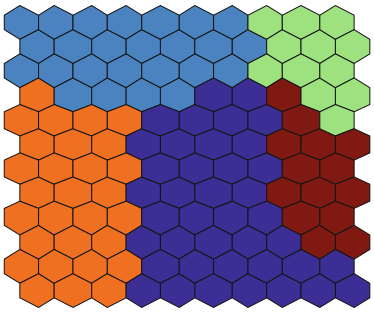

Fig. 4. Calibration plot of discovered target classes

In Fig. 5, the U-matrix: unified distance matrix (top leftmost map), is used to visualize the distances between the neurons. The distance between the adjacent neurons is computed and represented with different colourings between the adjacent nodes. Dark colours represent cluster separators while light colours represent the clusters. From the figure, presence of the clusters is noticed. The next four maps are the SOM component planes (of evaluation features) abstracted from PCA. We observe that naturalness and intelligibility followed same pattern, thus signifying modest correlation between them. In practice, intelligibility and naturalness are required to determine the speech quality, and one cannot preclude the other. Both require the evaluation of some segments of the utterance. While naturalness requires the evaluation of the entire utterance, intelligibility evaluates word segments for vowel, consonant or tone correctness. Also, comprehensibility and tone have similar patterns, but with a weaker correlation – most likely due to the small data (corpus) and uneven tone distribution. In practice, tone is a useful

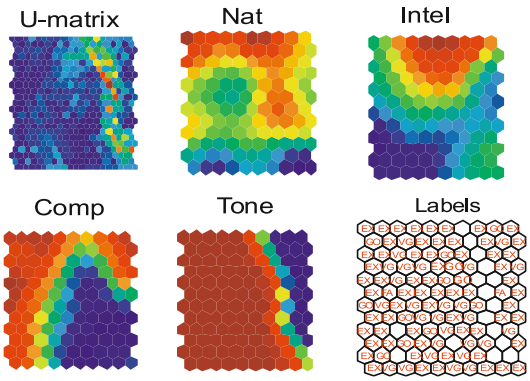

Fig. 5. U-matrix and feature component planes for selected features

determinant of speech comprehensibility. An extensive investigation into these claims with a larger corpus is expected in a future paper.

To ensure efficient knowledge synthesis and discovery, a confusion matrix which gives the classification of SQ patterns of the five distinct target categories (5-excellent, 4-very good, 3-good, 2-fair and 1-poor) is presented in Fig. 6. Samples belonging to "excellent" class were correctly classified. The classification accuracy of class "very good" was 90% (9) and 1 sample was incorrectly classified as a member of class "good'. Out of 78 (13.4% of the samples) instances of the target feature belonging to good SQ class, 67 samples were correctly classified while 4 and 9 samples were incorrectly classified as "very good" and "fair" SQ, respectively. The 'fair' and 'good' SQ dominated the speech samples used for this experiment and had the highest classification accuracy. While 161 (28.6%) samples of 'fair' class were correctly classified, 1.6% (9 samples) and 0.9% (5 samples) were incorrectly classified as members of "good" and "poor" classes respectively. Ten (10) samples of "poor" SQ were misclassified into "fair" quality with 286 (50.8%) samples correctly classified. Patterns of "excellent" SQ had 100% correct classification while samples belonging to 'poor" class, were only confused with "fair" class. "very good" class had the highest percentage of incorrectly classified patterns. The overall accuracy of classification is 93.1%, signifying an efficient speech quality classifier (Fig. 6).

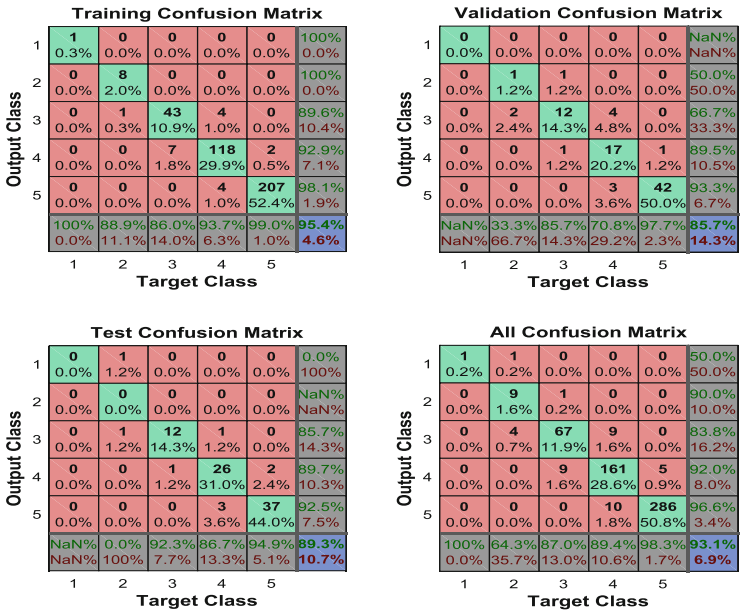

Fig. 6. Confusion matrix

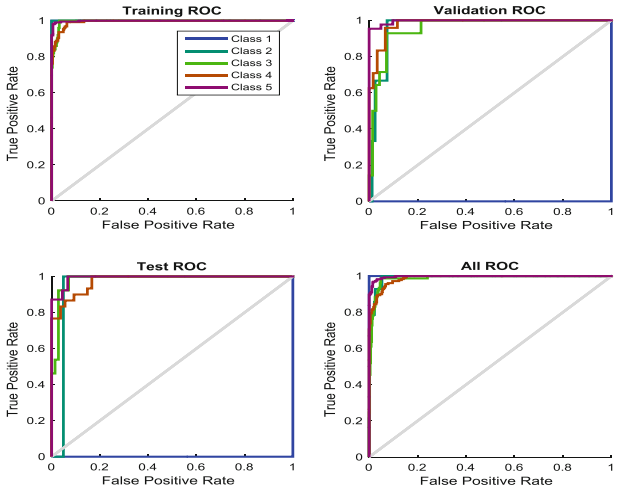

Fig. 7. ROC curve for training, validation and test data

5 Conclusion

Speech quality evaluation procedures for synthesized speech have played significant role in the development and enhancement cycle of synthesis systems. Although a host of factors can influence the speech quality, these factors depend (to a great extent) on the deployed application and can impair listening. Relevant feature identification has therefore become an essential task to apply data mining algorithms effectively in this context. This paper demonstrated the effectiveness of SOM and DNN in the evaluation of speech synthesis systems. The experiment conducted utilized speech quality data obtained from a collection of synthesized voices in a previous experiment [9]. A dimension reduction technique (PCA) was first employed to assess the relevance of each input feature, and select the relevant feature or feature subsets. Using SOM, the selected features were subjected to an unsupervised calibration and visualization – to cluster the target classes, and study the degree of association amongst these features. The SOM enabled the visualization of the selected dataset and evaluation of the feature dimensions of the overall speech quality. To ensure efficient knowledge discovery, DNN-Pattern recognition was exploited evaluated the performance of our classifier. An overall accuracy of 93.1% was achieved from a confusion matrix, signifying an efficient speech quality classifier.

References

1. Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., Kitamura, T.: Simultaneous modeling of spectrum, pitch and duration in HMM based speech synthesis. In: Proceedings of EUROSPEECH Conference (1999)
2. Zen, H., Oura, K., Nose T., Yamagishi, J., Sako, S., Toda, T., Masuko, T., Black, A.W., Tokuda, K.: Recent development of the HMM-based speech synthesis system (HTS). In: Proceedings of APSIPA Annual Summit and Conference, Sapporo, Japan, pp. 121–130 (2009)
3. Zen, H., Senior, A., Schuster, M.: Statistical parametric speech synthesis using deep neural networks. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 7962–7966 (2013)
4. Hunt, A., Black, A.: Unit selection in a concatenative speech synthesis system using a large speech database. In: Proceedings of ICASSP, Atlanta, Georgia, vol. 1, pp. 373–376 (1996)
5. Savargiv, M., Bastanfard, A.: Study on unit-selection and statistical parametric speech synthesis techniques. J. Comput. Robot. **2**(7–1), 19–25 (2014)
6. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. **11**, 3371–3408 (2010)
7. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
8. Deng, L., Dong, Y.: Deep Learning: Methods and Applications. Microsoft Research/NOW Publishers, UK (2014)
9. Ekpenyong, M.E., Inyang, U.G., Ekong, V.E.: A DNN framework for robust speech synthesis systems evaluation. In: Zygmunt, V., Mariani, H. (eds.) Proceedings of 7th Language and Technology Conference (LTC), Poznan, Poland, pp. 256261. Fundacja Uniwersytetu im. A. Mickiewicza (2015)
10. Cambell, N.: Evaluation of speech synthesis. In: Dybkjaer, L., Hamsen, H., Minker, W. (eds.) Evaluation of Text and Speech Systems. Text, Speech and Language Technology, vol. 37, pp. 29–64. Springer, The Netherlands (2007). https://doi.org/10.1007/978-1-4020-5817-2_2
11. Francis, A.L., Nusbaum, H.C.: Evaluating the quality of synthetic speech. In: Gardner-Bonneau, D. (ed.) Human Factors and Voice Interactive systems, pp. 63–97. Kluwer Academic, Boston (1999)
12. Morton, K.: Expectations for assessment techniques applied to speech synthesis. Proc. Inst. Acoust. **13**(2), 1–10 (1991)
13. Klatt, D.: Review of text-to-speech conversion for English. J. Acoust. Soc. Am. JASA **82**(3), 737–793 (1987)
14. Mariniak, A.: Global framework for the assessment of synthetic speech without subjects. In: Proceedings of Eurospeech, vol. 93, no. 3, pp. 1683–1686 (1993)
15. Goldstein, M.: Classification of methods used for assessment of text-to-speech systems according to the demands placed on the listener. Speech Commun. **16**, 225–244 (1995)
16. Logan, J., Greene, B., Pisoni, D.: Segmental intelligibility of synthetic speech produced by rule. J. Acoust. Soc. Am. JASA. **86**(2), 566–581 (1989)
17. Pisoni, D., Hunnicutt, S.: Perceptual evaluation of MITalk: the MIT unrestricted text-to-speech system. In: Proceedings of ICASSP, vol. 80, no. 3, pp. 572–575 (1980)
18. Bernstein, J., Pisoni, D.: Unlimited text-to-speech system: description and evaluation of a self organized maps. In: International Conference on Emerging Trends in Networks and Computer Communications (ETNCC), pp. 215–222 (1980)

19. Duffy, S.A., Pisoni, D.B.: Comprehension of synthetic speech produced by rule: a review and theoretical interpretation. Lang. Speech **35**, 351–389 (1992)
20. Kraft, V., Portele, T.: Quality evaluation of five German speech synthesis systems. Acta Acust. **3**(1995), 351–365 (1995)
21. Pavlovic, C., Rossi, M., Espesser, R.: Use of the magnitude estimation technique for assessing the performance of text-to-speech synthesis system. J. Acoust. Soc. Am. JASA **87**(1), 373–382 (1990)
22. Mannell, R.: Evaluation of speech synthesis systems. Macquarie University, Australia (2009). http://clas.mq.edu.au/speech/synthesis/synth_evaluation/. Accessed 26 June 2017
23. Clark, R.A., Dusterhoff, K.E.: Objective methods for evaluating synthetic intonation. In Proceedings of Eurospeech, vol. 4, pp. 1623–1626 (1999)
24. Kohonen, T.: Essential of self organizing maps. Neural Netw. **37**, 52–65 (2013)
25. Bengio, Y.: Learning deep architectures for AI. Found. Trends Mach. Learn. **2**(1), 1–127 (2009)
26. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: the RPROP algorithm. In: Proceedings of IEEE International Conference on Neural Networks, San Francisco, CA, USA, pp. 586–591 (1993)
27. Vasan, K., Surendiran, B.: Dimensionality reduction using Principal Component Analysis for network intrusion detection. Perspect. Sci. **8**, 510–512 (2016)