Speaker Dependent Hindi Speech Recognition using Optimized Classifiers

A Thesis

Submitted in fulfillment of the

requirements for the award of the degree of

Doctor of Philosophy

Submitted By

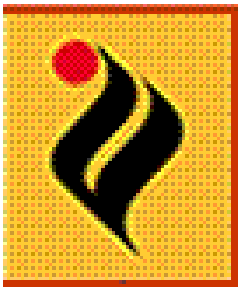
Teena Mittal

(Registration No. 950906017)

Under the supervision of

Dr. R.K. Sharma

Professor, CSED,
Thapar University, Patiala- 147004
Punjab, India.



Electronics & Communication Engineering Department
Thapar University
Patiala-147004

February 2016

To

My Family

CERTIFICATE

I, **Teena Mittal**, hereby certify that the work which is being presented in this thesis entitled **"Speaker Dependent Hindi Speech Recognition Using Optimized Classifiers"**, in fulfillment of requirements for the award of degree of the **DOCTOR OF PHILOSOPHY** submitted in the Electronics & Communication Engineering Department (ECED), Thapar University, Patiala, is an authentic record of my own work carried under the supervision of **Dr. R. K. Sharma** (Professor, CSED, Thapar University, Patiala).

The matter presented in this thesis has not been submitted either in part or full to any other University or Institute for the award of any degree.

(Teena Mittal)

Date: February 2, 2016

It is certified that the above statement made by the candidate is correct to the best of my knowledge and belief.

2.2.16

(Dr. R. K. Sharma)
Professor,
Computer Science and Engineering Department,
Thapar University, Patiala
PIN –147004 (INDIA).
Supervisor

Date: February 2, 2016

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and indebtedness to Professor R. K. Sharma, who has been the main driving force behind initialization and continuation of this research study and ultimately leading to its successful completion. Words are insufficient to acknowledge the help he has rendered in the form of his expert and valuable technical guidance, sparing of valuable time, untiring efforts and above all his timely help as and when required, which made the difficult terrain of research work inspiring, motivating and easier to pass through.

It is my duty to be thankful to the authorities of Thapar University, Patiala, Punjab, India for granting me permission to carry out the research work.

I express my special thanks and hearty feelings to my husbandDr. Nitin Narang who has been the main inspiring force to initialize this study and without whose moral support and co-operation this work would not have been completed.I feel a sense of apology to my son, Chinmaya to whom I could not provide his due share of love, attention and proper care during the period of research work.

I feel highly obliged to Mrs. Anupma Sharma w/o Dr. R. K Sharma for her being so kind, co-operative and generous during the whole period of research work.

I would also like to give thanks to Ms. Manju Gulati and Ms. Ramneet Kaur for the co-operation rendered by them in different ways.

I am indebted to my parents and in-laws for all the pain and sufferings, through which they have undergone to bring me up to this stage.

Finally, I am thankful to ‘Almighty’ who has showered his blessings in the form of providing me with such helpful and kind persons.

(Teena Mittal)

ABSTRACT

Speech is the most natural way of human communication. The variability in speech signal makes automatic speech recognition (ASR) a challenging task. The variability in speech depends on environmental conditions, speaker attributes such as emotion, age, gender and many other factors. Speech recognition is one of the most promising fields of current research due to its versatile applications. Many international organizations as well as research groups are working in this field. The performance of ASR systems has been improved since last decade and now it has been used for many practical applications.Even now, there are lot of possibilities to improve their recognition rate, speed, vocabulary and usefulness for the end-users.Another issue with ASR system is that it has not been developed for a good number of languages due to limited data availability and proper statistical framework of acoustic and language models. Hindi is the national language of India and people in several other Asian countries can easily understand and speak it. So there is a need to develop an efficient ASR system for Hindi language.

The main objective of ASR is to build a system that can map the acoustic signal into a string of words. An ASR system has two main elements,*i.e.*, front-end processor and back-end classifier. The front-end processor is used to extract speech features or parameters. These features are processed by a back-end classifier, for speech recognition.

The Artificial neural network (ANN), support vector machine (SVM) and hidden Markov model (HMM) classifiers have been widely used for speech recognition. An ANN is inspired by biological neural network and it processes input information using an interconnected group of artificial neurons and a connectionist approach to computation. Training of an ANN is a tedious task, because search space is high dimensional and multimodal. An ANN training needs efficient optimization techniques to search a set of weights and biases that minimizes the error. The most commonly used training algorithm is the back-propagation algorithm. It is based on gradient search, and may get trapped in local

optimum solution for non-linearly separable pattern classification problems. Another promising classifier is support vector machine. It works on the principle of structural risk minimization. SVM has generated a lot of interest in the pattern recognition community in recent years; still optimum parameter selection of SVM kernel is a vital issue for it.

In spite of universally acceptance of HMM to recognize speech, one of the main concerns with HMM is related to training phase. The training of HMM is computationally expensive and solution usually stagnated at local optimal solution. The Baum-Welch algorithm is widely used algorithm to train HMM, but it is conventional optimization method and quality of solution highly depends on initial search point. Normally the solution obtained from BW algorithm may converge to local optimum solution.

In this thesis work, various aspects of speech recognition system have been explored and some techniques have been proposed to improve speech recognition rate. The main contribution of this research work is as follows:

(i) Two databases, namely, Hindi speech words database and Hindi sentences database have been prepared in this work. The Hindi words database consists of twenty words with fifty utterances of each word spoken by two male and two female speakers. The Hindi sentences database consists of sixteen sentences with four utterances of each sentence spoken by two male and two female speakers. The recording has here been done in a quiet room environment with sampling frequency of 44.1 kHz.

(ii) To search optimum weights and biases of ANN, two optimization techniques have been proposed. First technique is predator influenced civilized swarm optimization (PCSO) in which swarm particles are divided into a number of societies and global best particle of the swarm is chased by predator particle. The predator effect helps to exploit the search area more effectively. Second technique is based on integration of global and local search techniques. In this technique, predator prey optimization (PPO) has been considered as the global search technique and Hooke-Jeeves method is undertaken as local search technique. In predator prey optimization with Hooke-Jeeves method (PPO-HJ), initial search is performed by PPO technique and in order to further enhance the search, global best solution obtained from PPO is given as input to Hooke-Jeeves method.

(iii) For SVM classifier, the hyper-parameters have been optimized by proposed PPO-HJ technique.

(iv) A mixed variable PPO (MVPPO) technique has also been proposed in this work. The mixed variable PPO with Hooke-Jeeves (MVPPO-HJ) method is applied for the selection of an appropriate feature set and also for the selection of optimized hyper-parameters.

(v) For training of HMM classifier, PPO and PCSO optimization techniques have been integrated with BW algorithm.

(vi) For continuous speech recognition, two hybrid classifier models have been proposed. These are optimized ANN-HMM and optimized SVM-HMM classifiers. In the optimized ANN-HMM hybrid model, the weights and biases of ANN are optimized with PPO-HJ technique and output of ANN is used to estimate the posterior probabilities of HMM. In optimized SVM-HMM hybrid model, SVM hyper-parameters are optimized with PPO-HJ technique and posterior probabilities of HMM are computed from SVM.

(vii) An Interface has also been developed for speech recognition system.

The chapter one presents the history of ASR system and detail of ASR system components. This chapter also enlists the need for study, objectives of the present study, and outlines the organization of thesis. The chapter two provides brief literature review on various aspects of ASR system. Besides this, review of Hindi speech recognition and optimization techniques in the field of pattern recognition has been done in this chapter.The intent of chapter three is to recognize isolated speech words using ANN classifiers. In this chapter, two hybrid optimization techniques are proposed to search optimum set of weights and biases of ANN. The linear predictive coding coefficient (LPCC), Mel-frequency cepstral coefficient(MFCC) and wavelet packet Mel-frequency cepstral coefficient(WPMFCC) features are extracted from speech signal to conduct the experiments. In chapter four, SVM classifier is explored for speech recognition purpose. In this chapter, the effect of dynamic frame size for feature extraction has been investigated. Other important issues are selection of appropriate feature set of speech and SVM kernel parameters. So, a hybrid optimization technique (MVPPO-HJ) is proposed to improve the learning ability of SVM and to select the most appropriate feature set. The experimental results obtained by proposed technique using

SVM classifier shows satisfactory recognition rate. Further, ROC curve has also been analyzed to verify sensitivity and specificity of the results obtained by MVPPO-HJ technique with SVM.In chapter five, Hindi speech recognition system for isolated words and continuous speech has been developed using HMM. In this chapter, two global search techniques have been integrated with BW algorithm to search HMM model parameters,*i.e.*, transition and emission probabilities. To evaluate the performance, average log likelihood values have been computed during training process. The intent of chapter six is to recognize isolated words and continuous speech using optimized hybrid classifiers. Two optimized hybrid classifiers, *i.e.*, ANN-HMM, SVM-HMM have been proposed. The PPO-HJ technique is applied to optimize weights and biases of ANN in ANN-HMM classifier and RBF kernel parameters in SVM-HMM classifier. Finally, chapter seven presents the inferences drawn from the results of the various experiments conducted in this thesis. Also, some pointers to the further research on the topic under consideration in this thesis are discussed briefly in this chapter.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACO	Ant Colony Optimization
ANFIS	Adaptive Neuro Fuzzy Inference System
ANN	Artificial Neural Network
ASR	Automatic Speech Recognition
AUC	Area Under the Curve
BP	Back-Propagation
BW	Baum-Welch
CL	Civilization Leader
CSO	Civilized Swarm Optimization
DBN	Dynamic Bayesian Network
DDBHMM	Duration Distributed Based HMM
DE	Differential Evolution
DFT	Discrete Fourier Transform
DP	Dynamic Programming
DTA	Dynamic Time Alignment
DWT	Discrete Wavelet Transform
FFA	FireFly Algorithm
FFT	Fast Fourier Transform
FN	False Negative
FNN	Feed-forward Neural Network
FP	False Positive
FPR	False Positive Rate
GA	Genetic Algorithm
GD	Gradient Descent
GFA-HMM	Generative Factor Analyzed HMM
GLRPNN	Generalize Local Recurrent Probabilistic Neural Network
HMM	Hidden Markov Model

HTK	HMM Toolkit
LM	Language Model
LPCC	Linear Predictive Coding Coefficient
LPC	Linear Predictive Coding
LSTM	Long Short Term Memory
MFDWC	Mel-Frequency Discrete Wavelet Coefficients
MFCC	Mel Frequency Cepstral Coefficient
MLP	Multilayer Perceptron
MSE	Mean Square Error
MVPSO	Mixed variable PSO
MVPPO	Mixed variable PPO
MVPSO-HJ	Mixed variable PSO with Hooke-Jeeves Method
MVPPO-HJ	Mixed-Variable PPO with Hooke-Jeeves method
NNE	Neural Network Ensemble
ORF	Optimal Relaxation Factor
PCSO	Predator influenced Civilized Swarm Optimization
PDP	Parallel Distributed Processing
PPO	Predator Prey Optimization
PPO-HJ	Predator Prey Optimization with Hooke-Jeeves Method
PSO	Particle Swarm Optimization
RASTA-PLP	Relative Spectral Perceptual Linear Prediction
RBF	Radial Basis Function
RLS	Recursive Least Squares
RWFB	Redundant Wavelet Filter-Banks
SCA	Society Civilization Algorithm
SL	Society Leader
SM	Society Member
SNR	Signal to Noise Ratio
SRM	Structural Risk Minimization

SRR	Sentence Recognition Rate
STFT	Short Term Fourier Transform
SVM	Support Vector Machine
SVM-PPO	SVM with PPO
SVM-PPO-HJ	SVM with PPO-HJ
SVM-PSO	SVM with PSO
TDNN	Time-Delayed Neural Network
TN	True Negative
TP	True Positive
TPR	True Positive Rate
VQ	Vector Quantization
WER	Word Error Rate
WPMFCC	Wavelet PacketMel Frequency Cepstral Coefficient
WPT	Wavelet Packet Transform
WRR	Word Recognition Rate
WT	Wavelet Transform

Chapter 1
Introduction

Speech is one of the most common sources of communication among human beings (Juang and Rabiner, 2005). The speech is generated by the vocal tract and is a continuous signal. The variability in speech signal makes automatic speech recognition (ASR) a challenging task. The variability in speech depends on environmental conditions, speaker attributes such as emotion, age, gender and many other factors. The main objective of ASR is to build a system that can map the acoustic signal into a string of words. An ASR system has two main elements, *i.e.*, front-end processor and back-end classifier. The front-end processor is used to extract speech features and these features are processed by a back-end classifier for speech recognition. Practically, a wide range of applications are possible with the ASR systems.

1.1 HISTORICAL BACKGROUND

Researchers are trying for automatic speech recognition for a very long period. In 1881, Graham Bell and his team invented a recording device, which responded to sound pressure. The first patent on automatic speech system has been allotted to Tihamer Nemes but this transcriber has been turned down in 1930 (Giridhar Rao, 1989). A very significant methodology shift in the field of ASR system started in the 1980's. In this technological shift, template-based pattern recognition strategies were replaced by a more rigorous statistical modelling framework, namely, Hidden Markov Model (HMM). HMMs are the most acceptable statistical models for ASR systems these days. The HMMs have extensively been used over the past two decades for ASR systems, with improvements in the methodology being made on a continual basis.

Another classification technique, artificial neural networks, was introduced in 1950's. It works on the principle of mimicking the human neural processing mechanism and initially it failed to produce any promising results (McCullough and Pitts, 1943). The artificial neural networks (ANNs) along with parallel distributed processing (PDP) model, introduced in the late 1980's, have produced notable results. The PDP is basically formed by interconnection of computational elements, and also employs a training method. A multi-layer perceptron is a particular form of PDP. The ANN is capable to approximate any function to arbitrary precision. The ANNs do not have any restrictions in the complexity of the processing configuration. In 1990, ANNs were applied to recognize only few phonemes or a few words (Lippmann, 1990). The speech recognition systems should be able to handle temporal variations and ANNs have not proven to be extensible to this task. The researchers have integrated ANNs with the HMMs to overcome this disadvantage.

A number of inventions have been introduced in the 1990's, for solving pattern recognition problems. The support vector machine (SVM) is one of the most promising pattern recognition techniques. The success some of statistical based methods revived the interest from DARPA. Some innovative ASR system including the Sphinx system from CMU (Lee, 1988), the DECIPHER system from SRI (Murveit *et al.*, 1989) and the BYBLOS system from BBN (Schwartz *et al.*, 1989) have been developed. Sphinx system has successfully been integrated with HMM, and it has achieved promising results for large-vocabulary continuous speech recognition.

The major achievements in the field of ASR systems started in 1960's. In the 1960's, it was possible to recognize small vocabularies of isolated words. In the 1970's ASR systems had the ability to recognize medium vocabularies tasks using template-based pattern recognition techniques. In the field of speech recognition, a major achievement started in the year 1980's with the addition of HMM technology, which allows recognizing large vocabulary speech dataset. Recently, researchers have proposed advancement in these technologies to improve speech recognition rate for practical applications. Some of the applications such as face recognition, stock trading, ECG analysis, *etc.* have already been introduced in commercial space and some new applications will continue to emerge to improve quality of life.

1.2 COMPONENTS OF ASR SYSTEM

An ASR system consists of four major stages. The first stage of an ASR system is data pre-processing stage. After data pre-processing, features are extracted from the speech signal at feature extraction stage. The third and fourth stages of the system are language modelling stage and classification stage. The block diagram of a speech recognition system is shown in Figure 1.1 and each stage is explained in further sub-sections.

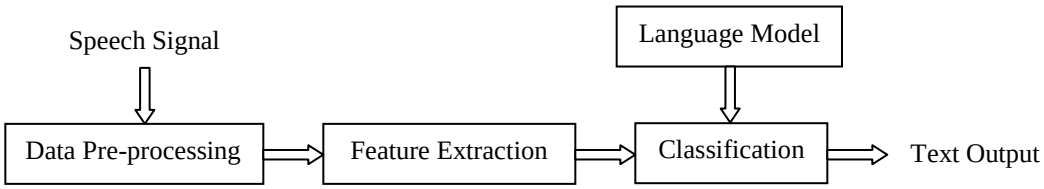


Figure 1.1: Block diagram of a speech recognition system

1.2.1 Data Pre-processing

The data pre-processing stage is the primary stage of an ASR system and it transforms the speech signal before passing it to feature extraction stage. The data pre-processing stage involves pre-emphasis, end point detection, framing and windowing as shown in Figure 1.2. The process of spectrally flatten the digitized speech signal by passing it through a first order finite impulse response filter is known as pre-emphasis. In speech processing, it is very important to detect the voice region. So, end point detection is applied to remove the silence region before and after the voice region (Shin *et al.*, 2010). For this purpose, energy and zero crossing rates are calculated (Faycal and Messaoud, 2014). After that, framing is done for segmenting the speech samples into small frames of approximately 20 to 40 ms. Framing allows the non-stationary speech samples to be segmented into quasi-stationary frames (Tan and Lindberg, 2010) and each frame overlaps its previous frame by a predefined size. After framing, windowing is done to each individual frame so as to minimize the signal discontinuities at the beginning and end of each frame. Normally, the Hamming window is used for windowing as it introduces the least amount of distortion.

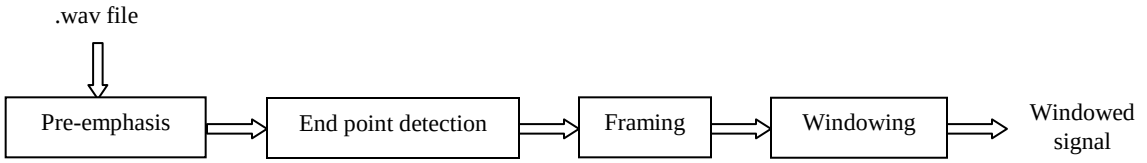


Figure 1.2: Block diagram of data pre-processing stage

1.2.2 Feature Extraction

Feature extraction is the process by which the speech signal is converted into a sequence of acoustic feature vector. The feature vector contains information of utterance which is having good discrimination property. These spectral feature vectors are used for distinguishing between the similar utterances. Another advantage of spectral feature vectors is that it is easy to model statistically, which require small amounts of training data. A number of feature extraction methods have been proposed by various researchers. The most commonly used features for ASR systems are Mel frequency cepstral coefficients (MFCCs), linear predictive coding coefficients (LPCCs), and wavelet based coefficients.

1.2.2.1 Linear predictive coding coefficients

Linear predictive coding (LPC) is a time domain approach in which future values of a digital signal are estimated as a linear function of previous samples.

$$s(n) \approx a_1 s(n-1) + a_2 s(n-2) + ... + a_p s(n-p) \quad (1.1)$$

where, n is the index of the current sample, $s(n)$ is the actual sample, p is the degree of the LPC model, and $a_i, i = 1, 2, ..., p$ is the filter predictor coefficient.

For each sample, a prediction error $e(n)$ is defined as:

$$e(n) = s(n) - \bar{s}(n) \quad (1.2)$$

where, $\bar{s}(n)$ is the linearly predictive sample. By minimizing the prediction error, $e(n)$ over a finite interval, a unique set of linear predictive coding coefficients can be determined.

1.2.2.2 Mel frequency cepstral coefficients

To extract the MFCCs, discrete Fourier transform (DFT) of the windowed signal is computed and is given to the Mel frequency warping block. During Mel frequency warping, the width of the triangular filters varies and also the log of total energy in a critical band around the center frequency is included. After warping, a numbers of coefficients are obtained. Finally the inverse DFT is used to compute the cepstral coefficients (Rabiner and Schafer, 1978). The steps involved in the MFCC feature extraction are shown in Figure 1.3. The first and second derivatives of MFCCs are termed as delta and acceleration coefficients, respectively.

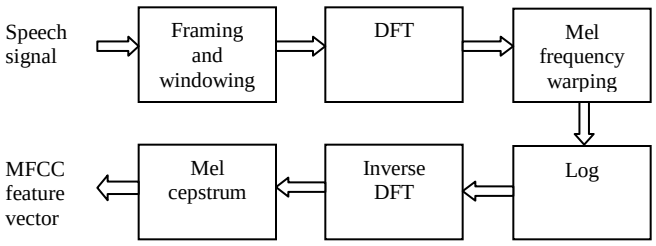


Figure 1.3: Block diagram of MFCC feature extraction process

1.2.2.3 Wavelet based features

Wavelet transform is a powerful tool for extracting features from a speech signal. It has the advantage of using variable size time-windows for different frequency bands that results in a high frequency resolution in low bands and low frequency resolution in high bands. Wavelets are applied in the forms, such as the discrete wavelet transform (DWT) (Nehe and Holambe, 2012; Patil and Dixit, 2012), and wavelet packet transform (WPT) (Coifman *et al.*, 1990; Avci and Akpolat, 2006; Wu and Lin, 2009).

DWT is the process of filtering the signal using a low pass filter and a high pass filter. Thus, first level of DWT decomposition of a signal splits it into two bands giving a low pass version and a high pass version of the signal (Mallat, 1989). In speech signals, the low pass filtered signal gives the approximate coefficients of the signal while the high pass filtered signal gives the details coefficients. The second level of decomposition is performed on the low pass signal obtained from the first level of decomposition (Figure 1.4). Thus wavelet decomposition results in a binary tree like structure which is left recursive (Nehe and

Holambe, 2012). Figure 1.5 (a) shows the three levels of decomposition of signal using discrete wavelet transform.

Wavelet packet transform, proposed by Coifman *et al.* (1990), is an extension of the DWT, in which the whole time-frequency plane is subdivided into different time-frequency pieces. Wavelet packet decomposition facilitates the partitioning of the higher frequency into smaller bands, which cannot be achieved by using discrete wavelet transform. Figure 1.5 (b) shows the three levels of decomposition of signal using wavelet packet transform. The WPT decomposes the approximate spaces as well as detail spaces. To extract WPMFCCs, first the wavelet packet transform coefficients of the speech signal are computed and then MFCCs of these coefficients are calculated as discussed in Section 1.2.2.2.

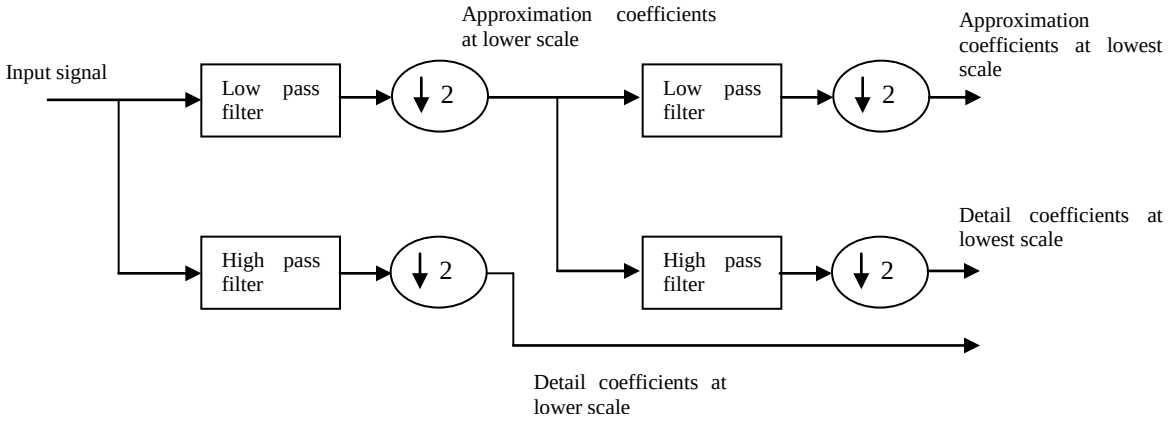
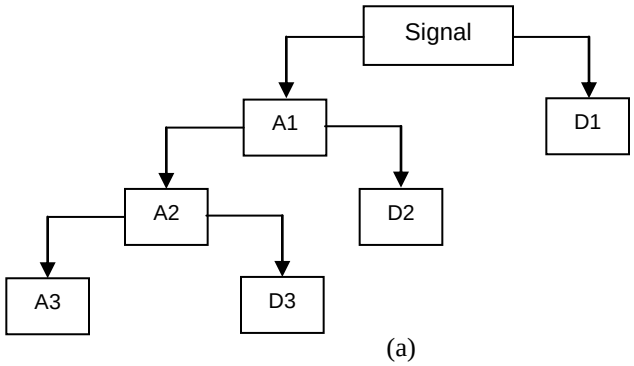
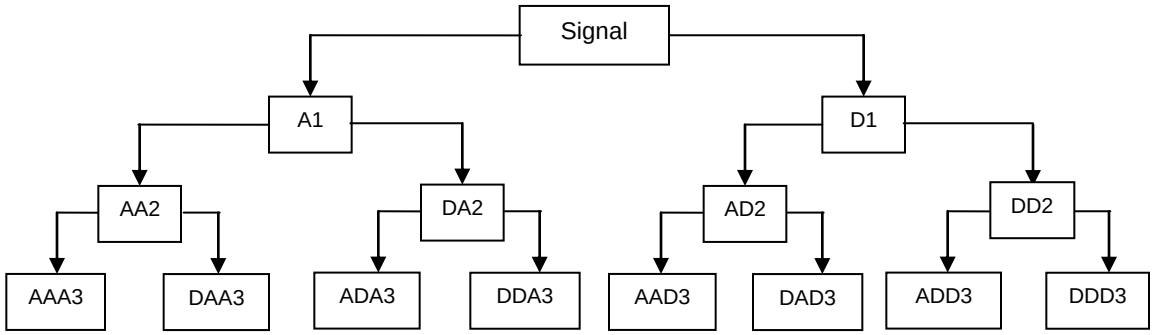


Figure 1.4: Decomposition of signal using discrete wavelet transform



(a)



(b)

Figure 1.5: Three level decomposition of a signal using (a) discrete wavelet transform (b) wavelet packet transform

1.2.3 Language Model

During early phase of speech recognition, it was assumed that only acoustic information is required to produce a text output. In the last two decades, language models have become equal partners with acoustic models. The language model refers to the constraints on word sequences imposed by syntactic, semantic and pragmatic rules of language being spoken (Shaughnessy, 2003). The language model is required to recognize not only the phonemes but also words or sentences.

1.2.4 Classification

The classification stage is the decision making stage of an ASR system. It uses the features extracted and the language model to recognize the speech signal. Classification methods are divided in two categories. The first is the generative approach and second is the discriminative approach. In the generative approach, joint probability distribution is computed over the given observations and the class labels. The resulting joint probability distribution is then used to predict the output for a new input (Cutajar, 2013). HMMs and Gaussian mixture models are the most commonly used generative approaches. In discriminative approach, the conditional distribution is found using a parametric model, where parameters are computed from a training set consisting of pairs of the input vectors and their corresponding target vectors. ANNs and SVMs are two well-liked methods based

on discriminative approach. Currently, researchers are focusing on hybrid models of generative and discriminative based methods, in order to combine the strengths of both approaches.

1.2.4.1 Hidden Markov model

HMM is a doubly stochastic process with an underlying stochastic process that is not directly observable but can be observed through another set of stochastic processes that produces the sequence of observations. HMM is a widely used statistical method of characterizing the spectral properties of the frames of a speech signal. HMM provides a natural and highly reliable way of recognizing speech for a wide range of applications.

HMMs are based on Markov chains that can be used to model a sequence of events. In the Markov model, each state corresponds to one observable event. But for a large number of observations the size of the model explodes. So, the hidden Markov concept extends the model by decoupling the observation sequence and the state sequence. For each state a probability distribution is defined that specifies how likely every observation symbol is to be generated in that particular state.

An HMM can be characterized by the following elements:

- The number of states N.
- A state space $Q = \{1, 2, \ldots, N\}$
- The number of observation symbols per state M.
- An output alphabet $V = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_M\}$.
- The state transition probability distribution $A = \{a_{ij}\}$ where
  $a_{ij} = P[q_{t+1} = j \mid q_t = i], \quad 1 \le i, \quad j \le N$
  where j is that state in which the model is in at a particular time t
- The observation symbol probability distribution, $B = \{b_j(k)\}$, in which
  $b_j(k) = P[o_t = v_k \mid q_t = j], \quad 1 \le k \le M$
  defines the symbol distribution in state j, $j = 1, 2, \ldots, N$.
- The initial state distribution $\pi = \{\pi_i\}$ in which
  $\pi_i = P[q_1 = i], \quad 1 \le i \le N$

To indicate the complete parameter set of the model, the notation $\lambda = (A, B, \pi)$ is used.

1.2.4.2 Artificial neural networks

ANNs are flexible mathematical structure and these are based on the neural structure of the brain. These networks are inspired by biological neural network systems in which input information is processed through interconnected group of neurons (Haykin, 1994). The ANNs are capable of mapping a nonlinear relation between input and output data sets. It is one of the effective learning tools to solve modelling, classification and recognition problems (Mehrotra*et al.,* 1997). A neural network consists of various layers of interconnected elements called neurons. Each neuron has a weighted sum of its inputs producing an output that is transformed via the use of a linear or non-linear function. The ANNs can easily be trained from samples. The other main advantages of ANNs are their good performance with noisy samples and possibility of parallel implementation.

A single layer feed-forward network consists of input and output layer. The input layer neurons take the input signals and the output layer neurons receive the output signals. Each input layer neuron is connected to every output layer neuron through synaptic links carrying the weights but not vice-versa. Figure 1.6 illustrates a single layer feed-forward network.

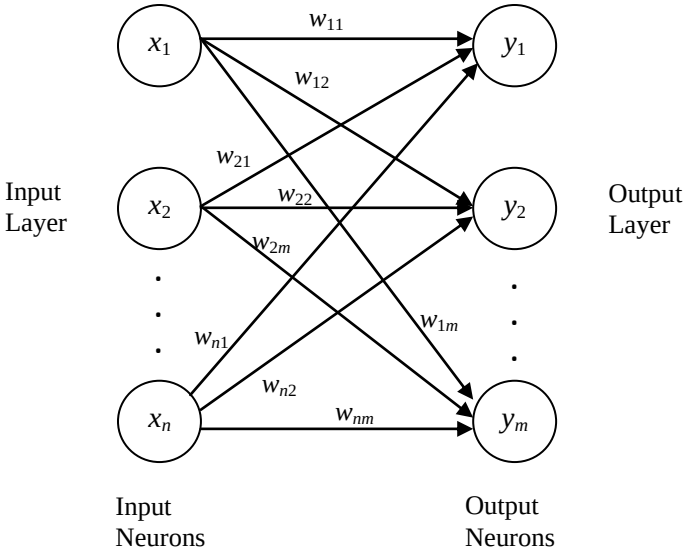


Figure 1.6: Single layer feed-forward network

A multilayer feed-forward ANN consists of input layer, output layer and one or more hidden layers (Jànos, 2012). The hidden layer performs intermediate computations between input and output layers. The input-hidden layer weights connect input layer neurons to hidden layer neurons. The hidden-output layer weights connect hidden layer neurons to output layer neurons. A multilayer feed-forward network having *l* input neurons, *m* neurons in one hidden layer and *n* output neurons is termed as *l-m-n* architecture. Figure 1.7 illustrates a multilayer feed-forward network with this architecture.

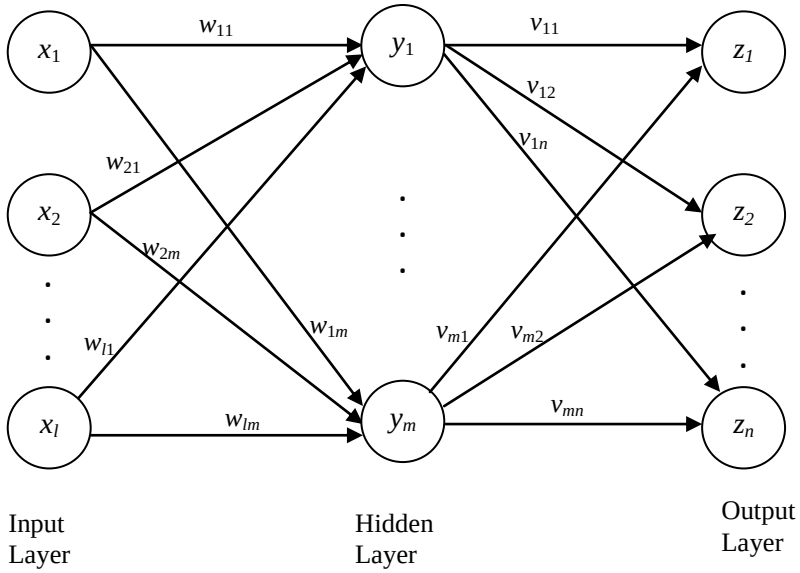


Figure 1.7: Multilayer feed-forward network

The net input at hidden layer is the weighted sum of its input and biases of each neuron and is defined as:

$$Y_j = \sum_{i=1}^{l} (x_i W_{ji}) + b_j \qquad ,(j = 1, 2, ..., m) \tag{1.3}$$

where W_{ji} is the weight representing the connection between j^{th} neuron of hidden layer and i^{th} neuron of input layer; x_i represents the input at neuron *i*; b_j is the bias of neuron *j*; *l* and *m* are the number of neurons in input layer and hidden layer, respectively.

To get the final output, the sum is passed through a transfer function. In this work, sigmoid transfer function is used between input layer and hidden layer; and linear transfer function is used between hidden layer and output layer. The output neuron computes the weighted sum of its inputs as:

$$Z_k = \sum_{j=0}^{m} V_{kj}\ y_j \quad , \qquad k = 1, 2, \ldots, n \tag{1.4}$$

whereV_{kj} is the weight associated with the connection between hidden neuron j and output neuron *k*; *m* and *n* are the number of neurons in hidden layer and output layer, respectively.
The training of ANN is normally carried out by applying iterative optimization process to minimize mean square error (MSE) by updating the weights and biases appropriately (Das *et al.*, 2014). The MSE is defined as the error between the actual output and the expected output and is defined as:

$$MSE = \frac{\sum_{i=1}^{N} (A_i - E_i)^2}{N} \tag{1.5}$$

where A_i is actual output and E_i is expected output; N is number of training points.

1.2.4.3 Support vector machine

Support vector machine, proposed by Vapnik (1995), is a binary classifier. It works on the principles of structural risk minimization (SRM) (Burges, 1998). SVMs have been successfully applied for many classification problems such as face identification, bioinformatics, database marketing and speech recognition (Bennett and Campbell, 2000). In support vector machine, the input space is mapped into a higher dimensional feature space. In higher dimensional feature space, SRM is used to maximize the margin between two classes and an optimal separating hyper plane is obtained. For a binary classification, the training dataset is $\{(x_i, y_i),\ i = 1, 2, ..., n\}$, where the input space $x_i \in R^n$ and $y_i \in \{1, -1\}$ are the labels of the input space x_i and n denotes the number of the data items in the training set. The hyper-plane is defined as $w.x_i + b = 0$, where x_i is a point on the separating hyper-plane, *w* is the normal vector to hyper-plane and *b* is bias value.

The optimal hyper-plane to separate the two classes is obtained by minimizing the regularized training error,

$$E=\frac{1}{2}\|w\|^2+C\sum_{i=1}^{n}\xi_i$$

subject to $y_i\left(\langle w,x_i\rangle+b\right)\geq 1-\xi_i,\ \xi_i\geq 0,\ i=1,2,...,n$ (1.6)

where $\langle,\rangle$ denotes the inner product, ξ_i is a slack variable, which defines the permitted misclassification error, C is the penalty parameter coefficient and it decides the trade-off between the regularization term and the empirical risk.

The key feature of SVM is the use of kernels (Burges, 1998; Cristianini and Taylor, 2000) that implicitly compute an inner product between the two data vectors in the high dimensional feature space. Various kernels such as linear, polynomial and radial basis function (RBF) can be used in SVM (Hwang and Kim, 2012). These kernels are given as:

Linear kernel:

$$K(x_i,x_j)=\langle x_i,x_j\rangle \quad (1.7)$$

Polynomial kernel:

$$K(x_i,x_j)=\left(\gamma\langle x_i,x_j\rangle+r\right)^d,\ \gamma>0 \quad (1.8)$$

Radial basis function (RBF) kernel:

$$K(x_i,x_j)=\exp\left(-\gamma\|x_i-x_j\|^2\right),\ \gamma>0 \quad (1.9)$$

where γ, r and d are kernel parameters.

The selection of a kernel function is an important issue, because kernel function has a bearing on the results.

There are two methods to solve multi-class problem using SVM. One method is one-versus-one, which learns to classify one class from another class (Solera-Ureña *et al.*, 2007) and another is one-versus-all, which learns to classify one class from all other classes (Manikandan and Venkataramani, 2011; Vapnik, 1995). If the number of classes is N then $N(N-1)/2$ one-versus-one classifiers are required as compared to N one-versus-all classifiers (Allwein *et al.*, 2000; Urena *et al.*, 2012; Weston and Watkins, 1999).

1.3 OPTIMIZATION TECHNIQUES

Optimization techniques are extensively used to solve real world optimization problems. These techniques are broadly divided into population based techniques and conventional search techniques. The population based techniques have multi-point search methods and include randomness to update the solution. One of the main advantages of population based search techniques is that these can be applied to solve non-differentiable, discontinues optimization problems and have a potential to search global optimum solutions. In the past decades, several global search techniques, such as genetic algorithm, differential evolution, particle swarm optimization, ant colony optimization and gravitational search algorithm have successfully been applied. Although population based search techniques are able to explore search area effectively but these techniques are not very efficient for fine tuning of solution. Particle swarm optimization technique is one of the most commonly used global optimization techniques. The PSO and its variants have been applied in various fields *i.e.*, to reduce higher order model of linera-time invariant system (Panda *et al.*, 2009), for optimum power flow problem (Abido, 2002), for optimal LQR tracking control of helicopter (Kumar *et al.*, 2016) and many more. The conventional search techniques follow certain mathematical rules to update the search. These techniques are well suited to exploit the search in nearby search spaces, but are not suitable to search in wide spaces. Due to their respective strengths and weaknesses, there is a motivation to integrate population based search technique with conventional search technique. In the field of pattern recognition, following are the major areas in which search based optimization techniques have been applied:

- Optimum feature selection
- HMM Classifier training
- Selection of optimum set of weights and biases for ANN classifier
- Optimum SVM hyper-parameters.

1.4 NEED FOR STUDY

Automatic speech recognition systems are now commercially available to recognize isolated words and sentences with high recognition rate for English, Arab and Chinese languages. A

few research papers are available to recognize Hindi words. Researchers have not explored ASR systems for the recognition of Hindi sentences. There is a possibility to explore hybrid features and to develop a speech recognition system for Hindi sentences with the help of existing techniques. The reason for choosing Hindi language is that it is widely spoken in various parts of India. The alphabets of Hindi language are very well categorized on the basis of similarities in articulation methods of its letters. This property of Hindi makes it free of homonyms thus reducing the complexity of handling them in design of speech recognition systems.

In recent years, optimization techniques are explored in the field of pattern recognition to improve recognition rate and to decrease computational time. Still, very few researchers have applied search based optimization techniques for ASR systems. The present work is an attempt towards the development of ASR system for Hindi speech. This work also involves few integrated optimization search techniques to improve speech recognition rate.

1.5 OBJECTIVES OF THIS RESEARCH

The objectives of the proposed study are outlined as follows:

- To explore existing features and their hybrid approach for Hindi speech word recognition.
- To recognize words with the help of existing techniques such as ANN, HMM, SVM and to propose an efficient recognition method based on the hybrid approach of these methods.
- To propose and implement a model for the recognition of speaker dependent simple Hindi sentences using small vocabulary language model and to develop an interface for the same.

1.6 ORGANIZATION OF THESIS

Rest of this thesis is organized as follows. In Chapter 2, review of literature has been presented. The review is organized according to various stages of speech recognition system, *i.e.*, feature extraction, feature selection, classification techniques and language model.

Besides this, review of Hindi speech recognition has been done in this chapter. The review related to optimization techniques in the field of feature selection, and classification techniques has also been presented. In Chapter 3, ANN is applied to recognize isolated words. To train the ANN, two hybrid optimization algorithms have been proposed. In Chapter 4, SVM classifier is used for recognition of isolated words. In this chapter, an optimization algorithm is applied to search optimal RBF kernel parameters and speech feature set. Chapter 5 presents HMM based speech recognition system for isolated and continuous Hindi speech. For training of HMM, global search technique along with Baum-Welch algorithm has been applied. Chapter 6 presents optimized hybrid classifiers for continuous speech recognition. Finally, in Chapter 7, thesis is concluded with a summary and discussion on further research work. Figure 1.8 contains a flowchart indicating the use of proposed techniques in variouschapters of this thesis.

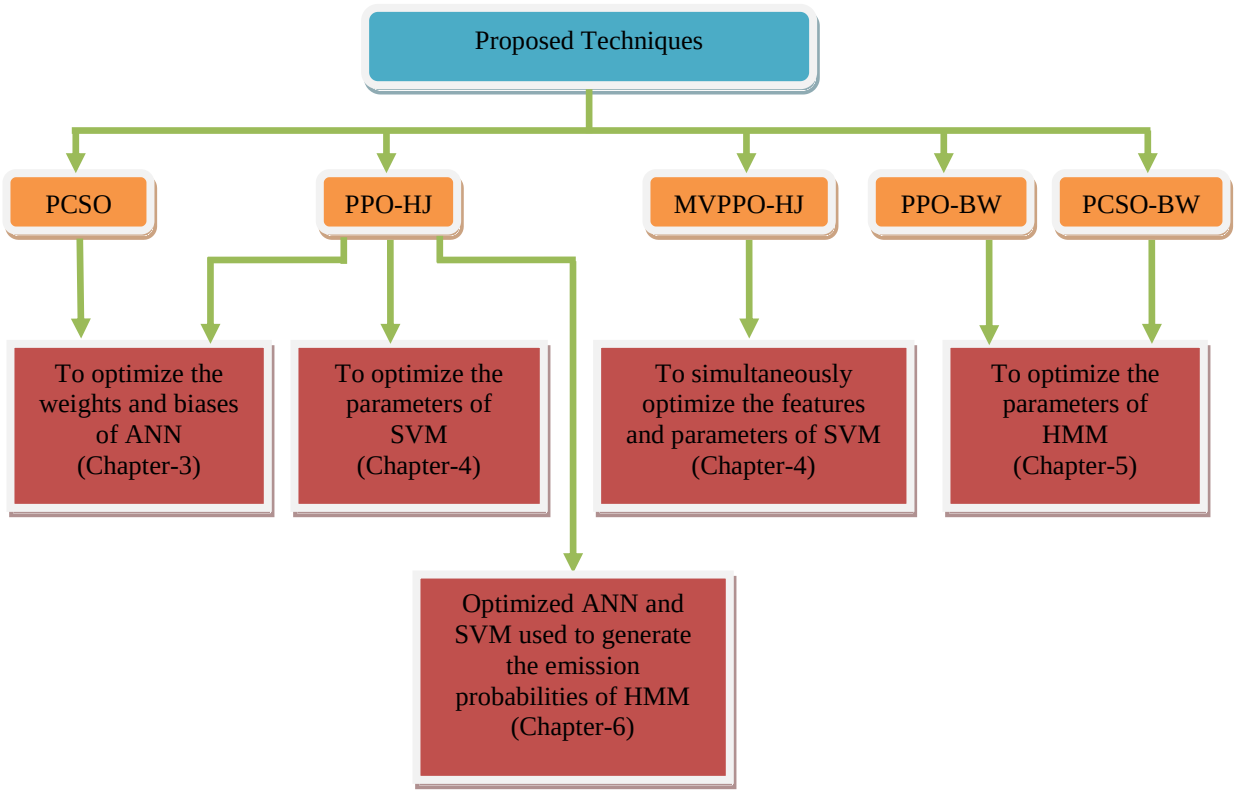


Fig 1.8: Flowchart showing various proposed techniques in different chapters

Chapter 2
Review of Literature

2.1 INTRODUCTION

Speech is the most natural way of human communication. For an efficient human-computer interaction, recognition system must be able to recognize the speech accurately. The performance of automatic speech recognition (ASR) systems has been improved during last decade and now it has been used for many practical applications. Even now, there are lot of possibilities to improve their recognition rate, speed, vocabulary and usefulness for the end-users. Another issue with ASR system is that it has not been developed for a good number of languages due to limited data availability and proper statistical framework of acoustic and language models. Hindi is the national language of India and people in several other Asian countries can easily understand and speak it. So there is a need to develop an efficient ASR system for Hindi language. Speech recognition comes under the category of pattern recognition and carries a considerable development prospect and practical value. A lot of research has been done on speech recognition in last sixty five years. Researchers have published many survey papers on the speech recognition. Shaughnessy (2008) has presented a tutorial regarding speech recognition problem. In this tutorial, he has discussed challenges in the field of speech recognition, successes and failures. Another review paper (Gaikwad *et al.*, 2010) presented an overview of main scientific perspectives and appreciations of the significant progress in the field of ASR system. A report is presented by Anusuya and Katti (2011) about various aspects of front end analysis of speech recognition. Vimala and Radha (2012) have presented some of the major challenges in the field of automatic speech recognition and also discussed regarding speech features and classification techniques. A recent review (Cutajar *et al.*, 2013) presented a comparative study on different classification approaches like HMMs, ANNs and SVMs for the task of ASR. Sanchez-Cortina *et al.* (2016) have shown that a word-dependent naïve Bayes classifier outperforms the conventional word

posterior probability as a confidence measure. Kim and Park (2016) have proposed an efficient emotion recognition approach that utilizes voice data accumulated on personal devices. Agarwalla and Sarma (2016) have reported certain approaches based on machine learning used for extraction of relevant samples from big data and apply them for ASR using certain soft computing for Assamese speech dialectal variations. Karpagavalli and Chandra (2015) have developed speaker independent isolated-phoneme and word recognition systems for the Indian regional language Tamil.An automatic speech recognition system mainly consists of four stages: pre-processing stage, feature extraction stage, classification stage and a language modelling stage.

The pre-processing stage is required before extracting any information from the speech signal. The pre-processing steps include filtering, sampling, quantization, and endpoint detection (Nouza *et al.*, 2010; Mporas *et al.*, 2007; Liu *et al.*, 2005). The literature regarding feature extraction, classification and language modelling has been reviewed in further sections.

2.2 FEATURE EXTRACTION

The feature extraction provides the acoustic feature vectors used to characterize the spectral properties of time varying speech signal. After pre-processing, characteristics parameters, *i.e.*, short time energy, short-term zero-crossing rate, and short-time autocorrelation coefficient are extracted from speech signal. Two most commonly used feature parameters are LPCC and MFCC.

The LPCCs were developed in 1960s and are widely used till today because LPCC features are accurate, computationally fast and easy to implement in hardware (Atal, 2006). But LPCC shows degraded performance in noisy conditions and its robustness, recognition rate and other aspects are less efficient for practical applications. The MFCCs are most commonly used speech features and have successfully been used in speech recognition from a long time (Chapaneri, 2012; Muda *et al.*, 2010; Wang *et al.*, 2010; Kopparapu and Laxminarayana, 2010; Milner and Darch, 2011; Milner *et al.*, 2009). The MFCCs are Cepstral based features that are compact and easily discriminable. The MFCC also has high noise immunity and the robustness (Xin Xing and Xu, 2012). Polur and Miller (2005) have extracted Fast Fourier transform (FFT), LPC, and MFCC features for speech recognition using HMM. They have found that a MFCC based model outperformed a FFT and linear

prediction based model. Aida-Zade *et al.* (2006) have explored combined use of MFCC and LPC features for ASR. The training and testing are performed for both MFCC and LPC based subsystems and the decision is based on the results obtained by these subsystems. Kopparapu and Bhuvanagiri (2013) have proposed a modified Mel filter bank that is based on a relationship between the MFCC features of original sampled speech and re sampled speech. Valentini-Botinhao *et al.* (2014) have presented a method for modifying the MFCC generated by statistical parametric models trained on plain speech.Jo *et al.* (2016) have presented an energy-efficient architecture to extract MFCC for real-time speech recognition systems.

The MFCC features are obtained by short term Fourier transforms (STFT) (Wu and Lin, 2009). The STFT uses fixed window length and because of that it is not able to detect sudden changes and transient parts of signal properly (Yang and Zhang, 2008; Farooq and Datta, 2001). To overcome this disadvantage, the wavelet transform (WT) is used in speech processing (Favero, 1994). The WT has the advantage of alterable window in time domain (Xueying and Zhiping, 2004). In the field of speech recognition, researchers have conducted various studies to integrate WT with auditory-model-based scales. The main advantage of discrete wavelet transform (DWT) features is that it shows high frequency resolution capability in low frequency range of signal while carrying time resolution in high frequency parts of speech signals (Xueying and Zhiping, 2004). Tufekci *et al.* (2006) have explored an idea in which they have placed DWT in MFCC features instead of cosine transform. Chang *et al.* (1998) have described adaptive wavelet feature for speech recognition. Farooq and Datta (2003) have proposed the use of DWT for the extraction of features from phonemes. Datta and Farooq (2001) have suggested that corresponding energies of the wave coefficients can be used as features. They have used the logarithm of output filter-bank energies as parameters representation. The wavelet packet transform (WPT) has been proposed for the feature extraction (Long and Datta, 1996; Long and Datta, 1998; Chang *et al.*, 1998). The only difference between WPT and DWT is that approximation and detail coefficients are further decomposed (Krishnan and Anto, 2009). In WPT, the problem of shift variance was present in the extracted features. Farooq and Datta (2001) have explored WPTs multi resolution capabilities to derive new set of features for speech recognition. To improve speech recognition rate for noisy speech, Farooq and Datta (2001) have explored a six-band filter structure by using permissible wavelet packet. Avci and Akpolat (2006) have developed a model based on integration of adaptive wavelet packet network with fuzzy inference system. Krishnan and Anto (2009) have used DWT and wavelet packet decomposition features for Malayalam language spoken words. Pavez and Silva (2012) have proposed wavelet-packet

cepstral coefficients for filter-bank energy based feature extraction technique. In recent years, researchers have explored some of the hybrid/modified feature extraction techniques. Pujol *et al.* (2005) have applied the relative spectral perceptual linear prediction (Rasta-PLP) features for recognition of clean and noisy speech. In Rasta-PLP, multi-layer perceptron and HMM/Gaussian mixer models have been integrated.

Tufekci and Gowdy (2000) have proposed a feature vector consisting of Mel-frequency discrete wavelet coefficients (MFDWC) by applying the DWT to the mel-scaled log filter-bank energies of a speech frame. Ricotti (2005) has introduced speech parameterization scheme and a modification of the MFCC processing scheme based on wavelets. Tohidypour *et al.* (2012) have proposed a speech representation based on redundant wavelet filter-banks (RWFB). They have shown that RWFB parameters are much less shift sensitive than those of DWT in speech recognition task. Debyeche *et al.* (2007) have presented modified vector quantization (VQ) approach for discrete HMM. The modified VQ codebook performs an optimal distribution of VQ codebook components on HMM states. The proposed technique is applied to recognize Arabic constants.Nguyen *et al.* (2016) have proposed a spectro-temporal transform for feature adaption as well as a minimum KL divergence based criterion for estimating the transform parameters. The feature adaption technique makes use of both spectral and temporal information which is suitable for dealing with both spectral variability and temporal variability. Kim *et al.* (2016) have presented a new feature extraction algorithm called power normalized cepstral coefficients (PNCCs). The PNCCs are motivated by auditory processing which includes the use of power-law nonlinearity.

The choice of a particular feature for ASR system depends on a number of criteria such as, clean or noisy environmental condition, temporal and frequency information of speech signals, memory storage size available and robustness (Cutajar *et al.*, 2013). The advantages and disadvantages of some of the commonly used features are listed in Table 2.1.

Table 2.1: Advantages and disadvantages of feature extraction techniques

Feature extraction technique	Advantages	Disadvantages
MFCC	(i) Perception based method (Anusuya and Katti, 2011). (ii) Good discrimination and small correlation between components (Anusuya and Katti, 2011). (iii) Mimic some of the human processing of the signal (Anusuya and Katti, 2011).	(i) Computationally expensive (Anusuya and Katti, 2011). (ii) Not suitable in noisy environment (Anusuya and Katti, 2011). (iii) Limitation in speech signal representation since only the power spectrum is considered, ignoring the phase spectrum of speech signals (Cutajar *et al.*, 2013).
LPC	(i) It is a production based model (Anusuya and Katti, 2011).	(i) The speech representation is not possible with linear scales (Shaughnessy, 2003).

	(ii) Provides linear characteristics (Anusuya and Katti, 2011). (iii) Provides a reasonable source-vocal tract separation (Anusuya and Katti, 2011).	(ii) LPC is highly correlated (Anusuya and Katti, 2011). (iii) Priori information cannot be included during test (Anusuya and Katti, 2011).
LPCC	(i) The feature components are decorrelated (Cutajar et al., 2013). (ii) Better robustness as compared to LPC (Anusuya and Katti, 2011).	(i) The speech representation is not possible with linear scales (Shaughnessy, 2003). (ii) Priori information cannot be included during test (Anusuya and Katti, 2011).
DWT	(i) Temporal information also consider along with frequency information (Muller *et al.*, 2006). (ii) Efficient time and frequency localisations are possible (Walker & Foo, 2003). (iii) De-noising tasks (Walker and Foo, 2003). (iv) Can compress a signal without major degradation (Anusuya and Katti, 2011).	(i) The same wavelet is used for all speech signals so it is not flexible (Anusuya and Katti, 2011).
WPT	(i) Same as DWT, but WPT shows also further detail present in the high frequency bands (Vimal Krishnan and Babu Anto, 2009).	(i) The same wavelet is used for all speech signals so it is not flexible (Anusuya and Katti, 2011).
PLP	(i) Based on short term spectrum of the speech (Anusuya and Katti, 2011). (ii) Computationally efficient and it yields a low-dimensional representation of the speech (Anusuya and Katti, 2011). (iii) Can mimic some of the human processing of the signal (Anusuya and Katti, 2011).	(i) Analysis depends of the result on the overall spectral balance on formant amplitudes (Anusuya and Katti, 2011). (ii) The spectral balance is easily affected by recording equipment, the communication channel and additive noise (Anusuya and Katti, 2011).
RASTA-PLP	(i) More robust as compared to PLP (Anusuya and Katti, 2011). (ii) Efficient in dealing with convolution noise (Anusuya and Katti, 2011). (iii) Can suppress the spectral components which change quickly or slowly than the range of change of speech (Anusuya and Katti, 2011).	(i) Poor performance under clean speech environments (Anusuya and Katti, 2011).
VQ	(i) Reduced storage for spectral analysis information (Anusuya and Katti, 2011). (ii) Less computation is required (Anusuya and Katti, 2011). (iii) Fast training speed (Krishnan *et al.*, 1994)	(i) Temporal information is not considered (Krishnan *et al*., 1994). (ii) Quantization error in the discrete representation of speech signals (Krishnan *et al*., 1994).

2.2.1 Feature Selection

In the field of pattern recognition, selection of relevant features is one of the most important issues (Uncu and Türkşen, 2007). The application of feature selection techniques can considerably reduce the number of features for ASR system without compromising recognition rate (Inza, 2001). To search optimal features, Kohavi and John (1997) have proposed wrapper method for a specific problem. They have also proposed some modified

versions of wrapper method. Wrapper method cannot be applied for a problem having a very large number of predictive attributes (Bermejo *et al.*, 2011). Uncu and Türkşen (2007) have proposed a feature selection algorithm for identification of important input variables. In the proposed feature selection algorithm, wrapper and filter methods have been combined. Maldonado and Weber (2009) have introduced a modified wrapper algorithm for feature selection, using SVMs with kernel functions. Kabir *et al.* (2010) have presented a modified feature selection algorithm based on wrapper approach using neural networks. One of the main advantages of this algorithm is the determination of neural network architectures during the feed forward process. Hsu *et al.* (2011) have introduced a hybrid feature selection method by combining filters and wrappers methods. Candidate features are first selected from the original feature set via computationally efficient filters and are further refined by more accurate wrappers. Bermejo *et al.* (2012) have proposed an iterative method that alternately shifts between filer ranking construction and wrapper method. Foithong *et al.* (2012) have proposed a hybrid model for feature selection. In the hybrid model, filter and wrapper methods have been integrated that uses the mutual information criterion without requiring a user-defined parameter for the selection of the candidate feature set.

2.3 CLASSIFICATION TECHNIQUES

Various researches have been carried out in order to find ideal classifier to recognize speech segments correctly. Three most widely used classifiers are HMMs, ANNs and SVMs. In this section, the literature on these three classifiers is presented with a focus on their implementations for ASR.

2.3.1 Hidden Markov Models

Hidden Markov model is a robust statistical method for ASR system. HMM has been successfully applied for a wide variety of applications. The utterances of speech is described by HMM model parameters (Kwong and Chau, 1997). Rabiner (1989) has presented the theoretical aspect of HMM. He has also illustrated the application of HMM for machine recognition of speech. In 1994, Renals *et al.* (1994) have reviewed the basis of HMM speech recognition and showed the benefits of incorporating connectionist network into a HMM network. In 1996, Ostendorf *et al.* (1996) have discussed a stochastic model that covers models presented in the literature, compare similarities of the models. Juang and Rabiner

(1991) have addressed the role of HMM in speech recognition and discussed a wide range of theoretical and practical issues of their importance. Wilpon *et al.* (1990) have suggested some modifications in a connected word speech recognition algorithm based on HMM. They have created statistical models of both the actual vocabulary words and the extraneous speech and background.

Pisarn and Theeramunkong (2007) have presented an approach to construct recognizer for the three commonly used Thai spelling methods based on HMMs. Saharia *et al.* (2009) have presented a work on part of speech tagging for Assmese language using HMM model. Latorre and Furui (2006) have presented a method for synthesizing multiple languages with the same voices, using HMM-based speech synthesis. In this approach, they have created a speaker-and language-independent acoustic model by mixing speech data from several speakers in different languages. Zeng and Liu (2006) have presented an extension of HMMs based on type-2 fuzzy set referred to T2FHMMs to recognize phonemes from TIMIT speech database. Membership functions of type-2 fuzzy set are three dimensional that offers additional degree of freedom to evaluate the HMMs fuzziness. Yao *et al.* (2005) have presented a generative factor analyzed HMM (GFA-HMM) for automatic speech recognition. In spite of HMM offering number of advantages, it has few shortcomings. To improve the capability of HMM, researchers have suggested various modifications (Trentin and Gori, 2006).

For real-world applications, robust acoustic model is required. Unfortunately, the HMMs are not noise tolerant. Ephraim (1992) has developed gain-adapted training and recognition algorithms for HMMs to recognize clean and noisy speech signals. Trentin and Gori (2006) have proposed an unsupervised maximum-likelihood gradient-ascent training algorithm for a neural feature adaption module, combined with a hybrid connectionist/HMM speech recognizer. For clean and noisy speech recognition, Chi *et al.* (2013) have explored Markov modeling of stereo speech features.

One of the main reasons for the inaccuracy of HMM-based automatic segmentation is the HMM training criterion and duration control. The discriminative training methods have achieved tremendous progress in automatic speech recognition (Jiang, 2010). Jiang (2010) has presented a reviewed report on discriminative training methods in speech recognition from both theoretical and practical perspectives. McDermott *et al.* (2007) have reported that discriminative training based on minimum classification error training applied to HMM, produces significant gains in recognition performance and model compactness. Wu *et al.* (2005) have proposed minimum segmentation error criteria and applied it along with

discriminative training method. In the proposed approach, a loss function is directly related to the segmentation error and parameter optimization is done by the generalized probabilistic descent algorithm. Champion and Houghton (2016) have described an optimal algorithm using continuous state HMM.Hoesen *et al.* (2016) have structured an Indonesian speech recognizer to handle both spontaneous and dictated speech. They have build recognizer based on Gaussian and HMMs.In spite of HMMs have been proved worthy for ASR system. Still, ASR is a challenging task under real world environment conditions (Trentin and Gori, 2001). To overcome some of the limitations of HMM, ANN has been explored from last two decades for ASR.

2.3.2 Artificial Neural Networks

Artificial neural networks have the potential to resolve pattern classification problems and also have the capability to extract useful information from data (Khan *et al.*, 2013). One of the main feature of ANN is its adaptability, so it is capable to classify new data effectively (Krishnan and Anto, 2009; Zhou, 2009). Some of the widely used neural network architectures are radial basis functions (RBF), multilayer perceptrons (MLP), self-organizing maps and recurrent neural networks. A historical survey on ANNs is presented by Schmidhuber (2015). He has reviewed deep supervised learning; unsupervised learning; reinforcement learning and evolutionary computation; and indirect search for short programs encoding deep and large networks. The researchers have extensively used ANN for classification purposes in various fields *i.e.* vowel classification, image processing, stock trading, face recognition, ECG analysis, market forcasting, *etc.* (Hagan *et al*., 1996). (Bhatt *et al.* (2014) have developed ANN-based apple classification. Bhatt and pant (2009) have presented a method to forecast the stock price using neural networks. Yılmaz *et al.* (2016) have described a code-switching ASR system built for the Frisian language. They have designed a bilingual deep NN based ASR system and investigated the impact of bilingual DNN training in the context of code-switching speech.

For automatic speech recognition, ANN is second most widely used classifier after HMM. Lippmann (1989) has presented a review of the application of ANN to ASR. ANNs are used either independently or as a combination with HMMs in order to get the advantages of both ANNs and HMMs (Hennebert *et al.*, 1994; Shaughnessy, 2003; Zhou, 2009). The major issue of designing an ANN for speech recognition is to deal with the dynamic properties of the speech signal. An ANN model with time-delayed neural network (TDNN)

has been proposed to process the dynamic information of speech (Waibel *et al.*, 1989). In TDNN, short-time delays are included in the input and hidden layers of a MLP in such a way that node responses over several time intervals are collectively fed forward to the neurons in the upper layer. The use of recurrent Boltzmann machines for speech recognition is explored by Prager *et al.* (1986). The networks with recurrent connections from output nodes to input nodes are studied by Robinson *et al.* (1988) and Anderson *et al.* (1988). Watrous and Shastri (1987) have proposed networks with recurrent self-looping connections on hidden and output nodes for speech recognition. Tebelskis and Waibel (1990) have proposed the ANN as a nonlinear pattern predictor rather than a discriminator for speech frames. Wu and Chan (1993) have presented an MLP network for speaker-independent isolated word speech recognition. The network has architecture of three subnets and associated adaptive learning algorithm is derived. Yu and Oh (2000) have introduced acoustic sub-word units to ANNs for speaker independent continuous speech recognition. They have separated segmentation and recognition and both are implemented by networks and spectral transition measure has been applied to decide unit boundaries. Ting *et al.* (2013) have presented a self-adjusted ANN to enable the network to adjust itself according to different data input sizes. They have applied proposed method to recognize Malay vowels and TIMIT isolated words. Dede and Sazli (2010) have applied ANNs to accomplish isolated speech recognition. The recognition is done in two parts, in first part pre-processing part is done with digital signal processing techniques and then post-processing part is done with ANN. They have designed three different neural network models; multi-layer back propagation, Elman neural network, and probabilistic neural networks. Ahad *et al.* (2002) have used MLP to recognize Urdu digits from a mono speaker database. The Persian digits were recognized by utilizing MLP by Pour *et al.* (2009).

Sivaram and Hermansky (2012) have introduced sparse MLP network. The sparse MLP network is similar to MLP network except that the outputs of one of the hidden layer are forced to be spare. The sparse MLP based system is applied on individual speech recognition feature streams. Fuzzy neural network (FNN) is applied by Zhou *et al.* (2009) for speech recognition. The FNN converges at optimal solution during training process since sounds in speech signal do not have clear boundaries. Adaptive neuro fuzzy inference system (ANFIS) has been implemented for recognition of isolated Persian words (Helmi and Helmi, 2008). Sabah and Aino (2009) have applied ANFIS for classification of speaker-independent isolated Malay digit speech signals. Graves and Schmidhuber (2005) have proposed long short term memory (LSTM) networks, and full gradient version of the LSTM learning

algorithm. To check the credibility of LSTM algorithm, it has been tested for frame wise phoneme classification. Siniscalchi *et al.* (2013) have applied deep neural networks to improve recognition rate of basic speech units. This network has higher flexibility and able to integrate both top-down and bottom-up knowledge into speech framework. Ganchev *et al.* (2007) have introduced generalize local recurrent probabilistic neural network (GLR PNN) by including a fully connected recurrent layer between the pattern and output layers.Ozyildirim and Avci (2013) have proposed modified radial basis function (RBF) based neural networks having five layers: input, pattern, summation, normalization and output layers. The network has gradient descent based optimization for smoothing parameters and diverge effect for calculation improvements.

In spite of several advantages of ANN, the main drawback of ANN is their inability in representing the time variability present in speech signal. HMMs have been proven successful for acoustic modeling in state-of-the-art speech recognition systems and for computation of acoustic parameters, ANNs have been found more suitable (Trentin and Gori, 2003). So, there is a need to hybridize ANN and HMM for ASR. Morgan and Bourlard (1995) have given a brief overview of ASR and statistical pattern recognition. They have reviewed HMMs and then described the use of ANNs as statistical estimators. The basic principles of HMM/ANN hybrid approach is also reviewed. Bengio *et al.* (1992) have integrated multilayered and recurrent ANNs with HMMs. In the integrated technique, ANN output constitute the sequence of observation vectors for the HMM. Rigoll (1994) has proposed a hybrid connectionist-HMM speech recognition system based on the use of a neural network as vector quantizer. Chen *et al.* (1996) have proposed a MLP based speech recognition system. In this method, the dynamic time warping capability of HMM is directly combined with the discriminant based learning of MLP to generate a sequence of MLPs as a word recognizer.

The ANN is frequently used to compute the posterior probabilities to improve the recognition rate obtained from HMMs. For ASR, Kim and Un (1995) have discussed various methods for estimating a probability distribution based on discrete HMM. Trentin and Gori (2001) have reviewed a number of significant hybrid ANN/HMM models for ASR. They have focused on ANNs to estimate posterior probabilities to the states of an HMM and on global optimization. Trentin and Gori (2003) have proposed hybrid ANN/HMM system, in which ANN is trained with gradient-ascent technique and is used to estimate emission probabilities for the HMM. Sivaram and Hermansky (2011) have applied hybrid HMM-SMLP for phoneme recognition where the posterior probabilities depend on the sparse hidden

features. The hybrid model based on RBF and HMM is applied for word recognition in a continuous speech environment (Umarani *et al.*, 2009). In this model, an HMM is constructed for each word in the database and a target value is associated with each HMM. Mohamed and Nair (2012) have combined HMM and ANN classifiers for the development of Malayalam speech recognition system.

2.3.3 Support Vector Machine

For pattern classification problems, support vector machine is one of the most popular approaches. Support vector machine has sound mathematical background and extraordinary performance (Doumpos *et al.*, 2007). SVM maps different objects into a high-dimensional space and then separates the classes with a hyperplane. The main motive of SVM framework is to maximize the separating margin between objects belonging in different classes. The most important aspect of SVM is the design of the inner product, the kernel, induced by the high dimensional mapping (Campbell *et al.*, 2006). Initially, SVM was designed for binary classification. There are two types of approaches used for multiclass SVM, one-against-all method and one-against-one method. In one-against-all method, the multiclass problem is divided into a number of binary SVM classifiers and each classifier builds a hyperplane between its own class and for remaining classes. In one-against-one method, classes are separated from each other by constructing hyperplane for each possible pair of classes. For both these methods, output class is decided by a majority voting scheme (Duan and Keerthi, 2005). Hsu and Lin (2002) have compared the different methods for multiclass SVMs. For multi-classification problems, SVMs are usually converted into binary ones, where unclassified regions may exist. To overcome this drawback, Liu *et al.* (2005) have presented nesting SVMs for multi-classification. An empirical study of multiclass SVM is presented by Duan and Keerthi (2005).

SVMs have extensively been used for pattern classification from last two decades. A tutorial on SVMs for pattern recognition is presented by Burges (1998). He has described practical implementation of SVM and discussed the kernel mapping technique to construct SVM solutions. Researchers have explored SVMs for ASR either independently or as a hybrid model with HMM. Ganapathiraju *et al.* (2004) have applied SVM for static pattern classification task based on the Deterding vowel data and achieved significant improvement. Gangashetty *et al.* (2004) have studied the SVM performance for recognition of context dependent subword units of speech in multiple languages. Khoo *et al.* (2005) have

investigated the performance of SVM for phoneme recognition with additive noise. Campbell *et al.* (2006) have examined the idea of using the GMM supervector in a SVM classifier and investigated the performance of classifier on NIST speaker recognition task. Campbell *et al.* (2006) have considered the applications of SVMs to speaker and language recognition. They have proposed sequence kernel based on generalized linear discriminants. To search endpoint detection of speech, Ramírez *et al.* (2006) have applied SVM-based technique. To improve the performance of speech recognition system, Ramírez *et al.* (2006) have presented a speech discrimination method to work in noisy conditions. They have applied their proposed method to train SVM that defines an optimized non-linear decision rule over different sets of speech features. Truong *et al.* (2007) have presented a multi-speaker segmentation technique by using wavelets features along with SVM classifier. Gales and Flego (2010) have described a scheme for overcoming the problem of mismatching between the training and test data. They have applied this scheme for speech recognition in noise by adapting the kernel rather than the SVM decision boundary. Manikandan and Venkataramani (2011) have proposed Texas Instruments DSP real-time speech recognition system using modified one-against-all SVM classifier. Zhang *et al.* (2013) have presented optimal relaxation factor (ORF) SVM kernel function for speech recognition. They have shown the effectiveness of ORF kernel functions on mapping trend, bi-spiral, and speech recognition problems. Sonkamble and Doye (2008) have provided an overview of speech recognition systems using SVMs.

The SVM is also hybridized with HMM for ASR applications, because of inability of SVM to deal with variable input vectors and high dimensionality of the sequences of speech feature vectors (Fernåndez-Lorenzana *et al.*, 2003). There are two approaches that have been used for hybridization of HMM and SVM. In one of the approaches, the distance measure of the separation between classified data points is used to generate posterior probabilities. These emission probabilities are used in HMM instead of GMMs. Sloin and Burshtein (2008) have applied SVM to improve classification of discrete and continuous output probability of HMMs. They have used HMM as a baseline system, and SVM is used to rescore the results obtained by HMMs. Ganapathiraju *et al.* (2004) have presented SVM with HMM to recognize large vocabulary speech. Liu *et al.* (2007) have proposed a hybrid SVM and duration distributed based HMM decision fusion model (DDBHMM) for robust continuous digital speech recognition. They have investigated the probable combinations of SVM and GMM in pattern recognition, and embed the fusion probability into the phone state level decision space of DDBHMM. Solera-Ureńa *et al.* (2007) have compared hybrid HMM-SVM and dynamic time alignment (DTA) kernel approaches in noisy environments for robust ASR

and concluded that DTA kernel provides important advantages over the baseline HMM. Clarkson and Moreno (1999) have applied hybrid combination of SVM and HMM for phonetic classification.

SVM is also hybridized with ANN for ASR. For pattern classification and regression, Xia *et al.* (2004) have used a one-layer recurrent neural network for SVM learning. Bresolin *et al.* (2008) have used WPT and neural classifier SVM to recognize spoken digits from zero to nine in Brazilian Portuguese.

2.4 LANGUAGE MODEL

In order to produce a meaningful representation of speech signal, knowledge of spoken language is necessary (Liddy, 2001). In 1960s and 1970s, it was assumed that the acoustic information of speech is needed to produce a text output. Speech recognition community has not explored the fact that most spoken utterances follow certain rules of grammar and semantics (Shaughnessy, 2003). In the last two decades, to implement these rules, researchers have suggested that, there is a need to incorporate language model (LM) in ASR systems along with acoustic models. The language model refers to a set of constraints on the words available in the vocabulary set and their corresponding sequences. The performance of an ASR system is significantly affected by the choice and scope of the LM (Deshmukh and Picone, 1995). It plays an important role to decide the search space, so there is a need of appropriate search strategy for the ASR process. For real speech, the formal grammar rules are not adequate because it can involve awkward phrasing and abbreviated word-forms. The LM should be able to incorporate such topical dependencies, grammatical constraints, *etc*. So, a good LM should not only be able to consider all these possibilities but also needs to be compact enough, for efficient real-time implementations (Forsberg, 1995; Deshmukh and Picone, 1995). In order to choose the optimal LM, a set of criteria, *i.e.*, perplexity, average log likelihood, cross entropy and resultant accuracy (Deshmukh and Picone, 1995; Rosenfeld, 2000) are explored.

The LMs are divided into static and dynamic LMs. A widely renowned static language technique is the *N*-gram model. In N-gram model, the probability of occurrence of the current word is computed by information obtained by *N-1* immediately preceding words (Bahl Lalit, 1983). For bigram model, *N* is set to 2 and for trigram model, *N* is set to 3. A trigram model is more popular than the bigram model, both in terms of accuracy and

perplexity (Deshmukh and Picone, 1995). The main limitation of static models is that these are incapable to adapt in diverse domains. To overcome this limitation, dynamic models need to be explored. In dynamic models, word probabilities are estimated on the part of document observed so far. So, it is capable to adapt in new speech domains. Some of the commonly used dynamic language models are long-distance *N*-grams, trigger-pairs, cache models and tree-based models (Deshmukh and Picone, 1995). After deciding the language model, a decoding search technique is required to select best hypothesis based on a specific criteria. The pruning algorithm is applied to prune the lowest score hypotheses (Deshmukh and Picone, 1995; Illina and Gong, 1996). The Viterbi search and *N*-best search are widely used algorithms. In Viterbi algorithm (David, 1973), all hypotheses correspond to the same portion of speech and hence these can be directly compared with each other. The *N*-best search algorithm considered as an extension of the Viterbi algorithm. The main difference between these two algorithms is that *N*-best search algorithm considers the *n*-best hypotheses instead of best hypothesis as in case of Viterbi algorithm.One of the main concerns of using *N*-best search algorithm is that there is a higher chance of choosing short hypotheses as compared to longer hypotheses because longer sentences will have more errors, thus resulting into lower scores. To improve the performance of *N*-best search algorithm, Deshmukh and Picone (1995) have suggested some modifications in the search algorithm and Illina and Gong (1996) have proposed modifications in pruning method.

2.5 OPTIMIZATION TECHNIQUES

The optimization techniques can be applied to search optimum solution for almost all practical applications. The ultimate aim to apply the optimization technique is to minimize the effort required or to maximize the desired benefits. The optimization techniques broadly divided into global and conventional search techniques. The global search techniques are population based technique and search here is performed by applying heuristics. Global search techniques are applied as a tool to solve complex optimization problems because of their exclusive advantages like global search capability, robust and reliable performance, and no requirement of differentiable and continuous objective function. In the past decades, several global search techniques, such as genetic algorithm (GA), differential evolution (DE), particle swarm optimization (PSO), ant colony optimization (ACO) and gravitational search algorithm have been proven to be efficient to solve complex optimization problems.

Conventional search techniques are further classified as gradient search techniques and direct search techniques. Gradient search techniques follow some specific mathematical rule to search optimal solution. These techniques use the gradient information of the function to get the promising search direction. Some of the common gradient search techniques include steepest descent, Newton's and quasi-Newton methods. Direct search techniques perform the search in favour of exploratory search direction and pattern search direction. These techniques do not require any information from derivative of objective function or constraints, so these techniques can be applied to solve non-differential problems. Powell's pattern search and Hooke-Jeeves methods are commonly used direct search techniques.The optimization techniques are extensively used in the field of pattern recognition. Major areas in which optimization techniques have been applied are: feature selection, HMM classifier training, selection of weights and biases for ANN classifier, and to choose SVM kernel parameters.

2.5.1 Optimization Techniques for Feature Selection

For feature selection, wrapper and filter methods are normally used as discussed in sub-section 2.2.1. The wrapper method generally achieves better recognition rate as compared to filter method because wrapper method utilizes the actual target learning algorithm and on the other side, filter method applies intrinsic data to select features while ignoring the target learning algorithm. But these methods are computationally impractical because of high computational time and memory requirement. To overcome these disadvantages of wrapper and filter methods, global search techniques are applied to select most relevant feature set. Vignolo *et al.* (2013) presented a multi-objective wrapper method based on GA to select the most relevant set of features. Rodrigues *et al.* (2014) have presented a wrapper feature selection approach based on bat algorithm and optimum-pat forest. Hua-chao *et al.* (2007) have proposed a feature selection and classification method for hyper-spectral images by integrating PSO algorithm with SVM. Liu *et al.* (2011) have designed a modified multi-swarm PSO for feature selection. To improve classification accuracy in binary problems, Sarafrazi and Nezamabadi-pour (2013) have proposed a hybrid system consisting of gravitational search algorithm with SVM. An ant colony optimization method with SVM has been applied to improve the classification accuracy with appropriate feature subset by Huang (2009). Hu *et al.*(2015) have applied firefly algorithm (FFA) to select proper input features for midterm interval load forecasting. Sivagaminathan and Ramakrishan (2007) have

presented a hybrid method based on ant colony optimization and ANN to address feature selection. Huang (2009) has proposed a hybrid ACO based classifier model to select appropriate feature set. In proposed hybrid model, ACO and SVM are integrated to improve classification accuracy. Bermejo *et al.* (2011) have proposed a stochastic algorithm based on the greedy randomized adaptive search procedure meta-heuristic to speed-up the feature subset selection process. Al-Ani *et al.* (2013) have proposed a feature selection method that utilizes DE to identify relevant feature subsets. Han *et al.* (2014) have presented modified gravitational search algorithm for feature subset selection. Kashef and Nezamabadi-pour (2015) have proposed a feature selection algorithm based on ant colony optimization. Yu and Cho (2006) have proposed an ensemble creation method based on GA wrapper feature selection. Santana and Canuto (2014) have proposed the application of PSO, ACO and GA to choose subsets of features for the individual components of ensembles.

2.5.2 Optimization Techniques for HMM Classifier

HMM is currently the most popular approach to speech recognition (Yang and Zhang, 2008). In recent years, various researchers have explored different search techniques for optimized model parameters. The Baum-Welch (BW) algorithm is one of the most common parameter estimation methods. The main advantages of BW algorithm are its reliability and efficiency. The BW algorithm is a conventional search method and it can easily trapped in local optimum solution. Kwong and Chau (1997) have presented GA for HMM training. Yang and Zhang (2008) have proposed an algorithm based on PSO and BW to train the continuous HMM in continuous speech recognition. To search an optimum number of states in HMM and its parameters, Kwong *et al.* (2001) have proposed an optimization method based on GA and BW algorithm. Rasmussen and Krink (2003) have applied a hybrid algorithm combining PSO with evolutionary algorithms to train HMMs for the alignment of protein sequences. Hassan *et al.* (2012) have introduced a hybrid of HMM, fuzzy logic and multi-objective evolutionary algorithm for building a fuzzy model to predict non-linear time series data. Sun *et al.* (2012) have proposed quantum-behaved PSO to train HMMs for multiple sequence alignment (Najkar *et al.*, 2010). The Viterbi algorithm is the main core for HMM based speech recognition system and dynamic programming (DP) is used to search the best alignment between the input speech and a given speech model. Najkar *et al.* (2010) have replaced DP with PSO algorithm to find out the best alignment between the input speech and

a given speech model. This idea is tested on an isolated word recognition and phone classification tasks.

2.5.3 Optimization Techniques for ANN Classifier

The ANN classifier takes a considerable time to choose the set of parameters that affects the final performance. The trail-and-error manual process is normally done to select number of hidden layers and nodes, transfer functions, parameters of training algorithm in order to find the best possible set of neural network parameters for a specific problem (Leandro and Teresa, 2010). The selection of weights usually involves a non-linear optimization problem that cannot be solved analytically (Romero and Alquèzar, 2007). Most models of ANN choose the weights in the first layer that correspond to hidden nodes in such a way that its associated output vector matches the previous residue as best as possible. For training of ANN, conventional search techniques such as back-propagation (BP) technique and the recursive least squares (RLS) learning algorithm are commonly used (Bilski and Rutkowski, 1998). The BP is a gradient based method having slow convergence characteristics and may be trapped at local optimum solution (Zhang *et al.*, 2012). The RLS algorithm is very fast as compared to BP technique, but it requires more complicated mathematical operations (Al-Batah, *et al.*, 2010). The gradient descent (GD) algorithm is normally preferred to optimize the parameters of feed-forward neural network (FNN). The GD algorithm shows higher precision and fast convergence speed around the global optimum solution. But it can easily trap to local optimum solution for the non-linear pattern classification problems (Gori and Tesi, 1992).

Researchers have suggested various approaches to optimize parameters of ANN. Wang *et al.* (2014) have applied teaching-learning based optimization algorithm with neighbourhood search to optimize weights and biases of ANN. Jànos (2012) has applied global optimization framework for parameterization of ANNs. Das *et al.* (2014) have applied ANN trained with PSO for the problem of channel equalization. For an ASR system, Almeida and Ludermir (2010) have applied evolution strategies to search optimum parameters of neural network. Zhang *et al.* (2007) have proposed integrated optimization technique based on PSO and BP to train the weights of FNN. Song *et al.* (2007) have presented tent mapping chaotic PSO to improve the performance of neural network for predictive control. The hybrid method based on fuzzy set theory, PSO and BP algorithm is applied to search the weights of the FNNs (Yuan *et al.*, 2011). Zhao *et al.* (2015) have trained the weights of neural network

ensemble (NNE) with the help of a combination of multi-population co-evolution PSO, artificial bee colony, and DE chaotic searching algorithms. Pauplin and Jiang (2012) have presented a Dynamic Bayesian Network (DBN) model for classification of hand written digits. They have applied a fixed DBN structure and evolutionary algorithm for selection and layout of the observations for each digit.

It has been observed from the literature that recurrent neural networks have not been used that extensively when compared with FNN for ASR. The training of recurrent neural network is a tedious task when error gradient based algorithms are used, as these are very unstable in their search and also require high computational time when the number of neurons is large (Blanco *et al.*, 2001). To overcome these deficiencies, Blanco *et al.* (2001) have presented real-coded GA to train recurrent neural network. Chalup and Blair (2003) have presented a hybrid technique based on integration of evolutionary hill climbing with incremental learning to train recurrent neural network for context-free and mildly context-sensitive languages. Rivero *et al.* (2010) have presented genetic programming for ANNs, with a few number of neurons and connections. Khan *et al.* (2013) have proposed a fast learning neuro-evolutionary algorithm for both feed-forward and recurrent networks. Their method is inspired by Cartesian genetic programming technique.

2.5.4 Optimization Techniques for SVM Classifier

A well-known problem in SVM is to choose the specific parameters for a kernel, since it has a high impact on the classification accuracy. Inappropriate parameter settings lead to poor classification results (Zhao *et al.*, 2011). To overcome this problem, parameters of SVM should be optimized. Guo *et al.* (2008) have proposed a hyper-parameter selection method for least-squares SVM based on PSO technique. Lin *et al.* (2008) have developed PSO based approach for parameter determination of the SVM. Ilhan and Tezel (2013) have developed a GA-SVM with parameter optimization. Huang and Dun (2008) have combined the discrete PSO with the continuous-value PSO to optimize the SVM kernel parameters. Vieira *et al.* (2013) have proposed a modified binary PSO method for optimization of SVM parameters. Xiang *et al.* (2006) have optimized SVM parameters by integration of PSO with simulated annealing algorithm. Yan-bin *et al.* (2009) have proposed modified PSO, by incorporating chaotic search to improve the performance of PSO. They have applied modified PSO to search optimum parameters of SVM. Wang *et al.* (2011) have presented modified SVM by combining principal component analysis and PSO to improve the detection rate of an

intrusion detection system. Huang and Wang (2006) have presented a GA based approach to optimize the parameters of SVM for pattern classification. Bao *et al.* (2013) have optimized SVM parameters by applying memetic algorithm based on PSO and pattern search. HE *et al.* (2012) have developed a method for classifying electronic nose data in rats wound infection detection based on SVM and wavelet analysis. They have applied PSO-SVM classifier for pattern recognition. Wei *et al.* (2011) have presented a method based on SVM and PSO for face recognition. Hsieh and Hu (2014) have presented an evolutionary technique to modify the SVM algorithm from a single objective optimization problem to a multi-objective optimization problem. The proposed technique is applied to classify fingerprint classification. Hu *et al.* (2013) have proposed a FFA based memetic algorithm to select the parameters of support vector regression forecasting model.

To improve the speech recognition rate, proper choice of SVM model parameters and most relevant feature subsets of speech is required (Zhao *et al.*, 2011). Shih-Wei *et al.* (2008) have developed PSO based approach for parameters determination of SVM along with feature selection. A combination of discrete and continuous PSO (Huang and Dun, 2008) is applied to select the most relevant feature subset and to search SVM kernel parameters. Huang and Wang (2006) have presented a GA based approach to simultaneously optimize the parameters of SVM and feature subset for pattern classification.

2.6 HINDI SPEECH RECOGNITION

Udhyakumar *et al.* (2004) have analyzed various issues in building a HMM based multilingual speech recognizer for Indian Languages. Malhotra and Khosla (2008) have presented an approach to identify gender and accent of a speaker using Gaussian Mixture Modeling technique. They have tested this approach to identify accent among four regional Indian accents in spoken Hindi and also identified the gender. Chourasia *et al.* (2005) have presented a report on methodology used in the generation of a phonetically rich Hindi text corpus. They have discussed the design, structure and phonetic analysis of text corpus for Hindi. Gupta and Sivakumar (2011) have created a basic building block of speech recognition software for Hindi speech recognition. Bansal *et al.* (2008) have proposed an optimum speaker-independent, isolated word HMM recognizer for Hindi language. Dutta *et al.* (2010) have presented the use of probabilistic neural networks to the classification scheme of demonstrative pronouns for indirect anaphora in Hindi corpus. Saha *et al.* (2012) have

presented a study on different dimensionality reduction approaches applied to the named entity recognition task and they have undertaken Hindi and Bengali languages to compare the performance of various approaches. Neti *et al.* (2004) have presented the work on building the acoustic and language models for Hindi language. Ranjan (2010) has proposed the application of discrete wavelet transform coefficients for recognition of isolated words in Hindi language speech. Biswas *et al.* (2014) have proposed a filter structure using admissible wavelet packet analysis for Hindi phoneme recognition. Anumanchipalli *et al.* (2005) have developed Tamil, Telugu and Marathi language speech databases to build large vocabulary speech recognition systems. Samudravijaya *et al.* (2000) have presented the design and development of an annotated and time-aligned speech database for Hindi language. Sarma *et al.* (2011) have developed Assamese speech corpus. They have mainly focused on some important issues and challenges in Assames language.Saharia *et al.* (2010) have presented a suffix based noun and verb classifier for Assmese language.Sharma *et al.* (2008) have discussed salient issues in Assamese morphology. Saharia *et al.* (2014) have discussed the problem of stemming several resource-poor languages from Eastern India, viz, Assamese, Bengali, Bishnupriya Manipuri and Bodo. Kishore *et al.* (2003) have presented a brief overview of unit selection speech synthesis and discussed the issues relevant to the development of voices for Indian languages.

2.7 COMPARISON OF SPEECH RECOGNITION RATES

It is a known fact that speech recognition rate depends on number of issues. The choice of a particular feature set depends on external conditions and many other factors. Selection of a classifier is also an important issue in ASR system. Each classifier has its own limitations and capabilities. The recognition rate also depends a lot on speech dataset size. This section presents some of the selected speech recognition rates from literature as listed in Table 2.2.

Table 2.2: Comparison of speech recognition rates

Reference	Year	Research work	Language	Feature extraction Technique	Classification Technique	Speech recognition rate (%)
(Prasad and Gerald, 2005)	2005	Isolated word recognition	Dysarthric speech	FFT, LPC, MFCC	HMM	LPC model-79.5, FFT model-89.0, MFCC model-92.0
(Aida-Zade *et al.*, 2006)	2006	Isolated word recognition	Azerbaijan Speech	LPC, MFCC, Combined use of LPC and MFCC	ANN	MFCC-87.5, LPC-91.6, Combined-84.6

(Chang *et al.*, 1998)	1998	Isolated word recognition	Korean Digits	Adaptive wavelet	HMM	81.7
(Avci *et al.*, 2006)	2006	Isolated word recognition	Turkish	WP	Adaptive NN	92.0
(Vimal Krishnan, and BabuAnto, 2009)	2009	Isolated word recognition	Malayalam	DWT, WPT	ANN	DWT-89.0 Wavelet packet-61.0
(Pujol *et al.*, 2005)	2005	Isolated word recognition	Numbers95 database	MFCC	HMM/GMM HMM/MLP	HMM/GMM-91.6 HMM/MLP-91.7
(Chen *et al.*, 1996)	1996	Isolated Mandarin digits	Database is provided by Telecommunication Laboratories	Energy spectra and Delta energy spectra	MLP/HMM	98.5
(Sloin and Burshtein, 2008)	2008	Isolated noisy digit recognition	TIDIGITS	MFCC	SVM/2-HMM	93.4
(de Andrade Bresolin, *et al.*, 2008)	2008	Digits (0 to 9) in Brazilian Portuguese	Self recorded	WPT	SVM	97.7
(Debyeche *et al.*, 2007)	2007	Isolated word recognition	Arabic Constants	VQ	Discrete HMM	91.0
(Pisarn and Theeramunkong, 2007)	2007	Isolated word recognition	Thai spellings	PLP	HMM	93.0
(Wilpon *et al.*, 1990)	1990	Isolated word recognition	5 word vocabulary	LPC	HMM	99.3
(Kwong, 2001)	2001	Isolated word recognition	TIMIT Database	MFCC	GA-HMM	93.1
(Yu and Oh, 2000)	2000	Isolated word recognition	TIMIT Database (520 words)	MFCC	ANN	75.1
(Ting *et al.*, 2013)	2013	Isolated word recognition	TIMIT Database (520 words)	MFCC	Self regulated ANN	92.5
(Farooq and Datta, 2003)	2003	Phonemes	TIMIT database	DWT	MLP	83.5
(Farooq and Datta, 2001)	2001	Phonemes	TIMIT database	Mel Filter-Like Admissible WP Structure	ND	86.0
(Tufekci and Gowdy, 2000)	2000	Phoneme recognition	TIMIT database	Mel-Frequency Discrete Wavelet Coefficients	HMM	61.5
(Prina Ricotti, 2005)	2005	Phoneme recognition	APASCI database	Modified MFCC	HMM	82.6
(Zeng and Liu, 2006)	2006	Phoneme recognition	TIMIT Database	MFCC	Type-2 fuzzy HMM	66.6
(Najkar,	2010	Phoneme	TIMIT	MFCC	PSO-HMM	70.8

2010)		recognition	Database			
(Sivaram and Hermansky, 2012)	2012	Phoneme recognition	TIMIT Database	PLP+FDLP+ MLDA	SMLP	81.4
(Graves and Schmidhuber, 2005)	2005	Phoneme recognition	TIMIT Database	MFCC	Bidirectional long short term memory networks	78.6
(Siniscalchi *et al.*, 2013)	2013	Phoneme recognition	TIMIT Database	MFCC	Deep NN	86.6
(Khoo *et al.*, 2005)	2005	Phoneme recognition under noisy environment	TIMIT Database	PLP, PLP with RASTA	SVM	68.0
(Truong *et al.*, 2007)	2007	TIdigits database	Aurora-2	Fourier transform and the Daubechies length-8 orthogonal wavelet	SVM with Wavelets	85.9
(Zhang *et al.*, 2013)	2013	Vowel recognition	UCI benchmark	MFCC	Optimal relaxation factor kernel function SVM	70.3
(Trentin and Gori, 2006)	2006	Continuous speech recognition	SPK Dataset	MFCC	HMM/ANN	68.6
(de Andrade Bresolin, *et al.*, 2008)	2008	Sentence Recognition	Self recorded sentences	MFCC	HMM	62.2
(Trentin, and Gori, 2003)	2003	Continuous speech recognition over Italian-digit string	SPK Database	MFCC	HMM/ANN	68.6
(Mohamed and Ramachandran Nair, 2012)	2012	Continuous Malayalam	Self recorded speech	MFCC	MLP/HMM	86.6
(Ganapathiraju *et al.*, 2004)	2004	Continuous alphadigit task	Self recorded speech	MFCC	SVM/HMM	92.4
(Liu *et al.*, 2007)	2007	Continuous mandarin digit speech	UCI repository	MFCC	FSVM/duration distributed based HMM	91.7

2.8 CONCLUSIONS

Speech recognition is one of the most complex pattern recognition problems due to variations in spoken speech. There is a very less work reported in literature for Hindi speech recognition in spite of the fact that of world population uses this language. In this chapter, contributions of various researchers related to feature extraction and feature selection have briefly been

reviewed. The work related to different classification techniques, *i.e.*, HMM, ANN, SVM has also been reviewed in this chapter. The review on Hindi speech recognition and language models has also further been included in this chapter. The chapter also reviews optimization techniques used in feature selection and classification processes of speech recognition. Finally, a brief survey is presented to compare the recognition rates reported by various researchers for speech recognition in different domains.

Chapter 3

Recognition of Isolated Words using Optimized ANN Classifier

Automatic speech recognition (ASR) is a challenging and difficult task. A lot of efforts have been made in the past to improve the recognition rate of ASR systems. In this chapter, we have explored ANN models to recognize isolated words. An ANN is a flexible mathematical structure and it is inspired by biological neural network. The ANN processes input information using an interconnected group of artificial neurons. It is able to handle complex nonlinear relations between input and output data sets. The main applications of ANN are in the field of classification, control systems and pattern recognition. Training of an ANN is a tedious task, because search space is high dimensional and multimodal. An ANN training needs efficient optimization techniques to search a set of weights and biases that minimizes the error. For training of ANN, back-propagation (BP) algorithm is the most commonly used algorithm. The BP algorithm is based on gradient search and may get trapped in local optimum solution for multimodal problem. To overcome thes problems of BP algorithm, two hybrid optimization techniques, predator influenced civilized swarm optimization (Technique-I); and predator prey optimization with Hooke-Jeeves method (Technique-II) are proposed in this chapter. The experiments have been carried out by taking conventional BP algorithm, proposed Technique-I and proposed Technique-II to train the ANN model parameters, namely, weights and biases for the development of isolated words recognition system. A benchmark database TI-46 has been used for experimentation. Experiments have also been conducted on a primary database consisting of isolated Hindi words.

3.1 EXPERIMENT 1: ANN TRAINED USING BP ALGORITHM

Artificial neural networks are inspired by biological neural network systems in which input information is processed through interconnected group of neurons (Haykin, 1994). ANN is

one of the effective learning tools to solve modelling, classification and recognition problems (Mehrotra *et al.*, 1997). A feed-forward ANN is illustrated in Figure 3.1, in which input layer, intermediate or hidden layer and output layer is connected in a systematic manner (Jànos, 2012). The net input is the summation of weighted sum of its input and biases of each neuron and is defined as:

$$Y_j = \sum_{i=0}^{n} W_{ji}\, x_i \ , \quad j = 1, 2, \ldots, m \tag{3.1}$$

where x_i represents the input at neuron i, W_{ji} is the weight associated between input layer neuron i and hidden layer neuron j; n and m are the number of neurons in input layer and hidden layer, respectively.

x_0 represents the bias and its value is always 1. W_{j0} is the weight associated with the bias x_0 and is updated like all other weights according to the BP algorithm.

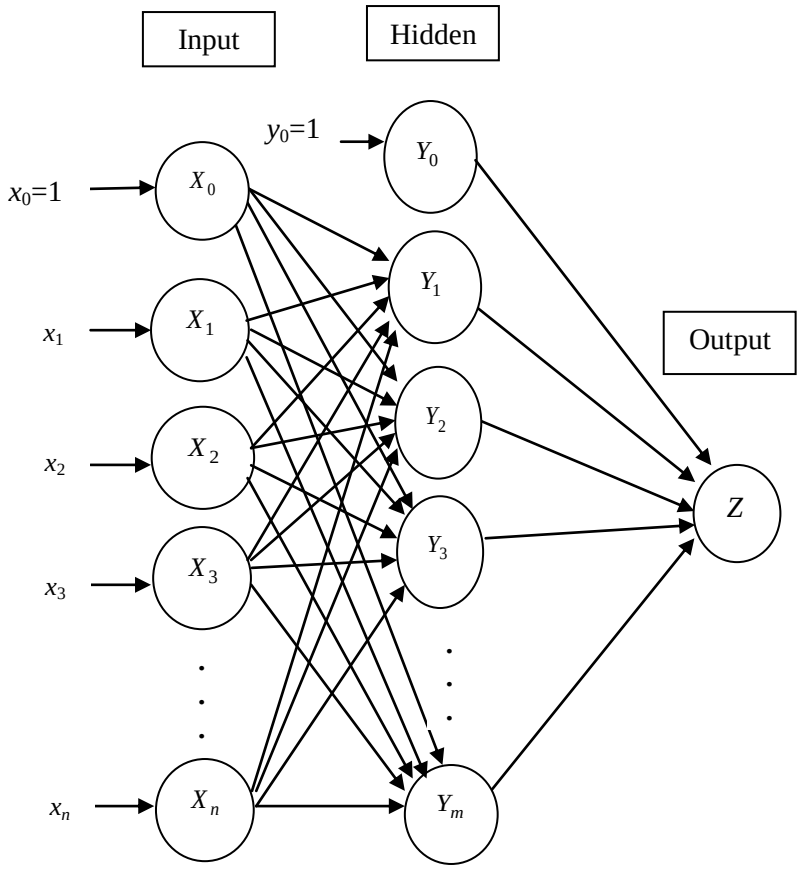


Figure 3.1: ANN articuture

The output of ANN model depends on the choice of transfer function between its layers as it affects the learning rate and performance of the model. Among all the transfer functions

tested in this work, the combination of hyperbolic tangent sigmoid function for hidden layer and linear transfer function for output layer resulted in better performance. At the hidden node, the hyperbolic tangent sigmoid function has been applied to the weighted sum of the inputs to the hidden node, so the output of hidden node is given as:

$$Y_j = \tan sig(Y_j) = \frac{\exp(Y_j) - \exp(-Y_j)}{\exp(Y_j) + \exp(-Y_j)} \tag{3.2}$$

The output neuron computes the weighted sum of its inputs as:

$$Z_k = \sum_{j=0}^{m} V_{kj}\ y_j \quad , \quad k = 1, 2, \ldots, p \tag{3.3}$$

where V_{kj} is the weight associated between hidden layer neuron j and output layer neuron k; m and $p$are the number of neurons in hidden layer and output layer, respectively. $y_0$represents the bias and its value is always 1. V_{k0} is the weight associated with the bias y_0 and is updated like all other weights according to the BP algorithm. For one output unit, so (3.3) can be written as:

$$Z = \sum_{j=0}^{m} V_j\ y_j \tag{3.4}$$

At the output node, linear transfer function has been applied to get the final output and is given as:

$$purelin(Z) = Z \tag{3.5}$$

The training of ANN is carried out by applying iterative optimization process to minimize mean square error (MSE) by updating the weights and biases appropriately (Das *et al.*, 2014). The MSE is defined as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(A_i - E_i)^2 \tag{3.6}$$

where A_i is actual output and E_i is expected output; N is number of training patterns.

3.1.1 Databases Used

In this work, as mentioned earlier, a benchmark database TI-46 (NIST Speech DB, 1991) under clean and noisy conditions and a self recorded isolated Hindi words database have been considered. The TI-46 database has two subsets, TI-20 and TI-ALPHA. The TI-20 database consists of ten English Digits "*zero*" to "*nine*" and ten control words "*yes*", "*no*", "*erase*",

"*rubout*", "*repeat*", "*go*", "*enter*", "*help*", "*stop*", and "*start*". The TI-ALPHA database consists of English alphabets "*a*" through "*z*". Both databases are speaker dependent and have sampling frequency of 12.5 kHz. The third database consists of self created Hindi numerals "*shoonya*", "*ek*", "*do*", "*teen*", "*chaar*", "*paanch*", "*cheh*", "*saat*", "*aath*" and "*nau*" recorded in a quiet room environment with sampling frequency of 44.1 kHz. For each database, the vocabulary size, number of instances and number of speakers are given in Table 3.1.

Table 3.1: Description of databases

Database	Vocabulary size	No. of instances	No. of speakers	
			Male	Female
TI-20	20	26	8	8
TI-ALPHA	26	26	8	8
Hindi digits	10	20	2	2

3.1.2 Implementation

Implementation of the speech recognition system using ANN trained with BP algorithm has been discussed in this section. The first stage of an ASR system is pre-processing. In pre-processing stage, the digitized speech signal is passed through a FIR filter to spectrally flatten the signal. Silence removal and end-point detection of speech signal have also been done in this stage. After pre-processing, to extract LPCC and MFCC features, the speech signal is divided into a fixed number of 40 frames each of 25ms with 50% superposition. After framing, the Hamming window is used for windowing. For each of the 40 frames, 13^{th} order LPCC features are extracted for all speech utterances, resulting into a feature vector of 520 dimensions for each utterance. For MFCC feature extraction, 20 triangular Mel filters are used. MFCC feature vector consists of 13 coefficients for each frame, resulting into a feature vector of 520 dimensions for each utterance. For WPMFCC features, initially the signal is decomposed into sub-bands using the WPT. Each speech sample has been decomposed into 2-level WPT. After decomposition, the speech signal is divided into sub-bands. It is further fed to the MFCC analysis block and 13 MFCC coefficients from each of the bands are extracted, obtaining a feature vector of 52 WPMFCCs for each utterance. The Daubechies 4 wavelet is used for the purpose of decomposition of the speech signal. These extracted acoustic features are given as input to ANN and BP algorithm is applied to optimize the weights and biases of ANN to recognize a given word.

Selection of optimum number of neurons in the hidden layer is a critical issue in creating an ANN model. In this work, to select the optimum number of neurons in the hidden layer, several experiments have been conducted using BP algorithm to train the ANN model. In these experiments, the number of neurons in the hidden layer has been varied from five to thirty with all other parameters fixed. The MSE with different number of hidden neurons for Hindi database with LPCC, MFCC and WPMFCC features is depicted in Figure 3.2. Minimum MSE has been found with 15 neurons in hidden layer for MFCC features, so network with 15 neurons in the hidden layer has been chosen as the best architecture for the MFCC features. For LPCC features, 20 neurons and for WPMFCC features, 10 neurons, in the hidden layer have been found giving minimum MSE. Similar experiments have been carried out for TI-20 and TI-ALPHA databases to find the suitable number of neurons in the hidden layer. For TI-20 database, the minimum MSE has been found with 24 neurons in hidden layer with MFCC features, 27 neurons with LPCC features, and 18 neurons with WPMFCC features (Figure 3.3). For TI-ALPHA database, MFCC features result into minimum MSE when 28 neurons are used in hidden layer; LPCC features result into minimum MSE when 25 neurons are used in hidden layer; and WPMFCC features result into minimum MSE when 22 neurons are used in hidden layer (Figure 3.4). The network is trained by using 10-fold cross validation approach, *i.e.*, the database is divided into ten equal parts, out of which nine parts are used for training and one part is reserved for testing of the model.

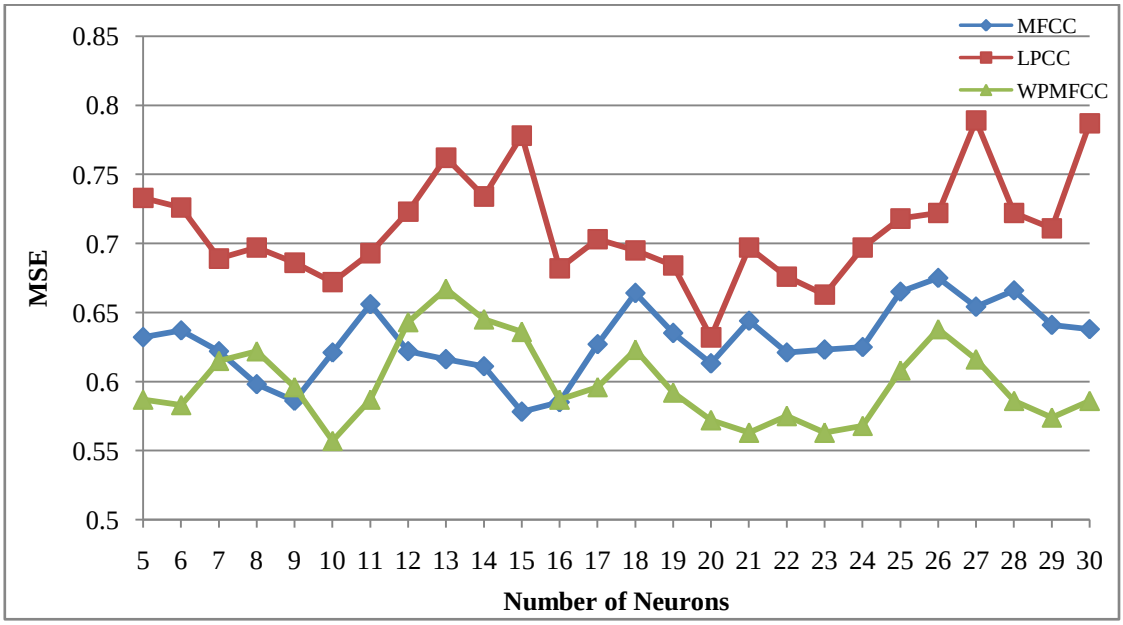


Figure 3.2: Variation in MSE with different number of neurons in the hidden layer for Hindi database

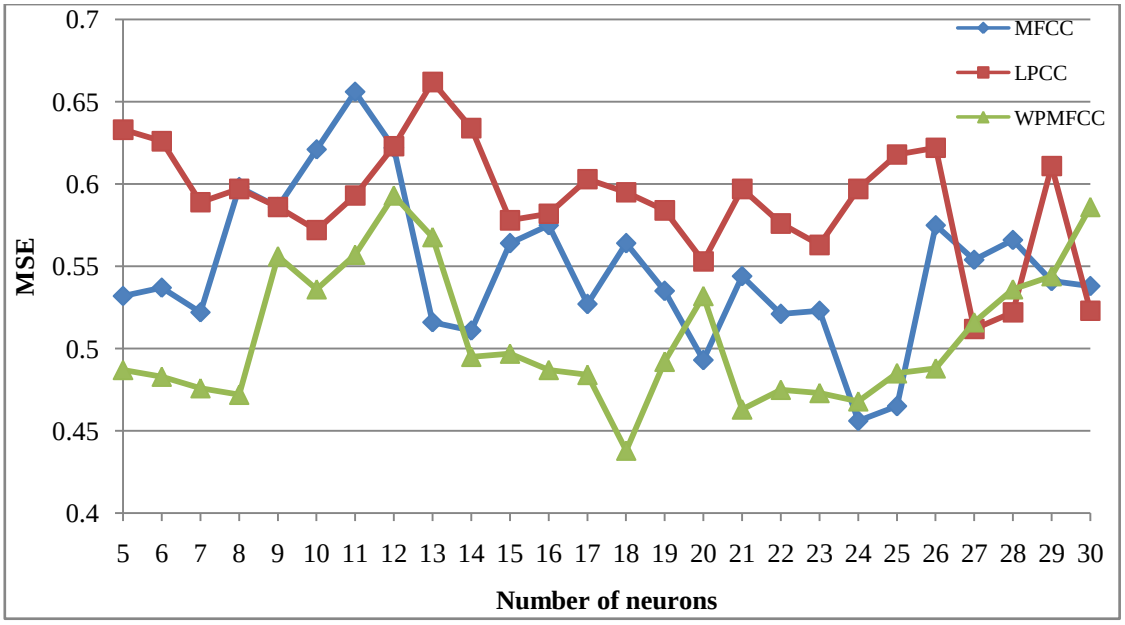


Figure 3.3: Variation in MSE with different number of neurons in the hidden layer for TI-20 database

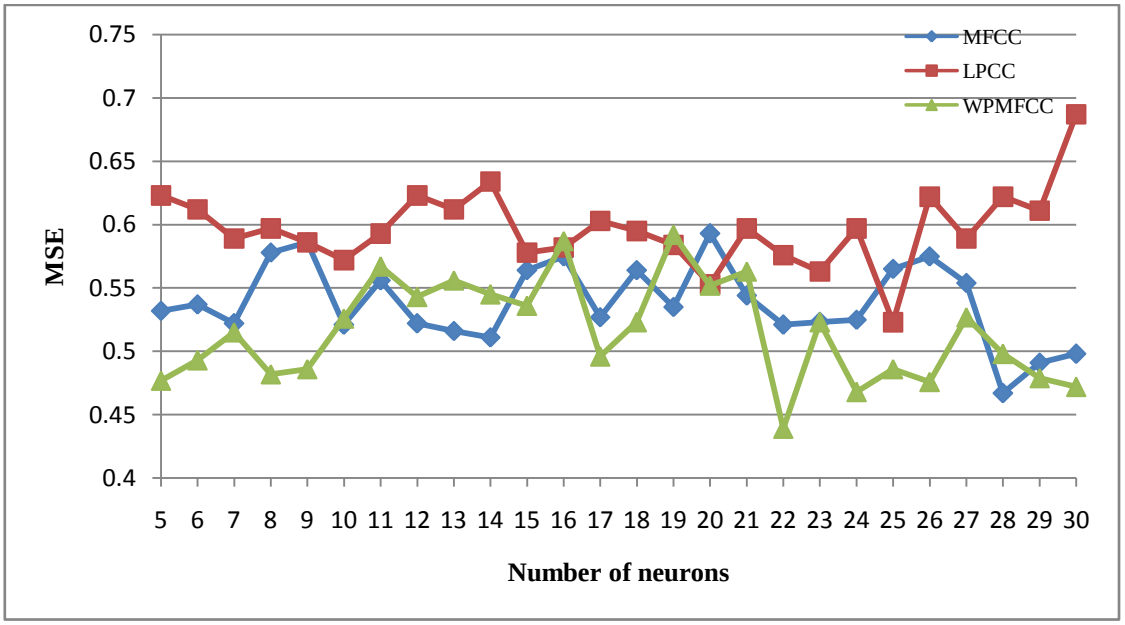


Figure 3.4: Variation in MSE with different number of neurons in the hidden layer for TI-ALPHA database

3.1.3 Results

The correlation coefficient (R) and MSE are two performance criteria that have been used to evaluate the performance of BP algorithm. The correlation coefficient measures the linear relation between expected output E_i and actual output A_i, while MSE measures the average squared error between the expected output and actual output. The MSE is computed using (3.6) and correlation coefficient is computed using:

$$R = \frac{\sum_{i=1}^{N}(E_i - \overline{E})(A_i - \overline{A})}{\sqrt{\sum_{i=1}^{N}(E_i - \overline{E})^2 \sum_{i=1}^{N}(A_i - \overline{A})^2}} \tag{3.7}$$

where $\overline{E}$ and $\overline{A}$ are average values of expected output and actual output, respectively.

The correlation coefficients and MSEs are computed for three databases considered in this work for LPCC, MFCC and WPMFCC features and are given in Table 3.2.

Table 3.2: The MSEs and correlation coefficients using ANN trained with BP algorithm

Database	Features	MSE	Correlation Coefficient
TI-20	LPCC	0.512	0.65
	MFCC	0.456	0.79
	WPMFCC	0.438	0.84
TI-ALPHA	LPCC	0.523	0.72
	MFCC	0.467	0.77
	WPMFCC	0.439	0.79
Hindi digits	LPCC	0.632	0.59
	MFCC	0.578	0.78
	WPMFCC	0.557	0.80

It is evident from Table 3.2 that for Hindi database, minimum MSE is 0.557 and the highest correlation coefficient is 0.80, obtained with WPMFCC features. For TI-20 database, minimum MSE achieved is 0.438 and the highest correlation coefficient is 0.84, obtained with WPMFCC features; and for TI-ALPHA database, minimum MSE and the highest correlation coefficient are 0.439 and 0.79, respectively, obtained with WPMFCC features.

3.2 EXPERIMENT 2: ANN TRAINED USING PROPOSED TECHNIQUE-I

This section presents the implementation of proposed Technique-I to optimize the weights and biases of ANN for speech recognition. The proposed Technique-I is a predator influenced civilized swarm optimization technique, formed by integrating civilized swarm optimization (CSO) and predator prey optimization (PPO) techniques. In this technique, predator always tries to chase the best solution in a controlled manner that maintains diversity in the

population and avoids local optimum solutions. The proposed Technique-I has been implemented on the speech databases used in Section 3.1.

3.2.1 Predator Influenced Civilized Swarm Optimization Technique

The predator influenced civilized swarm optimization (PCSO) is an integrating technique of CSO and PPO. The CSO is one of the potential global search techniques and is applied to solve various complex optimization problems (Selvakumar and Thanushkodi, 2009). The CSO is basically an integrated technique of society civilization algorithm (SCA) (Ray and Liew, 2003) and PSO (Kennedy and Eberhart, 1995).

In SCA, swarm is divided into civilized societies and every society has its own leader, society leader (SL). The SL is the best performing particle of the society and others particles are society members (SMs). The best performing SL is treated as civilization leader (CL). The SMs belonging to a particular society depend on their Euclidean distance in the parametric space and are computed as:

$$D_s = (\sum_{i=1}^{N} (SL_{is} - SM_{ir})^2)^{1/2},\ s = 1, 2, \ldots, N_s,\quad r = 1, 2, \ldots, N_r \tag{3.8}$$

The society member SM_r is assigned to society *'s'*, if it is closer to SL_s. Here N_s and N_r represent number of societies and society members, respectively and N represents the number of dimension.

In the proposed PCSO algorithm, predator particle is incorporated with civilized swarm. The predator always tries to chase CL, and it is difficult for SMs and CL to remain their favoured positions. The predator searches around CL in a concentrated manner, whereas swarm particles explore the search space escaping from predator. The effect of predator is managed through probability fear. The steps for PCSO procedure are as follows:

Step 1: The predator velocity $V_{Pi}(k)$ and position $X_{Pi}(k)$ at k^{th} iteration are updated as:

$$V_{Pi}(k+1) = C_p\left(CL_i(k) - X_{Pi}(k)\right),\ i = 1, 2, \ldots, N \tag{3.9}$$

$$X_{Pi}(k+1) = X_{Pi}(k) + V_{Pi}(k+1),\ i = 1, 2, \ldots, N \tag{3.10}$$

where C_p is uniform random number over (0, 1) and it decides how quickly predator particle can attack the CL; $CL_i(k)$ is civilization leader position at k^{th} iteration.

Step 2: The societies which do not have CL, update the velocity of society leaders $V_{is}^{SL}(k)$ by following CL and information acquired from their best positions as:

$$V_{is}^{SL}(k+1) = wV_{is}^{SL}(k) + C_{SL1}r_1(Pbest_{is}^{SL}(k) - SL_{is}(k)) + C_{SL2}r_2(CL_i(k) - SL_{is}(k)), \quad i=1, 2, \ldots, N, \quad s=1, 2, \ldots, N_s \tag{3.11}$$

where w is inertia weight and its value decreases from 0.9 to 0.4 with iteration; $C_{SL1}$and C_{SL2} are acceleration coefficients, which accelerate the SL towards its own best position and CL, respectively; r_1 and r_2 are uniformly distributed random numbers over (0,1); $Pbest_{is}^{SL}$ is personal best position of s^{th} SL; $SL_{is}(k)$ is s^{th} SL position at k^{th} iteration.

The velocity of society members $V_{ir}^{SM}(k)$ is updated by following corresponding SL and information acquire from their own best positions as:

$$V_{ir}^{SM}(k+1) = wV_{ir}^{SM}(k) + C_{SM1}r_3(Pbest_{ir}^{SM}(k) - SM_{ir}(k)) + C_{SM2}r_4(SL_{is}(k) - SM_{ir}(k)), \; i=1, 2, \ldots, N, \; r=1, 2, \ldots, N_r \tag{3.12}$$

where C_{SM1} and C_{SM2} are acceleration coefficients, which accelerate the SM towards its own best position and SL, respectively; r_3 and r_4 are uniformly distributed random numbers over (0, 1); $Pbest_{ir}^{SM}$ is personal best position of r^{th} SM; $SM_{ir}(k)$ is r^{th} SM position at k^{th} iteration.

Step 3: The society which contain CL particle, updates the velocity of civilized leader $V_i^{CL}(k)$ and society member $V_{ir}^{SM}(k)$ as:

For civilized leader:

$$V_i^{CL}(k+1) = \begin{Bmatrix} wV_i^{CL}(k) + C_{L1}r_5(Pbest_i^{CL}(k) - CL_i(k)) & pf < pf\max \\ wV_i^{CL}(k) + C_{L1}r_5(Pbest_i^{CL}(k) - CL_i(k)) + C_{L2}a_i\exp(-b_i d) & pf \geq pf\max \end{Bmatrix}, \; i=1, 2, \ldots, N, \; r=1, 2, \ldots, N_r \tag{3.13}$$

where $Pbest_i^{CL}(k)$ is personal best position of CL at k^{th} iteration; C_{L1} is acceleration coefficients, which accelerates the CL towards its own best position; C_{L2} is uniform random number over (0, 2), and it controls the predator influence on prey; a_i provides the maximum amplitude of the predator effect over SMs, and b_i controls the effect of the predator; d is the Euclidean distance between the predator and SMs; pf_{max} and pf are maximum probability fear and probability fear, respectively; pf is uniformly distributed random number over (0, 1); $r_5$is uniformly distributed random number over (0, 1).

For society members:

$$V_{ir}^{SM}(k+1) = \begin{cases} wV_{ir}^{SM}(k) + C_{SM1}r_3(Pbest_{ir}^{SM}(k) - SM_{ir}(k)) + C_{SM2}r_4(SL_{isr}(k) - SM_{ir}(k)); & pf < pf\max \\ wV_{ir}^{SM}(k) + C_{SM1}r_3(Pbest_{ir}^{SM}(k) - SM_{ir}(k)) + C_{SM2}r_4(SL_{isr}(k) - SM_{ir}(k)) + C_{SM3}a_i\exp(-b_i d); & pf \geq pf\max \end{cases}$$
$$i=1, 2, \ldots, N, \quad r=1, 2, \ldots, N_r \tag{3.14}$$

Step 4: The personal best positions $(Pbest)$ of CL, SLs and SMs are updated based on objective function evaluation and given as:

$$Pbest(k+1)=\begin{cases} X(k)+V(k+1); \varphi(Pbest(k+1)) < \varphi(X(k)) \\ Pbest(k); otherwise \end{cases} \quad (3.15)$$

where $\varphi(X(k))$ is objective function, evaluated at $X(k)$ position for k^{th} iteration; $X(k)$ and $V(k)$ are society particles position and velocity at k^{th} iteration, respectively.

Step 5: Formation of new swarm: Initially, swarm is taken as an empty set after that position of CL, SLs and SMs are updated and included to the new swarm. The positions are updated as:

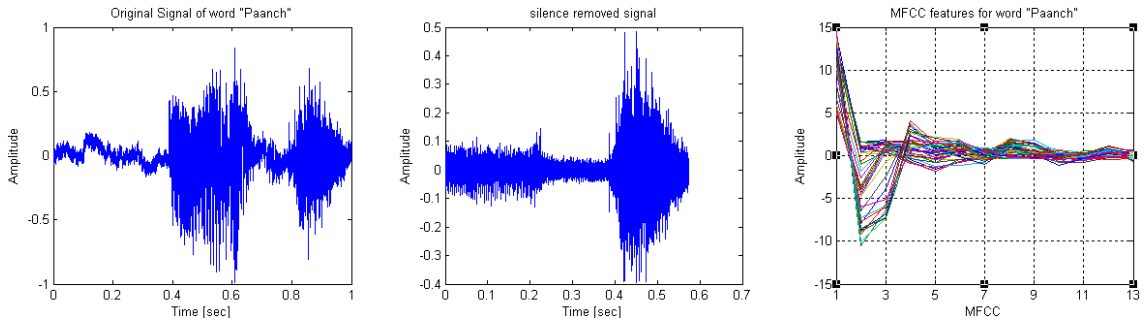
$$CL_i(k)=CL_i(k)+V_i^{CL}(k+1), \quad i=1, 2, \ldots, N \quad (3.16)$$

$$SL_{is}(k)=SL_{is}(k)+V_{is}^{SL}(k+1), \quad i=1, 2, \ldots, N, \quad s=1, 2, \ldots, N_s \quad (3.17)$$

$$SM_{ir}(k)=SM_{ir}(k)+V_{ir}^{SM}(k+1), \quad i=1, 2, \ldots, N, \quad r=1, 2, \ldots, N_r \quad (3.18)$$

3.2.2 Implementation

Initially, pre-processing of speech signal is required as discussed in sub-section 3.1.2. The acoustic features of speech signal are acquired by feature extraction technique as discussed in sub-section 3.1.2. The weights and biases of ANN are optimized by proposed Technique-I to minimize MSE and hence improving the speech recognition rate. Figure 3.5 shows the speech signal, silence removed signal and MFCC features extracted from that speech signal for Hindi word "*Paanch*". The extracted features are further given as input to ANN. The weights and biases of ANN are optimized using proposed Technique-I in order to minimize the MSE (Figure 3.6). The network is trained by using 10-fold cross validation approach.



(a) (b) (c)

Figure 3.5: Speech signal for Hindi word "*Paanch*"

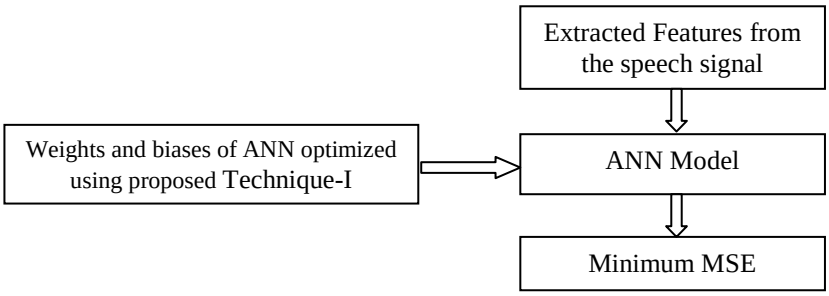


Figure 3.6: Speech recognition system using ANN trained with proposed Technique-I

A three layer ANN architecture having *n* input neurons, *m* hidden neurons and 1 output neuron has been undertaken in this experiment. For this model, the number of decision variables is computed as:

$$L = (n+1)m + (m+1) \tag{3.19}$$

$n \times m$ weights are used to connect input layer and hidden layer and *m* biases are used for hidden layer neurons. In the similar way, $m \times 1$ weights are used to connect hidden layer and output layer; and 1 bias is used for output layer neuron. All decision variables are set in the range (-1, 1).

The swarm having society particles and single predator is represented as:

$$Swarm = [[S^1]\ [S^2]\ \ldots [S^m]\ \ldots [S^{NP}]\ [PS]]$$

where the matrix $[S^m]$ represents the m^{th} particle of the swarm and it is defined as: $S^m = [W^m]$; where W^m represent weights and biases;$[PS]$ represents predator particle having same dimensions; *NP* represents number of society particles in a swarm.

The step-wise procedure for Technique-I based speech recognition system is described in Algorithm 3.1.

Algorithm 3.1: Proposed Technique-I for speech recognition

1. Read training data, expected outputs, parameters of algorithm, and set maximum number of iterations $k^{\max}$.
2. Randomly initialize ANN weights and biases as society and predator positions.
3. Initialize society and predator velocity randomly.
4. Initialize iteration index $k = 1$.
5. Compute net input as given by (3.4).
6. Compute the output by passing net input through the transfer function.
7. Compare the obtained and desired result; compute MSE by (3.6).
8. Arrange society particles on the basis of MSE, and best performing particle is selected as SL and remaining particles are treated as SMs.

9. The Euclidean distance between SMs and SL are computed by (3.8) and SMs are selected for a particular society.
10. Select CL among SLs on the basis of MSE.
11. Randomly generate probability fear.
12. Update predator velocity and position as given by (3.9) and (3.10), respectively.
13. For societies which do not have CL, update SL and SMs velocity by applying (3.11) and (3.12), respectively and their positions are updated by applying (3.16) and (3.18), respectively.
14. For society which has CL, update CL and SMs velocity by applying (3.13) and (3.14), respectively and their positions are updated by applying (3.16) and (3.18), respectively.
15. Update personal best positions as given by (3.15).
16. Repeat steps 4 to 15 until all training points are finished.
17. $k = k + 1$
18. IF $(k \leq k^{max})$THEN
    Select first training point and GOTO step 5
    ENDIF
19. STOP

3.2.2.1 Parameter setting for proposed Technique-I

In order to obtain more efficient results, parameters of a global search technique need to be adjusted. As such, for setting the parameter values of proposed Technique-I, 30 trials have been performed. In each trial run, the population size is set to 20 and one predator particle is undertaken. Parameters have been varied between minimum set values to maximum set values with a certain step size. Minimum and maximum values, step size and parameter values are given in Table 3.3. For each trial, maximum number of iterations is set to 100.

Table 3.3: Parameters range, step size and optimal value of parameters for proposed Technique-I

Parameter	Minimum value	Maximum value	Step size	Optimal value
N_s	2	10	1	4
$(C_{SL1}, C_{SL2}, C_{SM1}, C_{SM2}, C_{L1})$	0.25	2.0	0.25	(0.5, 0.5, 0.25, 0.75, 2.0)
C_{L2}	0.0	2.0	--	--
a_i	0.25	1.0	0.25	0.5
b_i	0.25	1.0	0.25	1.0
$pf_{\max}$	0.50	1.0	0.05	0.95

3.2.3 Results and Discussion

To select the optimum number of neurons in the hidden layer, several experiments have been conducted using proposed Technique-I to train the ANN model. The number of neurons at which the MSE has been found minimum for three databases with LPCC, MFCC and WPMFCC features is given in Table 3.4. The variations in MSE with iterations for ANN trained with proposed Technique-I is shown in Figure 3.7. The MSEs and correlation coefficients obtained using ANN trained with proposed Technique-I are given in Table 3.5. Figures 3.8-3.10 present the regression analysis between expected value (T) and actual value (Y) of output. Actual value of output has been obtained from ANN trained by proposed Technique-I for three databases using WPMFCC features. It is evident from Figures 3.8-3.10 that there are very small deviations between training, testing and validation performances. The correlation coefficients R for TI-20, TI-ALPHA and Hindi databases are 0.97, 0.94 and 0.91 respectively.

Table 3.4: Number of neurons in hidden layer for three databases with LPCC, MFCC and WPMFCC features

Database	Number of neurons in hidden layer		
	LPCC	MFCC	WPMFCC
Hindi digits	25	20	15
TI-20	15	15	25
TI-ALPHA	25	30	25

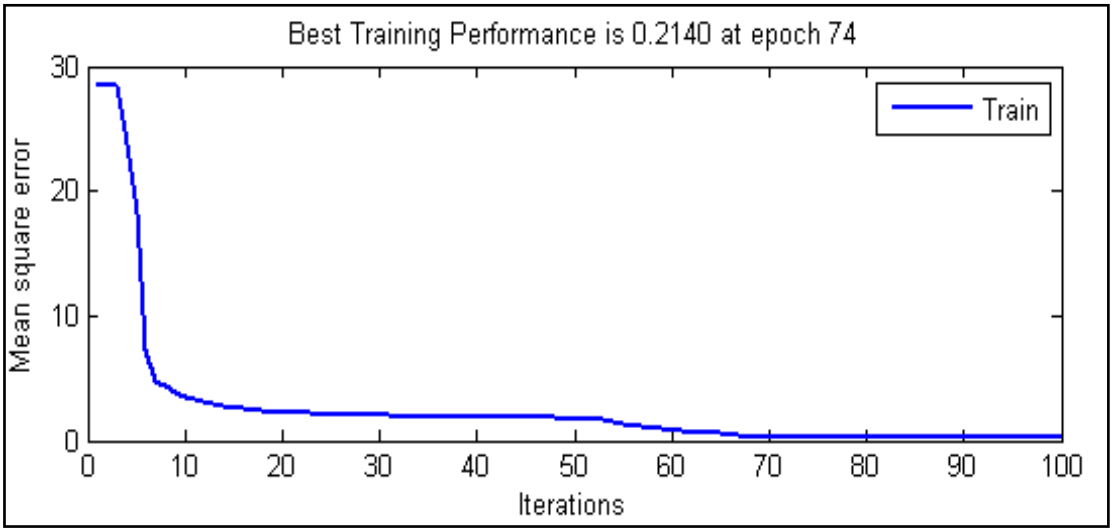


Figure 3.7: Variation in MSE with iterations obtained using ANN trained with proposed Technique-I

Table 3.5: The MSEs and correlation coefficients obtained by ANN trained with proposed Technique-I

Database	Features	MSE	Correlation Coefficient
TI-20	LPCC	0.256	0.92
	MFCC	0.174	0.96
	WPMFCC	0.125	0.97
TI-ALPHA	LPCC	0.236	0.89
	MFCC	0.178	0.91
	WPMFCC	0.246	0.95
Hindi digits	LPCC	0.432	0.79
	MFCC	0.416	0.91
	WPMFCC	0.347	0.92

To evaluate the robustness of the proposed Technique-I, a number of experiments have been conducted on noisy test samples. To obtain noisy test samples, artificially white noise is added, with a wide range of signal to noise ratio (SNR) from 0 to 40 dB in a step of 5 dB into the test samples of all the three databases. Figure 3.11 shows the variations in MSE with different SNRs using ANN trained with different algorithms for three databases with WPMFCC features.

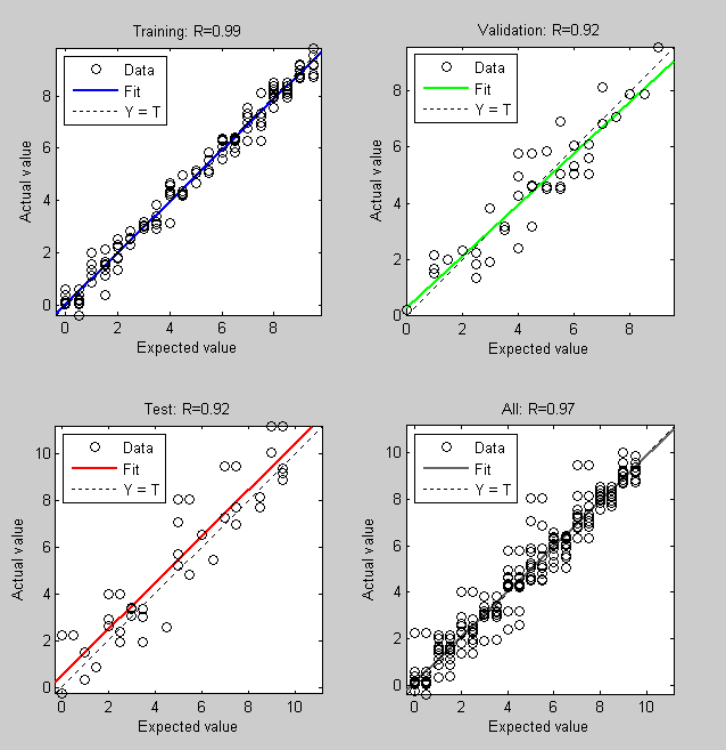


Figure 3.8: Regression plots for TI-20 database using Technique-I

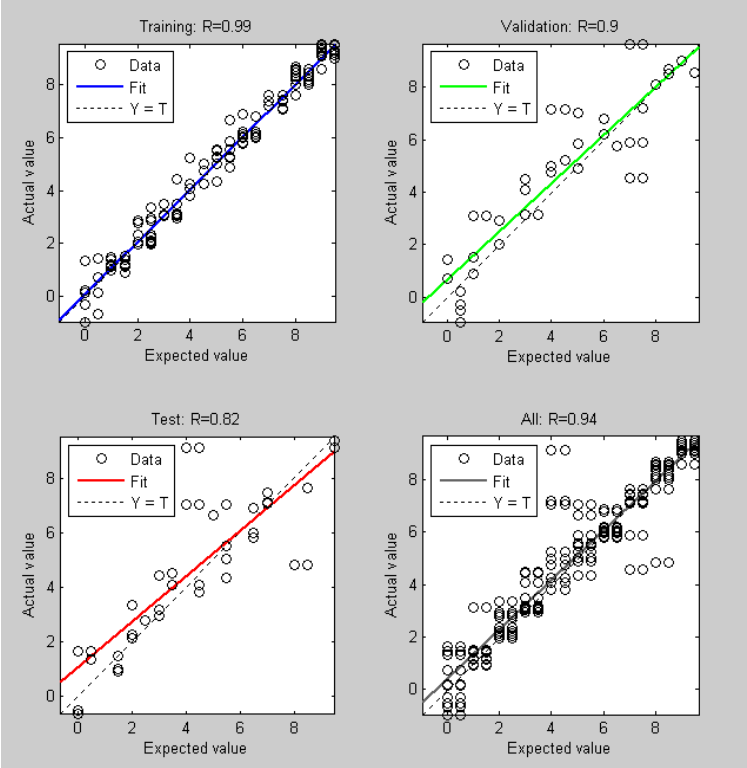


Figure 3.9: Regression plots for TI-ALPHA database using Technique-I

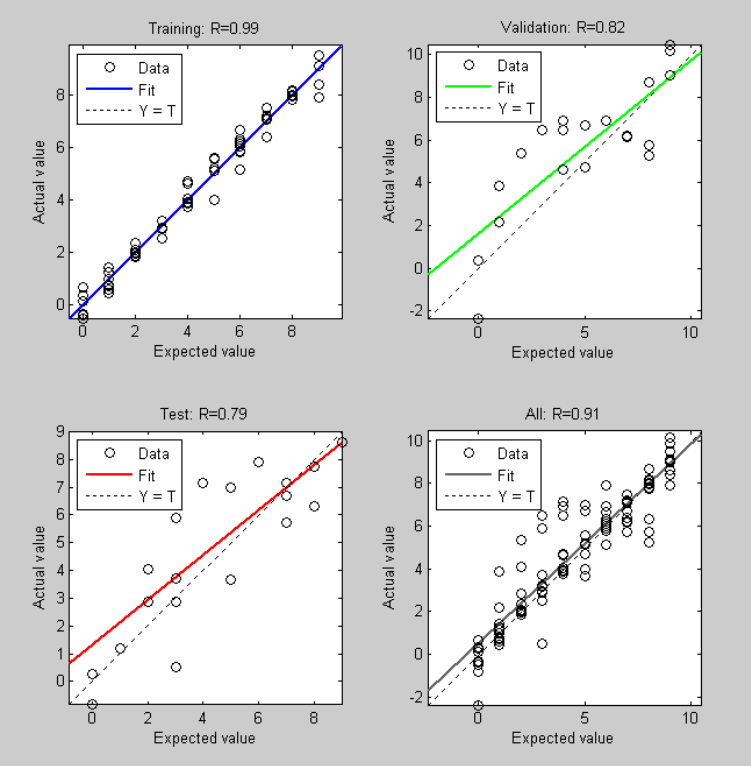


Figure 3.10: Regression plots for Hindi database using Technique-I

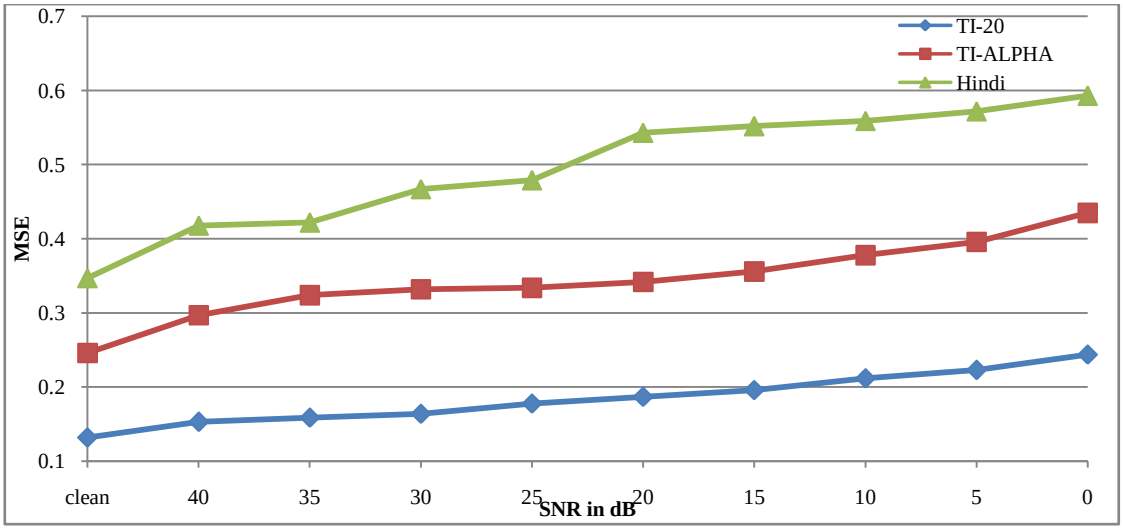


Figure 3.11: MSE versus SNR for three databases using ANN trained with proposed Technique-I

3.3 EXPERIMENT 3: ANN TRAINED USING PROPOSED TECHNIQUE-II

In this section, the results of the experiments that have been conducted for ANN training using Technique-II are presented. As mentioned earlier, Technique-II is an integration of PPO and Hooke-Jeeves method. Here, initial search is performed by global search PPO algorithm and after a set number of iterations; local search Hooke-Jeeves method is applied to avoid possible stagnation. The proposed Technique-II used for ANN training is tested on the speech databases as used in section 3.1. In the next sub-sections, a brief description of PPO technique and Hooke-Jeeves method is presented. The usage of these techniques in our work is reported in section 3.3.3.

3.3.1 Predator-Prey Optimization Technique

In the recent years, PSO has been successfully applied to solve various complex optimization problems. The PSO offered numerous advantages as compared to other global search techniques. In brief, the advantages of PSO are its easy implementation, better convergence characteristics as compared to other global search techniques. Still there are some unresolved issues of PSO are: parameter dependency, quality of solution depends on initial solutions, large computational times. One of the main problems is the diversity of particles. During the search process, particles may struck in to a local optimal solution for multimodal problems. The PSO researchers have proposed various modifications in basic PSO *i.e.* neighborhood topologies, parameter adaptation mechanism. In one of the attempt, Silva *et al.*(2002) have

proposed a model called predator prey model. In PPO model, search is performed by experience of prey particles along with consideration of predator particles. The predator chases the global best prey particle and global best particle searches another position to escape from predator. Prey particles search the optimum solution by sharing local and global best solution information and update their positions accordingly. In this process, prey particles efficiently explore the search area. The predator particle searches around global best prey particle and that helps to improve exploitation capability of search algorithm. To check the credibility of PPO, Silva *et al.* (2002) have tested it on multimodal functions. They have compared the obtained results with basic PSO and its other variants and concluded that PPO outperforms other algorithms in terms of solution quality and convergence characteristics. Recently, it has been used to solve various optimization problems (Narang *et al.*, 2012; 2014). The predator velocity V_{Pi}^{t} and position X_{Pi}^{t} are updated as:

$$V_{Pi}^{t+1} = C_4\left(Gbest_i^t - X_{Pi}^t\right),\ i=1,\ 2,\ \ldots,n \tag{3.20}$$

$$X_{Pi}^{t+1} = X_{Pi}^t + V_{Pi}^t\ ,\ i=1,\ 2,\ \ldots,n \tag{3.21}$$

where'n' represents number of dimensions; C_4 is uniform random number over (0, 1) and it controls the velocity of predator; $Gbest_i^t$ is global best prey position for i^{th} dimension, at t^{th} iteration.

The probability fear controls the effect of predator on prey particles. The velocity V_{il}^{t} and position X_{il}^{t} of prey particles are updated as:

$$V_{il}^{t+1} = \begin{cases} w\times V_{il}^t + C_1\left(Lbest_{il}^t - X_{il}^t\right)\times r_6 + C_2\left(Gbest_i^t - X_{il}^t\right)\times r_7 & pf < pf\max \\ w\times V_{il}^t + C_1\left(Lbest_{il}^t - X_{il}^t\right)\times r_6 + C_2\left(Gbest_i^t - X_{il}^t\right)\times r_7 + C_3 a_i \exp\left(-b_i d\right) & pf \geq pf\max \end{cases}$$

$$l=1,\ 2,\ \ldots,NP,\ i=1,\ 2,\ \ldots,n \tag{3.22}$$

$$X_{il}^{t+1} = X_{il}^t + V_{il}^t\ \ ,\ \ l=1,\ 2,\ \ldots,NP,\ \ \ i=1,\ 2,\ \ldots,n \tag{3.23}$$

where NP denotes number of swarm prey particles; constant 'a_i' represents predator maximum amplitude effect over prey; and 'b_i' represents effect of predator over prey; $C_1$and $C_2$are the acceleration constant; C_3 is uniform random number over (0, 1) and it controls the predator effect on prey; parameter w is inertia weight; $Lbest_{il}^t$ is local best prey position for i^{th} dimension of l^{th} particle at t^{th} iteration; $Gbest_i^t$ is global best prey position for i^{th} dimension, at t^{th} iteration; $pf\max$ is maximum probability fear; d is Euclidean distance between predator and prey; r_6 and r_7 are uniformly distributed random number over (0, 1).

3.3.2 Hooke-Jeeves Method

Hooke-Jeeves method (Rao, 1996) is a pattern search method and it does not require any information on derivative of fitness function and constraints. In this method, initial search is performed by exploratory move in which trail solution x_i^0 is perturbed in both directions and maximum fitness function is computed as:

$$f_{\max} = \max\begin{Bmatrix} f(x_1, x_2, ..., x_j, ..., x_n) \\ f^+(x_1, x_2, ..., x_j + \Delta, ..., x_n) \\ f^-(x_1, x_2, ..., x_j - \Delta, ..., x_n) \end{Bmatrix} \quad (3.24)$$

If the trail solution is updated then exploratory move is executed successfully. For the next cycle of optimization, the decision variable is updated by pattern move. The pattern search direction *S* is computed by using two successive points obtained from exploratory move and the decision variable is updated as:

$$x_i^{k+1} = x_i^k + \eta \times S \quad (3.25)$$

and

$$S = x_i^k - x_i^{k-1} \quad (3.26)$$

where k is the index of pattern search direction and η is the scaling factor.

The procedure continues until the termination criterion is met. The stepwise procedure of Hooke-Jeeves method is given in Algorithm 3.2.

Algorithm 3.2: Hooke-Jeeves method

1. Set step size $\Delta = 1$, scaling factor $\alpha = 1$ and tolerance band $\varepsilon = 0.2$.
2. Set decision variable counter $j = 1$.
3. Set pattern search counter $k = 1$.
4. Start with decision variable $x_i^k = x_i^{k-1}$ and compute fitness f, $f_{\max}^{k-1} = f$.
5. $\Delta = \Delta / \alpha$
6. Compute the maximum value of fitness function $f_{\max}^k$ as given by (3.24).
7. IF $(f_{\max}^k > f_{\max}^{k-1})$ THEN
   Update fitness function $f_{\max}^{k-1} = f_{\max}^k$ and decision variable x_i^{k-1}.
   ENDIF
8. $j = j + 1$
9. IF $(j \le n)$ THEN
   GOTO step 4.
   ENDIF
10. IF $(x_i^k \ne x_i^{k-1})$ THEN
    GOTO step 12.

ENDIF

11. IF $(\Delta > \varepsilon)$ THEN
    $\alpha = 2\alpha$
    GOTO step 2.
    ELSE
    STOP
    ENDIF
12. Set $f_{\max}^{k} = f_{\max}^{k-1}$
13. The pattern search direction is computed as given by (3.26).
14. Update the decision variable as given by (3.25).
15. Compute the fitness function $f_{\max}^{k+1}$.
16. IF $(f_{\max}^{k+1} > f_{\max}^{k})$ THEN
    Update decision variable.
    $k = k + 1$
    GOTO step 12.
    ELSE
    $x_i^{k-1} = x_i^k$
    GOTO step 11.
    ENDIF

3.3.3 Implementation

The extracted acoustic features are given as input to the ANN classification module. A feed-forward ANN model consisting of input layer, one hidden layer and output layer, has been undertaken. The network is trained by using 10-fold cross validation approach. The implementation of Technique-II to minimize MSE is explained as under.

The swarm having prey particles and single predator is represented as:

$$Swarm = [[P^1]\ [P^2]\ \ldots [P^m]\ \ldots [P^{NP}]\ [PS]]$$

where the matrix $[P^m]$ represents the m^{th} prey particle of the swarm and it is defined as: $P^m = [W^m]$; where W^m represents weights and biases;$[PS]$ represents predator particle having same dimensions; *NP* represents number of prey particles in a swarm.

The optimization process continues till the termination criterion is fulfilled. The stepwise procedure to implement Technique-II is given in Algorithm 3.3.

Algorithm 3.3: Proposed Technique-II for speech recognition

1. Read training data, expected outputs and parameters of algorithm.
2. Randomly initialize ANN weights and biases as prey position population and predator position.
3. Randomly initialize prey and predator velocity.
4. $k = 1$
5. Compute net input as given by (3.4).

6. Compute the output by passing net input through the transfer function.
7. Compare the actual output with the expected output and compute MSE as given by (3.6).
8. Select local and global best prey positions on the basis of MSE.
9. Randomly generate probability fear (pf).
10. Update predator velocity and position as given by (3.20) and (3.21), respectively.
11. Update prey velocity and position as given by (3.22) and (3.23), respectively.
12. Update local prey positions.
13. IF $\{MOD(k, 10) = 0\}$ THEN

    Apply Hooke-Jeeves method to modify local best prey positions.

    If modified local best position improves the solution then update local best positions.

    ENDIF
14. Update global best position.
15. Repeat steps 5 to 14 until all training points are finished.
16. $k = k + 1$
17. IF $(k \le k^{\max})$ THEN

    Select first training point and GOTO step 5.

    ENDIF
18. STOP

3.3.3.1 Parameter setting for proposed Technique-II

A number of trials have been executed to set various parameter values to obtain optimum value of MSE. In each trial, the prey population size is set to 20 and one predator is undertaken. Parameters have been varied between minimum set values to maximum set values with a certain step size. Maximum number of iterations is set to 200. Minimum values, maximum values and step size of these parameters are given in Table 3.6. Another parameter inertia weight w of PPO has been linearly decreasing with iterations from 0.9 to 0.4.

Table 3.6: Parameter range, step size and optimal value for proposed Technique-II

Parameter	Minimum Value	Maximum Value	Step Size	Optimal Value
C_1	1.0	2.5	0.50	2
C_2	1.0	2.5	0.50	2
a_i	0.05	0.2	0.05	0.1
b_i	0.50	2.0	0.50	1
$pf_{\max}$	0.10	0.95	0.05	0.95

The results achieved by ANN with Technique-II are compared with results obtained by ANN trained with BP, PSO, PPO, CSO, PSO with Hooke-Jeeves and Technique-I. To set the

optimum parameters of these techniques, the procedure as applied with Technique-II is considered. The optimum parameters for proposed technique-I are given in sub-section 3.2.2.1.

The optimum parameters of other algorithms are:

- PSO: acceleration constants C_1= 1.5, and C_2 = 2;
- PPO: C_1=1.5, C_2 =2, maximum value of C_3 =2.0, a_i =0.25, b_i=0.75, and pf_{max} =0.95;
- CSO: N_s =5, C_{SL1}= 0.25, C_{SL2} =0.5, C_{SM1}= 0.25, C_{SM2} =0.75, and C_{L1}= 1.75.

3.3.4 Results and Discussion

In this work, to select the optimum number of neurons in the hidden layer, several experiments have been conducted using Technique-II to train the ANN model. The number of neurons at which the MSE has been found minimum for three databases with LPCC, MFCC and WPMFCC features is given in Table 3.7.

Table 3.7: Number of neurons in hidden layer for three databases with LPCC, MFCC and WPMFCC features

Database	Number of neurons in hidden layer		
	LPCC	MFCC	WPMFCC
Hindi digits	20	10	5
TI-20	30	25	20
TI-ALPHA	25	25	15

To compare the performance of ANN trained with proposed techniques and ANN trained with other optimization algorithms, MSEs and correlation coefficients are evaluated and are presented in Table 3.8 and Table 3.11, respectively. It is evident from Table 3.8 that MSE obtained using ANN trained with Technique-II is less than MSE obtained using ANN trained with BP, PSO, PPO, PSO with Hooke-Jeeves and CSO algorithms for all databases considered in this work and when LPCC, MFCC and WPMFCC features are used. The MSE obtained using ANN trained with Technique-I outperforms the results obtained by Technique-II in some cases. Another important observation is that ANN trained with PSO and Hooke-Jeeves method has produced a satisfactorily result and sometimes better than Technique-I. It is due to excellent search capability of local search Hooke-Jeeves method.

Table 3.8: Comparison of MSEs obtained by different training algorithms for ANN

Database	Features	MSE						
		BP	PSO	PPO	PSO and Hooke-Jeeves	CSO	Proposed Technique-I	Proposed Technique-II
TI-20	LPCC	0.512	0.322	0.302	0.277	0.289	0.256	0.242
	MFCC	0.456	0.316	0.286	0.218	0.275	0.174	0.186
	WPMFCC	0.438	0.236	0.183	0.122	0.168	0.125	**0.097**
TI-ALPHA	LPCC	0.523	0.376	0.317	0.281	0.312	0.236	0.240
	MFCC	0.467	0.312	0.293	0.254	0.284	0.178	0.192
	WPMFCC	0.439	0.310	0.279	0.232	0.267	0.246	**0.112**
Hindi digits	LPCC	0.632	0.557	0.511	0.467	0.510	0.432	0.347
	MFCC	0.578	0.518	0.431	0.401	0.427	0.416	0.386
	WPMFCC	0.557	0.493	0.412	0.389	0.404	0.347	**0.235**

Statistical tests have been proposed by researchers to compare the performance of two algorithms. Pair-wise comparison tests are generally performed to verify an experimental study. Sometimes, the observations from an experimental study are not normally distributed. To deal with such observations, non-parametric tests are performed. One of the promising nonparametric test is Wilcoxon signed rank test (Deibold and Mariano, 1995; Demsar, 2006; Garca and Herrera, 2009). It is a simple, yet safe and robust test (Derrac *et al.*, 2011). By performing, the Wilcoxon signed rank test, one can judge whether two samples belong two different populations or not. The Wilcoxon signed rank test helps to detect significant differences between two samples means.

In this work, to detect the differences between two samples means for considered optimization techniques, Wilcoxon signed rank test has been performed. Each algorithm is run thirty times. In Table 3.9, performance of proposed Technique-I is compared with PSO with Hooke-Jeeves method, CSO, PPO and PSO techniques for TI-20 database by considering WPMFCC features at the level of significance $\alpha = 0.01$. From Table 3.9, it can be observed that there is no significant difference between the results obtained by proposed Technique-I when compared with, PSO with Hooke-Jeeves method, while proposed Technique-I is significantly better than CSO, PPO and PSO techniques. In Table 3.10, the performance of proposed Technique-II is compared with proposed Technique-I, CSO, PSO with Hooke-Jeeves method and PPO techniques for TI-20 database by considering WPMFCC

features at a level of significance α=0.01. It is evident from this table that proposed Technique-II is significantly better than other considered techniques.

Table 3.9: *p*-values for Wilcoxon signed rank test when applied to Proposed Technique-I; PSO with Hooke-Jeeves method; CSO; PPO and PSO

	Proposed Technique-I versus PSO with HJ method	Proposed Technique-I versus CSO	Proposed Technique-I versus PPO	Proposed Technique-I versus PSO
p-value	**0.881060**	0.000130	0.000088	0.000087

Table 3.10: *p*-values for Wilcoxon signed rank test when applied toProposed Technique-II; Proposed Technique-I; PSO with Hooke-Jeeves method; CSO; PPO and PSO

	Proposed Technique-II versus Proposed Technique -I	Proposed Technique-II versus PSO with Hooke-Jeeves method	Proposed Technique-II versus CSO	Proposed Technique-II versus PPO	Proposed Technique-II versus PSO
p-value	0.004840	0.003170	0.000088	0.000088	0.000088

The comparison of correlation coefficients is presented in Table 3.11 and it is observed that *R* obtained by ANN with Technique-II is better than *R* obtained by ANN trained with BP, PSO, PPO, CSO, PSO with Hooke-Jeeves method, and proposed Technique-I for the databases considered in this work and when LPCC, MFCC and WPMFCC features are used.The regression analysis for three databases with WPMFCC features obtained from ANN trained with Technique-II is presented in Figures 3.12-3.14. It can be observed from these figures that training, testing and validation performances have very small deviations. The best values of correlation coefficients for TI-20, TI-ALPHA and Hindi databases are 0.97, 0.96 and 0.95, respectively when Technique-II is used for ANN training with WPMFCC features.

To further establish the robustness of Technique-II, a series of experiments have been conducted by considering noisy speech test samples. The SNR has been varied from 0 to 40 dB in step of 5 dB into the test samples of all three databases. The MSEs are presented in Figures 3.15-3.17 for different values of SNR obtained by ANN optimized with various techniques with WPMFCC features. It has been observed from these figures that MSEs obtained using ANN trained with Technique-II is less than the MSEs obtained using ANN trained with other algorithms for all the three databases considered in this work.

Table 3.11: Comparison of correlation coefficients obtained using ANN trained with different algorithms

Database	Features	Correlation Coefficient (R)						
		BP	PSO	PPO	PSO with Hooke-Jeeves	CSO	Proposed Technique-I	Proposed Technique-II
TI-20	LPCC	0.65	0.82	0.86	0.87	0.90	0.92	0.93
	MFCC	0.79	0.86	0.89	0.92	0.92	0.96	0.96
	WPMFCC	0.84	0.91	0.94	0.93	0.95	0.97	**0.97**
TI-ALPHA	LPCC	0.72	0.80	0.82	0.86	0.84	0.89	0.91
	MFCC	0.77	0.83	0.85	0.89	0.88	0.91	0.93
	WPMFCC	0.79	0.87	0.91	0.92	0.93	0.95	**0.96**
Hindi digits	LPCC	0.59	0.63	0.68	0.74	0.77	0.79	0.83
	MFCC	0.78	0.81	0.86	0.88	0.89	0.91	0.92
	WPMFCC	0.80	0.85	0.88	0.89	0.91	0.92	**0.95**

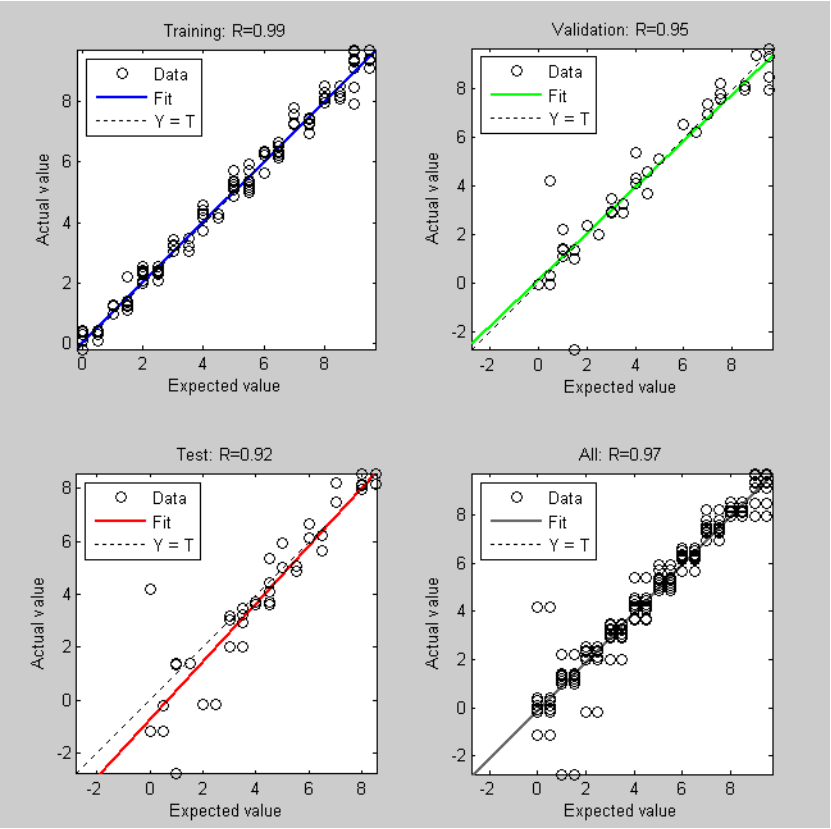


Figure 3.12: Regression plots for TI-20 database using Technique-II

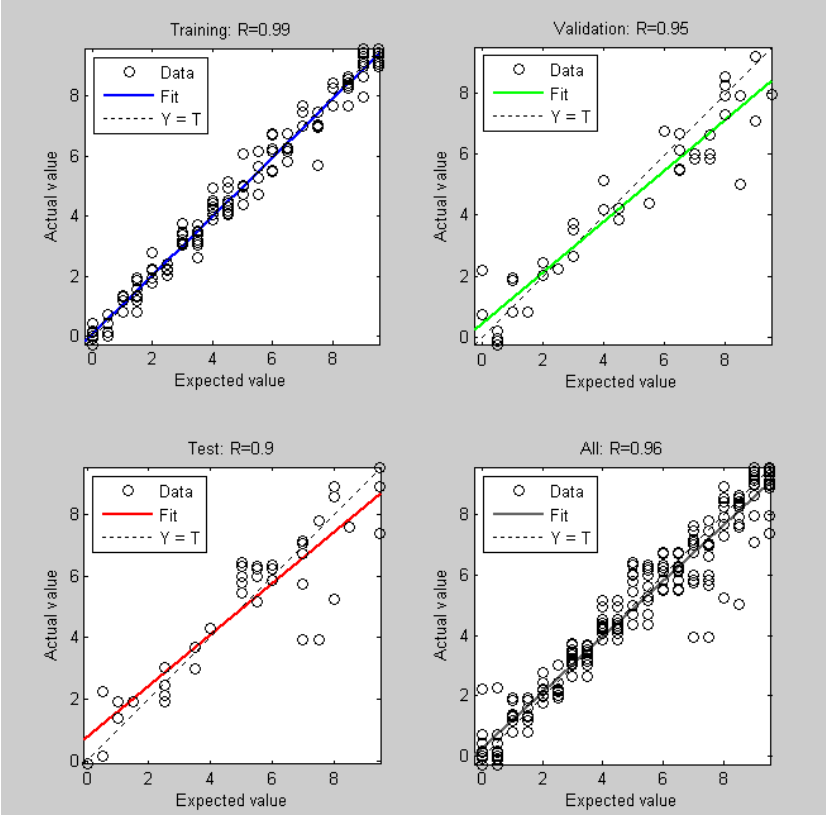


Figure 3.13: Regression plots for TI-ALPHA database using Technique-II

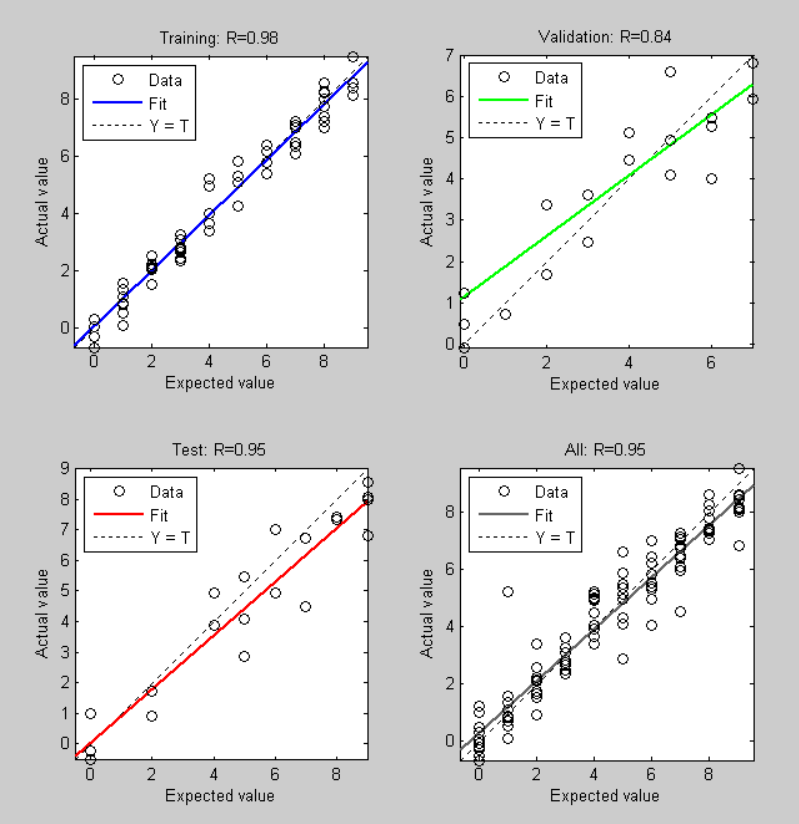


Figure 3.14: Regression plots for Hindi database using Technique-II

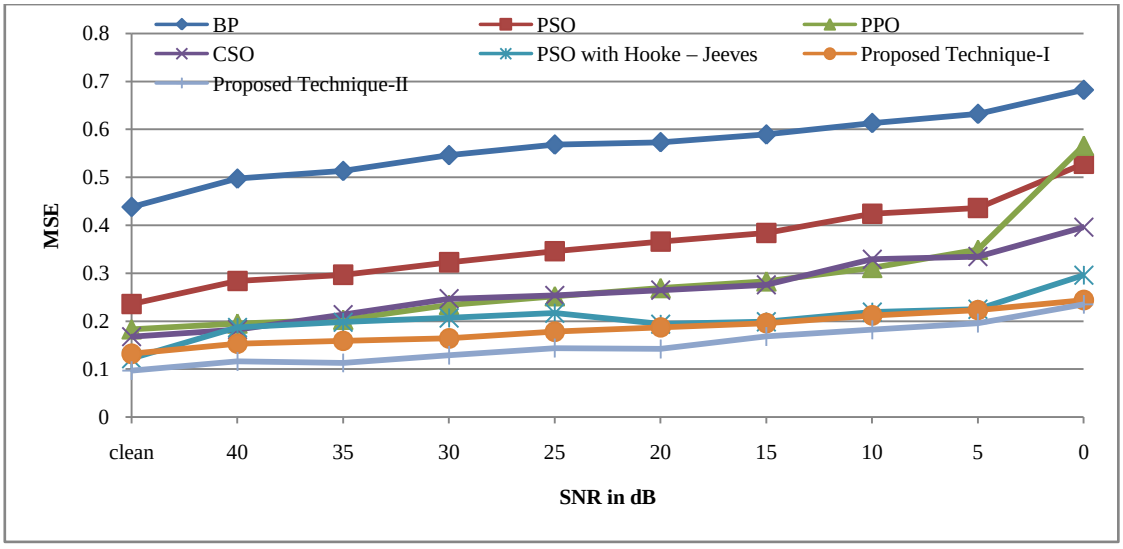


Figure 3.15: MSE versus SNR for TI-20 database with different training algorithms for ANN

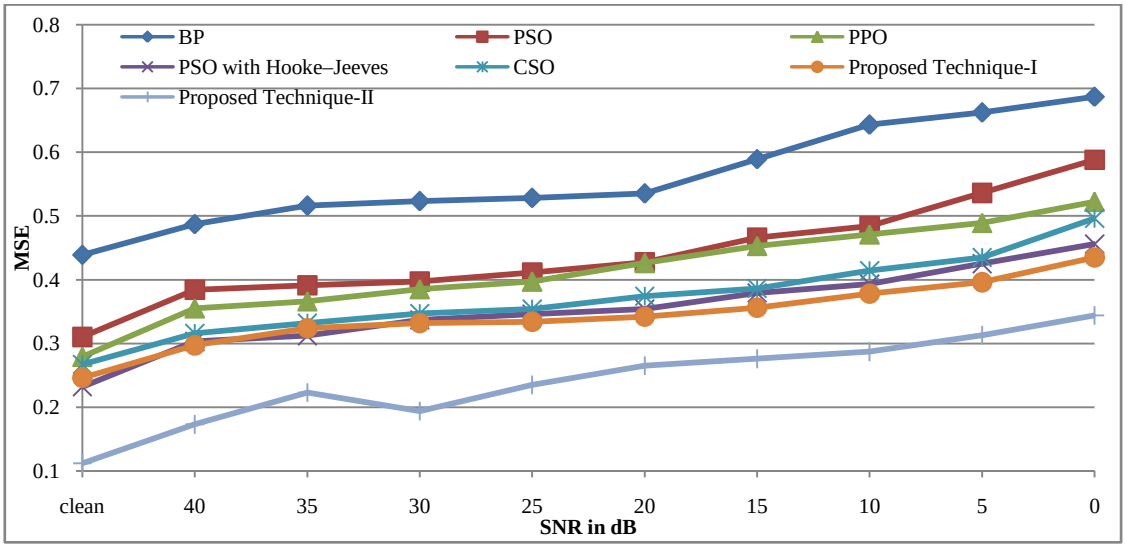


Figure 3.16: MSE versus SNR for TI-ALPHA database with different training algorithms for ANN

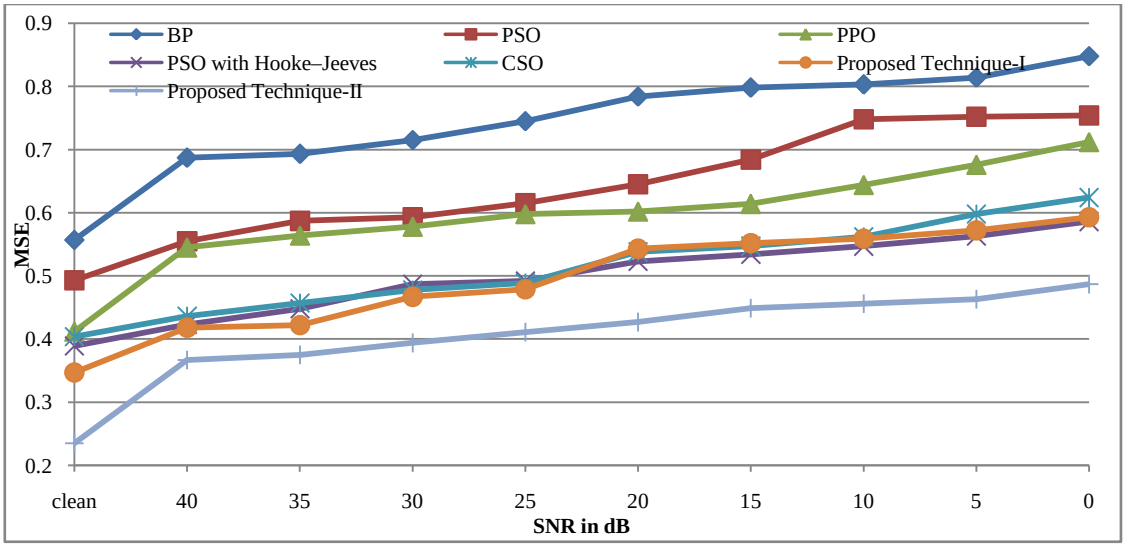


Figure 3.17: MSE versus SNR for Hindi database with different training algorithms for ANN

3.4 CONCLUSIONS

In this chapter, two optimization techniques have been proposed to train weights and biases of ANN. The ANN trained by these techniques is applied to improve the speech recognition rate. The proposed Technique-I is based on integration of CSO and PPO techniques. In this technique, predator particle chases the CL that includes an additional capability to escape from local optimum solution. The predator exploits the search around CL, whereas the society particles explore the solution space escaping from predator, so society particles play the role of diversification and predator particle exploits the search space. The proposed Technique-II is based on integration of PPO and Hooke-Jeeves method. In PPO algorithm, interaction between predator and prey particles improves the exploration and exploitation capability of search algorithm. The local best solutions obtained from PPO technique are further exploited by applying Hooke-Jeeves method to improve the search capability of proposed Technique-II. The experiments have been implemented on TI-20, TI-ALPHA clean speech word databases, and Hindi numerals database. For these speech databases, LPCC, MFCC and WPMFCC features have been explored. Two performance criteria MSE and correlation coefficient have been employed to test the validity of proposed techniques. After comparing the MSEs and correlation coefficients, it has been found that results obtained by ANN trained with proposed techniques outperforms the results obtained by ANN trained with PSO, PPO, CSO and BP algorithms. It has also been observed that proposed Technique-II achieves better results than proposed Technique-I in most of the cases because of excellent exploitation capability of Hooke-Jeeves method used in this technique. To further evaluate the robustness of proposed techniques, a white noise is added under a wide range of SNR in the test samples and MSE is computed. Finally, it has been concluded that MSEs obtained using ANN trained with Technique-II is less than the MSEs obtained using ANN trained with other algorithms for all the three databases considered in this work.

Chapter 4

Recognition of Isolated Words using Optimized SVM Classifier

Support vector machine (SVM) has been proved very promising learning machine for pattern recognition problems. This method has successfully been applied by several researchers in various fields like detection, verification, and recognition of faces, objects, handwritten characters, speech, *etc*. In this chapter, SVM classifier has been explored for speech recognition purpose. Three experiments have been covered in this chapter. In the first experiment, speaker-dependent, isolated words recognition system using one-vs-all SVM classifier has been developed. The linear, polynomial and RBF kernels with default values of their parameters are used for the construction of SVM for speech recognition. The effect of dynamic frame size for feature extraction has also been exploited for the three SVM kernels. In the second experiment, the emphasis is given to the application of optimization technique with SVM classifier for speech recognition. SVM has a good application prospects for speech recognition problems; still optimum parameter selection of SVM kernels is a vital issue for it. In this chapter, SVM hyper-parameters (RBF kernel parameter and penalty parameter) are searched by applying PPO with Hooke-Jeeves method. In speech recognition system, size of the speech feature set is an important aspect as it affects recognition accuracy, computational time, memory requirement and thus complexity of the model, so there is a need to choose an appropriate speech feature set. The third experiment has been conducted to select most appropriate speech feature set and appropriate SVM hyper-parameters. The selection of speech feature set is binary in nature and hyper-parameters are continuous in nature. So, an optimization technique is needed to deal with these two types of decision variables. In this chapter, a binary PPO technique is proposed and integrated with Hooke-Jeeves method. This technique along with technique proposed in Section 3.3 has been used to deal with these mixed type of variables. This proposed technique, mixed-variable PPO with Hooke-Jeeves method (MVPPO-HJ), along with RBF kernel is applied to recognize TI-46 standard speech database and Hindi words database. The experimental results obtained using SVM with

MVPPO-HJ confirm improved recognition rate. Further, ROC curve is analyzed to verify sensitivity and specificity of the results obtained by SVM with MVPPO-HJ technique.

4.1 EXPERIMENT 1: SPEECH RECOGNITION USING SVM

The implementation of a speech recognition system using SVM is presented in this section. SVM is a binary classifier, in which input space is mapped into a higher dimensional feature space. In SVM, structural risk minimization is used to construct an optimal separating hyper plane to maximize the margin between two classes. The optimal hyper plane is constructed by minimizing regularized training error as:

$$E=\frac{1}{2}\|w\|^2+C\sum_{i=1}^{n}\xi_i$$

subject to $y_i\left(\langle w, x_i\rangle+b\right)\geq 1-\xi_i,\ \xi_i\geq 0,\ i=1, 2, ..., n$ (4.1)

where $\langle,\rangle$ denotes the inner product, ξ_i is a slack variable, which defines the permitted misclassification error, C is the penalty coefficient and it defines the trade-off between the empirical risk and the regularization term.

Three SVM kernels, *i.e.*, linear, polynomial and RBF, have been undertaken for this work and these are given as (Hwang and Kim, 2012):

Linear kernel:

$$K(x_i, x_j)=\langle x_i, x_j\rangle \quad (4.2)$$

Polynomial kernel:

$$K(x_i, x_j)=\left(\gamma\langle x_i, x_j\rangle+r\right)^d,\ \ \gamma>0 \quad (4.3)$$

Radial basis function (RBF) kernel:

$$K(x_i, x_j)=\exp\left(-\gamma\|x_i-x_j\|^2\right),\ \gamma>0 \quad (4.4)$$

whereγ, r and d are kernel parameters.

The detail discussion related to SVM is given in Section 1.2.4.3. The highlight of this experiment is to explore the effect of dynamic size frame on speech recognition rate. The speech databases used for experimentation, the implementation details and the results obtained are discussed in further sub-sections.

4.1.1 Speech Database

In this work, as mentioned earlier, a benchmark database TI-46 (NIST Speech DB, 1991) and a self recorded isolated Hindi words database have been considered. The details of TI-46 (TI-20 and TI-ALPHA subsets) database have been discussed in Section 3.1.1 of Chapter 3. The Hindi database consists of 20 words with 50 utterances of each word spoken by a single female speaker. The vocabulary of Hindi database is given in Table 4.1. The database was recorded at 44.1 KHz in room conditions.

Table 4.1: Vocabulary used for Hindi database

S. No.	Word	S.No.	Word	S.No.	Word	S.No.	Word
(i)	shoonya	(vi)	Paanch	(xi)	gaana	(xvi)	paani
(ii)	ek	(vii)	Cheh	(xii)	dekho	(xvii)	kaam
(iii)	do	(viii)	Saat	(xiii)	baitho	(xviii)	anek
(iv)	teen	(ix)	Aath	(xiv)	khaao	(xix)	achaar
(v)	chaar	(x)	Nau	(xv)	raam	(xx)	jagat

4.1.2 Fitness Function

Normally, classification problems have positive and negative classes (Bao *et al.*, 2013). The confusion matrix usually divides the classification points into four categories *i.e.* true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). Table 4.2 represents the confusion matrix that illustrates relationship among these indices. The performance of a classifier is evaluated by its accuracy, which is defined on the basis of four categories of the confusion matrix as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.5}$$

Table 4.2: Confusion matrix

	P	N
P	True Positive	False Positive
N	False Negative	True Negative

4.1.3 Implementation

Initially, the pre-processing of speech signal has been done as discussed in Section 3.2.1. After pre-processing, LPCC, MFCC and WPMFCC features have been extracted. To obtain the LPCC and MFCC features, speech signal is divided into frames with 50% superposition. In this work, following Cerf and Compernolle (1994); and Tan and Lindberg (2010), dynamic size frames have been used to overcome the disadvantage with fixed size frames. To observe the influence of the choice of number of frames on the recognition rate, a study has been carried out with different values of frames while keeping other parameters fixed. The speech signal is divided into varying number of frames starting from 25 up to 50 in a step of 5. Here, 13 coefficients per frame are extracted, resulting into 325, 390, 455, 520, 585 and 650 features for the number of frames, respectively. The feature extraction procedure, as given in Section 3.2.2, has been used in this study. The WPMFCC features have also been extracted as discussed in Section 3.2.2.

The extracted features are given as input to the SVM. One-versus-all SVM (Figure 4.1) with linear, polynomial and RBF kernels is used for classification. The number of one-versus-all SVM classifiers used for Hindi, TI-20 and TI-APLHA databases are 20, 20 and 26, respectively. Each SVM classifier is separately trained by using LPCCs, MFCCs and WPMFCCs. In each case, the SVMs are trained by 10-fold cross validation approach (Stone, 1974).

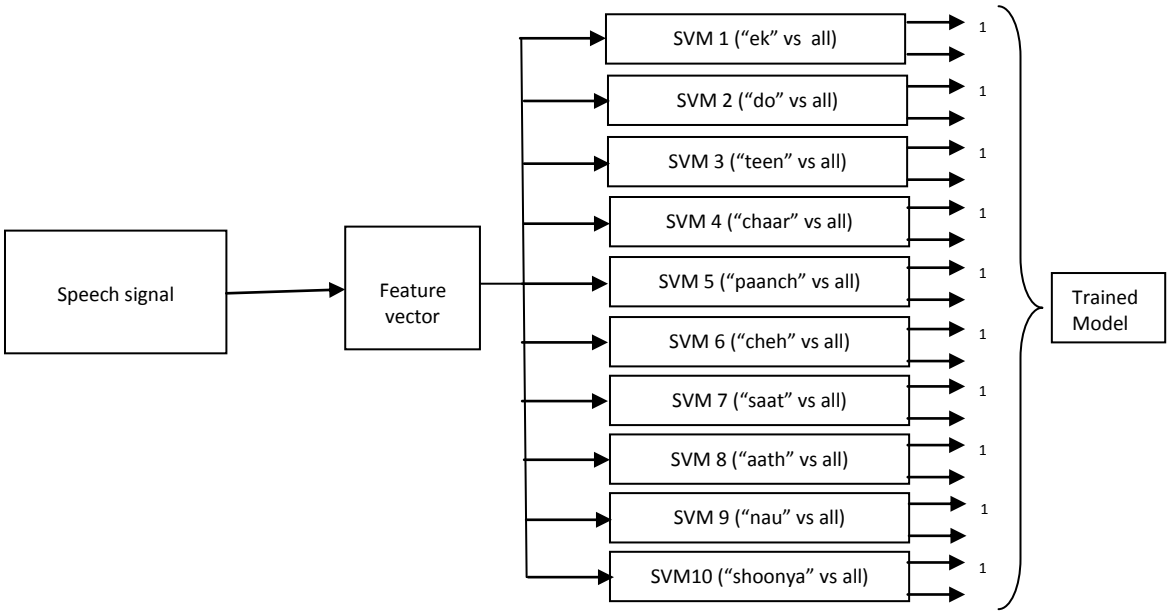


Figure 4.1: Training strategy of one-versus-all SVM

4.1.4 Results and Discussion

The effect of number of frames on the recognition rate for three considered databases with LPCC, MFCC and WPMFCC features is depicted in Figures 4.2-4.10. It has been observed from Figures 4.2-4.4 that for Hindi database, the highest recognition rate with linear, polynomial and RBF kernels is achieved with 30, 45 and 40 frames, respectively. The highest speech recognition rates achieved with linear, polynomial and RBF kernels are 66.2%, 58.0% and 64.0%, respectively for Hindi database with WPMFCC features.

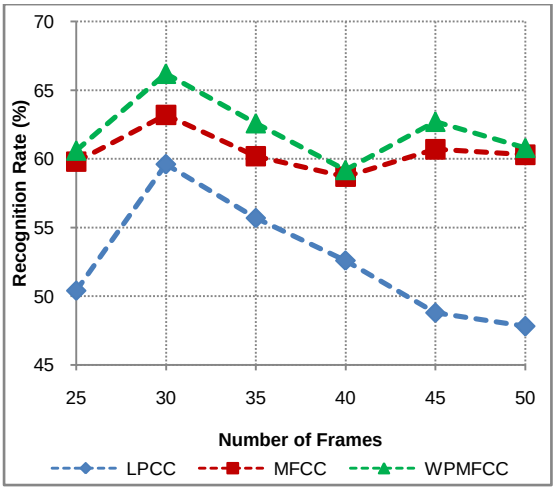


Figure 4.2: Recognition rate for different number of frames using SVM with linear kernel for Hindi database

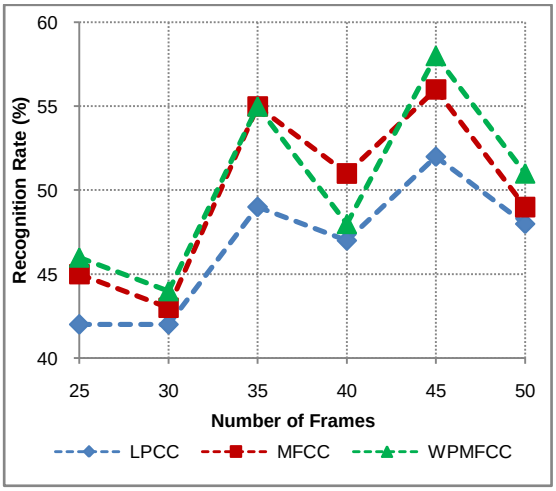


Figure 4.3: Recognition rate for different number of frames using SVM with polynomial kernel for Hindi database

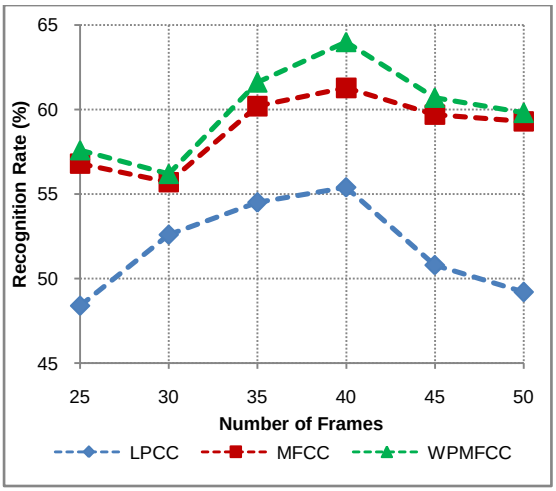


Figure 4.4: Recognition rate for different number of frames using SVM with RBF kernel for Hindi database

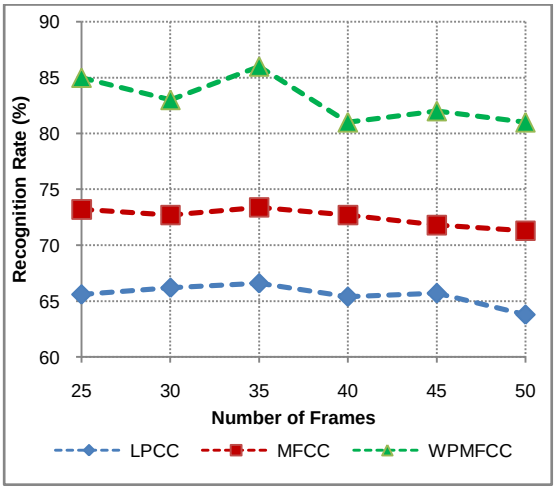


Figure 4.5: Recognition rate for different number of frames using SVM with linear kernel for TI-20 database

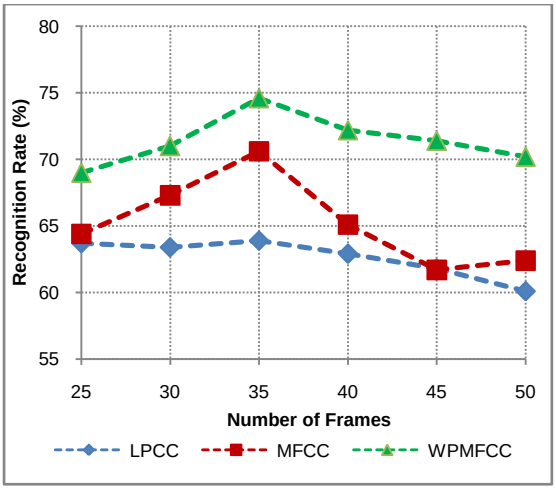


Figure 4.6: Recognition rate for different number of frames using SVM with polynomial kernel for TI-20 database

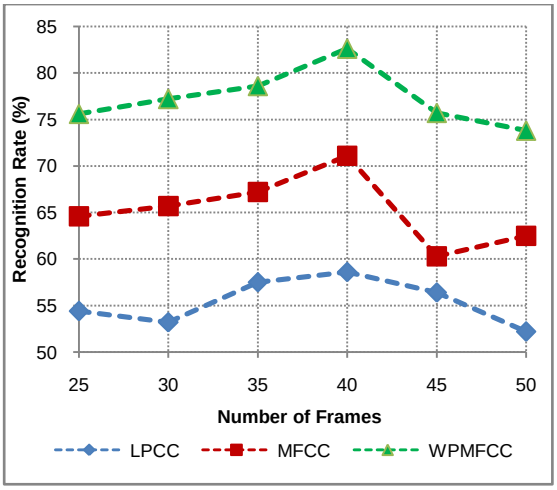


Figure 4.7: Recognition rate for different number of frames using SVM with RBF kernel for TI-20 database

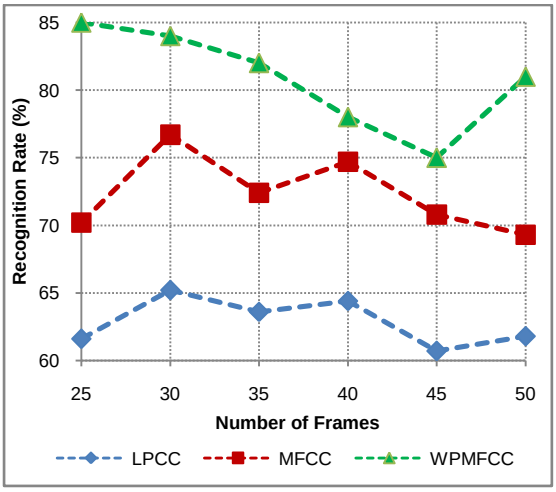


Figure 4.8: Recognition rate for different number of frames using SVM with linear kernel for TI-ALPHA database

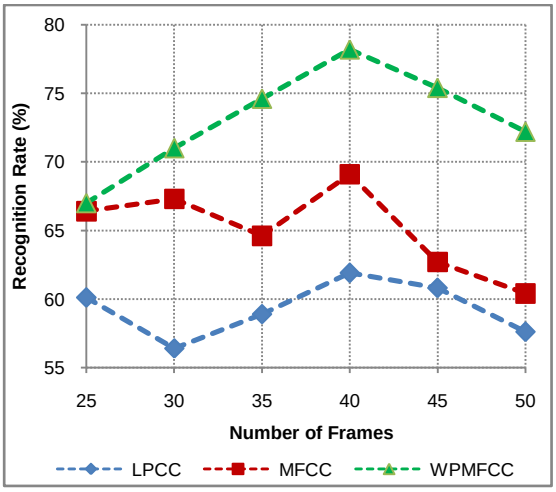


Figure 4.9: Recognition rate for different number of frames using SVM with polynomial kernel for TI-ALPHA database

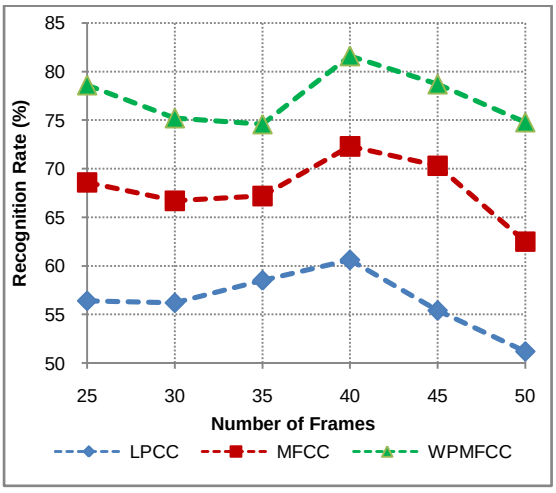


Figure 4.10: Recognition rate for different number of frames using SVM with RBF kernel for TI-ALPHA database

From Figures 4.5-4.7, it has been observed that for TI-20 database, the highest recognition rate with linear, polynomial and RBF kernels is achieved with 35, 35 and 40 frames, respectively. The highest speech recognition rates achieved with these kernels are 86.0%, 74.6% and 82.6%, respectively for TI-20 database with WPMFCC features.

It has also been observed from Figures 4.8-4.10 that for TI-ALPHA database, the highest recognition rate with linear, polynomial and RBF kernels is achieved with 30, 40 and 40 frames, respectively. The highest speech recognition rates achieved with linear, polynomial and RBF kernels are 85.0%, 78.2% and 81.6%, respectively for this database with WPMFCC features.

It has been observed that selection of appropriate value of frame is very important for better performance. It has also been observed that for LPCC, MFCC and WPMFCC features, the best recognition rate is achieved with linear kernel for all three considered databases. The SVM performance is highly affected by kernel parameters. In the case of polynomial or RBF kernel, kernel-parameters **(**γ**,** *r*, *d*) need to be adjusted to get the reasonably good performance. The advantage of linear kernel is that no kernel parameter is required to set. So sometimes, linear kernel outperforms other kernel performances.

It has been observed, as mentioned above that WPMFCC features outperform other features considered in this study. WPMFCC features are a combination of WPT and MFCC features, as illustrated in Section 3.1.2. This has further been observed that the results obtained using MFCC features are consistently better than the results obtained with LPCC features. This performance can be attributed to the fact that MFCCs are calculated using the concept of logarithmically spaced filter banks, clubbed with the concept human auditory system.

4.2 EXPERIMENT 2: SPEECH RECOGNITION USING SVM WITH OPTIMIZED PARAMETERS USING PPO AND HOOKE-JEEVES METHOD

SVM has good application prospects for speech recognition problems. However, selection of optimal values of parameters is an important issue in its implementation. To improve the learning ability of SVM, there is a need to apply effective optimization techniques to search optimal SVM hyper-parameters. In Chapter-3, two optimization techniques, *i.e.*, predator prey optimization with Hooke-Jeeves method (PPO-HJ) and predator influenced civilized swarm optimization (PCSO), have been proposed and tested on ANN based speech

recognition system. It has been concluded in Chapter-3, that PPO-HJ method outperforms the results obtained by PCSO technique. So, in this chapter PPO-HJ method is undertaken as an optimization technique to search SVM hyper-parameters. The PPO and Hooke-Jeeves method are discussed in Section 3.3.1 and 3.3.2, respectively. In this experiment, same speech databases have been undertaken as used in Experiment 1.

4.2.1 Implementation

Initially, pre-processing of speech signal is performed. The acoustic features of speech signal are acquired by feature extraction technique as discussed in Section 3.1.2. In this work, PPO-HJ method is applied to optimize the SVM hyper-parameters, *i.e.*, error penalty parameter *C* and kernel parameter γ. Optimization process is carried out in two phases. During the first phase, PPO technique is applied for a fixed number of iterations and then local best positions obtained by PPO technique are further improved by applying Hooke-Jeeves method. This proposed technique is described, in detail, in following sub-sections.

4.2.1.1 Swarm initialization and upgradation

In a global search technique, like PPO, each particle represents a probable solution to the optimization problem. In this experiment, SVM parameters (C and γ) represent dimensions of each particle. The PPO optimization technique has prey and predator particles. The role of predator particle is to influence the prey particle in effective manner so that prey can achieve best solution. The status of particles is characterized according to its position and velocity.

The prey population is randomly initialized within their respective limits as:

$$X_{il}^{0} = X_i^{\min} + r_1\left(X_i^{\max} - X_i^{\min}\right) \qquad i = 1, 2;\ \ l = 1, 2, \ldots, NP \tag{4.6}$$

$$V_{il}^{0} = V^{\min} + r_2\left(V^{\max} - V^{\min}\right) \tag{4.7}$$

where NP is number of prey particles; r_1 and r_2 are uniformly distributed random numbers over (0, 1).

Single predator position and velocity is randomly initialized within their respective limits as:

$$X_{Pi}^{0} = X_i^{\min} + r_3\left(X_i^{\max} - X_i^{\min}\right) \tag{4.8}$$

$$V_{Pi}^{0} = V^{\min} + r_4\left(V^{\max} - V^{\min}\right) \tag{4.9}$$

where r_3 and r_4 are again uniformly distributed random numbers over (0, 1).

The particles velocities and positions are updated as discussed in Section 3.3.1. The stepwise procedure to implement the PPO with Hooke-Jeeves method is elaborated in algorithm 4.1.

Algorithm 4.1: PPO with Hooke-Jeeves method

Step 1: Randomly generate position and velocity of swarm within predefined limits as given by (4.6) to (4.9).

Step 2: Initialize the iteration counter for proposed technique as $t = 1$.

Step 3: Train SVMs with prey positions.

Step 4: Compute the accuracy as given by (4.5) of each individual prey in the population.

Step 5: Update the *Pbest* position of every prey particle.

Step 6: $IF\{(t\ MOD\ TT) = 0\}$ apply Hooke-Jeeves method to improve *Pbest* location (Index *TT* is set to 10).

Step 7: Update the *Gbest* prey position.

Step 8: Update the predator velocity and position.

Step 9: Update the prey velocity and position. If updated velocity and position violate the limits then set these to its corresponding limits.

Step 10: Set $t = t + 1$

Step 11: Go to step 3 until the algorithm is evaluated for a given maximum number of iterations.

Step 12: STOP

4.2.1.2 Parameter setting

To obtain maximum speech recognition rate, it is necessary to set proper values of different parameters of PPO with Hooke-Jeeves method. A number of trials has been given to get optimum values of parameters. Parameters have been varied between minimum set values to maximum set values with a certain step size. Maximum number of iterations is set to 200. Minimum values, maximum values, step size and optimum value of these parameters are given in Table 4.3. Another parameter inertia weight w of PPO is linearly decreasing with iterations from 0.9 to 0.4.

In a population based search technique, population size is an important parameter that influences the results in terms of solution quality and simulation time. One has to choose the

value of population size with care as a large value or a small value of this parameter may not give desired accuracy. In this work, we have performed experimentation taking five values of this parameter (10, 15, 20, 25 and 30). For each value, 20 trials have been carried out in order to find the efficient population size. Out of the five population sizes considered in the experiments, we achieved the highest accuracy when the population size was taken as 20. The results reported in the next section have been obtained taking a population size of 20.

Table 4.3: Range, step size and optimal value of parameters for PPO-HJ technique

Parameter	Minimum Value	Maximum Value	Step Size	Optimal Value
C_1	1.0	2.5	0.5	2.0
C_2	1.0	2.5	0.5	2.0
a_i	0.05	0.2	0.05	0.1
b_i	0.50	2.0	0.5	1.5
$pf_{\max}$	0.10	0.95	0.05	0.90

4.2.2 Results and Discussion

The extracted features from speech signal have been given as input to SVM for recognition. The hyper-parameters of are optimized using SVM with PSO (SVM-PSO), SVM with PPO (SVM-PPO), and SVM with PPO-HJ (SVM-PPO-HJ) techniques to improve the speech recognition rate. The network is trained by using 10-fold cross validation approach. The results achieved by SVM with values of parameters optimized using PPO-HJ technique are compared with results obtained by SVM with default values of parameters and SVM with values of parameters optimized using PSO and PPO and are presented in Table 4.4. The default value of hyper-parameters are given in math library (MATLAB and Bioinformatics toolbox, 2009) and for RBF kernel, parameters are C = 1, σ = 1. In the experiments, the highest recognition rates of 92.2%, 90.3% and 81.5% have been achieved for TI-20, TI-ALPHA and Hindi speech databases, respectively using SVM-PPO-HJ technique with WPMFCC features.

To assess the effect of noise on recognition rate, experiments have also been carried out on noisy test samples. In the experimental work, artificial white noise is added to samples of all three databases to get wide range of SNR (0, 10, 20, 30 and 40 dB). For different SNR, the recognition rates obtained by SVM with various techniques for three databases using WPMFCC features are presented in Figures 4.11-4.13. It can be observed from Figures 4.11-

4.13, that recognition rate decreases with increasing value of noise and for each value of SNR, PPO-HJ technique outperforms other techniques.

Table 4.4: Recognition rates using default and optimized values of hyper-parameters

Database	Features	Recognition Rate (%)			
		SVM with default values	SVM-PSO	SVM-PPO	SVM-PPO-HJ
TI-20	MFCC	71.1	75.4	78.6	81.3
	DWT	74.7	77.9	81.8	87.5
	WPT	78.0	81.7	84.3	89.7
	WPMFCC	82.6	86.7	90.4	**92.2**
TI-ALPHA	MFCC	72.3	78.7	79.8	82.7
	DWT	68.4	75.4	78.2	84.6
	WPT	79.3	81.8	82.5	86.1
	WPMFCC	81.6	83.2	86.7	**90.3**
Hindi Speech	MFCC	61.2	66.7	73.8	75.3
	DWT	58.8	63.5	70.9	72.4
	WPT	62.6	65.1	72.9	77.9
	WPMFCC	64.0	69.7	77.1	**81.5**

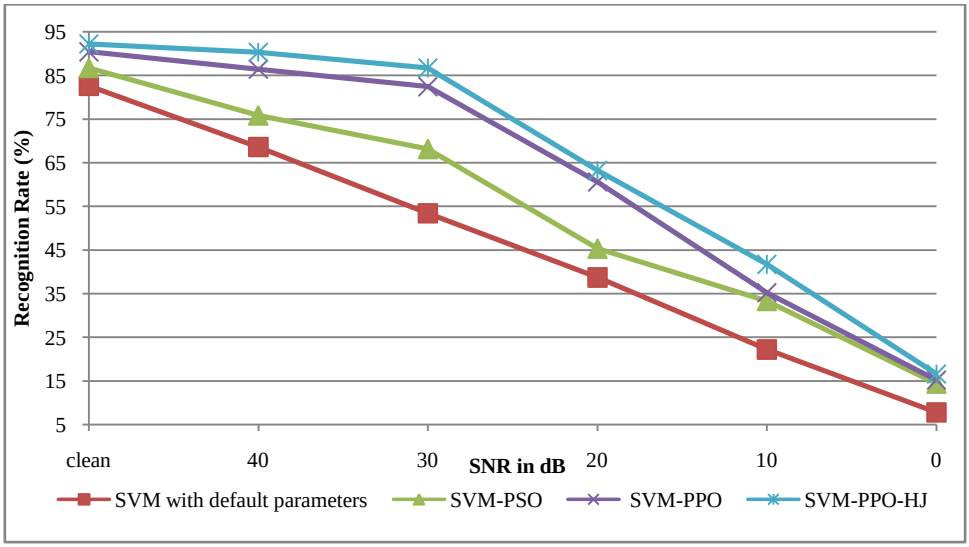


Figure 4.11: Recognition rate obtained with different techniques under noisy conditions for TI-20 database

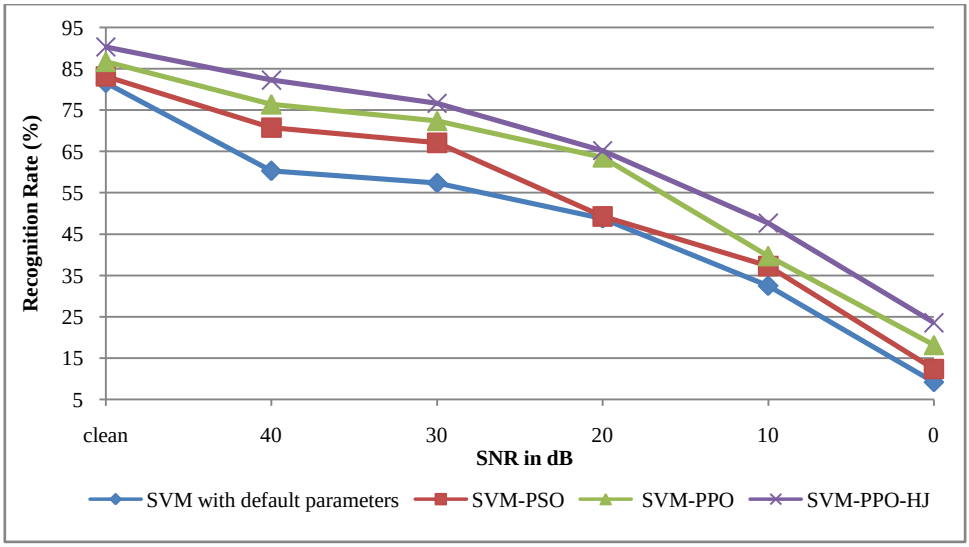


Figure 4.12: Recognition rate obtained with different techniques under noisy conditions for TI-ALPHA database

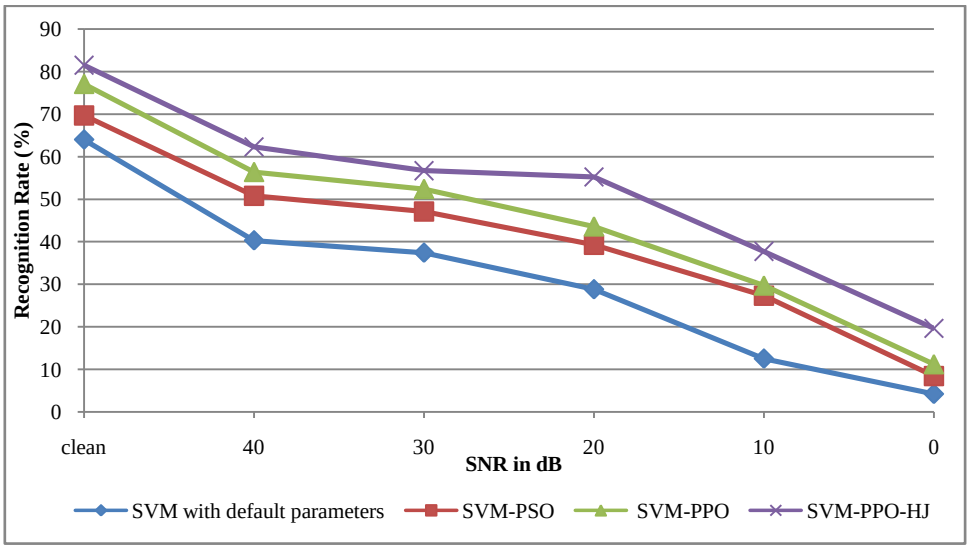


Figure 4.13: Recognition rate obtained with different techniques under noisy conditions for Hindi database

4.3 EXPERIMENT 3: SPEECH RECOGNITION USING SVM WITH MIXED-VARIABLE PPO AND HOOKE-JEEVES METHOD

In this section, experiments have been conducted to recognize isolated words using SVM classifier. The parameters of SVM have been optimized using mixed-variable PPO and Hooke-Jeeves method. This is an established fact that recognition rate and computation time are affected by the choice of feature set. So, feature selection algorithm has also been used in this section to select most significant features. As the parameters of ANN are weights and biases that have big dimension as compared to SVM that have only two parameters to be optimize. So, optimizing parameters and features simultaneously will become a very high

dimension and complex problem.In this section, an optimization technique is proposed to search the most relevant feature set of speech, and, optimal hyper-parameters of SVM. The proposed hybrid optimization technique is an integration of mixed-variable PPO technique and Hooke-Jeeves method (MVPPO-HJ). The SVM with MVPPO-HJ technique is implemented on the word databases as considered in Section 4.1.

4.3.1 Fitness Function

Two objectives behind proposing a fitness function are increased classification accuracy and reduced feature set. As such, there is a trade-off between these two objectives. Sarafrazi and Nezamabadi-pour (2013) have combined these two objectives to achieve high accuracy with less number of features as:

$$f = W_1 \times F_1 + W_2 \times F_2 \tag{4.10}$$

where f is fitness function; W_1 represents the weight for recognition accuracy function F_1 and W_2 is the weight for feature set function F_2. The values of W_1 and W_2 are set to 1.

The recognition accuracy function F_1 is defined as:

$$F_1 = \frac{CR}{CR + IR} \times 100 \tag{4.11}$$

Here, CR represents the number of correct recognitions and IR represents the number of incorrect recognitions. The accuracy function can also be defined as discussed in Section 4.1.2.

The function F_2 represents feature set and is defined as

$$F_2 = \left[1 - \frac{\sum_{k=1}^{N_F} f_k}{N_F} \right] \tag{4.12}$$

whereN_F is number of features and f_k is the value of feature mask defined as:

$$f_k = \begin{cases} 1 & if\ \ feature\, is\, selected \\ 0 & otherwise \end{cases} \tag{4.13}$$

For binary classifier, other performance measures are sensitivity and specificity. The sensitivity and specificity evaluate the discrimination capacity of classifier to classify between positive class and negative class. The sensitivity and specificity are measured by true

positive rate (TPR) and false positive rate (FPR), respectively. The TPR and FPR are computed as:

$$TPR = \frac{TP}{P};\ FPR = \frac{FP}{N} \tag{4.14}$$

where $P = TP + FN$ and $N = FP + TN$

4.3.2 Mixed-Variable PPO

The discussion related to continuous PPO is given in Section 3.3.1. In binary PSO, velocity is a probability vector and it computes the likelihood of the binary variables having a value of one (Bao *et al.*, 2014b). In binary PPO, predator and prey velocity is updated as for continuous PPO, but particles are restricted in the search area between zero and one, so the velocities of predator and prey particles are updated by sigmoid limiting transformation as:

$$S(Velocity) = \frac{1}{1 + \exp(-Velocity)} \tag{4.15}$$

The predator position is updated as:

$$X_{Pi} = \begin{cases} 1 & r_5 < S(V_{pi}) \\ 0 & otherwise \end{cases},\ i = 1, 2, \ldots, n \tag{4.16}$$

The prey positions are updated as:

$$X_{il} = \begin{cases} 1 & r_6 < S(V_{il}) \\ 0 & otherwise \end{cases},\ l = 1, 2, \ldots, NP;\ i = 1, 2, \ldots, n \tag{4.17}$$

where r_5 and r_6 are uniformly distributed random numbers over (0, 1); n represents number of dimensions; NP is number of prey particles.

4.3.3 Implementation

The aim of this research work is to improve the speech recognition rate by optimizing SVM hyper-parameters along with selection of most relevant feature subset. The experiments have been conducted by extracting MFCC features for each utterance as discussed in Section 3.1.2. The SVM technique with RBF kernel is implemented for classification. One-versus-all approach is used to construct the SVM classifier. The feature set and SVM hyper-parameters are randomly initialized as PPO particles. The SVM hyper-parameters (C, γ) are continuous in nature and feature mask is of a binary nature. The particles are divided in two parts; first part consists of feature mask and the second part contains RBF hyper-parameters. The feature

mask is either 0 or 1; if the mask is 0, then corresponding feature will not be selected and if it is 1, then corresponding feature will be selected (Bao *et al.*, 2014a; Xiong *et al.*, 2015). Table 4.5 shows the representation of particle *l* with a dimension of $N_F + 2$.

The initial prey and predator position and velocity representing feature mask are randomly generated as follows:

Prey particles:

$$X_{il}^0 = \begin{cases} 1 & if \;\; r_7 \geq 0.5 \\ 0 & otherwise \end{cases}, \quad i = 1, 2, ..., N_F; \; l = 1, 2, ..., NP \tag{4.18}$$

$$V_{il}^0 = \begin{cases} 1 & if \;\; r_8 \geq 0.5 \\ 0 & otherwise \end{cases}, \quad i = 1, 2, ..., N_F; \; l = 1, 2, ..., NP \tag{4.19}$$

where NP is number of prey particles; r_7 and r_8 are uniformly distributed random numbers over (0, 1).

Predator particles:

$$X_{Pi}^0 = \begin{cases} 1 & if \;\; r_9 \geq 0.5 \\ 0 & otherwise \end{cases}, \; i = 1, 2, ..., N_F \tag{4.20}$$

$$V_{Pi}^0 = \begin{cases} 1 & if \;\; r_{10} \geq 0.5 \\ 0 & otherwise \end{cases}, \; i = 1, 2, ..., N_F \tag{4.21}$$

where r_9 and r_{10} are again uniformly distributed random numbers over (0, 1).

Table 4.5: l^{th} prey particle representation

Input feature mask				C	γ
Discrete				Continuous	
$X_{1,l}$	--	--	$X_{N_F,l}$	$X_{N_F+1,l}$	$X_{N_F+2,l}$

where N_F represents number of features of speech; $X_{1,l}$ represents 1^{st} dimension of l^{th} prey particle.

The prey and predator position and velocity representing SVM hyper-parameters (C and γ) are randomly initialized as:

Prey particles:

$$X_{il}^0 = X_i^{\min} + r_{11}\left(X_i^{\max} - X_i^{\min}\right), \quad i = N_F + 1, \; N_F + 2; \;\; l = 1, 2, ..., NP \tag{4.22}$$

$$V_{il}^0 = V_i^{\min} + r_{12}\left(V_i^{\max} - V_i^{\min}\right), \quad i = N_F + 1, \; N_F + 2; \;\; l = 1, 2, ..., NP \tag{4.23}$$

Predator particles:

$$X_{Pi}^0 = X_i^{\min} + r_{13}\left(X_i^{\max} - X_i^{\min}\right), \; i = N_F + 1, \; N_F + 2 \tag{4.24}$$

$$V_{Pi}^{0} = V^{\min} + r_{14}\left(V_i^{\max} - V_i^{\min}\right), \quad i = N_F + 1, N_F + 2 \tag{4.25}$$

where r_{11}, r_{12}, r_{13} and r_{14} are uniformly distributed random numbers over (0, 1).

Prey particles are evaluated on the basis of fitness function as given in (4.10) and local and global best solutions are updated. To further improve the quality of solution, local best solutions are optimized by Hooke-Jeeves method. The Hooke-Jeeves method is discussed in Section 3.3.2. The flow chart of Hooke-Jeeves method to deal with mixed variable is shown in Figure 4.14. The flow chart of MVPPO-HJ technique for speech recognition is presented in Figure 4.15.

4.3.4 Results and Discussion

In order to test the performance of MVPPO-HJ technique, results achieved by SVM along with MVPPO-HJ method is compared with the results obtained by SVM along with mixed-variable PSO (MVPSO), mixed-variable PSO and Hooke-Jeeves method (MVPSO-HJ), mixed-variable PPO (MVPPO) and also with SVM having default values ofhyper-parameters. The recognition rates obtained by these approaches are given in Table 4.6. The experimental results indicate that recognition rate achieved by SVM along with MVPPO-HJ is better than the recognition rates obtained by other considered techniques.

For establishing that differences between two sample means is statistically significant for SVM-PPO-HJ technique when compared with SVM-PPO, SVM-PSO and SVM, Wilcoxon signed rank test has been performed. Each algorithm is run thirty times for TI-20 database considering WPMFCC features. The test is performed by taking the level of significance α as 0.01. It is observed from*p*-value reported in Table 4.7, that SVM-PPO-HJ technique is significantly better than other considered techniques.

Table 4.6: Comparison of recognition rates (%)

Applied Technique / Database	SVM with default values	SVM with MVPSO	SVM with MVPSO-HJ	SVM with MVPPO	SVM with MVPPO-HJ
TI-20	91.6	94.7	97.3	96.5	98.8
TI-ALPHA	86.2	90.1	94.0	91.7	96.6
Hindi database	83.7	89.2	91.1	90.0	93.4

Table 4.7: *p*-values for Wilcoxon signed rank test when appliedto SVM-PPO-HJ technique; SVM-PPO; SVM-PSO and SVM techniques

	SVM-PPO	SVM-PSO	SVM with default values
p-value	0.00008770	0.00008708	0.00008671

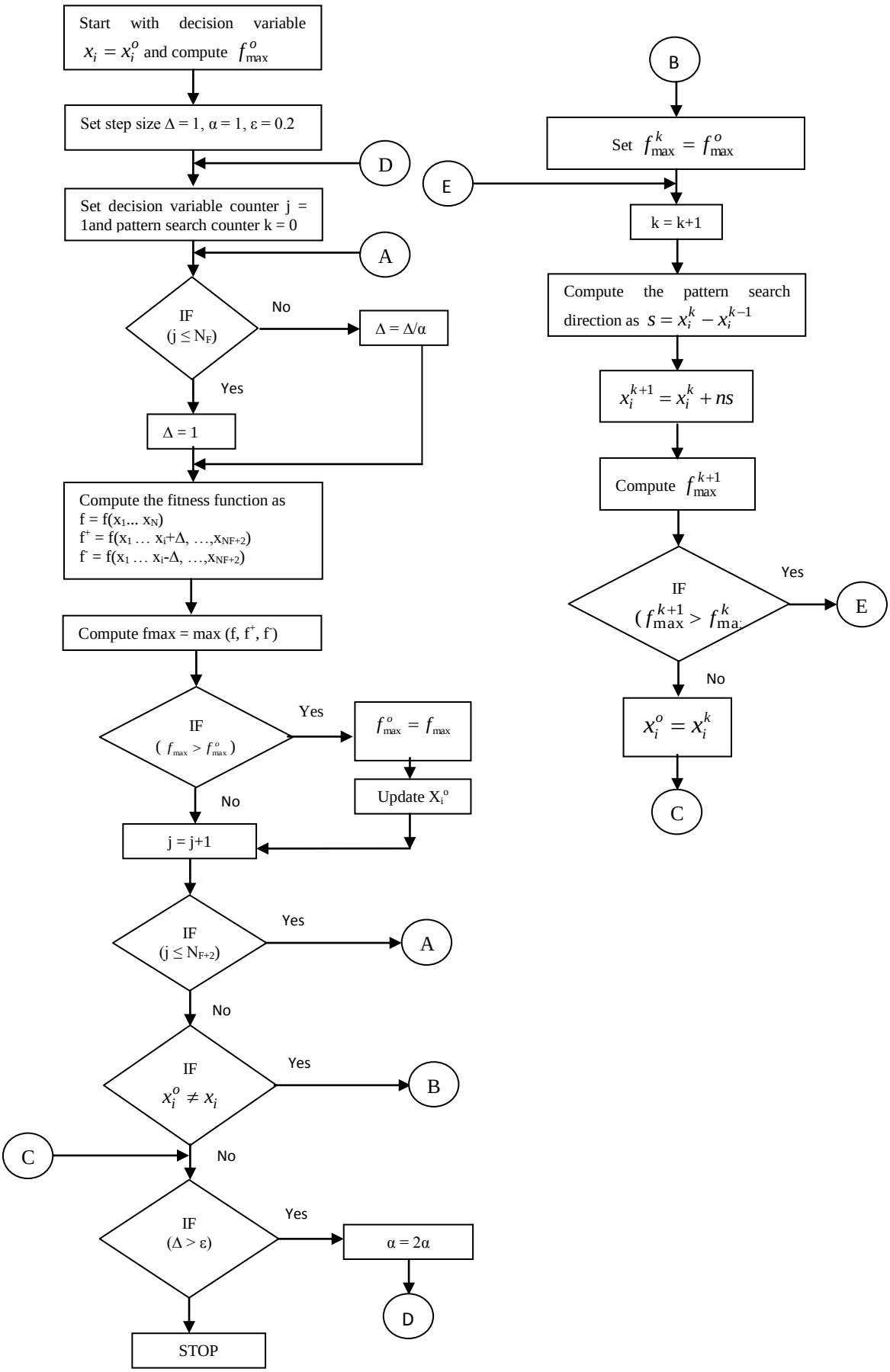


Figure 4.14: Flow chart of Hooke-Jeeves method for mixed type of decision variable

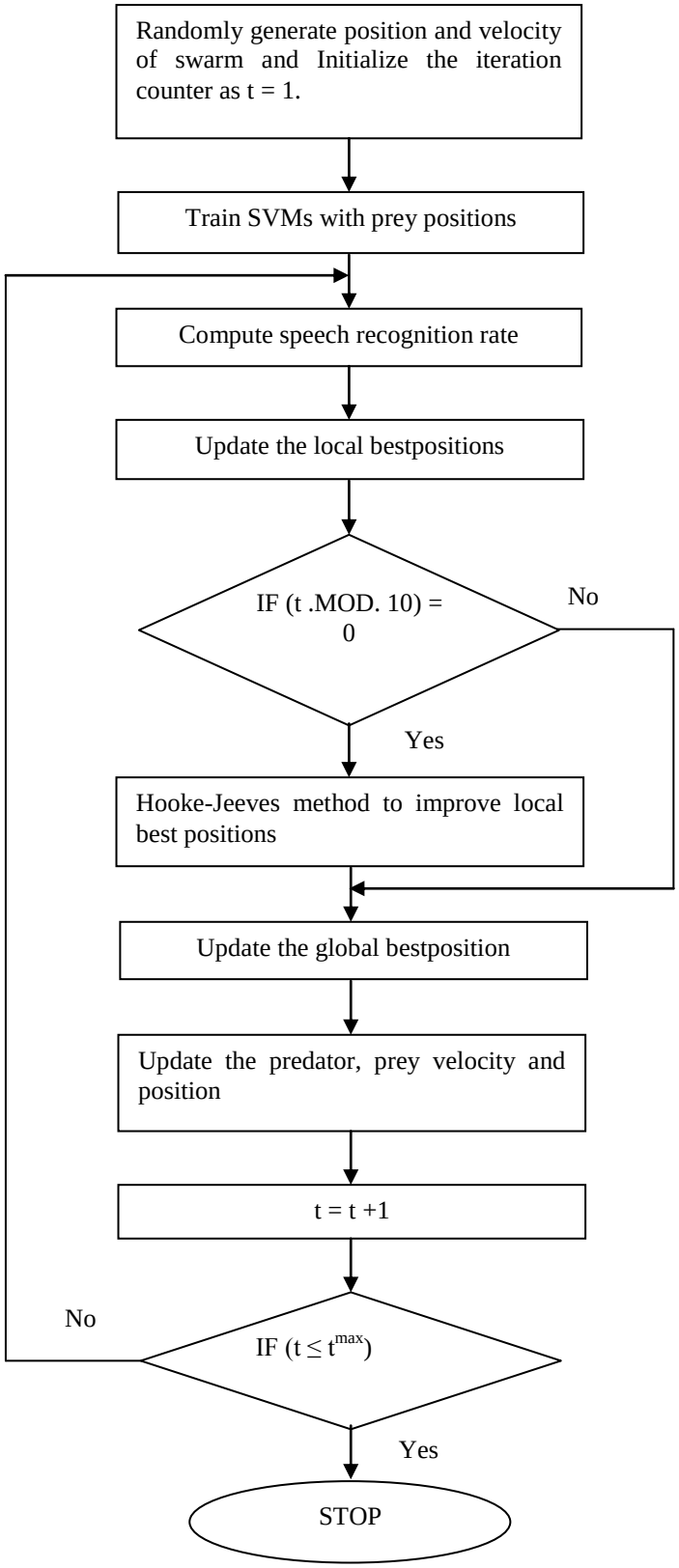


Figure 4.15: Flowchart for MVPPO-HJ technique for speech recognition

To check the validity of MVPPO-HJ technique, ROC curves have also been implemented. The ROC is a graphical plot between TPR and FPR that illustrates the performance of binary classifier. The TPR represents 'sensitivity' and FPR represents '1 - specificity'. Figures 4.16-4.18 show the ROC curves plotted for various techniques for three databases. For Hindi database, 400 test samples and ten classes gives a total of 400 positive samples and 3600 negative samples. For TI-20 database, 8320 samples give a total of 832 positive samples and 7488 negative samples. For TI-ALPHA database, 10816 samples give a total of 1082 positive samples and 9734 negative samples. Based on these statistics, the ROC curves have been drawn by varying the threshold level for the techniques under consideration and the area under the curves(AUC) has also been computed and is given in Table 4.7 for Hindi, TI-20 and TI-ALPHA databases. It can be observed that AUC of SVM with MVPPO-HJ technique is larger than the AUC for other techniques. So, the performance of MVPPO-HJ technique is better than other techniques considered in this work.

Table 4.8: Area under the ROC curve using SVM with different techniques for three databases

Technique	Area under the curve		
	Hindi	TI-ALPHA	TI-20
SVM with default parameters	0.8090	0.8096	0.8161
SVM with MVPSO	0.8715	0.8759	0.8891
SVM with MVPSO-HJ	0.9303	0.9045	0.9438
SVM with MVPPO	0.9236	0.9166	0.9287
SVM with MVPPO-HJ	**0. 9527**	**0.9539**	**0.9544**

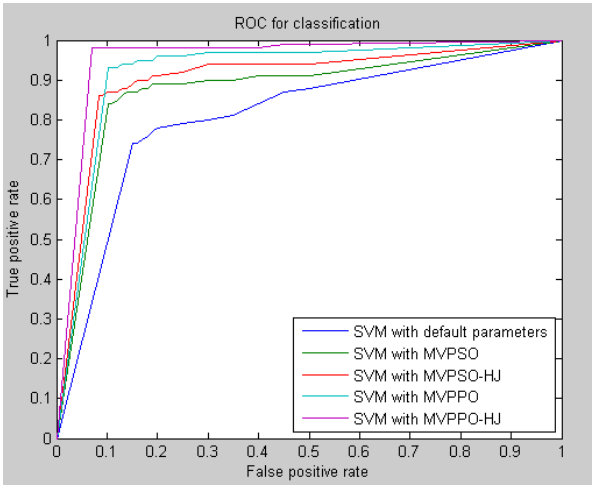


Figure 4.16: ROC curves for different techniques for Hindi database

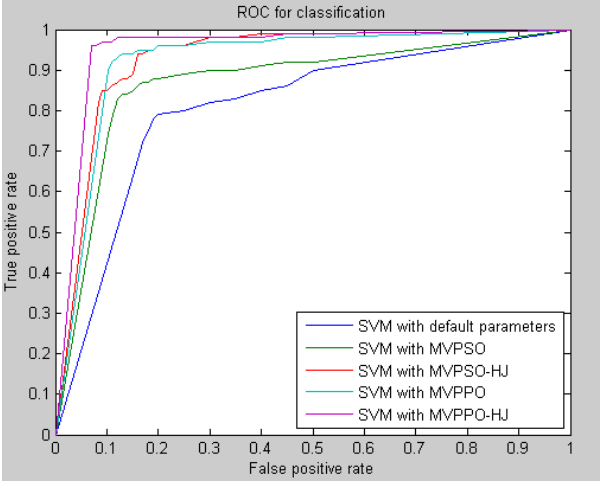


Figure 4.17: ROC curves for different techniques for TI-20 database

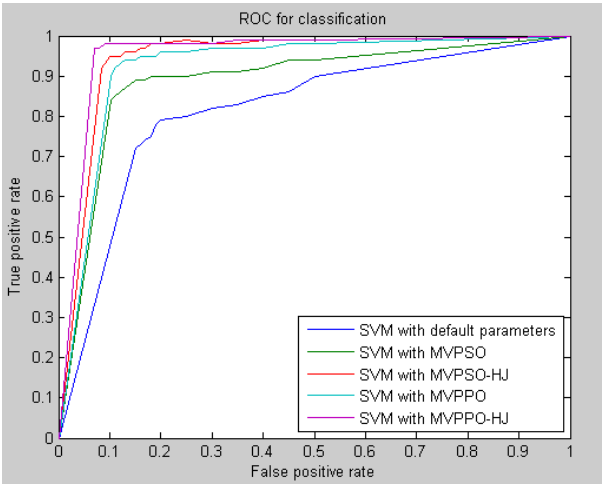


Figure 4.18: ROC curves for different techniques for TI-ALPHA database

To evaluate the robustness of MVPPO-HJ technique, experiments have also been carried out with noisy samples. Noisy samples have been obtained by artificially adding white noise with different SNR (0 to 40 dB) to all three databases. Figures 4.19-4.21 represent the recognition rates for different SNR achieved by various applied techniques with SVM model for the three databases. It has been observed that recognition rate achieved by SVM with MVPPO-HJ technique is higher than other techniques.

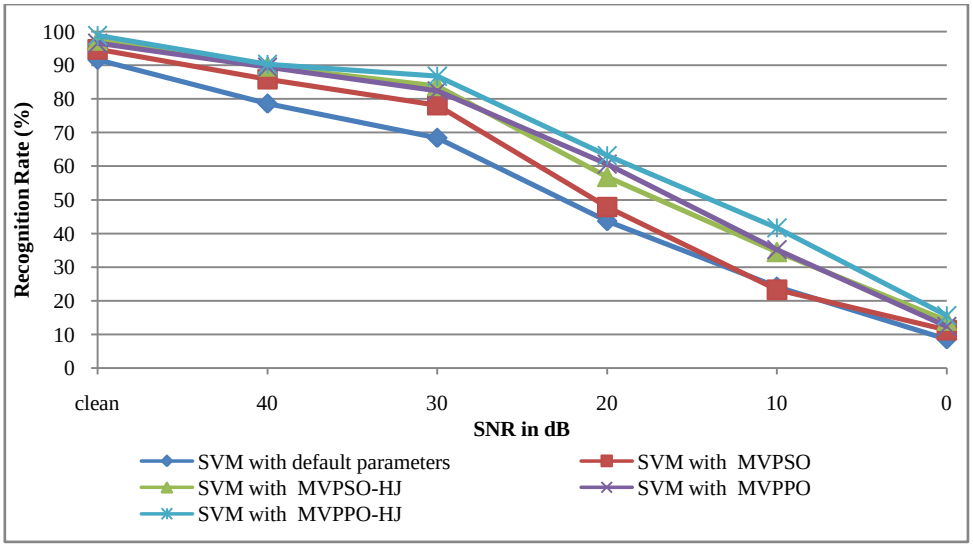


Figure 4.19: Recognition rate obtained with various techniques under noisy conditions for TI-20 database

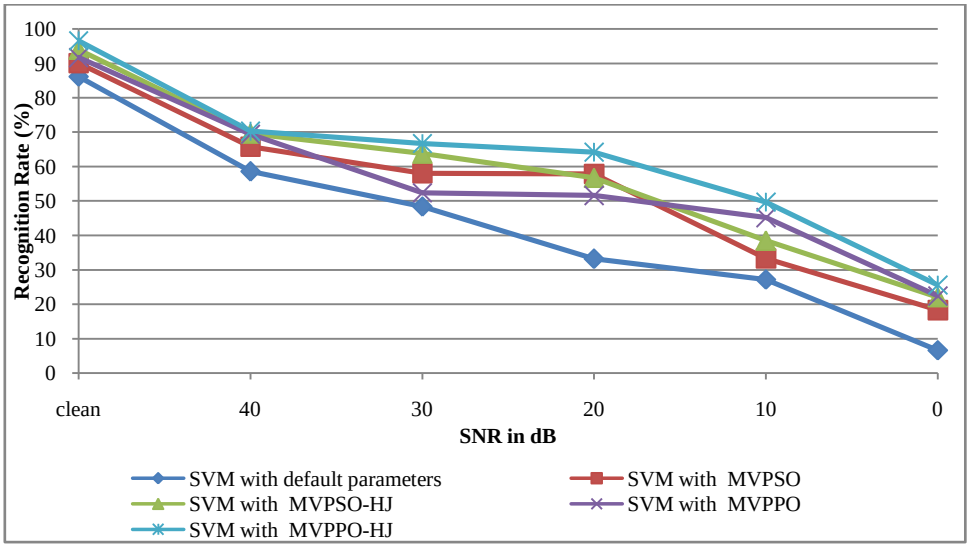


Figure 4.20: Recognition rate obtained with various techniques under noisy conditions for TI-ALPHA database

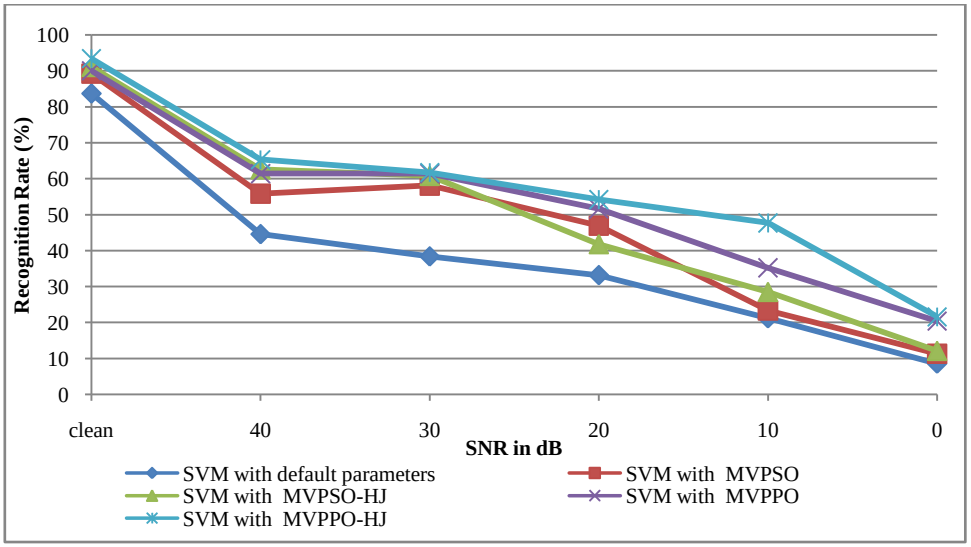


Figure 4.21: Recognition rate obtained with various techniques under noisy conditions for Hindi database

4.4 CONCLUSIONS

In this chapter, three experiments have been conducted for speech recognition with SVM classifier. The effect of changing the number of frames on recognition rate has been examined in first experiment. In this work, three SVM kernels, *i.e.*, linear, polynomial, and RBF; and LPCC, MFCC and WPMFCC speech features have been considered. The experiment has been conducted on Hindi, TI-20 and TI-ALPHA databases. For all these databases, the highest recognition rate is achieved with linear kernel along with WPMFCC features. The highest recognition rates for Hindi, TI-20 and TI-ALPHA databases are 66.2%, 86.0% and 85.0%, respectively. The number of frames that helped achieving these recognition rates is 30, 35 and 35, for Hindi, TI-20 and TI-ALPHA databases, respectively. In the first experiment, it has also been observed that, SVM hyper-parameters significantly influence the recognition rate. So, in second experiment, SVMhyper-parameters have been optimized by applying PPO-HJ method. The speech recognition rate obtained by SVM with PPO-HJ method is compared with recognition rates achieved by SVM-PSO, SVM-PPO and SVM with default values of hyper-parameters. The highest recognition rates of 81.5%, 92.2% and 90.3% have been achieved for Hindi, TI-20 and TI-ALPHA speech databases, respectively under clean environment by applying PPO-HJ method using WPMFCC features. In third experiment, another important issue, *i.e.*, selection of relevant speech features along with optimal hyper-parameters of SVM has been undertaken. The selection of speech features is binary in nature and hyper-parameters are continuous. In this research work, binary PPO is proposed and integrated with continuous PPO to deal with mix type of variables. The mixed-variable PPO with Hooke jeeves method is applied for aforementioned work. The recognition rate achieved by proposed technique with SVM model is compared with the results obtained by SVM model along with MVPSO, MVPSO-HJ, MVPPO, and with default values of SVM hyper-parameters. It has been observed that SVM with MVPPO-HJ technique gives higher accuracy for all the databases used in this work.The statistical test, ROC has also been implemented and it has been seen that AUC for MVPPO-HJ technique is better than that of other techniques.

Chapter 5

Recognition of Hindi Speech Using Optimized HMM Classifier

In this chapter, Hindi speech recognition system for isolated words and continuous speech has been developed using HMM. HMM has been used to train and recognize the Hindi speech that uses MFCCs and their first and second derivatives as features extracted from the speech utterances. To carry out this, Hidden Markov Model Toolkit(HTK) designed for speech recognition is used.In spite of universally acceptance of HMM to recognize speech, one of the main concerns with HMM is related to training phase. The training of HMM is computationally expensive and solution usually stagnated at local optimal solution. The Baum-Welch (BW) algorithm is widely used algorithm to train HMM, but it is a conventional optimization method and quality of solution highly depends on initial search point. Normally, the solution obtained from BW algorithm may converge to local optimum solution. In this chapter, two global search techniques, *i.e.*, predator-prey optimization (PPO) and predator influenced civilized swarm optimization (PCSO) have been integrated with BW algorithm to search HMM model parameters, *i.e.*, transition and emission probabilities.

5.1 ISSUES IN HINDI LANGUAGE

Hindi language is originated from Sanskrit and belongs to the Indo-European family. Devanagari script is used to write the Hindi language (Shaughnessy, 2008). The Hindi language alphabets are divided into two groups: vowels and consonants. The Hindi language contains 12 vowels and these have two forms, the dependent form and the independent form. The independent form vowels are standalone, which means the vowels are used when pronunciation is unattached, isolated and not associated with any other consonant. When the vowels are attached to any consonants, then these vowels are in dependent

form(Shaughnessy, 2008). The Hindi language has 36 consonants. A list of vowels and consonants is given in Figure 5.1.

अ	आ	इ	ई	उ	ऊ
ए	ऐ	ओ	औ	अं	अः

(a) Vowels

क	ख	ग	घ	ड.
च	छ	ज	झ	
ट	ठ	ड	ढ	ण
त	थ	द	ध	न
प	फ	ब	भ	म
य	र	ल	व	
श	स	ष	ह	
क्ष	त्र	ज्ञ		

(b) Consonants

Figure 5.1: Hindi vowels and consonants

5.2 EXPERIMENT 1: ISOLATED WORD RECOGNITION USING HMM

Hidden Markov model is a doubly stochastic process in which one process is not directly observable (Rabiner, 1989). This hidden stochastic process can be observed only through another set of stochastic processes that can produce the observation sequence. HMMs are widely used acoustic models for speech recognition systems because of their better performance than other methods. HMM consists of nodes that represent hidden states and the nodes are interconnected by links which describe conditional transition probabilities between the states. Each hidden state has an associated set of probabilities of emitting particular visible states. An HMM model with q states is given in Figure 5.2.

States are connected to each other by transition probability a_{ij} where

$$a_{ij} \geq 0 \ ; \ (1 \leq i \leq q; 1 \leq j \leq q) \tag{5.1}$$

$$\sum_{j=1}^{q} a_{ij} = 1 \ (1 \leq i \leq q) \tag{5.2}$$

A match or insert state emits an observation from an output alphabet with a probabilities $b_j(k)$ where

$$b_j(k) \geq 0 \ (1 \leq j \leq q, 1 \leq k \leq m) \tag{5.3}$$

$$\sum_{k=1}^{M} b_j(k) = 1 \tag{5.4}$$

here$M$ is number of observations.

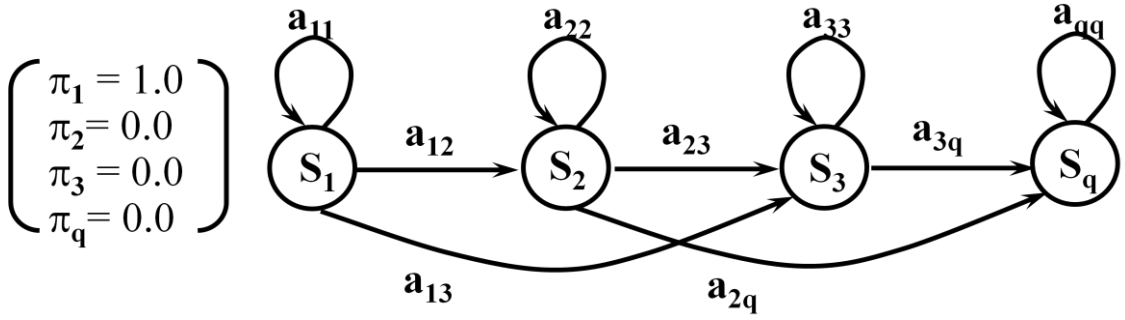


Figure 5.2: A left-to-right HMM topology

The block diagram of an isolated word recognizer using HMM is shown in Figure 5.3(Rabiner and Juang, 1993). For a vocabulary size of V, an HMM λ^v is build for each word v in the vocabulary that estimate the model parameters (A, B, π) such as the likelihood of the training set observation vector is optimized for the v^{th} word. For a word to be recognized, the observation sequence O is measured using feature set of the corresponding word. The observation sequence is given to each HMM model to calculate model likelihoods $P(O|\lambda^v)$, $1 \leq v \leq V$; and the word is selected whose model likelihood is highest.

5.2.1 Database Used

A speech recognition system needs a collection of utterances for training and testing. In this work, a self-recorded isolated Hindi words database has been considered for experimentation. The database consists of 20 Hindi words, as given in Table 4.1, with 50 utterances each, spoken by two male and two female speakers. The data is recorded in a quiet room environment at a sampling rate of 44.1 kHz.

5.2.2 Fitness Function

An HMM must be trained to describe the observational sequence of the word accurately. One of the most commonly used criteria for ASR is to maximize the probability $P[O|\lambda]$ of the observation sequence O, generated by the given HMM λ. However, the dynamic range of the $P[O|\lambda]$ is normally very small and to improve the precision range, the logarithm of the likelihood is used instead. The average likelihood probability of the HMM solution λ that generates the training observation sequences $O_1, O_2, ..., O_M$ is given as:

$$lP = \frac{1}{M}\left(\sum_{i=1}^{M} \log(P(O_i|\lambda)\right) \tag{5.5}$$

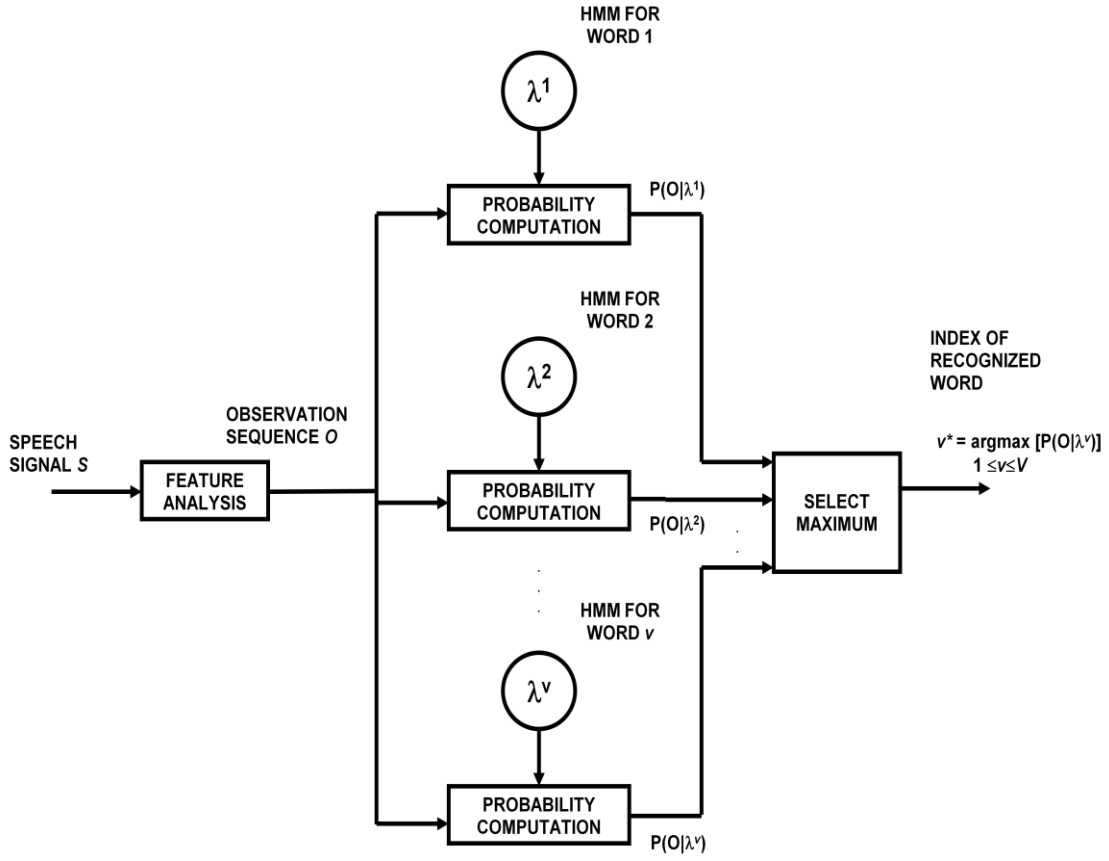


Figure 5.3: Block diagram of an isolated word recognizer

5.2.3 Implementation

In this section, the implementation of Hindi speech recognition system has been presented. Hindi speech recognition system is developed using HTK toolkit v3.4. Firstly, the HTK

training tools are used to estimate the parameters of a set of HMMs using training utterances and their associated transcriptions. Secondly, unknown utterances are transcribed using the HTK recognition tools. System is trained for 20 Hindi words. Word model has been used to recognize the speech.

To recognize a word using HTK requires that a task grammar and a dictionary must be created. The task grammar consists of a set of variable definitions followed by a regular expression describing the words to recognize. The snapshot of task grammar for Hindi words is shown in Figure 5.4, where the vertical bars denote alternatives and the angle braces denote one or more repetitions. The representation of a task grammar shown in Figure 5.4 is for user convenience. The HTK recognizer actually requires a word network in which each word instance and each word-to-word transition is listed. This word network (wdnet) can be created (Figure 5.5) automatically from the grammar using the HParse tool.

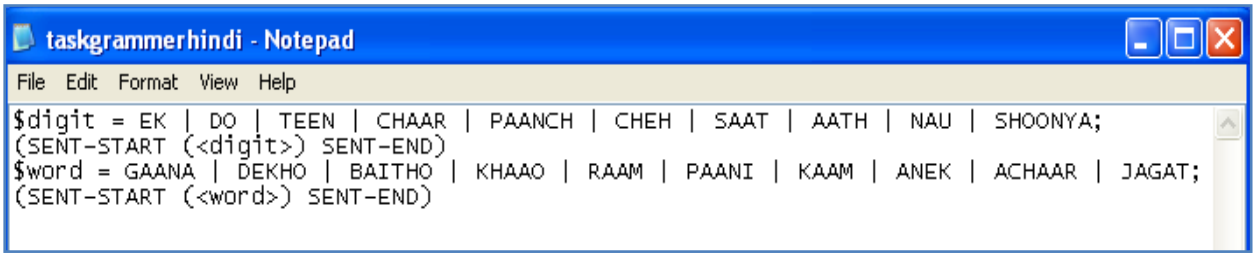


Figure 5.4: Task grammar for Hindi words

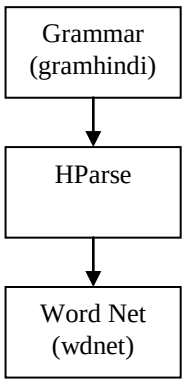


Figure 5.5: Steps to create word network

After creating the word network, the next step is to construct a pronunciation dictionary. The pronunciation dictionary defines which combination of phones gives valid words for the recognition. It contains information about different pronunciation variants of the same word. Figure 5.6 shows the snapshot of an extract of the dictionary. The words in the left column are related to their pronunciation in the right column. The dictionary itself can

be built using HDMan tool of HTK. The training and test data is recorded using the HTK tool HSLab. The input speech sample is processed at 10 ms frame rate with a Hamming window of 25ms. After that the speech signal is parameterized into a sequence of features. MFCCs have been used to parameterize the speech signal. The acoustic feature vector consists of 39 coefficients in which 13 MFCCs; and their first and second order derivatives have been used. For this purpose, HTK tool HCopy is used.

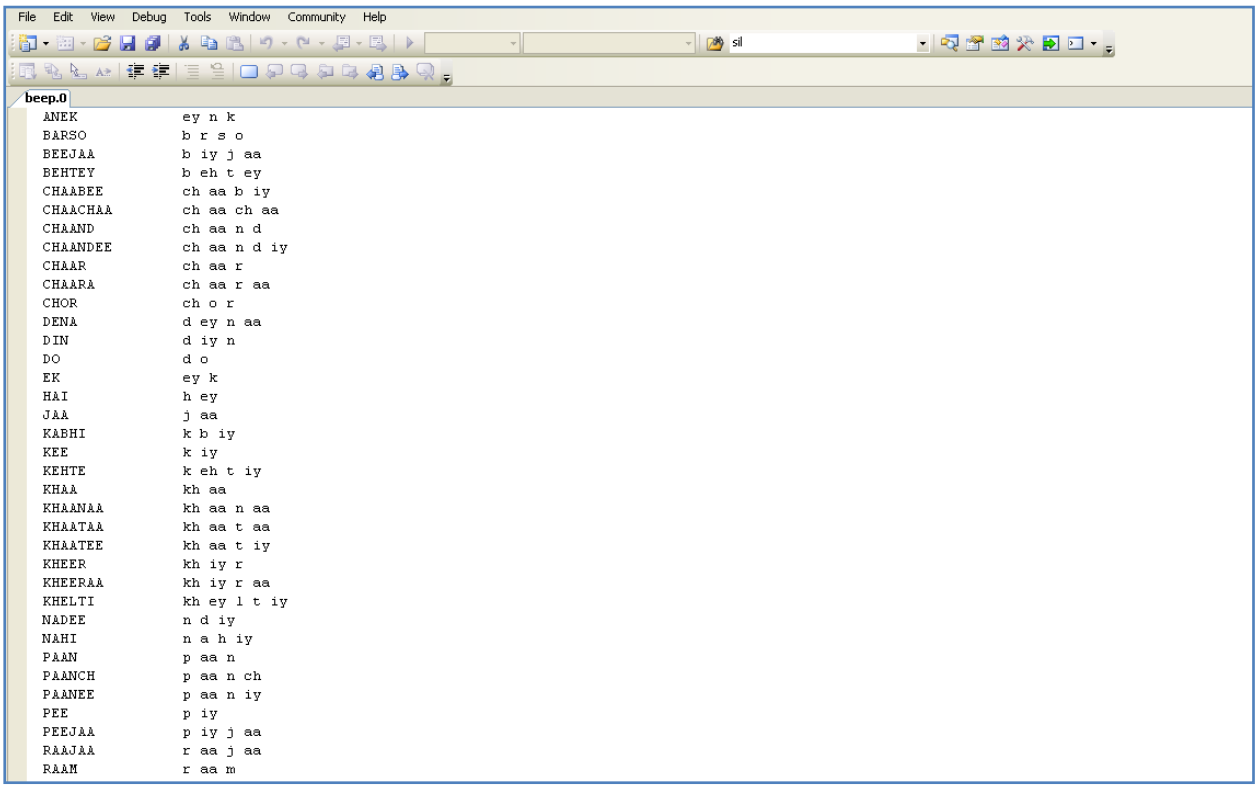


Figure 5.6: Dictionary for Hindi words

For training the HMM, a prototype HMM model is created, which is then re-estimated using the data from the speech files.There are several methods to decide the number of states in the HMM of each word. Levinson *et al.* (2001) have recommend that the number of states in the HMM model should be nearly same as number of sound units in the word. Bakis *et al.* (2001) have suggested that number of states is decided according to the average number of observations in spoken word. Rabiner and Juang (2001) have suggested that utterance of the same word varied from occurrence to occurrence so optimal number of states can only be found by refining the model after each HMM training.For prototype model, a 5-state HMM in which the first and last states are non-emitting states having left-to-right topology with no skips has been used.

The snapshot of the prototype model is shown in Figure 5.7. Apart from the models of vocabulary words, model for silence (Figure 5.8) is also included. The prototype models are initialized using the HTK tool HCompV which scan a set of data files, compute the global mean and variance and set all of the Gaussians in a given HMM to have the same mean and variance. Then HERest is used to re-estimate the parameters of the HMM model using BW algorithm based on the other samples in the training set and final set of monophones has been stored. Figure 5.9 shows a snapshot of monophones created.

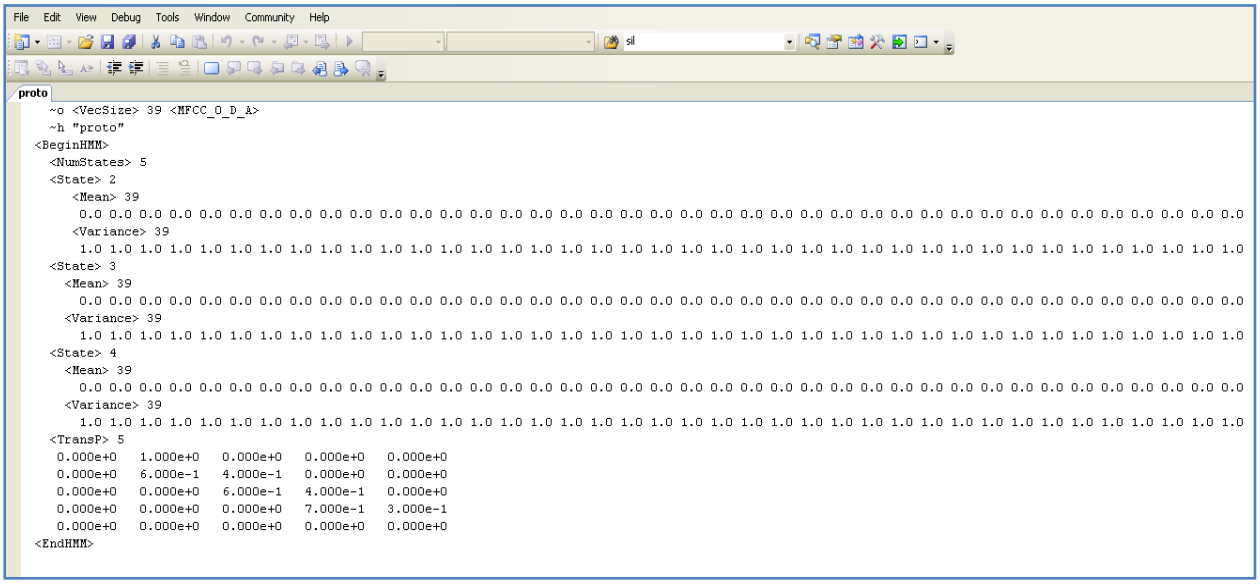


Figure 5.7: A prototype model

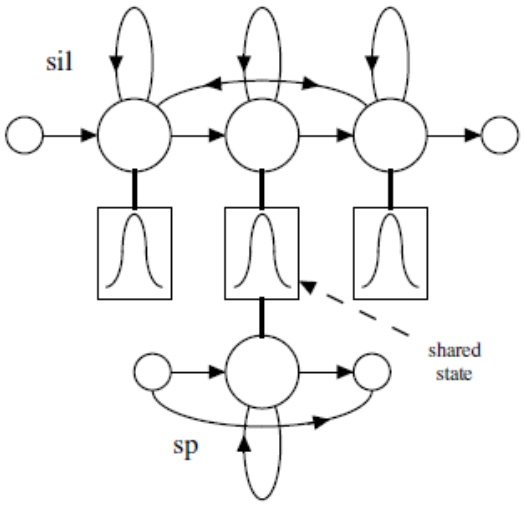


Figure 5.8: Silence model

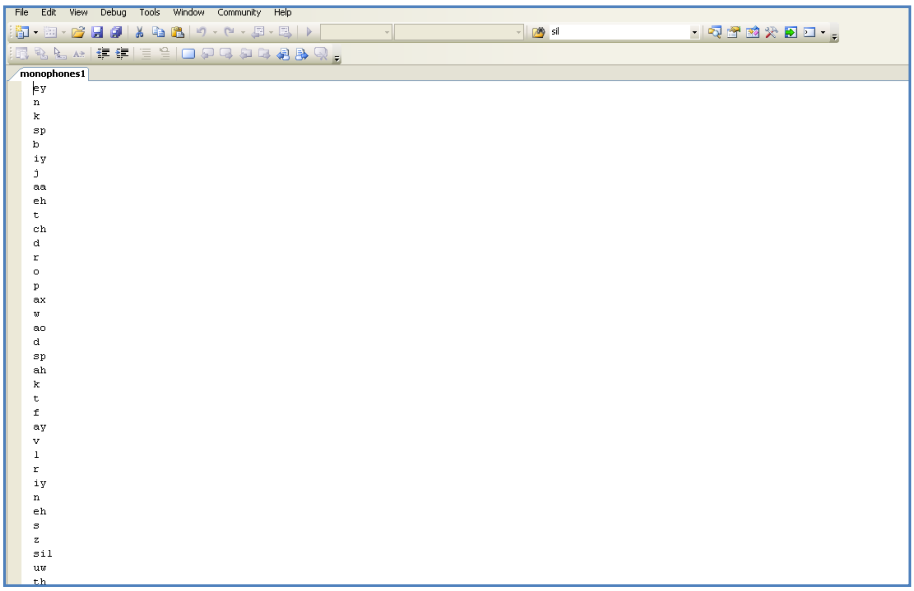


Figure 5.9: Snapshot of monophones

The dictionary contains multiple pronunciations of some words. So, the phone models can be used to realign the training data and create new transcriptions. For this purpose the HTK tool HVite is used. This command used the re-estimated *hmmdefs* and *macros* files (Figure 5.10) to transform the word level transcriptions to phone level transcriptions (Figure 5.11). Once the new alignments are created, two passes of HERest have been applied to estimate the HMM set parameters.

macros

```
~o
   <VecSize> 39
   <MFCC_0_D_A>
~v "varFloor1"
   <Variance> 39
      0.0012 0.0003 ...
```

hmmdefs

```
~h "aa"
   <BeginHMM> ...
   <EndHMM>
~h "eh"
   <BeginHMM> ...
   <EndHMM>
... etc
```

Figure 5.10: Form of master macro file

The final stage of building the model is to create context-dependent triphone HMMs. Triphone models have been created by converting the monophone transcriptions into triphone transcriptions and similar acoustic states of these triphones are tied to estimate all state distributions. After creating the context-dependent triphone HMMs, the new triphone set has

been re-estimated using HERest by replacing the monophone list with triphone list and monophone transcriptions with triphone transcriptions. Now the performance of the recognizer has been evaluated using the test data. The results of HTK recognizer have been analyzed using HResults tool.

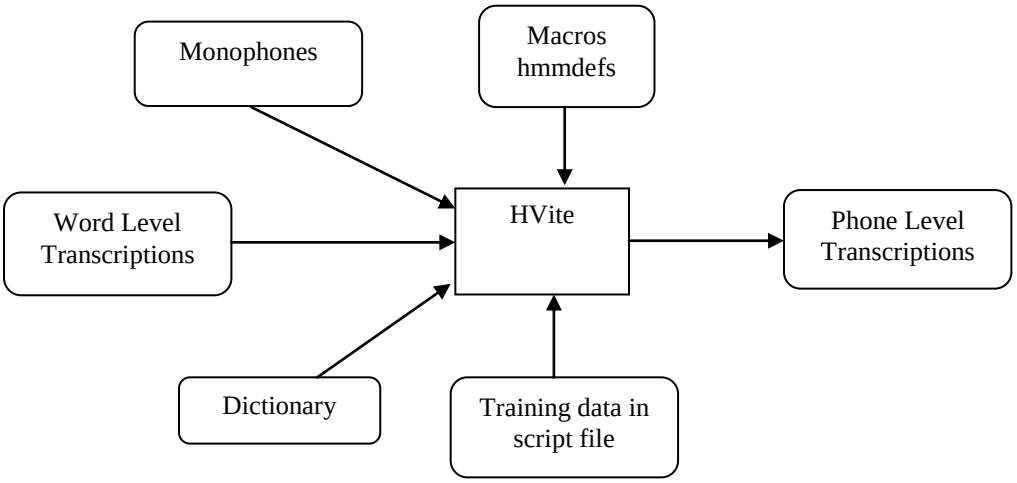


Figure 5.11: Creation of phone level transcriptions using HVite tool

5.2.4 Results and Discussion

In this work, three set of features have been undertaken, *i.e.*, MFCCs, MFCCs and their first derivative (Delta), and, MFCCs and their first and second derivative (Double Delta). The BW algorithm has been applied to search optimum model parameters. The results have been obtained using the HTK toolkit. Tables 5.1-5.3 show the average log likelihood values for twenty words of Hindi database using MFCC features, MFCC and Delta features, and, MFCC, Delta and Double Delta features, respectively. It has been observed from results that average log likelihood values using MFCC, Delta and Double Delta features is better than its counterparts for every considered word.

Table 5.1: Average value of log likelihood using BW algorithm with MFCC features

Word	*lP*	Word	*lP*	Word	*lP*	Word	*lP*
shoonya	-292.37	paanch	-278.12	gaana	-296.23	paani	-345.86
ek	-303.45	cheh	-265.67	dekho	-296.89	kaam	-313.52
do	-267.54	saat	-325.78	baitho	-287.47	anek	-294.65
teen	-322.86	aath	-213.27	khaao	-324.68	achaar	-315.86
chaar	-284.67	nau	-323.83	raam	-317.92	jagat	-313.42

Table 5.2: Average value of log likelihood using BW algorithm with MFCC and Delta features

Word	*lP*	Word	*lP*	Word	*lP*	Word	*lP*
shoonya	-251.12	paanch	-269.25	gaana	-282.24	paani	-319.35
ek	-271.45	cheh	-261.64	dekho	-275.06	kaam	-305.77
do	-265.25	saat	-301.40	baitho	-267.93	anek	-281.57
teen	-318.07	aath	-196.24	khaao	-314.56	achaar	-287.33
chaar	-271.44	nau	-312.24	raam	-276.75	jagat	-291.37

Table 5.3: Average value of log likelihood using BW algorithm with MFCC, Delta and Double Delta features

Word	*lP*	Word	*lP*	Word	*lP*	Word	*lP*
shoonya	-230.31	paanch	-262.47	gaana	-275.37	paani	-311.73
ek	-267.37	cheh	-259.37	dekho	-271.26	kaam	-297.68
do	-264.64	saat	-298.65	baitho	-258.68	anek	-274.58
teen	-307.84	aath	-192.62	khaao	-306.85	achaar	-279.89
chaar	-263.89	nau	301.84	raam	-247.74	jagat	-284.48

5.3 EXPERIMENT 2: ISOLATED WORD RECOGNITION USING OPTIMIZED HMM

HMM is an effective tool for speech recognition. In speech recognition by HMM, two modules are mainly involved in the experiment, *i.e.*, training and recognition. Initially, HMM model is trained with training set and then test data is recognized by the trained model. So, the performance of training process is very important to improve the speech recognition rate. For speech recognition, each Hindi word has been uttered 50 times by two male and two female speakers. Three sets of features have been extracted, *i.e.*, MFCCs; MFCCs and Delta; MFCCs, Delta and Double Delta, separately. The model parameters of HMM consists of two matrices, transition probability matrix *A* and observational symbol probability matrix *B*. The size of matrix *A* is 'number of states-by-number of states' and size of matrix *B* is 'number of states-by-number of observations'.

The optimum model parameters can accurately represent the training utterances. Various researchers (Baum and Egon, 1967; Baum *et al.*, 1970; Baum, 1972) have applied BW algorithm to set model parameters. But, BW algorithm is a local search method so optimum solution is highly dependent on initial estimates of the model parameters. BW

algorithm can search global optimum solution if initial model parameters are near to global optimum solution. Rabiner and Juang (1993) have suggested segmental *K*-means segmentation to avoid this problem, but this procedure is computationally intensive. The application of global search optimization technique with BW algorithm can be one of the effective methods to search optimum model parameters. In this section, two global search techniques, *i.e.*, PPO and PCSO are integrated with BW algorithm to search optimum model parameters.

5.3.1 Particle Representation

The transition and emission probabilities are randomly initialized as global search algorithm particles. Table 5.4 shows the representation of particle *l*. The particle has two parts; the first part of particle represents transition probabilities and second part represents emission probabilities.

Table 5.4: l^{th} particle representation

Transition probabilities			Emission probabilities		
a_{11}	..	a_{1q}	b_{11}	..	b_{1M}
..	..	..	..	..	..
a_{q1}	..	a_{qq}	b_{q1}	..	b_{qM}

where q represents number of states and M represents number of observations.

5.3.2 Implementation of Optimized HMM With PPO

In this experiment, left-to-right HMM model has been undertaken. The model parameters (transition and emission probabilities) have been searched with hybrid optimization technique based on integration of PPO and BW algorithm. The particle representation for hybrid optimization technique is given in Table 5.4. The detail discussion regarding PPO is given in Section 3.3.1. Initial search is performed by PPO algorithm and global best solution obtained from PPO is given as input to BW algorithm for further improvement. The training of HMM is done with each prey particle and fitness function (5.5) is evaluated. Based on fitness function evaluation, global best and local best positions are selected. The position and velocity of prey and predator particles are updated as per PPO algorithm. During each iteration, the constraints on the transition and emission probabilities ((5.2) and (5.4)) are

satisfied. After updating the model parameters from PPO algorithm, BW algorithm is applied to updated global best solution obtained from PPO. The hybrid algorithm based on PPO and BW techniques to search optimum model parameters is given in Algorithm 5.1.

Algorithm 5.1: HMM trained by PPO with BW method

Step 1: Randomly generate position and velocity of swarm.

Step 2: Initialize the iteration counter as $k = 1$.

Step 3: Compute fitness function as given by (5.5) for each prey position.

Step 4: Update the predator velocity and position.

Step 5: Update the prey velocity and position.

Step 6: Set $k = k + 1$

Step 7: IF ($k \leq k^{max}$) THEN

GOTO step 3.

ENDIF

Step 8: Input the global best position of parameters to BW algorithm for further improvement.

Step 9: STOP

5.3.2.1 Parameter setting of PPO algorithm

To set parameters of PPO technique, the approach given in Section 3.3.3.1 has been used. The minimum and maximum values, step size and optimal value of parameters are given in Table 5.5. For each trial, maximum number of iterations is set to 50.

Table 5.5: Parameter range, step size and optimal value for PPO technique

Parameter	Minimum Value	Maximum Value	Step Size	Optimal Value
C_1	1.0	2.5	0.50	2.0
C_2	1.0	2.5	0.50	2.0
a_i	0.05	0.2	0.05	0.1
b_i	0.50	2.0	0.50	1.0
$pf_{\max}$	0.10	0.95	0.05	0.85

5.3.2.2 Results and discussion

In this experiment, three sets of features as discussed in sub-section 5.2.4have been considered. Tables 5.6-5.8 have presented the average log likelihood values for twenty words of Hindi database using MFCC; MFCC and Delta; MFCC, Delta and Double Delta features, respectively. After comparing the results given in Tables 5.6-5.8, it has been observed that average log likelihood values using MFCC, Delta and Double Delta features is better than its counterpart. A drawback of this approach is that it requires more training time as compared to BW algorithm. However, it is worth mentioning that training of ASR system is performed offline.

Table 5.6: Average value of log likelihood using PPO and BW algorithm with MFCC features

Word	*lP*	Word	*lP*	Word	*lP*	Word	*lP*
shoonya	-228.79	paanch	-264.36	gaana	-274.68	paani	-314.57
ek	-265.65	cheh	-258.84	dekho	-274.72	kaam	-294.63
do	-261.73	saat	-293.78	baitho	-263.75	anek	-279.47
teen	-314.83	aath	-202.87	khaao	-311.38	achaar	-281.68
chaar	-262.56	nau	-306.48	raam	-303.48	jagat	-287.52

Table 5.7: Average value of log likelihood using PPO and BW algorithm with MFCC and Delta features

Word	*lP*	Word	*lP*	Word	*lP*	Word	*lP*
shoonya	-225.64	paanch	-261.72	gaana	-272.37	paani	-304.57
ek	-263.48	cheh	-255.58	dekho	-268.94	kaam	-292.83
do	-258.33	saat	-286.79	baitho	-257.83	anek	-273.48
teen	-302.34	aath	-199.69	khaao	-302.55	achaar	-275.85
chaar	-257.68	nau	-298.38	raam	-287.85	jagat	-281.29

Table 5.8: Average value of log likelihood using PPO and BW algorithm with MFCC, Delta and Double Delta features

Word	*lP*	Word	*lP*	Word	*lP*	Word	*lP*
shoonya	-220.25	paanch	-256.67	gaana	-268.43	paani	-301.73
ek	-258.26	cheh	-245.82	dekho	-264.47	kaam	-283.32
do	-252.63	saat	-278.67	baitho	-252.68	anek	-267.48
teen	-292.43	aath	-196.96	khaao	-297.65	achaar	-265.85
chaar	-253.48	nau	-287.43	raam	-284.56	jagat	-278.12

5.3.3 Implementation of Optimized HMM With PCSO

In this experiment, PCSO and BW algorithms are integrated to search optimum model parameters. As discussed in Chapter 3, PCSO is a global search technique and it helps to explore the search area effectively. The initial search is performed by PCSO and after that, to fine tune the optimum parameters, BW algorithm is applied. The BW is a local search technique based on strong mathematical foundation. It requires few iteration to search nearby optimum solution. While implementing the integrated approach, the transition and emission probabilities have been taken as society particles. The representation of society particles is given in Table 5.4. The training of HMM is done with each society particle position and each particle position is updated based on evaluation of fitness function as given in (5.5). During each iteration, the constraints on the transition and emission probabilities ((5.2) and (5.4)) are satisfied. After training of HMM with PCSO, the civilized leader of the society represents the optimized parameters of the HMM, *i.e.*, transition and emission probabilities. These optimum parameters are further optimized by BW algorithm. The procedure to select number of states in HMM model is same as discussed in Section 5.2.3. The hybrid algorithm based on PCSO and BW techniques to search optimum model parameters is given in Algorithm 5.2.

Algorithm 5.2: HMM trained by PCSO with BW algorithm

1. Read training data, expected outputs, parameters of algorithm, and set maximum number of iterations $k^{\max}$.
2. Randomly initialize transition and emission probabilities as society and predator positions.
3. Initialize society and predator velocity randomly.
4. Initialize iteration index $k = 1$.
5. Compute fitness function as given in (5.5) for each society particle position.
6. Arrange society particles on the basis of fitness function, and best performing particle is selected as society leader and remaining particles are treated as society members.
7. Compute the Euclidean distance between society members and society leader; and select society members for a particular society.
8. Select civilized leader among society leaders on the basis of fitness function.
9. Randomly generate probability fear.
10. Update predator particle velocity and position.

11. Update society leader and society member's velocity and positions.
12. Update civilized leader velocity and position.
13. Update personal best positions.
14. Set $k = k + 1$
15. IF ($k \leq k^{max}$) THEN
    GOTO step 5.
    ENDIF
16. The civilized leader position represents global best optimum solution.
17. Input the civilized leader position to BW algorithm for further improvement.
18. STOP

5.3.3.1 Parameter setting for PCSO algorithm

To set the parameters of PCSO technique, the approach given in Section 3.2.2.1 has been used. The minimum and maximum values, step size and optimal values of parameters are given in Table 5.9. For each trial, maximum number of iterations is set to 50.

Table 5.9: Parameter range, step size and optimal value for PCSO algorithm

Parameter	Minimum value	Maximum value	Step size	Optimal value
N_s	2	10	1	4
$(C_{SL1}, C_{SL2}, C_{SM1}, C_{SM2}, C_{L1})$	0.25	2.0	0.25	(0.5, 0.5, 0.25, 0.50, 2.0)
C_{L2}	0.0	2.0	--	--
a_i	0.25	1.0	0.25	0.25
b_i	0.25	1.0	0.25	0.50
$pf_{\max}$	0.50	1.0	0.05	0.95

5.3.3.2 Results and discussion

The HMM is trained using PCSO with BW algorithm using MFCC; MFCC and Delta; MFCC, Delta and Double Delta features, respectively. The average log likelihood has been computed and given in Tables 5.10-5.12 for the database as used in sub-section 5.2.1. It is evident from all the experiments that set of MFCC, Delta and Double Delta features produce better results. It is also evident from the results given in Tables 5.1, 5.6 and 5.10 that HMM

model gives good results when trained with PCSO and BW algorithm for MFCC features. The same observations have been found after comparing the results given in Tables 5.2, 5.7 and 5.11; and Tables 5.3, 5.8 and 5.12 for MFCC and Delta; MFCC, Delta and Double Delta features, respectively.

Table 5.10: Average value of log likelihood using PCSO and BW algorithm with MFCC features

Word	*lP*	Word	*lP*	Word	*lP*	Word	*lP*
shoonya	-223.45	paanch	-254.35	gaana	-271.24	paani	-299.36
ek	-259.68	cheh	-247.67	dekho	-266.74	kaam	-284.57
do	-257.83	saat	-281.23	baitho	-257.67	anek	-266.37
teen	-297.45	aath	-198.78	khaao	-296.75	achaar	-267.83
chaar	-256.84	nau	-289.73	raam	-284.64	jagat	-277.63

Table 5.11: Average value of log likelihood using PCSO and BW algorithm with MFCC and Delta features

Word	*lP*	Word	*lP*	Word	*lP*	Word	*lP*
shoonya	-218.36	paanch	-246.68	gaana	-263.26	paani	-284.48
ek	-255.67	cheh	-242.58	dekho	-261.29	kaam	-275.26
do	-251.28	saat	-274.27	baitho	-243.42	anek	-258.73
teen	-288.79	aath	-188.46	khaao	-284.68	achaar	-261.89
chaar	-246.96	nau	-279.49	raam	-276.49	jagat	-272.84

Table 5.12: Average value of log likelihood using PCSO and BW algorithm with MFCC, Delta and Double Delta features

Word	*lP*	Word	*lP*	Word	*lP*	Word	*lP*
shoonya	-212.28	paanch	-242.47	gaana	-257.68	paani	-282.24
ek	-248.26	cheh	-239.39	dekho	-254.28	kaam	-270.17
do	-247.85	saat	-268.83	baitho	-238.74	anek	-249.90
teen	-280.93	aath	-185.57	khaao	-276.89	achaar	-255.69
chaar	-241.57	nau	-273.35	raam	-272.38	jagat	-263.45

To compare the performance of BW algorithm with hybrid techniques, *i.e.*, PPO with BW algorithm, PCSO with BW algorithm, 100 utterances have been randomly chosen from Hindi database and speech recognition rate is compared. The speech recognition rate obtained by HMM trained with hybrid techniques and BW algorithm for all considered features are

given in Table 5.13. The highest recognition rate of 96.3% has been obtained, when MFCC and its first and second derivatives are considered as feature vector. So, it is concluded that a better speech recognition rate can be obtained with this feature set.

Table 5.13: Recognition rates using HMM with different features for isolated Hindi words

Features	Recognition rates (%)		
	BW algorithm	PPO and BW algorithm	PCSO and BW algorithm
MFCC	82.4	92.3	94.2
MFCC and Delta	85.8	93.7	94.9
MFCC, Delta andDouble Delta	86.6	94.8	96.3

5.4 EXPERIMENT 3: RECOGNITION OF HINDI SENTENCES

In this section, Hindi sentences have been recognized using HMM trained with three techniques, namely, BW algorithm, PPO with BW algorithm, and PCSO with BW algorithm. The database used for experimentation, implementation details and results and discussion have been discussed in further sub-sections.

5.4.1 Database Used

In this work, self-recorded Hindi sentences database has been considered for experimentation. The database consists of 16 Hindi sentences with 4 utterances each, spoken by two male and two female speakers. The data is recorded in a quiet room environment at a sampling rate of 44.1 kHz. The snapshot of Hindi sentences is given in Figure 5.12.

5.4.2 Implementation

In this section, implementation of speech recognition system for Hindi sentences has been presented. The recognition system has been developed using HTK toolkit v3.4. Firstly, the HTK training tools are used to estimate the parameters of a set of HMMs using training utterances and their associated transcriptions. Secondly, unknown utterances are transcribed using the HTK recognition tools. System is trained for 16 Hindi sentences. For recognition purpose, a sentence is to be modeled in a sequence of sub-word units, such as, words or

phonemes. In this work, word model is used and each word is represented by a continuous density HMM with transition and emission probabilities. A left-to-right topology has been used and the BW algorithm is used for computing the transition and emission probabilities of HMM. In decoding, the Viterbi algorithm is used to find the sequence of words having the highest probability. In the second technique the transition probability and emission probability of HMM have been optimized using PPO technique. These optimized probabilities are given to HMM. In the third technique the transition probability and emission probability of HMM have been optimized using PCSO technique. These optimized probabilities are given to HMM. Remaining implementation details are same as discussed in Section 5.2.3.

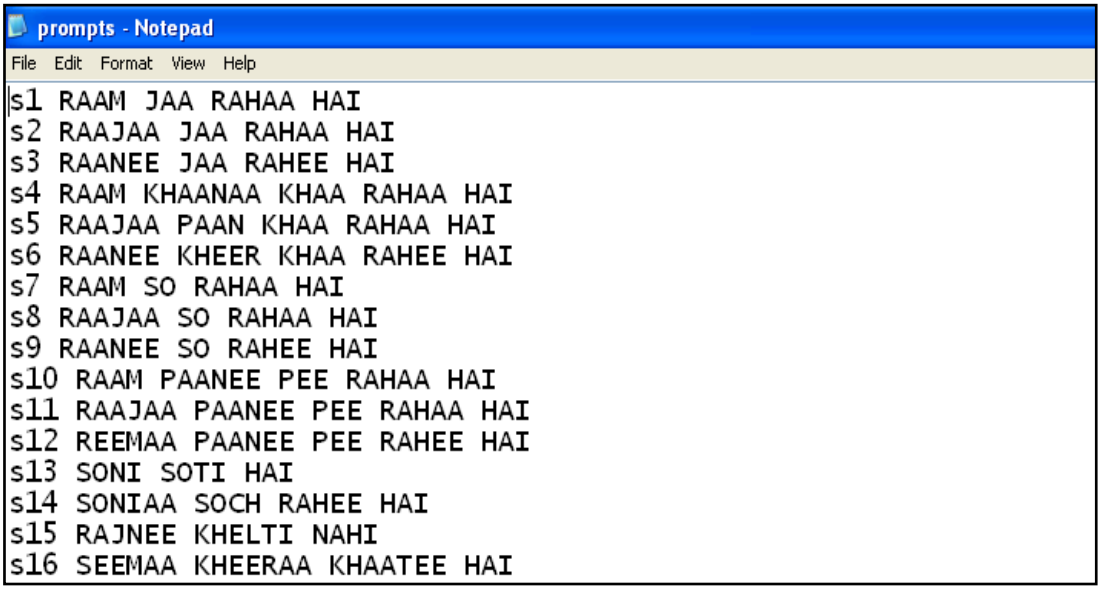


Figure 5.12: Snapshot of Hindi sentences

5.4.3 Results and Discussion

Three sets of features have been undertaken in this work, *i.e.*, MFCCs; MFCCs and Delta; and MFCCs, Delta and Double Delta. The BW algorithm has been applied to search optimum model parameters. The results obtained from BW algorithm, PPO with BW algorithm and PCSO with BW algorithm are given in Tables 5.14-5.16.It has been observed from Table 5.14 that a recognition rate of 89.4% for words and 34.7% for sentences has been obtained using BW algorithm with MFCC, Delta and Double Delta features. Table 5.15 shows that a recognition rate of 91.4% for words and 48.4% for sentences has been achieved using PPO and BW algorithm with MFCC, Delta and Double Delta features. One can also observe from

Table 5.16 that a recognition rate of 95.8% for words and 54.7% for sentences has been achieved using PCSO and BW algorithm with MFCC, Delta and Double Delta features.

Table 5.14: Recognition rates for different features with BW algorithm

Features	Word recognition rate (%)	Sentence recognition rate (%)
MFCC	82.4	23.6
MFCC and Delta	86.2	26.2
MFCC, Delta and Double Delta	89.4	34.7

Table 5.15: Recognition rates for different features with PPO and BW algorithm

Features	Word recognition rate (%)	Sentence recognition rate (%)
MFCC	85.3	43.1
MFCC and Delta	90.8	44.2
MFCC, Delta and Double Delta	91.4	48.4

Table 5.16: Recognition rates for different features with PCSO and BW algorithm

Features	Word recognition rate (%)	Sentence recognition rate (%)
MFCC	87.4	45.6
MFCC and Delta	92.6	49.2
MFCC, Delta and Double Delta	95.8	54.7

To detect the significant differences between two samples means for PCSO with BW technique along with PPO with BW and BW techniques, Wilcoxon signed rank test has been performed. Each algorithm is run thirty times for Hindi database considering MFCC, Delta and Double Delta features to recognize word. The test is performed by taking a level of significance $\alpha=0.01$. It is observed from p-value reported in Table 5.17, that PCSO with BW technique is significantly better than PPO with BW and BW technique.

Table 5.17: Wilcoxon signed rank test results. PCSO with BW technique versus PPO with BW and BW technique

	PPO with BW	BW
p-value	0.0000875	0.0000875

5.5 CONCLUSIONS

In this chapter, Hindi speech recognition system for isolated words and continuous speech with limited vocabulary has been developed using HMM. To carry out this, Hidden Markov Model toolkit (HTK) designed for speech recognition is used. The task grammar and pronunciation dictionary for Hindi language has been build for this purpose. For speech recognition, three sets of acoustic features, MFCCs; MFCCs and Delta; and MFCCs, Delta and Double Delta, have been considered which consists of 13, 26 and 39 coefficients, respectively. In this work, three experiments have been conducted. In first experiment, HMMs have been trained with BW algorithm. To improve the performance of BW algorithm during training process of HMM, the BW algorithm is integrated with global optimization technique. In second experiment, PPO and PCSO techniques have been integrated with BW algorithm. These integrated techniques have been applied to search HMM model parameters, *i.e.*, transition and emission probabilities. These experiments have been tested on isolated Hindi words and continuous Hindi speech. To evaluate the performance, average log likelihood values have been computed during training process. It is concluded from results that HMM model gives better results when trained with PCSO and BW algorithm with MFCC and its first and second derivatives as feature vector. For testing purpose, 100 utterances of isolated words have been randomly chosen from Hindi database and speech recognition rate is compared. The highest recognition rate of 96.3% has been obtained, when MFCC and its first and second derivatives are considered as feature vector. So, it is concluded that a better speech recognition rate is possible with a proper training. In Experiment 3, Hindi sentences have been recognized. It has been found that highest recognition rate of 95.8% for words and 54.7% for sentences have been achieved using PCSO and BW algorithm with MFCC, Delta and Double Delta features.

Chapter 6

Optimized Hybrid Classification Models for Speech Recognition

In this chapter, optimized hybrid classification models have been explored for Hindi speech recognition. The credibility of ANN and SVM classifiers has already been proved for pattern classification. Both the classifiers have sound theoretical foundations and use a discriminative approach for classification. In the field of speech recognition, the application of ANN and SVM classifiers is mainly limited to recognition of isolated words. The ANNs are not found suitable to cover dynamic aspects of speech recognition and one of the main limitations of SVM in the field of speech recognition is its ability to classify only fixed length data vectors.To overcome the limitations of ANN and SVM classifiers, HMM classifier has been integrated with these classifiers in this research work.The HMM classifier has a great capability of dealing with the variability in speech signal. As such, it has emerged as an efficient tool for continuous speech recognition.

In the optimized hybrid model of ANN with HMM classifier, the ANN is trained to estimate the posterior probabilities of HMM states so as to optimize the posterior probabilities of a Markov model. For training of ANN, hybrid optimization technique based on integration of predator prey optimization (PPO) with Hooke-Jeeves method is applied. In optimized hybrid SVM-HMM model, posterior probabilities of HMM are computed from SVM. The RBF kernel has been undertaken and its parameters are optimized using PPO with Hooke-Jeeves optimization technique. The performance of hybrid models have been tested on isolated Hindi words and Hindi sentences.

6.1 HYBRID ANN-HMM SYSTEM

The main objective of a continuous speech recognition system is to arrange the words according to the spoken utterances. Recognition of continuous speech is a complex task, because word boundaries in continuous speech are not clear. Another important issue with

continuous speech is dominance of co-articulatory effects (Mohamed and Nair, 2012). In last decade, researchers have explored the ANN to develop automatic speech recognition (ASR) systems and it has been found that ANN techniques are suitable where the variety and separability of the speech patterns are important. However, ANNs by themselves are not found very suitable for large scale recognition of continuous speech. The Hidden Markov Model is an efficient tool for continuous speech recognition which can deal with the variability in speech signal. One of the important features of HMM is its capability to model any speech units, such that each parameter of it can be estimated based on training data (Bahl *et al*., 1983; Rabiner and Juang, 1986). However, discriminative learning algorithm of HMM makes it suitable only for small problems.

Considering these facts, Bengio *et al.* (1992) suggested that a combination of ANN and HMM classifiers can improve speech recognition rate. In Hybrid ANN-HMM system, emission probabilities for HMM are generated by ANN, instead of Gaussian Mixtures. The benefit of this combination is two fold: (i) the combination of HMM with ANN will add some dynamic features to the ANN so that it is capable of handling static pattern recognition problems as well as dynamic speech recognition problems with the same accuracy and (ii) the combination will help HMM to get better preprocessing capabilities so that it is more capable of transforming an acoustic speech pattern.

6.1.1 Estimation of Posterior Probabilities from ANN

The ANN is trained to estimate the posterior probabilities of HMM states so as to maximize the posterior probability of a left-to-right Markov model λ^{v}, given an acoustic observation sequence O. For a speech recognition problem, the maximum posterior criterion has been adopted to find the sequence of words $\tilde{W}$ which maximizes the quantity $P(W \mid O)$, where O is the sequence of input observation features. The quantity $P(W \mid O)$ is factorized using Bayes' theorem, as:

$$P(W|O) = \frac{P(O|W)P(W)}{P(O)} \tag{6.1}$$

where $P(W)$ is prior probability and $P(O|W)$ are the likelihoods estimated by the HMMs.

6.1.2 Implementation of Optimized ANN-HMM Hybrid System

The implementation of an optimized ANN-HMM hybrid speech recognition system is presented in this section. The block diagram of a speech recognition system using optimized ANN-HMM hybrid model is shown in Figure 6.1.Initially, the pre-processing of speech signal has been done as discussed in Section 3.2.1. The extracted acoustic features are given as input to the ANN classification module.The PPO with Hooke-Jeeves method has been applied to search optimum weights and biases of ANN, the details have been discussed in Section 3.3. The emission probabilities are computed from optimized ANN. The outputs of ANN have been interpreted as estimates of posterior probabilities of output classes conditioned on the input. In the hybrid model, the posterior probabilities are used to estimate the state emission probabilities of HMM by applying Bayes' theorem, *i.e.*, by dividing the posterior estimates from the ANN outputs by estimates of class priors. Training of other probabilistic quantities in the HMM, the initial and transition probabilities, has been done using BW algorithm. The Viterbi algorithm is applied to obtain final recognition rate as discussed in Section 5.2.3.

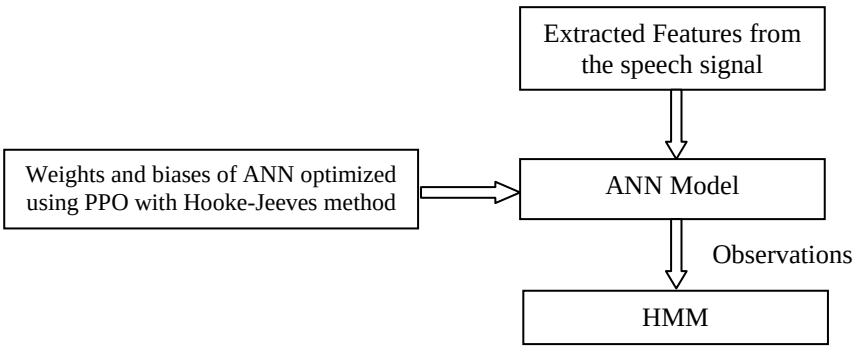


Figure 6.1: Block diagram of optimized ANN-HMM hybrid model for speech recognition

In this work, two databases, a self-recorded isolated Hindi words database and a self-recorded Hindi sentences database, have been considered for experimentation. The isolated Hindi words database consists of 20 Hindi words, as given in Table 4.1, with 50 utterances each, spoken by two male and two female speakers. The self-recorded Hindi sentences database consists of 16 Hindi sentences with 4 utterances each, spoken by two male and two female speakers.

The performance of the system has been obtained by computing the word error rate (WER). The WER can be computed as:

$$WER = \frac{S + D + I}{N} \tag{6.2}$$

where$S$ is the number of substitutions, D is the number of the deletions, I is the number of the insertions and N is the number of words in the reference. The word recognition rate (WRR) is given by $(1 - WER)$. The sentence recognition rate (SRR) is defined as total number of sentences correctly identified by the recognizer and is computed by dividing the number of sentences recognized correctly by the total number of sentences in the test set.

6.1.3 Results and Discussion

In this experiment, MFCC, Delta and Double Delta features as discussed in Section 5.2.4 have been considered. During training phase, average log likelihood values are computed as given in Section 5.2.2.Table 6.1 presents the average log likelihood values for isolated twenty words of Hindi database using optimized ANN-HMM classifier.

Table 6.1: Average values of log likelihood for isolated Hindi words database using optimized ANN-HMM classifier

Word	*lP*	Word	*lP*	Word	*lP*	Word	*lP*
shoonya	-221.53	Paanch	-258.36	gaana	-257.85	paani	-307.27
ek	-225.32	Cheh	-243.92	dekho	-258.38	kaam	-284.57
Do	-246.78	Saat	-274.42	baitho	-254.49	anek	-263.56
Teen	-287.27	Aath	-211.01	khaao	-294.48	achaar	-264.23
Chaar	-255.37	Nau	-263.27	raam	-273.24	jagat	-276.56

To compute the speech recognition rate, 100 utterances have been randomly chosen from isolated Hindi words database. The recognition rate has been computed using HMM, ANN-HMM and optimized ANN-HMM classifier. Table 6.2 shows the recognition rates for isolated Hindi words database and Hindi sentences database. A recognition rate of 96.0% has been obtained for isolated words with optimized ANN-HMM hybrid system. For the Hindi sentences database, 50 utterances have been taken for testing and recognition rate of 92.6% for words and 48.6% for sentences has been achieved by optimized ANN-HMM classifier.

Wilcoxon signed rank test has also been applied to establish the differences between samples means for optimized ANN-HMM technique, with ANN-HMM and HMM techniques. Each algorithm is run thirty times to recognize Hindi words. The test is performed by taking the level of significance α as 0.01. It is observed from *p*-values reported

in Table 6.3 that optimized ANN-HMM technique is significantly better than ANN-HMM and HMM techniques.

Table 6.2: Hindi words and sentences recognition rates using different classifiers

Database	HMM classifier		ANN-HMM classifier		Optimized ANN-HMM classifier	
	WRR (%)	SRR (%)	WRR (%)	SRR (%)	WRR (%)	SRR (%)
Isolated Hindi words	86.6	-	89.0	-	96.0	-
Hindi Sentences	89.4	34.7	90.6	37.8	92.6	48.6

Table 6.3: *p*-values for Wilcoxon signed rank test results.Optimized ANN-HMM versus ANN-HMM and HMM technique

	ANN-HMM	HMM
p-value	0.00008745	0.00008609

6.2 HYBRID SVM-HMM SYSTEM

Support vector machine is one of the most popular approaches that use a discriminative approach. It has sound theoretical foundation and high generalizing performance (Doumpos *et al.*, 2007). One of the main limitations of SVM is its ability to classify only fixed length data vectors so it cannot model the temporal structure of speech effectively. To overcome the limitations of SVM, a hybrid SVM-HMM classifier has been proposed. The motive behind the hybrid SVM-HMM approach is to enhance the discrimination abilities of HMM to improve speech recognition rate. The posterior probabilities of HMM have been computed from output of SVM. The details regarding computation of posterior probabilities from SVM is discussed in next sub-section.

6.2.1 Estimation of Posterior Probabilities from SVM

In SVM, there is a no clear relationship given between distance from the margin and the posterior class probability. Researchers have proposed various approaches to compute posterior probabilities from SVM, *i.e.*, Gaussian fits and histogram approaches. But these methods are not Bayesian in nature so they do not represent the variability in the estimates of the SVM parameters (Tipping, 2000). Kwok (1999) and Platt (1999) have extensively studied the use of moderated SVM outputs as estimates of the posterior probability. In this work, SVM classifier is used to generate posterior probabilities during training phase. A sigmoid

function is used to transform the distance into condition probability $P(j \mid x)$ and then Bayes' theorem is applied to get HMM emission probability $P(x \mid j)$ as:

$$P(j \mid x) = \frac{1}{1+\exp(-k\ f_j(x))} \tag{6.3}$$

$$P(x \mid j) = \frac{P(j \mid x)P(x)}{P(j)} \tag{6.4}$$

where $P(j)$ is the probability of the class j and f_j is the function that identifies the class j from the others and that returns the distance from the margin.

6.2.2 Implementation of Optimized SVM-HMM Hybrid System

Implementation of the optimized SVM-HMM hybrid speech recognition system is presented in this section. The block diagram of a speech recognition system using optimized SVM-HMM hybrid model is shown in Figure 6.2. Initially, the pre-processing of speech signal has been done as discussed in Section 3.2.1. After pre-processing, speech features have been extracted. The extracted features have been given as input to SVM classifier. In this work, RBF kernel has been undertaken and SVM hyper-parameters, *i.e.*, penalty parameter *C* and kernel parameter γ, are searched by applying PPO with Hooke-Jeeves method. The detailed discussion regarding this is given in Section 4.2.

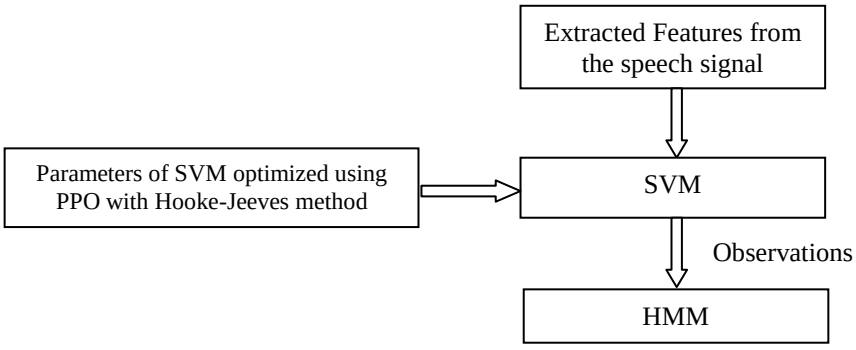


Figure 6.2: Block diagram of optimized SVM-HMM hybrid model for speech recognition

In the optimized SVM-HMM hybrid system, SVM estimates the emission probabilities of HMM. Training of other probabilistic quantities in the HMM, the initial and transition probabilities, has been done using BW algorithm. The Viterbi algorithm is applied to get final recognition rates as discussed in Section 5.2.3.

6.2.3 Results and Discussion

The two databases, a self-recorded isolated Hindi words database and a self recorded Hindi sentences database, have been considered for experimentation. The details regarding these databases are given in Section 6.1.2. In this experiment, MFCC, Delta and Double Delta features of speech signal, as discussed in Section 5.2.4, have been considered. Table 6.3 presents the average log likelihood values for twenty isolated words of Hindi database using optimized SVM-HMM classifier.

Table 6.4: Average values of log likelihood for isolated words Hindi database using optimized SVM-HMM classifier

Word	*lP*	Word	*lP*	Word	*lP*	Word	*lP*
shoonya	-220.27	paanch	-254.83	gaana	-253.27	paani	-298.12
ek	-221.58	cheh	-242.67	dekho	-247.19	kaam	-282.26
do	-242.63	saat	-273.15	baitho	-253.37	anek	-260.38
teen	-277.16	aath	-206.33	khaao	-266.83	achaar	-255.51
chaar	-246.94	nau	-261.44	raam	-248.35	jagat	-268.16

To compute speech recognition rates, 100 utterances have randomly been chosen from Hindi database. The recognition rate is computed by using HMM, SVM-HMM and optimized SVM-HMM classifiers. The recognition rate of 98.0% has been obtained with optimized SVM-HMM hybrid system. For Hindi sentences database, 50 utterances have been taken for testing and a recognition rate of 93.8% for words and 54.2% for sentences have been achieved with optimized SVM-HMM classifier as presented in Table 6.4.

Table 6.5: Hindi words and sentences recognition rates using different classifiers

Database	HMM classifier		SVM-HMM classifier		Optimized SVM-HMM classifier	
	WRR (%)	SRR (%)	WRR (%)	SRR (%)	WRR (%)	SRR (%)
Isolated Hindi words	86.6	-	95.0	-	98.0	-
Hindi Sentences	89.4	34.7	93.0	47.3	93.8	54.2

6.3 ASR SYSTEM INTERFACE

In this work, an interface has also been developed for speech recognition system using MATLAB tool. The snapshot of this interface is shown in Figure 6.3. This interface window

is having a “record” button, to record a speech signal; a “save” button, to save the recorded signal; a “play” button, to play a recorded speech; a “load” button, to load a already saved speech signal; a “predict” button, to recognize a speech signal. The interface window also shows the waveform of speecg signal and its spectrogram. The interface window also consists of a length bar to display the length of the speech signal. A text box is also there in the window to display the predicted result.

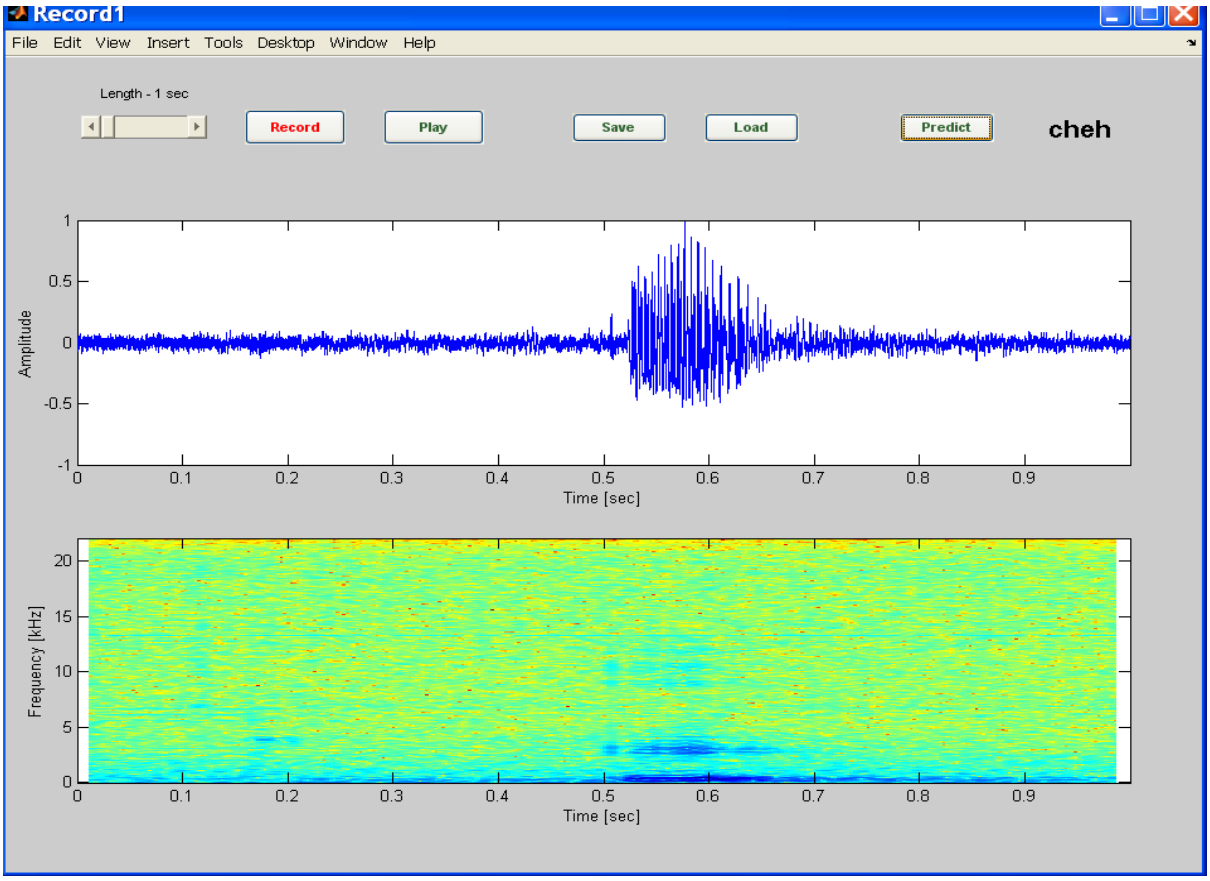


Figure 6.3: ASR system interface

6.4 CONCLUSIONS

In this chapter, two hybrid classifiers, *i.e.*, optimized ANN-HMM and optimized SVM-HMM have been explored for recognition of isolated Hindi words and Hindi sentences. To improve the performance of ANN-HMM and SVM-HMM models, PPO with Hooke-Jeeves method has been applied to search optimum weights and biases of ANN and optimum value of hyper-parameters of SVM, respectively. In hybrid classifiers, ANN and SVM classifier are used to estimate the posterior probabilities of HMM states. The MFCC, Delta and Double Delta speech features have been used. The recognition rates achieved for isolated Hindi words by

optimized ANN-HMM classifier and optimized SVM-HMM classifiers are 96.0% and 98.0%, respectively. For Hindi sentences database, recognition rates achieved by optimized ANN-HMM classifier are 92.6% for words and 48.6% for sentences. The optimized SVM-HMM classifier proposed in this work has produced an accuracy of 93.8% for words and 54.2% for sentences.

Chapter 7
Conclusions and Future Scope

This chapter presents salient and significant contributions of the work carried out in this thesis as well as a brief statement including recommendations for further research.In this thesis, an attempt has been made to improve the efficiency of speech recognition systems.For this purpose, optimization techniques have been explored to improve the performance of various classifiers. In this work, optimization techniques have also beenexplored to select most appropriate speech feature set. Proposed methodologies have been implemented to recognize isolated Hindi words database, TI-46 database and Hindi sentences database. Section 7.1 contains a brief illustration on the contributions of the work and Section 7.2 lists some directions for the further research on this topic.

7.1 SIGNIFICANT CONTRIBUTIONS

This thesis has made the following contributions in the field of Hindi speech recognition:

(viii) Two databases, namely, Hindi speech words databaseand Hindi sentences database have been prepared in this work. The Hindi words database consists of twenty words with fifty utterances of each word spoken bytwo male and two female speakers. The Hindi sentences database consists of sixteen sentences with four utterances of each sentence spoken bytwo male and two female speakers. The recording has here been done in a quiet room environment with sampling frequency of 44.1 kHz for both the databases.

(ix) To search optimum weights and biases of ANN, two optimization techniques have been proposed. Firsttechnique is predator influenced civilized swarm optimization (PCSO) in which swarm particles are divided into a number of societies and global best particle of the swarm is chased by predator particle. The predator effect helps to exploit the search area more effectively. Second

technique is based on integration of global and local search techniques. In this technique,predator prey optimization (PPO) has been considered as the global search technique and Hooke-Jeeves method is undertaken as local search technique. In predator prey optimization with Hooke-Jeeves method (PPO-HJ),initial search is performed by PPO technique and in order to further enhance the search, global best solution obtained from PPO is given as input to Hooke-Jeeves method.

(x) For SVM classifier, the hyper-parametershave been optimized by proposed PPO-HJ technique.The recognition rates of 81.5%, 92.2% and 90.3% have been achieved for Hindi, TI-20 and TI-ALPHA speech databases, respectively under clean environment by applying PPO-HJ method.

(xi) A mixed variable PPO (MVPPO) technique has also been proposed in this work. The mixed variable PPO with Hooke-Jeeves (MVPPO-HJ) method is applied for the selection of an appropriate feature set and also for the selction of optimized hyper-parameters. The recognition rates of 93.4%, 98.8% and 96.6% have been achieved for Hindi, TI-20 and TI-ALPHA speech databases, respectively under clean environment.

(xii) For training of HMM classifier, PPO and PCSO optimization techniques have been integrated with BW algorithm.We are able to achieve a recognition rate of 95.8% for words and 54.7% for sentences employing PCSO and BW algorithm with MFCC, Delta and Double Delta features.

(xiii) For continuous speech recognition, two hybrid classifier models have been proposed. These areoptimized ANN-HMM and optimized SVM-HMM classifiers. In the optimized ANN-HMM hybrid model, the weights and biases of ANN are optimized with PPO-HJ technique and output of ANN is used to estimate the posterior probabilities of HMM. In optimized SVM-HMM hybrid model, SVM hyper-parameters are optimized with PPO-HJ technique and posterior probabilities of HMM are computed from SVM.The recognition rates achieved for isolated Hindi words by optimized ANN-HMM classifier and optimized SVM-HMM classifiers are 96.0% and 98.0%, respectively. For Hindi sentences database, recognition rates achieved by optimized ANN-HMM classifier are 92.6% for words and 48.6% for sentences, respectively. The optimized SVM-HMM classifier proposed in this work has produced an accuracy of 93.8% for words and 54.2% for sentences.

(xiv) In this work, an Interface has also been developed for speech recognition system using MATLAB tool.

7.2 FUTURE SCOPE

No work is complete in itself. We have realized that attempts can be made in the following directions to improve the work presented in this thesis.

(i) **Enhancement of database**: The current work can be expanded in future by increasing the vocabulary of Hindi words and as well as Hindi sentences.

(ii) **Enhancing number of speakers**: we have included four speakers in creating Hindi words and sentences database. One canextend this work by including more number of speakers.

(iii) **Speaker independent system:**The recognition rates reported in this work are based on the four speakers who contributed in the creation of databases. One can extend this work by including a larges number of speakers and then obtaining the speaker independent recognition results.

(iv) **Hybridization of other techniques:** In order to increase the accuracy of ASR system, integration of global optimization techniques can also be explored to optimize the parameters of different classifiers. Fuzzy logic based techniques to optimize the parameters of different classifiers can be explored to improve the efficiency of ASR systems.

REFERENCES

1. Abido, M.A., 2002. Optimal power flow using particle swarm optimization.International Journal of Electrical Power and Energy Systems 24(7), 563-571.
2. Agarwalla, S., Sarma, K.K., 2016. Machine learning based sample extraction for automatic speech recognition using dialectal Assamese speech. Neural Networks 78, 97-111.
3. Ahad, A., Fayyaz, A., Mehmmod, T. 2002. Speech recognition using multilayer perceptron. In: Proceedings of IEEE Students Conference, 2002, ISCON'02, Lahore, Pakistan, Aug.16-17, 2, 103-109.
4. Aida-Zade, K.R., Ardil, C., Rustamov, S.S. 2006. Investigation of combined use of MFCC and LPC Features in speech recognition systems. World Academy of Science, Engineering and Technology 19, 74-80.
5. Al-Ani, A., Alsukker, A., Khushaba, R.N. 2013. Feature subset selection using differential evolution and a wheel based search strategy. Swarm and Evolutionary Computation 9(April), 15-26.
6. Al-Batah, M.S., Mat Isa, N.A., Zamli, K.Z., Azizli, K.A. 2010. Modified recursive least squares algorithm to train the hybrid multilayered perceptron network. Applied Soft Computing 10(1), 236-244.
7. Allwein, E., Schapire, R.E., Singer, Y. 2000. Reducing multiclass to binary: A unifying approach for margin classifiers. Journal of Machine Learning Research 1(Dec), 113-141.
8. Anderson, S., Merrill, J., Port, R. 1998. Dynamic speech categorization with recurrent networks. Technical Report TR258, Department of Linguistic, Department of Computer Science, Indiana University, August.
9. Anumanchipalli, G., Chitturi, R., Joshi, S., Kumar R., Singh S.P., Sitaram, R.N.V., Kishore, S.P. 2005. Development of Indian language speech database for large vocabulary speech recognition systems. In: Proceedings of International Conference on Speech and Computer SPECOM, Patras, Greece, Oct.
10. Anusuya, M.A., Katti, S.K. 2011. Comparison of different speech feature extraction techniques with and without wavelet transform to Kannada speech recognition. International Journal of Computer Applications, 26(4), 19-23.

11. Anusuya, M.A., Katti, S.K. 2011. Front end analysis of speech recognition: a review. International Journal of Speech Technology 14(2), 99-145.
12. Atal B. S. The history of linear prediction, IEEE Signal Processing Magazine. 23, 154-161.
13. Avci, E., Akpolat, Z.H. 2006. Speech recognition using a wavelet packet adaptive network based fuzzy inference system. Expert Systems with Applications 31(3), 495-503.
14. Bahl Lalit R., Jelinek, F., Mercer, R. 1983. A maximum likelihood approach to continuous speech recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 5(2), 179-190.
15. Bao, Y., Hu, Z., Xiong, T. 2013. A PSO and pattern search based memetic algorithm for SVMs parameters optimization. Neurocomputing 117(October), 98-106.
16. Bao, Y., Xiong, T., Hu, Z. 2014a. Multi-step-ahead time series predictionusing multiple-output support vector regression. Neurocomputing 129(April),482-493.
17. Bao, Y., Xiong, T., Hu, Z. 2014b. PSO-MISMO modeling strategy formultistep-ahead time series prediction. IEEE Transactions on Cybernetics 44(5), 655-668.
18. Baum, L.E., Egon, J.A. 1967. An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology. Bulletin of the American Meteorological Society, 73, 360-363.
19. Baum, L.E., Petrie, T., Soules, G., Weiss, N. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. The Annals of Mathematical Statistics, 41(1), 164-171.
20. Baum, L.E. 1972. An inequality and associated maximization technique in statistical estimation for probabilistic functions of markov process. Inequalities, 3, 1-8.
21. Bengio, Y., De Mori, R., Flammia G., Kompe R., 1992. Global optimization of a neural network-Hidden Markov Model hybrid. IEEE Transactions on Neural Networks, 3(2), 252-259.
22. Bermejo, P., Jose, A.G., Puerta, J.M. 2011. A GRASP algorithm for fast hybrid (filter-wrapper) feature subset selection in high-dimensional datasets. Pattern Recognition Letters 32(5), 701-711.
23. Bermejo, P., De la ossa, L., Gåmez, J.A., Puerta, J.M. 2012. Fast wrapper feature subset selection in high-dimensional datasets by means of re-ranking. Knowledge-Based Systems 25(1), 35-44.

24. Bhatt, A., Pant, D. 2009. Back propagation neural networks in financial analysis of stock market returns. Journal of Computer Science, Karpagam Jcs 4(1), 1354-1361.
25. Bhatt, A., Pant, D., Singh, R. 2014. An analysis of the performance of Artificial Neural Network technique for apple classification. AI & Society 29(1), 103-110.
26. Bilski, J., Rutkowski, L. 1998. A fast training algorithm for neural networks. IEEE Trans Circuits and Systems-II: Analog and Digital Signal Processing 45(6), 749-753.
27. Biswas, A., Sahu, P.K. Chandra M., 2014. Admissible wavelet packet features based on human inner era frequency response for Hindi constant recognition. Computers and Electrical Engineering 40(4), 1111-1122.
28. Blanco, A., Delgado, M., Pegalajar, M.C. 2001. A real-coded genetic algorithm for training recurrent neural networks. Neural Networks 14 (1), 93-105.
29. Burges, C.J.C. 1998. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2(2), 121-167.
30. Campbell, W.M., Sturim, D.E., Reynolds, D.A. 2006. Support vector machines using GMM supervectors for speaker verification. IEEE Signal Processing Letters 13(5), 308-311.
31. Campbell, W.M., Campbell, J.P., Reynolds, D.A., Singer, E., Torres-Carrasquillo, P.A. 2006. Support vector machines for speaker and language recognition. Computer Speech and Language 20(2-3), 210-229.
32. Castellani, A., Botturi D., Bicego M., Fiorini P. 2004. Hybrid HMM/SVM model for the analysis and segmentation of teleperation tasks. In: Proceedings of the IEEE International Conference on Robotics & Automation, New Orleans, LA, April, 2004, pp. 2918-2923.
33. Cerf, P.L., Compernolle, D.V. 1994. A new variable frame rate analysis method for speech recognition. IEEE Signal Processing Letters 1(12), 85-187.
34. Champion, C., Houghton, S.M., 2016. Application of continuous state hidden Markov models to a classical problem in speech recognition. Computer Speech and Languages 36, 347-364.
35. Chang, S., Kwon, Y., Yang, S. 1998. Speech feature extracted from adaptive wavelet for speech recognition. Electronics Letters 34(23), 2211-2213.
36. Chapaneri, S.V. 2012. Spoken digits recognition using weighted MFCC and improved features for dynamic time warping. International Journal of Computer Applications 40(3), 6-12.

37. Chau, C.W., Kwong S., Diu, C.K., Fahrner, W.R. 1997. Optimization of HMM by a genetic algorithm. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP-97, Munich, Apr. 21-24, vol. 3, 1727-1730.

38. Chaurasia, V., Samudravijaya, K., Chandwani, M. 2005. Phonetically rich Hindi sentence corpus for creation of speech database. In: Proceedings of International Symposium on Speech Technology and Processing Systems and Oriental COCOSDA-2005, Indonesia, Dec. 6-8, 132-137.

39. Chen, W.-Y., Chen, S.-H., Lin, C.-J. 1996. A speech recognition method based on the sequential multi-layer perceptrons. Neural Networks 9(4), 655-669.

40. Clarkson, P., Moreno, P.J. 1999. On the use of support vector machines for phonetic classification. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP99, Phoenix, AZ, USA, Mar.15-19, 2, 585-588.

41. Cristianini, N., Taylor, J.S. 2000. An Introduction to support vector machines and other kernel-based learning methods. Cambridge University Press.

42. Cui, X., Afify, M., Gao, Y., Zhou, B. 2013. Stereo hidden Markov modeling for noise speech recognition. Computer Speech and Language 27(2), 407-419.

43. Cutajar, M., Gatt, E., Grech, I., Casha, O., Micallef, J. 2013. Comparative study of automatic speech recognition techniques. IET Signal Processing 7(1), 25-46.

44. Das, G., Pattnaik, P.K., Padhy, S.K. 2014. Artificial neural network trained by particle swarm optimization for non-linear channel equalization. Expert Systems with Applications 41(7), 3491-3496.

45. David Forney, G.JR. 1973. The Viterbi Algorithm. In: Proceedings of the IEEE 61(3), 268-278.

46. de Andrade Bresolin, A., Dòria Neto A.D., Alsina, P.J. 2008. Digit recognition using wavelet and SVM in Brazilian Portuguese. In: Proceedings of IEEE International Conference on Acoustic, Speech and Signal Processing, ICASSP'2008, Las Vegas, NV, Mar. 31-Apr. 4,1545-1548.

47. de Andrade Bresolin, A., Doria Neto, A.D., Alsina, P.J. 2008. Digit recognition using wavelet and SVM in Brazilian Portuguese, In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, Las Vegas, USA, Mar. 31-Apr. 4, 1545-1548.

48. Debyeche, M., Haton, J.-P., Houacine, A. 2007. Improved vector quantization approach for discrete HMM speech recognition system. The International Arab Journal of Information Technology 4(4), 338-344.

49. Dede, G., Sazli, M.H. 2010. Speech recognition with artificial neural networks. Digital Signal Processing 20(3), 763-768.

50. Demšar, J. 2006. Statistical comparisons of classifiers over multiple datasets. Journal of Machine Learning Research 7, 1-30.

51. Derrac, J., Garcia, S., Molina, D., Herrera, F. 2011. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary ans swarm intelligence algorithms. Swarm and Evolutionary Computation 1, 3-18.

52. Deshmukh, N., Picone, J. 1995. Methodologies for language modeling and search in continuous speech recognition. In: Proceedings of IEEE Southeastcon 95, Visualize the Further, Raleigh, NC, Mar.26-29, 192-198.

53. Diebold, F.X., Mariano, R.S. 1995. Comparing predictive accuracy. Journal of Business and Economic Statistics 13(3), 134-144, 1995.

54. Doumpos, M., Zopounidis, C., Golfinopoulou, V. 2007. Additive support vector machines for pattern classification. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics 37(3), 540-550.

55. Duan, K.-B., Sathiya Keerthi, S. 2005. Which is the best multiclass SVM method? An empirical study. In: Proceedings of Multiple Classifier Systems 6th International Workshop, MCS 2005, Seaside, CA, USA, June 13-15, 278-285.

56. Dutta, K., Prakash, N., Kaushik, S. 2010. Probabilistic neural network approach to the classification of demonstrative pronouns for indirect anaphora in Hindi. Expert Systems with Applications 37(8), 5607-5613.

57. Ephraim, Y. 1992. Gain-adapted hidden Markov models for recognition of clean and noisy speech. IEEE Transactions on Signal Processing 40(6), 1303-1316.

58. Farooq, O., Datta, S. 2001. Mel filter-like admissible wavelet packet structure for speech recognition. IEEE Signal Processing Letters 8(7), 196-198.

59. Farooq, O., Datta, S. 2001. Robust features for speech recognition based on admissible wavelet packets. Electronics Letters 37(25), 1554-1556.

60. Farooq. O., Datta, S. 2003. Phoneme recognition using wavelet based features. Information Sciences 150(1-2), 5-15.

61. Farsi, H., Saleh, R. 2014. Implementation and optimization of a speech recognition system based on hidden Markov Model using genetic algorithm. In: Proceedings of Iranian Conference on Intelligent Systems, ICIS' 14, Bam, Iran, Feb. 4-6, 1-5.

62. Favero, R. F. 1994. Compound wavelets: wavelets for speech recognition. In: Proceedings of IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis, Philadelphia, PA, Oct. 25-28, 600-603.
63. Faycal, Y., Messaoud, B. 2014. Comparative performance study of several features for voiced/unvoiced classification. The International Arab Journal of Information and Technology 11(3), 293-299.
64. Fernåndez-Lorenzana R., Pẽrez-Cruz F., Garcỉa-Cabellos J.M., Palåez-Moreno C., Gallardo-Antolin A., Dỉaz-de-Marỉa F. 2003. Some experiments on speaker-independent isolated digit recognition using SVM classifiers. In: ITRW on Non-Linear Speech Processing, Le Croisic, France, May 20-23.
65. Flynn R., Jones E. 2010. Robust distributed speech recognition in noise and packet loss conditions. Digital Signal Processing 20(6), 1559-1571.
66. Foithong, S., Pinngern, O., Attachoo, B. 2012. Feature subset selection wrapper based on mutual information and rough sets. Expert Systems with Applications 39(1), 574-584.
67. Forsberg, M. 2003. Why is speech recognition difficult? Chalmers University of Technology.
68. Giridhar Rao, B.S. 1989. A fractal characterization of speech waveform graphs, Master of Science Thesis, Texas Tech University.
69. Gaikwad, S.K., Gawali, B.W., Yannawar P. 2010. A review on speech recognition technique. International Journal of Computer Applications 10(3), 16-24.
70. Gales, M.J.F., Flego, F. 2010. Discriminative classifiers with adaptive kernels for noise robust speech recognition. Computer Speech and Language 24(4), 648-662.
71. Ganapathiraju, A., Hamaker, J.E., Picone, J. 2004. Applications of support vector machines to speech recognition. IEEE Transactions on Signal Processing 52(8), 2348-2355.
72. Gangashetty, S.V., Chandra Sekhar, C., Yegnanarayana, B. 2004. Acoustic model combination for recognition of speech in multiple language using support vector machines. In: IEEE International Joint Conference on Neural Networks, Budapest, Hungary, July 25-29, 2004, 4, 3065-3069.
73. Goh, J., Tang, H.L., Peto, T., Saleh, G. 2012. An evolutionary approach for determining hidden Markov model for medical image analysis. In: Proceedings of IEEE World Congress on Computational Intelligence, CEC'12, Brisbane, Australia, June 10-15, 1-8.

74. Gori, M., Tesi, A. 1992. On the problem of local minima in back-propagation. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(1), 76-86.

75. Graves, A., Schmidhuber, J. 2005. Framewise phoneme classifier with bidirectional LSTM and other neural network architectures. Neural Networks 18(5-6), 602-610.

76. Guo, X.C., Yang, J.H., Wu, C.G., Wang, C.Y., Liang, Y.C. 2008. A novel LS-SVMs hyper-parameter selection based on particle swarm optimization. Neurocomputing 71(16-18), 3211-3215.

77. Gupta, R., Sivakumar, G. 2011. Speech recognition for Hindi, M.Tech Project Report, Department of computer science and engineering, M.Tech Project Report, Indian Institute of Technology, Bombay.

78. Han, X., Chan, X., Quan, L., Xiong, X., Li, J., Zhang, Z., Liu, Y. 2014. Feature subset selection by gravitational search algorithm optimization. Information Sciences 281(10), 128-146.

79. Hassan Md., R., Nath, B., Kirley, M., Kamruzzman, J. 2012. A hybrid of multiobjective evolutionary algorithm and HMM-fuzzy model for time series prediction. Neurocomputing, 81(April), 1-11.

80. Haykin, S. 1994. Neural Networks: A comprehensive foundation. Macmillan, NY, USA.

81. He, Q., Yan, J., Shen, Y., Bi, Y., Ye, G., Tian, F., Wang, Z. 2012. Classification of electronic nose data in wound infection detection based on PSO-SVM combined with wavelet transform. Intelligent Automation & Soft Computing 18(7), 967-979.

82. Helmi, N., Helmi, B.H. 2008. Speech recognition with fuzzy neural network for discrete words. In: Fourth International Conference on Natural Computation, Shandong University, China, Oct. 18-20, 7, 265-269.

83. Hennebert, J., Hasler, M., Dedieu, H. 1994. Neural networks in speech recognition. In: Proceedings of the 6th Microcomputer School of Neural Networks, Theory and Applications , Prague, Czech Republic, Sept. 18-23, 23-40.

84. Hoesen, Devin, Satriawan, Cil Hardianto, Lestari, Dessi Puji, Khodra, Masayu Leylia, 2016. Towards robust Indonesian speech recognition with spontaneous-speech adapted acoustic models. Procedia Computer Science, 81, 167-173.

85. Hsieh, C.-T., Hu, C.-S. 2014. Fingerprint recognition by multi-objective optimization PSO hybrid with SVM. Journal of Applied Research and Technology 12(6), 1014-1024.

86. Hsu, H.-H., Hsieh, C.-W., Lu, M.-D. 2011. Hybrid feature selection by combining filters and wrappers. Expert Systems with Applications 38(7), 8144-8150.

87. Hsu, C.-W., Lin, C.-J. 2002. A comparison of methods for multiclass support vector machines. IEEE Transactions on Neural Networks 13(2), 415-425.

88. Hu, Z., Bao, Y., Chiong, R., Xiong, T. 2015. Mid-term interval load forecastingusing multi-output support vector regression with a memetic algorithm forfeature selection. Energy 84 (May), 419-431.

89. Hu, Z., Bao, Y., Xiong, T. 2013. Electricity load forecasting usingsupport vector regression with memetic algorithms. The Scientific WorldJournal 2013(292575).

90. Hua-chao, Y., Shu-bi, Z., Ka-zhong, D., Pei-Jun, D. 2007. Research into a feature selection method for hyperspectral imagery using PSO and SVM. Journal of China University Mining & Technology 17(4), 473-478.

91. Huang, C.-L. 2009. ACO-based hybrid classification system with feature subset selection and model parameters optimization. Neurocomputing 73(1-3), 438-448.

92. Huang, C.-L., Wang, C.-J. 2006. A GA-based feature selection and parameters optimization for support vector machines. Expert Systems with Applications 31(2), 231-240.

93. Huang, C.-L., Dun, J.-F. 2008. A distributed PSO–SVM hybrid system with feature selection and parameter optimization. Applied Soft Computing 8(4), 1381-1391.

94. Hwang, D., Kim, D. 2012. Near-boundary data selection for fast support vector machines. Malaysian Journal of Computer Science 25(1), 23-37.

95. Illina, I., Gong, Y. 1996. Improvement in N-best search for continuous speech recognition. In: Proceedings of Fourth International Conference on Spoken Language, Philadelphia, PA, Oct. 3-6, 4, 2147-2150.

96. Ilhan, I., Gulay, T. 2013. A genetic algorithm–support vector machine method with parameter optimization for selecting the tag SNPs. Journal of Biomedical Informatics 46(2), 328-340.

97. Inza, I., Merino, M., Larrñaga, P., Quiroga, J., Sierra, B., Girala, M. 2001. Feature subset selection by genetic algorithms and estimation of distribution algorithms: A case study in the survival of cirrhotic patient treated with TIPS. Artificial Intelligence in Medicine 23(2), 187-205.

98. Jànos, D. P. 2012. Calibrating artificial neural networks by global optimization. Expert Systems with Applications 39(1), 25-32.

99. Jiang, H. 2010. Discriminative training of HMMs for automatic speech recognition: A survey. Computer Speech and Language 24(4), 589-608.

100. Jin, W., Zhang, J.-Q., Zhang, X. 2011. Face recognition method based on support vector machine and particle swarm optimization. Expert Systems with Applications 38(4), 4390-4393.

101. Juang, B.H., Rabiner, L.R. 1991. Hidden Markov models for speech recognition. Technometrics 33(3), 251-272.

102. Juang, B.H., Rabiner, L.R. 2005. Automatic speech recognition-A brief history of the technology. Elsevier Encyclopedia of Language and Linguistics, second edition.

103. Kabir Md., M., Islam Md., M., Murase, K. 2010. A new wrapper selection approach using neural network. Neurocomputing 73(16-18), 3273-3283.

104. Karpagavalli, S., Chandra, E., 2015. Phoneme and word based model for Tamil speech recognition using GMM-HMM.Int. Conf. on Advanced Computing and Communication Systems, Coimbatore, India.

105. Kashef, S., Nezamabadi-pour, H. 2015. An advanced ACO algorithm for feature subset selection. Neurocomputing 147(January), 271-279.

106. Kennedy, J., Eberhart, R.C. 1995. Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Network, Perth, WA, Australia, Nov. 27-Dec.1,4:1942-1948.

107. Kennedy, J., Eberhart, R.C. 1997. A discrete binary version of the particle swarm algorithm. In: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, Washington, DC, USA, Oct. 12-15, 4104-4108.

108. Khan, M.M., Ahmad, A.M., Khan, G.M., Miller, J.F. 2013. Fast learning networks using Cartesian genetic programming. Neurocomputing 121(December), 274-289.

109. Khoo, L., Cvetkovic, Z., Sollich, P. 2005. Robustness of phoneme recognition using support vector machines. In: Proceedings of International Conference on Acoustics, Speech, and Signal Processing, Toulouse, France, May 14-19.

110. Kim, N.S., Un, C.K. 1995. On estimating robust probability distribution in HMM-based speech recognition. IEEE Transactions on speech and audio processing 3(4), 279-285.

111. Kim, C., Stern, Richard M., 2016. Power-normalized cepstral coefficients for robust speech recognition. IEEE/ACM Transactions on Audio, Speech and Language Processing 24(7), 1315-1329.

112. Kim, Jae-Bok, Park, Jeong-Sik, 2016. Multistage data selection-based unsupervised speaker adaption for personalized speech emotion recognition. Engineering Applications of Artificial Intelligence 52, 126-134.

113. Kishore, S.P., Black Alan, W., Kumar R., Sangal R. 2003. Experiments with unit selection speech database for Indian languages. In: Presented at National Seminar on Language Technology Tools: Implementation of Telugu, Oct., Hyderabad, India.

114. Kohavi, R., John, G.H. 1997. Wrappers for feature selection, Artificial Intelligence 97(1-2), 273-324.

115. Kopparapu, S.K., Bhuvanagiri, K.K. 2013. Recognition of subsampled speech using a modified Mel filter bank. Computers and Electrical Engineering 39(2), 655-662.

116. Kopparapu, S.K., Laxminarayana, M. 2010. Choice of Mel filter bank in computing MFCC of a resampled speech. In: Proceedings of 10th International Conference on Information Sciences Signal Processing and their Applications, ISSPA'2010, Kuala Lumpur, Malaysia, May 10-13, 121-124.

117. Krishnan, M., Neophytou, C.P., Prescott, G. 1994. Wavelet transform speech recognition using vector quantization, dynamic time warping and artificial neural networks. Computer Aided Systems Engineering and Telecommunications & Information Science Laboratory.

118. Kumar E.V., Raaja, G.S., Jerome, J. 2016. Adaptive PSO for optimal LQR tracking control of DoF laboratory helicopter. Applied Soft Computing 41, 77-90.

119. Kwok, J. 1999. Moderating the outputs of support vector machine classifiers. IEEE Transactions on Neural Networks, 10, 1018-1031.

120. Kwong, S., Chau, C.W. 1997. Analysis of parallel genetic algorithms on HMM based speech recognition system. IEEE Transactions on Consumer Electronics 43(4), 1229-1233.

121. Kwong, S., Chau, C.W., Man, K.F., Tang, K.S. 2001. Optimization of HMM topology and its model parameters by genetic algorithms. Pattern Recognition 34(2), 509-522.

122. Latorre, J., Iwano, K., Furui, S. 2006. New approach to the polyglot speech generation by means of an HMM-based speaker adaptable synthesizer. Speech Communication 48(10), 1227-1242.

123. Leandro, M.A., Teresa, B.L. 2010. A multi-objective memetic and hybrid methodology for optimizing the parameters and performance of artificial neural networks. Neurocomputing 73(7-9), 1438-1450.

124. Lee, K.-F. Large-vocabulary speaker-independent continuous speech recognition: The Sphinx system, Ph. D. Thesis, Carnegie Mellon University, 1988.

125. Li, X., Yang, S.-D., Qi, J.-X. 2006. A new support vector machine optimized by improved particle swarm optimization and its application. Journal of Central South University of Technology 13(5), 567-571.

126. Li, Y.-B., Zhang, N., Li, C.-B. 2009. Support vector machine forecasting method improved by chaotic particle swarm optimization and its application. Journal of Central South University of Technology 16(3), 478-481.

127. Liddy, E.D. 2001. Natural language processing. In: Encyclopedia of Library and Information Sciences, 2nded. New York, Marcel Decker. Inc.

128. Lin, S.-W., Ying, K.-C., Chen, S.-C., Lee, Z.-J. 2008. Particle swarm optimization for parameter determination and feature selection of support vector machines. Expert Systems with Applications 35(4), 1817-1824.

129. Lippmann, R.P. 1989. Review of neural networks for speech recognition. Neural Computation 1(1), 1-38.

130. Liu, B., Hao, Z.-F., Yang, X.-W. 2005. Nesting support vector machine for multi-classification. In: Proceedings of the IEEE Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, China, Aug. 18-21, 4220-4225.

131. Liu, J., Wang, Z., Xiao, X. 2007. A hybrid SVM/DDBHMM decision fusion modeling for robust continuous digital speech recognition. Pattern Recognition Letters 28(8), 912-920.

132. Liu, Y., Wang, G., Chen, H., Dong, H., Zhu, X., Wang, S. 2011. An improved particle swarm optimization for feature selection. Journal of Bionic Engineering 8 (2): 191-200.

133. Long, C.J., Datta, S. 1996. Wavelet based feature extraction for phoneme recognition. In: Proceedings of Fourth International Conference on Spoken Language Processing, Philadelphia, PA, Oct 3-6, 1, 264-267.

134. Long, C.J., Datta, S. 1998. Discriminant wavelet basis construction for speech recognition. In: Proceedings of Fifth International conference on Spoken Language Processing, Sydney, Australia, Nov. 30-Dec. 4, 3,1047-1049.

135. Maldonado, S., Weber, R. 2009. A wrapper method for feature selection using support vector machines. Information Sciences 179(13), 2208-2217.

136. Manikandan, J., Venkataramani, B. 2011. Evaluation of multiclass support vector machine classifiers using optimum threshold-based pruning technique. IET Signal Processing 5(5), 506-513.

137. MATLAB Toolboxversion 7.8.0**,**2009.Natick, Massachusetts: The MathWorks Inc.

138. McCullough, W.S., Pitts, W.H. 1943. A logical calculus of idea immanent in nervous activity. The Bulletin of Mathematical Biophysics 5, 115-133.

139. McDermott, E., Hazen, T.J., Roux, J.L., Nakamura, A., Katagiri, S. 2007. Discriminative training for large-vocabulary speech recognition using minimum classification error. IEEE Transactions on Audio, Speech and Language Processing 15(1), 203-223.

140. Mehrotra, K., Mohan, C.K., Ranka, S. 1997. Artificial Neural Networks, The MIT Press.

141. Milner, B., Darch, J. 2011. Robust acoustic speech feature prediction from noisy Mel-Frequency cepstral coefficients. IEEE Transactions on Audio, Speech and Language Processing 19(2), 338-347.

142. Milner, B., Darch, J., Almajai, I. 2009. Reconstructing clean speech from noisy MFCC vectors. In: Proceedings of 10th Annual Conference of the International Speech Communication Association, ISCA' 2009, Brighton, United Kingdom, Sep. 6-10, 1943-1946.

143. Mnikandan, J., Venkataramani, B. 2011. Design of a real time automatic speech recognition system using modified one against all SVM classifier. Microprocessors and Microsystems 35(6), 568-578.

144. Mohamed, A., Ramachandran Nair, K.N. 2012. HMM/ANN hybrid model for continuous Malayalam speech recognition. Procedia, (Engineering International Conference on Communication Technology and System Design, 2011) 30, 616-622.

145. Morgan, N., Bourlard, H.A. 1995. Neural networks for statistical recognition of continuous speech. In: Proceedings of the IEEE 83(5), 742-770.

146. Mporas, I., Ganchev, T., Siafarikas, M., Fakotakis, N. 2007. Comparison of speech features on the speech recognition task. Journal of Computer Science 3(8), 608-616.

147. Muda, L., Begam, M., Elamvazuthi, I. 2010. Voice recognition algorithms using Mel-Frequency cepstral coefficient and dynamic time warping techniques. Journal of Computing 2(3), 138-143.

148. Muller, D.N., de Siqueira, M.L., Navaux P.O.A. 2006. A connectionist approach to speech understanding. In: International Joint Conference on Neural Networks, IJCNN'06, Vancouver, BC, Jul.16-21, 3790-3797.

149. Murveit, H., Cohen, M., Price, P., Baldwin, G., Weintraub, M., Bernstein, J. SRI's DECIPHER System. In: Proceedings of the speech and Natural Language Workshop, Philadelphia, PA, 1989.

150. Najkar, N., Razzazi, F., Sameti, H. 2010. A novel approach to HMM-based speech recognition systems using particle swarm optimization. Mathematical and Computer Modelling 52(11-12), 1910-1920.

151. Narang, N., Dhillon, J.S., Kothari D.P. 2012.Multi-objective short-term hydrothermal generation scheduling using predator-prey optimization. Electric Power Components and Systems 40(15) 1708-1730.

152. Narang, N., Dhillon, J.S., Kothari D.P. 2014. Multiobjective Fixed head hydrothermal scheduling using predator-prey optimization and Powell search method. Energy 47(1), 237-252.

153. Neti, C., Rajput, N., Verma, A. 2004. A large vocabulary continuous speech recognition for Hindi. IBM Journal of Research and Development 48(5-6), 703-716.

154. Nouza, J., Zdansky, J., Cerva, P. 2010. System for automatic collection, annotation and indexing of Czech broadcast speech with full-text search. In: Proceedings of 15th IEEE Mediterranean Electrotechnical Conference, MELECON 2010, Valletta, Apr. 26-28, 202-205.

155. Nguyen, Duc, Hoang, Ha, Xiong, Xiao, Siong, Chng, Haizhou Li, 2016. Feature adaption using linear spectro-temporal transform for robust speech recognition, IEEE/ACM Transactions on Audio, Speech and Language Processing. 24(6), 1006-1019.

156. Ostendorf, M., Digalakis, V.V., Kimball, O.A. 1996. From HMM's to segment models: A unified view of stochastic modeling for speech recognition. IEEE Transactions on Speech and Audio Processing 4(5), 360-378.

157. ỌShaughnessy, D. 2003. Interacting with computers by voice: automatic speech recognition and synthesis. In: Proceedings of the IEEE 91(9), 1272-1305.

158. ỌShaughnessy, D. 2008. Automatic speech recognition: History, methods and challenges. Pattern Recognition 41(10), 2965-2979.

159. Ozyildirim, B.M., Avci, M. 2013. Generalized classifier neural network. Neural Networks 39(March), 18-26.

160. Panda, S., Tomar, S.K., Prasad, R., Ardil, C. 2009. Reduction of linear time-invariant systems using Routh-approximation and PSO. International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering 3(9), 1775-1782.

161. Pauplin, O., Jiang, J. 2012. DNB-based structure learning and optimisation for automated handwritten character recognition. Pattern Recognition Letters 33(6), 685-692.

162. Pavez, E., Silva, J.F. 2012. Analysis and design of wavelet-packet cepstral coefficients for automatic speech recognition. Speech Communication 54(6), 814-835.

163. Pisarn, C., Theeramunkong, T. 2007. An HMM-based method for Thai spelling speech recognition. An International Journal Computers & Mathematics with Applications 54(1), 76-95.

164. Platt J. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in Large Margin Classifiers. Cambridge; MA: MIT Press.

165. Pour, M.M., Farokhi, F. 2009. A new approach for Persian speech recognition. In: Proceedings of the IEEE International Advance Computing Conference, Patiala, India, 6-7 March, 153-158.

166. Prager, R.W., Harrison, T.D., Fallside, F. 1986. Boltzmann machines for speech recognition. Computer Speech and Language 1(1), 3-27.

167. Prasad, D.P., Gerald, E.M. 2005. Experiments with Fast Fourier Transform, Linear Predictive and Cepstral coefficients in Dysarthric speech recognition algorithms using hidden Markov model. IEEE Transactions on Neural Systems and Rehabilitation Engineering 13(4), 558-561.

168. Prina Ricotti, L. 2005. Multitapering and a wavelet variant of MFCC in speech recognition. IEE Proceedings Vision, Image and Signal Processing 152(1), 29-35.

169. Pujol, P., Pol, S., Nadeu, C., Hagen, A., Bourlard, H. 2005. Comparison and combination of features in a hybrid HMM/MLP and a HMM/GMM speech recognition system. IEEE Transactions on Speech and Audio Processing 13(1), 14-22.

170. Qiao, J.F., Han, H.G. 2010. A repair algorithm for radial basis function neural network and its application to chemical oxygen demand modelling. International Journal of Neural Systems 20(1), 63-74.

171. Rabiner, L.R., Schafer, R.W. 1978. Digital Processing of Speech Signals. Prentice Hall, Upper Saddle River, NJ.

172. Rabiner, L.R., Juang, B.H. 1986. An introduction to hidden Markov models. IEEE Acoustic Speech and Signal Processing Magazine, 4-16.

173. Rabiner, L.R., Wilpon J.G., Juang, B.H. 1986. A segmental k-means training procedure for connected word recognition,AT& T Techn. J. 65, 21-31.

174. Rabiner, L.R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 77(2), 257-286.

175. Rabiner, L.R., Juang, B.H. 1993. Fundamentals of speech recognition, Prentice-Hall, Englewood Cliff, NJ.

176. Ramĭrez, J., ´Yelamos, P., Gòrriz, J.M., Segura, J.C. 2006. SVM-based speech endpoint detection using contextual speech features. Electronics Letters 42(7), 426-428.

177. Ramĭrez, J., Yĕlamos, P., Gòrriz, J.M., Segura, J.C., Garcĭa, L. 2006. Speech/Non-Speech discrimination combining advanced feature extraction and SVM learning. In: Proceedings of the Ninth International Conference on Spoken Language Processing, ICSLP, Pittsburgh, Pennsylvania, USA, Sep. 17-21, 4, 1662-1665.

178. Ranjan, S. 2010. A discrete wavelet transform based approach to Hindi speech recognition. In: Proceedings of International Conference on Signal Acquisition and Processing, ICSAP'10, Bangalore, India, Feb. 9-10, 345-348.

179. Rao, S.S. 1996. Engineering Optimization: theory and practice. 3rded. New York: John Wiely & Sons.

180. Rasmussen, T.K., Krink, T. 2003. Improved hidden Markov model training for multiple sequence alignment by a particle swarm optimization-evolutionary algorithm hybrid. BioSystems 72 (1-2), 5-17.

181. Ray, T., Liew, K.M. 2003. Society and Civilization: an optimization algorithm based on the simulation of social behaviour. IEEE Transactions on Evolutionary Computation 7(4), 386-396.

182. Renals, S., Morgan, N., Bourlard, H., Cohen, M., Franco, H. 1994. Connectionist probability estimators in HMM speech recognition. IEEE Transactions on Speech and Audio processing 2(1-II), 161-174.

183. Rigoll, G. 1994. Maximum mutual information Neural networks for hybrid connection-HMM speech recognition systems. IEEE Transactions on Speech and Audio Processing 2(1), 175-184.

184. Rivero, D., Dorado, J., Rabuñal, J., Pazos, A. 2010. Generation and simplification of artificial neural networks by means of genetic programming. Neurocomputing 73(16-18), 3200-3223.

185. Robinson, A.J., Fallside, F. 1988. Static and dynamic error propagation networks with application to speech coding. In: D. Anderson, (Ed.), Neural Information Processing Systems, Denver CO, American Institute of Physics, New York, 632-641.

186. Rodrigues, D., Pereira Luıs, A.M., Nakamura, R.Y.M., Costo, K.A.P., Yang, X.-S., Souza, A.N., Papa, J.P. 2014. A wrapper approach for feature selection based on bat algorithm and optimum-path forest. Expert Systems with Applications 41(5), 2250-2258.

187. Romero, E., Alquèzar, R. 2007. Heuristics for the selection of weights in sequential feed-forward neural networks: An experimental study. Neurocomputing 70(16-18), 2735-2743.

188. Rosenfeld, R. 2000. Two decades of statistical language modeling: where do we go from here? In: Proceedings of the IEEE 88(8), 1270-1278.

189. Salomon, R. 1998. Evolutionary algorithms and gradient search: similarities and differences. IEEE Transactions on Evolutionary Computation 2(2), 45-55.

190. Sabah, R., Aino, R.N. 2009. Isolated digit speech recognition in Malay language using neuro-fuzzy approach. In: Proceedings of Third Asia International Conference on Modelling & Simulation, Bandung/Bali, Indonesia, May. 25-29, 336-340.

191. Saha, G., Chakraborty, S., Senapati, S. 2005. A new silence removal and endpoint detection algorithm for speech and speaker recognition applications. In: Proceedings of the Eleventh National Conference on Communications, Kharagpur, India, January, 291-295.

192. Saha, S.K., Mitra, P., Sarkar, S. 2012. A comparative study on feature reduction approaches in Hindi and Bengali named entity recognition. Knowledge-Based Systems 27(March), 322-332.

193. Saharia, N., Das, D., Sharma, U., Kalita, J. 2009. Part of Speech Tagger for Assamese Text. In: Proceeding of the Association for Computational Linguistic and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL-IJCNLP'2009, Singapore, 4 Aug 33-36.

194. Saharia, N., Sharma, U., Kalita, J. 2010. A suffix-based Noun and Verb classifier for an inflectional language. In: International Language on Asian Language Processing, IALP'10, Harbin Heilongjjiang China, Dec. 28-30, 19-22.

195. Saharia, N., Sharma, U., Kalita, J. 2014. Stemming resource-poor Indian languages. ACM Transactions on Asian Information Processing 13(3), 14:1-14:26.

196. Samudravijaya, K., Rao, P.V.S., Agrawal, S.S. 2000. Hindi speech database. In: Proceedings of International Conference on Spoken Language Processing ICSLP00, Beijing, China, October 2000.CDROM paper:00192.

197. Sanchez-Cortina, I., Jesús, Andrés-Ferrer, Sanchis, A., Juan A., 2016. Speaker-adapted confidence measure for speech recognition of video lectures. Computer Speech and Language 37,11-23.

198. Santana, L.E.A.S., Canuto, A.M.P. 2014. Filter-based optimization techniques for selection of feature subsets in ensemble systems. Expert Systems with Applications 41(4), 1622-1631.

199. Sarma, H., Saharia, N., Sharma, U., Sinha, S.K., Malakar, M.J. 2011. Development and Transcription of Assamese Speech Corpus. In: National seminar cum Conference on Recent threads and Techniques in Computer Sciences.

200. Schwartz, R., Barry, C., Chow, Y.-L., Derr, A., Feng, M.-W., Kimball, O., Kubala, F., Makhoul, J., Vandegrift, J. 1989. The BBN BYBLOS continuous speech recognition system. In: Proceddings of the Speech and Natural Language Workshop, Philadelphia, PA, 94-99.

201. Selvakumar, A.I., Thanushkodi, K. 2009. Optimization using civilized swarm: Solution to economic dispatch with multiple minima. Electric Power Systems Research 79(1), 8-16.

202. Seman, N., Jamil, N. 2015. Blending sentence optimization weights of unsupervised approaches for extractive speech summarization. Procedia Computer Science 51, 620-629.

203. Shin, J.W., Chang, J.H., Kim, N.S. 2010. Voice activity detection based on statistical models and machine learning approaches. Computer Speech and Language 24(3), 515-530.

204. Sarafrazi, S., Nezamabadi-pour, H. 2013. Facing the classification of binary problems with GSA-SVM hybrid system. Mathematical and Computer Modelling 57 (1-2), 270-278.

205. Schmidhuber, J. 2015. Deep learning in neural networks: An overview. Neural Networks 61(January), 85-117.

206. Sharma, U., Kalita, J.K., Das, R.K. 2008. Acquisition of morphology of an Indic language from text corpus. ACM Transactions on Asian Language Information Processing 7(3), 9:1-9:33.

207. Silva, A, Ana, N., Ernesto, C. 2002. An empirical comparison of particle swarm and predator prey optimization. In: Proceedings of the 13th Irish International Conference on Artificial Intelligence and Cognitive Science, Limerick, Ireland, Sep. 12-13, 24(64), 103-110.

208. Siniscalchi, S.M., Yu, D., Deng, L., Lee, C.-H. 2013. Exploiting deep neural networks for detection-based speech recognition. Neurocomputing 106(April), 148-157.

209. Sivagaminathan, R.K., Ramakrishnan, S.A. 2007. Hybrid approach for feature subset selection using neural networks and ant colony optimization. Expert Systems with Applications 33(1), 49-60.

210. Sivaram, G.S.V.S., Hermansky, H. 2011. Multilayer perceptron with sparse hidden outputs for phoneme recognition. In: Proceedings of IEEE International Conference on Acoustic Speech and Signal Processing, Prague, May 22-27, 5336-5339.

211. Sivaram, G.S.V.S., Hermansky, H. 2012. Sparse multilayer perceptron for phoneme recognition. IEEE Transactions on Audio, Speech and Language Processing 20(1), 23-29.

212. Sloin, A., Burshtein, D. 2008. Support vector machine training for improved hidden markov modeling. IEEE Transactions on Signal Processing 56(1), 172-188.

213. Solera-Ureña, R., Martın̂-Iglesias, D., Gallardo-Antolın̂, A., Pelàez-Moreno, C., Dıaz-de-Marıa, F. 2007. Robust ASR using support vector machines. Speech Communication 49(4), 253-267.

214. Song, Y., Chen, Z., Yuan, Z. 2007. New chaotic PSO-based neural network predictive control for nonlinear process. IEEE Transactions on Neural Networks 18(2), 595-601.

215. Sonkamble, B.A., Doye, D.D. 2008. An overview of speech recognition system based on the support vector machines. In: Proceedings of the International Conference on Computer and Communication Engineering, Kuala Lumpur, Malaysia, May 13-15, 768-771.

216. Soruri, M., Hamid Zahiri, S., Sadri, J. 2013. A new approach of training hidden Markov model by PSO algorithm for gene sequence modeling. In: Proceedings of

First Iranian Conference on Pattern Recognition and Image Analysis, PRIA'13, Birjand, Iran, Mar. 6-8, 1-4.

217. Stephan, K.C., Alan, D.B. 2003. Incremental training of first order recurrent neural networks to predict a context-sensitive language. Neural Networks 16 (7) 955-972.
218. Stone, M. 1974. Cross validation choice and assessment of statistical predictions. Journal of the Royal Statistical Society B 36, 111-147.
219. Sun, J., Wu, X., Fang, W., Ding, Y., Long, H., Xu, W. 2012. Multiple sequence alignment using the Hidden Markov Model trained by an improved quantum-behaved particle swarm optimization. Information Sciences 182(1), 93-114.
220. Tan, Z.H., Lindberg, B. 2010. Low-complexity variable frame rate analysis for speech recognition and voice activity detection. IEEE Journal of Selected Topics in Signal Processing 4(5), 798-807.
221. Tebelskis, J., Waibel, A. 1990. Large vocabulary recognition using linked predictive neural networks. In: Proceedings of IEEE International Conference Acoustic, Speech, and Signal Processing, ICASSP-90, April 3-6, Albuquerque, NM, USA, 1, 437-440.
222. Thatphithakkul, N., Kanokphara S. 2004. HMM Parameter optimization using tabu search. In: Proceedings of International Symposium on Communications and Information Technologies, ISCIT 2004, Sapporo, Japan, Oct. 26-29, 904-908.
223. TI 46-Word speaker dependent isolated word corpus, 1991. NSIT speech disc (1991):7-1.1.
224. Ting, H.-N., Yong, B.-F., Mirhassani, S.M. 2013. Self-adjustable neural network for speech recognition. Engineering Applications of Artificial Intelligence 26(9), 2022-2027.
225. Tipping, M.E. 2000. The relevance vector machine. Advances in Neural Information Processing Systems 12. Cambridge, MA: MIT Press, 652-665.
226. Todor, D.G., Dimitris, K.T., Michael, N.V., Nikos, D.F. 2007. Generalized locally recurrent probabilistic neural networks with application to text-independent speaker verification. Neurocomputing 70(7-9), 1424-1438.
227. Tohidypour, H.R., Seyyedsalehi, S.A., Behbood, H., Roshandel, H. 2012. A new representation for speech frame recognition based on redundant wavelet filter banks. Speech Communication 54(2), 256-271.
228. Trentin, E., Gori, M. 2001. A survey of hybrid ANN/HMM models for automatic speech recognition. Neurocomputing 37(1-4), 91-126.

229. Trentin, E., Gori, M. 2003. Robust combination of neural networks and Hidden Markov models for speech recognition. IEEE Transactions on neural networks 14(6), 1519-1531.

230. Trentin, E., Gori, M. 2006. Inversion-based nonlinear adaptation of noisy acoustic parameters for a neural/HMM speech recognizer. Neurocomputing 70(1-3), 398-408.

231. Truong, T.K., Lin, C.-C., Chen, S.-H. 2007. Segmentation of specific speech signals from multi-dialog environment using SVM and wavelet. Pattern Recognition Letters 28(11), 1307-1313.

232. Tufekci, Z., Gowdy, J.N. 2000. Feature extraction using discrete wavelet transform for speech recognition. In: Proceedings of the IEEE Southeastcon 2000, Nashville, TN, 9-9 April, 116-123.

233. Tufekci, Z., Gowdy, J.N., Gurbuz, S., Patterson, E. 2006. Applied mel-frequency discrete wavelet coefficients and parallel model compensation for noise-robust speech recognition. Speech Communication 48(10), 1294-1307.

234. Udhyakumar, N., Swaminathan, R., Ramakrishnan, S. K. 2004. Multilingual speech recognition for information retrieval in Indian context. In: Proceedings of the Student Research Workshop at Human Language Technology-North American Chapter of the Association for Computational Linguistics, Boston, Massachusetts, USA, May 6-7, 1-6.

235. Umarani, S.D., Raviram, P., Wahidabanu, R.S.D. 2009. Implementation of HMM and radial basis function for speech recognition. In: International Conference on Intelligent Agent and Multi-Agent Systems, July 22-24, Chennai, India 1-4.

236. Uncu, Ö., Türkşen, I.B. 2007. A novel feature selection approach: combining feature wrappers and filters. Information Sciences 177(2), 449-466.

237. Urena, R.S., Moral, A.I.G., Moreno, C.P., Ramon, M.M., Maria, F.D. 2012. Real-time robust automatic speech recognition using compact support vector machines. IEEE Transactions on Audio, Speech and Language Processing 20(4), 1347-1361

238. Valentini-Botinhao, C., Yamagishi, J., King, S., Maia, R. 2014. Intelligibility enhancement of HMM-generated speech in additive noise by modifying Mel cepstral coefficients to increase the glimpse proportion. Computer Speech and Language 28(2), 665-686.

239. Vapnik, V., 1995. The Nature of Statistical Learning Theory, Springer-Verlag, New York.

240. Vieira, S.M., Mendonca, L.F., Farinhaa, G.J., Sousa, J.M.C. 2013. Modified binary PSO for feature selection using SVM applied to mortality prediction of septic patients. Applied Soft Computing 13(8), 3494-3504.

241. Vignolo, L.D., Milone, D.H., Scharcanski, J. 2013. Feature selection for face recognition based on multi-objective evolutionary wrappers. Expert Systems with Applications 40(13), 5077-5084.

242. Vimala, C., Radha, V. 2012. A review on speech recognition challenges and approaches. World of Computer Science and Information Technology Journal 2(1), 1-7.

243. Vimal Krishnan, V.R., Babu Anto, P. 2009. Features of Wavelet packet decomposition discrete Wavelet transform for Malayalam speech recognition. International Journal of Recent Trends in Engineering 1(2), 93-96.

244. Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., Lang, K. 1989. Phoneme recognition using time-delay neural networks. IEEE Transactions on Acoustics Speech and Signal Processing 37(1), 328-339.

245. Walker, S.L., Foo, S.Y. 2003. Optimal wavelets for speech signal representations. Journal of Systemics, Cybernetics and Informatics 1(4), 44-46.

246. Wang, L., Minami, K., Yamamoto, K., Nakagawa, S. 2010. Speaker identification by combining MFCC and phase information in noisy environments. In: Proceedings of International Conference on Acoustic Speech and Signal Processing ICASSP'2010, Dallas, TX, Mar.14-19, 4502-4505.

247. Wang, H., Zhang, G., Mingjie, E., Sun, N. 2011. A novel intrusion detection method based on improved SVM by combining PCA and PSO. Wuhan University Journal of Natural Sciences 16(5), 409-413.

248. Wang, L., Zou, F., Yang, D., Chen, D., Jiang, Q. 2014. An improved teaching-learning-based optimization with neighborhood search for application of ANN. Neurocomputing 143(November), 231-247.

249. Watrous, R. L., Shastri, L. 1987. Learning phonetic features using connectionist networks: An experiment in speech recognition. In: Proceedings of First IEEE International Conference on Neural Networks, San Diego, California, June 21-24, 2, 619-627.

250. Weston, J., Watkins, C. 1999. Support vector machines for multi-class pattern recognition. In: Proceedings of Seventh European Symposium Artificial Neural Networks, Bruges, Belgium, Apr. 21-23, 219-224.

251. Wilpon, J.G., Rabiner, L.R., Lee, C.-H., Goldman, E.R. 1990. Automatic recognition of keywords in unconstrained speech using hidden Markov models. IEEE Transactions on Acoustics, Speech, and Signal processing 38(11), 1870-1878.

252. Wu, J., Chan, C. 1993. Isolated word recognition by neural network models with cross-correlation coefficients for speech dynamics. IEEE Transactions on Pattern Analysis and Machine Intelligence 15(11), 1174-1185.

253. Wu, J.-D., Lin, B.-F. 2009. Speaker identification using discrete wavelet packet transform technique with irregular decomposition. Expert Systems with Applications 36(2), 3136-3143.

254. Wu, Y.-J., Kawai, H., Ni, J., Wang, R.-H. 2005. Discriminative training and explicit duration modeling for HMM-based automatic segmentation. Speech Communication 47(4), 397-410.

255. Xia, Y., Wang, J. 2004. A one-layer recurrent neural network for support vector machine leaning. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics 34(2), 1261-1269.

256. XinXing, J., Xu, S. 2012. Speech recognition based on effective DTW algorithm and its DSP implementation. Procedia Engineering (2012 International Workshop on Information and Electronics Engineering) 29, 832-836.

257. Xiong, T., Bao, Y., Hu, Z., Chiong, R. 2015. Forecasting interval timeseries using fully complex-valued RBF neural network with DPSO and PSOalgorithm. Information Science 305 (June), 77-92.

258. Xueying, Z., Zhiping, J. 2004. Speech recognition based on auditory wavelet packet filter. In: Proceedings of Seventh International Conference on Signal Processing, Beijing, Aug. 31-Sep. 4, 1, 695-698.

259. Yang, F., Zhang, C. 2008. An effective hybrid optimization algorithm for HMM. In: Proceedings of IEEE Fourth International Conference on Neural Computation, ICNC'08, Jinan, Oct. 18-20, 4, 80-84.

260. Yao, K., Paliwal, K.K., Lee, T.-W. 2005. Generative factor analyzed HMM for automatic speech recognition. Speech Communication 45(4), 435-454.

261. Yılmaz, E., Heuvel, Henk van den, Leeuwen, David van, 2016. Investigating bilingual deep neural networks for automatic recognition of code-switching Frisian speech. Procedia Computer Science 81, 159-166.

262. Yu, E., Cho, S. 2006. Ensemble based on GA wrapper feature selection. Computers & Industrial Engineering 51(1), 111-116.

263. Yu, H.-J., Oh, Y.-H. 2000. A neural network for 500 word vocabulary word spotting using non-uniform units. Neural Networks 13 (6), 681-688.

264. Yuan, H., Zhi, J., Liu, J. 2011. Application of particle swarm optimization algorithm-based fuzzy BP neural network for target damage assessment. Scientific Research and Essays 6(15), 3109-3121.

265. Zeng, J., Liu, Z.-Q. 2006. Type-2 fuzzy hidden Markov models and their application to speech recognition. IEEE Transactions on Fuzzy Systems 14(3), 454-467.

266. Zhang, J.R., Zhang, J., Lokt Liu, M.R. 2007. A hybrid particle swarm optimization-back-propagation algorithm for feedforward neural network train. Applied Mathematics and Computation 185(2), 1026-1037.

267. Zhang, R., Xu, Z.B., Huang, G.B., Wang, D.H. 2012. Global convergence of online BP training with dynamic learning rate. IEEE Transactions on Neural Networks and Learning Systems 23(2), 330-341.

268. Zhang, X., Wang, Y., Zhao, Z. 2007. A hybrid speech recognition training method for HMM based on genetic algorithm and Baum Welch algorithm. In: Proceedings of second International conference on Innovative Computing, Information and Control, ICICIC'07, Kumamoto, Japan, Sept. 5-7, 572-577.

269. Zhang, X., Liu, X., Wang, Z. J. 2013. Evaluation of a set of new ORF kernel functions of SVM for speech recognition. Engineering Applications of Artificial Intelligence 26(10), 2574-2580.

270. Zhao, M., Fu, C., Ji, L., Tang, K., Zhou, M. 2011. Feature selection and parameter optimization for support vector machines: A new approach based on genetic algorithm with feature chromosomes. Expert Systems with Applications 38(5), 5197-5204.

271. Zhao, Z.-S., Feng, X., Lin, Y.-Y., Wei, F., Wang, S.-K., Xiao, T.-L., Cao, M.-Y., Hou, Z.-G. 2015. Evolved neural network ensemble by multiple heterogeneous swarm intelligence. Neurocomputing 149(February), 29-38.

272. Zhou, P., Tang, L.Z., Xu, D.F. 2009. Speech recognition algorithm of parallel subband HMM based on wavelet analysis and neural network. Information Technology Journal 8(5), 796-800.

LIST OF PUBLICATIONS

- Teena Mittal and R. K. Sharma, "Multiclass SVM based spoken Hindi numerals recognition", *The International Arab Journal of Information Technology*, 12(6A), 2015, 666-71.

- Teena Mittal and R.K. Sharma, "Integrated search technique for parameter determination of SVM for speech recognition", *Journal of Central South University*, 23(6), 2016, 1390-98.

- Teena Mittal and R.K. Sharma, "Feature subset selection and parameter determination of SVM using a hybrid optimization technique for speech recognition", *Journal of Engineering Research*, 4(1), 2016, 1-20.

- Teena Mittal and Rajendra Kumar Sharma, "Speech recognition using ANN and predator influenced civilized swarm optimization algorithm", *Turkish Journal of Electrical Engineering and Computer Sciences*, In Press.

- Teena Mittal and R.K. Sharma, "SVM parameter optimization using PSO", National Conference on Advanced Computational Methods in Electrical Engineering, 25-26 March 2016, SLIET, Longowal.

- Teena Mittal and R.K. Sharma, "Recognition of Hindi numerals using ANN and SVM", National Conference on Advances in Science and Technology, 3-4 March 2016, RIMT-IET, Mandi Gobindgarh.