7. TEMPERATURE CONTROLLED PSO PRE-TRAINED DBN FOR PHONEME RECOGNITION

The complexity and invariability lying in the speech recognition problem provoked the thought of using a generative model which is capable of handling complex problems. In the previous work discussed in Chapter 6, the process of pre-training DBN using contrastive divergence has been replaced by the variants of PSO to handle the problem of the solution being trapped in local minima and to speed up the process of DBN training. The PSO-DBN and SGPSO-DBN models discussed in the previous chapter are able to identify the better optimal solution in addition to the advantage of lowering the time complexity of pretraining phase. The variants of PSO, as expected reduced the training time of DBN, also turning up with better optimized connection weights and bias parameters for DBN. This was achieved with the property of PSO to quickly find and explore the promising regions in the global search space. Among the variants of PSO experimented, PSO was found as a worthy option with better phoneme error rates. But, due to the lack of momentum, PSO risk in tending to stagnate at some point preventing it from reaching the optimal solution.

This chapter proposes a novel optimization procedure, through an improved version of PSO to support faster and promote active convergence of particles towards the solution. This is achieved through a well known characteristic of particle velocity with respect to the temperature of the molecule. The velocity of the particle increases with temperature. With this intuition, a new optimization algorithm called temperature controlled PSO is proposed to gear up the movement of particles in the optimization phase and to initialize the parameters of DBN. The proposed methodology called TPSO-DBN is used to build phoneme recognition model for continuous Tamil speech.

7.1 TAMIL PHONEME RECOGNITION MODEL USING TPSO-DBN

Parameter optimization is a crucial task involved in improving the efficiency of any model, which holds true even for deep neural networks. The proposed methodology uses the proposed TPSO parameter optimization algorithm to identify the optimized parameters for the deep belief network. The complete model building process is depicted in Fig. 7.1. The dataset is passed to TPSO-DBN to get the model trained. The pre-processed labelled phoneme dataset built from the continuous Tamil speech using a graphcut based segmentation algorithm is used to build the phoneme recognition model. The phonetic dataset is passed to the TPSO-DBN module which comprises of two phases in building the model namely, (i) DBN parameter optimization phase

using TPSO - the pretraining phase and (ii) the training and building DBN phase. The proposed TPSO is introduced before discussing the pretraining and training phases of the architecture.

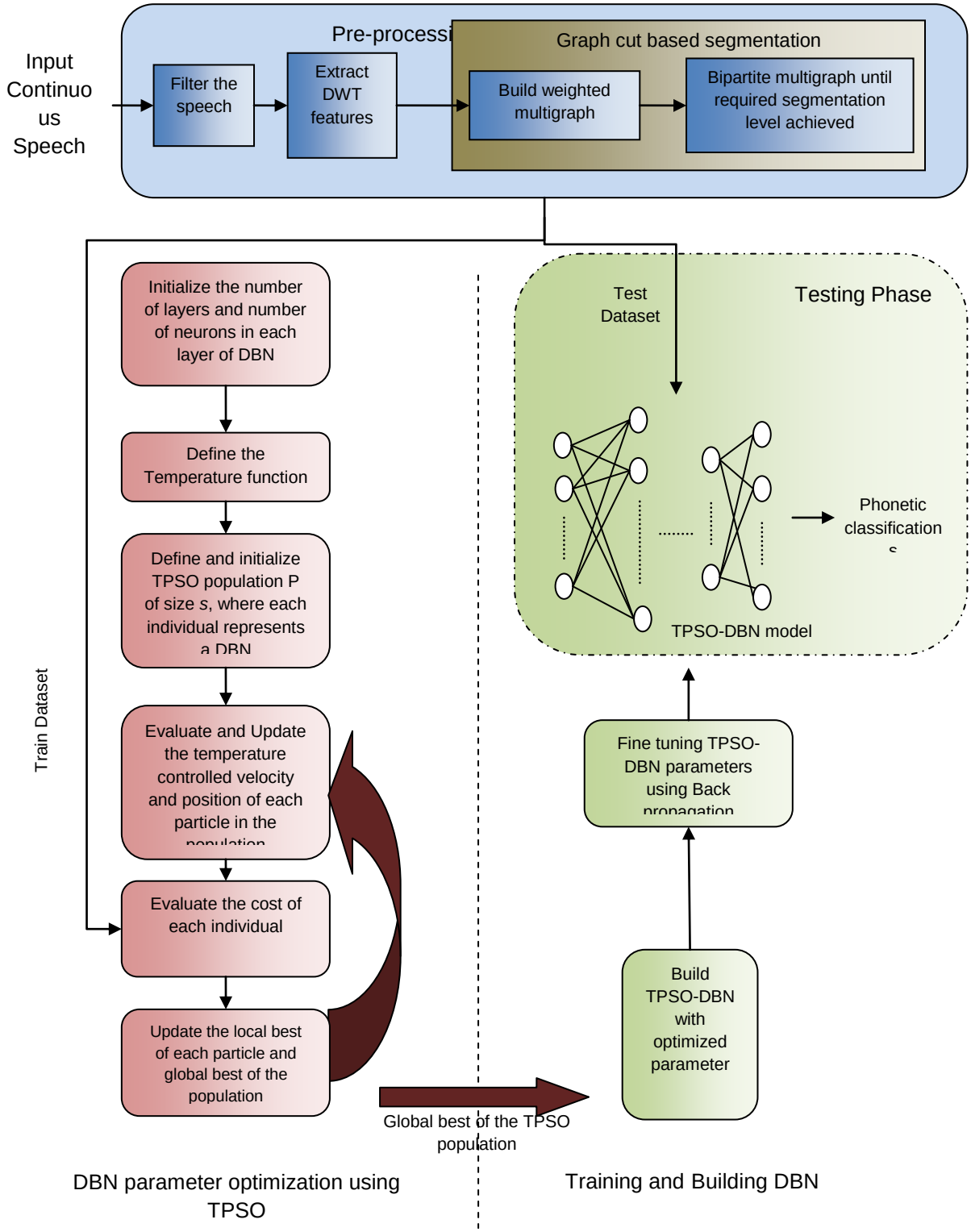

Fig. 7.1 Architecture of Proposed TPSO-DBN Phoneme Recognition Model

Temperature controlled particle swarm optimization algorithm is proposed with the intuition of the relationship existing between the temperature of the particles and the acceleration of particles. The velocity of the particles is directly proportional to the temperature of the molecules. The velocity of the particle in a population during PSO training is defined with two terms namely, local best and global best term. In addition to those terms, the TPSO includes a temperature term to control the new velocity of the particle in each of the iteration. The velocity updation equation of TPSO that replaces the velocity equation 2.22 of basic PSO is given as follows,

$$v_i(t+1)=wv_i(t)+h(t)v_i(t)+c_1r_1(p_i(t)-x_i(t))+c_2r_2(p_g(t)-x_i(t) \quad (7.1)$$

where *h(t)* is temperature function whose values are in the range [0,1]. The second term in the equation 7.1 is called the temperature term. The temperature function is a trapezoidal function as shown in Fig. 7.2. The temperature function *h(t)* gradually increases with *t* for certain generations and stays stable for generations, which then declines during the last few generations. This change in the temperature controls the increase and decrease of velocity over generations. Once the velocity of the particles is updated in each generation using equation 7.1, the position update of the particles in the population is done using equation 7.2.

$$\boldsymbol{x_i(t+1) = x_i(t) + v_i(t+1)} \quad (7.2)$$

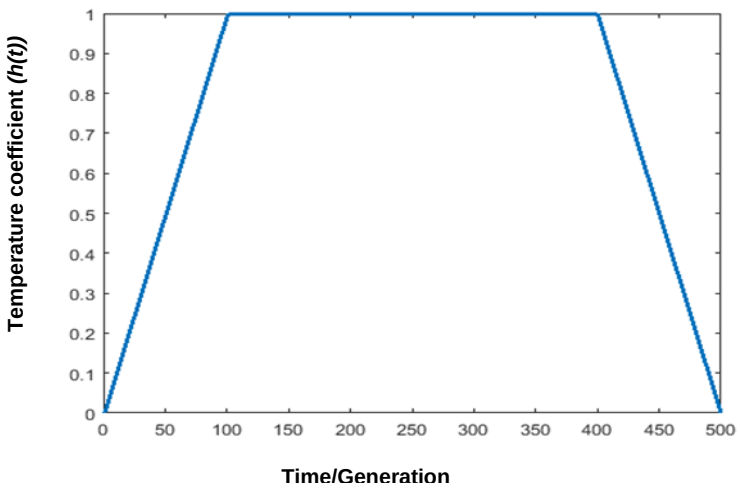

Fig. 7.2 Trapezoidal Function

In the pretraining phase the dataset first enters the TPSO parameter optimization process where the optimized parameters are identified for the DBN being built. The algorithm 7.1

elucidates the steps to optimize the DBN parameters using TPSO. The algorithm takes the DBN architectural parameters such as number of layers, number of neurons activation function and dataset to produce the evolved population and global optimal solution for the given problem.

Algorithm 7.1 Optimizing DBN parameters using TPSO

Step 1: Initialize the TPSO parameters namely, inertia coefficient ω, personal acceleration coefficient c_1 and global acceleration coefficient c_2

Step 2: Define the temperature function, h

Step 3: Define the lower bounds Ímin and upper bounds Ímax of the decision variables

Step 4: Instantiate population P of size M whose individuals are of the form as shown in Fig. 7.3

Step 5: for all individuals i in the population P do

a. Initialize the position vector $x_i(1)$ as uniform random values in the range $(X_{min};X_{max})$

b. Initialize the velocity vector $v_i(1)$ as zero

Step 6: for each generation *t* do

Step 7: for each individual *i* in the population *P* do

a. Calculate the new velocity,

$$v_i(t+1)=wv_i(t)+h(t)v_i(t)+c_1r_1(p_i(t)-x_i(t))+c_2r_2(p_g(t)-x_i(t)$$

b. Calculate the new position, $x_i(t+1) = x_i(t) + v_i(t+1)$

c. Evaluate the cost of the individual for the given training dataset using algorithm 7.2

d. Update the personal best cost and position pi

e. Update the global best cost and position pg

f. Update inertia ω

In this TPSO parameter optimization phase, population of particles are defined randomly for the initial generation. Based on the architectural parameters given, the individuals of the TPSO population are formulated as shown in Fig. 6.2. Each particle in the population represents a DBN, thus defining its connection weight and bias parameters, referred as the position of a

particle. These parameters are initialized with uniform random values of range ($Ɪ_{min}$, $Ɪ_{max}$), where $Ɪ_{min}$ and $Ɪ_{max}$ are lower and upper bound vectors for decision parameters. The velocity of each particle is initialized as a zero vector. Next, the temperature function is defined, which is a trapezoidal function that increases gradually, then stays stable and then declines finally to denote the temperature of the particles in the population over time. The TPSO phase runs for a fixed number of generations. During each generation, new velocity of each particle is evaluated using equation 7.1, which is followed by the position update of each particle in the population using equation 7.2. At the end of each generation the cost of every individual in the population representing the DBNs are evaluated with the given training dataset and the personal best cost of the each particle and the global best cost of the population are updated as defined in algorithm 7.2.

Algorithm 7.2 Evaluating the cost of each individual in TPSO

Step 1: for each individual *i* in population do

Step 2: Construct a DBN $₦_i$ by transforming individual *i* coded as in Fig. 6.2 into a DBN structure

Step 3: Pass the train dataset through the layers of $₦_i$ using $p(h_j = 1|v, \theta) = \sigma(a_j + \sum_{i=1}^{V} w_{ij} v_i)$

Step 4: Evaluate $Cost(₦_i) = (y_i - o_i)^2/ɱ$, where y_i, o_i and $ɱ$ are predicted output vector, actual output vector and number of training samples.

Step 5: end for

TPSO is used here to evolve the connection weights of the DBN. Each generation evolves by updating the population with new velocity and position of the individuals under consideration. The candidate models represented by the population are evaluated for each generation with the training data to find the global best and get ready for the evolution of next generation.

Once the pre-training of the DBN is successfully completed by applying TPSO, the global best of the TPSO is considered as an optimized solution. The values obtained as the position parameters of the global best mapped to initialize the bias and connection weights of DBN and then subjected to back-propagation training process. The steps to initialize and train the DBN are portrayed in algorithm 7.3.

Algorithm 7.3 Building TPSO-DBN with global best of TPSO

Step 1: Construct TPSO-DBN ₦ with the best individual obtained in TPSO parameter optimization phase

Step 2: for i = 1 to MaxIteration do

a. Pass the train dataset through the layers of DBN ₦ using equation 2.22

b. Backpropagate the error from the output layer to all the hidden layers by updating their bias and weights using change in bias and weight obtained through equation 5.2 to 5.6

Step 3: end for

The output obtained in TPSO explained in algorithm 7.1 is given as input to the above algorithm, where the values of the global best individual is mapped to the bias and connection weights of DBN with reference to the coding scheme used. The training dataset is then passed to make the DBN learn with the forward phase and backward phase that uses backpropagation training. This process is repeated for MaxIteration to build the phoneme recognition model.

7.2 EXPERIMENT AND RESULTS

The DBN based model for phonetic recognition for Tamil continuous speech is built and experimented using TPSO based pretraining. The DWTFS dataset of speech corpus 'Kazhangiyam' discussed in chapter 3 is used in the experiment which is a labelled dataset of Tamil phonetic units. The experiments discussed here builds 7-layer DBNs. The performance of the different DBNs that are either pretrained using contrastive divergence or pre-trained by one of the variations of PSO namely PSO, SGPSO, NMPSO or TPSO are compared.

The hyperparameters of the DBN model to be built are decided. The experiments are done by building 7-layer DBNs with 90, 100, 120, 120, 100, 70 and 39 neurons in each layers respectively using various pretraining methods. The sigmoid function is used as the activation function. The hyperparameters defined for DBN are passed to the TPSO module along with the training dataset which is contributed with 70% of the DWTFS dataset. The TPSO parameters namely, maximum number of iterations, population size, inertia coefficient ω, damping inertia coefficient, personal acceleration co-efficient c_1, global acceleration co-efficient c_2 are set as 500, 35, 1, 0.99, 2 and 2 respectively. The temperature function *h(t)*, used in TPSO is defined as

shown in Fig. 7.2. Mean Square Error (MSE) is set as the cost function in the pre-training phase of TPSO-DBN model building.

With the population set to 35, each individual is randomly initialized adhering to the defined limits, each representing a solution to initialize the parameters of the DBN model. The movement of particles towards the global optimal solution during TPSO pretraining, captured at every hundredth generation for DWTFS training dataset is shown in the Fig. 7.3. The evolution of particles in TPSO continues for 500 iterations. The movement of the particles are highly realised during this pretraining.

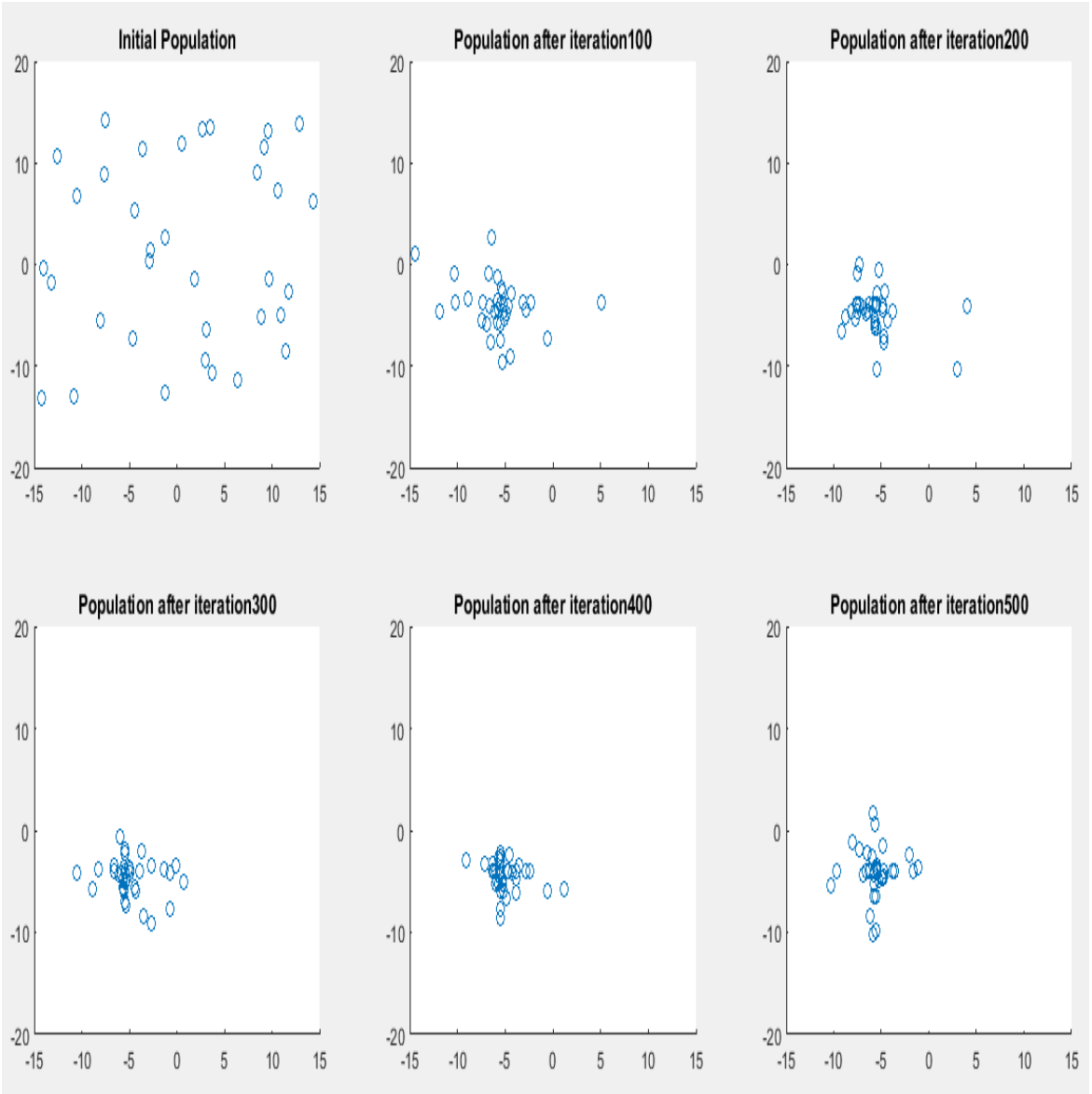

Fig. 7.3 Population of Initial, 100th, 200th, 300th, 400th and 500th TPSO's Generation

The best cost observed over generations during TPSO pretraining is compared with the best costs observed for various other pretraining procedures in Fig. 7.4. The best cost achieved in this experiment is recorded as 0.0297 for TPSO pretraining and is found to be lower when compared to best cost for various other pre-training procedures recorded as 0.0465, 0.0442 and 0.0449 for PSO, SGPSO and NMPSO respectively. The best costs observed in various pretraining procedures have converged in generations 220, 453, 129 and 290 respectively. The experiments SGPSO and NMPSO converge to the solution space earlier when compared to PSO and TPSO. The steepness of best cost curve shows that the variations of PSO namely SGPSO and NMPSO converge faster towards the optimum solution than PSO and TPSO for the problem under consideration. Among the variants of PSO considered, the best costs achieved through generations is observed less stagnate for TPSO when compared to its competing variants.

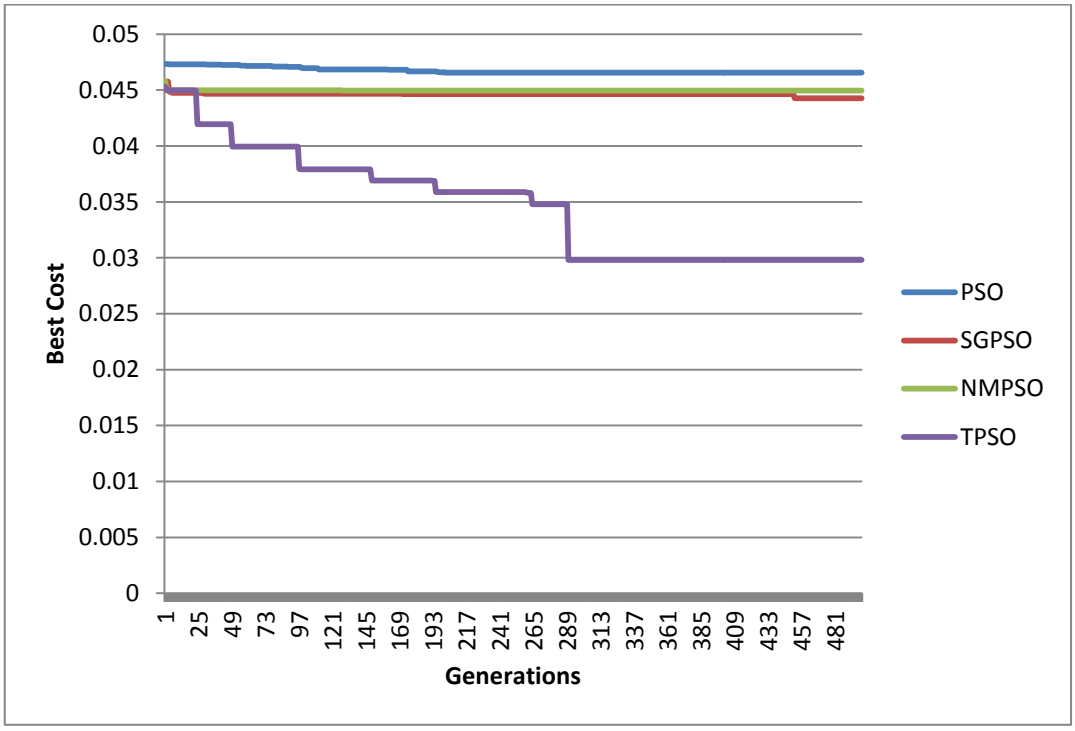

Fig. 7.4 Best Cost Comparison of PSO, SGPSO, NMPSO and TPSO Pre-training

The global best solution arrived in TPSO is then decoded to represent the DBN. A 7-layer DBN with 90, 100, 120, 120, 100, 70 and 39 neurons from layer 1 to 7 is built with sigmoid activation function defined for each neuron. The DBN modelled so, is then trained using backpropagation. For all DBN models, the back propagation based training lasts for 1000 iterations with a batch size of 100. During DBN training, the initial and final momentum is set to 0.5 and 0.9 respectively, where the initial momentum lasts for first five iterations. The weight cost of the DBN is set as 0.0002. The backpropagation training is accomplished as discussed in chapter 5.

The TPSO-DBN model is evaluated for its effectiveness using measures such as root mean square error, phoneme error rate, precision, recall and F1-score. Eventhough the pretraining phase helps to reach the optimal solution, there is a need to fine tune the network parameters through backpropagation to improve the accuracy rate of the model built. Table XXIV projects the RMSE and PER values for various models under consideration before fine tuning with back propagation algorithm for the training and testing datasets. The results show that the RMSE and PER values for TPSO is observed as 0.01031, 0.01272, 13.91% and 16.72% for training and testing datasets respectively, which is lower than other models proving the efficiency of TPSO in parameter optimization and the effect of using the well optimized parameters in building DBNs.

Table XXIV Performance Comparison of Various Models without Back Propagation Fine tuning

Method	Training Data		Testing Data		Average PER
	RMSE	PER	RMSE	PER	
CD-DBN	0.04526	34.21	0.04918	36.23	35.22
PSO-DBN	0.03209	21.31	0.03802	25.01	23.16
SGPSO-DBN	0.02941	18.68	0.03901	28.11	23.39
NMPSO-DBN	0.04654	28.37	0.04932	32.69	30.53
TPSO-DBN	0.01031	13.91	0.01272	16.72	15.32

Further the RMSE and PER values experimented for various models with backpropagation fine tuning are portrayed in Table XXV, for both training and testing phases. The results show an improvement in the RMSE and PER values for various models proving the support of back propagation in performance. Among the results obtained, the best phoneme error rate of 9.5% and 10.81% are observed for SGPSO-DBN model in training phase and TPSO-DBN model in testing phase respectively. The RMSE in both training and testing phases are observed as 0.0094 and 0.01058 respectively and found to be lower in TPSO-DBN model when compared to the other models. Table XXIV shows that the TPSO-DBN has also recorded the lowest average PER with 10.65% when compared to other methods with 14.79%, 12.59%, 15.76% and 15.45%. The comparative results of RMSE and PER observed during training and testing phases of various models is shown in Fig. 7.5 and 7.6.

Table XXV Performance Comparison of Various Models Built with Back Propagation Fine tuning

Method	Training Phase		Testing Phase		Average PER
	RMSE	PER	RMSE	PER	
CD-DBN	0.01523	14.62	0.01528	14.97	14.79
PSO-DBN	0.01549	12.2	0.01620	12.98	12.59
SGPSO-DBN	0.01196	9.5	0.02112	22.03	15.76
NMPSO-DBN	0.01891	13.03	0.0210	17.87	15.45
TPSO-DBN	0.0094	10.5	0.0105	10.81	10.65

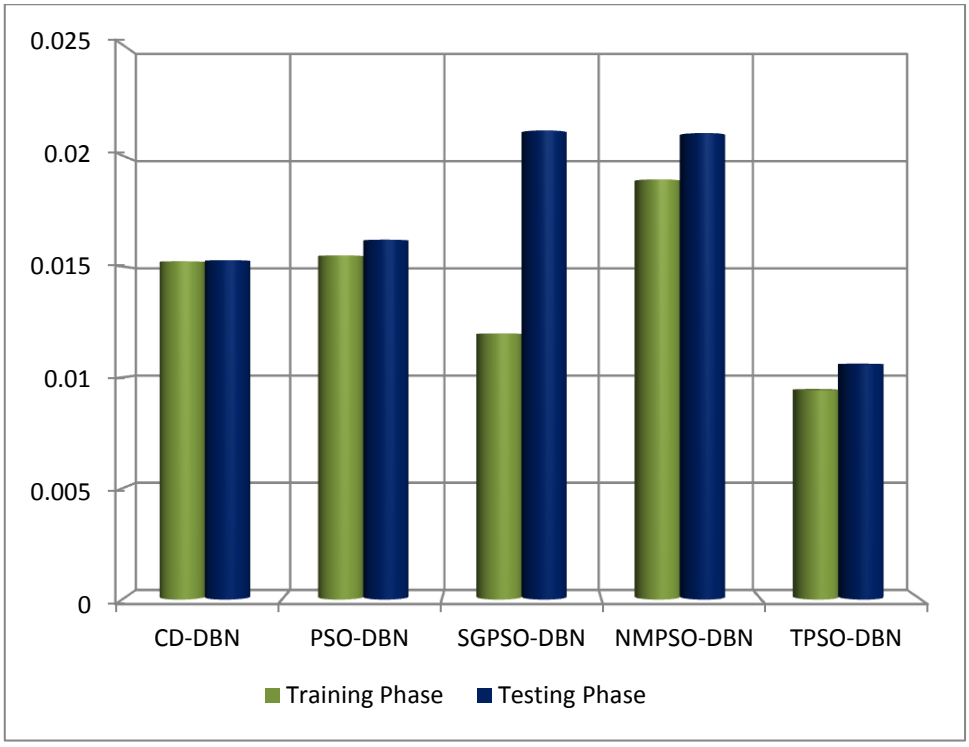

Fig. 7.5 Performance Comparison of TPSO-DBN Model with Other Models using RMSE

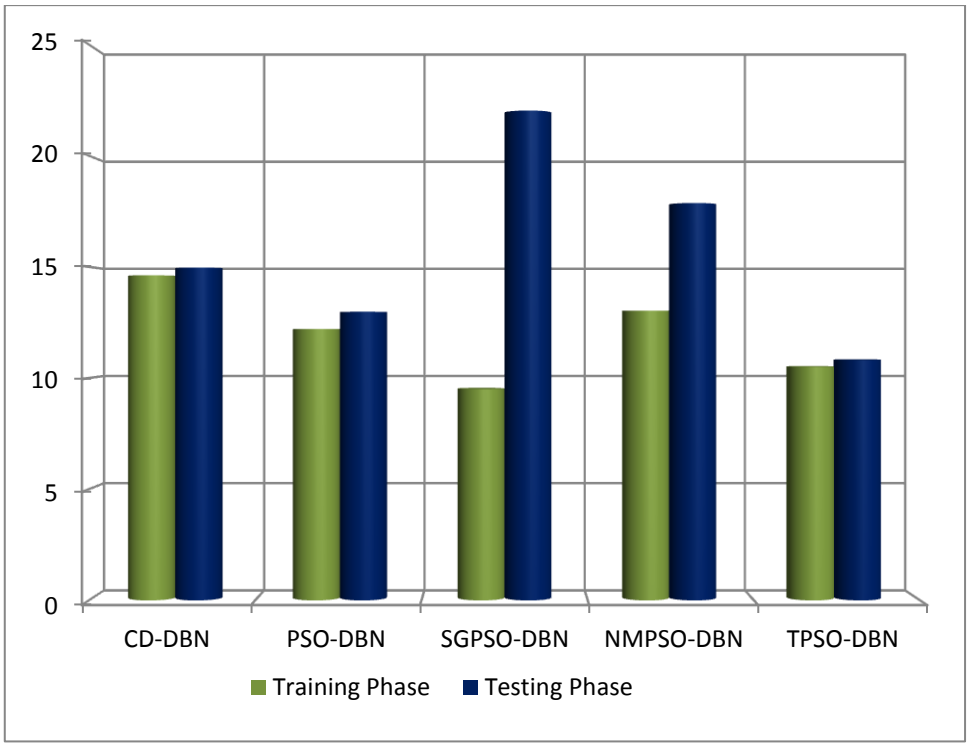

Fig. 7.6 Performance Comparison of TPSO-DBN Model with Other Models using PER

Comparison of average PER before and after backpropagation training in Fig. 7.7 shows a decline in average PER for the five models under experimentation. Their PER declination after backpropagation fine tuning are observed as 20.43%, 10.57%, 7.63%, 15.08% and 4.67% for CD-DBN, PSO-DBN, SGPSO-DBN and TPSO-DBN respectively. It can be seen that the average PER of TPSO-DBN declines much lesser when compared to the other models showing ability of TPSO in identifying the better optimal solution for DBN.

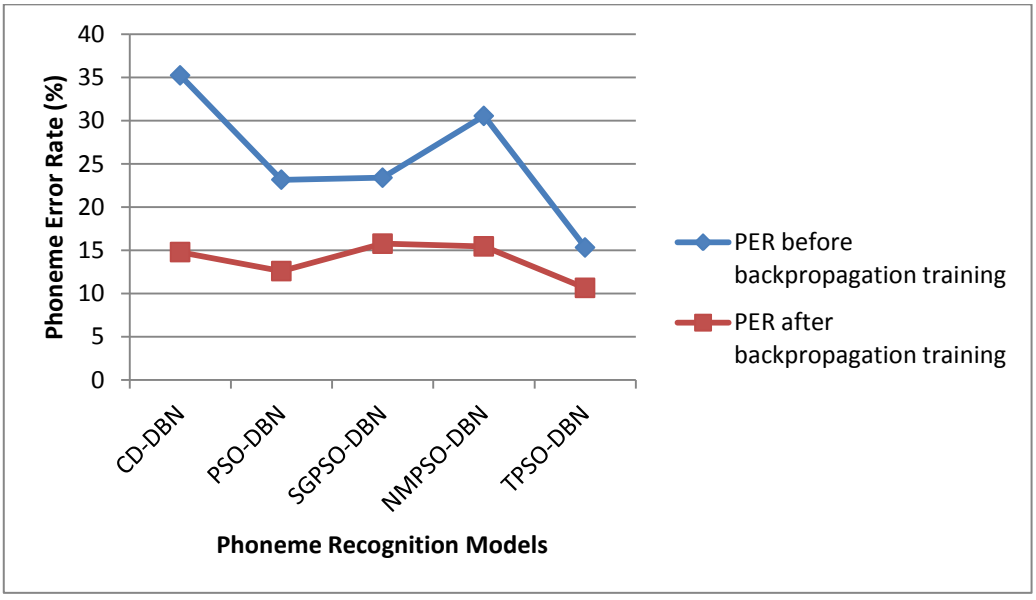

Fig. 7.7 PER Comparison for Various DBN Before and After Backpropagation Training

The precision, recall and F-measure of the models under study are listed in the Table XXVI. The best precision achieved is 0.807888, 0.769021 and 0.687681 for TPSO-DBN, SGPSO-DBN and PSO-DBN. The best recall is observed as 0.778787, 0.773511 and 0.611481 for TPSO-DBN, SGPSO-DBN and PSO-DBN respectively. The recognition performance of TPSO-DBN performs comparatively competitive with SGPSO-DBN and is observed as 0.790325. The accuracy achieved for TPSO-DBN acoustic model is 89.35% and is found to outperform CD-DBN, PSO-DBN, SGPSO-DBN and NMPSO-DBN which have achieved accuracy of 85.21%, 87.41%, 86.62% and 47.33% respectively.

Table XXVI Performance Comparison of TPSO-DBN with CD-DBN, PSO-DBN, SGPSO-DBN, NMPSO-DBN Models

Method	Precision	Recall	F-Measure	Accuracy (%)
CD-DBN	0.299081	0.253322	0.274306	85.21
PSO-DBN	0.687681	0.611481	0.647346	87.41
SGPSO-DBN	0.769021	0.778787	0.773872	86.62
NMPSO-DBN	0.198742	0.239211	0.217105	47.33
TPSO-DBN	0.807888	0.773511	0.790325	89.35

The comparison of precision, recall, F-measure and accuracy various models with TPSO-DBN phoneme recognition model is shown in Fig. 7.8. It is found that TPSO-DBN transcends the other models under comparison.

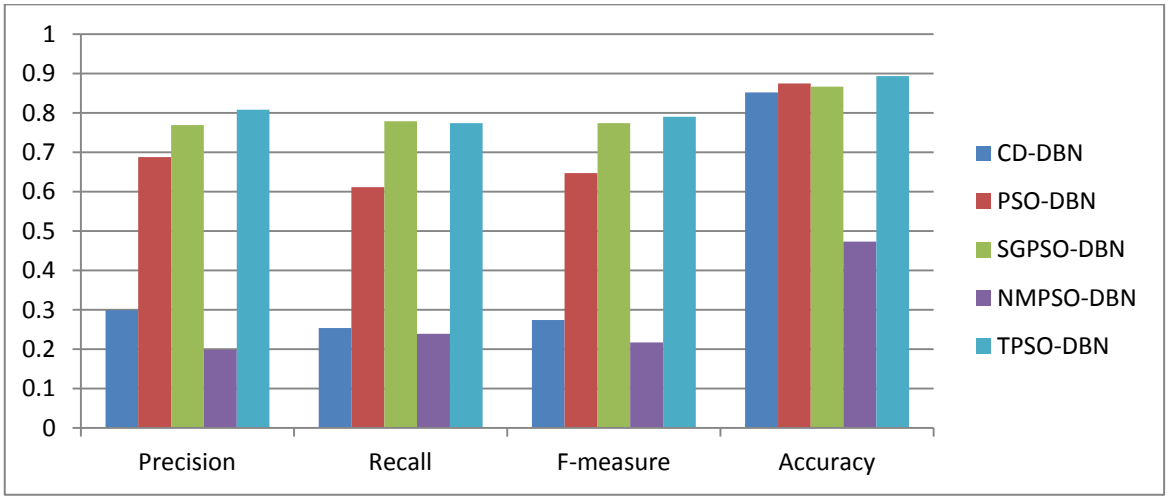

Fig. 7.8 Performance Comparison of TPSO-DBN with DBN, PSO-DBN, SGPSO-DBN and NMPSO-DBN Models

The recognition performances of the models are further analyzed using receiver-operating characteristic curve and area under curve measures. The performances of the models under consideration are depicted as ROC curves in the Fig. 7.9. It shows ROC curves of some majority classes namely, /a/, /u/, /nn/, /th/, /p/ and /m/ for the classifiers built using CD-DBN, PSO-DBN, SGPSO-DBN, NMPSO-DBN and TPSO-DBN.

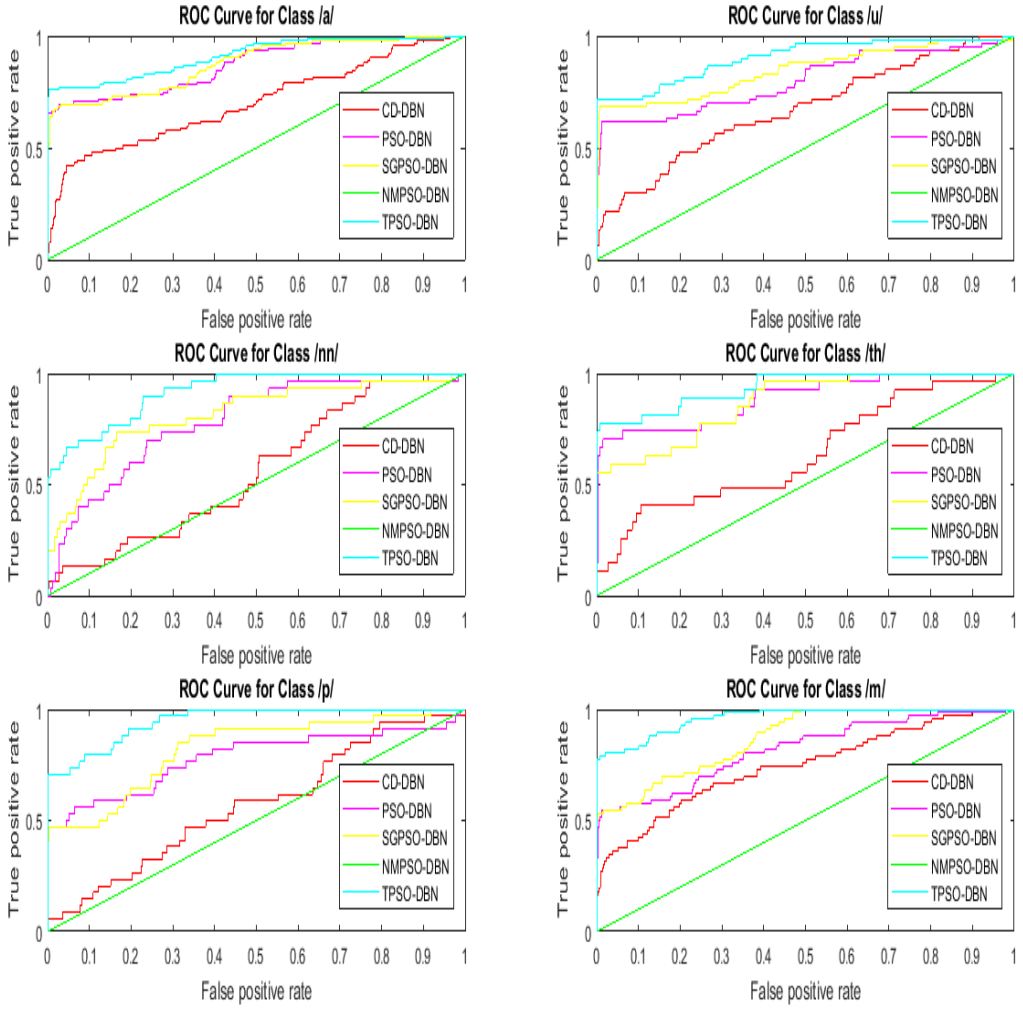

Fig. 7.9 ROC of Few Majority Classes /a/,/u/,/nn/,/th/,/p/ and /m/ for Models Built using CD-DBN, PSO-DBN, SGPSO-DBN, NMPSO-DBN and TPSO-DBN Respectively

The ROC curves depicting the performance of the models built for minority classes namely, /uu/, /au/, /sh/, /nj/, /zh/ and /j/ is projected in Fig. 7.10. It is observed that for most cases the classification performance in terms of true positive rate to the false positive rate is not as promising as observed for the majority classes as shown in Fig. 7.9.

s

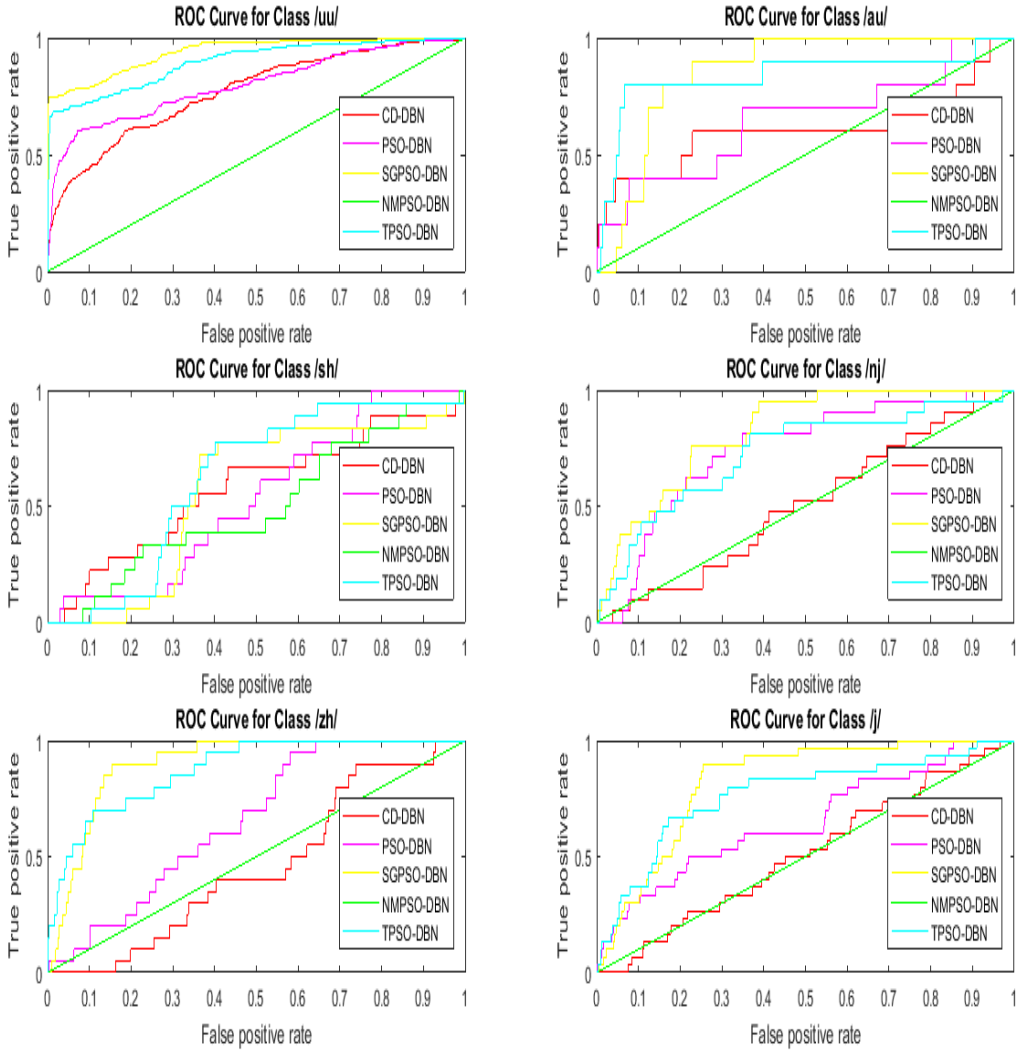

Fig. 7.10 ROC of Few Minority Classes /uu/,/au/,/sh/,/nj/,/zh/ and /j/ for Models Built using CD-DBN, PSO-DBN, SGPSO-DBN, NMPSO-DBN and TPSO-DBN Respectively

The ROC curve of TPSO-DBN and SGPSO-DBN are more towards 1 confirming them to be better classifiers when compared to the other models under study. The AUC values of the various classifiers for the classes visualized as ROCs in Fig. 7.9 and Fig 7.10 are listed in Table XXVII. The AUC measure is identified to be greater than 0.9 for TPSO-DBN and greater than 0.8 for SGPSO-DBN for all majority classes. The highest AUC is observed as 0.9618 for class /m/ of TPSO-DBN and the lowest AUC is observed as 0.4583 for CD-DBN class /zh/. The AUC of maximum classes for NMPSO-DBN is observed to be constant showing the model reacts same for both majority and minority classes of the dataset.

Table XXVII AUC Comparison of Few Classes for Various Models

Class/Model	CD-DBN	PSO-DBN	SGPSO-DBN	NMPSO-DBN	TPSO-DBN
/a/	0.7016	0.8724	0.8755	0.5	0.9117
/u/	0.6723	0.7997	0.8466	0.5	0.9041
/nn/	0.5609	0.7840	0.8164	0.5	0.9208
/th/	0.6384	0.8875	0.8711	0.5	0.9392
/p/	0.5692	0.7818	0.8221	0.5	0.9484
/m/	0.7417	0.8170	0.8713	0.5	0.9618
/uu/	0.7702	0.800	0.9407	0.5000	0.8973
/au/	0.5967	0.6502	0.8587	0.5000	0.8389
/sh/	0.5736	0.5312	0.5598	0.4992	0.8205
/nj/	0.5041	0.7424	0.8244	0.5000	0.7285
/zh/	0.4583	0.6577	0.9026	0.5000	0.8776
/j/	0.5089	0.6526	0.8236	0.5000	0.7673

In consolidation, the best cost and convergence time achieved by SGPSO is better than the other pre-training methods used but its RMSE and PER values shows greater differences for training and testing datasets which shows the overfitness of the model. On the other hand, the recognition performance of DBN pre-trained using TPSO gives better precision, recall, F-measure and recognition accuracy when compared to the one pre-trained with SGPSO. The ROC curves and AUC values of classes for various models proves the improved efficiency of proposed TPSO-DBN model to other models. The performance of the proposed TPSO-DBN is evaluated to be the best of various aforesaid experiments conducted on DWTFS speech dataset.

Findings

Eventhough the performance of DBNs pretrained using PSO and SGPSO are performing comparatively during training to TPSO-DBN, they are observed to be overfitted models during the testing process. The statement is strongly supported with the respective AUC values and corresponding ROCs. The difference in the AUC values is observed and identified to be lesser for minority classes when compared to majority classes for all models. It is proved that TPSO solved the problem of particle stagnation in the pretraining phase which is realised through the

steady improvement in the best cost curve over generations. Thus it helps more in the movement of the particles in the population towards the global optimal solution providing better initialization to the DBN which intern improve the performance of the model built.

SUMMARY

The proposed TPSO algorithm and the Tamil phoneme recognition model built using TPSO pretrained DBN was elucidated in this chapter. The performance of TPSO-DBN was evaluated and compared with the previously implemented models namely, CD-DBN, PSO-DBN, SGPSO-DBN and NMPSO-DBN. As problem under consideration uses a Tamil phoneme dataset which is highly imbalanced, the influence PER contributed by the minority class was evinced. So, it is believed that the accuracy of the model can be further improved by using techniques to handle the problem of imbalanced data and is proposed in forth coming chapter.

Remarks

- The article titled ‘Temperature Controlled PSO on Optimizing the DBN Parameters for Phoneme Classification’ is published in International Journal of Speech Technology, Springer, Vol. 22, No. 1, pp. 143-156, 2019, ISSN online: 1572-8110, ISSN print: 1381-2416. Springer US. DOI: 10.1007/s10772-018-09586-2 (Published-Scopus Indexed)