Arabic Phoneme Recognition using Hierarchical Neural Fuzzy Petri Net and LPC Feature Extraction

Ghassaq S. Mosa ghassaqsaeed@yahoo.com
College of Engineering/Department of Computer Engineering
University of Basrah
Basrah, Iraq.

Abduladhem Abdulkareem Ali aduladhem@compengbas.net
College of Engineering/Department of Computer Engineering
University of Basrah
Basrah, Iraq.

Abstract

The basic idea behind the proposed hierarchical phoneme recognition is that phonemes can be classified into specific phoneme types which can be organized within a hierarchical tree structure. The recognition principle is based on "divide and conquer" in which a large problem is divided into many smaller, easier to solve problems whose solutions can be combined to yield a solution to the complex problem. Fuzzy Petri net (FPN) is a powerful modeling tool for fuzzy production rules based knowledge systems. For building hierarchical classifier using Neural Fuzzy Petri net (NFPN), Each node of the hierarchical tree is represented by a NFPN. Every NFPN in the hierarchical tree is trained by repeatedly presenting a set of input patterns along with the class to which each particular pattern belongs. The feature vector used as input to the NFPN is the LPC parameters.

Keywords: Hierarchical networks, Linear predictive coding, Neural fuzzy Petri net, phoneme recognition, Speech recognition.

1. INTRODUCTION

The Arabic Language is one of the oldest living languages in the world. The bulk of classical Islamic literature was written in classical Arabic (CA), and the Holy Qur'an was revealed in the Classical Arabic language. Standard Arabic is the mother (spoken) tongue for more than 200 million people living in the vast geographical area known as the Arab world, which includes countries such as Iraq, Syria, Jordan, Egypt, Saudi Arabia, Morocco, and Sudan. Arabic is one of the world's oldest Semitic languages, and it is the fifth most widely used. Arabic is the language of communication in official discourse, teaching, religious activities, and in literature.

Many works have been done on the recognition of Arabic phonemes. These studies include the use of neural networks [1-3], Hidden Markov Model [4] and Fuzzy system [5].

Hierarchical approaches based on neural networks were employed in other languages with different techniques [5-7] . In this paper hieratical Arabic phoneme recognition system is proposed based on LPC feature vector and neural Fuzzy Petri Net (NFPN). The principle is based on proposing a decision tree for the Arabic phonemes. NFPN is used as a decision network in each

node in the tree.

2. ARABIC LANGUGE ALPHABET

Every language is typically partitioned into two broad categories: vowels and consonants. Vowels are produced without obstructing air flow through the vocal tract, while consonants involve significant obstruction, creating a nosier sound with weaker amplitude. The Arabic language consists of 28 letters, Arabic is written from right to left and letters take different forms depending on their position in a word; some letters are similar to others except for diacritical points placed above or beneath them. Arab linguists classify Arabic letters into two categories: sun and moon. Sun letters are indicated by an asterisk. When the sun letters are preceded by the prefix Alif-Laam in nouns, the Laam consonant is not pronounced. The Arabic language has six different vowels, three short and three long. The short vowels are fatha (a), short kasrah (i), and short dammah (u). No special letters are assigned to the short vowels; however special marks and diacritical notations above and beneath the consonants are used. The three long vowels are durational allophones of the above short vowels, as in mad, meet, and soon and correspond to long fatha, long kasrah, and long dammah respectively. Consonants can be also un-vowelised (not followed by a vowel); in this case a diacritic sakoon is placed above the Consonant. Vowels and their IPA (International Phonetic Alphabet) equivalents .

.

3. SPEECH RECOGNITION SYSTEM

The general model for speech recognition system here are five major phases recording & digitalizing speech signal, segmentation, pre-processing signal, feature extraction and decision-making, each phase will be explained in more details along with the approaches used to enhance the performance of the speech recognition systems.

A. A/D Conversion: The input speech signal is changed into an electrical signal by using a microphone. Before performing A/D conversion, a low pass filter is used to eliminate the aliasing effect during sampling. A continuous speech signal has a maximum frequency component at about 16 KHz.

B. Segmentation: Speech segmentation plays an important role in speech recognition in reducing the requirement for large memory and in minimizing the computation complexity in large vocabulary continuous speech recognition systems. [8].

C. Preprocessing: Preprocessing includes filtering and scaling of the incoming signal in order to reduce the noise and other external effect. Filtering speech signal before recognition task is an important process to remove noise related to speech signal which may be either low frequency or high frequency noise. Figure (1) shows the effect of preprocessing on signal.

D. Feature Extraction: The goal of feature extraction is to represent any speech signal by a finite number of measures (or features) of the signal. This is because the entirety of the information in the acoustic signal is too much to process, and not all of the information is relevant for specific tasks. In present ASR systems, the approach of feature extraction has generally been to find a representation that is relatively stable for different examples of the same speech sound, despite differences in the speaker or environmental characteristics, while keeping the part that represents the message in the speech signal relatively intact [9].

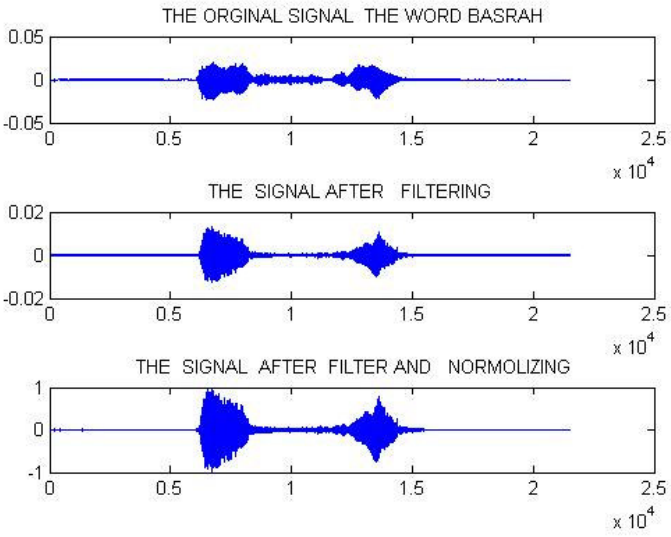


FIGURE 1: The word Basrah before and after filtering and normalizing.

4. LINEAR PRIDICTIVE CODING

Linear predictive analysis has been one of the most powerful speech analysis techniques since it was introduced in the early 1970s [10]. The LPC is a mode1 based on the vocal tract of human beings [11]. Figure (2) shows the block diagram of the LPC calculations.

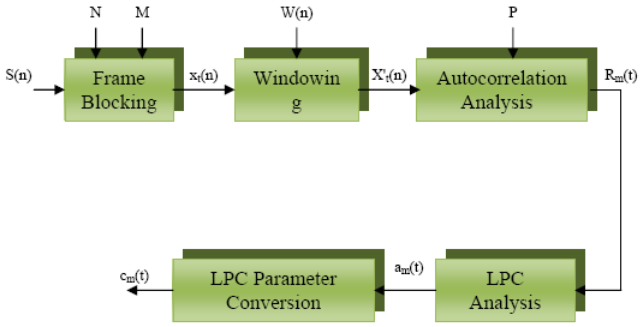


FIGURE 2: The LPC block diagram

A. **Frame Blocking:** The digitalized speech signal, S (n), is blocked into frames of N samples, with adjacent frames being separated by M samples. If we denote the lth frame of speech by xl(n), and there are L frames within the entire speech signal, then

$$x_l(n) = s(M_l + n), \qquad n = 0,1,....N-1, l = 0,1,...L-1$$

B. **Windowing:** To minimize the discontinuity and therefore preventing spectral leakage of a signal at the beginning and end of each frame, every frame is multiplied by a window function.

$$w(n) = 0.54 - 0.45\cos\left(\frac{2\pi n}{N-1}\right) \quad (1)$$

$$0 \le n \le N-1$$

Figure (3) shows the effect of windowing.

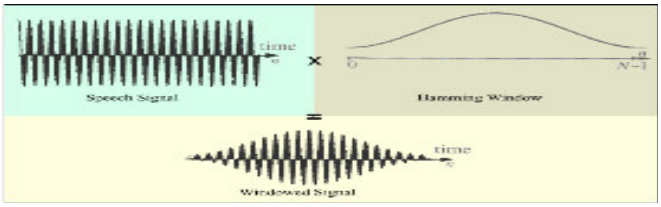


FIGURE 3: Effect of window on signal.

C. Autocorrelation Analysis: In this step, each frame of Windowed signal is auto correlated to give:

$$r_l = \sum_{n=0}^{N-1-M} x_l'(n)x_l'(n+m)$$

$$m = 0,1,2,\ldots\ldots,p \quad (2)$$

Where $x_l'(n)$ is the windowed signal,

Where highest correlation value, p, is the order of LPC analysis. Typically, values of p from 8 to 16 are used. It is interesting to note that the zeroth autocorrelation, $r_0(a)$, is the energy of l^{th} frame. The frame energy is an important parameter for speech-detection. In our case [10,11].

$$r_{xx}(1) = \sum_{n=-\infty}^{\infty} x(n)x(n-1) = \sum_{n=-\infty}^{\infty} x(n+1)x(n)$$

For l=0,1,2,3…….. (3)

D. LPC Analysis: The next processing step is the LPC calculation, which convert each of the auto correlated frame into an "LPC parameter set", in which the set might be the LPC coefficients, the reflection (or PARCOR) coefficients, the log are ratio coefficients and the cepstral coefficients. The formal method for converting from autocorrelation coefficients to an LPC parameter set (for the LPC autocorrelation method) is known as Durbin's method and can formally given as the following algorithm for convenience omitting the subscript l or $r_l(n)$,[10].

$$E^0 = r(0) \quad (4)$$

$$K_i = \left\{r(i) - \Sigma_{j=1}^{i-1} a^{i-1}{}_j\, r(|i-j|)\right\} \Big/ E^{(i-1)} \quad (5)$$

$$1 \le i \le p$$

$$a^{(i)}{}_i = K_i \quad (6)$$

$$a^{(i)}{}_j = a^{(i-1)}{}_j - K_i a_{i-j}{}^{i-1} \quad (7)$$

$$E^{(i)} = (1 - K_i^2)E^{(i-1)} \quad (8)$$

Where, the summation in Equation (5) is omitted for i=1, the set of Equation(4 -8) are solved recursively for i=1,2,…P, and the final solution is given as[11] :

a_m = LPC coefficients = $a^{(p)}{}_m$,

$1 \leq m \leq p$ (9)

K_m =PARCOR Coefficients (10)

g_m = Log area ratio coefficients

$= log\left[\frac{1-K_m}{1+K_m}\right]$ (11)

D. LPC Parameter Conversion to Cepstral Coefficients: Avery important LPC parameter set, which can be driven from the LPC coefficient set, is the LPC cepstral coefficients Cm, that is calculated by the following Equation:

$$C_{m=}a_m + \sum_{k=1}^{m-1}\left[\frac{k}{m}\right]c_k a_{m-k}:$$

$$1 \leq m \leq p$$

Where a_m is LPC coefficients

5. FUZZY NEURAL PETRI NET

After extracting the desired features from the input data, they are applied to the decision making stage in order to make the appropriate decision on the specific class that the input data belongs to. In this work NFPN is used as the decision making network.

Petri nets, developed by Carl Adam Petri in his Ph.D. thesis in 1962, are generally considered as a tool for studying and modeling of systems. A Petri net (PN) is foremostly a mathematical description, but it is also a visual or graphical representation of a system. The application areas of Petri nets being vigorously investigated involve knowledge representation and discovery, robotics, process control, diagnostics, grid computation, traffic control, to name a few high representative domains [12]. Petri nets (PNs) are a mathematical tool to describe concurrent systems and model their behavior. They have been used to study the performance and dependability of a variety of systems.

Petri Nets essentially consist of three key components: places, transitions, and directed arcs [13]. The directed arcs connect the places to the transitions and the transitions to the places. There are no arcs connect transitions to transitions or places to places directly. Each place contains zero or more tokens. A vector representation of the number of tokens over all places defines the state of the Petri Net. A simple Petri Net graph is shown in Figure (4). The configuration of the Petri Net combined with the location of tokens in the net at any particular time is called the Petri Net s

Petri Net structure is formally described by the five-tuple (P, T, I, O, M),

Where P is the set of places {p1, .., pn},

T is the s I is the set of places connected via arcs as inputs to transitions,

O is the set of places connected via arcs as outputs from transitions,

and M is the set of places that contain tokens et of transitions {t1, .., tm}, Formal Definition and State of Figure (4)

(P, T, I, O, M):

P = { p1, p2, p3, p4 }

T = { t1, t2 }

I = {{ p1 }, { p2, p3 } }

O = {{ p2, p3 }, { p4 } }

M = {1, 0, 0, 0}

The structure of the proposed Neural Fuzzy Petri Net is shown in figure (5) and (6). The network has the following three layers

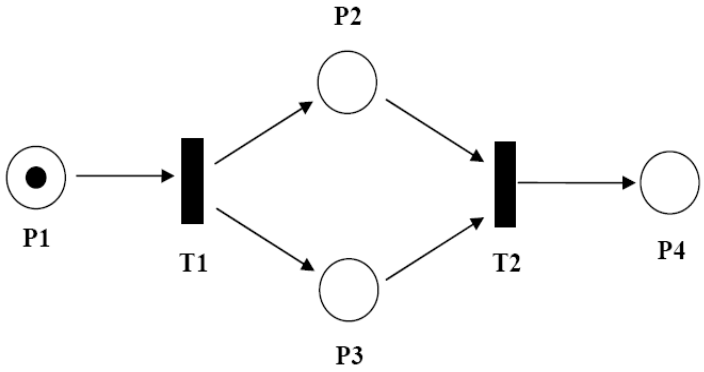


FIGURE 4: Petri Net graph.

:
- an input layer composed of n input
- a transition layer composed of hidden transitions;
- an output layer consisting of m output places.

The input place is marked by the value of the feature. The transitions act as processing units. The firing depends on the parameters of transitions, which are the thresholds, and the parameters of the arcs (connections), which are the weights. Each output place corresponds to a class of pattern. The marking of the output place reflects a level of membership of the pattern in the corresponding class [14].

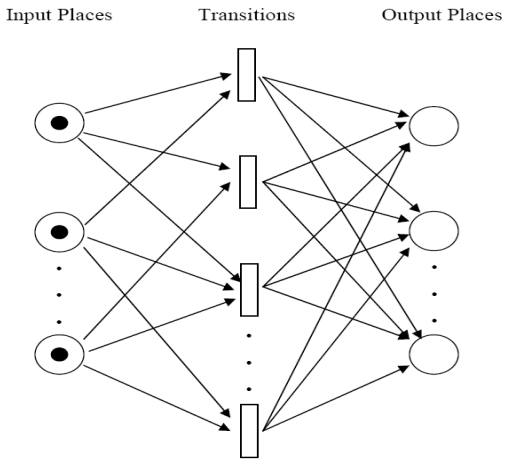


FIGURE 5: The structure of the Neural Fuzzy Petri Net.

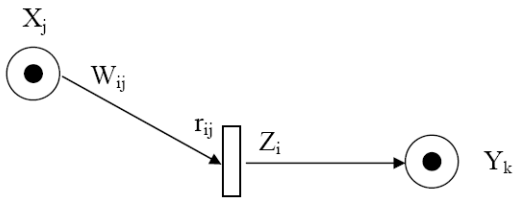


FIGURE 6: Section of the net outlines the notations.

The specifications of the network are as follows:
- Xj is the marking level of j-th input place produced by a triangular mapping function. The top of the triangular function is centered on the average point of the input values. The length of triangular base is calculated from the difference between the minimum and maximum values of the input. The height of the triangle is unity. This process keep the input of the network within the period [0,1]. This generalization of the Petri net will be in full agreement with the two-valued generic version of the Petri net [14].

$$x_j = f(input(j)) \tag{13}$$

Where f is a triangular mapping function

$$f(x) = \begin{cases} \dfrac{x - \min(x)}{average(x) - \min(x)}, & if \quad x < average(x) \\ \dfrac{\max(x) - x}{\max(x) - average(x)}, & if \quad x > average(x) \\ 1, & if \quad x = average(x) \end{cases}$$

- Wij is the weight between the i-th transition and the j-th input place;
- rij is a threshold level associated with the level of marking of the j-th input place and the i-thtransition;
- Zj is the activation level of i-th transition and defined as follows:

$$Z_i = \prod_{j=1}^{n} [W_{ij} S(r_{ij} \to X_j)], \quad j = 1,2,\ldots, n;\ i = 1,2,\ldots, \text{hidden} \tag{14}$$

Y_k is the marking level of the k-th output place produced by the transition layer and performs a nonlinear mapping of the weighted sum of the activation levels of these transitions (Zi) and the associated connections Vjk

$$Y_k = f\left(\sum_{i=1}^{No.ofTransitions} V_{ki} Z_i \right), \qquad j = 1,2,...,m \tag{15}$$

Where " f " is a nonlinear monotonically increasing function from R to [0,1].

Learning Procedure

The learning process depends on minimizing certain performance index in order to optimize the network parameters (weights and thresholds). The performance index used is the standard sum of squared errors[4].

$$E = \frac{1}{2}\sum_{k=1}^{m}(t_k - y_k)^2 \quad (16)$$

Where tk is the k-th target;
yk is the k-th output. The updates of the parameters are performed according to the gradient method

$$\text{param(iter +1)} = \text{param(iter)} - \alpha \nabla_{\text{param}} \text{E} \quad (17)$$

Where ∇ Eparam E is a gradient of the performance index E with respect to the network parameters, α is the learning rate coefficient, and iter is the iteration counter.
The nonlinear function associated with the output place is a standard sigmoid described as[14]:

$$y_k = \frac{1}{1+\exp(-\sum Z_i V_{ki})} \quad (18)$$

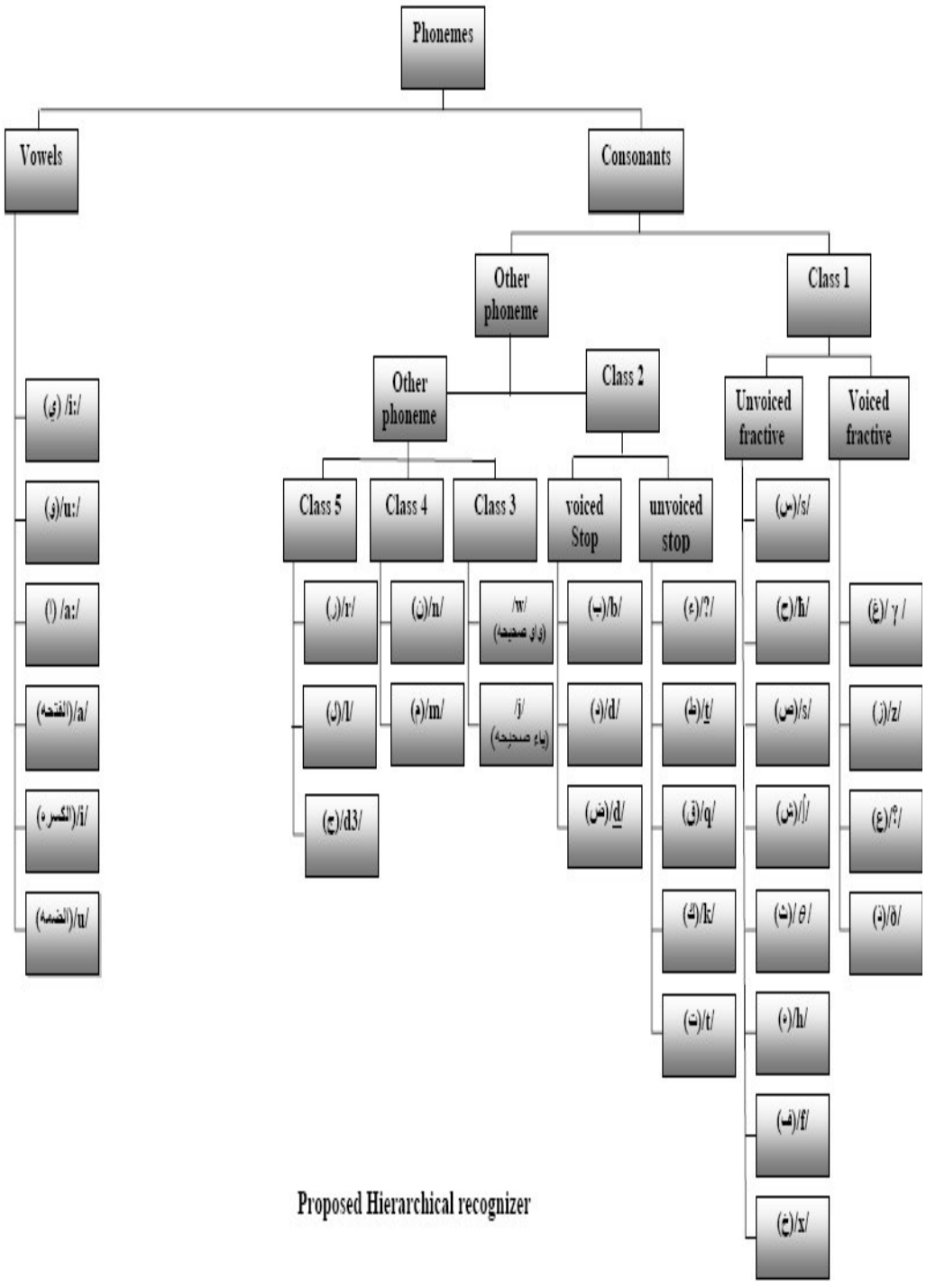
In this paper we used the fuzzy Petri net in Arabic phoneme recognition and using LPC (liner predictive code) technique as feature extracting for speech signal.

6. EXPERMENTAL RESULTS

For each phoneme, there are 24 recorded words. In eight of the 24 words, the target comes as the initial letter in word. In the second group, 8 phonemes come in the middle of the words. In other eight phonemes, the target phoneme comes at the end of the words. Manual segmentation is used to extract target phoneme from the recorded words. Half of the phonemes used in the training and the other half is used for testing the resulting system. Adobe Audition software1.5 is used to record and save the data files. The recorded data is stored as (.WAV) files with 16-bit per sample precision and a sampling rate of 16 kHz.
A hierarchical tree is formed for the phonemes as shown in figure (7). Five classes are exist in this tree. The first class is the fricative which contain both voiced and un voiced, the second class is the stop which contain both voiced and unvoiced, the third class contain the semi vowel phoneme, the fourth class contain nasal, and fifth class contain affricative, lateral, and trail. Each node in the tree is recognized using a separate NFPN with LPC parameters as inputs to these networks. The recognition principle is based on divide and contour. Class 1 is identified with a net consist of 18 input place, 56 transition and one output place. For class2 is a net consist of 18 input place, 47 transitions and one output place. For class 3 the input place is 18, the hidden layer is 22 and one output place. For class 18 the input place is 2, the hidden layer is 22 and one output place. For class 5 is a net consist of 18 input place, 36 hidden layer and one output. Table 1 shows the recognition accuracy for each phoneme and class recognition. It is found that the total recognition accuracy reached 79.6378%.

7. Conclusion

Arabic phoneme recognition system is proposed in this work. The decision is based on LPC feature vector and hierarchical NFPN as a decision network. The experimental results shows that it is possible to use the hierarchical structure to recognize phonemes using NFPN.



Proposed Hierarchical recognizer

FIGURE 7: The proposed hierarchical tree.

NO	phonemes	IPA symbols	Phoneme recognition accuracy %	class recognition accuracy %
1	ء	/ʔ/	71.66	85.83
2	ب	/b/	66.67	87.5
3	ت	/t/	81.67	100
4	ث	/θ/	61.46	81.25
5	ج	/d3/	91.67	91.67
6	ح	/ħ/	87.5	100
7	خ	/x/	56	75
8	د	/d/	69.44	83.33
9	ذ	/ð/	60.41	75
10	ر	/r/	58.8	75
11	ز	/z/	86.45	90.27
12	س	/s/	91.67	100
13	ش	/ʃ/	100	100
14	ص	/s/	86.85	100
15	ض	/d̲/	58.33	83.33
16	ط	/t̲/	56.67	75
17	ع	/ʕ/	90.97	100
18	غ	/ɣ/	91.67	91.67
19	ف	/f/	75	87.5
20	ق	/q/	71.67	87.5
21	ك	/k/	90.33	91.67
22	ل	/l/	70.83	91.67
23	م	/m/	83.33	83.33
24	ن	/n/	66.67	83.33
25	ه	/h/	89.58	91.67
26	و	/w/	83.33	100
27	ي	/j/	91.667	100
28	ا علة	/a:/	81.25	87.5
29	و علة	/u:/	100	100
30	ي علة	/i:/	91.5833	100
31	فتحة	/a/	86.667	91.667
32	ضمة	/u/	86.667	91.667
33	كسرة	/i/	91.5833	100
Total accuracy %			79.6378	90.3744

Recognition accuracy for the phonemes in the Hierarchical based LPC.

TABLE 1: The phonemes recognition accuracy

8. REFERENCES

1. S. Ismail, and A. Ahmad, "*Recurrent neural network with back propagation through time algorithm for Arabic recognition*," In Proceedings of the 18th ESM Magdeburg, Germany, 13-16 June 2004.

2. S. Al-Sayegh, and A. AbedEl-Kader, "*Arabic phoneme recognizer based on neural network"* , In Proceedings of International Conf. Intelligent Knowledge Systems (IKS-2004), August 16-20,2004.

3. Y. Alotaibi, S. Selouani, and D. O'Shaughnessy, "*Experiments on Automatic Recognition of Nonnative Arabic Speech*", EURASIP Journal on Audio, Speech, and Music Processing, Vol. 2008, pp.1-6, 2008.

4. M. Awais, and Habib-ur-Rehman, "*Recognition of Arabic phonemes using fuzzy rule base system"*, In Proceedings of 7th Int. Multi Topic Conf. INMIC-2003, pp.367-370, 8-9 Dec. 2003.

5. P. Schwarz, P. Matejka, and J. Cernocky, "*Hierarchical Structures of Neural Networks for Phoneme Recognition*", In Proceedings of IEEE Int. Conf. Acoustics, Speech and Signal Processing, ICSP-2006, 14-19 May 2006.

6. J. Pinto and H. Hermansky, "*Combining Evidence from a Generative and a Discriminative Model in Phoneme Recognition*", In Proceedings of Interspeech. Brisbane, Australia 22-26 September 2008

7. M. Scholz, and R. Vigario, " *Nonlinear PCA: a new hierarchical approach*", In Proceedings of European Symposium on Artificial Neural Networks ESANN-2002, pp. 439-444, Bruges, Belgium, 24-26 April 2002.

8. Y. Suh and Y. Lee, "*Phoneme Segmentation of Continuous Speech using Multi-Layer perceptron*", In Proceedings of 4th Int. Conf. Spoken Language, ICSLP-96,3, pp.1297-1300 , 1996.

9. Y.Gong, "Speech Recognition in Noisy Environments: A SurveSpeech Communication ,16, p:261-291, 1995.

10. N. Awasthy, J.P.Saini and D.S.Chauhan "*Spectral Analysis of Speech: A New Technique", Int*. J. Signal Processing ,2(1), p: 19-28, 2005.

11. L.Rabinar and R.W.Schafar "*Fundamental of Speech Recognition* ",Prentice Hall, 1993.

12. S. I. Ahson "*Petri net models of fuzzy neural networks*," IEEE Trans. Syst. Man Cybern., 25(6), pp. 926–932, Jun. 1995.

13. A. Seely, *"Petri Net Implementation of Neural Network Elements*", M.Sc. Thesis, Nova Southeastern University, 2002.

14. H. M. Abdul-Ridha," *ECG Signal Classification using Neural, Neural Fuzzy and Neural Fuzzy Petri Networks*" Ph.D. Thesis,Department of Electrical Engineering, University of Basrah, 2007.