An Improved Voice Activity Detection Algorithm with an Echo State Network Classifier for On-device Mobile Speech Recognition

MOHAMMED K. MUSTAFA
Baze University, Abuja, Nigeria

TONY ALLEN & KOFI APPIAH
Nottingham Trent University, United Kingdom

ABSTRACT Speech recognition is a complex process that involves quite a number of steps to achieve. The complexity of this process becomes a bigger concern when speech recognition is to be implemented for on-device mobile speech recognition. The aim of this paper is to develop a mobile speech recognition system that does the entire processing on the mobile device. To achieve this, a two-stage speech recognition system is introduced. This system is developed from experimentation using available techniques in the literature and trying to improve on this and develop one that is suitable for a mobile device use. The first stage presents a novel Voice Activity Detection (VAD) technique that adopts Linear Predictive Coding Coefficients (LPC) that can be easily applied to on-device isolated word recognition on a mobile device. With recognition performance of 90% in comparison to a previous algorithm and recognition rate of 97.7% for female users in some of the experiments. The second stage adopts an Echo State network as the speech classifier with a performance of 76.82%. The developed system is suitable for mobile device use with the VAD showing significant improvement in terms of performance and processing speed.

Keywords: Speech recognition, Linear Predictive Coding, Mel Frequency Cepstral Coefficients, Linear cross correlation, Voice activity detection

Introduction

Speech recognition is the process by which a device understands and translates human utterances into a machine understandable format for further processing. Speech recognition by nature involves quite a number of processes to achieve an effective and efficient recognition system. However, there is a general convention within the speech recognition research field, that it is acceptable to establish a high performing speech recognition process without the need to undertake an effective evaluation of the impact of these high-performance algorithms within this process on the computational resources of the target device. In the generality of speech recognition, the processes can be divided into two stages. The pre-processing stage
and the recognition stage. Both stages are very important in order to attain a very good speech recognition system. The pre-processing stage serves as a precursor to the recognition stage (Mustafa et al., 2015), it is in this stage that the prospective audio signal that will be used for recognition is processed for use by the recognizer. There are different ranges of pre-processing which have laid the foundation for different speech recognition systems. These different pre-processing techniques are achieved either by the use of different filters or even much more sophisticated algorithms that use a lot of statistical analysis. The recognition stage on the other hand, takes the output of the pre-processing stage and uses it to recognize the spoken utterance. Techniques used as speech recognisers are probabilistic models such as Hidden Markov models (Schuller et al. 2003) and Gaussian mixture models (So and Paliwal 2006), and statistical such as Artificial neural networks (Skowronski and Harris, 2007a; 2007b; 2007c).

The predominant application of speech recognition on mobile devices adopt an on-line approach in order to get around the challenges of mobile device power, computation time and memory. Speech recognition on mobile devices is a very lucrative venture and as such the specific details of the activities involved in the process are hidden from the general public and literature due to the market competition involved between the predominant speech recognition systems for mobile phones which include Google’s Talk to Google, Apple’s SIRI and the most recent one to join the group being The Cortana from Microsoft for Microsoft mobile devices which all adopt the on-line approach.

It is obvious there are a good amount of techniques that can be used in the speech recognition process. However, because this research is interested in the deployment of speech recognition for on-device mobile speech recognition, it is very important to investigate and apply techniques that are both good in performance and at the same time offer real-time processing on the target mobile device.

**Voice Activity Detection (VAD)**

Voice Activity Detection (VAD) can be regarded as one of the pre-processing techniques that is utilised in some speech recognition systems. In VAD the principal objective is to find the respective regions of speech within an audio signal while ignoring the regions of silence for use in further processing. Not only does this help in reducing the computational cost but the processing time as well because it is a well-known fact that speech recognition is a computationally exhaustive task that imposes heavy demands on the processing power and storage capacity of any target device (Waheed et al. 2002) and there are no extra merits for passing a huge amount of speech data to such a device (Alarifi et al. 2011).

In the paper of (Huang and Lin 2009) LPC was used but instead of the standard deviation, the mean of the noise was used and then the speech was binarized with a certain threshold. This highlights the importance of noise in any VAD algorithm. In (Ghaemmaghami et al., 2010), auto correlation was used in conjunction with the Zero Crossing rate. However, this poses a dual calculation situation which is not desirable for the ultimate goal of mobile based VAD application. The consideration of the long-term speech information is important to a VAD algorithm as
argued by (Ramirez et al. 2004; Ghosh et al., 2011). In (Ramirez et al., 2004), the long-term speech divergence between the speech and non-speech frames of the entire signal is considered. This is then used to come up with thresholds by comparing the long term spectral envelope to the average noise spectrum. However, in (Ghosh et al., 2011) it is argued that the adoption of the average noise spectrum magnitude information is not attainable because in practice there is no stability or rather stationary noise. In (Ghosh et al., 2011) long term signal variability is considered. This is a measure computed using the last frames of the observed signal with respect to the current frame of interest. The only problem with a consideration of the last few frames of the speech signal in (Ghosh et al., 2011) or even the first few ones as described by (Ramirez et al., 2004) is that the availability of the desired noise spectrum cannot be guaranteed to lie in those regions.

We put emphasis on (Tashan et al., 2014) where it introduced the concept of linear cross correlation (LCC). This technique was very effective for VAD purposes in a speech verification system. The proposed VAD algorithm improves the processing time for the linear cross correlation algorithm for use on a mobile device. We also adopt the concept of the long term spectral information. However, the proposed algorithm does not use a predetermined region of the speech signal to find the noise spectrum as argued by (Ramirez et al. 2004; Ghosh et al., 2011). We adopt a dynamic approach to find the region with a higher noise spectrum. The choice of speech features for the VAD is the Linear Predictive Coding technique.

### Linear Predictive Coding (LPC)

The main idea behind this method is that a speech sample is approximated as a linear combination of past speech samples (Rabiner and Schafer 1978). Linear prediction is mostly used in low bit rate transmission of speech or storage (Rabiner and Schafer 1978). This is achieved by minimizing the sum of squared differences between the actual speech samples and the linearly predicted ones. LPC has also become the predominant technique for estimating the basic speech parameters such as formants, spectra, vocal tract area functions (Rabiner and Schafer 1978). There are several different methods of formulation of the linear predictive coding analysis. The Levinson–Durbin approach is used to compute the LPC features in this paper. The prediction of a speech sample \( x[n] \) is given in equation 1:

\[
x(n) = \sum_{i=1}^{P} a_i x(n-i)
\]

Where \( x(n) \) is the sample being predicted, \( x(n - i) \) is the previous sample, \( P \) is the order and \( a_i \) is the prediction coefficient.

### Proposed VAD System

In this paper we present a two stage/process based Voice Activity Detection algorithm. This algorithm is broken into two stages so they can be independently
implemented. This can provide a multipurpose outlook algorithm, where the user or researcher can use the first stage for further processing with their own algorithm whilst ignoring the second stage which is strictly our choice of implementation in this paper. The two stages involved are;

- Linear cross correlation stage
- K-means clustering stage (Modified)

**Linear Cross correlation Stage**

This technique was applied to Discrete Fourier Transform (DFT) speech frames where it correlates the DFT frames against each other, thereby resulting in a higher correlation value for frames containing speech and low value for frames without (Tashan et al. 2014). The correlation value is within the range of 0 and 1. The Linear cross correlation between two variables X and Y is given in equation 2

$$LCC(x, y) = \frac{\sum_{i=1}^{N} x_i y_i - \left(\sum_{i=1}^{N} x_i \right) \left(\sum_{i=1}^{N} y_i \right)}{\sqrt{\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i \right)^2}} \sqrt{\sum_{i=1}^{N} y_i^2 - \left(\sum_{i=1}^{N} y_i \right)^2}$$

where $x_i$ and $y_i$ are two vectors of N samples.

This concept was directly applied to the LPC features but the results were not as distinctive as the DFT results. Further work applied the same principle to the LPC residual signal (prediction error) which was even more inconclusive (Mustafa et al. 2014).

In this paper LCC being applied to the LPC was applied differently. After the computation of the LPC features using the Levinson – Durbin method. The frames containing LPC features of the entire signal are taken and equation 3 is used to compute the standard deviation (SD) of the respective frames. SD is a measure of noise and interference within a signal (Smith, 1997).

$$\sqrt{\sigma^2} = \frac{1}{N-1} \sum_{i=0}^{N-1} (x_i - \mu)^2$$  \hspace{1cm} (3)

where $\sigma^2$ is the variance, $x_i$ is the signal and $\mu$ is the mean of the signal and N total number of signals.

The frame with the minimum standard deviation is chosen as the primary frame. This is then used as the candidate frame to be correlated with every other frame including itself. This gives a correlation value for every frame within the audio signal. The frames with the lower correlation value are the frames with speech and vice versa. A pseudo code implementation is given in table 1.
A pseudo code Implementation Of the LCC Algorithm

Break A into Frames of 128 samples

for every F

    Compute LPC with Order 12 using Eqn. 1

end

for every LF

    Compute the SD using Eqn. 3

end

Using MinSD as x in Eqn. 2

for every LF as y in Eqn. 2

Compute LCC using Eqn. 2

Table 1. Pseudo Implementation of LCC code

Where A is audio signal, F is the frame, LF is the frame containing LPC, SD is used to denote the standard deviation and MinSD is the calculated minimum SD.

Figure 1 gives the initial output of the correlation algorithm, where the frames with speech are at the lower regions of the plot.

Figure 1. LCC on LPC Results

Reversing the vertical axis because the regions with speech are in the lower side of the plot will give figure 2.

Figure 2. LCC on LPC Results flipped
The reversed plot shows the position of the respective digits in the plot. This result can then be used for the stage 2 of our proposed system. However, this result can be used for a different algorithm to pick out the respective digits.

**K-means Clustering Stage**

The K-means algorithm was applied to the correlation results in figure 2. The K-means algorithm was used to cluster the correlation values into two clusters of speech and silence. K-means assigns feature vectors to clusters by the minimum distance assignment principle (Carl and Looney 1999; Žalik, 2008). The cluster centers are calculated using equation 4.

\[
J = \sum_{j=1}^{k} \sum_{i=1}^{n} ||x^{(j)}_i - c_j||^2
\]

\(||x^{(j)}_i - c_j||^2\) is a chosen distance between a data point \(x^{(j)}_i\) and the cluster centre \(c_j\).

**Modifications to K–Means**

The standard K-means algorithm was modified with data specific choices for each of the correlated results. A bias was introduced to help lower the threshold in order to do a more effective separation of the correlation results. This bias pushes the threshold for the algorithm above the median of the data. This allows enough of the speech data in the lower region to be captured. The equations for the bias calculation are given in equations 5 – 7.

\[
C = \left( \sum_{i=1}^{N} X_i \right)/N
\]

\[
\text{bias} = (C_A - C_B) \times 0.33
\]

\[
C_A = C_B + \text{bias}
\]

Where \(X_i\) is the mean of the \(i^{th}\) frame within the cluster, \(C\) is the centroid of the respective clusters and \(N\) the total number of frames within the cluster.
Table 2. K-means implementation

A Pseudo code Implementation Of the K-means Algorithm
for Iteration = 1
Compute Frame with Min and Max Correlation Value of A
Assign Min as centroid of CA & Max as centroid of CB
for every CF applying bias in Eqn. 6
  if ED of (CA - CF) < (CB - CF)
    Assign to CA
  else if ED of (CB - CF) < (CA - CF)
    Assign to CB
end
Calculate CA and CB using Eqn. 5
If(new CA and CB = Previous CA and CB) or (Iteration = 100) terminate
end

Where $A$ is the audio signal, $C_A$ is cluster $A$, $C_B$ is cluster $B$, $CF$ is the correlated frame and $ED$ is the Euclidean distance.

The pseudo code implementation of the Modified K-means is given in table 2 for a clear understanding of how the decisions are made. After the implementation of the K-means clustering algorithm, the result is two clusters, one with speech frames and the second with silence. The cluster with speech is used for further processing or verification.

Experimental Data

A total of 20 samples from 20 different speakers with the digits 1 – 9 (total of 180 digits) were used in total for these algorithms. These samples are from the Centre for Spoken Language Understanding database (CSLU2002). This is a commercially available database from the Oregon Health & Science University. These audio samples have a sampling rate of 8kHz and encoded in 16 bits. The first 10 were used as a guide set to modify and apply the different variable adjustments to come up with an initial guide with regards to the adjustable variables of the experiment explained a below with the results in table 3. The second set of 10 audio samples was used to test the algorithm against a previous work (Mustafa et al. 2014), where the same samples were used. The results are given in table 4.

A third algorithm was developed to test the respective digits. These digits were grouped using a decision algorithm that uses 2 variables to make its decision. This is also an improvement to the test algorithm in (Mustafa et al. 2014). Only the cluster with speech is passed on to this algorithm to try and find out where the respective digits are. These two variables are:
• Frame Distance: This variable is the distance/pause between the frames containing the respective digits. There are 9 digits with pauses in between them. This can be adjusted and there is no definite size to this particular variable as the speed of speech cannot be controlled because it depends completely on the individual speaking. However we made attempts to try to generalize this as this will be shown in the results.

• Inter Frame Distance: This particular variable is used to complement the effort of the first in the grouping of the digits. There are cases where there is a little silence in the utterance of a digit. An example of this is the digit 6, there is a slight break between the “SI” part of it and the “X” part of 6. This break using the first variable could group them into two different groups as separate digits. The inter frame distance checks every respective group to see if it meets a set minimum. If it does not attain the set minimum this then tries to assign it to the group closest to it using the Euclidean Distance measure.

Different adjustments to these variables were used to come up with an acceptable optimal number. A number of experiments were conducted in this regard and then the same 90 were used with our previous work to compare and see if there was any improvement. However, it is worthy of note that as a mobile device intended system it is important that we have variables that can be adjusted. This can be adopted to the system as measures of customization.

After the adjustments of the respective variables, the algorithm developed is in the form of a mobile application. This is deployed on the mobile phone and this is then used to verify the respective digits. The application plays the respective digits and this is audibly verified. Only the set of digits that were fully recognised were recorded as recognised. The digits that were still broken or split by the algorithm were ignored in the computation of the final result numbers.

Echo State Networks (ESN)

Echo state networks have recently garnered a lot of interests as a good choice of recurrent neural network (RNN) architecture for use with time series prediction. However, the application of the ESN to speech recognition is still very new and research is still on going in this field. As such, there is no final best architecture for using such networks. Echo State networks are fundamentally Recurrent Neural Networks, however, the difference is that, only the output weights of the network are trained. (Skowronski and Harris 2007a) presents an architecture that adopts a left-right arrangement of a competitive set of network filters. These filters are arranged to form a state, with transitions between the states governed by a winner take all strategy. In (Skowronski and Harris 2007c), the same network is used for a noise robust system. A different set of 12 human factor cepstral coefficients and the log frame energy are combined as the input features. A further extension is applied in (Skowronski and Harris 2007b) by adopting a more discriminative approach which employs an offset asymptote using the hyperbolic tangent transfer function. A teacher forced approach is used in (Hmad and Allen 2013) to train the output weights. These teacher output weights are produced using a maximum like-
likelihood criterion stage on the network output before being fed back into the network. A Pseudo-inverse method is used to train the network output weights.

The ESN architecture adopted in this work is based on the original concept suggested by (Jaeger, 2001) and the adoption of a single feedback structure for the network creates a transitional behaviour between the frames presented to the network representing the left to right ordering of the speech signals (Schuller et al., 2003). This network structure is used for phoneme based recognition. This network architecture does not necessarily present the best Echo State Network for speech recognition but it presents a plausible network architecture that can be used on a mobile device with minimum emphasis on the computational and memory resources. The network is trained on the computer and the weights are then transferred to the mobile device for testing.

The reservoir units are calculated using equation 8 and the output units are calculated using equation 9:

\[
    h(n + 1) = f(W^{in}i(n + 1) + W^{h(n)})
\]

\[
    o = f(W^{out}(i(n + 1), h(n + 1)))
\]

where \( h \) is a unit in the hidden layer, \( f \) is the activation function, \( W^{in} \) is the weight connecting the input and reservoir units, \( i \) is the input to the network, \( W^{h(n)} \) is the weight of the previous hidden unit and \( W^{out} \) is the weight connecting the reservoir unit and output unit.

The training procedure adopted for the output weights of the network is the pseudo-inverse training algorithm. The pseudo inverse of the reservoir activated units are used to train the output weights (Lukoševičius, 2012). The concept of the pseudo inverse training algorithm is to concatenate the input and hidden nodes into a single column matrix. This is done for all the respective time steps that constitute the output, although for the case of the architecture adopted here, there is only one time step. One of the reasons for adopting the single time step and this particular architecture was to limit the memory intensive calculations and storage of previous time steps within the network. To calculate the weight updates during training, equation 10 is used.

\[
    W^{out} = \hat{O}^{\text{desired}}X^T(XX^T)^{-1}
\]

where \( \hat{O}^{\text{desired}}X^T \) is a matrix containing the targets of the network with regards to all the stored concatenated values of \( X \) and \( XX^T \) is a second matrix containing all the concatenated values with regards to previous states. \( W^{out} \) is the same as equation 9.

**Echo State Network Data Preparation**

The data used for the ESN experimentations is from the same CSLU database. The same 20 speakers were used. However, the number of samples per speaker was
increased from 1 to a total of 16 samples per speaker. This results in a total of 2880
digits for training the network. This was initially split into a training dataset of
2520 and a testing dataset of 360 digits. These digits are also made up of the digits
1 – 9. The different number of samples from each speaker compensates for differ-
et speech speed and different environmental factors. To train the ESN, a teacher
forced training is used to train the ESN in this work. Specifically, a Self-organising
map is used on MATLAB to obtain the teacher labels for the respective audio sam-
ples. Table 3 gives a brief classification used for the respective labels., resulting in
a total of 17 phonemes.

Table 3 - Digit Phoneme labels

<table>
<thead>
<tr>
<th>Digit</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 3 - Digit Phoneme labels

These phoneme labels are used as the total output of the SOM classification.
These outputs are used as the teacher output of the ESN network. The SOM experi-
ment is done by using a single SOM for the entire audio data. This is to create co-
hesion between the data the network is being trained with. However, this presents a
very difficult controlled labelling of the output of the SOM.

**ESN Experimentations**

The intended adoption of ESN is so that it can easily adopt the output of the VAD
algorithm as input. The ESN is evaluated such that the output of a sequence is cal-
culated to have recognised the sequence if it is able to identify at least 70% of the
frames within this sequence. This is important especially in cases of digits with
more than 2 phonemes. The 70% evaluation allows some of the 3rd phonemes
frames to be captured in the evaluation. This evaluation is also due to the fact that
there is no clustering done prior to feeding the frames of a particular digit to the
network. This invariably means there is no classification of the frames into an aver-
aged single phoneme representation but many frames representing a single pho-
neme. This as explained is to avoid a clustering or segmentation stage between the
VAD algorithms developed in (Mustafa et al., 2014; 2015) and the speech recogni-
iser to avoid extra calculations.
Table 4 - Echo State Network Forward Pass

<table>
<thead>
<tr>
<th>Echo State Network Forward Pass</th>
</tr>
</thead>
<tbody>
<tr>
<td>For every $D \in N$</td>
</tr>
<tr>
<td>For every $F \in D$</td>
</tr>
<tr>
<td>Calculate $R$ using equation 3</td>
</tr>
<tr>
<td>$Net_1 = f(R)$</td>
</tr>
<tr>
<td>save $Net_1$ as $PR$</td>
</tr>
<tr>
<td>Propagate to the Output</td>
</tr>
<tr>
<td>Calculate $O$ using equation 4</td>
</tr>
<tr>
<td>$Net_2 = f(O)$</td>
</tr>
<tr>
<td>End</td>
</tr>
<tr>
<td>End</td>
</tr>
</tbody>
</table>

Table 4 - Echo State Network Forward Pass

$N$ is the total number of digits, $D$ is used to denote the digit, $F$ is the frame, $R$ represents the reservoir, $PH$ is the previous reservoir unit, $NF$ is the new frame passed through the network and $Win$ is the input weight.

The forward pass through the network is summarised in table 4. During training this pass is applied to every digit sequence until the entire training data is exhausted. After the forward pass is calculated for all the digits within the training data, the weights of the network are updated. The respective reservoir states are stored during the forward pass procedure. The stored reservoir states are then passed into the training part of the algorithm where the weights are updated using the pseudo inverse procedure. The different experiments conducted are presented in terms of the global parameters to be modified. The following section defines these parameters and explain the experiments conducted for this purpose. There are no set globally accepted values for these parameters. As argued by (Lukoševičius, 2012) the only way to go about modifying these parameters is to do it one at a time. This is because the network configuration also affects how these parameters work in unison with each other.

Parameter Modifications

- Spectral radius: It is defined as the maximum eigenvalues of the reservoir weights. As originally proposed this has to be less than 1 (usually between 0 and 1) to maintain the echo state property of the network as well as the stability of the network. As recommended by (Venayagamoorthy and Shishir 2009) this should be modified in successive multiples of 0.2 between 0.8 and 0.2. However, a higher number was adopted to cover the range better.
• Input Connectivity: This is the connectivity of the input to the reservoir i.e. the weights connecting the input layer to the reservoir layer. This is 1 for full connectivity and 0 for no connectivity.
• Interconnectivity: This variable is used to control the amount of connection that exists between the current reservoir node and the previous reservoir node of the network.
• Weight range: This is the range of the random weights to be generated for the input and output weights.

The respective parameters of the network defined above are then modified one at a time. The spectral radius of the reservoir weights connectivity was the first choice of the parameters to be modified. The other parameters of the network were maintained as suggested by (Lukoševičius, 2012) to modify one parameter at a time. Thus, the other parameters were modified after the spectral radius and subsequently this followed all the way till the weight range was modified. The concept applied is to modify one of the parameters and after obtaining the best performance within the range maintain it and move to the next parameter and so on until the last parameter.

VAD Results

<table>
<thead>
<tr>
<th>Frame Distance</th>
<th>Inter Frame Distance</th>
<th>Recognized Male (SD)</th>
<th>Recognized Female (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Minimum</td>
<td>Next minimum</td>
<td>Minimum</td>
</tr>
<tr>
<td>&gt;=2 &lt;=6</td>
<td>77.70%</td>
<td>71.70%</td>
<td>91.10%</td>
</tr>
<tr>
<td>&gt;=2 &lt;=7</td>
<td>77.70%</td>
<td>66.60%</td>
<td>93.30%</td>
</tr>
<tr>
<td>&gt;=2 &lt;=8</td>
<td>75.50%</td>
<td>71.10%</td>
<td>91.10%</td>
</tr>
<tr>
<td>&gt;=2 &lt;=9</td>
<td>73.30%</td>
<td>68.80%</td>
<td>93.30%</td>
</tr>
<tr>
<td>&gt;=2 &lt;=10</td>
<td>75.50%</td>
<td>66.60%</td>
<td>93.30%</td>
</tr>
<tr>
<td>&gt;=3 &lt;=6</td>
<td>75.50%</td>
<td>71.10%</td>
<td>86.60%</td>
</tr>
<tr>
<td>&gt;=3 &lt;=7</td>
<td>75.50%</td>
<td>68.80%</td>
<td>86.60%</td>
</tr>
<tr>
<td>&gt;=3 &lt;=8</td>
<td>80%</td>
<td>71.10%</td>
<td>88.80%</td>
</tr>
<tr>
<td>&gt;=3 &lt;=9</td>
<td>82.20%</td>
<td>71.10%</td>
<td>88.80%</td>
</tr>
<tr>
<td>&gt;=3 &lt;=10</td>
<td>77.70%</td>
<td>68.80%</td>
<td>88.80%</td>
</tr>
<tr>
<td>&gt;=4 &lt;=6</td>
<td>82.20%</td>
<td>62.20%</td>
<td>84.40%</td>
</tr>
<tr>
<td>&gt;=4 &lt;=7</td>
<td>80%</td>
<td>57.70%</td>
<td>82.20%</td>
</tr>
<tr>
<td>&gt;=4 &lt;=8</td>
<td>84.40%</td>
<td>62.20%</td>
<td>84.40%</td>
</tr>
<tr>
<td>&gt;=4 &lt;=9</td>
<td>80.00%</td>
<td>62.20%</td>
<td>82.20%</td>
</tr>
<tr>
<td>&gt;=4 &lt;=10</td>
<td>75.50%</td>
<td>53.30%</td>
<td>82.20%</td>
</tr>
</tbody>
</table>

Table 5. 10 Initial Samples
Table 6. 10 samples from the previous paper (Mustafa et al., 2014)

**ESN Results**

As explained earlier the different parameters were modified in different combinations. The experiments conducted used the same LPC coefficients as the feature choices. The results in table 7 are based on automatic phonetic labelling of the frames. All the data representing the entire digit sequences of 1 – 9 are presented, and they are all labelled in one process.
Different applications of the parameters for the network have been suggested in
different papers. However, due to the difference in architecture these parameters
are not suitable for direct adoption. The output of these experiments is the respec-
tive recognised frames. However, for evaluation purposes, a post processing stage
was applied to get a final word recognition performance for a particular digit. The
chosen method of evaluation was that, the network was deemed to have recognised
a digit if 70% of the frames in a particular digit were correct.

Using the 70% evaluation of frames, the highest digit recognition achieved
was 86.46% on the training data using 0.5 weight range, 0.4 spectral radius, 0.8
input connectivity and 0.8 interconnectivity with 400 reservoir units. Given this
rather low test performance on the training data and in addition to the computa-

<table>
<thead>
<tr>
<th>Spectral Radius</th>
<th>Input</th>
<th>Inter connectivity</th>
<th>Weight Range</th>
<th>Recognition %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.9</td>
<td>0.97</td>
<td>0.5</td>
<td>75.41%</td>
</tr>
<tr>
<td>0.8</td>
<td>0.9</td>
<td>0.97</td>
<td>0.5</td>
<td>75.92%</td>
</tr>
<tr>
<td>0.6</td>
<td>0.9</td>
<td>0.97</td>
<td>0.5</td>
<td>69.97%</td>
</tr>
<tr>
<td>0.4</td>
<td>0.9</td>
<td>0.97</td>
<td>0.5</td>
<td>54.14%</td>
</tr>
<tr>
<td>0.2</td>
<td>0.9</td>
<td>0.97</td>
<td>0.5</td>
<td>76.82%</td>
</tr>
<tr>
<td>0.4</td>
<td>0.8</td>
<td>0.97</td>
<td>0.5</td>
<td>74.73%</td>
</tr>
<tr>
<td>0.4</td>
<td>0.6</td>
<td>0.97</td>
<td>0.5</td>
<td>75.15%</td>
</tr>
<tr>
<td>0.4</td>
<td>0.4</td>
<td>0.97</td>
<td>0.5</td>
<td>70.33%</td>
</tr>
<tr>
<td>0.4</td>
<td>0.2</td>
<td>0.97</td>
<td>0.5</td>
<td>67.37%</td>
</tr>
<tr>
<td>0.4</td>
<td>1</td>
<td>0.97</td>
<td>0.5</td>
<td>76.35%</td>
</tr>
<tr>
<td>0.4</td>
<td>0.8</td>
<td>1</td>
<td>0.5</td>
<td>71.70%</td>
</tr>
<tr>
<td>0.4</td>
<td>0.8</td>
<td>0.8</td>
<td>0.5</td>
<td>74.58%</td>
</tr>
<tr>
<td>0.4</td>
<td>0.8</td>
<td>0.6</td>
<td>0.5</td>
<td>54.16%</td>
</tr>
<tr>
<td>0.4</td>
<td>0.8</td>
<td>0.4</td>
<td>0.5</td>
<td>74.93%</td>
</tr>
<tr>
<td>0.4</td>
<td>0.8</td>
<td>0.2</td>
<td>0.5</td>
<td>73.61%</td>
</tr>
<tr>
<td>0.4</td>
<td>0.8</td>
<td>0.8</td>
<td>0.4</td>
<td>54.30%</td>
</tr>
<tr>
<td>0.4</td>
<td>0.8</td>
<td>0.8</td>
<td>0.3</td>
<td>54.30%</td>
</tr>
<tr>
<td>0.4</td>
<td>0.8</td>
<td>0.8</td>
<td>0.2</td>
<td>71.22%</td>
</tr>
<tr>
<td>0.4</td>
<td>0.8</td>
<td>0.8</td>
<td>0.1</td>
<td>69.52%</td>
</tr>
<tr>
<td>0.4</td>
<td>0.8</td>
<td>0.8</td>
<td>0</td>
<td>69.07%</td>
</tr>
</tbody>
</table>

Table 7. 17 output SOM used for ESN with LPC features on training data
time of 29.65 secs for each minute of audio data on the mobile phone, there is a need for further experimentation using this technique.

**Discussion**

It can be seen from the VAD results that this algorithm and detection outperforms the algorithm and detection algorithm in the last paper (Mustafa, Allen et al. 2014). The optimal performance of the algorithm in that paper was 86.6% and that result compared to the result in this experiment on the same data with 90% shows a significant improvement with regards to the algorithm detection technique.

The performance of the female samples over the male samples in table 3 is not unconnected to the smoothness and pitch of the female voice. However, due to the average sound of the male speaker, it can be noted that there is a lot of vibrations in the male speech. This can be seen by the performance of this algorithm by simply comparing the female speakers which is 97.7% at best with the male speakers which is 84.4%. It is also worthy of note that the only comparable leap was achieved with the minimum SD where the performance for both was 84.4%. After computing the LPC features, the LCC stage of this experiment takes 2.76 secs as compared to the original DFT in (Mustafa, Allen et al., 2014) which took 49.75 secs. This also shows a faster processing time on the same mobile device.

The task of assigning the frame distance and inter frame distance is very tricky. However, for an on–device mobile device implementation of speech recognition these variables can prove very important as they allow a degree of customization to the particular user of that mobile phone. This will improve the problem of generalization with regards to speech recognition and as such helps to concentrate a system down to the user.

It is worthy of note that both the minimum and next minimum standard deviation cannot be ignored for implementation purposes. The results in table 5 shows that in female cases most especially the next minimum outperforms the minimum. This can also be an advantage to the implementation of a user specific system for this detection. These variables can be implemented in a dynamic approach to try and adjust them to a user. The results of the VAD are carried over to the ESN stage.

Table 7 presents the performance of the ESN, with a best performance of 76.92%. ESN have been shown to perform better and in some cases close to probabilistic models such as HMM. However, this performance is subject to how the system is applied to a task. The pre-processing applied to this method adopted a Self-organising map classification. In evaluating the ESN, it is worthy of note to consider this fact. As such, this cannot be taken as the best performance of an ESN system to speech recognition. However, this is indicative of limitations in applying ESN.

The ESN is trained using the classification outcome of the SOM in table 3. After training the ESN, the output frames of the VAD are passed into it. This has some disparities. In order to have an effective full system, the ESN has to be trained using the VAD outcome.
Conclusion

The VAD algorithm developed has shown great improvement as compared to the previous paper. This shows that it is a very good candidate for performing VAD. The ESN network on the other hand requires improvement to adopt it for mobile device use. This paper is based on a mobile device implementation of a speech recognition system. This means there are limitations in terms of memory and processing speed. The algorithms developed in this paper adopted speeding up techniques and not high memory consuming architecture for the ESN. The results have shown that there is very good improvement for the VAD algorithm but there is need for further improving the ESN.

Correspondence
Mohammed K. Mustafa
Baze University, Cadastral Zone C 00
686 Jabi Airport Road Bypass
Abuja, Nigeria
E-mail: mohammed.mustafa@bazeuniversity.edu.ng
References


Smith, S.W., (1997.) The Scientist and Engineer's guide to Digital Signal Processing.


