Summer 2017

# Speech Based Machine Learning Models for Emotional State Recognition and PTSD Detection

Debrup Banerjee
*Old Dominion University*

# SPEECH BASED MACHINE LEARNING MODELS FOR EMOTIONAL STATE RECOGNITION AND PTSD DETECTION

by

Debrup Banerjee
B.E. July 1999, Shivaji University, India
M.S. July 2004, Hampton University, USA

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

ELECTRICAL AND COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY
August 2017

Approved by:

Jiang Li (Director)

Frederic McKenzie (Member)

Dean Krusienski (Member)

Vishnu Lakdawala (Member)

# ABSTRACT

## SPEECH BASED MACHINE LEARNING MODELS FOR EMOTIONAL STATE RECOGNITION AND PTSD DETECTION

Debrup Banerjee
Old Dominion University, 2017
Director: Dr. Jiang Li

Recognition of emotional state and diagnosis of trauma related illnesses such as post-traumatic stress disorder (PTSD) using speech signals have been active research topics over the past decade. A typical emotion recognition system consists of three components: speech segmentation, feature extraction and emotion identification. Various speech features have been developed for emotional state recognition which can be divided into three categories, namely, *excitation*, *vocal tract* and *prosodic*. However, the capabilities of different feature categories and advanced machine learning techniques have not been fully explored for emotion recognition and PTSD diagnosis. For PTSD assessment, clinical diagnosis through structured interviews is a widely accepted means of diagnosis, but patients are often embarrassed to get diagnosed at clinics. The speech signal based system is a recently developed alternative. Unfortunately, PTSD speech corpora are limited in size which presents difficulties in training complex diagnostic models. This dissertation proposed *sparse coding* methods and *deep belief network* models for emotional state identification and PTSD diagnosis. It also includes an additional *transfer learning* strategy for PTSD diagnosis. Deep belief networks are complex models that cannot work with small data like the PTSD speech database. Thus, a *transfer learning* strategy was adopted to mitigate the small data problem. *Transfer learning* aims to extract knowledge from one or more source tasks and apply the knowledge to a target task with the intention of improving the learning. It has proved to be useful when the target task has limited high quality training data. We evaluated the proposed methods on the *speech under simulated and actual stress database* (SUSAS) for emotional state recognition and on two PTSD speech databases for PTSD diagnosis. Experimental results and statistical tests showed that the proposed models outperformed most state-of-the-art methods in the literature and are potentially efficient models for emotional state recognition and PTSD diagnosis.

# ACKNOWLEDGEMENTS

I would like to offer my sincerest appreciation for the direction, assistance and guidance of Dr. Jiang Li, who is my advisor and whose support and patience has been invaluable. I would also like to sincerely thank the committee and express my deepest gratitude for their valuable guidance. I would also like to acknowledge the support from my wife, my parents and extended family, without which the dissertation would not have been possible. I also gratefully acknowledge the support of NVIDIA Corporation for the donation of the TESLA K40 GPU used in this research.

## TABLE OF CONTENTS

Chapter Page

Chapter Page

]

# LIST OF TABLES

Table Page

Table Page

## LIST OF FIGURES

Figure Page

Figure Page

]

# CHAPTER 1

# INTRODUCTION

## 1.1 BACKGROUND

Speaker *emotional state* identification from speech signal is aimed at identifying the underlying *emotional state.* Emotion recognition from speech is important for human-machine or human-computer interaction. *Emotional state* is expressed by a variety of physiological variations, such as heart beat rate, changes in blood pressure, degree of sweating and can also manifest in shaking, changes in skin coloration, facial expression and the acoustics of speech [1]. It has been shown that emotions such as anger, sadness and fear can be recognized through voice [2].

*Emotional state* recognition could be utilized in various applications. For example, it could be used to judge the authenticity and urgency of an emergency call. It can also be used to route emergency call services for high priority emergency calls. In aircraft cockpits, recognition of stressed speech between air-traffic control and pilots can improve aviation safety [3]. In forensic speech analysis, emotional state identification can also help law enforcement assess the state of telephone callers or aid them in suspect interviews [3].

A typical emotion recognition system takes a speech signal as input and performs feature extraction to extract features. Sometimes, it also conducts feature selection to identify most effective features. Finally, it classifies the speech signal into different emotion categories. In the literature, different types of speech corpora, features and classifiers have been utilized for emotion recognition.

Speech features popularly used in the literature can be categorized into three groups: Vocal tract, prosodic and excitation [4]. Vocal tract characteristics are better described in frequency domain [5], and are strongly correlated with the shape of the vocal tract and the articulator movement [6]. Examples of tract features are Mel-frequency cepstrum coefficients (MFCC), foramants, etc. Examples of excitation features are linear prediction coefficients (LPC) and glottal features [1]. In human speech production, duration, intonation and different intensity patterns are produced which constitute the prosodic features [7]. Examples of prosodic features include minimum, maximum, mean, variance, range and standard-deviation of energy and pitch of the signal [8]. Casale *et al.* in [3], proposed using the genetic algorithm to fuse vocal-tract, prosodic and excitation features to recognize *emotional state.* The three categories of features were combined as a vector and the genetic algorithm worked as a feature selection module to identify a feature subset for the recognition. Features not selected were discarded. For classification, many linear and non-linear classifiers have been explored such as linear discriminant analysis (LDA), Naïve-Bayes classifiers, SVM, Gaussian mixture model (GMM), neural network and Hidden Markov model (HMM), etc. [9].

PTSD is a traumatic-stressor related disorder. It is developed by exposure to a traumatic or an adverse environmental event that caused serious harm or injury. Examples of such events may include torture, severe war zone stress and others. PTSD is a serious problem for military, affecting 30% of military service members who have spent time in war zones. This makes it an important problem to resolve. Currently, many clinical approaches have been explored to diagnose PTSD, while only a few studies have focused on PTSD diagnosis using EEG data or speech. For assessment of PTSD, clinical diagnosis through structured interviews is the only widely accepted means of diagnosis. These diagnoses suffer from certain limitations. The diagnostic criteria for PTSD assessment is questionable and the objective and qualitative measures are limited. Distortions in memory and self-perception of patients also make diagnosis difficult. Patients are often embarrassed and not willing to spend time to come to clinics for diagnosis.

Human speech is affected by the presence of PTSD which makes it a very useful indicator of PTSD status. This can be exploited to build a speech based system for PTSD detection. Speech is non-invasive and can be obtained remotely via telephone or recording media. It can also be used to monitor patient treatment progress. Although speech based diagnosis presents few advantages, a major drawback related to PTSD speech corpora are their limited size. This presents difficulties in training complex diagnostic models. In addition, the capabilities of different feature categories and advancements in the field of machine learning have not been fully exploited for emotion recognition and PTSD diagnosis. This dissertation has attempted to address these limitations by using *sparse coding*, *deep belief network* models and a *transfer learning* strategy to achieve emotion recognition and PTSD diagnosis. *Transfer learning* is proposed specifically to address and mitigate the small data size problem.

Sparse Coding algorithms have recently shown state-of-the-art performances in many applications [54, 55, 56, 57, 58, 59]. In sparse coding, a set of basis functions, named dictionary, was first learned from the data. The dictionary was then used, which served as a building block, in order to reconstruct all the original data samples. Finally, the reconstruction weights were new representations of original data for subsequent classification. Basis functions in sparse coding are learned or selected from data, making the feature extraction process adaptive. We evaluated the proposed sparse coding based method on the SUSAS speech database and the PTSD speech corpus.

Deep learning is a revived technique, which emerged as a result of decades of research in artificial neural networks and have been shown to perform extremely well [79, 80, 81, 82, 83]. These methods can automatically learn features from raw data, without prior knowledge. The downside of using deep learning networks is that they require massive amounts of training data. In such cases, another recent technique known as *transfer learning* can help improve the learning performance when the amount of training data available is small. It works by transferring knowledge in other deep models learned from massive data sets and multiple label categories, using the learned model as a generic feature extractor.

*Transfer learning* was originally defined in 2005, by the Broad Agency Announcement (BAA) 05-29 of the Defense Research Projects Agency (DARPA)'s Information Processing, Technology Office (IPTO) who gave a new mission for transfer learning as the *"the ability of a system to recognize and apply knowledge and skills learned in previous tasks to novel tasks"* [85]. In this definition, *transfer learning* aims to extract the knowledge from one or more *source tasks* and applies the knowledge to a target task with the intention of improving the learning. It aims to extract the knowledge from one or more *source* tasks and applies that knowledge to a *target* task, when the *target* task has fewer high-quality training data. Many machine learning methods work well under the assumption that the training and test data are drawn from the same feature space and same distribution. Most statistical models need to be rebuilt from scratch using newly collected training data if the distribution changes. It becomes an expensive and huge task to acquire and recollect the new training data and rebuild the models [84]. In such a case *transfer learning* finds applicability and becomes very feasible to improve the learning between task domains. *Transfer Learning* is categorized into three major categories, based on different situations between the *source* and *target* domains and tasks. They are *inductive transfer learning*, *transductive transfer learning* and *unsupervised transfer learning.*

The remainder of the dissertation is organized as follows. Section 2 presents related work on speech emotion recognition and PTSD diagnosis. Section 3 describes the proposed method used for emotion recognition and PTSD diagnosis. Section 4 describes, in detail, the experimental procedures, results and discussion related to research on emotion recognition. Section 5 describes the experimental procedures, results and discussion for PTSD diagnosis followed by conclusions in section 6.

## 1.2 CONTRIBUTIONS

The contributions are the following. 1) An efficient speech-driven *sparse coding* framework was developed for emotion recognition which did not exist before. The proposed system, evaluated on the SUSAS data set, outperformed other state-of-the-art algorithms. 2) A speech-driven *sparse coding* and *deep belief net* framework was developed for PTSD detection for the first time. It addressed the limitation of current clinical diagnostic methods heavily reliant on assessment of PTSD based on structured interviews conducted in clinics. 3) The small data size challenge was resolved using the method of *transfer learning.* 4) Novel feature extraction techniques were performed for PTSD detection.

# CHAPTER 2

# RELATED WORK

This chapter introduces the related work done in the field of speech based emotion recognition over the past decade. It describes the features and classification schemes used for emotion recognition from speech. It then provides a detailed description of the related work done on PTSD diagnosis, its history and overview. Features and classification schemes used for PTSD diagnosis are also discussed in this section.

## 2.1 RELATED WORK IN EMOTION RECOGNITION

A lot of research in the emerging area of emotion identification from speech has been carried out in the past decade. This section summarizes most of the work done in this field. The first subsection describes the features and classification schemes used for identifying emotion. The second, deals with the vocal acoustic characteristics used. The third section describes the neural network models and it's extension known as deep learning for emotion recognition.

### 2.1.1 FEATURES AND CLASSIFICATION SCHEMES FOR EMOTION RECOGNITION FROM SPEECH

This section describes the types of speech features and classification schemes used in emotion recognition and the related work done so far. First, an overview of the speech feature extraction process and it's details are presented followed by related work done. Speech signals include emotion information as well as data. Speech signals are not stationary, meaning that their amplitudes have a lot of variance over time. It is common in speech processing to divide a speech signal into short time-duration units called 'frames' over which they are approximated to be stationary [11]. Features such as pitch and energy are extracted from each speech frame and are called local features. On the other hand, global features are computed as statistics of all speech features extracted from an utterance [12]. There is disagreement about whether local or global features are more suitable for emotion recognition, but most researchers agree that global features seem to be superior when applying cross validation and feature selection algorithms. They also take less time to execute than local features. Although they are more efficient, researchers have claimed that global features are effective only in discriminating between high-arousal emotions like *anger*, *fear* and *joy* versus low-arousal emotions such as *sadness* [13]. Another approach for feature extraction, is based on segmenting speech signals to the underlying phonemes and then calculating one feature vector for each segmented phoneme [14]. This approach relies on a study that observes variation in the spectral shapes of the same phone under different emotions [15]. Overall, speech features can be grouped into three major categories: prosodic, vocal-tract and excitation features.

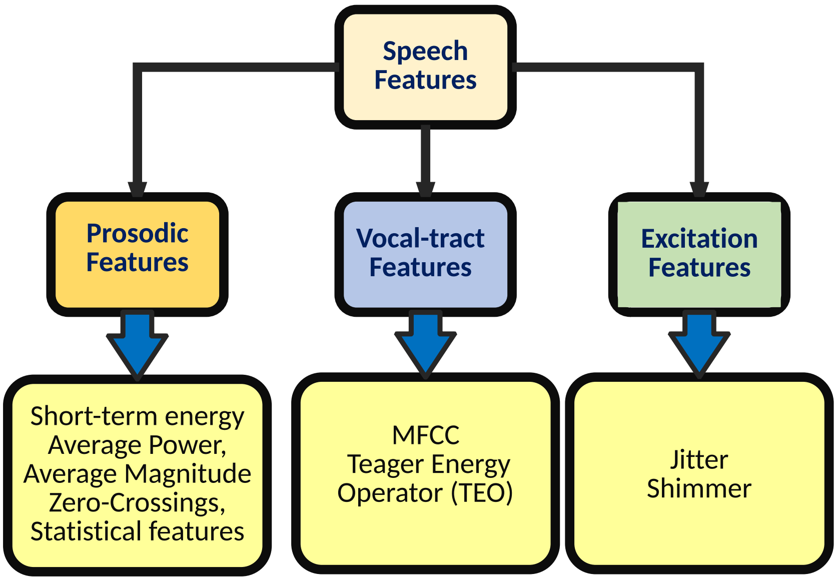
Speech Features
Prosodic Features
Vocal-tract Features
Excitation Features
Short-term energy Average Power, Average Magnitude Zero-Crossings, Statistical features
MFCC Teager Energy Operator (TEO)
Jitter Shimmer

**Figure 1:** Three different categories of speech features.

In [16], I. Luengo *et al.* proposed using continuous prosodic features on the *Basque* speech database with three different feature-classifier combinations. The first, using spectral features and the Gaussian Mixture Model (GMM) classifier, a second combination, using other prosodic features and the support vector machine (SVM) and a third, using prosodic features and GMM. Feature selection was carried out on 86 extracted features. The first classifier gave the best result with 98.4% accuracy when using 512 mixtures, but the best 6 prosodic features achieve 92.3% showing that they are effective in identifying emotions. In [17], Wu et al., proposed using features computed from the long term, spectro-temporal speech representation and comparing them to short-term spectral features as well as popular prosodic features on the *Berlin* speech database. It showed that these computed spectro-temporal features outperformed the others and achieved an overall accuracy of 88.6% by using a combination of the proposed and prosodic feature set for classifying the seven discrete emotions in the *Berlin* database.

Out of the several studies on the *Berlin* speech database, Iliou *et al.* in [28] focused on comparing classifiers for emotion recognition. Speaker-dependent and speaker-independent scenarios were considered. 133 speech features were obtained out of which a subset of 35 features were selected using the statistical method and classified using the ANN and the random forest classifiers. Seven emotion categories were used. In speaker dependent framework, ANN classification reached an accuracy of 83%, and random forest reached 77%. In the speaker independent framework, for ANN classification, a mean accuracy of 55% was reached, while random forest reached a mean accuracy of 48% [28].

Nwe *et al.* in [18] proposed comparing three different feature-classifier combinations in terms of classification performance in detecting emotional stress. The first system made use of linear short-time log frequency power coefficients (LFPC). The second employed Teager Energy Operator (TEO) based non-linear frequency domain LFPC features (NFD-LFPC) and a third system used TEO based non-linear time domain LFPC features (NTD-LFPC). The classifier used was a continuous density five state hidden markov model (HMM) with two Gaussian mixtures per states for each stress cycle. For the system using LFPC features, an average accuracy of 84% and a best accuracy of 95% were obtained.

Extraction of features from word-level utterances by animated conversational agents was proposed by Hoque *et al.* in [25]. The features included a total of 22 prosodic and acoustic features. Utterance level statistics related to the fundamental frequency were also computed. The speech processing software called *Praat* was used for this purpose. Then the extracted features are projected on to a lower dimensional space using principal component analysis and linear discriminant analysis (LDA) is applied for a clustered representation of the computed features. Finally, the models are learned using machine learning techniques from the training samples by using *WEKA*, a machine learning toolbox to classify between two states of emotion, the positive and negative states. An evaluation of the models is also carried out. The first model fed the raw 22 features directly into the classifier. The second and the third model applied PCA on the raw features and took the first 15 and 20 eigenvectors respectively to de-correlate the base features. In the fourth model, LDA is directly used on the raw features to project them directly onto the lower dimension. The fifth model consisted of the combination of principal component analysis (PCA) and Linear Discriminant Analysis (LDA). A 10-fold cross validation technique was used. Results showed that the combination of data projection techniques such as PCA and LDA yielded better performance as opposed to using raw features or using LDA or PCA alone. An average accuracy of 83.33% was achieved using the combination of PCA and LDA. The performance of combining PCA and LDA is higher than PCA or LDA itself mainly because PCA de-correlates the data, whereas LDA projects the data onto a lower dimension. Therefore, the combination of PCA and LDA is expected to work better. Robust autonomous recognition of emotion is gaining attention due to the widespread applications into various domains, including those with animated conversational agents.

In [19], Neiberg *et al.* proposed modeling pitch by utilizing mel-frequency cepstrum coefficients (MFCC). A 25.6 ms hamming window for every 10ms shift was used and a variant of that called MFCC-low (filter banks placed in the 20-300 Hz region) was also utilized. Plain pitch features were also extracted using the average magnitude difference function algorithm (AMDF). These features were modeled using a GMM classifier over two sets of speech databases and languages, *Swedish Voice Controlled Telephone Services* and *English Meetings.* Results indicated that using GMM's at frame level was a feasible technique for emotion classification. It has been observed that current text-to-speech systems have very good intelligibility, but most are still easily identified as artificial voices and no commercial system incorporates prosodic variation resulting from emotion and related factors [20].

A large breadth of objectively measurable features in discriminating depressed speech was explored by Moore II *et al.* in [10]. Features included those related to prosody, the vocal tract features and parameters extracted directly from the glottal waveform. Feature combinations were formed which included prosodic features, prosodic and vocal tract features, prosodic and glottal features and prosodic, vocal tract and glottal features. Results of classification using the fisher discriminant analysis indicated that the combination of glottal and prosodic features produced better performance overall showing that glottal descriptors are vital components of vocal affect analysis.

Prosody related features were explored by Bozkurt *et al.* in [21]. They included mean and normalized values of pitch, first derivative of pitch and intensity, spectral features like MFCC and line spectral frequency features and their derivatives. Additionally, HMM based features were also explored for the evaluation of emotion recognition with a GMM based classifier. A fusion of different feature sets and classifiers was applied to evaluate classification performance based on the *InterSpeech 2009 Emotion Challenge Corpus* containing highly emotional and spontaneous recordings.

In [22], Zhou *et al.* proposed three new derivative features of the non-linear Teager Energy Operator (TEO) feature as good stress indicators. It is believed that the TEO based features are able to better model the non-linear airflow structure of speech production under adverse stressful situations. The proposed features included TEO-decomposed-FM-Variation (TEO-FM-Var), normalized TEO autocorrelation envelope area (TEO-Auto-Env) and critical band based TEO autocorrelation envelope area. These features are evaluated for simulated and actual stressed speech and it was demonstrated that the TEO-CB-Auto-Env feature outperformed pitch and MFCC features by a very large margin. The overall neutral-stress classification rates were also shown to be more consistent across different stress styles.

In [23], Koolagudi, *et al.* proposed using linear prediction (LP) residual samples as features on a semi-natural speech database called *GEU Semi Natural Speech Corpus* (GEU-SNESC) for obtaining emotion specific information. The emotions considered for this work included *sadness*, *anger*, *happiness* and *neutral* emotions. The linear prediction (LP) residual of the speech signal (obtained by inverse filtering of the speech signal) was used for characterizing the basic emotions present in speech. GMM's were used to capture the higher order relations present in the LP residual. The emotion recognition performance achieved was about 50-60%.

The use of the k-nearest neighbor method to classify utterances was proposed by Lee *et al.* in [24] to classify emotions as either being *negative* or *non-negative.* Also, linear discriminant classification with Gaussian class-conditional probability distribution was used. The features used by the classifiers were utterance level statistics of the fundamental frequency and energy of the speech signal. Two feature selection methods, promising first selection and forward feature selection, were used. Principal component analysis, PCA, was used to reduce the dimensionality of the features. Gender specific experiments were carried out since pitch related features are very different between male and female genders, especially the mean, *max* and *min* of the fundamental frequency.

Dellaert *et al.* in [9] proposed a new method of extracting prosodic features based on a smoothing spline approximation of the pitch contour. They built a speech corpus containing emotional speech containing 1000 utterances from different speakers. Majority voting of subspace specialists, a novel pattern recognition technique was used to obtain a good classification performance.

Features based on the glottal airflow signal were utilized to evaluate the classification performance in seven different classification schemes by Iliev *et al.* in [26]. It's effectiveness was tested on the new optimum path classifier (OPF) as well as on six other previously established classification methods such as the Gaussian mixture model (GMM), support vector machine (SVM), artificial neural networks–multi layer perceptron (ANN-MLP), k-nearest neighbor rule (k-NN), Bayesian classifier (BC) and the C4.5 decision tree. The speech database used in this work was collected in an anechoic environment with ten speakers, of which five were male and five female speakers, each speaking ten sentences in four different emotions: *happy*, *angry*, *sad*, and *neutral*. The glottal waveform was extracted from fluent speech via inverse filtering. The investigated features included the glottal symmetry and MFCC vectors of various lengths both for the glottal and the corresponding speech signal. Experimental results indicated that the best performance was obtained for the glottal-only features with SVM and OPF generally providing the highest recognition rates. For GMM, or the combination of glottal and speech features, performance was relatively inferior [26]. For this text dependent, multi speaker task the top performing classifiers achieved perfect recognition rates for the case of 6th order glottal MFCCs. Results confirmed that glottal information is rich in emotional clues and presents a very effective source for achieving recognition for spoken emotion. Best classification performance was provided by SVM and OPF. The lowest performance was that of GMM. In terms of computation time, k-NN was the fastest. It was also observed that OPF was much faster than SVM.

In [27], Lugger *et al.* proposed utilizing the bayesian classifier to classify six emotion categories, based on the extraction of over 200 prosodic features like pitch, energy and duration from a speech corpus. Voice quality parameters (VQP) describing the properties of the glottal source were also used. The feature set used was a parameterization of the voice quality in the frequency domain by spectral gradients. The VQP are reported to have good discrimination capacity with regard to emotion. Around eight VQP features were extracted. Feature selection was applied to reduce the number of features by using the sequential floating forward selection algorithm. It's an iterative method to find the best subset of features. A total of six emotion categories were classified [27]. A leave-one-speaker-out cross validation (LOSO-CV) was used for speaker independent classification. The Bayesian classifier was used for all scenarios. The class conditional densities were modeled as unimodal gaussians. A change in classification performance was observed by altering the number of gaussians in the GMM. Classification was also carried out using combined feature sets. It was seen that the parameters of voice quality show a contribution in addition to the well-known prosodic features.

### 2.1.2 VOCAL ACOUSTIC CHARACTERISTICS APPLIED TO EMOTION RECOGNITION FROM SPEECH

In this section, work done on the correlation between vocal acoustic features and depression or negative mental states is described. Speech recordings were made of sixteen depressed patients during depression after clinical improvement. The recordings were then analyzed using a computer program that extracts acoustic parameters from the fundamental frequency contour of the voice. The percent pause time, the standard deviation of the voice fundamental frequency distribution, the standard deviation of the rate of change of the voice fundamental frequency and the average speed of voice change were found to correlate to the clinical state of the patient. The mean fundamental frequency, the total reading time and the average rate of change of the voice fundamental frequency did not differ between the depressed and the improved group [29]. The acoustic measures were less strongly correlated to the depressive symptoms such as retardation or agitation and more pronounced in correlation to the clinical state of the patient as measured by global depression scores.

Several studies have documented speech motor impairment in the case of patients suffering from Parkinson's disease (PD). In this study, a retrospective analysis of speech was conducted on two well-known individuals with PD and two matched controls to determine if certain acoustic measures were sensitive markers of early pathophysiologic changes or treatment response in PD. Acoustic analyses were conducted on samples of speech produced over a 10-year period surrounding the time of disease diagnosis. Analyses revealed that, for both PD cases, a decrease in fundamental frequency variability during free speech was detected prior to clinical diagnosis. Changes in fundamental frequency variability and voice onset time (VOT) were also detected upon the initiation of symptomatic treatment. In a second experiment, an acoustical analysis of speech production was conducted on four newly diagnosed persons with PD and four matched controls, using a standard speech examination protocol [30].

Among the many empirical studies conducted to investigate the relationship between acoustical measures of voice and speech to that of severity of clinical depression, one study focused on exploring this relationship using the 17 item Hamilton Depression Rating Scale (HDRS). Pilot data were obtained from seven subjects that included five males and two females, from videotapes used to train expert raters on the administration and scoring of the HDRS. Several speech samples were isolated for each subject and processed to obtain the acoustic measurements. Acoustic measures were selected on the basis that they were correlated with HDRS ratings of symptom severity as seen under ideal voice recording conditions in previous studies. The findings corroborate earlier reports that speaking rate is well correlated (negatively) with HDRS scores, with a strong correlation and nearly significant trend seen for the measure of pitch variability. A moderate pairwise correlation between percent pause time and HDRS score was also revealed, although this relationship was not statistically significant [31].

In recent years, the problem of automatic detection of mental illness from the speech signal has gained some initial interest; however, questions remain that include how speech segments should be selected, what features provide good discrimination, and what benefits feature normalization might bring given the speaker-specific nature of mental disorders. Feature normalization is applied to reduce the mismatch between different speakers. In this work, classifier configurations are employed in emotion recognition from speech, evaluated on a 47-speaker depressed/neutral read sentence speech database. Results demonstrate that detailed spectral features are well suited to the task and that speaker normalization provides benefits mainly for less detailed features. It also shows that dynamic information appears to provide little benefit. Classification accuracy using a combination of MFCC and formant based features approached 80% for this database.

### 2.1.3 NEURAL NETWORKS AND DEEP LEARNING MODELS FOR IDENTIFYING EMOTION FROM SPEECH

Although automatic emotion recognition systems have seen improvements by way of crafting features that give reasonable good accuracy by using feature selection methods, still they are only able to capture only linear relationships between the features a majority of the time. Neural networks and deep learning techniques can capture complex non-linear feature interactions in the data. Deep belief network models thus show an improvement in classification accuracy over baselines that do not use these models. In one such study, two methodologies are compared, unsupervised feature learning using DBN and secondary supervised feature selection. First an unsupervised two-layer DBN is built, enforcing multi-modal learning. The DBN is augmented with two types of feature selection: 1) before DBN training to assess the benefit of feature learning exclusively from an emotionally-salient subset of the original features and 2) after DBN training to assess the advantage of reducing the learned feature space in a supervised context. This is compared to the performance of a three-layer DBN model [32]. The baseline is an SVM that uses subsets of the original feature space selected using supervised and unsupervised feature selection. The results provide important insight into feature learning methods for multimodal emotion data [32]. The results show that the DBN models outperform the baseline models. This suggests that unsupervised feature learning can be used in lieu of supervised feature selection for this data type.

In addition, the relative performance improvement of the three-layer model for subtle emotions suggests that these complex feature relationships are particularly important for identifying subtle emotional cues. Deep neural networks (DNN) denote multilayer artificial neural networks with more than one hidden layer and millions of free parameters. Another study proposed a Generalized Discriminant Analysis (GerDA) based on DNN to learn discriminative features of low dimension optimized with respect to a fast classification from a large set of acoustic features for emotion recognition. On nine frequently used emotional speech corpora, the study compares the performance of GerDA features and their subsequent linear classification with previously reported benchmarks obtained using the same set of acoustic features classified by the SVM. Results show that low-dimensional GerDA features capture hidden information from the acoustic features leading to a significantly raised unweighted average recall and considerably raised weighted average recall [33].

In another study, a novel staged hybrid model for emotion detection in speech is proposed. A hybrid model is used since hybrid models exploit the strength of discriminative classifiers along with the representational power of generative models. Temporal deep networks are capable of capturing the representation of a more temporally rich set of problems. Temporal deep networks include conditional RBM's (CRBMs), and temporal RBMs (TRBMs). CRBMs and TRBMs have been successfully used in the audio domain, for example, *phone* recognition, and polyphonic music generation. Recently, deep stacking networks, a special type of deep model equipped with parallel and scalable learning, have been successfully used for frame-level phone classification, *phone* recognition, and information retrieval. A brief summary of the related work in emotion recognition is presented in Table 1.

## 2.2 RELATED WORK ON PTSD DIAGNOSIS

### 2.2.1 BACKGROUND AND OVERVIEW

PTSD is a traumatic-stressor related disorder. It is developed by some people when they are exposed to a traumatic or an adverse environmental event that caused serious harm or injury. Examples of such events may include genocide, torture, severe war zone stress and others. Symptoms are marked by negative cognitions and mood states as well as disruptive behavioral symptoms [35]. PTSD is a serious problem for the military, affecting 30% of military service members who have spent time in war zones. Today, PTSD is recognized as a psychobiologial mental disorder that can affect survivors of combat experience, terrorist attacks, natural disasters and serious accidents, assaults, abuses and sudden major emotional losses. In 1980, the American Psychiatric Association (APA) added PTSD to the third edition of Diagnostic and Statistical Manual of Mental Disorders *DSM-3* nosologic classfication scheme. Currently, *DSM-5*, is the latest revised criteria for assessment of PTSD. It contains several criteria defined in the form of alphabets, A-H, each alphabet corresponding to a certain criterion. The significant change from a historical perspective was that the causing agent, for example a traumatic event, was outside the individual rather than an inherent individual weakness.

The study of how strong emotions such as fear are linked to memory formation and retrieval are key to PTSD clinical research [36]. Diagnosis of PTSD is mostly based on patient-self reporting during clinical interviews. Few objective or qualitative measures are available

**Table 1:** A brief summary of related work on emotion recognition based on speech.

| Method Used | Features Used | Results |
|---|---|---|
| TEO based framework | TEO based features extracted from SUSAS | 92.9% for pairwise text-dependent scenario, 89% for pairwise text-independent scenario, 88.85% for text-independent multistyle scenarios |
| Adaptive sinusoidal model based | Sinusoidal based extracted from SUSAS | Average 64.25% for multiclass |
| Multi-level classification framework on resting-state fMRI (Multi-kernel SVM) | Univariate, bivariate and multivariate features derived from fMRI | 92.5% classification accuracy |
|  | Pitch, log energies, MFCC's, velocity and acceleration features extracted from SUSAS | 91.3% for pairwise text-independent scenario, 70.1% for text-independent multistyle scenario |
| Integration framework | MFCC, delta and acceleration coefficients extracted from SUSAS | Best accuracy of 83.8% |
| Long Short Term Memory Neural network framework | MFCC and Lyon Cochleagram Model extracted from SUSAS | Best accuracy of 75.41% |
| Three different feature classifier combinations (spectral features + GMM, prosodic features + SVM, prosodic features + GMM) | Features extracted from BERLIN speech database | (Best of 98.4% by spectral features + GMM) |
| Spectro-temporal framework | Long term spectro-temporal features proposed and comparing to short-term spectral features and prosodic features on BERLIN speech database | Overall accuracy of 88.6% using combination of proposed and prosodic feature set |

to help clinicians diagnose this condition. Certain factors make diagnosis a more challenging proposition. Some of these factors are distortions in memory and self-perception. Currently, standardized diagnostic interviews such as the Structured Clinical Interview for *DSM-4 Axis*

*I* Disorders (SCID) is used for PTSD diagnosis. Another example of such an interview is the *Clinician-Administered PTSD Scale* (CAPS) interview which is also a gold standard in PTSD diagnosis. To address the growing needs, it is necessary to find a more objective and time-efficient way to diagnose PTSD. In some cases, patients are embarrassed and not willing to visit the clinic for clinical interviews and spend time being interviewed. One such modality is speech. Speech is a non-invasive, inexpensive and useful indicator of PTSD status of a person. It provides an indicator of the patient's condition and can also be used to monitor patient treatment progress. Another advantage is that it can also be obtained remotely via phone for analysis.

### 2.2.2 FEATURES AND CLASSIFICATION SCHEMES FOR PTSD DIAGNOSIS

This section describes the types of speech features and classification schemes used for PTSD diagnosis and the related work done so far. In [37], Vergyri *et al.* explored three feature categories which included lexical, spectral and longer-range prosodic features. PTSD recordings from the standardized CAPS interview taken by military personnel were used to extract the features. Classification schemes included the gaussian backend, decision tree and neural network classifiers. An overall accuracy of 77% was achieved. It also concluded that spectral and prosodic features outperformed lexical features. Multi-view learning, a genre of learning that uses heterogeneous subsets of a data collection was utilized by Zhuang *et al.* in [38]. Both speech and EEG data are used during training while only speech data is used for detection. Results show that multi-view learning outperforms both speech-only and EEG-only methods. Two classifiers, the gaussian naive-bayes and the linear SVM are used in this study. It was demonstrated that there was a net relative increase between 20% and 37% in speech-based PTSD detection.

Liu *et al.* in [39] proposed applying a multi-level classification framework on resting-state, functional magnetic resonance imaging (fMRI) for emotion detection. A multi-kernel SVM was used for this purpose. A classification accuracy of 92.5% was achieved.

Zhang *et al.* in [40], performed multi-modal MRI based classification of PTSD. Structural and resting state fMRI were collected from three categories of individuals. These included PTSD patients, trauma-exposed controls without PTSD (TEC) and non-traumatized healthy controls (HC). Three different types of features were extracted to integrate the information of structural and functional MRI data. The extracted features were combined by a multi-kernel combination strategy. An SVM classifier was trained to distinguish the subjects at the individual level. The performance of the classifier was evaluated using the leave-one-out cross-validation (LOOCV) method. In the pairwise comparison of PTSD, TEC, and HC groups, classification accuracies obtained by the proposed approach were 2.70%, 2.50%, and 2.71% higher than the best single feature way, with accuracies of 89.19%, 90.00%, and 67.57% for PTSD against HC, TEC versus HC, and PTSD versus TEC respectively. The proposed approach was found to improve PTSD identification at the individual level.

A sparse, combined regression-classification scheme for learning a physiological alternative to clinical PTSD scores was proposed by Brown *et al.* in [41]. This work utilized a novel experimental set-up, exploiting virtual reality videos and peripheral physiology for PTSD diagnosis. In pursuit of an automated physiology-based objective diagnostic method, a learning formulation that integrates the description of the experimental data and expert knowledge on desirable properties of a physiological diagnostic score was proposed by Brown *et al.* in [41]. The physiological score produced by the sparse, combined regression-classification is assessed with respect to three sets of criteria chosen to reflect design goals for an objective, physiological PTSD score, parsimony and context of selected features, diagnostic score validity, and learning generalizability. For these criteria, the work demonstrated that sparse, combined regression-classification performs better than more generic learning approaches.

Karstoft *et al.* in [42] used the *target information equivalence* approach to identify a set of features based on *markov boundary* and used SVM for classification. The *target information equivalence* algorithm identified all minimal sets of features, *markov boundaries*, that maximized the prediction of a non-remitting PTSD symptom trajectory when integrated in a support vector machine (SVM). The predictive accuracy of each set of predictors was evaluated in a repeated 10-fold cross-validation and expressed as average area under the Receiver Operating Characteristics curve for all validation trials [42]. The study concluded the hypothesized existence of multiple and interchangeable sets of risk indicators that equally and exhaustively predict non-remitting PTSD.

A markov boundary based feature selection scheme was proposed by Levy *et al.* in [43], known as the *markov boundary induction algorithm for generalized local learning.* Six different classification schemes were used which included variations of the linear SVM, polynomial SVM, random forests, adaboost, kernel-ridge regression with the radial basis function and the bayesian binary regression. The study concluded that machine-learning algorithms were feasible for PTSD diagnosis and that the approach was a promising one. A brief summary of the related work on PTSD diagnosis is presented in Table 2.

**Table 2:** Summary of related work on PTSD detection based mainly on speech and electroencephalogram (EEG).

| Method Used | Features Used | Results |
|---|---|---|
| Gaussian backend, decision tree and neural network | Lexical, spectral and longer-range prosodic features | Overall 77% accuracy. Spectral and prosodic outperformed lexical |
| Multi-view learning (speech+ EEG) Gaussian naïve bayes and linear SVM | Common speech and EEG features | Net relative increase of 20% to 37% in speech-based PTSD detection |

This dissertation makes several contributions. 1) An efficient speech-driven, *sparse coding* framework was developed for emotion recognition which did not exist before. The proposed system which was evaluated on the SUSAS data set achieved better performance, compared to other state-of-the-art algorithms. 2) A speech-driven *sparse coding* and *deep belief net framework* was developed for PTSD detection for the first time. It addressed the limitation of current clinical diagnostic methods discussed previously. 3) The small data size challenge was resolved by adopting a *transfer learning* strategy. 4) Novel feature extraction methods were also employed for PTSD diagnosis.

# CHAPTER 3

# PROPOSED METHOD

This chapter starts with an introduction to the speech feature extraction process. It specifies the different categories of speech features utilized for emotional state recognition and PTSD diagnosis. It also provides detailed feature descriptions. The next section briefly describes the TIMIT and PTSD feature extraction processes. The proposed method which includes a total of three different models is then discussed in detail. A brief discussion of principal components analysis is then presented. Relevant details of classification using the support vector machine are then presented in the final section of this chapter.

## 3.1 PROPOSED SPEECH BASED EMOTION RECOGNITION AND PTSD DIAGNOSTIC MODELS

Figure 2 shows the diagram of the proposed system. First, pre-emphasis filtering is applied to the input speech signal as a pre-processing step. The system then extracts features from the pre-processed speech segments and separates out the voiced frames. Using these voiced frames, it then performs emotion recognition and PTSD diagnosis by applying either *sparse coding*, *deep belief network* or *transfer learning* models. In the following subsections, the components of the system are described in detail.

### 3.1.1 SPARSE CODING MODEL FOR EMOTION RECOGNITION AND PTSD DIAGNOSIS FROM SPEECH

Sparse-coding has achieved state-of-the-art performance in many applications including computer vision [54, 55, 56, 57, 58]. The goal of sparse coding is to represent input vectors approximately as a weighted linear combination of a small number of 'basis' functions. This basis set is usually overcomplete (number of basis functions is larger than its dimension) and therefore can capture a large number of patterns in the input data. Given a training dataset $A = (a^i)_{i=1}^{N_v}$, a dictionary $D = (d^i)_{i=1}^{K_d}$, consisting of a set of basis functions, $d^i$ , can be learnt based on an L1-penalized sparse coding formulation by optimizing the following cost function,

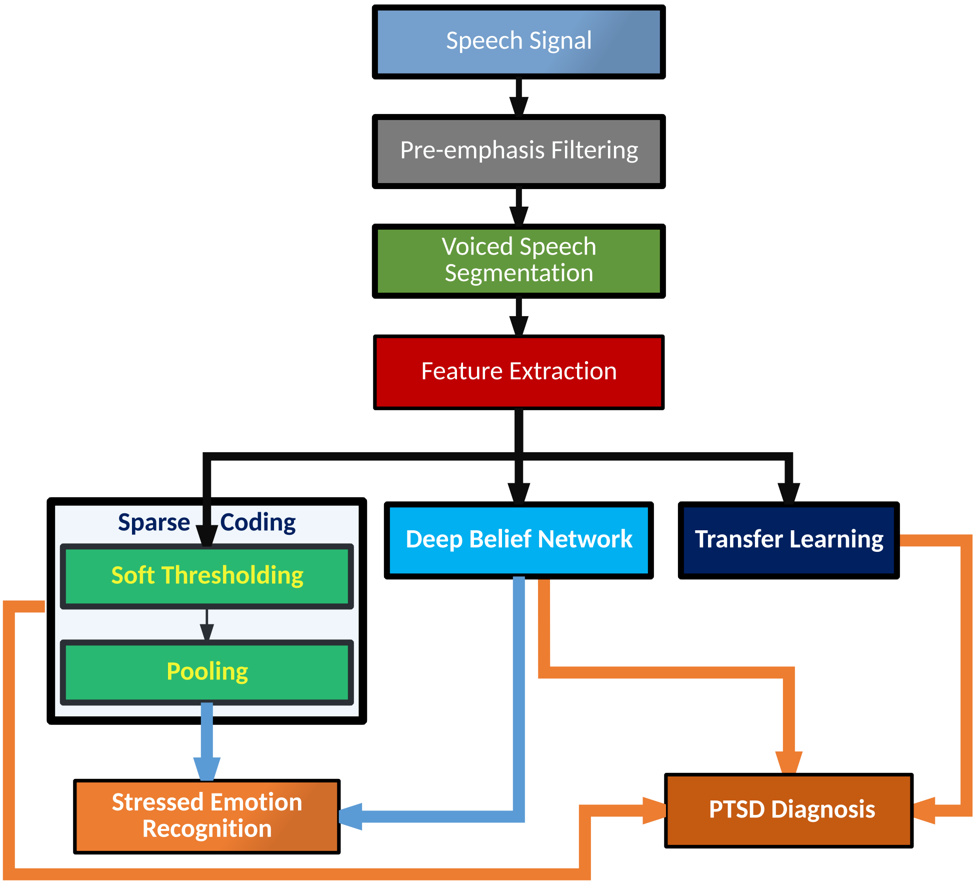

**Figure 2:** The proposed models.

$$min_{D,\beta} \|D\beta^{(i)} - a^{(i)}\|^2 + \lambda\|\beta^{(i)}\|_1, \tag{1}$$

$$subjectto \|d^{(i)}\|_2^2 = 1 \qquad \forall i \tag{2}$$

where $a^{(i)}$ represents the $i-th$ data sample in $A$, $\beta^{(i)}$ denotes the reconstruction weight for $a^{(i)}$ using the basis function in $D$ and $\lambda$ is a trade-off parameter. Because of the $L_1$ norm penalty, the resulting weights, $\beta^{(i)}$ will be sparse, meaning that most of them will be zeros. The solution of the above equation can be obtained using alternating minimization over the sparse codes and dictionary while holding the other fixed. Dictionary learning plays an important role in sparse coding framework because it will identify those building blocks from data [59]. However, the learning process is time-consuming and difficult. Recent research has shown that randomly selected dictionaries can also perform well [59]. In this work, we will apply the random dictionary learning method. Once the dictionary was learnt, an encoding step was performed to transform the input data samples into desirable representations based on the learnt dictionary. For a particular data sample, $a^{(i)}$, its representation $\beta^{(i)}$ can be obtained either by solving equation 1 with $D$ fixed or by the soft-thresholding method which achieves the sparse representation for $a^{(i)}$ by the following operation,

$$\beta^{(i)} = sgn(z^{(i)})max(0, |z^{(i)} - t|) \tag{3}$$

where t is an adjustable threshold and,

$$z^{(i)} = D^T a^{(i)} \tag{4}$$

Finally, a feature pooling step was applied to reduce the high dimensionality of the new feature space. In the feature pooling step, we divided the feature vector into separate quadrants and then computed the average of the four quadrants as new features such that the dimensionality was reduced by a factor of four.

### 3.1.2 DEEP BELIEF NETWORK MODEL FOR EMOTION RECOGNITION AND PTSD DIAGNOSIS FROM SPEECH

#### 3.1.2.1 MOTIVATION

Hinton proposed the first successful deep learning system in 2006 by applying RBM to pre-train the deep structure layer by layer. In the past few years, deep learning became more and more popular in both academia and industry because of its superior performance in many different applications. For instance, using deep learning models, Deng et al. achieved state-of-the-art performance in speech recognition on several benchmark datasets [81] and Dieleman et al. obtained excellent results on music signal processing [82]. This method can be used for reducing dimensionality [84] and also for classification [83]. Figure 3 shows the general layout of the deep belief network with multiple RBM's stacked together.

#### 3.1.2.2 METHODOLOGY

The method of constructing the network is described as follows.,

1) Initialize the weights using random numbers. 2) Pretain multiple layers of feature detectors by learning a stack of restricted Boltzmann machines (RBMs) in an unsupervised way. 3) Use the labels as ground truth and perform supervised fine-tuning using the backpropagation algorithm.

#### 3.1.2.3 PRE-TRAINING

Once the first layer of RBM is built, its outputs (feature detectors) are used as inputs of the next layer to learn the next RBM. The procedure can be repeated so that feature representations are learned layer by layer. With this greedy, layer-by-layer learning mechanism, a deep structure with any number of layers of RBMs can be built. Such a stacking of RBM's forms a deep belief net. It is a type of undirected, generative, energy-based deep neural network composed of multiple layers of latent variables with connections between the layers but no intra-layer connections, or connections between the units themselves in a given layer.

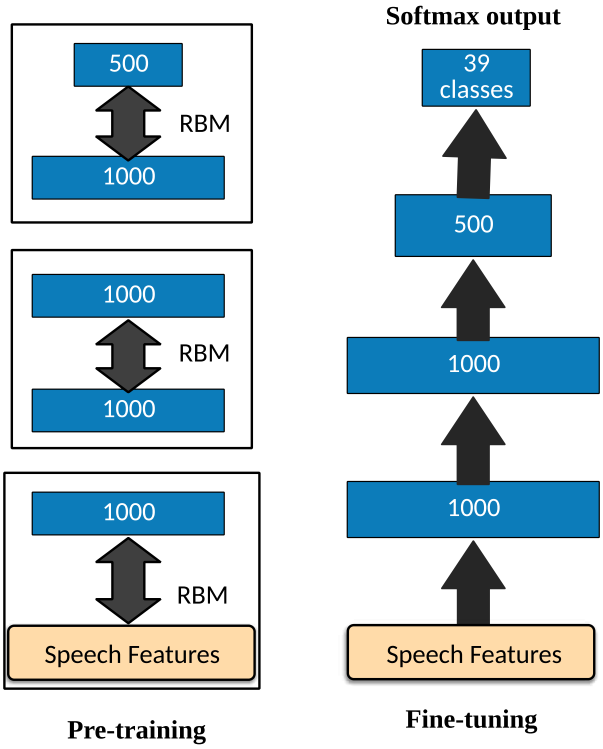

**Figure 3:** General layout of a deep belief network used for classification. Multiple RBM's are stacked together. The network is initially generatively pre-trained, where the stacked RBM's are trained greedily, layer by layer and then discriminatively fine-tuned using labels.

The procedure of pre-training consists of learning stacks of restricted Boltzmann machines, each of which is two-layered. One layer consists of the *visible* units whose states are observable and the other layer consists of *hidden* units whose states are unobservable and are the feature detectors. RBM's use symmetrically weighted connections to transform visible units to stochastic binary feature detectors. An energy function for the visible and hidden units is defined as,

$$E(v,h) = -\sum_{i \in input} b_i v_i - \sum_{j \in fea} b_j h_j - \sum_{i,j} v_i h_j w_{ij} \tag{5}$$

where $v_i$ and $h_j$ are the binary states of input unit $i$ and feature $j$, $b_i$ and $b_j$ are the biases for input $i$ and feature $j$ and $w_{ij}$ is the weight between them.

The probability of the visible vector can be determined from the following equation,

$$p(v) = \sum_{h \in H} p(v,h) = \frac{\sum_h exp(-E(v,h))}{\sum_{u,g} exp(-E(u,g))} \tag{6}$$

where $H$ is the set of all possible binary hidden vectors.

The parameters of the RBM are determined as,

1) Given a visible vector (speech input), the binary state of $h_j$ of the feature is set to 1 with a probability of $\sigma(b_j + \sum_i v_i w_{ij})$, where $\sigma(x)$ is the sigmoid function $\frac{1}{1+exp(-x)}$.

2) Given $h_j$ from previous step, a reconstruction of the visible vector is achieved by setting each $v_i$ to 1 with a probability of $\sigma(b_i + \sum_j h_j w_{ij})$.

3) Update weights using the following update rule,

$$\Delta w_{ij} = \epsilon(< v_i h_j >_{data} - < \widehat{v_i} hj >_{recon}) \tag{7}$$

where $\epsilon$ is the learning rate, $< v_i h_j >_{data}$ is the fraction of times the input unit $i$ in the visible vector and feature detector are on together, and $< \widehat{v_i} h_j >_{recon}$ is the fraction for the reconstruction of the input. After the first RBM layer has been built, its feature detectors now become the visible units to learn the next RBM. The procedure can repeat, so that higher levels of RBM's are learnt one by one. Using this mechanism, a network with any number of RBM layers can be built. Features in deeper layers tend to capture strong and high order correlations between units in the lower layers.

#### 3.1.2.4 FINE-TUNING

After pre-training, the weights of the pre-trained model are fine-tuned using a final label layer. The label layer is considered as the ground truth. The label layer is added on top of the pre-trained structure. Performing supervised fine-tuning of the network using the label information forms a deep classifier. The layer prior to the label layer in the deep classifier is the new representation of the original features. The standard backpropagation algorithm optimizes the weights and the mini-batch gradient descent algorithm is used to optimize the network with respect to a supervised training criterion.

### 3.1.3 TRANSFER LEARNING FOR PTSD DIAGNOSIS

*Transfer learning* was originally defined in 2005, by the BAA 05-29 of the Defense Research Projects Agency (DARPA)'s Information Processing, Technology Office, who gave a new mission for transfer learning as the *the ability of a system to recognize and apply knowledge and skills learned in previous tasks to novel tasks* [85]. In this definition, *transfer learning* aims to extract the knowledge from one or more *source tasks* and applies the knowledge to a target task with the intention of improving the learning. It aims to extract the knowledge from one or more *source* tasks and applies that knowledge to a *target* task, when the *target* task has fewer high-quality training data. Many machine learning methods work well under the assumption that the training and testing data are drawn from the same feature space and same distribution. Most statistical models need to be rebuilt from scratch using newly collected training data if the distribution changes. It becomes an expensive and huge task to acquire and recollect the new training data and rebuild the models [84]. In such a case *transfer learning* finds applicability and becomes very feasible between task domains.

*Transfer Learning* is categorized into three major categories, based on different situations between the *source* and *target* domains and tasks. They are *inductive transfer learning*, *transductive transfer learning* and *unsupervised transfer learning.* In the *inductive transfer learning* setting, the *source* and the *target* are different irrespective of the *source* and *target* domains. In this setting, some amount of labeled data is required to be available in the *target* domain in order to induce a predictive model in the *target* domain. Further, depending on different situations related to labeled and unlabeled data in the *source* domain a further categorization can be made for the *inductive transfer learning.* In the first scenario, lots of labeled data are available in the source domain. This is similar to the *multitask learning* setting. The difference is while *multitask learning* simultaneously tries to learn both the *source* and *target* tasks, *inductive transfer learning* tries to achieve a high performance in the *target* task by transferring knowledge from the *source* task. In the second scenario, the source domain does not have any labeled data available, and in this case the *inductive transfer learning* setting mimics the *self-taught learning* [87] setting.

The second category is called *transductive transfer learning*, where the *source* and *target* tasks are identical but the *source* and *target* domains are different. In this scenario, a lot of labeled data exists in the source domain while absolutely no labeled data is available for the *target* domain. It can be further sub-categorized into two different cases, based on the situation between *source* and *target* domains. In the first case, the feature spaces for the *source* and the *target* domains are different. In the second case, the feature spaces between *source* and *target* domains are the same but the marginal probability of the distributions of the input data are different.

Finally, the third category of *transfer learning* is known as *unsupervised transfer learning*, where the *target* task is related to the *source* task but different. No labeled exists in both the *source* and the *target* domains. The *unsupervised transfer learning* tries to solve unsupervised learning tasks in the *target* domain like clustering, dimensionality reduction and others.

Since deep belief network architectures require large amounts of data for training, we can gain advantage by using *transfer learning.* Layer-wise model adaptation is by far the most popular representational transfer method in deep learning architectures [86]. In this method, a model of the source task is built first. The representation obtained is then used to re-train the model for the target task layer by layer. It enables the method to learn good mid-level representations from the *source* task to improve the learning of the *target* task. Sometimes the representation built from the *source* may cause a negative impact on the learning of the *target* task. This is known as *negative transfer* which needs to be avoided and more research is being conducted in this direction.

In general, three reasons can be cited as to why *transfer learning* can benefit from deep learning architectures. 1) *Shared Internal Representation*: Deep learning can learn shared internal representation in an unsupervised fashion from the examples of a number of different tasks, which enables the learner to generate useful features of the task domain and the context of the problem [88]. 2) *Hierarchical levels of representation*: Deep learning builds hierarchical levels of representation where each layer generates features from the representation in the layer below it. Even for two different but related tasks, it is very probable that these tasks can share some lower levels of representations. This suggests that if we fix the lower levels of representation and re-train the higher levels of representation, we would be able to improve learning with relatively small training data [89]. 3) *Learning from unlabeled data*: Since the layer-wise training in deep learning architectures is unsupervised, it enables us to leverage a small number of labeled data with a large amount of unlabeled data [87].

## 3.2 SPEECH CORPORA

This section describes the three different types of speech corpora utilized for feature extraction. The *Stress Under Simulated and Actual Stress Database* (SUSAS) speech corpus is used for emotion recognition and the *Texas Instruments* and *Massachusetts Institute of Technology* (TIMIT) and PTSD speech databases are utilized for performing PTSD diagnosis. These are described in detail as follows.

### 3.2.1 SPEECH UNDER SIMULATED AND ACTUAL STRESS DATABASE (SUSAS) SPEECH CORPUS

We evaluated the proposed method on the SUSAS database [61]. This database consists of a total of 32 speakers including 13 female and 19 male subjects with ages ranging from 22 to 76. The subjects were recruited to generate over 16,000 utterances from a 35 word aircraft communication vocabulary set. Later in 1993, utterances from four additional male pilots operating ‘apache’ helicopters were added to the database [61]. Speech recordings from two of the pilots were from the 35 word vocabulary, and those from another two pilots consisted of continuous tactical communication (other than the 35 word vocabulary) between the pilots and an air-control operator during an actual night flying mission with the helicopter low on fuel, creating a real stress on the pilots. The SUSAS database comprises five different domains depending upon whether the stress was simulated or was generated under actual stress conditions. The ‘simulated’ domain consists of speech recordings from nine speakers in a quiet environment simulating speech under stress. The ‘actual’ domain uses recordings from seven speakers in states of actual roller-coaster stress [3].

These five domains include: (i) Speaking styles consists of speakers simulating different speech styles such as ‘slow’, ‘fast’, ‘soft’, ‘loud’, ‘angry’, ‘clear’, ‘neutral’ and ‘questioning’ styles of speech.

(ii) Single-tracking task contains speech recordings of subjects undergoing computer workload stress or the ‘lombard’ effect. The stress condition referred to as Lombard effect results when a speaker attempts to modify his or her speech production system while speaking in a noisy environment [62]. Computer workload stress was simulated by displaying an error command to a subject and the subject tried to correct the error while producing speech utterances from the 35 word randomized vocabulary set. The ambient noise was created by generating pink noise and presenting it binaurally to simulate the ‘lombard’ effect.

(iii) A dual tracking task was developed by USAF School Of Aerospace Medicine to simulate actual stress when subjects performing both compensation and acquisition tasks [63]. The primary task was compensatory where subjects had to perform simulated flight control and the secondary task was that of target acquisition.

(iv) The subject-motion fear task consists of speech recordings from subjects riding two types of roller coaster in an amusement park. This task was designed to simulate sudden changes in altitude and direction sometimes experienced in an aircraft cockpit.

(v) Psychiatric analysis includes speech recordings from patient interviews with different emotions at Emory Medical University. In this work, four different styles of speech, (‘angry’, ’loud’, ’lombard’ and ’neutral’) were selected from the first two domains to represent ‘simulated’ stress. Two additional styles of speech, (‘roller coaster stress’ and ‘actual neutral’), were selected from the third and fourth domains to represent ‘actual’ stress.

### 3.2.2 TIMIT SPEECH CORPUS

TIMIT is an acoustic-phonetic speech corpus. It was jointly developed by *Texas Instruments* (TI), *SRI International* (SRI) and *Massachusetts Institute Of Technology* (MIT) to provide speech data for the acquisition of acoustic-phonetic knowledge and for the development and evaluation of automatic speech recognition systems. The objective of using the TIMIT speech corpus is discussed here. A major limitation in training the DBN on the PTSD feature data set is the small size of the feature data set due to which the network is prone to being *overtrained.* Exploiting the large size of the TIMIT speech corpus, we trained the DBN for phone recognition so that we could eventually utilize the trained network to run transfer learning on PTSD data sets, overcoming the small data problem related to PTSD.

The TIMIT corpus includes 16bit, 16kHz, speech waveform data from 630 speakers representing 8 major dialect divisions of american english, each speaking 10 phonetically rich sentences resulting in 6,300 utterances. It also includes time-aligned, orthographic, phonetic and word transcriptions [78]. The speech was recorded at TI, transcribed at MIT and the data has been verified and prepared for cd-rom production by *National Institute of Standards and Technology* (NIST). 70% of the speakers are male and 30% are female. Training and testing subsets, balanced for phonetic and dialectal coverage, are specified.

A "core" test set contains speech data from 24 speakers, 2 male and 1 female from each dialect region and a "complete" test set contains utterances spoken by 168 speakers resulting in about 27% of the total speech material in the corpus. The speech was directly digitized at a sampling rate of 20kHz using a *Digital Sound Corporation (DSC) 200* with the anti-aliasing filter at 10KHz. The speech was then digitally filtered, debiased and downsampled to 16Khz.

### 3.2.3 PTSD SPEECH CORPUS

There are a total of 26 PTSD patient audio data files (speech signals) and audio data from 26 control subjects collected from Youtube and an Ohio hospital. Each set of recordings from each source contains recordings from PTSD and non-PTSD patients split equally. The duration of the recordings from the subjects varies approximately between 51 seconds and 480 seconds while a large fraction of the recordings are between 120-140 seconds. Each recording is from a particular subject and the recordings were sampled at 44.1kHz.

## 3.3 FEATURE EXTRACTION

This section begins with a description of the process of feature extraction for emotional state recognition from the SUSAS speech database. Following this description it provides details of the feature extraction process from TIMIT and PTSD speech databases for utilizing them in PTSD detection.

### 3.3.1 FEATURE EXTRACTION FROM SUSAS FOR EMOTIONAL STATE RECOGNITION

*Speech Under Simulated and Actual Stress Database* (SUSAS) was the first comprehensive speech database to be recorded under stressful conditions and is the database of choice for stressed emotion recognition. As a pre-processing step, the speech signal was passed through a pre-emphasis filter. Following this, speech frames of a length of 25 milliseconds were extracted first from an entire speech signal consisting of a word. The voiced frames present in the speech signal were segmented by short-term energy thresholding. A frame shift of 10 milliseconds was applied. Excitation, vocal tract and prosodic features were then computed for each of these frames. The process is shown in figure 4.

Additionally, time derivative features were also computed, generating a total of 162 features for each speech frame. Finally, we computed the mean, standard deviation and skewness of each feature based on all speech frames of a single word, resulting in 486 features for a word. Feature details are summarized in Table 3.

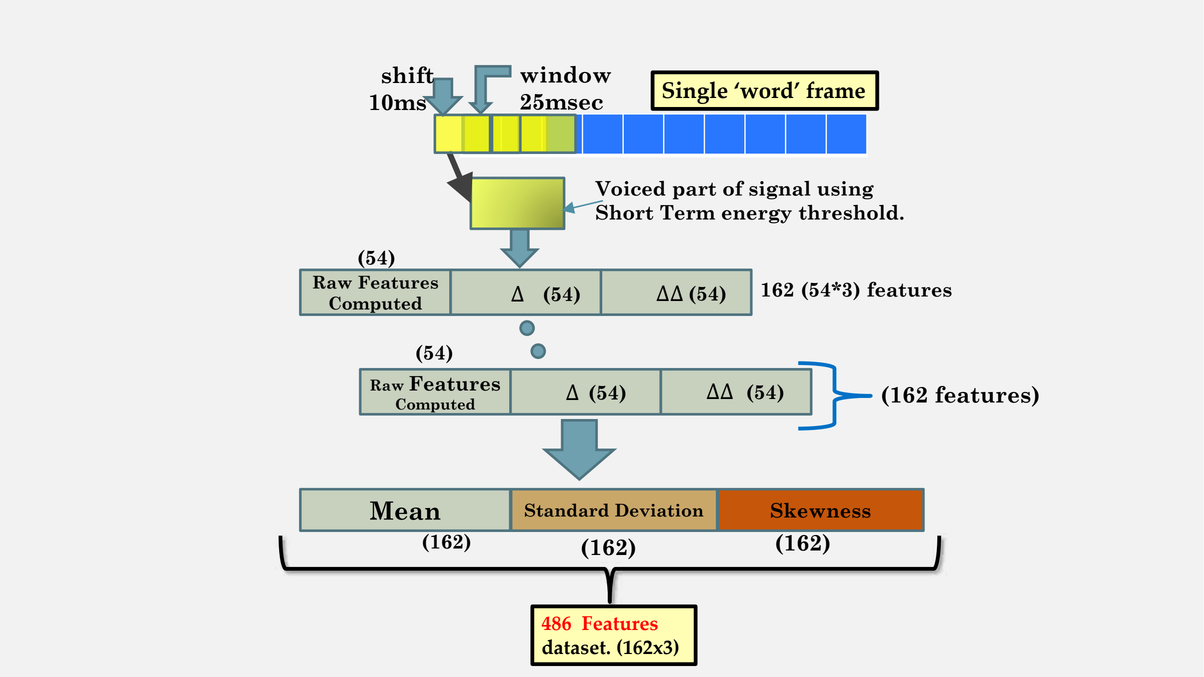

**Figure 4:** The process of feature extraction from SUSAS and TIMIT speech databases.

The types of features used are described in the following sections.

### 3.3.1.1 PROSODIC FEATURES

**3.3.1.1.1 SHORT TERM ENERGY:** Short-term energy is the energy of a short speech segment. For the *n-th* speech frame, $x_n$ of length $N$,

$$x_n(m) = x(m)w(n-m) \tag{8}$$

where $w(n-m)$ is a windowing function such as the Hamming window and $N$ is the window length, $x(m)$ is the $m-th$ sample in the whole speech signal. The short-term energy of $x_n$ is computed as,

$$E_n = \sum_{m=n-N+1}^{m=n} [x_n(m)]^2 \tag{9}$$

The short-term energy feature extracted from the SUSAS speech corpus for a speech utterance

**Table 3:** Description of speech frame features. The same raw features have been extracted across all the three different speech corpora.

| **Feature** | **Position** | **Number Of Features** |
|---|---|---|
| **Prosodic features** | | |
| Short-time energy | 1 | 1 |
| Average power | 2 | 1 |
| Average magnitude | 3 | 1 |
| No of Zero crossings | 4 | 1 |
| Mean | 5 | 1 |
| Median | 6 | 1 |
| Standard deviation | 7 | 1 |
| Minimum | 8 | 1 |
| Maximum | 9 | 1 |
| Range | 10 | 1 |
| Dynamic range | 11 | 1 |
| Interquartile range | 12 | 1 |
| **Vocal-tract features** | | |
| MFCC (Mel Frequency Cepstrum Coefficients) | 13-51 | 39 |
| Teager Energy Operator | 52 | 1 |
| **Excitation features** | | |
| Jitter | 53 | 1 |
| Shimmer | 54 | 1 |
| Total number of Original features | 1-54 | 54 |
| 1st order time derivative features | 55-108 | 54 |
| 2nd order time derivative features | 109-162 | 54 |
| Total number of features per frame | **162** | |

of the word *freeze* is shown in figure 5.

**3.3.1.1.2 AVERAGE POWER:** The average power of a short speech segment is the short-term energy divided by the number of speech samples in that segment and it is computed as,

$$P_n = \frac{1}{N} \sum_{m=n-N+1}^{m=n} [x_n(m)]^2 \tag{10}$$

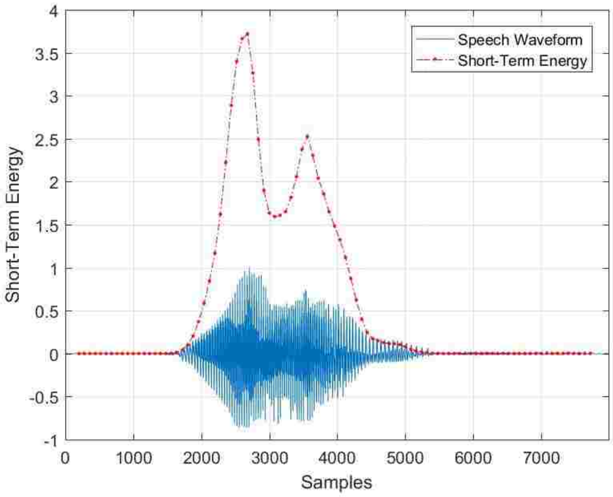

**Figure 5:** Short-term energy feature extracted from SUSAS for speech utterance of the word *freeze*.

**3.3.1.1.3 AVERAGE MAGNITUDE:** This measure does not emphasize larger signal amplitudes like the short-time energy measure since it eliminates the squaring. It is defined as,

$$M_n = \frac{1}{N} \sum_{m=n-N+1}^{m=n} [x_n(m)] \tag{11}$$

The average magnitude feature extracted from the SUSAS speech corpus for a speech utterance of the word *freeze* is shown in figure 6.

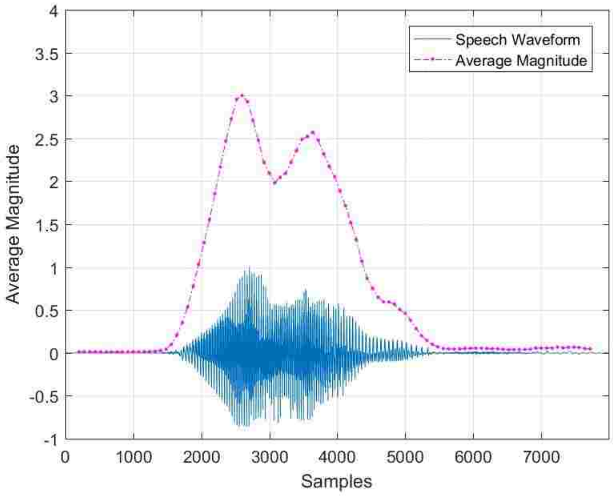

**Figure 6:** Average magnitude feature extracted from SUSAS for speech utterance of the word *freeze.*

**3.3.1.1.4 ZERO CROSSINGS:** A zero-crossing is said to occur if successive samples have different algebraic signs. The number of zero crossings is a simple measure of the frequency content of a signal and defined as,

$$Z_n = \frac{1}{2N} \sum_{m=n-N+1}^{m=n-1} |sgn[x_n(m+1)] - sgn[x_n(m)]| \tag{12}$$

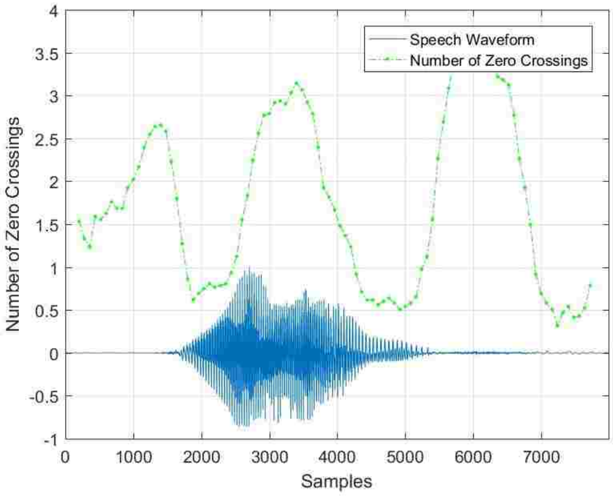

**Figure 7:** Number of zero-crossings feature extracted from SUSAS for speech utterance of the word *freeze.* Feature extraction is carried out only on the voiced segment identified using short-term energy thresholding.

The number of zero-crossings features extracted from the SUSAS speech corpus for a speech utterance of the word *freeze* is shown in figure 7. Feature extraction is carried out only on the voiced segment identified using short-term energy thresholding.

**3.3.1.1.5 DYNAMIC RANGE:** Dynamic range is computed as the difference in base-10 logarithm of the maximum and the minimum amplitudes given by,

$$DynamicRange = Log_{10}(\alpha_{max}) - Log_{10}(\alpha_{min}) \tag{13}$$

where $\alpha_{max}$ and $\alpha_{min}$ are the maximum and minimum amplitudes of the speech signal frame.

**3.3.1.1.6 INTERQUARTILE RANGE:** The interquartile range is expressed as the difference between the 75th and 25th percentile, and is given by,

$$IQR = P_{75} - P_{25} \tag{14}$$

where $P_{75}$ and $P_{25}$ are the $75^{th}$ and $25^{th}$ percentile respectively and IQR denotes the interquartile range.

#### 3.3.1.2 VOCAL-TRACT FEATURES

**3.3.1.2.1 MEL FREQUENCY CEPSTRAL COEFFICIENTS:** MFCC is computed based on frequency bins on Mel-scale [49], which is a frequency binning method based on the human ear's frequency resolution. The mel-scale mimics the human ear in terms of the way in which frequencies are sensed and resolved. The general procedure of extracting the MFCC features involves several steps. First, a pre-emphasis filter is applied to boost high frequencies. Second, frequency spectrum is obtained using the fast fourier transform (FFT). Third, the spectrum is passed through Mel-filters to obtain the Mel Spectrum and finally, cepstral analysis is performed on the Mel-Spectrum to obtain MFCC features. The spectrum $\boldsymbol{X}_n$ for speech frame $\boldsymbol{x}_n$ is computed as,

$$\mathbf{X}_n(k) = X_n(k), \qquad k = 1, 2...K \tag{15}$$

$$X_n(k) = \frac{1}{N}\sum_{m=1}^{m=N}(x_n(m) * h_p(m))e^{-j2\pi km/N} \tag{16}$$

where $h_p(m) = \delta(m) - \lambda\delta(m)$is the impulse response of the two-tap pre-emphasis filter, "$*$" denotes convolution and $N$ is the length of the discrete Fourier transform (DFT). The periodogram-based power spectral estimate for the speech frame $\mathbf{x}_n$ is given by,

$$P_n(k) = \frac{1}{N}|X_n(k)]|^2 \tag{17}$$

The power spectral estimate is then converted to mel scale by triangular overlapping windows of the mel filterbank [50] that gives a measure called the log-spectral energy envelope $\boldsymbol{E}$, given by,

$$\boldsymbol{E} = E(j), \qquad j = 1, 2...M \tag{18}$$

at the output of each filter and is defined as,

$$E(j) = \frac{1}{N}\sum_{k=1}^{k=K} ln[|(P_n(k)||H_j(k)|], \qquad j = 1, 2...M \tag{19}$$

where $H_j(k)$ is the transfer function of the *j-th* filter in the mel filterbank and $M$ denotes the number of filterbank channels in the filterbank. Finally, MFCC is obtained by taking the discrete cosine transform (DCT) of the mel log powers,

$$C_p = \sqrt{\frac{2}{M}}\sum_{j=1}^{M} E(j)\cos\frac{\pi p}{M}(m - 0.5) \tag{20}$$

where $p$ is the number of computed MFCC features.

The speech frame signal is transformed using a fast fourier transform (FFT) algorithm and the resulting spectrum is converted to logarithmic scale. The logarithmic scale is then transformed to the resulting cepstrum after taking the inverse discrete fourier transform (IDFT). The cepstral coefficients, $\mathbf{c}_n$ can be computed using the following relation,

$$\mathbf{c}_n = \mathit{Real}[IDFT(ln|FFT(\mathbf{x}_n)^2|)] \tag{21}$$

The value of the fundamental frequency $F_0$ can be evaluated by using the following relation,

$$F_0 = \frac{f_{samp}}{t} \tag{22}$$

where $f_{samp}$ is the sampling frequency and $t$ is the order of the given cepstral coefficient corresponding to the local maximum (peak) of the cepstrum.

**3.3.1.2.2 TEAGER ENERGY OPERATOR:** Teager showed that airflow separates in the vocal-tract when it propagates, instead of just flowing as a plane wave. During stress, a change occurs in the vocal system physiology during speech production which is further seen to affect the vortex-flow interactions in the vocal tract [51]. This feature has been shown to successfully detect these changes in speech production. This feature has been found to be responsive to speech under stress using audio from the SUSAS corpus [52]. The Teager Energy operator, is a non-linear energy tracking operator, $\Psi[.]$ and is computed using the following relation,

$$\Psi[x_n(m)] = x_n(m)^2 - x_n(m+1)x_n(m-1) \tag{23}$$

The teager energy operator feature extracted from the SUSAS speech corpus for a speech utterance of the word *freeze* is shown in figure 8.

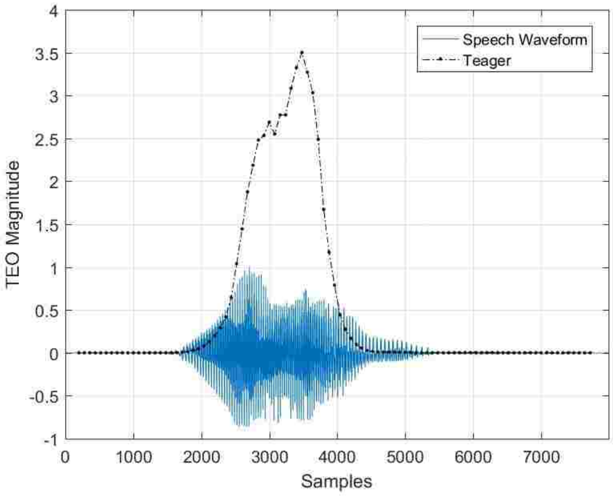

**Figure 8:** Teager energy operator feature extracted from SUSAS for speech utterance of the word *freeze*.

#### 3.3.1.3 EXCITATION FEATURES

**3.3.1.3.1 JITTER:** Jitter is defined as average absolute difference of consecutive pitch periods. It is a measure of the variation of successive pitch periods. It is defined by,

$$jitter = \frac{1}{Q-1}\sum_{i=1}^{Q-1} |T_i - T_{i-1}| \tag{24}$$

where $T_i$ is the $i-th$ extracted pitch period and $Q$ is the total number of extracted pitch periods in the segment.

**3.3.1.3.2 SHIMMER:** Shimmer is the average absolute difference (in dB) between amplitudes of consecutive periods,

$$shimmer = \frac{1}{L-1}\sum_{i=1}^{L-1} |20log(\frac{U_{i+1}}{U_i})| \tag{25}$$

where $U_i$ is the amplitude of the $i-th$ period and $L$ is the total number of extracted periods.

**3.3.1.3.3 TIME DERIVATIVE FEATURES:** The first and second order time derivative features were computed as,

$$\Delta f(n) = f(n+1) - f(n-1) \tag{26}$$

$$\Delta^2 f(n) = \Delta f(n+1) - \Delta f(n-1) \tag{27}$$

where $f(n)$ denotes a feature computed from the $n-th$ speech frame. Once we determined the above features for each frame, we computed the mean, median, max, min, range and skewness for each feature based on all speech frames of the word. In total, 486 features were computed for each word.

### 3.3.2 FEATURE EXTRACTION FROM TIMIT

TIMIT is an acoustic-phonetic speech corpus. It was jointly developed by *Texas Instruments* (TI), *SRI International* (SRI) and *Massachusetts Institute Of Technology* (MIT) to provide speech data for the acquisition of acoustic-phonetic knowledge and for the development and evaluation of automatic speech recognition systems. The process of single-frame feature extraction is as described here and shown in figure 4. The process of multiple-frame feature extraction is shown in figure 10. The speech signal was first pre-emphasized using a first order FIR filter. Then the speech signal was divided into a set of frames of length 25ms with an overlap of 10ms between two consecutive frames. Speech frames overlapping two different phones were deleted. Features identical to those shown in Table 3 are extracted from the TIMIT database. They comprise of a combination of prosodic, vocal-tract and excitation features typically used in speech recognition. A total of 54 raw features along with their first and second order temporal derivatives are combined to form a total feature vector with a length of 162 features. There are 39 phone classes in this dataset. Training data is separate from testing. The output phone classification is carried out using the logistic regression classifier.

A second data set was also built for TIMIT derived form the first data set. This process is shown in figure 10. Using the feature vector having 162 features obtained from the data set described in the previous paragraph, 15 contiguous feature vector segments, are concatenated column-wise to form a 2,430 (162x15) dimensional feature vector. Each 15-segment feature vector is extracted using a constant time shift of one frame. The phone label of the central frame was considered to be the resulting phone label of this feature vector. There are 39 phone classes in this dataset. Training data is separate from testing. There are a total number of almost 439,000 data points for training, and approximately 161,000 samples for testing. The output phone classification is carried out using the logistic regression classifier.

### 3.3.3 FEATURE EXTRACTION FROM PTSD SPEECH CORPUS

There are a total of 26 PTSD patient audio data files (speech signals) and audio data from 26 control subjects collected from youtube and an Ohio hospital. The duration of the recordings from the subjects varies approximately between 51 seconds and 480 seconds while a large fraction of the recordings are between 120-140 seconds. Each recording is from a particular subject. Frame sizes of 1, 2 and 3 seconds and forward frame shifts of 0.1, 0.5 and 1 seconds are used to extract word-level segments from the entire audio recording of the given duration. Given a subject audio recording, we select a specific combination of frame size and shift to obtain multiple word-level segments for that recording. The process of single-frame feature extraction process is shown in figure 9.

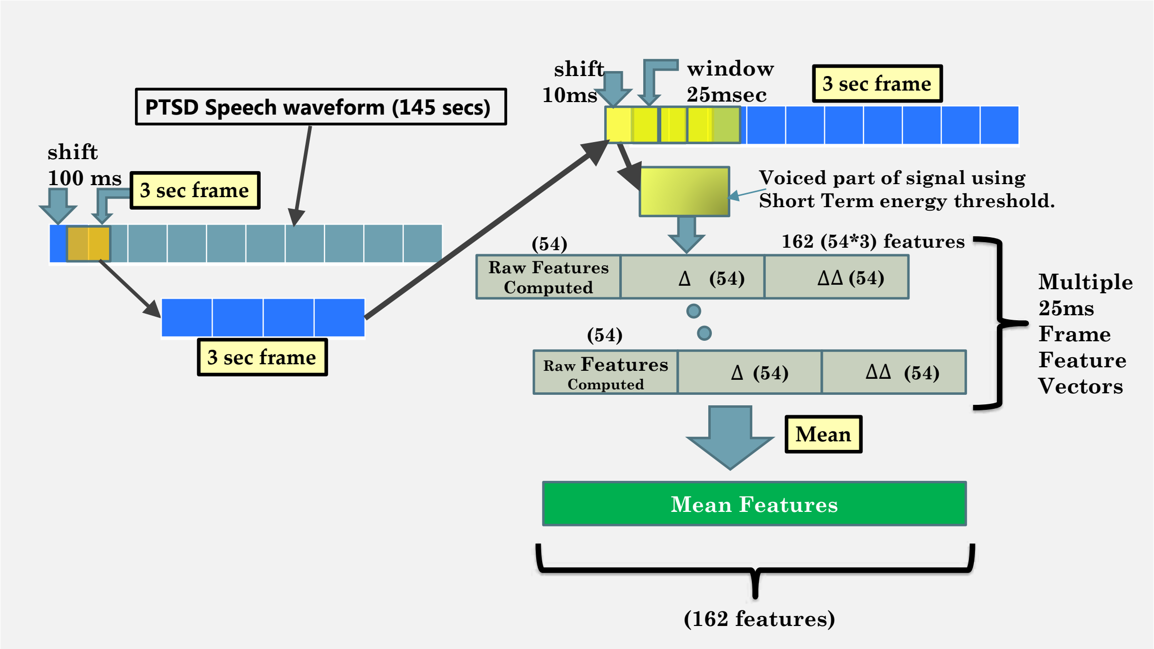

**Figure 9:** The process of single frame feature extraction having 162 features from the PTSD speech database.

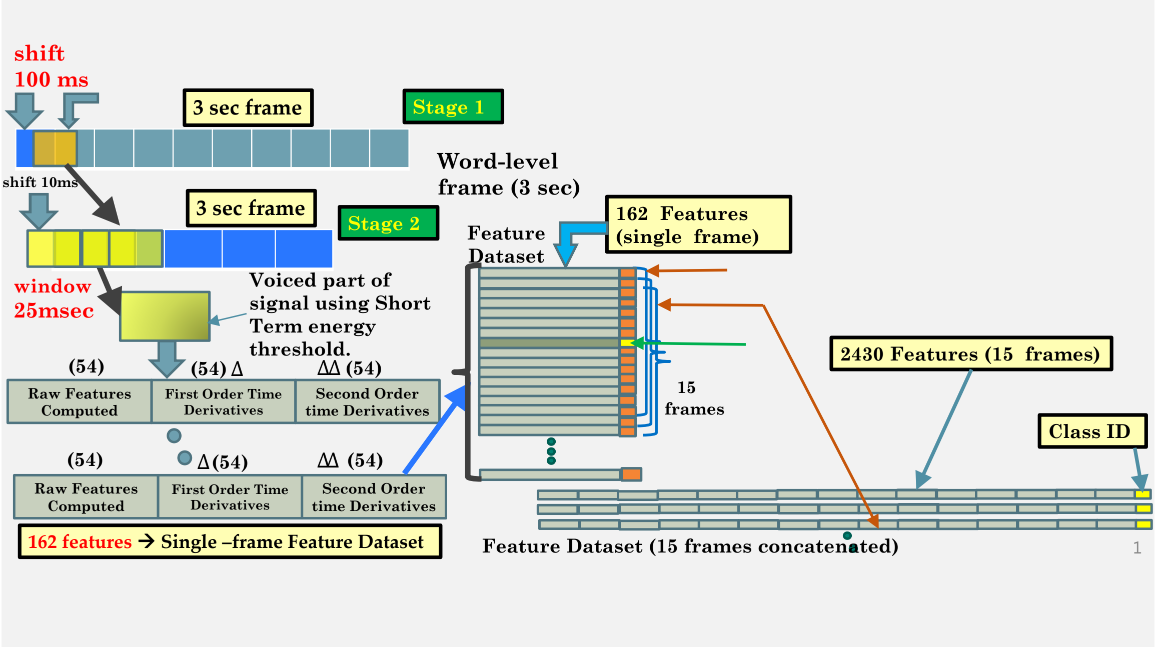

**Figure 10:** The process of concatenation of 15 frames, each having 162 features to form the multi-frame TIMIT and PTSD datasets. For PTSD the process begins from stage 1 shown in the figure whereas for TIMIT, it begins from stage 2.

Subsequently, for each word-level segment, we apply a 25ms frame-size and 10ms frame-shift to compute the previously mentioned raw features and the first and second order time derivatives for each frame to give us 162 (54 raw features + 54 first-order time derivatives + 54 second-order time-derivative) features. In the process we obtain multiple frames. The types of features computed are the same as those mentioned in Table 3. After the features have been extracted from all the frames of this word segment, we take the mean of all these frames. For each word-level segment we obtain a single frame. This process is then repeated for all of the word-level segments for that given audio file which gives us as many frames as the number of word-level segments in the recording. This process is repeated for variable word-level frame sizes and shifts. This forms the single-frame dataset.

We built a second multiple frame data set using the above feature vector, 162 features in length as a starting point, by concatenating 15 frames. This is shown in figure 10. A total of 15 contiguous frames were concatenated to form a 2,430 (162x15) dimensional feature vector. The class label of the central frame was appended to the end of this newly created feature vector. To build the subsequent feature vector, we applied a shift of one frame and repeated the process. Each subject has a matrix of features with each row representing a feature vector containing 15 speech frames. For a given recording, the class-label was from the same subject so, neither the class-label nor the subject-label changed. The PTSD features were extracted for a total of 9 different datasets of frame-length and frame-shift combinations as shown in Table 3. One of these datasets corresponding to a 3 sec frame-length and 1 sec frame-shift was used for feature selection experiments described in section 3.4. Hypothesis testing was performed to identify if the differences were significant. Detailed results are described in the 6.3.2 section.

We built a third data set using MFCC features extracted from PTSD audio recordings. The initial process of extracting a word-level segment is the same as described in the first paragraph of this section. In this context, a word-level segment is a long-time speech segment typically 1, 2 or 3 seconds in duration. For each word-level segment, a short-time, 25ms frame-size and 10ms frame-shift was then applied to compute the *MFCC* features and the first and second order temporal derivatives generating 39 (13 raw *MFCC* features + 13 first-order time derivatives + 13 second-order time-derivative) features. This feature computation was repeated for all the short-time frames present in this long-time speech window. After all the short-time speech frames of this word-level segment were processed with each frame having 39 dimensions, we concatenated 15 short-time frames without any overlap and continued this process across the entire word-level segment. This generated multiple 15-frame segments with each frame having 585 dimensions (39x15). These 15-frame segments were then averaged to get only one feature vector or sample for each word-level segment. This resulted in a total number of feature vector samples equal to the number of word-level segments in a recording with each sample having 585 features. This process was then repeated for all the word-level speech segments using variable word-level window sizes and shifts, for a given audio file. This process, when repeated for all audio files across all subjects, gave us the required dataset. For a given recording, the class-label was from the same subject. Neither the class label nor the subject label changed. The PTSD features were extracted for a total of 9 different sets of long-time window length and window shift combinations. *Transfer learning* was then applied to the network similar to that described in the previous section. All the layers were utilized in evaluating classification performance. Results are shown in Tables 48 through 52.

## 3.4 FEATURE SELECTION IN TRANSFER LEARNING

We investigated which of the three categories of features or feature combinations was the most effective for PTSD detection. We used all the three categories of features as inputs to a deep belief model that fused the features for PTSD diagnosis. The process of feature selection in *transfer learning* is shown in figure 11.

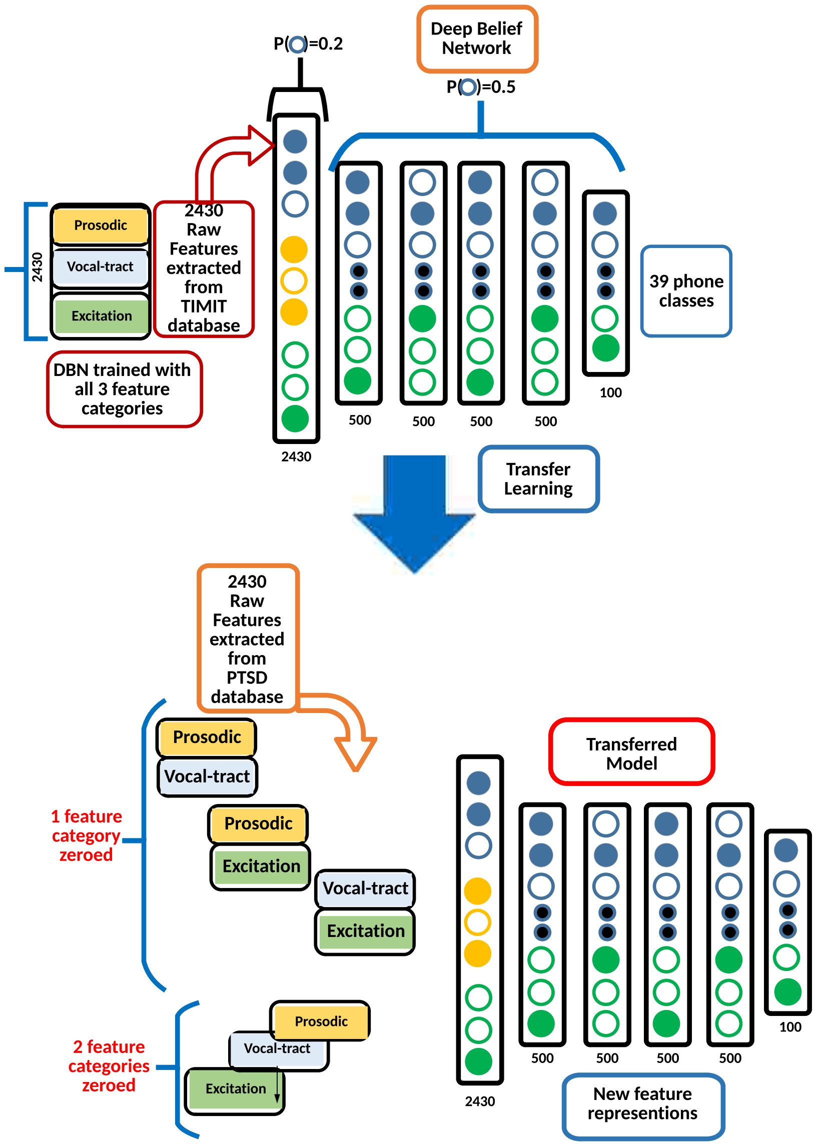

**Figure 11:** Feature category selection in *transfer learning*.

The input PTSD speech features dataset described earlier in section 3.3.3 was used for feature selection experiments. The model used for the feature selection experiments was 2430-500-500-500-500-100.

First, we trained the 5-layer deep model on TIMIT and then we used the trained model as a feature extractor to obtain new representations for PTSD features in each of the 5 hidden layers. The modified inputs for PTSD diagnosis were then transferred by the deep structure. We then evaluated the performance of each layer representation on both Youtube and Ohio datasets by utilizing the LOSO-CV. In LOSO-CV, we left one subject for testing and remaining data were used to train a SVM classifier. This process was repeated until each subject was tested once, and then training and testing accuracies were computed. We have used two performance metrics: segment-wise accuracy and subject-wise accuracy. If a subject's segment-wise accuracy surpassed 50%, the subject was considered to be correctly classified.

Secondly, one feature category was excluded each time by zeroing those features in the PTSD data set at the input layer in the deep model. LOSO-CV was then applied to each the new feature representations obtained from all the hidden layers. Lastly, all these steps were again performed excluding two feature categories each time. The first hidden layer showed the best overall performance in classification so the results presented in section 6.2.3 were obtained from the first hidden layer only. All possible combinations of the three feature categories were investigated, with the aim of identifying the best input feature combination for PTSD diagnosis.

## 3.5 PRINCIPAL COMPONENT ANALYSIS

We performed *principal component analysis* (PCA) to reduce the dimensionality of the features. PCA finds $d$ orthogonal vectors that encompass the most variance in the data. Consider $F$ as a $mxn$ data matrix containing the $m$ samples in $n$ dimensions and $V$ as a $mxd$ mapping matrix ( $d < n$ to achieve dimensionality reduction) that maximizes $V^T cov(F)V$. Finding $V$ can be done by solving the eigen-decomposition problem shown below,

$$cov(F)V = \gamma V \tag{28}$$

The resultant $V$ contains $d$ orthogonal basis vectors spanning the data. These vectors are known as principal components that correspond to eigenvalues of $\gamma_1 \geq ... \geq \gamma_d$, with the first principal component corresponds to the largest eigenvalue, the second principal component corresponds to the second largest eigenvalue etc.. To achieve dimensionality reduction, the mapping matrix $V$ was applied on $F$ as,

$$A = VF \tag{29}$$

where A has a dimensionality $d$ that is less than $n$. To compare with the proposed method, we performed PCA using the 585 features as input for dimensionality reduction. We kept those leading principal components (PCs), so that they can account for 99.9999% of the variance in the data. The selected PCs were then input to a linear SVM for classification.

## 3.6 CLASSIFICATION

In this dissertation, we used support vector machine (SVM), for classification. The SVM classifier usually classifies data into two classes by finding the optimal hyperplane that separates data points of one class from those of another class [60]. The optimal hyperplane has the largest or maximal margin between the two classes. If the data is linearly separable, a linear SVM may be used whereas for non-linearly separable data, non-linear SVM's are applied using different kernel types. Considering a two-class linearly separable data set $\zeta$, a decision boundary can be found such that all data points will satisfy the constraint,

$$k_j(\omega^T\phi(\zeta_j)+b) \geq 1, \qquad j=1,...,N_v \tag{30}$$

where $N_v$ is the number of input vectors, $k_j \in \{-1,1\}$ , denotes the target classification, $\omega$ is a normal vector to the hyperplane, $\phi(\zeta_j)$ denotes a fixed feature-space transformation and $b$ is the bias parameter. Maximizing the margin to the hyperplane is equivalent to maximizing $||\omega||^{-1}$ or equivalently minimizing $||\omega||^2$. Then the optimization problem becomes,

$$arg\ min_{\omega,b}\frac{1}{2}||\omega||^2 \tag{31}$$

under the constraints [62]. In real datasets where the class distributions are overlapping a relaxation term must be included. A slack variable $\xi_j \geq 0$ is introduced for each data point and is defined as,

$$\xi_j = |k_j - y(\zeta_j)| \tag{32}$$

where $y(\zeta_j)$ is the predicted classification by the SVM. The slack variable will be zero if the point is inside the correct margin boundary and positive otherwise. The constraint is modified to:

$$k_j y(\zeta_j) \geq 1-\xi_j, \qquad j=1,..,N_v \tag{33}$$

The minimization of the model therefore now becomes:

$$C\sum_{j=1}^{N_v}\xi_j + \frac{1}{2}||\omega||^2 \tag{34}$$

where $C \geq 0$ is an adjustable parameter regulating the trade-off between the margin and the slack variable.

# CHAPTER 4

# EMOTIONAL STATE RECOGNITION BASED ON SUSAS SPEECH CORPUS

This chapter discusses the results of emotional state recognition based on the SUSAS speech corpus. They are described in the experiments and results section in which the experimental protocols and results are discussed in detail. *PCA* was performed to compare emotional state recognition performance and is discussed in this section. *Sparse coding* was also carried out in three different evaluation contexts and its details are discussed in this section. The results achieved by the *deep belief network* model are also discussed. Hypothesis testing results showed that *sparse coding* did not achieve significantly better results than the baseline for the text-dependent pairwise scenario. *Sparse coding* achieved better performance for the text-independent pairwise and the text-independent multistyle scenarios.

We conducted emotion state recognition for three different scenarios including (i) Text-dependent pairwise stress classification, (ii) Text-independent pairwise stress classification and (iii) Text-independent multistyle stress classification. The proposed system was compared with other baseline methods.

## 4.1 EXPERIMENTS AND RESULTS

### 4.1.1 PRINCIPAL COMPONENTS ANALYSIS RESULTS

The principal components analysis was only performed for the SUSAS data set. For the text-dependent scenario, 17 PC's were selected for the simulated domain whereas for the actual domain, 15 PCs were chosen. For the text-independent scenario 107 PCs were selected for the simulated domain whereas for the actual domain, 95 PCs were chosen. For the text-independent multistyle scenario, 40 PCs were selected for the simulated domain.

### 4.1.2 SPARSE CODING MODEL FOR EMOTION RECOGNITION

#### 4.1.2.1 TEXT DEPENDENT PAIRWISE EMOTION RECOGNITION

In the first experiment, we conducted text-dependent pairwise stress classification with different variations. First, we chose the same subset of six vocabulary words - ‘freeze’, ‘mark’, ‘nav’, ‘help’, ‘oh’ and ‘zero’ as those used in [3]. In the simulated domain, each word was spoken by nine speakers with the four different stress styles and each style was repeated once, resulting in a total of 108 (6x9x2) recordings for each emotion. In the actual domain, each word was spoken by seven speakers and was repeated a variable number of times and sometimes some words were completely omitted. This resulted in a total of 94 recordings for the actual stress style and 94 recordings for the actual neutral style. We conducted four classification tasks including ‘angry’, ‘loud’ and ‘lombard’ versus ‘simulated neutral’, respectively, and ‘roller coaster stress’ versus ‘actual neutral’. For each word, we applied the leave-one-out-cross-validation (LOOCV) scheme to evaluate the proposed method.

In LOOCV, one recording of a word was left for testing and the remaining data were used to train a classification model. This procedure was repeated till each recording was tested once. In the experiment, a linear SVM was used for classification, whose parameters were optimized by grid search. The same procedure was repeated for all the six words thereby generating a total of six test accuracies for both simulated and actual domains. Mean accuracy and standard deviation for each of the classification tasks were then calculated. To test if more training data will improve the classification, we performed the second variation of the experiment by increasing the data set to include 22 words from the SUSAS database. These words are ‘freeze’, ‘mark’, ‘nav’, ‘help’, ‘oh’, ‘zero’, ’steer’, ’strafe’, ’ten’, ’thirty’, ’three’, ’white’, ’wide’, ’enter’, ’fifty’, ’gain’, ’go’, ’hello’, ’hot’, ’point’, ’six’ and ’south’. The data set contains a total of 396 (22x9x2) recordings for each emotion in the simulated domain, and 244 actual stress style recordings and 434 actual neutral style recordings in the actual domain.

For the text-dependent pairwise stress classification case, results of LOOCV are shown in Tables 4, 5 and 6. They are also represented in figures 12, 13 and 14. In the text-dependent experimental scenario where the input consists of only 6 chosen words, the results are discussed. Figure 12 is used to depict the results graphically for this particular scenario and tabulated in Table 4. The best mean accuracy is 90.86% with a standard deviation of 6.13% by the proposed method. The proposed method marginally outperforms the baseline SVM which achieves 90.69%. However, the proposed method fails against the PCA method which achieves 91.83%. The mean and standard deviation were computed across all the four classification tasks.

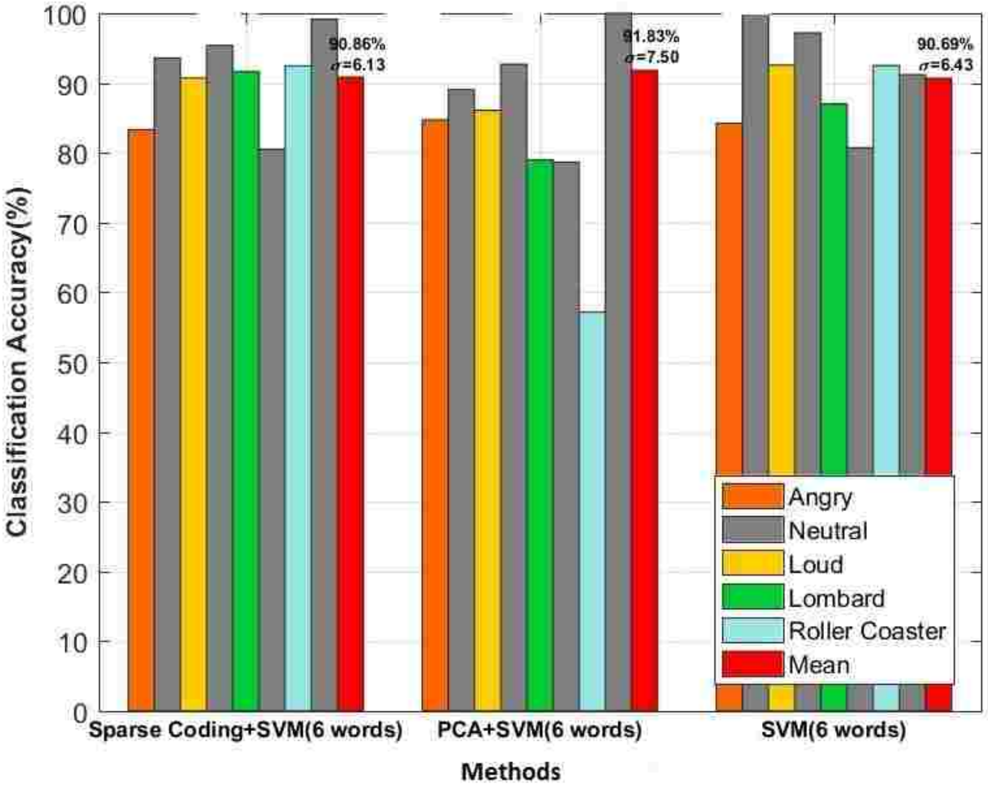

**Figure 12:** LOOCV results for the text-dependent case by different methods. Six words are used to extract all the features which are used as input to the classifier.

**Table 4:** LOOCV results for the text-dependent case by different methods. Six words are used to extract all the features which are used as input to the classifier.

| **Method** | **Experiment Variations** | **Angry + Simulated Neutral (%)** | **Loud + Simulated Neutral (%)** | **Lombard + Simulated Neutral (%)** | **Roller Coaster Stress + Actual Neutral(%)** | **Overall Mean + Std(%)** |
|---|---|---|---|---|---|---|
| All Fea+SC+SVM (Proposed Model) | 6 words with all features | Angry=83.33<br>Neutral=93.60 | Loud=90.73<br>Neutral=95.41 | Lombard=91.66<br>Neutral=80.53 | Roller Coaster Stress=92.48<br>Neutral=99.16 | Mean=90.86<br>($\sigma = 6.13$) |
| PCA+SVM | 6 words with all features | Angry=84.72<br>Neutral=89.08 | Loud=86.10<br>Neutral=92.70 | Lombard=78.98<br>Neutral=78.67 | Roller Coaster Stress=57.16<br>Neutral=100 | Mean=91.83<br>($\sigma = 7.5$) |
| All Fea+SVM | 6 words with all features | Angry=84.25<br>Neutral=100 | Loud=92.58<br>Neutral=97.20 | Lomsbard=87.03<br>Neutral=80.73 | Roller Coaster Stress=92.51<br>Neutral=91.22 | Mean=90.69<br>($\sigma = 6.43$) |

For the text-dependent pairwise stress classification case, results of LOOCV are shown in Tables 4, 5 and 6. They are also represented in figures 12, 13 and 14. In the text-dependent experimental scenario where the input consists of only 6 chosen words, the results are discussed. Figure 12 is used to depict the results graphically for this particular scenario and tabulated in Table 4. The best mean accuracy is 90.86% with a standard deviation of 6.13% by the proposed method. The proposed method marginally outperforms the baseline SVM which achieves 90.69%. However, the proposed method fails against the PCA method which achieves 91.83%. The mean and standard deviation were computed across all the four classification tasks.

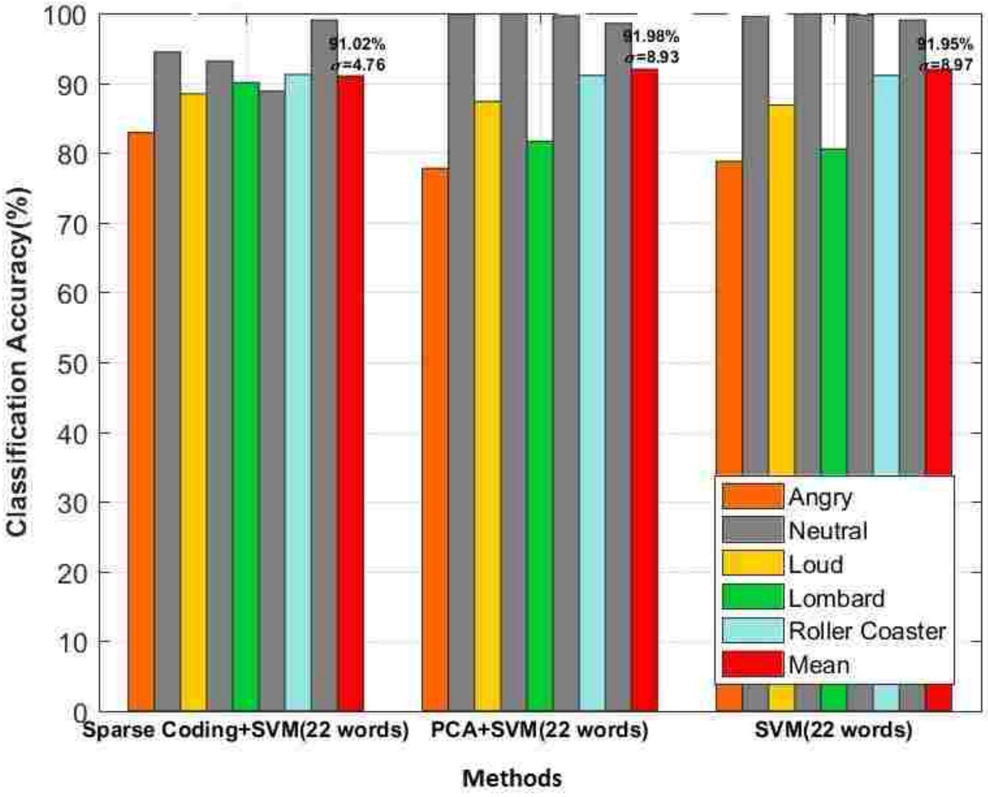

**Figure 13:** LOOCV results for the text-dependent case by different methods. Twenty-two words are used to extract all the features which are used as input to the classifier.

Another experiment was conducted in the text-dependent context, using 22 types of words as input. Each word was paired with a larger number of neutral recordings, the results of which are discussed. Figure 13 is used to depict the results graphically for this particular scenario and tabulated in Table 5. Table 5 shows that the proposed method achieves the best accuracy of 91.02% but fails to outperform the baseline SVM and PCA methods which achieve 91.95% and 91.98% respectively. It is also noted that increasing the training set size improves the classification accuracy of the proposed method by 0.16%.

**Table 5:** LOOCV results for the text-dependent case by different methods. Twenty-two words are used to extract all the features which are used as input to the classifier.

| **Method** | **Experiment Variations** | **Angry + Simulated Neutral (%)** | **Loud + Simulated Neutral (%)** | **Lombard + Simulated Neutral (%)** | **Roller Coaster Stress + Actual Neutral(%)** | **Overall Mean + Std(%)** |
|---|---|---|---|---|---|---|
| Proposed Model | 22 words with all features | Angry=82.90<br>Neutral=94.46 | Loud=88.46<br>Neutral=93.15 | Lombard=90.07<br>Neutral=88.85 | Roller Coaster Stress=91.24<br>Neutral=99.04 | Mean=91.02<br>($\sigma$ = 4.76) |
| PCA+SVM | 22 words with all features | Angry=77.77<br>Neutral=99.87 | Loud=87.36<br>Neutral=99.95 | Lombard=81.64<br>Neutral=99.63 | Roller Coaster Stress=91.10<br>Neutral=98.57 | Mean=91.98<br>($\sigma$ = 8.93) |
| All Fea+SVM | 22 words with all features | Angry=78.78<br>Neutral=99.59 | Loud=86.86<br>Neutral=99.95 | Lomsbard=80.55<br>Neutral=99.78 | Roller Coaster Stress=91.10<br>Neutral=99.04 | Mean=91.95<br>($\sigma$ = 8.97) |

A third scenario of this experiment was carried out in which the same set of words was used as in the second scenario. In the simulated domain, a different and much larger set of recordings of the same words exists only for the 'neutral' emotion style, in which each speaker repeats each word 12 times. This results in a total of 2,376 (22x9x12) recordings. This set of recordings was used to train the simulated 'neutral' style, while a total of 396 (22x9x2) recordings were used for each of 'angry', 'loud' and 'lombard' emotion. The number of recordings used in the actual domain were identical to that used in the second scenario. We also conducted similar variations as those in experiment 1 to test if more training datasets will improve the classification and the discriminating capabilities of different feature categories. This variation was employed only for cases where each feature category was excluded for evaluation. Note that the text-independent experiment has a separate testing data set and those added words were used for training only.

We also tested the discriminating capabilities of different feature sets. In the fourth to sixth scenarios, we excluded the prosodic, vocal-tract and excitation features, resulting in feature vector lengths of 378(126*3), 126(42*3) and 396(132*3) respectively.

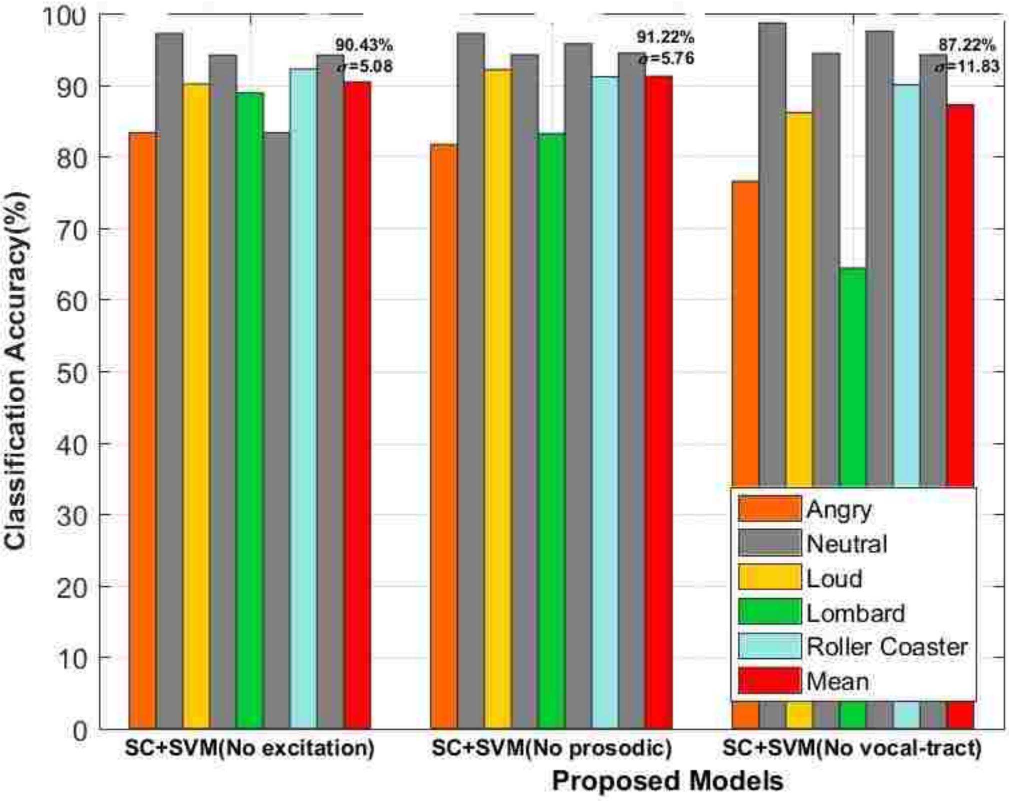

**Figure 14:** LOOCV results for the text-dependent case across different feature combinations.

**Table 6:** LOOCV results for the text-dependent case across different feature combinations.

| **Method** | **Experiment Variations** | **Angry + Simulated Neutral (%)** | **Loud + Simulated Neutral (%)** | **Lombard + Simulated Neutral (%)** | **Roller Coaster Stress + Actual Neutral(%)** | **Overall Mean + Std(%)** |
|---|---|---|---|---|---|---|
| Proposed Model | Excitation features excluded | Angry=83.33<br>Neutral=97.22 | Loud=90.14<br>Neutral=94.18 | Lombard=88.89<br>Neutral=83.33 | Roller Coaster Stress=92.22<br>Neutral=94.18 | Mean=90.43<br>($\sigma = 5.08$) |
| Proposed Model | Prosodic features excluded | Angry=81.67<br>Neutral=97.22 | Loud=92.14<br>Neutral=94.22 | Lombard=83.18<br>Neutral=95.77 | Roller Coaster Stress=91.15<br>Neutral=94.45 | Mean=91.22<br>($\sigma = 5.76$) |
| Proposed Model | Vocal-tract features excluded | Angry=76.51<br>Neutral=98.65 | Loud=86.12<br>Neutral=94.41 | Lombard=64.38<br>Neutral=97.51 | Roller Coaster Stress=90.02<br>Neutral=94.22 | Mean=87.22<br>($\sigma = 11.83$) |
| Average (Tables 5 + 6) | | 89.32 | 92.05 | 85.56 | 92.00 | |

Table 6 and figure 14 show the results for the experimental scenario in which classification is performed by excluding a selected category of features. Since there are three major categories of speech features used in this dissertation, it results in three distinct variations of the training set. Results show the best accuracy is 91.22% with a standard deviation of 5.76% which achieves an improvement of 0.2% over the previous result using 22 words and all features. It, however, fails to significantly outperform the baseline SVM or PCA result in the text-dependent evaluation context.

#### 4.1.2.2 TEXT INDEPENDENT PAIRWISE EMOTION RECOGNITION

The second experiment involved text-independent pairwise stress classification to verify whether and to what extent the classification performance depends on information contained in a text or a phoneme [3]. We conducted the same classification tasks as in experiment 1. However, the data selected for training and testing were from different vocabulary words.

Tables 7 and 8 show the results from the text-independent pairwise stress classification experiment. Figures 15 and 16 show the results in corresponding Tables 7 and 8 graphically. The proposed method fails to achieve a better result than the baseline SVM or PCA in both the scenarios where the training set size includes 6 words and 22 words. It is observed that the performance of the proposed method is improved by a margin of 2.73% when using a larger training set size.

Tables 7 and 8 show the results from the text-independent pairwise stress classification experiment. Figures 15 and 16 show the results in corresponding Tables 7 and 8 graphically. The proposed method fails to achieve a better result than the baseline SVM or PCA in both the scenarios where the training set size includes 6 words and 22 words. It is observed that the performance of the proposed method is improved by a margin of 2.73% when using a larger training set size.

For the simulated domain the training set was identical to those used in experiment 1 but the test set consisted of 270 stressful recordings (’angry’, ’loud’ and ’lombard’) from vocabulary words that were different from the training set and 272 neutral style recordings from out-of-vocabulary words. In the actual domain, the training set comprised 94 speech recordings from within vocabulary by seven speakers and the corresponding 94 ‘actual neutral’ style recordings. The test set included 140 out-of-vocabulary recordings under ‘actual stress’ conditions and 272 neutral style recordings using out-of-vocabulary words. A linear SVM was used to perform binary classification.

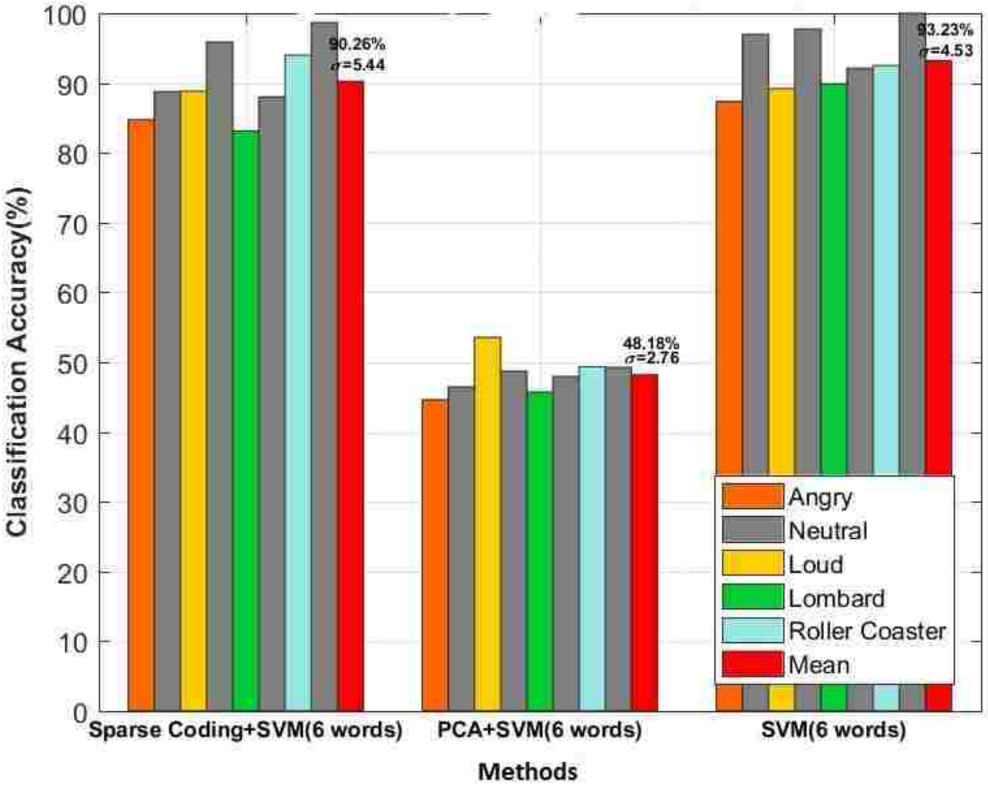

**Figure 15:** Comparison of proposed method against baseline and PCA methods in the text-independent pairwise scenario. Six words are used to extract all the features which are used as input.

**Table 7:** Classification accuracies for the text-independent pairwise scenario by using different methods. Six words are used to extract all the features which are used as input.

| **Method** | **Experiment Variations** | **Angry + Simulated Neutral (%)** | **Loud + Simulated Neutral (%)** | **Lombard + Simulated Neutral (%)** | **Roller Coaster Stress + Actual Neutral(%)** | **Overall Mean + Std(%)** |
|---|---|---|---|---|---|---|
| All Fea+SC+SVM (Proposed Model) | 6 words with all features | Angry=84.75 Neutral=88.76 | Loud=88.84 Neutral=95.88 | Lombard=83.14 Neutral=88.01 | Roller Coaster Stress=94.02 Neutral=98.69 | Mean=90.26 ($\sigma = 5.44$) |
| PCA+SVM | 6 words with all features | Angry=44.60 Neutral=46.44 | Loud=53.53 Neutral=48.68 | Lombard=45.69 Neutral=47.94 | Roller Coaster Stress=49.36 Neutral=49.25 | Mean=48.18 ($\sigma = 2.76$) |
| All Fea+SVM | 6 words with all features | Angry=87.36 Neutral=97.00 | Loud=89.21 Neutral=97.75 | Lombard=89.88 Neutral=92.13 | Roller Coaster Stress=92.53 Neutral=100 | Mean=93.23 ($\sigma = 4.53$) |

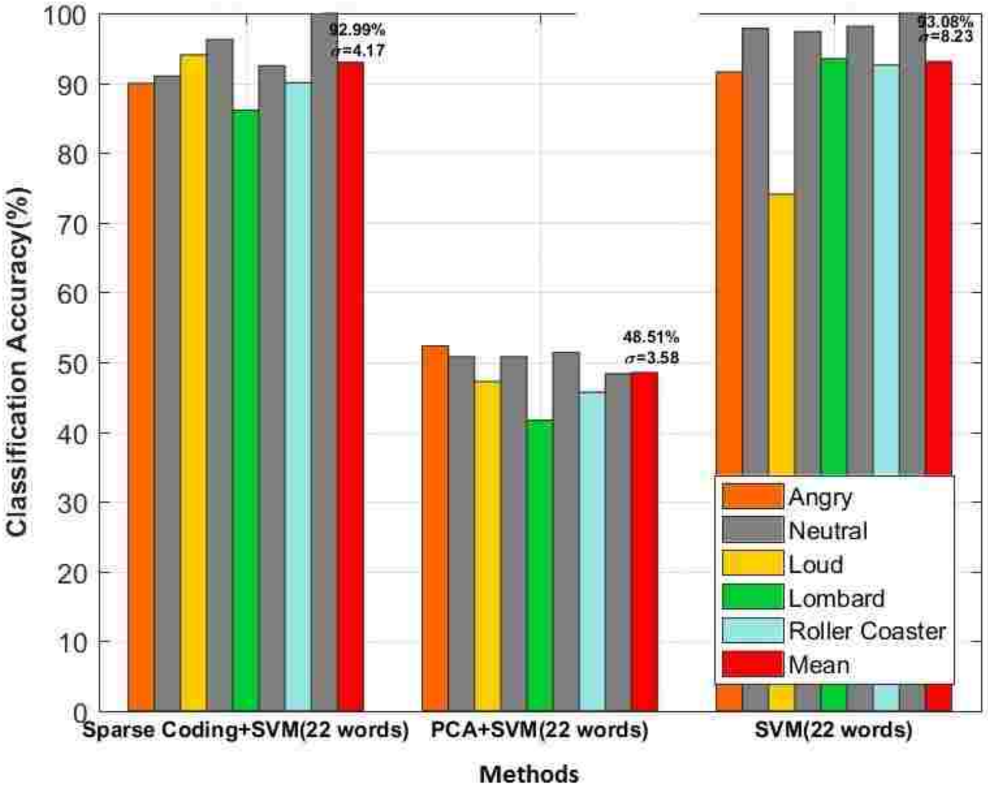

**Figure 16:** Comparison of proposed method against baseline and PCA methods in the text-independent pairwise scenario. Twenty-two words are used to extract all the features which are used as input.

**Table 8:** Classification accuracies for the text-independent pairwise scenario by using different methods. Twenty-two words are used to extract all the features which are used as input.

| **Method** | **Experiment Variations** | **Angry + Simulated Neutral (%)** | **Loud + Simulated Neutral (%)** | **Lombard + Simulated Neutral (%)** | **Roller Coaster Stress + Actual Neutral(%)** | **Overall Mean + Std(%)** |
|---|---|---|---|---|---|---|
| Proposed Model | 22 words with all features | Angry=89.96<br>Neutral=91.01 | Loud=94.05<br>Neutral=96.25 | Lombard=86.14<br>Neutral=92.50 | Roller Coaster Stress=90.04<br>Neutral=100 | Mean=92.99<br>($\sigma = 4.17$) |
| PCA+SVM | 22 words with all features | Angry=52.33<br>Neutral=50.77 | Loud=47.22<br>Neutral=50.77 | Lombard=41.66<br>Neutral=51.38 | Roller Coaster Stress=45.67<br>Neutral=48.33 | Mean=48.51<br>($\sigma = 3.58$) |
| All Fea+SVM | 22 words with all features | Angry=91.58<br>Neutral=97.83 | Loud=74.07<br>Neutral=97.37 | Lombard=93.51<br>Neutral=98.14 | Roller Coaster Stress=92.59<br>Neutral=100 | Mean=93.08<br>($\sigma = 8.23$) |

Figure 17 depicts the results corresponding to Table 17 graphically. From Table 9 it is observed that the proposed method achieves a best classification accuracy of 94.86% with a standard deviation of 5.36% on the training set in which the excitation features were excluded. This result outperforms the baseline SVM result of 93.08% by a margin of 1.78%. The above result implies that excitation features do not contribute appreciably to an improvement in the classification performance suggesting a weak discrimination capacity. It is also observed that excluding vocal-tract features from the training set results in a lower classification accuracy of 86.46% suggesting a strong discrimination capability of this category of features.

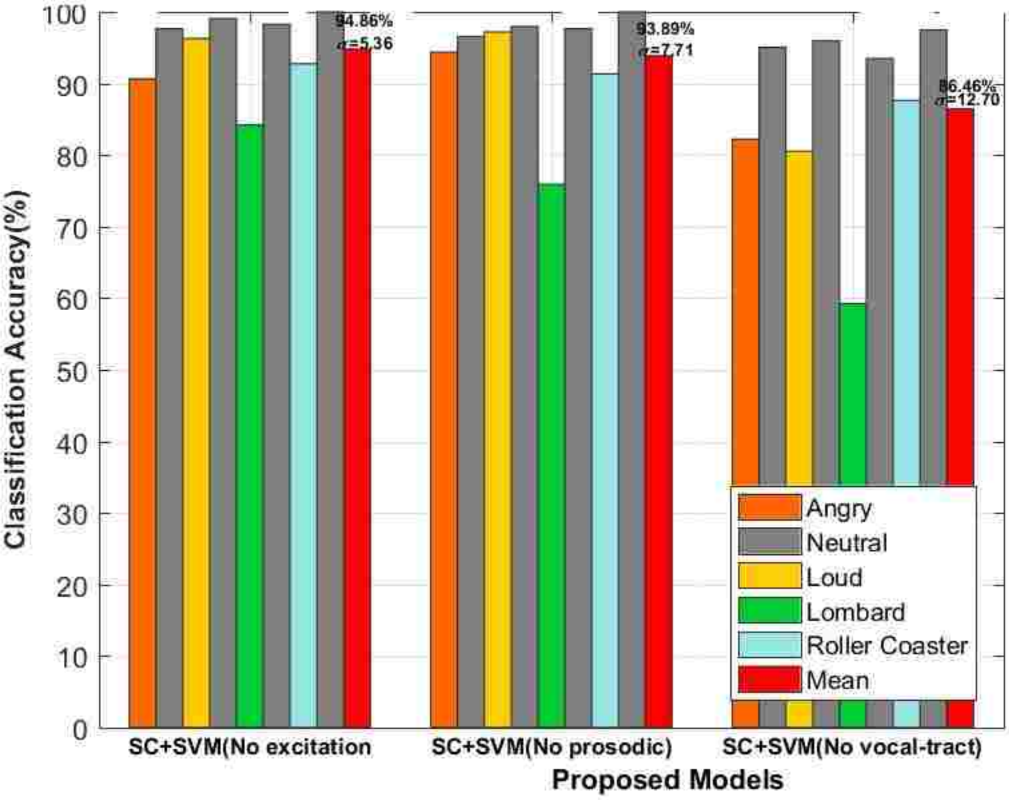

**Figure 17:** Classification accuracies achieved by the proposed method in the text-independent pairwise scenario across different feature combinations.

#### 4.1.2.3 TEXT INDEPENDENT MULTISTYLE EMOTION RECOGNITION

In this scenario, the aim was to assess the features in discriminating multiple stress styles. Actual domain data was not considered as the stress content in the voice tones was more conspicuous making it easily detectable. For the simulated domain, a multi-class SVM classifier was trained to discriminate among the four different speech styles simultaneously. In this experiment, the training data originally utilized in experiment 2 for discriminating "angry",

**Table 9:** Classification accuracies for the text-independent pairwise scenario across different feature combinations.

| **Method** | **Experiment Variations** | **Angry + Simulated Neutral (%)** | **Loud + Simulated Neutral (%)** | **Lombard + Simulated Neutral (%)** | **Roller Coaster Stress + Actual Neutral(%)** | **Overall Mean + Std(%)** |
|---|---|---|---|---|---|---|
| Proposed Model | Excitation features excluded | Angry=90.65<br>Neutral=97.67 | Loud=96.29<br>Neutral=99.07 | Lombard=84.21<br>Neutral=98.29 | Roller Coaster Stress=92.77<br>Neutral=100 | Mean=94.86<br>($\sigma$ = 5.36) |
| Proposed Model | Prosodic features excluded | Angry=94.39<br>Neutral=96.59 | Loud=97.22<br>Neutral=97.98 | Lombard=75.92<br>Neutral=97.67 | Roller Coaster Stress=91.35<br>Neutral=100 | Mean=93.89<br>($\sigma$ = 7.71) |
| Proposed Model | Vocal-tract features excluded | Angry=82.24<br>Neutral=95.06 | Loud=80.55<br>Neutral=95.98 | Lombard=59.25<br>Neutral=93.51 | Roller Coaster Stress=87.65<br>Neutral=97.50 | Mean=86.46<br>($\sigma$ = 12.70) |
| Average (Tables 7 + 8) | | 86.83 | 89.97 | 82.36 | 90.67 | |

**Table 10:** Classification results of text-independent multistyle stress classification by different sparse-coding based methods. Six words are used to extract all the features which are used as input.

| Method | Experiment Variations | Distribution of Speech Style Detection Rate(%) | | | | | Neutral-Stressed(%) | |
|---|---|---|---|---|---|---|---|---|
| | | **Input Test Speech Style** | **Neutral** | **Angry** | **Loud** | **Lombard** | **Neutral** | **Stressed** |
| Proposed Model | 6 words with all features | Neutral | **78.65** | 3.74 | 2.62 | 14.98 | **78.65** | 21.35 |
| | | Angry | 7.80 | **61.71** | 19.33 | 11.15 | 7.80 | **92.20** |
| | | Loud | 5.57 | 18.21 | **55.76** | 20.44 | 5.57 | **94.43** |
| | | Lombard | 9.36 | 14.23 | 14.23 | **62.17** | 9.36 | **90.64** |
| | | Average | **64.57** | | | | **88.98** | |
| PCA + SVM | 6 words with all features | Neutral | **25.46** | 25.09 | 25.09 | 24.34 | **25.46** | 74.54 |
| | | Angry | 19.70 | **24.90** | 28.25 | 27.13 | 19.70 | **80.30** |
| | | Loud | 24.90 | 23.79 | **24.53** | 26.76 | 24.90 | **75.10** |
| | | Lombard | 27.34 | 26.21 | 25.84 | **20.59** | 27.34 | **72.60** |
| | | Average | **23.87** | | | | **63.36** | |
| All features + SVM | 6 words with all features | Neutral | **89.13** | 0.76 | 0.00 | 10.11 | **89.13** | 10.87 |
| | | Angry | 12.63 | **62.08** | 16.74 | 8.55 | 12.63 | **87.37** |
| | | Loud | 12.63 | 62.08 | **16.74** | 8.55 | 12.63 | **87.37** |
| | | Lombard | 8.18 | 18.97 | 52.41 | **20.44** | 8.18 | **91.82** |
| | | Average | **47.09** | | | | **91.82** | |

"loud" and "Lombard" versus "simulated neutral" were combined to train a multiclass SVM classifier. The same testing data used in experiment 2 for the simulated domain was used for testing. Again, similar variations as those in experiments 1 and 2 were conducted.

**Table 11:** Classification results of text-independent multistyle stress classification by different sparse-coding based methods. Twenty-two words are used to extract all the features which are used as input.

| Method | Experiment Variations | Distribution of Speech Style Detection Rate(%) | | | | | Neutral-Stressed(%) | |
|---|---|---|---|---|---|---|---|---|
| | | **Input Test Speech Style** | **Neutral** | **Angry** | **Loud** | **Lombard** | **Neutral** | **Stressed** |
| Proposed Model | 22 words with all features | Neutral | **85.39** | 4.11 | 2.24 | 8.23 | **85.39** | 14.61 |
| | | Angry | 7.80 | **62.08** | 19.70 | 10.40 | 7.80 | **92.20** |
| | | Loud | 1.48 | 17.84 | **65.42** | 15.24 | 1.48 | **98.52** |
| | | Lombard | 8.98 | 12.73 | 14.23 | **64.04** | 8.98 | **91.02** |
| | | Average | **69.23** | | | | **91.78** | |
| PCA + SVM | 22 words with all features | Neutral | **67.43** | 17.12 | 8.79 | 6.63 | **67.43** | 32.57 |
| | | Angry | 60.74 | **22.42** | 5.60 | 11.21 | 60.74 | **39.26** |
| | | Loud | 62.03 | 16.67 | **8.33** | 12.96 | 62.03 | **37.97** |
| | | Lombard | 60.18 | 15.74 | 17.59 | **6.48** | 60.18 | **39.82** |
| | | Average | **26.16** | | | | **46.12** | |
| All features + SVM | 22 words with all features | Neutral | **97.68** | 1.69 | 0.46 | 0.15 | **97.68** | 2.32 |
| | | Angry | 8.41 | **67.28** | 14.95 | 9.34 | 8.41 | **91.59** |
| | | Loud | 5.55 | 17.59 | **57.40** | 19.44 | 5.55 | **94.45** |
| | | Lombard | 32.40 | 9.25 | 5.55 | **52.77** | 32.40 | **67.60** |
| | | Average | **68.78** | | | | **87.83** | |

Tables 10 and 11 show accuracies for the text-independent multistyle classification of stress. For each method, detection rate distribution of each stress style is also displayed. It can be observed from Table 12 that the proposed method achieved a best mean accuracy of 92.23% for the case in which the training set size excluded excitation features. On the other hand, excluding vocal-tract features from the training set results in a much lower neutral-stressed detection rate of 84.78%.

It is also observed from Table 12 that the best emotion-style detection rate of 74.83% was achieved in the case where the training set excluded excitation features, significantly higher than the baseline SVM accuracy of 68.78%. Similar conclusions are drawn as in the case of the text-independent scenario. When compared to the baseline method (last row in Table 12), the proposed method results in much better classification performance. Principal component analysis performed poorly in this context achieving only a best stressed-neutral detection rate of 63.36% and a best emotion-style detection rate of 26.16%.

**Table 12:** Classification results of text-independent multistyle stress classfication using different sparse-coding based methods across different feature combinations.

| Method | Experiment Variations | Distribution of Speech Style Detection Rate(%) | | | | | Neutral-Stressed(%) | |
|---|---|---|---|---|---|---|---|---|
| | | **Input Test Speech Style** | **Neutral** | **Angry** | **Loud** | **Lom-bard** | **Neutral** | **Stressed** |
| Proposed Model | Excitation features excluded | Neutral | **95.97** | 2.79 | 0.32 | 0.92 | **95.97** | 4.03 |
| | | Angry | 6.54 | **64.48** | 19.64 | 9.34 | 6.54 | **93.46** |
| | | Loud | 2.77 | 15.75 | **74.07** | 7.41 | 2.77 | **97.23** |
| | | Lombard | 21.29 | 5.55 | 8.35 | **64.81** | 21.29 | **78.71** |
| | | Average | **74.83** | | | | **91.34** | |
| Proposed Model | Prosodic features excluded | Neutral | **93.96** | 3.09 | 0.47 | 2.48 | **93.96** | 6.04 |
| | | Angry | 5.60 | **66.35** | 19.64 | 8.41 | 5.60 | **94.40** |
| | | Loud | 1.85 | 20.37 | **62.96** | 14.82 | 1.85 | **98.15** |
| | | Lombard | 17.59 | 6.49 | 10.18 | **65.74** | 17.59 | **82.41** |
| | | Average | **72.25** | | | | **92.23** | |
| Proposed Model | Vocal-tract features excluded | Neutral | **86.41** | 4.32 | 2.48 | 6.79 | **86.41** | 13.59 |
| | | Angry | 9.34 | **59.81** | 24.29 | 6.56 | 9.34 | **90.66** |
| | | Loud | 9.25 | 23.16 | **49.07** | 18.52 | 9.25 | **90.75** |
| | | Lombard | 28.70 | 14.83 | 19.44 | **37.03** | 28.70 | **71.30** |
| | | Average | **58.08** | | | | **84.78** | |

#### 4.1.2.4 PARAMETER SELECTION

The performance of the sparse coding model depends greatly on the choices of parameters. Important ones include the size of the basis functions, stride step size and soft-thresholding parameter. Multiple experiments were performed on a single input file chosen from the text-dependent scenario containing 36 data points with *stressed emotion* and *neutral* labels. While applying sparse coding on the raw input features, one parameter was varied while keeping all others fixed. Leave-one-out cross validation was performed in determining the classification performance. This process was repeated for multiple combinations of parameters. The most optimal set of parameters was found to be 3000 basis functions with a size of 65, a stride step of 60, and a soft-thresholding parameter of 0.8. The results are shown in tables 71 through 79 which can be found in the Appendix section of this dissertation.

### 4.1.3 DEEP BELIEF NETWORK MODEL FOR EMOTION RECOGNITION

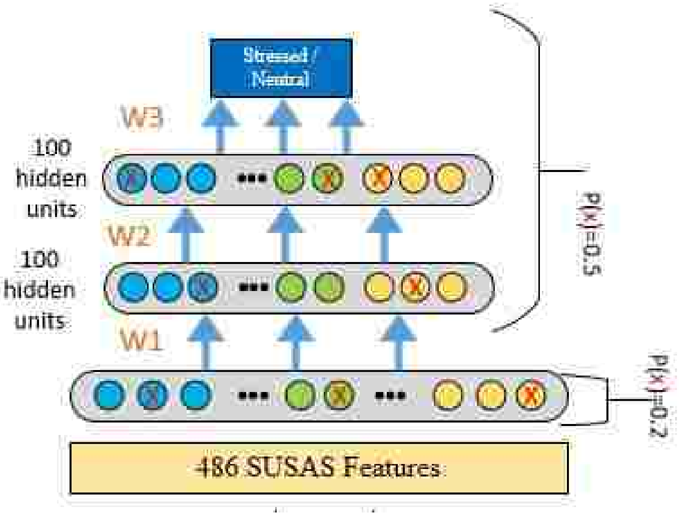

**Figure 18:** DBN network used for emotion recognition. The architecture was 486-100-100-2.

#### 4.1.3.1 DEEP BELIEF NETWORK MODEL BASED TEXT INDEPENDENT PAIRWISE EMOTION RECOGNITION

In this experiment, the proposed deep belief network (DBN) based classification model is applied to the SUSAS text-independent pairwise datasets to discriminate between stressed and neutral emotions. An example is shown in Figure 18. The datasets are selected and organized in the same way as described earlier in the sparse-coding model experiment using pairwise, text-independent data.The network architecture is 486-100-100-2 where the input layer has 486 features and the output layer has 2 label types, stressed emotion and neutral. There are two hidden layers with each containing 100 units. Initially, the network is pre-trained and dropout is applied with a probability of 0.2 for the input layer and 0.5 for the hidden layers. 100 epochs are used for pre-training and 200 epochs for fine-tuning. An initial momentum of 0.5 is used for the first 20 epochs and a final momentum of 0.9 is used. A learning rate of 0.001 is applied during pre-training. The type of activation function used is a sigmoid. A mini-batch size of 100 is used. For the fine-tuning step, an initial momentum of 0.5 is used for the first 10 epochs and 0.95 for the final momentum. The dropout applied is similar to the pre-training step. The number of epochs is 200. The output label layer consists of a logistic regression layer to classify between two distinct labels. A deeper architecture with the following configuration (486-2000-1000-500-2) is also applied for evaluation.

#### 4.1.3.2 DEEP BELIEF NETWORK MODEL BASED TEXT INDEPENDENT MULTI-STYLE EMOTION RECOGNITION

In this experiment, the proposed deep belief network based classification model is applied to the SUSAS text-independent multistyle datasets to discriminate between four different types of emotions. The datasets are selected and organized in the same way as described earlier in section 4.1.2.2. The text-dependent and text-independent pairwise evaluation scenarios are left out as the data sets are small and cannot be used to train DBNs. The network architecture and parameters are similar to that used in the case for applying the DBN to pairwise, text-independent data as described in the previous experiment. Many different configurations of the network are applied in order of increasing complexity that include 486-100-100-4, 486-500-100-4, 486-500-300-4, 486-500-300-100-4, 486-1000-500-200-100-50-4, 486-1000-500-200-100-4, 486-2000-500-4, 486-1000-500-4, 486-500-500-4, 486-2000-1000-500-4 and 486-4000-2000-1000-4.

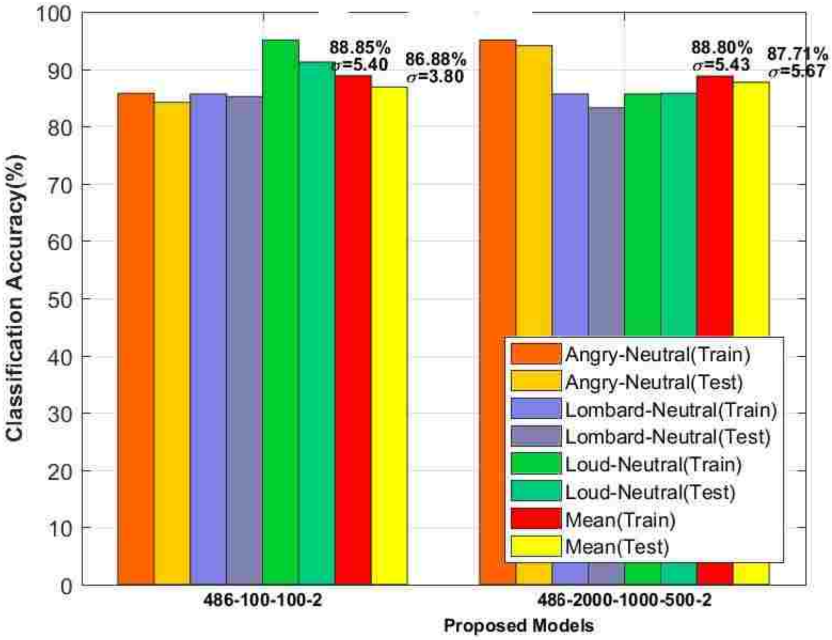

**Figure 19:** Classification results for different deep belief network architectures applied to the SUSAS text-independent pairwise datasets.

Table 13 shows the classification accuracies for the deep belief network model applied to SUSAS text-independent pairwise datasets. This is shown graphically in figure 19. Two types of network architectures have been used, each with a different number of hidden units in each hidden layer and varying numbers of layers. It is observed that the best accuracy of 94.11% is achieved for the case of the loud and neutral emotional pairwise dataset. The architecture, 486-100-100-2 achieves a marginally higher mean test accuracy of 86.88% with a standard deviation of 3.80%,

**Table 13:** Classification results for different deep belief network architectures applied to the SUSAS text-independent pairwise datasets.

| Type of Pairwise-Stress Input | Deep Belief Network Architecture (No. of Units Per Hidden Layer) | Training Accuracy(%) | Test Accuracy (%) |
|---|---|---|---|
| Angry + Simulated Neutral | 486-100-100-2 | 85.79 | 84.19 |
| Lombard + Simulated Neutral | | 85.67 | 85.22 |
| Loud + Simulated Neutral | | 95.09 | 91.23 |
| | Average: | 88.85<br>($\sigma = 5.40$) | 86.88<br>($\sigma = 3.80$) |
| Angry + Simulated Neutral | 486-2000-1000-500-2 | 95.08 | 94.11 |
| Lombard + Simulated Neutral | | 85.67 | 83.27 |
| Loud + Simulated Neutral | | 85.67 | 85.77 |
| | Average: | 88.80<br>($\sigma = 5.43$) | 87.71<br>($\sigma = 5.67$) |
| | Overall Average: | 88.82<br>($\sigma = 4.84$) | 87.29<br>($\sigma = 4.34$) |

across all the different types of pairwise input datasets. Also an overall accuracy of 87.29% with a standard deviation of 4.34% is obtained using this proposed deep belief network model.

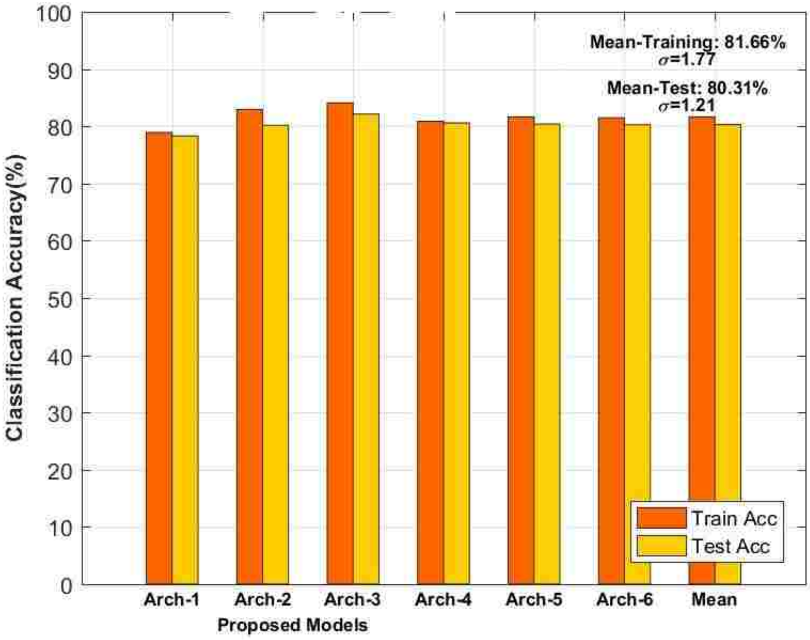

**Figure 20:** Classification results for different deep belief network architectures applied to the SUSAS text-independent pairwise datasets.

**Table 14:** Classification results for different deep belief network architectures applied to the SUSAS text-independent multistyle dataset.

| **Deep Belief Network Architecture** | **Training Accuracy(%)** | **Test Accuracy(%)** |
|---:|:---:|:---:|
| 486-100-100-4 | 78.92 | 78.32 |
| 486-500-100-4 | 82.94 | 80.18 |
| 486-500-300-4 | 84.10 | **82.14** |
| 486-1000-500-4 | 80.87 | 80.59 |
| 486-2000-500-4 | 81.64 | 80.39 |
| 486-1000-500-200-100-4 | 81.49 | 80.28 |
| Average: | 81.66<br>($\sigma = 1.77$) | 80.31<br>($\sigma = 1.21$) |

Figure 20 shows the results of applying the DBN model on the text-independent multistyle dataset graphically. Table 14 shows the tabulated results for this scenario. In this experiment, the datasets encompass four different emotional stress types all of which belong to the *simulated* domain of the SUSAS speech corpus. These include *anger*, *lombard*, *loud* and *neutral*.

The motivation for this particular experiment is that it is desired to test the effectiveness of the DBN based classification model in discriminating between these four different types of emotional stress. It is observed that best testing accuracy is obtained using the following network architecture of 486-500-300-4, where the input layer contains 486 features, the first hidden layer contains 500 stochastic binary hidden units and the final hidden layer has 300 hidden units. The output label layer consists of a logistic regression classifier to discriminate between the four different emotion labels. An average test accuracy of 80.31% with a standard deviation of 1.21% is obtained across all the different architectures. It is noted that the best accuracy of 82.14% is obtained with the following network architecture of 486-500-300-4.

## 4.2 DISCUSSION

### 4.2.1 SUMMARY OF BEST TESTING ACCURACIES FOR ALL MODELS ACROSS ALL EXPERIMENTS

Table 16 shows the results of the best test accuracies achieved by all the models in the proposed method across all the experiments. It is observed that for the text-dependent pairwise evaluation context, PCA achieved the best classification accuracy of 91.98% surpassing the proposed *sparse coding* model which achieved the best accuracy of 91.22%. It is observed that for the text-independent pairwise evaluation context, the proposed *sparse coding* model achieved the best classification accuracy of 94.86% outperforming the baseline which achieved the best accuracy of 93.23%. In text-independent multistyle scenario, the *sparse coding* model achieved the best emotion detection accuracy of 74.83% outperforming the baseline which achieved the best accuracy of 68.78%. In the text-independent pairwise scenario, the *deep belief network* model achieved a best of 94.11%. The *sparse-coding* model achieved the best overall performance.

### 4.2.2 SUMMARY OF HYPOTHESIS TESTING

Table 15 shows the hypothesis testing results in order to compare *sparse coding* against SVM for emotion recognition for the text-dependent pairwise evaluation scenario. The features are a combination of vocal-tract, prosodic and excitation feature categories. The level of significance, $\alpha$ is 5%. None of the cases show significantly different results.

**Table 15:** Application of hypothesis testing to compare the *sparse coding* against SVM for emotion recognition for the text-dependent pairwise evaluation scenario. The features are a combination of vocal-tract, prosodic and excitation feature categories. The level of significance, $\alpha$ is 5%. In the case of excluding vocal-tract features, it shows a statistically significant result, but the baseline accuracy is higher than that of sparse coding. The null hypothesis is that the accuracies from both methods come from a normal population distribution consisting of independent random samples with equal means and unknown variances. The alternative hypothesis is that the means are unequal.

| **Sparse Coding Methods** | **SVM Method**<br>**p-Value(SC% vs SVM%)** |
|---:|:---:|
| Sparse Coding (6 words) | $0.7208(90.86, 90.69)$ |
| Sparse Coding (22 words) | $0.6827(91.02, 91.95)$ |
| Sparse Coding (22 words + No prosodic) | $0.6422(91.22, 91.95)$ |

The null hypothesis is that the accuracies from both methods come from a normal population distribution consisting of independent random samples with equal means and unknown variances. The alternative hypothesis is that the means are unequal. It is also observed that when sparse coding is performed on a feature set in which the prosodic category is excluded, it achieved the best accuracy of 91.19%.

For the sparse-coding based text-dependent pairwise stress classification using SUSAS, the proposed method achieved a best accuracy of 92.06%. The baseline method, which trained a SVM classifier directly using all extracted features obtained an accuracy of 90.69% that is comparable to all other methods. In the text-dependent experiment, training and testing were conducted word-wise such that text information was implicitly utilized. It is observed that most of these methods can discriminate different stress types. It was also evident that sparse coding did not improve the performance significantly.

In the sparse-coding based text-independent experiment conducted on SUSAS, the proposed method achieved an accuracy of 94.86% with the excitation features being excluded, and the accuracy is higher than the baseline method (All Fea + SVM) which gives 93.23%. It implies that the excitation features are not very effective in detecting stress in speech. It is also observed that omitting prosodic features gives a classification accuracy of 93.89% also suggesting that perhaps the prosodic features are not very good stress detectors in speech. However, eliminating vocal-tract features reduces the classification accuracy to 86.46% showing that vocal-tract features are highly effective in detecting stress in speech. The accuracy of the PCA method (PCA features + SVM) is 48.36%, which is even slightly worse than a random guess (50%). In the text-independent experiment, speech recordings used for training and testing were from different words such that text information contained in the recordings was not utilized. The proposed method can discriminate different stress types without knowing the text information.

**Table 16:** Summary of best testing accuracies achieved by the models in the proposed method across all the experiments.

| **Evaluation Context** | **Model Type** | **Test Accuracy (Best) %** |
|---|---|---|
| Text-Dependent Pairwise | Sparse Coding (prosodic features excluded) | 91.22 |
| | PCA + SVM (22 words) | **91.98** |
| | All Features + SVM (22 words) | 91.95 |
| Text-Independent Pairwise | Sparse Coding (excitation features excluded) | **94.86** |
| | PCA + SVM (22 words) | 48.51 |
| | All Features + SVM (6 words) | 93.23 |
| | Deep belief Network (22 words) | 87.71 |
| Text-Independent Multistyle | Sparse Coding (excitation features excluded) | 74.83 |
| | PCA + SVM (22 words) | 26.16 |
| | All Features + SVM (22 words), (Baseline) | 68.78 |
| | Deep belief Network (22 words) | **80.31** |

In the sparse-coding based multistyle stress classification experiment on the SUSAS speech database, the proposed method achieved a best accuracy of 75.08% in the simulated domain. The baseline method did not perform well with an accuracy of 67.66%. Accuracies from all sparse coding based methods are in the range of 58% - 75% for the simulated domain and of 84% - 92% for the actual domain. Discrimination of the multiple stress types simultaneously is a more challenging task. Table 16 shows a summary of the best test accuracies of all models across all the experimental scenarios for emotion recognition.

Text information could be independent of stress, i.e., we may express the same emotion type using different words. In our experiment, we showed that if the discriminating model was carefully designed (Table 6, the proposed method), a text-independent stress type classification system is feasible. We also showed that it is generally helpful for a system to perform stress type discrimination if it utilized text information, with an average accuracy of 89.92% across all methods in Tables 4, 5 and 6 combined, as compared to 80.29% in Tables 7, 8 and 9, if text information was not used. The baseline method utilized all features and SVM and performed reasonably well in our experiments. In Table 5, improvements are noted when PCA features (PCA+SVM) are directly used as input to the SVM classifier. PCA reduces the dimensionality of the data from 486 to 17 for a simulated domain and 15 for the actual domain. Classification accuracies were improved from 91.83% to 91.98% with PCA in Table 5. However, the classification accuracies are noted to drop to 48.36% with PCA in Table 7. In sparse coding, we learned 3000 basis functions with a size of 65 and a step size of 60. These hyperparameters were found after conducting multiple experiments, described in section 4.1.2.4, to observe which combination produced the best classification accuracy. With an initial feature vector size of 486, the final feature dimensionality after sparse coding is 24,000. The classifier used was an SVM, which is based on a regularized optimization procedure that aims to maximize its generalization capability [60]. It is well known that it is specifically effective for small-sized data sets. In our experiments, training data sets usually contain a couple hundred data samples, which can be considered as small as compared to the number of features in the data.

Feature selection algorithms such as the one in [3] usually select a subset of features that are most effective for a given task and a predefined objective. If data sets contain irrelevant or redundant features and do not have enough data samples for training, i.e., the number of data samples is less than the number of features, the classifier will tend to be over-trained leading to degraded performance. Dimensionality of data sets can also be reduced by the PCA technique by just keeping those top ranked principal components (PC). Each PC is a linear combination of all of the original features.

In the deep belief network based text-independent pairwise experiment conducted using SUSAS, the proposed deep belief network model achieved the emotion-wise best accuracy of 94.11% and an overall best accuracy of 87.71%. The performance is slightly better than the case in which sparse-coding based method was applied and vocal-tract features were excluded. However, it is to be noted that in this case the input training data consisted of only 6 words as against 22 words which formed a much larger training set in the case of the sparse-coding based SUSAS experiment. The result for the case in which vocal-tract features were excluded is 86.46%. For the case in which the deep belief network was applied for the text-independent multistyle scenario in the *simulated* domain of SUSAS, it achieved the best accuracy of 82.14% and an overall accuracy of 80.28%. This is higher than that obtained using sparse-coding which achieved an overall accuracy of 75.08%. Compared to the baseline which achieved only 67.66%, the deep belief network achieved a much better performance.

Statistical hypothesis testing was carried out for the text-dependent scenario, between the different sparse coding methods and SVM to assess the relative performance of emotion detection. Table 15 shows the results of hypothesis testing. The results are not significantly different in any of the cases. In the text-dependent scenario, it is concluded that sparse coding is not significantly better than the baseline. The results from these two tables suggests that the sample accuracies from the two different methods come from normally distributed populations with equal means. However, as discussed earlier, sparse coding outperforms the baseline and other methods in the text-independent pairwise scenario.

## 4.3 COMPARISON OF PROPOSED METHOD WITH RELATED WORK

**Table 17:** Comparison of related work with proposed method for emotion recognition based on speech.

| Method Used | Features Used | Results |
|---|---|---|
| TEO based framework | TEO based features extracted from SUSAS | 92.9% for pairwise text-dependent scenario, 89% for pairwise text-independent scenario, 88.85% for text-independent multistyle scenarios |
| Adaptive sinusoidal model based | Sinusoidal based extracted from SUSAS | Average 64.25% for multiclass |
| Multi-level classification framework on resting-state fMRI (Multi-kernel SVM) | Univariate, bivariate and multivariate features derived from fMRI | 92.5% classification accuracy |
| LDA Classification Framework | Pitch, log energies, MFCC's, velocity and acceleration features extracted from SUSAS | 91.3% for pairwise text-independent scenario, 70.1% for text-independent multistyle scenario |
| Integration framework | MFCC, delta and acceleration coefficients extracted from SUSAS | Best accuracy of 83.8% |
| Long Short Term Memory Neural network framework | MFCC and Lyon Cochleagram Model extracted from SUSAS | Best accuracy of 75.41% |

This section compares the performance of the proposed sparse-coding based model against those in the literature that used the same SUSAS benchmark speech corpus. The study conducted in [70] utilizing wavelet features achieved a mean accuracy of 90% for all combinations of pairwise stressed speech classification for the simulated domain. No results were reported for the actual domain in this study. As a comparison, our proposed method achieved an accuracy of 93.29% for the pairwise classifications combined in the simulated domain. In [71], nonlinear Teager Energy

Operator (TEO) based features were utilized for stress classification and achieved 92.9%, 89% and 88.85% accuracies for pairwise text-dependent, pairwise text-independent and text-independent multistyle classifications, respectively. In this study, we obtained 92.06%, 94.86% and 92.23% for the three cases and our method performed much better for the latter two cases.

The experiments conducted in [72] used pitch, log-energies, MFCC's, velocity and acceleration features of pitch as features. A mean accuracy of 91.3% (as compared to 94.86% in our experiment) was reported for the pairwise text-independent classification. For the multistyle scenario, a much lower accuracy of 70.1% was reported as compared to 92.23% in our experiment.

The study in [74] proposed to integrate frame-level information in speech into a large feature-space emotion recognition engine. The method utilized MFCC, its delta and acceleration coefficients as features and an SVM as classifier for classification. A maximum accuracy of 83.8% was reported on the SUSAS database. Our proposed system achieves higher classification accuracies in the three different evaluation contexts described earlier.

A biologically inspired emotion recognition system was proposed in [75], in which features were derived from MFCC and the Lyon-cochleagram model. Classification was performed using a long short-term memory (LSTM) recurrent neural network. A multistyle classification involving five different emotions was performed for which the best achieved accuracy was reported to be 75.41% in the case of the Lyon-cochleagram model. Our proposed method achieved a much higher accuracy of 94.86% in the text-independent scenario. Another biologically inspired method was proposed in [76], which extracted vowel information from an input speech signal and converted it to features. The MFCC features were mapped into an appropriate spike representation after which a spiking neural network was used to discriminate five different emotion states in the SUSAS database. An average classification accuracy of 72% was reported. Our proposed method, in contrast, achieved a higher accuracy of 92.23% in the multistyle scenario. We did not perform feature selection in the current study; this is our planned future work.

## 4.4 CONCLUSION OF PROPOSED APPROACH

We compared the proposed *sparse-coding* based method using SUSAS and PTSD speech corpora with the following methods: 1) using all extracted features and then passing them to an SVM for classification, and 2) using the selected *PCA* features and an SVM for classification. The *PCA* method was only applied to the SUSAS speech corpus. The proposed *sparse-coding* based method, applied to SUSAS, used seven different sets of training data in order to compare the classification results. We also compared our results with those in the literature using similar subsets of the SUSAS database. In the case of the *proposed deep belief network* model, we also compared it to this method: 1) using all extracted features and passing them to SVM as our baseline.

In the text-dependent scenario, the *sparse coding* model achieved an increase of 1.37% over the baseline. The proposed *deep belief network* based text-independent experiment conducted on SUSAS, achieved the best accuracy of 94.11% which is comparable to the baseline result of 93.30%, while the *sparse-coding* model achieved a best accuracy of 94.86%. In the multistyle scenario, the *sparse coding* model achieved an overall 75.08%, while the *deep belief network* model achieved a best accuracy of 80.31%. Overall, *sparse coding* achieves the best performance in the text-independent pairwise scenario whereas the *deep belief network* model performs best in the text-independent multistyle scenario which could be attributed to the availability of larger training data.

# CHAPTER 5

# PTSD DIAGNOSIS

This chapter begins with a discussion of the two different speech corpora utilized for PTSD diagnosis. A description of the experiments, results and discussion of the results follows in the next section. PTSD diagnosis using SVM on single-frame and multiple-frame raw feature sets is presented first. *Sparse coding* and the *deep belief network* models for PTSD detection are presented next. A *transfer learning* strategy was adopted to solve the small data size problem which is discussed in the final section. This section also summarizes important results and presents relevant discussion followed by conclusions. Hypothesis testing showed that when comparing *transfer learning* with SVM, *transfer learning* performed significantly better than the baseline in more than 31% of the cases tested. It performed significantly better than the *deep belief network* model in nearly 30% of the cases that were tested. *Transfer learning* also achieved significantly better results in 24% of the cases, compared to *sparse coding* for PTSD diagnosis.

## 5.1 EXPERIMENTS RESULTS AND DISCUSSION

### 5.1.1 PTSD DIAGNOSIS USING SVM MODEL

#### 5.1.1.1 PTSD DIAGNOSIS USING SVM ON SINGLE FRAME MULTI-CATEGORY RAW FEATURES

In order to set a reference for comparison, we used the raw PTSD features directly and applied SVM for classification. A total of 162 PTSD features were computed. To compute these features, various categories of features widely used in emotion recognition literature were used. The details of the type and number of each are shown in Table 3. The total number of raw features per frame is 54. Adding the first and second order time derivative features give us a total of 162 features for each frame.

Table 18 shows the results of classifying raw PTSD features directly with an SVM using the leave-one-subject-out cross validation. The PTSD features have 162 dimensions. Different sets of PTSD features have been sampled from the available recordings using variable speech frame sizes and frame shifts. While an overall subject-wise accuracy of 53.62% is achieved, the mean subject-wise accuracy on the Youtube dataset is 55.98% while that for Ohio is 51.27%. The overall mean segment-wise accuracy on Youtube is 56.84%, while for Ohio the mean segment-wise

**Table 18:** Classification results of applying SVM directly on raw features extracted from PTSD data set using the leave-one-subject-out cross validation. The raw input feature data set has 162 dimensions consisting of a combination of prosodic, vocal-tract and excitation features.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26 No of Subjects from Ohio:26 Total:52 subjects | 3.0 | 0.1 | 55.36<br>($\sigma = 49.73$) | 42.36<br>($\sigma = 47.18$) | 53.84<br>(14/26) | 38.46<br>(10/26) | 46.15 |
| | 3.0 | 0.5 | 55.88<br>($\sigma = 46.66$) | 57.61<br>($\sigma = 42.57$) | 57.69<br>(15/26) | 57.69<br>(15/26) | 57.69 |
| | 3.0 | 1.0 | 56.25<br>($\sigma = 49.25$) | 46.87<br>($\sigma = 46.39$) | 57.69<br>(15/26) | 46.15<br>(12/26) | 51.92 |
| | 2.0 | 0.1 | 55.95<br>($\sigma = 44.66$) | 51.43<br>($\sigma = 48.37$) | 57.69<br>(15/26) | 50.00<br>(13/26) | 53.84 |
| | 2.0 | 0.5 | 55.86<br>($\sigma = 45.36$) | 57.94<br>($\sigma = 44.43$) | 61.53<br>(16/26) | 61.53<br>(16/26) | 61.53 |
| | 2.0 | 1.0 | 50.35<br>($\sigma = 46.88$) | 50.41<br>($\sigma = 43.03$) | 50.00<br>(13/26) | 50.00<br>(13/26) | 50.00 |
| | 1.0 | 0.1 | 63.98<br>($\sigma = 44.98$) | 53.18<br>($\sigma = 46.35$) | 57.69<br>(15/26) | 53.84<br>(14/26) | 55.76 |
| | 1.0 | 0.5 | 64.92<br>($\sigma = 46.75$) | 54.18<br>($\sigma = 43.16$) | 57.69<br>(15/26) | 53.84<br>(14/26) | 55.76 |
| | 1.0 | 1.0 | 53.05<br>($\sigma = 45.83$) | 54.95<br>($\sigma = 40.62$) | 50.00<br>(13/26) | 50.00<br>(13/26) | 50.00 |
| | | Average: | 56.84<br>($\sigma = 4.71$) | 52.10<br>($\sigma = 5.04$) | 55.98 | 51.27 | **53.62** |

accuracy is 52.10%. A best segment-wise accuracy of 64.92% is obtained which is seen to be from the Youtube data set.

#### 5.1.1.2 PTSD DIAGNOSIS USING SVM DIRECTLY ON MULTIPLE FRAME MULTI-CATEGORY RAW FEATURES

In order to set a reference for comparison, we used 15-frame raw PTSD features directly and applied SVM for classification. A total of 2,430 PTSD features were computed. More details about the feature extraction process are included in the third paragraph of section 3.3.3.

Table 19 shows the results of raw PTD feature classification directly with an SVM using the leave-one-subject-out cross validation. The results are shown for different sets of PTSD data sampled differently using variable frame sizes and shifts. The mean subject-wise accuracy on the Youtube dataset is 51.36% while that for Ohio is 57.26%. The overall mean segment-wise accuracy on Youtube is 57.05% with a standard-deviation of 3.77% while for Ohio the mean

**Table 19:** Classification results of applying SVM directly on raw features extracted from PTSD data set using the leave-one-subject-out cross validation. The raw input feature data set has 2430 dimensions consisting of prosodic, vocal-tract and excitation features.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26<br>No of Subjects from Ohio:26<br>Total:52 subjects | 3.0 | 0.1 | 59.28<br>($\sigma = 50.18$) | 47.03<br>($\sigma = 48.46$) | 53.84<br>(14/26) | 50.00<br>(13/26) | 51.92 |
| | 3.0 | 0.5 | 61.93<br>($\sigma = 47.32$) | 43.03<br>($\sigma = 44.44$) | 61.53<br>(16/26) | 46.15<br>(12/26) | 53.84 |
| | 3.0 | 1.0 | 50.12<br>($\sigma = 50.36$) | 40.18<br>($\sigma = 46.14$) | 50.00<br>(13/26) | 42.30<br>(11/26) | 46.15 |
| | 2.0 | 0.1 | 55.73<br>($\sigma = 49.51$) | 51.28<br>($\sigma = 46.20$) | 57.69<br>(15/26) | 53.84<br>(14/26) | 55.76 |
| | 2.0 | 0.5 | 56.34<br>($\sigma = 49.52$) | 52.95<br>($\sigma = 47.29$) | 57.69<br>(15/26) | 53.94<br>(14/26) | 55.81 |
| | 2.0 | 1.0 | 59.53<br>($\sigma = 45.81$) | 52.95<br>($\sigma = 47.29$) | 61.53<br>(16/26) | 53.84<br>(14/26) | 57.68 |
| | 1.0 | 0.1 | 56.04<br>($\sigma = 48.36$) | 52.13<br>($\sigma = 46.31$) | 57.69<br>(15/26) | 53.84<br>(14/26) | 55.76 |
| | 1.0 | 0.5 | 60.97<br>($\sigma = 47.22$) | 69.78<br>($\sigma = 41.25$) | 61.53<br>(16/26) | 76.92<br>(20/26) | 69.22 |
| | 1.0 | 1.0 | 53.57<br>($\sigma = 50.60$) | 52.98<br>($\sigma = 49.61$) | 53.84<br>(14/26) | 53.84<br>(14/26) | 53.84 |
| | | Average: | 57.05<br>($\sigma = 3.77$) | 51.36<br>($\sigma = 8.37$) | 57.26 | 53.85 | **55.55** |

segment-wise accuracy is 51.36% with a standard-deviation of 8.37%. The overall subject-wise accuracy is 55.55% which is higher than the case of applying SVM on raw features having 162 dimensions that achieved 53.62%.

#### 5.1.1.3 PTSD DIAGNOSIS USING SVM DIRECTLY ON MULTIPLE FRAME MFCC FEATURES

Raw MFCC features are also used since they have been found to be very useful in the literature. A raw MFCC feature data set with 15 frames, each frame having 39 features forming a total of 585 (39x15) features in total is computed. An SVM is applied directly on this raw data set to serve as our baseline. It is observed from Table 20 that the overall subject-wise test accuracy for Youtube data is 73.07% and for Ohio data is 56.53%. The average segment-wise accuracy for Youtube data is 79.10% with a standard deviation of 2.16% and for Ohio data, 53.05% with a standard deviation of 1.90%. An overall subject-wise accuracy of 64.74% is obtained.

**Table 20:** Classification results of applying SVM directly on raw MFCC features extracted from PTSD dataset using the leave-one-subject-out cross validation. The raw input feature data set extracted from PTSD speech corpus consists of multiple frame MFCC features having 585 dimensions.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26 No of Subjects from Ohio:26 Total:52 subjects | 3.0 | 0.1 | 81.58<br>($\sigma = 25.08$) | 55.18<br>($\sigma = 26.11$) | 73.08<br>(19/26) | 61.53<br>(16/26) | 67.30 |
| | 3.0 | 0.5 | 81.90<br>($\sigma = 24.99$) | 50.37<br>($\sigma = 24.88$) | 76.92<br>(20/26) | 53.84<br>(14/26) | 65.38 |
| | 3.0 | 1.0 | 79.43<br>($\sigma = 25.57$) | 54.27<br>($\sigma = 24.34$) | 73.08<br>(19/26) | 57.69<br>(15/26) | 65.38 |
| | 2.0 | 0.1 | 80.06<br>($\sigma = 24.57$) | 54.23<br>($\sigma = 24.25$) | 73.08<br>(19/26) | 57.69<br>(15/26) | 65.38 |
| | 2.0 | 0.5 | 80.22<br>($\sigma = 24.31$) | 51.44<br>($\sigma = 23.39$) | 69.23<br>(18/26) | 50.00<br>(13/26) | 59.61 |
| | 2.0 | 1.0 | 77.49<br>($\sigma = 25.47$) | 55.32<br>($\sigma = 26.08$) | 73.08<br>(19/26) | 61.53<br>(16/26) | 67.30 |
| | 1.0 | 0.1 | 77.95<br>($\sigma = 23.58$) | 53.40<br>($\sigma = 22.20$) | 76.92<br>(20/26) | 57.69<br>(15/26) | 67.30 |
| | 1.0 | 0.5 | 78.29<br>($\sigma = 23.26$) | 52.79<br>($\sigma = 21.86$) | 69.23<br>(18/26) | 57.69<br>(15/26) | 63.46 |
| | 1.0 | 1.0 | 75.02<br>($\sigma = 24.25$) | 50.49<br>($\sigma = 20.56$) | 73.08<br>(19/26) | 50.00<br>(13/26) | 61.54 |
| | | Average: | 79.10<br>($\sigma = 2.16$) | 53.05<br>($\sigma = 1.90$) | 73.07 | 56.53 | **64.74** |

### 5.1.2 SPARSE CODING MODEL FOR PTSD DIAGNOSIS

The sparse coding model discussed in section 3.1.1 is applied to the PTSD raw data. More details about the feature extraction process are included in the third paragraph of section 3.3.3. A single- frame data set was created using single frames by combining the three distinct categories of speech features such as prosodic, vocal-tract and excitation speech features. Each frame consisted of 162 features. The multi-frame data set comprised of 15 single frames concatenated together to form a total of 2,430 features. Similar single frame and multi-frame features were computed for for MFCC features. The single frame consisted of 39 features. The multi-frame MFCC data set consisted of 15 single frames concatenated together to form a total of 585 features. Sparse coding was applied to the extracted single and multiple frame feature data sets. We learned 3000 basis functions, randomly selected from the data with a size of 65 and a step size of 55. The hyperparameters were found after conducting multiple experiments in order to determine the optimal set of parameters based on classification accuracy. The classification was performed using a linear kernel SVM whose optimal parameters were found using gridsearch.

#### 5.1.2.1 SPARSE CODING MODEL FOR PTSD DIAGNOSIS BASED ON SINGLE FRAME MULTI-CATEGORY FEATURES

In this scenario, the input feature data set consisted of single frame features with each frame having 162 features. The raw input features extracted from the PTSD speech database consisted of a combination of three different categories of speech features, namely the prosodic, vocal-tract and excitation speech features.

**Table 21:** Classification results of applying the sparse coding based model for PTSD diagnosis. The raw input features extracted from PTSD speech corpus have 162 dimensions which consisted of prosodic, vocal-tract and excitation features.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26 No of Subjects from Ohio:26 Total:52 subjects | 3.0 | 0.1 | 68.40<br>$(\sigma = 32.23)$ | 57.85<br>$(\sigma = 28.37)$ | 73.07<br>(19/26) | 61.53<br>(16/26) | 67.30 |
| | 3.0 | 0.5 | 71.62<br>$(\sigma = 28.59)$ | 55.52<br>$(\sigma = 29.13)$ | 76.92<br>(20/26) | 61.53<br>(16/26) | 69.22 |
| | 3.0 | 1.0 | 69.52<br>$(\sigma = 30.57)$ | 56.66<br>$(\sigma = 24.87)$ | 76.92<br>(20/26) | 65.38<br>(17/26) | 71.15 |
| | 2.0 | 0.1 | 67.11<br>$(\sigma = 28.00)$ | 56.53<br>$(\sigma = 25.90)$ | 73.07<br>(19/26) | 57.69<br>(15/26) | 65.38 |
| | 2.0 | 0.5 | 65.79<br>$(\sigma = 29.30)$ | 57.11<br>$(\sigma = 25.65)$ | 73.07<br>(19/26) | 61.53<br>(16/26) | 67.30 |
| | 2.0 | 1.0 | 65.59<br>$(\sigma = 28.74)$ | 55.43<br>$(\sigma = 22.72)$ | 69.23<br>(18/26) | 61.53<br>(16/26) | 65.38 |
| | 1.0 | 0.1 | 66.24<br>$(\sigma = 26.03)$ | 55.06<br>$(\sigma = 20.89)$ | 73.07<br>(19/26) | 65.38<br>(17/26) | 69.22 |
| | 1.0 | 0.5 | 66.17<br>$(\sigma = 24.95)$ | 54.89<br>$(\sigma = 18.89)$ | 73.07<br>(19/26) | 57.69<br>(15/26) | 65.38 |
| | 1.0 | 1.0 | 64.19<br>$(\sigma = 25.57)$ | 55.01<br>$(\sigma = 20.25)$ | 69.23<br>(18/26) | 61.53<br>(16/26) | 65.38 |
| | | Average: | 67.18<br>$(\sigma = 2.29)$ | 56.00<br>$(\sigma = 1.06)$ | 73.07 | 61.53 | **67.30** |

Table 21 shows that the segment-wise accuracy for Youtube subjects is 67.18% while for Ohio it is 56.00%. The subject-wise accuracy is 73.07% for Youtube while for Ohio subjects it is 61.53%. The overall subject-wise accuracy is found to be 67.30% which is considerably higher than the baseline achievement of 53.62%.

#### 5.1.2.2 SPARSE CODING MODEL FOR PTSD DIAGNOSIS BASED ON MULTIPLE FRAME MULTI-CATEGORY FEATURES

In this scenario, the input feature data set consisted of multiple frame features with each frame having 2,430 features. The raw input features extracted from the PTSD speech database consisted of a combination of three different categories of speech features, namely the prosodic, vocal-tract and excitation speech features.

**Table 22:** Classification results of applying the sparse coding based model for PTSD diagnosis. The raw input features extracted from PTSD speech corpus have 2430 dimensions which consisted of prosodic, vocal-tract and excitation features.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26 No of Subjects from Ohio:26 Total:52 subjects | 3.0 | 0.1 | 64.10 ($\sigma = 32.77$) | 63.81 ($\sigma = 24.29$) | 73.07 (19/26) | 73.07 (19/26) | 73.07 |
| | 3.0 | 0.5 | 62.98 ($\sigma = 36.26$) | 63.48 ($\sigma = 32.34$) | 61.53 (16/26) | 65.38 (17/26) | 63.45 |
| | 3.0 | 1.0 | 58.18 ($\sigma = 37.62$) | 61.40 ($\sigma = 34.48$) | 50.00 (13/26) | 73.07 (19/26) | 61.53 |
| | 2.0 | 0.1 | 65.04 ($\sigma = 30.97$) | 62.56 ($\sigma = 24.34$) | 73.07 (19/26) | 65.38 (17/26) | 69.22 |
| | 2.0 | 0.5 | 62.19 ($\sigma = 36.32$) | 64.31 ($\sigma = 32.80$) | 61.53 (16/26) | 61.53 (16/26) | 61.53 |
| | 2.0 | 1.0 | 62.84 ($\sigma = 34.16$) | 64.31 ($\sigma = 35.27$) | 57.69 (15/26) | 69.23 (18/26) | 63.46 |
| | 1.0 | 0.1 | 66.55 ($\sigma = 31.51$) | 59.54 ($\sigma = 21.71$) | 76.92 (20/26) | 69.23 (18/26) | 73.07 |
| | 1.0 | 0.5 | 64.31 ($\sigma = 35.09$) | 64.95 ($\sigma = 28.41$) | 65.38 (17/26) | 73.07 (19/26) | 69.22 |
| | 1.0 | 1.0 | 61.88 ($\sigma = 36.39$) | 65.56 ($\sigma = 30.86$) | 61.53 (16/26) | 69.23 (18/26) | 65.38 |
| | | Average: | 63.11 ($\sigma = 2.36$) | 63.32 ($\sigma = 1.88$) | 64.52 | 68.79 | **66.65** |

Table 22 shows the results. It is observed from Table 22 that the segment-wise accuracy for Youtube subjects is 63.11% while for the Ohio data it is 63.32%. The subject-wise accuracy is 64.52% for Youtube while for Ohio subjects it is 68.79%. The overall subject-wise accuracy is found to be 66.65% which is higher than that of the baseline achievement of 55.55%.

#### 5.1.2.3 SPARSE CODING MODEL FOR PTSD DIAGNOSIS BASED ON MULTIPLE FRAME MFCC FEATURES

In this scenario, the input feature data set consisted of single frame features with each frame having 162 features. The raw input features extracted from the PTSD speech database consisted of MFCC speech features.

**Table 23:** Classification results of applying the sparse coding based model for PTSD diagnosis. The raw input features extracted from PTSD speech corpus consisted of multiple frame MFCC features having 585 dimensions.

| Number of Subjects | Frame Length (seconds) | Frame Shift (seconds) | Segment-wise Acc (%) on Youtube | Segment-wise Acc (%) on Ohio | Subject-wise Acc (%) on Youtube | Subject-wise Acc (%) on Ohio | Overall Subject-wise Accuracy (%) |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26 No of Subjects from Ohio:26 Total:52 subjects | 3.0 | 0.1 | 74.95<br>($\sigma = 23.26$) | 50.83<br>($\sigma = 20.96$) | 80.76<br>(21/26) | 61.53<br>(16/26) | 71.14 |
| | 3.0 | 0.5 | 74.11<br>($\sigma = 22.00$) | 51.15<br>($\sigma = 17.14$) | 88.46<br>(23/26) | 65.38<br>(17/26) | 76.92 |
| | 3.0 | 1.0 | 73.32<br>($\sigma = 22.47$) | 49.03<br>($\sigma = 16.50$) | 84.61<br>(22/26) | 53.84<br>(14/26) | 69.22 |
| | 2.0 | 0.1 | 71.76<br>($\sigma = 21.99$) | 49.86<br>($\sigma = 19.02$) | 76.92<br>(20/26) | 65.38<br>(17/26) | 71.15 |
| | 2.0 | 0.5 | 71.30<br>($\sigma = 20.66$) | 49.90<br>($\sigma = 15.66$) | 80.76<br>(21/26) | 65.38<br>(17/26) | 73.07 |
| | 2.0 | 1.0 | 72.64<br>($\sigma = 19.96$) | 48.88<br>($\sigma = 13.80$) | 84.61<br>(22/26) | 53.84<br>(14/26) | 69.22 |
| | 1.0 | 0.1 | 71.41<br>($\sigma = 20.41$) | 47.76<br>($\sigma = 16.45$) | 80.76<br>(21/26) | 53.84<br>(14/26) | 67.30 |
| | 1.0 | 0.5 | 68.80<br>($\sigma = 17.95$) | 47.71<br>($\sigma = 12.64$) | 80.76<br>(21/26) | 57.69<br>(15/26) | 69.22 |
| | 1.0 | 1.0 | 68.07<br>($\sigma = 18.28$) | 49.79<br>($\sigma = 10.83$) | 76.92<br>(20/26) | 38.46<br>(10/26) | 57.69 |
| | | Average: | 71.81<br>($\sigma = 2.28$) | 49.43<br>($\sigma = 1.20$) | 81.61 | 57.26 | **69.43** |

Table 23 shows that the segment-wise accuracy for Youtube subjects is 59.40% while for the Ohio data it is 63.94%. The subject-wise accuracy for Youtube is 66.23% while for Ohio it was 70.93%. The overall subject-wise accuracy was found to be 68.58% which is slightly higher than the baseline achievement of 67.30%. The sparse coding model with input MFCC features is seen to perform the best, compared to other sparse-coding based scenarios, with an overall subject-wise accuracy if 69.43%. It suggests that MFCC features possess good discriminatory capability. Hypothesis testing was also carried out to compare the performance of sparse-coding against those of DBN and Transfer Learning and are reported in a later section.

### 5.1.3 DEEP BELIEF NETWORK MODEL FOR PTSD DIAGNOSIS

In this experiment, the proposed DBN model was applied to three different PTSD feature datasets for PTSD detection using leave-on-subject-out cross validation. These datasets were extracted using the feature extraction procedure described in section 3.3.3. A total of five DBN architectures were used, 162-100-50-2, 2430-100-50-2, 2430-1000-1000-500-2, 2430-500-500-500-500-100-2 and 585-2000-2000-2000-2 some of which are shown in figures 21, 22 and 23. The architectures were selected to be of increasing complexity and also included variation in the type and dimension of input features. The type of features computed for the first four cases is the same as described in section 3 whereas the final scenario considered taking MFCC features as input. MFCC features are used as they have been found to work well in the past. We have speech recordings from 26 PTSD patients, and another set of recordings from 26 control subjects collected from Youtube and an Ohio hospital. To detect presence of PTSD, we utilized the leave-one-subject-out-cross-validation (LOSO-CV) to evaluate the DBN framework. In LOSO-CV, we left one subject for testing and trained a DBN model on the remaining subjects. If the testing accuracy on the testing subject is above 50%, the subject was correctly diagnosed. This procedure was repeated so that each subject was tested once and just once. These experiments were run using the *NVIDIA Tesla K40* GPU.

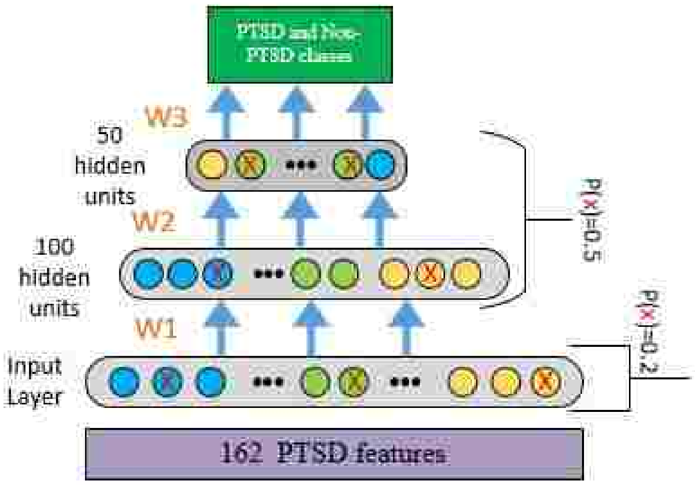

**Figure 21:** DBN network used for PTSD detection was trained with 162 features in the input layer, extracted on PTSD, using leave-one-subject-out-cross validation. The architecture was 162-100-50-2.

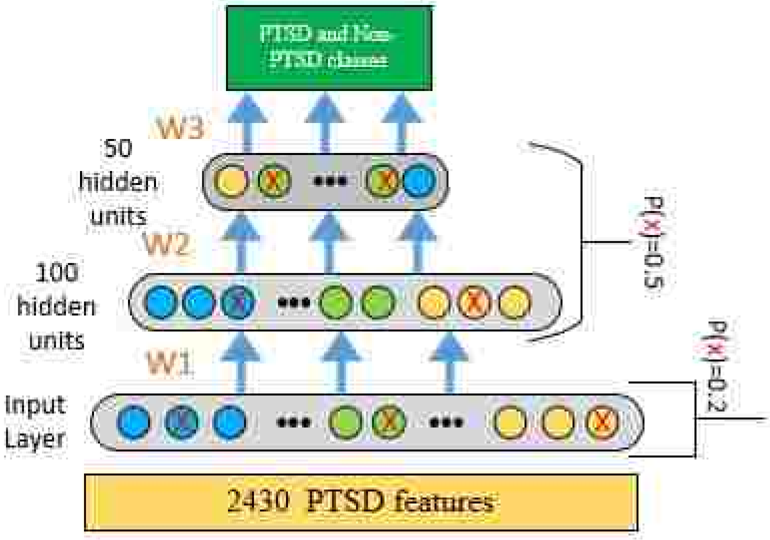

**Figure 22:** DBN network used for PTSD detection was trained with 2430 features extracted on PTSD, in the input layer, using leave-one-subject-out-cross validation. The architecture was 2430-100-50-2.

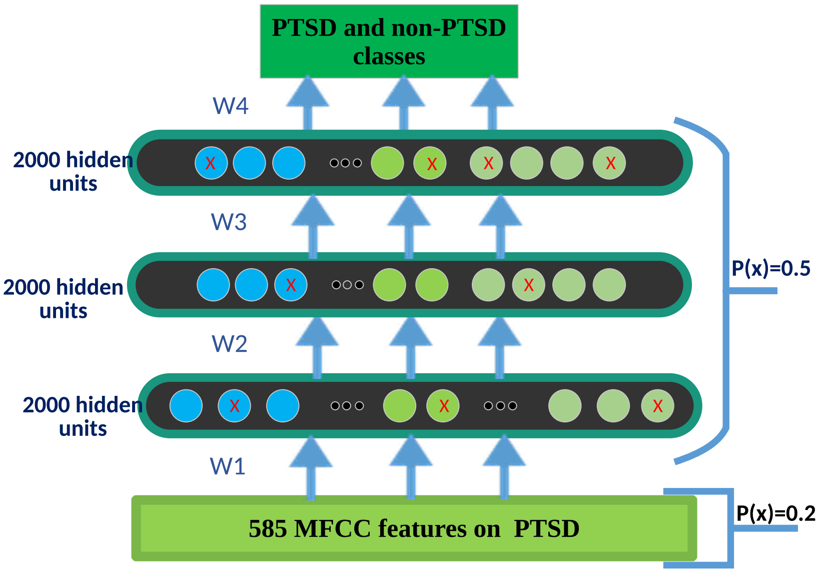

**Figure 23:** DBN network used for PTSD detection was trained with 585 MFCC features extracted on PTSD, in the input layer, using leave-one-subject-out-cross validation. The architecture was 585-2000-2000-2000-2.

#### 5.1.3.1 DEEP BELIEF NETWORK MODEL FOR PTSD DIAGNOSIS BASED ON SINGLE FRAME MULTI-CATEGORY FEATURES

**Table 24:** Classification results of the deep belief network model using leave-one-subject-out cross-validation applied for PTSD diagnosis. The DBN architecture is 162-100-50-2.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26 No of Subjects from Ohio:26 Total:52 subjects | 3.0 | 0.1 | 58.98 ($\sigma = 23.47$) | 56.74 ($\sigma = 27.11$) | 57.69 (15/26) | 53.84 (14/26) | 55.76 |
| | 3.0 | 0.5 | 57.71 ($\sigma = 22.64$) | 55.83 ($\sigma = 27.10$) | 73.07 (19/26) | 57.69 (15/26) | 65.38 |
| | 3.0 | 1.0 | 58.98 ($\sigma = 23.47$) | 56.48 ($\sigma = 27.44$) | 57.69 (15/26) | 57.69 (15/26) | 57.69 |
| | 2.0 | 0.1 | 43.44 ($\sigma = 20.37$) | 55.32 ($\sigma = 27.42$) | 34.61 (9/26) | 53.84 (14/26) | 44.22 |
| | 2.0 | 0.5 | 55.20 ($\sigma = 25.23$) | 54.65 ($\sigma = 27.13$) | 50.00 (13/26) | 53.84 (14/26) | 51.92 |
| | 2.0 | 1.0 | 51.79 ($\sigma = 25.55$) | 56.06 ($\sigma = 26.45$) | 53.84 (14/26) | 53.84 (14/26) | 53.84 |
| | 1.0 | 0.1 | 59.64 ($\sigma = 29.62$) | 54.52 ($\sigma = 25.07$) | 76.92 (20/26) | 61.53 (16/26) | 69.22 |
| | 1.0 | 0.5 | 49.29 ($\sigma = 23.54$) | 54.60 ($\sigma = 25.67$) | 50.00 (13/26) | 57.69 (15/26) | 53.84 |
| | 1.0 | 1.0 | 50.72 ($\sigma = 21.04$) | 53.54 ($\sigma = 26.04$) | 53.84 (14/26) | 61.53 (16/26) | 57.68 |
| Average: | | | 53.97 ($\sigma = 5.54$) | 55.30 ($\sigma = 1.05$) | 56.40 | 56.83 | **56.61** |

Table 24 outlines the results of applying the DBN based model with a simple network for diagnosing patients afflicted with PTSD using single frame features. The architecture used was 162-100-50-2. The mean subject-wise accuracy for Youtube data was 56.40% and for Ohio it was 56.83%. The overall subject-wise accuracy was 56.61%. It is observed from Table 24 that a maximum subject-wise accuracy of 65.38% was achieved, corresponding to three different cases of frame sizes and shifts. The performance marginally exceeds the baseline result of 55.55% by 1.06%.

**Table 25:** Classification results of the deep belief network model using leave-one-subject-out cross-validation applied for PTSD diagnosis. The DBN architecture is 2430-100-50-2.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26<br>No of Subjects from Ohio:26<br>Total:52 subjects | 3.0 | 0.1 | 54.23<br>($\sigma = 25.25$) | 47.33<br>($\sigma = 17.74$) | 53.84<br>(14/26) | 46.15<br>(12/26) | 49.99 |
| | 3.0 | 0.5 | 58.71<br>($\sigma = 22.64$) | 47.78<br>($\sigma = 16.40$) | 73.07<br>(19/26) | 42.30<br>(11/26) | 57.68 |
| | 3.0 | 1.0 | 59.60<br>($\sigma = 23.47$) | 51.23<br>($\sigma = 14.79$) | 57.69<br>(15/26) | 57.69<br>(15/26) | 57.69 |
| | 2.0 | 0.1 | 43.44<br>($\sigma = 20.37$) | 51.41<br>($\sigma = 14.28$) | 34.61<br>(9/26) | 65.38<br>(17/26) | 49.99 |
| | 2.0 | 0.5 | 55.20<br>($\sigma = 25.23$) | 48.07<br>($\sigma = 18.48$) | 50.00<br>(13/26) | 46.15<br>(12/26) | 48.07 |
| | 2.0 | 1.0 | 51.79<br>($\sigma = 25.55$) | 49.76<br>($\sigma = 17.51$) | 53.84<br>(14/26) | 50.00<br>(13/26) | 51.92 |
| | 1.0 | 0.1 | 57.68<br>($\sigma = 22.29$) | 49.44<br>($\sigma = 15.30$) | 61.53<br>(16/26) | 42.30<br>(11/26) | 51.92 |
| | 1.0 | 0.5 | 49.29<br>($\sigma = 23.54$) | 48.45<br>($\sigma = 18.22$) | 50.00<br>(13/26) | 46.15<br>(12/26) | 48.07 |
| | 1.0 | 1.0 | 50.72<br>($\sigma = 21.04$) | 45.08<br>($\sigma = 14.07$) | 53.84<br>(14/26) | 34.61<br>(9/26) | 44.22 |
| | | Average: | 53.40<br>($\sigma = 5.18$) | 48.72<br>($\sigma = 1.99$) | 54.26 | 47.85 | **51.06** |

#### 5.1.3.2 DBN MODEL FOR PTSD DIAGNOSIS BASED ON MULTIPLE FRAME MULTI-CATEGORY FEATURES

Table 25 shows the evaluation of 15-frame features by the DBN network for PTSD detection. The architecture used was 2430-100-50-2. The mean subject-wise accuracy for Youtube data was 54.26%, while for Ohio it was 47.85%. The overall subject-wise accuracy was 51.06%. It is observed that a maximum subject-wise accuracy of 57.69% was achieved, corresponding to three different cases of frame sizes and shifts. The overall subject-wise accuracy was 51.06%. This result is inferior to the case when using the architecture, 162-100-50-2 which achieved 56.61% and is also inferior to the baseline result of 55.55%.

Table 26 shows the evaluation of 15-frame features by the DBN network for PTSD detection using a more complex architecture. The architecture was 2430-1000-1000-500-2. The mean subject-wise accuracy for Youtube data was 64.09%, while for Ohio it was 52.98%. The overall subject-wise accuracy was 58.54%. It is observed that a maximum subject-wise accuracy of 65.38% was achieved, corresponding to two different cases of frame sizes and shifts. This result of the overall subject-wise accuracy is superior to the case when using the architecture, 2430-100-50-2 which achieved 51.06% by a margin of 7.48%. This can possibly be attributed to using multiple frames which may have captured the temporal information more effectively and also the complexity of the network. It is also seen to outperform the baseline using the same feature data set which achieved an overall subject-wise accuracy of 55.55% by a margin of 2.99%.

**Table 26:** Classification results of the deep belief network model using leave-one-subject-out cross-validation applied for PTSD diagnosis. The DBN architecture is 2430-1000-1000-500-2.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26 No of Subjects from Ohio:26 Total:52 subjects | 3.0 | 0.1 | 47.65 ($\sigma = 20.66$) | 50.83 ($\sigma = 16.91$) | 34.61 (9/26) | 50.00 (13/26) | 42.30 |
| | 3.0 | 0.5 | 63.70 ($\sigma = 19.52$) | 52.15 ($\sigma = 16.29$) | 76.92 (20/26) | 53.84 (14/26) | 65.38 |
| | 3.0 | 1.0 | 61.81 ($\sigma = 21.44$) | 53.08 ($\sigma = 16.53$) | 69.23 (18/26) | 61.53 (16/26) | 65.38 |
| | 2.0 | 0.1 | 64.69 ($\sigma = 19.34$) | 48.43 ($\sigma = 13.40$) | 76.92 (20/26) | 42.30 (11/26) | 59.61 |
| | 2.0 | 0.5 | 66.72 ($\sigma = 17.39$) | 46.64 ($\sigma = 15.08$) | 76.92 (20/26) | 46.15 (12/26) | 61.53 |
| | 2.0 | 1.0 | 58.35 ($\sigma = 22.55$) | 53.84 ($\sigma = 14.85$) | 57.69 (15/26) | 65.38 (17/26) | 61.53 |
| | 1.0 | 0.1 | 58.85 ($\sigma = 21.55$) | 47.63 ($\sigma = 15.72$) | 61.53 (16/26) | 46.15 (12/26) | 53.84 |
| | 1.0 | 0.5 | 59.87 ($\sigma = 21.63$) | 50.37 ($\sigma = 16.39$) | 61.53 (16/26) | 53.84 (14/26) | 57.68 |
| | 1.0 | 1.0 | 60.05 ($\sigma = 20.25$) | 52.97 ($\sigma = 16.01$) | 61.53 (16/26) | 57.69 (15/26) | 59.61 |
| | | Average: | 60.18 ($\sigma = 5.48$) | 50.66 ($\sigma = 2.59$) | 64.09 | 52.98 | **58.54** |

Table 27 shows the evaluation of 15-frame features by the DBN network for PTSD detection using a more complex architecture. The architecture was 2430-500-500-500-500-100-2. The mean subject-wise accuracy for Youtube data was 67.09%, while for Ohio it was 61.10%. The overall subject-wise accuracy was 64.09%. The segment-wise accuracy for Youtube was 59.78% while for Ohio it was 60.22%. It is observed that a maximum subject-wise accuracy of 64.09% was achieved, corresponding to three different cases of frame sizes and shifts. Comparing the result from the network 2430-1000-1000-500, which achieved 58.54%, an improvement is noticed by a margin of 5.55%. It is also seen to outperform the baseline that achieved 55.55% by a margin of 8.54%.

**Table 27:** Classification results of the deep belief network model using leave-one-subject-out cross-validation applied for PTSD diagnosis. The DBN architecture is 2430-500-500-500-500-100-2.

| Number of Subjects | Frame Length (seconds) | Frame Shift (seconds) | Segment-wise Acc (%) on Youtube | Segment-wise Acc (%) on Ohio | Subject-wise Acc (%) on Youtube | Subject-wise Acc (%) on Ohio | Overall Subject-wise Accuracy (%) |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26<br>No of Subjects from Ohio:26<br>Total:52 subjects | 3.0 | 0.1 | 61.97<br>($\sigma = 18.99$) | 61.24<br>($\sigma = 29.42$) | 73.07<br>(19/26) | 61.53<br>(16/26) | 67.30 |
| | 3.0 | 0.5 | 60.21<br>($\sigma = 16.15$) | 61.54<br>($\sigma = 30.70$) | 73.07<br>(19/26) | 61.53<br>(16/26) | 67.30 |
| | 3.0 | 1.0 | 57.59<br>($\sigma = 22.51$) | 61.09<br>($\sigma = 30.12$) | 57.69<br>(15/26) | 61.53<br>(16/26) | 59.61 |
| | 2.0 | 0.1 | 61.35<br>($\sigma = 17.92$) | 60.60<br>($\sigma = 29.33$) | 73.07<br>(19/26) | 61.53<br>(16/26) | 67.30 |
| | 2.0 | 0.5 | 63.65<br>($\sigma = 16.32$) | 60.38<br>($\sigma = 29.93$) | 76.92<br>(20/26) | 61.53<br>(16/26) | 69.22 |
| | 2.0 | 1.0 | 60.41<br>($\sigma = 19.01$) | 60.13<br>($\sigma = 29.81$) | 73.07<br>(19/26) | 61.53<br>(16/26) | 67.30 |
| | 1.0 | 0.1 | 57.87<br>($\sigma = 20.22$) | 59.25<br>($\sigma = 28.25$) | 69.23<br>(18/26) | 61.53<br>(16/26) | 65.38 |
| | 1.0 | 0.5 | 55.61<br>($\sigma = 20.45$) | 59.02<br>($\sigma = 28.78$) | 50.00<br>(13/26) | 61.53<br>(16/26) | 55.76 |
| | 1.0 | 1.0 | 59.38<br>($\sigma = 21.57$) | 58.75<br>($\sigma = 28.94$) | 57.69<br>(15/26) | 57.69<br>(15/26) | 57.69 |
| | | Average: | 59.78<br>($\sigma = 2.47$) | 60.22<br>($\sigma = 1.01$) | 67.09 | 61.10 | **64.09** |

#### 5.1.3.3 DBN MODEL FOR PTSD DIAGNOSIS BASED ON MULTIPLE FRAME MFCC FEATURES

Table 28 shows the evaluation of MFCC features by the DBN network for PTSD classification. The architecture used was 585-2000-2000-2000-2. The feature extraction details can be found in the fourth paragraph of section 3.3.3. The mean subject-wise accuracy for Youtube data was 63.92%, while for Ohio it was 57.18%. The overall subject-wise accuracy was 71.79%. It was observed that a maximum subject-wise accuracy of 80.76% was achieved, corresponding to a single case of frame size and shift combination. It is seen to outperform the baseline using the same feature data set which achieved an overall subject-wise accuracy of 55.55%. It lends substantial credence to the effective discrimination capability of the MFCC features for PTSD detection.

**Table 28:** Classification results of the deep belief network model using leave-one-subject-out cross-validation applied for PTSD diagnosis. The DBN architecture is 585-2000-2000-2000-2.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segme nt-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accu-racy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26<br>No of Subjects from Ohio:26<br>Total:52 subjects | 3.0 | 0.1 | 63.46<br>($\sigma = 14.56$) | 56.69<br>($\sigma = 7.94$) | 73.07<br>(19/26) | 69.23<br>(18/26) | 71.15 |
| | 3.0 | 0.5 | 65.70<br>($\sigma = 12.86$) | 57.41<br>($\sigma = 9.30$) | 84.61<br>(22/26) | 76.92<br>(20/26) | 80.76 |
| | 3.0 | 1.0 | 63.02<br>($\sigma = 12.16$) | 57.85<br>($\sigma = 7.34$) | 88.46<br>(23/26) | 53.85<br>(14/26) | 71.15 |
| | 2.0 | 0.1 | 62.20<br>($\sigma = 12.78$) | 57.26<br>($\sigma = 8.19$) | 80.76<br>(21/26) | 65.38<br>(17/26) | 73.07 |
| | 2.0 | 0.5 | 63.70<br>($\sigma = 12.57$) | 55.97<br>($\sigma = 9.18$) | 80.76<br>(21/26) | 50.00<br>(13/26) | 65.38 |
| | 2.0 | 1.0 | 64.47<br>($\sigma = 13.79$) | 56.55<br>($\sigma = 7.53$) | 80.76<br>(21/26) | 53.85<br>(14/26) | 67.30 |
| | 1.0 | 0.1 | 61.63<br>($\sigma = 13.85$) | 58.21<br>($\sigma = 6.74$) | 69.23<br>(18/26) | 69.23<br>(18/26) | 69.23 |
| | 1.0 | 0.5 | 67.02<br>($\sigma = 12.11$) | 57.06<br>($\sigma = 7.49$) | 80.76<br>(21/26) | 65.38<br>(17/26) | 73.07 |
| | 1.0 | 1.0 | 64.12<br>($\sigma = 13.78$) | 57.66<br>($\sigma = 8.17$) | 80.76<br>(21/26) | 69.23<br>(18/26) | 74.99 |
| | | Average: | 63.92<br>($\sigma = 1.67$) | 57.18<br>($\sigma = 0.69$) | 79.90 | 63.67 | **71.79** |

## 5.2 DISCUSSION

Table 29 summarizes the average PTSD diagnostic accuracies achieved across all the baseline, sparse-coding and DBN models. Table 30 summarizes the best PTSD diagnostic accuracies achieved across all the baseline, sparse-coding and DBN models. From Table 29 it is observed that the best overall subject-wise accuracy of 71.79% is achieved by the DBN model using MFCC multiple frame features as input. From Table 30 we can observe that the best overall subject-wise accuracy obtained is 80.76% by the DBN model.

## 5.3 CONCLUSION OF THE PROPOSED APPROACH

The best segment-wise accuracy achieved by SVM was 77.95% for Youtube and 69.78% for Ohio when using multiple-frame, multi-category features. Using the same input features, *sparse coding* achieved a best segment-wise accuracy of 74.95% for Youtube and 67.72% for Ohio. A best segment-wise accuracy of 66.72% for Youtube and 61.54% for Ohio was achieved by the *deep belief network* method.

The subject-wise accuracy using multi-frame, multi-category features by SVM was 55.55% whereas *sparse coding* achieved 68.58%, an increase of 13.03%. Utilizing the same features, the deep belief network model achieved 64.09%, an increase of 8.54% compared to the SVM. A best overall subject-wise accuracy of 71.79% was obtained by the DBN model whereas the sparse-coding model achieved an overall subject-wise accuracy of 76.92%. From Table 30 it is observed that when using DBN for PTSD diagnosis, increasing the network complexity and change in the input features is accompanied by an increase in the subject-wise accuracy from 56.61% to 80.76%. Overall, the *sparse coding* and *deep belief network* models achieved better subject-wise performance than the baseline SVM. Comparison of these models with *transfer learning* by way of hypothesis testing follows in the next chapter.

**Table 29:** Summary of average PTSD diagnostic accuracies across all the baseline and DBN experiments.

| **Method** | **Mean Segment-Wise Acc (%) on Youtube** | **Mean Segment-wise Acc (%) on Ohio** | **Mean Subject-wise Acc (%) on Youtube** | **Mean Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|
| 162 Raw Features (prosodic + excitation + vocal-tract) + SVM | 56.84 ($\sigma = 4.71$) | 52.10 ($\sigma = 5.04$) | 55.98 | 51.27 | **53.62** |
| 2430 Raw Features (prosodic + excitation + vocal-tract) + SVM | 57.05 ($\sigma = 3.77$) | 51.36 ($\sigma = 8.37$) | 57.26 | 53.85 | **55.55** |
| 585 Raw MFCC Features + SVM | 79.10 ($\sigma = 2.16$) | 53.05 ($\sigma = 1.90$) | 88.07 | 56.53 | **67.30** |
| 162 Raw Features (prosodic + excitation + vocal-tract) + Sparse Coding | 67.18 ($\sigma = 2.29$) | 56.00 ($\sigma = 1.06$) | 73.07 | 61.53 | **67.30** |
| 2430 Raw Features (prosodic + excitation + vocal-tract) + Sparse Coding | 59.40 ($\sigma = 2.31$) | 63.94 ($\sigma = 2.21$) | 66.23 | 70.93 | **68.58** |
| 585 Raw MFCC Features + Sparse Coding | 71.81 ($\sigma = 2.28$) | 49.43 ($\sigma = 1.20$) | 81.61 | 57.26 | **69.43** |
| DBN + LOSO-CV (Architecture: 162-100-50-2) | 53.97 ($\sigma = 5.54$) | 55.30 ($\sigma = 1.05$) | 56.40 | 56.83 | **56.61** |
| DBN + LOSO-CV (Architecture: 2430-100-50-2) | 57.21 ($\sigma = 14.49$) | 48.72 ($\sigma = 1.99$) | 54.26 | 47.85 | **51.06** |
| DBN + LOSO-CV (Architecture: 2430-1000-1000-500-2) | 60.18 ($\sigma = 5.48$) | 50.66 ($\sigma = 2.59$) | 64.09 | 52.98 | **58.54** |
| DBN + LOSO-CV (Architecture: 2430-500-500-500-500-100-50-2) | 59.78 ($\sigma = 2.47$) | 60.22 ($\sigma = 1.01$) | 67.09 | 61.10 | **64.09** |
| DBN + LOSO-CV (Architecture: 585-2000-2000-2000-2) | 63.92 ($\sigma = 1.67$) | 57.18 ($\sigma = 0.69$) | 79.90 | 63.67 | **71.79** |

**Table 30:** Summary of best PTSD diagnostic accuracies across the baseline and DBN methods.

| **Method** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|
| 162 Raw Features (prosodic + excitation + vocal-tract) + SVM | 55.86 ($\sigma = 45.36$) | 57.94 ($\sigma = 44.43$) | 61.53 | 61.53 | **61.53** |
| 2430 Raw Features (prosodic + excitation + vocal-tract) + SVM | 60.97 ($\sigma = 47.22$) | 69.78 ($\sigma = 41.25$) | 61.53 | 76.92 | **69.22** |
| 585 Raw MFCC Features + SVM | 77.95 ($\sigma = 23.58$) | 53.40 ($\sigma = 22.20$) | 88.46 | 57.69 | **74.99** |
| 162 Raw Features (prosodic + excitation + vocal-tract) + Sparse Coding | 69.52 ($\sigma = 30.57$) | 56.66 ($\sigma = 24.87$) | 76.92 | 65.38 | **71.15** |
| 2430 Raw Features (prosodic + excitation + vocal-tract) + Sparse Coding | 56.86 ($\sigma = 36.71$) | 67.72 ($\sigma = 26.59$) | 61.53 | 80.76 | **71.14** |
| 585 Raw MFCC Features + Sparse Coding | 72.64 ($\sigma = 19.96$) | 48.88 ($\sigma = 13.80$) | 84.61 | 53.84 | **69.22** |
| DBN + LOSO-CV (Architecture: 162-100-50-2) | 59.64 ($\sigma = 5.54$) | 55.30 ($\sigma = 1.05$) | 56.40 | 56.83 | **56.61** |
| DBN + LOSO-CV (Architecture: 2430-100-50-2) | 59.60 ($\sigma = 23.47$) | 51.23 ($\sigma = 14.79$) | 57.69 | 57.69 | **57.69** |
| DBN + LOSO-CV (Architecture: 2430-1000-1000-500-2) | 61.81 ($\sigma = 21.44$) | 53.08 ($\sigma = 16.53$) | 69.23 | 61.53 | **65.38** |
| DBN + LOSO-CV (Architecture: 2430-500-500-500-500-100-50-2) | 63.65 ($\sigma = 16.32$) | 60.38 ($\sigma = 29.93$) | 76.92 | 61.53 | **69.22** |
| DBN + LOSO-CV (Architecture: 585-2000-2000-2000-2) | 65.70 ($\sigma = 12.86$) | 57.41 ($\sigma = 9.30$) | 84.61 | 76.92 | **80.76** |

# CHAPTER 6

# TRANSFER LEARNING FOR PTSD DIAGNOSIS

## 6.1 DEEP BELIEF NETWORK MODEL FOR PHONE RECOGNITION USING TIMIT

The *deep belief network* model is discussed with regard to phone recognition in this section. The phone recognition is performed as an initial step for a technique known as *transfer learning*. The PTSD data set is too small to train the DBN network efficiently and achieve a very good performance. *Transfer learning* is utilized to solve this problem. It mitigates the small data challenge associated with the PTSD data set. Initially, the DBN network is trained on the extensively large TIMIT speech corpus for phone classification. The TIMIT speech corpus has over 6,300 utterances and has been extensively used for such purposes. The trained model would then be applied for PTSD detection using transfer learning. With this objective, a DBN model was trained using several types of TIMIT data sets, to explore and compare their performance. Table 32 shows the results of training the DBN network using the TIMIT speech database. It was intended to move from simpler to more complex configurations of the network to see if it resulted in higher phone classification performance. The first part of this section discusses the aspects of DBN-TIMIT phone recognition and latter part describes the *transfer learning* details.

In the first experiment, we trained a DBN model using the TIMIT speech corpus. In this experiment, the TIMIT speech features were extracted as follows. The speech signal was first pre-emphasized using a first order FIR filter. Then the speech signal was divided into a set of frames of length 25ms with an overlap of 10ms between two consecutive frames. Speech frames which overlapped between two phones were deleted. The same features as shown in Table 3 were extracted from the TIMIT database. They comprised of a combination of prosodic, vocal-tract and excitation features typically used in speech recognition. A total of 54 raw features along with their first and second order temporal derivatives were combined to form a total feature vector with a length of 162 features. There were 39 phone classes in this dataset. The training data set comprised of almost 439,000 data points and approximately 161,000 samples for the test data set. The output phone classification is carried out using the logistic regression classifier.

In all of the experiments, the following procedure was followed in order to train the DBN network. We first pre-trained the structure layer by layer utilizing the restricted Boltzmann machine [79], using raw features as input. The first hidden layer containing multiple hidden units was trained with the dropout technique [80] and weights were stored. Once the first hidden layer was trained, the outputs of the first layer hidden layer were used as inputs for the second hidden layer and were trained again by the restricted Boltzmann machine. Using this principle, the deep structure can be built up to any number of layers. By following the work in [80], we built a deep structure in the pre-training step. In the fine-tuning step, we attached the class labels to the training data set and we added a single logistic regression layer on top of the deep structure for phone classification. The dropout technique was also used in fine-tuning. Once the deep structure was trained, the output before the last layer was used as a new representation for the original raw input features and was used to train a SVM classifier for classification. We used a large-scale linear SVM for training. The computation was performed using a *NVIDIA K40 Tesla* GPU.

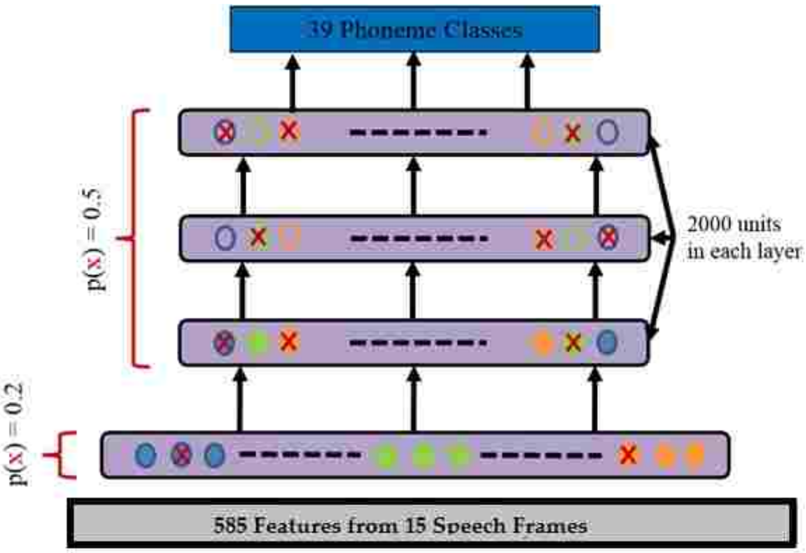

**Figure 24:** DBN network trained with 585 MFCC features using TIMIT in the input layer. 15 speech frames are combined together, each having a dimension 39 to form 585 features. The architecture is 585-2000-2000-2000-39.

**Table 31:** Details of the training and test feature data sets extacted from TIMIT.

| Type of Input | No.of Features | No.of Data Points | No.of Classes |
|---|---|---|---|
| Entire training set | 585 | ~ 1.056M | 39 |
| Development data set | 585 | ~ 114K | 39 |
| Core testing data set | 585 | ~ 54.5K | 39 |

**Table 32:** Classification results of applying different architectures of the proposed DBN model on the TIMIT speech corpus.

| **Architecture Used** | **Training data set** | **Testing data set** | **No of epochs (pre-train/fine-tune)** | **Training Accuracy (%)** | **Testing Accuracy (%)** |
|---|---|---|---|---|---|
| 162-100-50-39 | Entire Training set | Entire test set | 100/200 | 44.55 | 43.63 |
| 2430-100-50-39 | Entire Training set | Entire test set | 100/200 | 39.12 | 38.50 |
| 2430-1000-1000-500-39 | Entire Training set | Entire test set | 100/200 | 49.18 | 47.44 |
| 2430-500-500-500-500-100-39 | Entire Training set | Entire test set | 100/200 | 26.48 | 26.53 |

Network configurations similar to those used in the case of applying the deep belief network directly on PTSD data were used, to be eventually used for applying *transfer learning* for comparing performance. In the first experiment, we used a simple configuration of 162-100-50-39. Related feature extraction details are mentioned in section 3.3.2. It did not result in good phone recognition. From Table 32 it is observed that a test accuracy of 43.63% was achieved. A second experiment was performed using the same network configuration but with the TIMIT speech feature dataset having 2,430 features, obtained by concatenating 15 frames together. In the third experiment, we used a more complex network configuration of 2430-1000-1000-500-39. It achieved the best test accuracy of 47.44%. In the fourth experiment, we trained a deeper DBN model using TIMIT with the following network configuration, 2430-500-500-500-500-100-39. The classification was observed to be 26.53%. A deeper network did not necessarily imply a better classification performance.

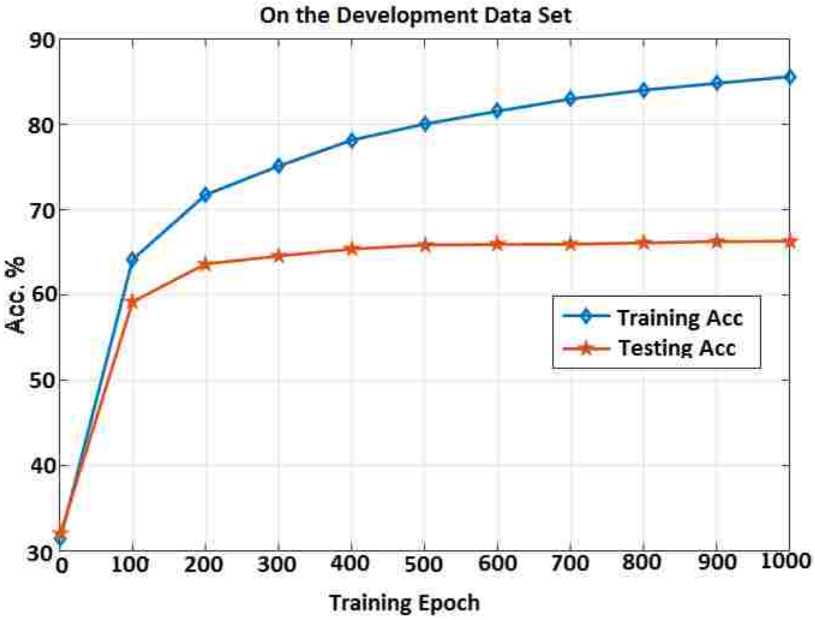

**Figure 25:** Classification results if the DBN network was trained using the TIMT *development* set for phone classification.

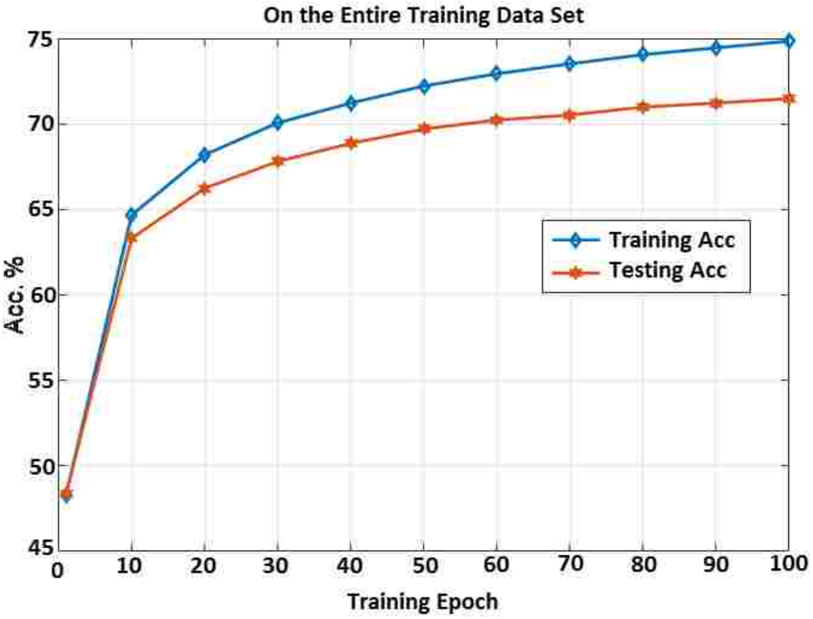

**Figure 26:** Classification results if the DBN network was trained using the *entire* TIMIT training set.

In the fourth experiment, MFCC features were used in the input layer since they have been found to work very well in phone recognition in the past. A configuration of 585-2000-2000-2000-39 was used as shown in Figure 24. The first layer is the input layer consisting of 585 MFCC features and the final layer is the output layer corresponding to 39 phone labels. For this particular scenario of using MFCC features as input, the configuration was also run with two different types of training data sets. The *development* set was of smaller size than the *entire* training set, while the test sets for both were identical. Figures 25 and 26 show the difference in results between training using the *entire* training set and the *development* set.

**Table 33:** Classification results of applying the deep belief network model on TIMIT with MFCC features as input. The DBN architecture is 585-2000-2000-2000-39.

| **Training data set** | **Testing data set** | **No. of epochs (pre-train/finetune)** | **Training Accuracy(%)** | **Testing Accuracy(%)** | **CPU time(hours)** |
|---|---|---|---|---|---|
| Development set | Core test set | 100/100 | 85.60 | 66.31 | $\sim 20$ |
| Entire training set | Core test set | 100/100 | 74.88 | 71.51 | $\sim 48$ |

The results of using these two different types of training sets for training the DBN network are shown in Table 33. It is observed that using the *entire* training set resulted in a classification performance gain of 5.2% compared to that of using the *development* set with an overall test accuracy of 71.51%. Since a larger training set gave favorable results, it was decided to use the *entire* TIMIT training set to train the DBN network. Table 31 shows the relevant details associated with these data sets.

## 6.2 TRANSFER LEARNING FOR PTSD DIAGNOSIS

### 6.2.1 TRANSFER LEARNING FOR PTSD DIAGNOSIS USING SINGLE FRAME MUTLI-CATEGORY FEATURES

Six experiments based on *transfer learning* have been conducted with the goal of diagnosing PTSD. These experiments differed in three respects: 1) type of input features 2) the input feature dimensionality and 3) the depth of the network architectures used. A shallow network with a single frame feature as input was used in the beginning while subsequent experiments were carried out using multiple frame features and deeper networks. The first four experiments were carried out using the features described in Table 3 while the last two experiments used MFCC features computed on PTSD as input described in section 3.3.3 previously. Figure 27 demonstrates the concept of transfer learning.

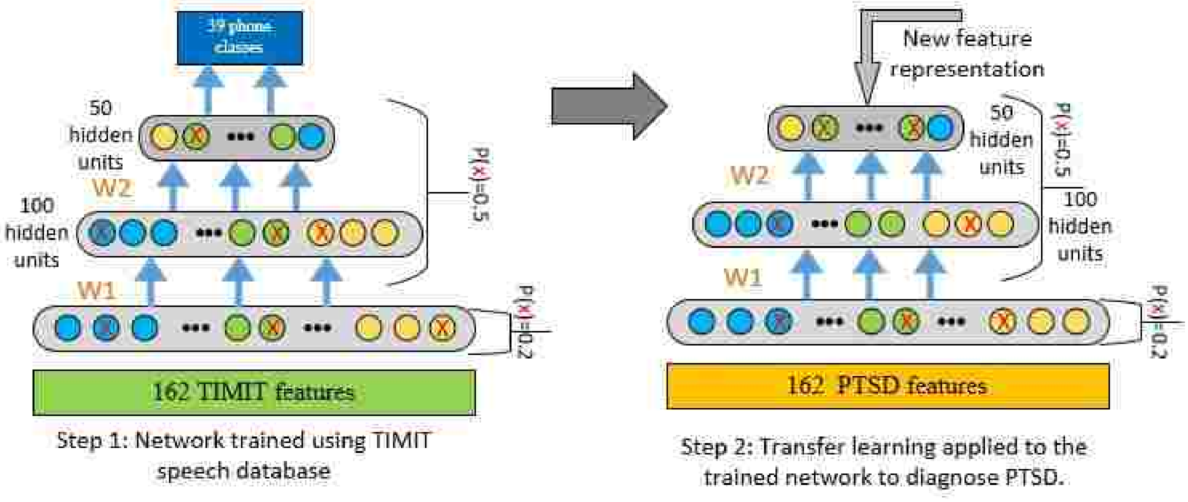

**Figure 27:** Concept of *transfer learning.*

Initially the DBN network was trained on the TIMIT data set for phone recognition. Generative pre-training was carried out using the DBN model where the label information was not used to build the DBN model greedily, layer by layer first. The pretrained model was then discriminatively fine-tuned, using the label information by using a logistic regression classifier which was built on top of the deep structure. Once the deep structure was trained, the outputs before the last layer were used as new representations for the original raw features and were used to train a SVM classifier for classifying output phone labels. The architecture of the DBN network is 162-100-50-39.

The first layer is the input layer consisting of 162 features. The first hidden layer contains 100 hidden units which was trained using dropout and the final hidden layer has a total of 50 hidden units. The output label layer contains a total of 39 class labels. The dropout technique was used in fine-tuning with a dropout probability of 0.2 for input layer and 0.5 for all hidden layers. Results were shown in Table 32.

After the DBN network was trained on the TIMIT data set, the knowledge or the model was applied and transferred to several different PTSD datasets for classifying between PTSD and non-PTSD patients using the leave-one-subject-out cross-validation. The concept of transfer learning depicted in Figure 27, was applied in the following manner. First, single frame PTSD feature datasets having 162 dimensions each, were extracted as described in section 3.3.3. The label layer was removed resulting in a network configuration of 162-100-50. Then raw PTSD features were used to make a forward pass through the network utilizing the previously trained, fine-tuned weights by the TIMIT database. Both the first layer of 100 features and the final layer are selected for classification. Utilizing these features, a linear-kernel SVM classifier was trained for classification whose parameters were found by performing gridsearch. In LOSO-CV, we left one subject for testing and used the remaining subjects to train the SVM. If the testing accuracy on the testing subject is above 50%, the subject was considered to be correctly diagnosed. This procedure was repeated so that each subject was tested once and just once. In this experiment, all hidden layers are also used for classification to compare performance.

Tables 34 and 35 outline the results of the first experiment where the proposed transfer learning framework is applied to the PTSD datasets. The network configuration is 162-100-50. Different sets of PTSD features have been sampled from the available recordings using variable speech frame sizes and frame shifts as shown.

**Table 34:** Classification results using the first hidden layer of 100 features obtained after applying *transfer learning* and applying leave-one-subject-out cross-validation. The transfer learning architecture is 162-100-50.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26<br>No of Subjects from Ohio:26<br>Total:52 subjects | 3.0 | 0.1 | 78.93<br>($\sigma = 28.46$) | 54.02<br>($\sigma = 28.72$) | 88.46<br>(23/26) | 42.30<br>(11/26) | 65.38 |
| | 3.0 | 0.5 | 78.92<br>($\sigma = 29.00$) | 53.62<br>($\sigma = 28.82$) | 88.46<br>(23/26) | 42.30<br>(11/26) | 65.38 |
| | 3.0 | 1.0 | 74.98<br>($\sigma = 34.05$) | 60.62<br>($\sigma = 25.66$) | 73.07<br>(19/26) | 61.53<br>(16/26) | 67.30 |
| | 2.0 | 0.1 | 77.91<br>($\sigma = 27.53$) | 53.63<br>($\sigma = 26.65$) | 84.61<br>(22/26) | 38.46<br>(10/26) | 61.53 |
| | 2.0 | 0.5 | 77.49<br>($\sigma = 27.95$) | 53.22<br>($\sigma = 26.82$) | 88.46<br>(23/26) | 38.46<br>(10/26) | 63.46 |
| | 2.0 | 1.0 | 74.79<br>($\sigma = 33.40$) | 58.16<br>($\sigma = 23.25$) | 76.92<br>(20/26) | 57.69<br>(15/26) | 67.30 |
| | 1.0 | 0.1 | 76.15<br>($\sigma = 25.51$) | 53.27<br>($\sigma = 22.89$) | 92.30<br>(24/26) | 38.46<br>(10/26) | 65.38 |
| | 1.0 | 0.5 | 75.85<br>($\sigma = 25.73$) | 53.19<br>($\sigma = 22.83$) | 92.30<br>(24/26) | 34.61<br>(9/26) | 63.45 |
| | 1.0 | 1.0 | 72.82<br>($\sigma = 30.79$) | 54.59<br>($\sigma = 20.57$) | 80.76<br>(21/26) | 50.00<br>(13/26) | 65.38 |
| | | Average: | 76.42<br>($\sigma = 2.06$) | 54.92<br>($\sigma = 2.64$) | 85.03 | 44.86 | **64.95** |

For the case in which the first hidden layer of 100 features, is used for classification, it is observed from Table 34 that the overall subject-wise test accuracy for Youtube data is 85.03% and for Ohio data is 44.86%. The average segment-wise accuracy for Youtube data is 76.42% with a standard deviation of 2.06% and for the Ohio data, 54.92% with a standard deviation of 2.64%. The overall subject-wise accuracy is 64.95%.

For the case in which the final hidden layer of 50 features, is used for classification,

**Table 35:** Classification results using the final hidden layer of 50 features obtained using *transfer learning* and applying leave-one-subject-out cross-validation. The network configuration for *transfer learning* is 162-100-50.

| Number of Subjects | Frame Length (seconds) | Frame Shift (seconds) | Segment-wise Acc (%) on Youtube | Segment-wise Acc (%) on Ohio | Subject-wise Acc (%) on Youtube | Subject-wise Acc (%) on Ohio | Overall Subject-wise Accuracy (%) |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26<br>No of Subjects from Ohio:26<br>Total:52 subjects | 3.0 | 0.1 | 75.81 ($\sigma = 27.07$) | 53.08 ($\sigma = 28.04$) | 84.61 (22/26) | 42.30 (11/26) | 63.45 |
| | 3.0 | 0.5 | 75.36 ($\sigma = 28.02$) | 52.84 ($\sigma = 28.10$) | 80.76 (21/26) | 42.30 (11/26) | 61.53 |
| | 3.0 | 1.0 | 72.64 ($\sigma = 33.08$) | 56.23 ($\sigma = 23.17$) | 76.92 (20/26) | 57.69 (15/26) | 67.30 |
| | 2.0 | 0.1 | 75.70 ($\sigma = 26.33$) | 53.00 ($\sigma = 26.34$) | 88.46 (23/26) | 46.15 (12/26) | 67.30 |
| | 2.0 | 0.5 | 75.24 ($\sigma = 26.80$) | 52.66 ($\sigma = 26.37$) | 88.46 (23/26) | 42.30 (11/26) | 65.38 |
| | 2.0 | 1.0 | 72.32 ($\sigma = 31.46$) | 54.50 ($\sigma = 21.37$) | 80.76 (21/26) | 57.69 (15/26) | 69.22 |
| | 1.0 | 0.1 | 73.52 ($\sigma = 23.91$) | 52.37 ($\sigma = 22.63$) | 88.46 (23/26) | 42.30 (11/26) | 65.38 |
| | 1.0 | 0.5 | 73.36 ($\sigma = 23.59$) | 52.09 ($\sigma = 22.72$) | 88.46 (23/26) | 46.15 (12/26) | 67.30 |
| | 1.0 | 1.0 | 70.02 ($\sigma = 28.12$) | 52.32 ($\sigma = 19.55$) | 84.61 (22/26) | 50.00 (13/26) | 67.30 |
| | | Average: | 73.77 ($\sigma = 1.94$) | 53.23 ($\sigma = 1.32$) | 84.61 | 47.43 | **66.02** |

it is observed from Table 34 that the overall subject-wise test accuracy for Youtube data is 84.61% and for Ohio data is 47.43%. The average segment-wise accuracy for Youtube data is 73.77% with a standard deviation of 1.94% and for the Ohio data, 53.23% with a standard deviation of 1.32%. The overall subject-wise accuracy is 66.02%. Compared to the baseline which achieved 53.62%, the performance improved by a margin of 12.4%. It is also observed that, when compared to the case for transfer learning using 100 features, there is a marginal improvement in performance for this layer.

### 6.2.2 TRANSFER LEARNING FOR PTSD DIAGNOSIS USING MUTLIPLE FRAME MULTI-CATEGORY FEATURES

In the second experiment, we trained a simple DBN model using 15 frame features derived from TIMIT as input. Transfer learning was then applied having the architecture, 2430-100-50 with input PTSD features. In this experiment, the multiple frame TIMIT speech features were extracted as described in the second paragraph of section 3.3.2.

**Table 36:** Classification results utilizing the first hidden layer of 100 features obtained by applying *transfer learning* and applying leave-one-subject-out cross-validation. The architecture is 2430-100-50.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26 No of Subjects from Ohio:26 Total:52 subjects | 3.0 | 0.1 | 72.04 ($\sigma = 35.63$) | 59.48 ($\sigma = 27.72$) | 76.92 (20/26) | 57.69 (15/26) | 67.30 |
| | 3.0 | 0.5 | 73.81 ($\sigma = 37.08$) | 62.98 ($\sigma = 30.80$) | 76.92 (20/26) | 69.23 (18/26) | 73.07 |
| | 3.0 | 1.0 | 77.04 ($\sigma = 35.97$) | 64.93 ($\sigma = 33.71$) | 80.76 (21/26) | 69.23 (18/26) | 74.99 |
| | 2.0 | 0.1 | 71.98 ($\sigma = 34.72$) | 58.45 ($\sigma = 26.21$) | 76.92 (20/26) | 61.53 (16/26) | 69.22 |
| | 2.0 | 0.5 | 73.82 ($\sigma = 36.64$) | 62.32 ($\sigma = 30.97$) | 76.92 (20/26) | 61.53 (16/26) | 69.22 |
| | 2.0 | 1.0 | 77.77 ($\sigma = 35.47$) | 64.32 ($\sigma = 33.91$) | 76.92 (20/26) | 69.23 (18/26) | 73.07 |
| | 1.0 | 0.1 | 71.74 ($\sigma = 33.22$) | 57.14 ($\sigma = 24.27$) | 76.92 (20/26) | 57.69 (15/26) | 67.30 |
| | 1.0 | 0.5 | 73.56 ($\sigma = 36.63$) | 61.33 ($\sigma = 30.95$) | 76.92 (20/26) | 65.38 (17/26) | 71.15 |
| | 1.0 | 1.0 | 76.32 ($\sigma = 35.74$) | 62.40 ($\sigma = 34.06$) | 76.92 (20/26) | 53.84 (14/26) | 65.38 |
| | | Average: | 74.23 ($\sigma = 2.28$) | 61.48 ($\sigma = 2.64$) | 77.34 | 62.81 | **70.08** |

In the third experiment, the complexity of the network was increased. A deeper DBN model was trained on the TIMIT speech corpus using 15 frame features. Multiple frame TIMIT speech features were extracted as described in the second paragraph of section 3.3.2. Subsequently, transfer learning was applied to PTSD input features, whose architecture was 2430-1000-1000-500.

The pre-training and fine tuning was carried out similar to that described in section 4.1.3.1. Generative pre-training was carried out using the DBN model where the label information was not used to build the DBN model greedily, layer by layer first. The pre-trained model was then discriminatively fine-tuned, using the label information by using a logistic regression classifier which is built on top of the deep structure. The dropout technique was used in both pre-training and fine-tuning with a dropout probability of 0.2 for input layer and 0.5 for all hidden layers. Once the deep structure was trained, the outputs before the last layer were used as new representations for the original raw features and were used to train a SVM classifier for classifying output phone labels.

Once the DBN network was trained using the TIMIT speech corpus, *transfer learning* was

**Table 37:** Classification results utilizing the second hidden layer of 50 features obtained by applying *transfer learning* and applying leave-one-subject-out cross-validation. The architecture is 2430-100-50.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26 No of Subjects from Ohio:26 Total:52 subjects | 3.0 | 0.1 | 69.59<br>($\sigma = 31.41$) | 57.97<br>($\sigma = 20.47$) | 69.23<br>(18/26) | 69.23<br>(18/26) | 69.23 |
| | 3.0 | 0.5 | 62.82<br>($\sigma = 33.64$) | 60.52<br>($\sigma = 22.51$) | 69.23<br>(18/26) | 69.23<br>(18/26) | 69.23 |
| | 3.0 | 1.0 | 69.01<br>($\sigma = 37.08$) | 63.11<br>($\sigma = 27.45$) | 69.23<br>(18/26) | 69.23<br>(18/26) | 69.23 |
| | 2.0 | 0.1 | 69.76<br>($\sigma = 30.32$) | 56.92<br>($\sigma = 19.68$) | 76.92<br>(20/26) | 65.38<br>(17/26) | 71.15 |
| | 2.0 | 0.5 | 69.98<br>($\sigma = 33.06$) | 60.13<br>($\sigma = 22.34$) | 69.23<br>(18/26) | 69.23<br>(18/26) | 69.23 |
| | 2.0 | 1.0 | 68.70<br>($\sigma = 36.04$) | 64.45<br>($\sigma = 27.31$) | 69.23<br>(18/26) | 69.23<br>(18/26) | 69.23 |
| | 1.0 | 0.1 | 69.86<br>($\sigma = 29.30$) | 54.51<br>($\sigma = 17.85$) | 80.76<br>(21/26) | 65.38<br>(17/26) | 73.07 |
| | 1.0 | 0.5 | 70.72<br>($\sigma = 32.46$) | 59.14<br>($\sigma = 22.86$) | 69.23<br>(18/26) | 65.38<br>(17/26) | 67.30 |
| | 1.0 | 1.0 | 69.64<br>($\sigma = 35.83$) | 64.08<br>($\sigma = 28.15$) | 69.23<br>(18/26) | 69.23<br>(18/26) | 69.23 |
| | | Average: | 68.89<br>($\sigma = 2.35$) | 60.09<br>($\sigma = 3.36$) | 71.36 | 67.94 | **69.65** |

applied to the same network, where the input data were features extracted from the PTSD data set. The type of features were identical to those described in section 3.3.3 and shown in Table 3. The input data is the first layer of the network, consisting of 2,430 features. The concept of transfer learning is depicted in Figure 28. The first and second hidden layers contain 1000 hidden units which were trained using dropout and the final hidden layer has a total of 500 hidden units. The output label layer contains a total of 39 class labels. *Transfer learning* was applied in the same way as described in the previous section. In this experiment, all outputs were used for classification to compare performance.

For the case in which the first hidden layer of 100 features, is used for classification, it is observed from Table 37 that the overall subject-wise test accuracy for Youtube data is 71.36% and for Ohio data is 67.94%. The average segment-wise accuracy for Youtube data is 68.89% with a standard deviation of 2.35% and for the Ohio data, 60.09% with a standard deviation of 3.36%. The overall subject-wise accuracy is 69.65%. Statistical tests of significance between transfer learning and SVM are presented and discussed towards the end of this section.

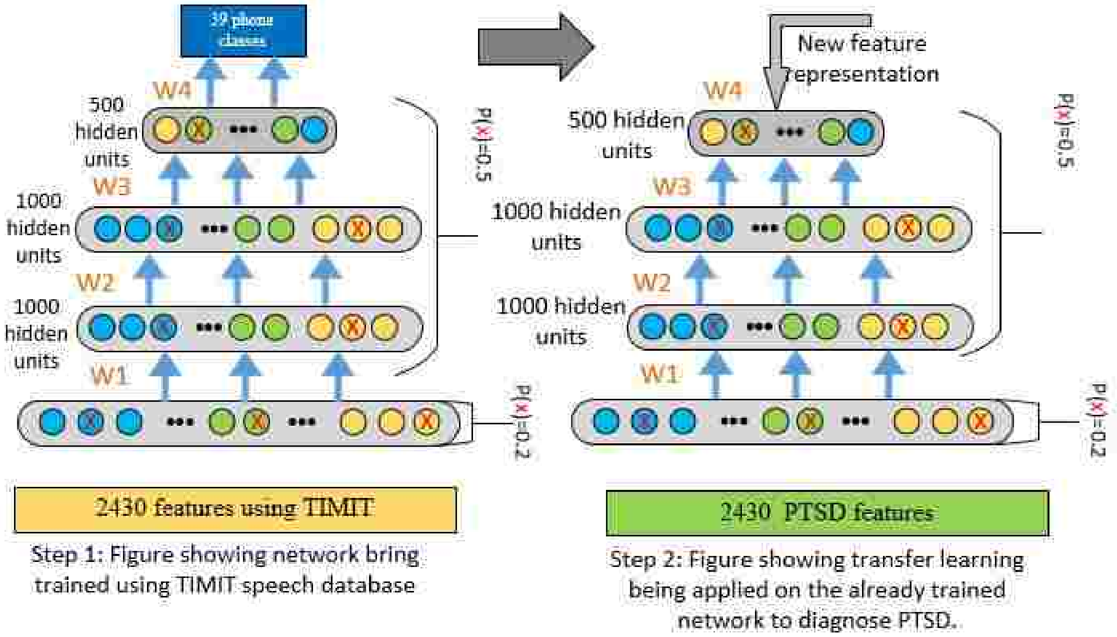
39 phone classes
W4
500 hidden units
W3
1000 hidden units
W2
1000 hidden units
W1
P(x)=0.5
P(x)=0.2
2430 features using TIMIT
Step 1: Figure showing network bring trained using TIMIT speech database
New feature representation
W4
500 hidden units
W3
1000 hidden units
W2
1000 hidden units
W1
P(x)=0.5
P(x)=0.2
2430 PTSD features
Step 2: Figure showing transfer learning being applied on the already trained network to diagnose PTSD.

**Figure 28:** Concept of transfer learning and the network architectures used for training with TIMIT and for PTSD diagnosis. The input layer has 2430 features.

For the case in which the first hidden layer of 1000 features is used for classification, it is observed from Table 38 that the overall subject-wise test accuracy for Youtube data is 78.20% and for Ohio data is 62.81%. The average segment-wise accuracy for Youtube data is 76.46% with a standard deviation of 2.12%. This is the highest achieved segment-wise accuracy among all the three layers. For the Ohio data, 61.02% with a standard deviation of 2.28%. The overall subject-wise accuracy is 70.50%. This is the highest accuracy, compared to the performance by the other two layers and the baseline SVM. Compared to the baseline there is an improvement in performance by a margin of 14.90%.

Classification, performed using the second hidden layer of 1000 features, produces the results shown in Table 39. The overall subject-wise test accuracy for Youtube data is 77.34% and for Ohio data is 59.39%. The average segment-wise accuracy for Youtube data is 73.48% with a standard deviation of 1.68% and for the Ohio data, 57.41% with a standard deviation of 1.16%. An overall subject-wise accuracy of 68.37% is achieved which is higher than the baseline achievement of 55.55% by a margin of 12.82%. The result is inferior when compared to the first layer achievement of 70.50%.

Classification, performed using the final hidden layer of 500 features, produces the results shown in Table 40. The overall subject-wise test accuracy for Youtube data is 78.19% and for Ohio data is 60.67%. The average segment-wise accuracy for Youtube data is 71.56% with a standard deviation of 1.88% and for the Ohio data, 58.21% with a standard deviation of 1.00%. An overall subject-wise accuracy of 69.43% is achieved which is higher than the baseline achievement of 55.55% by a margin of 13.88%. The result is inferior when compared to the best

**Table 38:** Classification results utilizing the first hidden layer of 1000 features obtained by applying *transfer learning* and applying leave-one-subject-out cross-validation. The architecture is 2430-1000-1000-500.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26<br>No of Subjects from Ohio:26<br>Total:52 subjects | 3.0 | 0.1 | 74.87 ($\sigma = 35.22$) | 58.67 ($\sigma = 29.46$) | 76.92 (20/26) | 53.84 (14/26) | 65.38 |
| | 3.0 | 0.5 | 75.71 ($\sigma = 37.89$) | 62.96 ($\sigma = 32.09$) | 76.92 (20/26) | 61.53 (16/26) | 69.22 |
| | 3.0 | 1.0 | 78.92 ($\sigma = 36.70$) | 63.13 ($\sigma = 33.14$) | 80.76 (21/26) | 65.38 (17/26) | 73.07 |
| | 2.0 | 0.1 | 74.65 ($\sigma = 34.41$) | 58.20 ($\sigma = 28.07$) | 76.92 (20/26) | 57.69 (15/26) | 67.30 |
| | 2.0 | 0.5 | 75.83 ($\sigma = 37.64$) | 62.40 ($\sigma = 31.87$) | 76.92 (20/26) | 61.53 (16/26) | 69.22 |
| | 2.0 | 1.0 | 79.50 ($\sigma = 36.36$) | 63.05 ($\sigma = 32.73$) | 80.76 (21/26) | 69.23 (18/26) | 74.99 |
| | 1.0 | 0.1 | 73.94 ($\sigma = 33.74$) | 57.82 ($\sigma = 26.69$) | 76.92 (20/26) | 61.53 (16/26) | 69.22 |
| | 1.0 | 0.5 | 75.63 ($\sigma = 37.39$) | 60.23 ($\sigma = 31.69$) | 76.92 (20/26) | 65.38 (17/26) | 71.15 |
| | 1.0 | 1.0 | 79.11 ($\sigma = 36.14$) | 62.79 ($\sigma = 33.37$) | 80.76 (21/26) | 69.23 (18/26) | 74.99 |
| | | Average: | 76.46 ($\sigma = 2.12$) | 61.02 ($\sigma = 2.28$) | 78.20 | 62.81 | **70.50** |

subject-wise accuracy of 70.50% achieved by the first layer. Statistical significance tests are discussed in the latter part of this section.

The fourth experiment is identical to the second experiment in methodology, except that the network configuration was increased in depth to five hidden layers and the transfer learning network architecture was 2430-500-500-500-500-100.

The results of applying *transfer learning* using a deeper network architecture of 2430-500-500-500-500-100 are shown in Table 41. In this scenario, 2,430 is the feature dimensionality of the input layer. When the first hidden layer of 500 features are used for classification, it is observed from Table 41 that the overall subject-wise test accuracy for Youtube data is 78.62% and for Ohio data is 70.08%. The average segment-wise accuracy for Youtube data is 66.94% with a standard deviation of 3.94% and for the Ohio data, 57.50% with a standard deviation of 2.89%. The overall subject-wise accuracy is 74.35% which outperforms the SVM baseline result of 55.55% by a margin of 18.80%.

Using the second hidden layer of 500 features for classification, it is observed from Table 45 that the overall subject-wise test accuracy for Youtube data is 76.49% and for Ohio data is 61.53%. The average segment-wise accuracy for Youtube data is 73.05% with a standard deviation of 1.24% and for the Ohio data, 59.19% with a standard deviation of 2.03%. The overall subject-wise accuracy is 69.01% which also outperforms the baseline SVM result of 55.55%.

For the case in which the third hidden layer of 500 features is used for classification, Table 43 shows that the overall subject-wise test accuracy for Youtube data is 75.63% and for Ohio data is 66.23%. The average segment-wise accuracy for Youtube data is 70.50% with a standard deviation of 1.02% and for the Ohio data, 60.34% with a standard deviation of 1.83%. The overall subject-wise accuracy is 70.93% which also outperforms the SVM baseline result of 55.55%.

**Table 39:** Classification results using the second hidden layer of 1000 features and applying leave-one-subject-out cross-validation. The 1000 features are obtained by applying *transfer learning* to the PTSD data sets. The architecture is 2430-1000-1000-500.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26<br>No of Subjects from Ohio:26<br>Total:52 subjects | 3.0 | 0.1 | 74.12 ($\sigma = 35.37$) | 57.91 ($\sigma = 30.03$) | 80.76 (21/26) | 61.53 (16/26) | 71.14 |
| | 3.0 | 0.5 | 74.98 ($\sigma = 37.14$) | 58.92 ($\sigma = 33.25$) | 80.76 (21/26) | 61.53 (16/26) | 71.14 |
| | 3.0 | 1.0 | 71.23 ($\sigma = 39.66$) | 55.94 ($\sigma = 34.86$) | 69.23 (18/26) | 53.84 (14/26) | 61.53 |
| | 2.0 | 0.1 | 73.78 ($\sigma = 34.49$) | 57.57 ($\sigma = 28.31$) | 76.92 (20/26) | 61.53 (16/26) | 69.22 |
| | 2.0 | 0.5 | 75.72 ($\sigma = 36.59$) | 59.04 ($\sigma = 32.89$) | 80.76 (21/26) | 61.53 (16/26) | 71.14 |
| | 2.0 | 1.0 | 72.04 ($\sigma = 38.03$) | 57.43 ($\sigma = 33.85$) | 76.92 (20/26) | 57.69 (15/26) | 67.30 |
| | 1.0 | 0.1 | 72.95 ($\sigma = 33.39$) | 56.24 ($\sigma = 25.83$) | 76.92 (20/26) | 57.69 (15/26) | 67.30 |
| | 1.0 | 0.5 | 75.18 ($\sigma = 36.64$) | 57.73 ($\sigma = 32.37$) | 80.76 (21/26) | 61.53 (16/26) | 71.14 |
| | 1.0 | 1.0 | 71.32 ($\sigma = 38.83$) | 55.96 ($\sigma = 32.79$) | 73.07 (19/26) | 57.69 (15/26) | 65.38 |
| | | Average: | 73.48 ($\sigma = 1.68$) | 57.41 ($\sigma = 1.16$) | 77.34 | 59.39 | **68.37** |

**Table 40:** Classification results using the final hidden layer of 500 features and applying leave-one-subject-out cross-validation. The 500 features are obtained by applying *transfer learning* to the PTSD data sets. The architecture is 2430-1000-1000-500.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26 No of Subjects from Ohio:26 Total:52 subjects | 3.0 | 0.1 | 71.83<br>($\sigma = 35.20$) | 58.64<br>($\sigma = 28.77$) | 80.76<br>(21/26) | 61.53<br>(16/26) | 71.14 |
| | 3.0 | 0.5 | 73.69<br>($\sigma = 36.86$) | 59.49<br>($\sigma = 32.58$) | 80.76<br>(21/26) | 65.38<br>(17/26) | 73.07 |
| | 3.0 | 1.0 | 68.92<br>($\sigma = 37.74$) | 58.77<br>($\sigma = 33.07$) | 69.23<br>(18/26) | 61.53<br>(16/26) | 65.38 |
| | 2.0 | 0.1 | 71.24<br>($\sigma = 34.53$) | 57.79<br>($\sigma = 26.82$) | 80.76<br>(21/26) | 57.69<br>(16/26) | 69.22 |
| | 2.0 | 0.5 | 74.13<br>($\sigma = 36.56$) | 59.23<br>($\sigma = 32.07$) | 80.76<br>(21/26) | 61.53<br>(16/26) | 71.14 |
| | 2.0 | 1.0 | 69.30<br>($\sigma = 37.09$) | 57.95<br>($\sigma = 31.96$) | 76.92<br>(20/26) | 57.69<br>(15/26) | 67.30 |
| | 1.0 | 0.1 | 70.18<br>($\sigma = 33.50$) | 56.29<br>($\sigma = 24.27$) | 76.92<br>(20/26) | 57.69<br>(15/26) | 67.30 |
| | 1.0 | 0.5 | 73.36<br>($\sigma = 36.05$) | 58.52<br>($\sigma = 30.91$) | 80.76<br>(21/26) | 61.53<br>(16/26) | 71.14 |
| | 1.0 | 1.0 | 71.44<br>($\sigma = 36.71$) | 57.26<br>($\sigma = 33.39$) | 76.92<br>(20/26) | 61.53<br>(16/26) | 69.22 |
| | | Average: | 71.56<br>($\sigma = 1.88$) | 58.21<br>($\sigma = 1.00$) | 78.19 | 60.67 | **69.43** |

When the fourth hidden layer of 500 units is used for classification, it is observed from Table 44 that the overall subject-wise test accuracy for Youtube data is 73.92% and for Ohio data is 66.66%. The average segment-wise accuracy for Youtube data is 69.34% with a standard deviation of 2.03% and for the Ohio data, 60.26% with a standard deviation of 1.92%. The overall subject-wise accuracy is 70.29% outperforming the SVM baseline result of 55.55%.

Classification using the final hidden layer of 100 features produces the results shown in Table 42 that the overall subject-wise test accuracy for Youtube data is 74.35% and for Ohio data is 66.23%. The average segment-wise accuracy for Youtube data is 69.44% with a standard deviation of 2.57% and for the Ohio data, 59.98% with a standard deviation of 1.92%. The overall subject-wise accuracy is 70.29% which is above the SVM baseline by a margin of 14.74%.

**Table 41:** Classification results using the first hidden layer of 500 features generated after applying *transfer learning* and applying leave-one-subject-out cross-validation. The architecture is 2430-500-500-500-500-100.

| Number of Subjects | Frame Length (seconds) | Frame Shift (seconds) | Segment-wise Acc (%) on Youtube | Segment-wise Acc (%) on Ohio | Subject-wise Acc (%) on Youtube | Subject-wise Acc (%) on Ohio | Overall Subject-wise Accuracy (%) |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26 No of Subjects from Ohio:26 Total:52 subjects | 3.0 | 0.1 | 72.95<br>($\sigma = 36.27$) | 59.18<br>($\sigma = 29.33$) | 76.92<br>(20/26) | 61.53<br>(16/26) | 69.22 |
| | 3.0 | 0.5 | 74.52<br>($\sigma = 38.35$) | 63.67<br>($\sigma = 31.18$) | 76.92<br>(20/26) | 65.38<br>(17/26) | 71.15 |
| | 3.0 | 1.0 | 78.82<br>($\sigma = 36.38$) | 61.46<br>($\sigma = 34.44$) | 80.76<br>(21/26) | 61.53<br>(16/26) | 71.14 |
| | 2.0 | 0.1 | 72.83<br>($\sigma = 35.25$) | 58.75<br>($\sigma = 27.79$) | 76.92<br>(20/26) | 57.69<br>(15/26) | 67.30 |
| | 2.0 | 0.5 | 74.48<br>($\sigma = 38.10$) | 63.34<br>($\sigma = 31.17$) | 76.92<br>(20/26) | 69.23<br>(18/26) | 73.07 |
| | 2.0 | 1.0 | 79.33<br>($\sigma = 35.99$) | 63.11<br>($\sigma = 33.24$) | 80.76<br>(21/26) | 65.38<br>(17/26) | 73.07 |
| | 1.0 | 0.1 | 72.24<br>($\sigma = 34.24$) | 57.75<br>($\sigma = 26.53$) | 76.92<br>(20/26) | 61.53<br>(16/26) | 69.22 |
| | 1.0 | 0.5 | 74.05<br>($\sigma = 38.35$) | 61.76<br>($\sigma = 31.92$) | 76.92<br>(20/26) | 69.23<br>(18/26) | 73.07 |
| | 1.0 | 1.0 | 78.86<br>($\sigma = 35.42$) | 62.80<br>($\sigma = 32.97$) | 84.61<br>(22/26) | 69.23<br>(18/26) | 76.92 |
| | | Average: | 75.34<br>($\sigma = 2.85$) | 61.31<br>($\sigma = 2.21$) | 78.62 | 64.52 | **71.57** |

### 6.2.3 TRANSFER LEARNING FOR PTSD DIAGNOSIS WITH FEATURE SELECTION

Feature selection experiments were performed using the transfer learning framework as described previously in section 3.4. Figure 11 shows the concept of feature category selection in *transfer learning*. Only the first hidden layer was used each time as it produced the best performance. In the tables 46 and 47 we abbreviated prosodic features as “P”, vocal-tract features as “V”, and excitation features as “E”. For the six experimental groups, we defined “Out” as the feature category being excluded, and “Only” as the feature category remains while the others being excluded. To provide a more intelligible view, we bolded all data that show statistical significance, or that fit the criterion of rejecting the null hypothesis, for every table in this section.

Table 46 summarizes segment-wise accuracies using different feature combinations computed from transfer learning for PTSD diagnosis. While other average test accuracies are approximately between 60% and 80%, the results obtained by using excitation features only are potential outliers in this data set. This category had test accuracies less than 50%. Table 47

**Table 42:** Classification results using the second hidden layer of 500 features obtained after the application of *transfer learning* and applying leave-one-subject-out cross-validation. The architecture is 2430-500-500-500-500-100.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26<br>No of Subjects from Ohio:26<br>Total:52 subjects | 3.0 | 0.1 | 73.35<br>($\sigma = 34.55$) | 58.58<br>($\sigma = 27.34$) | 76.92<br>(20/26) | 65.38<br>(17/26) | 71.15 |
| | 3.0 | 0.5 | 74.14<br>($\sigma = 36.11$) | 61.86<br>($\sigma = 30.03$) | 76.92<br>(20/26) | 69.23<br>(18/26) | 73.07 |
| | 3.0 | 1.0 | 71.35<br>($\sigma = 36.35$) | 58.21<br>($\sigma = 33.58$) | 73.07<br>(19/26) | 57.69<br>(15/26) | 65.38 |
| | 2.0 | 0.1 | 73.25<br>($\sigma = 33.65$) | 57.53<br>($\sigma = 26.03$) | 76.92<br>(20/26) | 61.53<br>(16/26) | 69.22 |
| | 2.0 | 0.5 | 73.57<br>($\sigma = 36.06$) | 61.91<br>($\sigma = 29.47$) | 76.92<br>(20/26) | 69.23<br>(18/26) | 73.07 |
| | 2.0 | 1.0 | 75.04<br>($\sigma = 34.62$) | 59.04<br>($\sigma = 33.67$) | 80.76<br>(21/26) | 57.69<br>(15/26) | 69.22 |
| | 1.0 | 0.1 | 72.39<br>($\sigma = 32.56$) | 56.71<br>($\sigma = 24.12$) | 76.92<br>(20/26) | 53.84<br>(14/26) | 65.38 |
| | 1.0 | 0.5 | 73.20<br>($\sigma = 36.09$) | 61.51<br>($\sigma = 29.29$) | 76.92<br>(20/26) | 61.53<br>(16/26) | 69.22 |
| | 1.0 | 1.0 | 71.17<br>($\sigma = 36.81$) | 57.43<br>($\sigma = 32.94$) | 73.07<br>(19/26) | 57.69<br>(15/26) | 65.38 |
| | | Average: | 73.05<br>($\sigma = 1.24$) | 59.19<br>($\sigma = 2.03$) | 76.49 | 61.53 | **69.01** |

summarizes subject-wise accuracies using different feature combinations computed from transfer learning for PTSD diagnosis. It can be easily observed that the fractions of subjects correctly classified using excitation features only are half that of using other feature categories in the same dataset. The results for excitation features only are less than 50%[90, 91].

In summary, excitation features are least effective in detecting PTSD, in a majority of the experiments, and prosodic features seem to be the most effective feature category. To determine whether any of the results are statistically significant, we carried out Paired T-test for the individual results using the first hidden layer for classification, presented in section 6.3.2.

**Table 43:** Classification results utilizing the third hidden layer of 500 features, obtained through *transfer learning* and applying leave-one-subject-out cross-validation. The architecture is 2430-500-500-500-500-100.

| Number of Subjects | Frame Length (seconds) | Frame Shift (seconds) | Segment-wise Acc (%) on Youtube | Segment-wise Acc (%) on Ohio | Subject-wise Acc (%) on Youtube | Subject-wise Acc (%) on Ohio | Overall Subject-wise Accuracy (%) |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26<br>No of Subjects from Ohio:26<br>Total:52 subjects | 3.0 | 0.1 | 71.03<br>($\sigma = 35.80$) | 59.95<br>($\sigma = 25.90$) | 76.92<br>(20/26) | 65.38<br>(17/26) | 71.15 |
| | 3.0 | 0.5 | 71.34<br>($\sigma = 36.26$) | 62.69<br>($\sigma = 28.83$) | 76.92<br>(20/26) | 69.23<br>(18/26) | 73.07 |
| | 3.0 | 1.0 | 68.51<br>($\sigma = 36.70$) | 61.27<br>($\sigma = 32.11$) | 73.07<br>(19/26) | 65.38<br>(17/26) | 69.22 |
| | 2.0 | 0.1 | 70.85<br>($\sigma = 34.91$) | 58.74<br>($\sigma = 24.59$) | 76.92<br>(20/26) | 69.23<br>(18/26) | 73.07 |
| | 2.0 | 0.5 | 71.94<br>($\sigma = 36.36$) | 62.34<br>($\sigma = 28.73$) | 76.92<br>(20/26) | 69.23<br>(18/26) | 73.07 |
| | 2.0 | 1.0 | 70.10<br>($\sigma = 36.07$) | 60.56<br>($\sigma = 31.53$) | 76.92<br>(20/26) | 65.38<br>(17/26) | 71.15 |
| | 1.0 | 0.1 | 70.45<br>($\sigma = 33.69$) | 57.44<br>($\sigma = 22.92$) | 76.92<br>(20/26) | 69.23<br>(18/26) | 73.07 |
| | 1.0 | 0.5 | 70.81<br>($\sigma = 36.55$) | 61.65<br>($\sigma = 27.87$) | 76.92<br>(20/26) | 65.38<br>(17/26) | 71.15 |
| | 1.0 | 1.0 | 69.53<br>($\sigma = 36.34$) | 58.44<br>($\sigma = 31.42$) | 69.23<br>(18/26) | 57.69<br>(15/26) | 63.46 |
| | | Average: | 70.50<br>($\sigma = 1.02$) | 60.34<br>($\sigma = 1.83$) | 75.63 | 66.23 | **70.93** |

### 6.2.4 TRANSFER LEARNING FOR PTSD DIAGNOSIS USING MULTIPLE FRAME MFCC FEATURES

The fifth and final experiments used MFCC features as input. In the fifth experiment, the TIMIT speech features were extracted as follows. Instead of extracting the previously stated type of features, only *MFCC* features were extracted. The speech signal was first pre-emphasized using a first order FIR filter. Then the speech signal was divided into a set of frames of length 25ms with an overlapping of 10ms between two consecutive frames. For each frame, 13 *MFCC* features were extracted using discrete cosine transform (DCT) based on 40 *mel* scale frequency band energies. For each frame, the first and second time derivatives were also computed making the total number of features for each frame, 39. Fifteen (15) frames were used to predict the phoneme class of the center frame. For this configuration, there were 585 (39x15) features and 39 classes in this data set. The network architecture used was 585-500-500-500-500-100-39. There were approximately 440,000 data points for training and approximately 50,000 samples for testing. The hyperparameters used for pre-training and fine tuning were the same as those described in section 4.1.3.1. Output phone classification was carried out using the logistic regression classifier.

**Table 44:** Classification results using the fourth hidden layer of 500 features generated after applying *transfer learning* and applying leave-one-subject-out cross-validation. The architecture is 2430-500-500-500-500-100.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26 No of Subjects from Ohio:26 Total:52 subjects | 3.0 | 0.1 | 70.73 ($\sigma = 35.32$) | 59.16 ($\sigma = 24.26$) | 76.92 (20/26) | 69.23 (18/26) | 73.07 |
| | 3.0 | 0.5 | 71.28 ($\sigma = 35.62$) | 62.71 ($\sigma = 27.37$) | 76.92 (20/26) | 69.23 (18/26) | 73.07 |
| | 3.0 | 1.0 | 66.45 ($\sigma = 37.23$) | 59.69 ($\sigma = 30.18$) | 69.23 (18/26) | 61.53 (16/26) | 65.38 |
| | 2.0 | 0.1 | 70.15 ($\sigma = 34.69$) | 58.44 ($\sigma = 23.08$) | 76.92 (20/26) | 69.23 (18/26) | 73.07 |
| | 2.0 | 0.5 | 71.26 ($\sigma = 35.61$) | 62.80 ($\sigma = 26.83$) | 76.92 (20/26) | 73.07 (19/26) | 74.99 |
| | 2.0 | 1.0 | 66.73 ($\sigma = 36.50$) | 60.37 ($\sigma = 29.95$) | 69.23 (18/26) | 65.38 (17/26) | 67.30 |
| | 1.0 | 0.1 | 69.97 ($\sigma = 33.54$) | 57.25 ($\sigma = 20.85$) | 76.92 (20/26) | 65.38 (17/26) | 71.15 |
| | 1.0 | 0.5 | 70.61 ($\sigma = 36.29$) | 62.00 ($\sigma = 26.43$) | 76.92 (20/26) | 69.23 (18/26) | 73.07 |
| | 1.0 | 1.0 | 66.88 ($\sigma = 36.09$) | 59.93 ($\sigma = 28.72$) | 65.38 (17/26) | 57.69 (15/26) | 61.53 |
| | | Average: | 69.34 ($\sigma = 2.03$) | 60.26 ($\sigma = 1.92$) | 73.92 | 66.66 | **70.29** |

For the case of using the first hidden layer of 500 features for classification, as shown by Table 48, the overall subject-wise test accuracy for Youtube data is 74.35% and for Ohio data is 64.95%. The average segment-wise accuracy for Youtube data is 77.41% with a standard deviation of 2.69% and for the Ohio data, 52.31% with a standard deviation of 2.04%. The overall subject-wise accuracy for all subjects is 69.65% which is higher than that achieved by the baseline(67.30%) by a margin of 2.35%.

Classification using the second hidden layer of 500 features, Table 49 shows that the overall subject-wise test accuracy for Youtube data is 76.91% and for Ohio data is 57.68%. The average segment-wise accuracy for Youtube data is 76.24% with a standard deviation of 1.98% and for the Ohio data, 52.41% with a standard deviation of 3.26%. The overall subject-wise accuracy for all subjects is 67.30% which equals that achieved by the baseline.

**Table 45:** Classification results using the final hidden layer of 100 features and applying leave-one-subject-out cross-validation. The 500 features are generated by applying *transfer learning* on the PTSD data sets. The architecture is 2430-500-500-500-500-100.

| Number of Subjects | Frame Length (seconds) | Frame Shift (seconds) | Segment-wise Acc (%) on Youtube | Segment-wise Acc (%) on Ohio | Subject-wise Acc (%) on Youtube | Subject-wise Acc (%) on Ohio | Overall Subject-wise Accuracy (%) |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26<br>No of Subjects from Ohio:26<br>Total:52 subjects | 3.0 | 0.1 | 70.59 ($\sigma = 36.02$) | 58.03 ($\sigma = 22.18$) | 76.92 (20/26) | 65.38 (17/26) | 71.15 |
| | 3.0 | 0.5 | 71.23 ($\sigma = 37.56$) | 60.48 ($\sigma = 25.20$) | 73.07 (19/26) | 65.38 (17/26) | 69.22 |
| | 3.0 | 1.0 | 68.14 ($\sigma = 37.56$) | 62.17 ($\sigma = 26.51$) | 73.07 (19/26) | 65.38 (17/26) | 69.22 |
| | 2.0 | 0.1 | 70.00 ($\sigma = 34.92$) | 56.61 ($\sigma = 21.09$) | 76.92 (20/26) | 65.38 (17/26) | 71.15 |
| | 2.0 | 0.5 | 71.21 ($\sigma = 37.08$) | 60.68 ($\sigma = 24.71$) | 73.07 (19/26) | 69.23 (18/26) | 71.15 |
| | 2.0 | 1.0 | 63.38 ($\sigma = 38.12$) | 64.06 ($\sigma = 26.75$) | 73.07 (19/26) | 69.23 (18/26) | 71.15 |
| | 1.0 | 0.1 | 69.87 ($\sigma = 33.61$) | 55.72 ($\sigma = 18.78$) | 76.92 (20/26) | 61.53 (16/26) | 69.22 |
| | 1.0 | 0.5 | 71.86 ($\sigma = 36.91$) | 60.25 ($\sigma = 23.59$) | 73.07 (19/26) | 69.23 (18/26) | 71.15 |
| | 1.0 | 1.0 | 68.73 ($\sigma = 37.93$) | 61.84 ($\sigma = 27.03$) | 73.07 (19/26) | 65.38 (17/26) | 69.22 |
| | | Average: | 69.44 ($\sigma = 2.57$) | 59.98 ($\sigma = 1.92$) | 74.35 | 66.23 | **70.29** |

**Table 46:** Summaries of segment-wise accuracies using different feature combinations computed from transfer learning for PTSD diagnosis. The architecture was 2430-500-500-500-500-100.

| Segment-wise Accuracies | | Original | P Out | V Out | E Out | P Only | V Only | E Only |
|---|---|---|---|---|---|---|---|---|
| Youtube | Average(Test) | 78.82 | 73.96 | 80.05 | 80.87 | 78.49 | 75.36 | **49.56** |
| | Std(Test) | 36.38 | 35.99 | 33.64 | 33.62 | 36.47 | 34.47 | 37.56 |
| Ohio | Average(Test) | 61.46 | 63.71 | 61.80 | 63.49 | 67.69 | 65.26 | **35.69** |
| | Std(Test) | 34.44 | 34.90 | 37.93 | 33.51 | 36.10 | 33.56 | 32.66 |
| Overall Average: | | 70.14 | 68.83 | 70.92 | 72.18 | 73.09 | 70.31 | 42.62 |

Using the third hidden layer of 500 features for classification, it is observed from Table 50 that the overall subject-wise test accuracy for Youtube data is 80.76% and for Ohio data is 56.40%. The average segment-wise accuracy for Youtube data is 75.11% with a standard deviation of 3.05% and for the Ohio data, 52.42% with a standard deviation of 3.53%. The overall subject-wise accuracy for all subjects is 68.58% which is higher when compared to the baseline accuracy of 67.30%.

**Table 47:** Summaries of subject-wise accuracies using different feature combinations computed from transfer learning for PTSD diagnosis. The architecture was 2430-500-500-500-500-100.

| **Subject-wise Accuracies** | | **Original** | **P Out** | **V Out** | **E Out** | **P Only** | **V Only** | **E Only** |
|---|---|---|---|---|---|---|---|---|
| Youtube | In-Fractions | 21/26 | 20/26 | 22/26 | 22/26 | 21/26 | 20/26 | **12/26** |
| | In (%) | 80.77 | 76.92 | 84.62 | 84.62 | 80.77 | 76.92 | **46.15** |
| Ohio | In-Fractions | 16/26 | 16/26 | 15/26 | 19/26 | 18/26 | 16/26 | **6/26** |
| | In (%) | 61.54 | 61.54 | 57.69 | 73.08 | 69.23 | 61.54 | **23.08** |
| Overall Average: | | 71.15 | 69.23 | 71.15 | 78.85 | 75.00 | 69.23 | 34.61 |

**Table 48:** Classification results using the first hidden layer of 500 features, obtained through *transfer learning* and applying leave-one-subject-out cross-validation. The MFCC features, extracted from PTSD audio recordings are fed as input to the network and have 585 dimensions. The architecture is 585-500-500-500-500-100.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26 No of Subjects from Ohio:26 Total:52 subjects | 3.0 | 0.1 | 80.69 ($\sigma = 25.95$) | 54.39 ($\sigma = 25.22$) | 84.61 (22/26) | 61.53 (16/26) | 73.07 |
| | 3.0 | 0.5 | 80.85 ($\sigma = 25.96$) | 54.35 ($\sigma = 24.78$) | 80.76 (21/26) | 61.53 (16/26) | 71.14 |
| | 3.0 | 1.0 | 76.77 ($\sigma = 27.74$) | 51.70 ($\sigma = 22.92$) | 84.61 (22/26) | 53.84 (14/26) | 69.22 |
| | 2.0 | 0.1 | 78.91 ($\sigma = 25.96$) | 53.76 ($\sigma = 23.61$) | 80.76 (21/26) | 61.53 (16/26) | 71.14 |
| | 2.0 | 0.5 | 79.04 ($\sigma = 25.96$) | 53.90 ($\sigma = 23.47$) | 80.76 (21/26) | 57.69 (15/26) | 69.22 |
| | 2.0 | 1.0 | 75.48 ($\sigma = 26.27$) | 50.59 ($\sigma = 22.04$) | 80.76 (21/26) | 50.00 (13/26) | 65.38 |
| | 1.0 | 0.1 | 76.05 ($\sigma = 25.06$) | 52.26 ($\sigma = 21.16$) | 65.38 (17/26) | 80.76 (21/26) | 73.07 |
| | 1.0 | 0.5 | 76.34 ($\sigma = 24.91$) | 51.67 ($\sigma = 21.43$) | 65.38 (17/26) | 80.76 (21/26) | 73.07 |
| | 1.0 | 1.0 | 72.55 ($\sigma = 25.35$) | 48.22 ($\sigma = 21.17$) | 46.15 (12/26) | 80.76 (21/26) | 61.53 |
| | | Average: | 77.41 ($\sigma = 2.69$) | 52.31 ($\sigma = 2.04$) | 74.35 | 64.95 | **69.65** |

When using the fourth hidden layer of 500 features for classification, it is observed from Table 51 that the overall subject-wise test accuracy for Youtube data is 79.90% and for Ohio data is 53.84%. The average segment-wise accuracy for Youtube data is 75.95% with a standard

**Table 49:** Classification results using the second hidden layer of 500 features, obtained through *transfer learning* and applying leave-one-subject-out cross-validation. The MFCC features, extracted from PTSD audio recordings are fed as input to the network and have 585 dimensions. The architecture is 585-500-500-500-500-100.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26<br>No of Subjects from Ohio:26<br>Total:52 subjects | 3.0 | 0.1 | 72.88<br>($\sigma = 25.38$) | 56.92<br>($\sigma = 24.08$) | 80.76<br>(21/26) | 57.69<br>(15/26) | 69.22 |
| | 3.0 | 0.5 | 80.02<br>($\sigma = 22.95$) | 54.58<br>($\sigma = 24.02$) | 80.76<br>(21/26) | 61.53<br>(16/26) | 71.14 |
| | 3.0 | 1.0 | 77.22<br>($\sigma = 24.65$) | 49.88<br>($\sigma = 23.86$) | 76.92<br>(20/26) | 57.69<br>(15/26) | 67.30 |
| | 2.0 | 0.1 | 76.85<br>($\sigma = 22.66$) | 54.75<br>($\sigma = 21.11$) | 80.76<br>(21/26) | 61.53<br>(16/26) | 71.14 |
| | 2.0 | 0.5 | 77.12<br>($\sigma = 25.45$) | 53.15<br>($\sigma = 20.55$) | 65.38<br>(17/26) | 53.84<br>(14/26) | 59.61 |
| | 2.0 | 1.0 | 76.54<br>($\sigma = 24.65$) | 47.20<br>($\sigma = 22.88$) | 80.76<br>(21/26) | 50.00<br>(13/26) | 65.38 |
| | 1.0 | 0.1 | 75.68<br>($\sigma = 20.12$) | 51.19<br>($\sigma = 16.98$) | 65.38<br>(17/26) | 61.53<br>(16/26) | 63.45 |
| | 1.0 | 0.5 | 74.96<br>($\sigma = 21.19$) | 55.11<br>($\sigma = 19.99$) | 80.76<br>(21/26) | 65.38<br>(17/26) | 73.07 |
| | 1.0 | 1.0 | 74.96<br>($\sigma = 21.99$) | 48.97<br>($\sigma = 18.82$) | 80.76<br>(21/26) | 50.00<br>(13/26) | 65.38 |
| | | Average: | 76.24<br>($\sigma = 1.98$) | 52.41<br>($\sigma = 3.26$) | 76.91 | 57.68 | **67.30** |

deviation of 2.78% and for the Ohio data it is 52.20% with a standard deviation of 3.83%. The overall subject-wise accuracy for all subjects is 66.87% which is inferior to that achieved by the baseline.

The results of applying *transfer learning* using a deeper network architecture of 585-500-500-500-500-100 are shown in Tables 52 through 48. In this scenario, 585 is the feature dimensionality of the input layer consisting of *MFCC* features, extracted from PTSD audio files. When using the final hidden layer of 100 features for classification, it is observed from Table 52 that the overall subject-wise test accuracy for Youtube data is 88.03% and for Ohio data is 55.97%. The average segment-wise accuracy for Youtube data is 79.17% with a standard deviation of 2.26% and for the Ohio data it is 52.16% with a standard deviation of 2.43%. The overall subject-wise accuracy for all subjects is 72.00%. Compared to the baseline which achieved 67.30%, the performance is higher by a margin of 4.70%.

The sixth experiment used the same parameter values and the same feature extraction process used in the fifth experiment. A network configuration of 585-2000-2000-2000, was used for carrying out transfer learning. *Transfer learning* was then applied. All three layers of features

**Table 50:** Classification results using the third hidden layer of 500 features, generated by applying *transfer learning* and applying leave-one-subject-out cross-validation. The MFCC features, extracted from PTSD audio recordings are fed as input to the network and have 585 dimensions. The architecture is 585-500-500-500-500-100.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26 No of Subjects from Ohio:26 Total:52 subjects | 3.0 | 0.1 | 78.11<br>($\sigma = 21.68$) | 55.11<br>($\sigma = 20.12$) | 84.61<br>(22/26) | 57.69<br>(15/26) | 71.15 |
| | 3.0 | 0.5 | 80.12<br>($\sigma = 20.45$) | 50.26<br>($\sigma = 21.88$) | 80.76<br>(21/26) | 57.69<br>(15/26) | 69.22 |
| | 3.0 | 1.0 | 75.47<br>($\sigma = 24.26$) | 55.16<br>($\sigma = 21.92$) | 80.76<br>(21/26) | 50.00<br>(13/26) | 65.38 |
| | 2.0 | 0.1 | 74.77<br>($\sigma = 23.00$) | 54.18<br>($\sigma = 20.12$) | 84.61<br>(22/26) | 57.69<br>(15/26) | 71.15 |
| | 2.0 | 0.5 | 77.12<br>($\sigma = 21.40$) | 53.75<br>($\sigma = 19.18$) | 80.76<br>(20/26) | 53.84<br>(14/26) | 67.30 |
| | 2.0 | 1.0 | 74.29<br>($\sigma = 21.92$) | 54.88<br>($\sigma = 18.85$) | 84.61<br>(22/26) | 53.84<br>(14/26) | 69.22 |
| | 1.0 | 0.1 | 70.21<br>($\sigma = 22.92$) | 52.04<br>($\sigma = 18.23$) | 76.92<br>(20/26) | 65.38<br>(17/26) | 71.15 |
| | 1.0 | 0.5 | 74.15<br>($\sigma = 20.07$) | 52.35<br>($\sigma = 18.23$) | 76.92<br>(20/26) | 65.38<br>(17/26) | 71.15 |
| | 1.0 | 1.0 | 71.81<br>($\sigma = 22.26$) | 44.09<br>($\sigma = 18.25$) | 76.92<br>(20/26) | 46.15<br>(12/26) | 61.53 |
| | | Average: | 75.11<br>($\sigma = 3.05$) | 52.42<br>($\sigma = 3.53$) | 80.76 | 56.40 | **68.58** |

were used to evaluate the classification performance. The results are presented in Tables 53 through 55.

For the case in which the first hidden layer of 2000 features, is used for classification, it is observed from Table 53 that the overall subject-wise test accuracy for Youtube data is 88.03% and for Ohio data is 56.46%. The average segment-wise accuracy for Youtube data is 78.94% with a standard deviation of 2.27% and for the Ohio data, 53.14% with a standard deviation of 1.94%. The overall subject-wise accuracy is 63.02% which is inferior to the baseline result of 67.30%. The highest attained subject-wise accuracy of 69.43% by this network is achieved by the second hidden layer.

Classification, performed using the second hidden layer of 2000 features, produces the results shown in Table 54. The overall subject-wise test accuracy for Youtube data is 82.47% and for Ohio data is 56.83%. The average segment-wise accuracy for Youtube data is 76.98% with a standard deviation of 1.97% and for the Ohio data, 52.45% with a standard deviation of 1.53%. Compared to the baseline which achieved 67.30%, the overall subject-wise accuracy is superior at 69.65%.

**Table 51:** Classification results using the fourth hidden layer of 500 features obtained through *transfer learning* and applying leave-one-subject-out cross-validation. The MFCC features, extracted from PTSD audio recordings are fed as input to the network and have 585 dimensions. The architecture is 585-500-500-500-500-100.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26 No of Subjects from Ohio:26 Total:52 subjects | 3.0 | 0.1 | 79.03<br>($\sigma = 24.68$) | 54.82<br>($\sigma = 21.64$) | 84.61<br>(22/26) | 53.84<br>(14/26) | 69.22 |
| | 3.0 | 0.5 | 79.01<br>($\sigma = 25.45$) | 49.56<br>($\sigma = 21.97$) | 80.76<br>(21/26) | 50.00<br>(13/26) | 65.38 |
| | 3.0 | 1.0 | 78.55<br>($\sigma = 22.45$) | 58.15<br>($\sigma = 20.12$) | 76.92<br>(20/26) | 53.84<br>(14/26) | 65.38 |
| | 2.0 | 0.1 | 77.18<br>($\sigma = 24.00$) | 54.18<br>($\sigma = 20.04$) | 84.61<br>(22/26) | 57.69<br>(15/26) | 71.15 |
| | 2.0 | 0.5 | 76.96<br>($\sigma = 24.40$) | 53.75<br>($\sigma = 20.00$) | 80.76<br>(21/26) | 53.84<br>(14/26) | 67.30 |
| | 2.0 | 1.0 | 73.75<br>($\sigma = 25.51$) | 48.90<br>($\sigma = 21.37$) | 76.92<br>(20/26) | 46.15<br>(12/26) | 61.53 |
| | 1.0 | 0.1 | 73.69<br>($\sigma = 21.92$) | 53.04<br>($\sigma = 17.10$) | 80.76<br>(21/26) | 61.53<br>(16/26) | 71.14 |
| | 1.0 | 0.5 | 73.97<br>($\sigma = 22.07$) | 52.35<br>($\sigma = 17.92$) | 76.92<br>(20/26) | 65.38<br>(17/26) | 71.15 |
| | 1.0 | 1.0 | 71.49<br>($\sigma = 22.26$) | 45.09<br>($\sigma = 18.18$) | 76.92<br>(20/26) | 42.37<br>(11/26) | 59.64 |
| | | Average: | 75.95<br>($\sigma = 2.78$) | 52.20<br>($\sigma = 3.83$) | 79.90 | 53.84 | **66.87** |

In the sixth experimental scenario, *transfer learning* is applied using a network architecture of 585-2000-2000-2000. In this experiment, the final hidden layer of 2000 features is obtained. When this layer of features is used for classification, it is observed from Table 55 that the overall subject-wise test accuracy for Youtube data is 81.18%. For Ohio data the mean subject-wise accuracy is 55.97%. The average segment-wise accuracy for Youtube data is 76.53% with a standard deviation of 2.45% and for the Ohio data it is 51.82% with a standard deviation of 1.58%. Compared to the baseline which achieved 67.30%, the overall subject-wise accuracy is marginally higher at 68.58%.

## 6.3 DISCUSSION

From tables 56 and 57 we can see that for *transfer learning* the best overall subject-wise accuracy of 72.00% is achieved by the final layer of the network architecture, 585-500-500-500-500-500-100. Tables 58 and 57 show that for *transfer learning* the best overall subject-wise accuracy achieved was 76.92% by the first layer of the network architecture 2430-500-500-500-500-100. A comparison of the performance of the three different models in the proposed method is presented

**Table 52:** Classification results using the final hidden layer of 100 features obtained through *transfer learning* and applying leave-one-subject-out cross-validation. The MFCC features, extracted from PTSD audio recordings are fed as input to the network and have 585 dimensions. The architecture is 585-500-500-500-500-100.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26 No of Subjects from Ohio:26 Total:52 subjects | 3.0 | 0.1 | 81.69 ($\sigma = 25.20$) | 55.50 ($\sigma = 25.72$) | 88.46 (23/26) | 61.53 (16/26) | 74.99 |
| | 3.0 | 0.5 | 81.87 ($\sigma = 25.28$) | 55.58 ($\sigma = 25.70$) | 88.46 (23/26) | 61.53 (16/26) | 74.99 |
| | 3.0 | 1.0 | 79.55 ($\sigma = 25.88$) | 50.25 ($\sigma = 26.30$) | 88.46 (23/26) | 53.84 (14/26) | 71.15 |
| | 2.0 | 0.1 | 80.16 ($\sigma = 24.63$) | 54.27 ($\sigma = 23.96$) | 88.46 (23/26) | 57.69 (15/26) | 73.07 |
| | 2.0 | 0.5 | 80.37 ($\sigma = 24.57$) | 53.88 ($\sigma = 23.96$) | 88.46 (23/26) | 57.69 (15/26) | 73.07 |
| | 2.0 | 1.0 | 77.11 ($\sigma = 25.86$) | 49.88 ($\sigma = 24.86$) | 88.46 (23/26) | 50.00 (13/26) | 69.23 |
| | 1.0 | 0.1 | 77.72 ($\sigma = 23.67$) | 53.01 ($\sigma = 21.88$) | 88.46 (23/26) | 57.69 (15/26) | 73.07 |
| | 1.0 | 0.5 | 77.97 ($\sigma = 23.52$) | 52.45 ($\sigma = 21.90$) | 88.46 (23/26) | 57.96 (15/26) | 73.07 |
| | 1.0 | 1.0 | 75.03 ($\sigma = 24.44$) | 49.07 ($\sigma = 22.08$) | 84.61 (22/26) | 46.15 (12/26) | 65.38 |
| | | Average: | 79.17 ($\sigma = 2.26$) | 52.16 ($\sigma = 2.43$) | 88.03 | 55.97 | **72.00** |

in details by means of hypothesis testing in the latter part of this section.

### 6.3.1 SUMMARY OF MEAN AND BEST TEST ACCURACIES BY ALL MODELS ACROSS ALL EXPERIMENTS

Tables 56 and 57 summarize the average results achieved by all the scenarios of *transfer learning* for PTSD diagnosis. Tables 58 and 59 summarize the best subject-wise accuracies across all the *transfer learning* experiments.

### 6.3.2 HYPOTHESIS TESTING

We applied hypothesis tests to first identify if the performance differences between the competing models were statistically significant. We then extended it to test for statistical significance between competing feature categories utilizing *transfer learning*. The two types of

**Table 53:** Classification results using the first hidden layer of 2000 features, obtained through *transfer learning* and applying leave-one-subject-out cross-validation. The MFCC features, extracted from PTSD audio recordings are fed as input to the network which have 585 dimensions. The architecture is 585-2000-2000-2000.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26 No of Subjects from Ohio:26 Total:52 subjects | 3.0 | 0.1 | 81.47 ($\sigma = 24.98$) | 55.44 ($\sigma = 25.98$) | 88.46 (23/26) | 61.53 (16/26) | 66.34 |
| | 3.0 | 0.5 | 81.72 ($\sigma = 25.11$) | 55.58 ($\sigma = 25.97$) | 88.46 (23/26) | 61.53 (16/26) | 65.37 |
| | 3.0 | 1.0 | 79.29 ($\sigma = 25.55$) | 50.79 ($\sigma = 24.74$) | 88.46 (23/26) | 53.84 (14/26) | 61.53 |
| | 2.0 | 0.1 | 80.01 ($\sigma = 24.46$) | 54.22 ($\sigma = 24.19$) | 88.46 (23/26) | 57.69 (15/26) | 64.42 |
| | 2.0 | 0.5 | 80.22 ($\sigma = 24.22$) | 54.39 ($\sigma = 23.95$) | 88.46 (23/26) | 57.69 (15/26) | 63.45 |
| | 2.0 | 1.0 | 77.27 ($\sigma = 25.66$) | 51.14 ($\sigma = 24.07$) | 88.46 (23/26) | 50.00 (13/26) | 59.61 |
| | 1.0 | 0.1 | 77.97 ($\sigma = 23.62$) | 53.35 ($\sigma = 21.96$) | 88.46 (23/26) | 57.69 (15/26) | 62.49 |
| | 1.0 | 0.5 | 78.14 ($\sigma = 23.67$) | 52.83 ($\sigma = 21.65$) | 88.46 (23/26) | 57.69 (15/26) | 64.42 |
| | 1.0 | 1.0 | 74.45 ($\sigma = 24.34$) | 50.56 ($\sigma = 20.39$) | 84.61 (22/26) | 50.00 (13/26) | 59.61 |
| | | Average: | 78.94 ($\sigma = 2.27$) | 53.14 ($\sigma = 1.94$) | 88.03 | 56.46 | **63.02** |

hypothesis tests we used were parametric and non-parametric tests. Parametric tests assume a normal population but the non-parametric tests do not make any such assumption. We used a *paired t-test* for parametric and the *Wilcoxon Signed-Rank test* to test non-parametrically.

First, we applied the *paired t-test* between one of the transfer learning networks and the competing SVM, DBN and sparse coding models to analyze if the performance by *transfer learning* showed statistically significant differences compared to the other models in the proposed method. Secondly, we applied the *paired-t test* in the case of *transfer learning* utilizing feature selection, where different competing feature categories have different performances. Lastly, we applied the *Wilcoxon Signed-Rank test* in the case with differing feature categories.

**Table 54:** Classification results using the second hidden layer of 2000 features, obtained through *transfer learning* and applying leave-one-subject-out cross-validation. The MFCC features, extracted from PTSD audio recordings are fed as input to the network which have 585 dimensions. The architecture is 585-2000-2000-2000.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26 No of Subjects from Ohio:26 Total:52 subjects | 3.0 | 0.1 | 79.31<br>($\sigma = 27.19$) | 53.90<br>($\sigma = 25.90$) | 84.61<br>(21/26) | 57.69<br>(15/26) | 71.15 |
| | 3.0 | 0.5 | 79.60<br>($\sigma = 26.63$) | 53.96<br>($\sigma = 25.44$) | 88.46<br>(23/26) | 53.84<br>(14/26) | 71.15 |
| | 3.0 | 1.0 | 75.54<br>($\sigma = 26.48$) | 53.41<br>($\sigma = 21.95$) | 80.76<br>(21/26) | 57.69<br>(15/26) | 69.22 |
| | 2.0 | 0.1 | 78.05<br>($\sigma = 26.67$) | 52.87<br>($\sigma = 23.62$) | 84.61<br>(22/26) | 57.69<br>(15/26) | 71.15 |
| | 2.0 | 0.5 | 78.45<br>($\sigma = 26.51$) | 53.47<br>($\sigma = 23.55$) | 80.76<br>(21/26) | 57.69<br>(15/26) | 69.22 |
| | 2.0 | 1.0 | 75.74<br>($\sigma = 26.46$) | 52.21<br>($\sigma = 20.37$) | 80.76<br>(21/26) | 57.69<br>(15/26) | 69.22 |
| | 1.0 | 0.1 | 76.15<br>($\sigma = 25.68$) | 51.52<br>($\sigma = 20.87$) | 80.76<br>(21/26) | 53.84<br>(14/26) | 67.30 |
| | 1.0 | 0.5 | 76.37<br>($\sigma = 25.55$) | 51.49<br>($\sigma = 20.92$) | 80.76<br>(21/26) | 57.69<br>(15/26) | 69.22 |
| | 1.0 | 1.0 | 73.67<br>($\sigma = 24.32$) | 49.22<br>($\sigma = 17.54$) | 80.76<br>(21/26) | 57.69<br>(15/26) | 69.22 |
| | | Average: | 76.98<br>($\sigma = 1.97$) | 52.45<br>($\sigma = 1.53$) | 82.47 | 56.83 | **69.65** |

#### 6.3.2.1 PAIRED T-TEST

Paired t-test compares two quantitative population means where observations from one population are paired with the observations from the other population. The assumption is that the observations follow a normal distribution. In this case, we paired up the initial test accuracies (the control group) and accuracies from each individual attempt at removal (the paired groups), assuming that the data are normally distributed. Paired t-test is used to assess if two sets of segment-wise accuracies from the control group and the paired group are significantly different. The t statistic is determined based on the mean and standard deviation of the group difference, and the degree of freedom. Based on the t statistic, we calculate the two-tailed p-value using the t-distribution with degree of freedom. If p-value is smaller than 0.05, we would reject the null hypothesis that the means of two distributions are equal.

Statistical hypothesis testing utilizing the Paired t-test was carried out between one of the transfer learning networks and SVM to assess the relative performance of PTSD detection between the competing models. Tables 60 through 63 show the results of hypothesis testing. With regard to hypothesis testing, it can be seen from Table 60 that in 13 of 27 cases, the results

**Table 55:** Classification results using the final hidden layer of 2000 features, obtained through *transfer learning* and applying leave-one-subject-out cross-validation. The MFCC features, extracted from PTSD audio recordings are fed as input to the network which have 585 dimensions. The architecture is 585-2000-2000-2000.

| **Number of Subjects** | **Frame Length (seconds)** | **Frame Shift (seconds)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| No of Subjects from Youtube:26<br>No of Subjects from Ohio:26<br>Total:52 subjects | 3.0 | 0.1 | 79.62<br>($\sigma = 24.94$) | 53.36<br>($\sigma = 24.70$) | 80.76<br>(21/26) | 57.69<br>(15/26) | 69.22 |
| | 3.0 | 0.5 | 79.76<br>($\sigma = 24.76$) | 53.80<br>($\sigma = 24.87$) | 80.76<br>(21/26) | 57.69<br>(15/26) | 69.22 |
| | 3.0 | 1.0 | 75.51<br>($\sigma = 25.76$) | 51.91<br>($\sigma = 21.54$) | 80.76<br>(21/26) | 53.84<br>(14/26) | 67.30 |
| | 2.0 | 0.1 | 78.32<br>($\sigma = 24.04$) | 52.56<br>($\sigma = 22.51$) | 80.76<br>(21/26) | 57.69<br>(15/26) | 69.22 |
| | 2.0 | 0.5 | 77.05<br>($\sigma = 24.11$) | 52.70<br>($\sigma = 22.09$) | 80.76<br>(21/26) | 57.69<br>(15/26) | 69.22 |
| | 2.0 | 1.0 | 74.79<br>($\sigma = 24.19$) | 49.52<br>($\sigma = 19.47$) | 84.61<br>(22/26) | 53.84<br>(14/26) | 69.22 |
| | 1.0 | 0.1 | 75.93<br>($\sigma = 23.80$) | 51.44<br>($\sigma = 19.86$) | 80.76<br>(21/26) | 53.84<br>(14/26) | 67.30 |
| | 1.0 | 0.5 | 75.81<br>($\sigma = 23.56$) | 51.91<br>($\sigma = 19.70$) | 80.76<br>(21/26) | 53.84<br>(14/26) | 67.30 |
| | 1.0 | 1.0 | 72.06<br>($\sigma = 22.67$) | 49.19<br>($\sigma = 16.89$) | 80.76<br>(21/26) | 57.69<br>(15/26) | 69.22 |
| | | Average: | 76.53<br>($\sigma = 2.45$) | 51.82<br>($\sigma = 1.58$) | 81.18 | 55.97 | **68.58** |

are statistically significant with *p-values* less than the chosen significance level of 5%. It is also observed that for those cases, the performance of *transfer learning* is better than the baseline. Table 61 shows that in 4 of 27 cases, the classification accuracies are also statistically significant. These statistically significant results from these two tables suggests that the sample accuracies from the two different methods come from normally distributed populations with unequal means. The hypothesis testing experiment is also repeated between *transfer learning* and DBN.

With regard to hypothesis testing between *transfer learning* and DBN, it can be seen from Table 62 that in 8 of 27 cases, the results are significantly different with *p-values* less than the significance level. From Table 63 it is observed that in 2 of 27 cases, the classification accuracies achieve statistical significance. Results from these two tables suggests that the sample accuracies from the two different methods come from normally distributed populations with unequal means. It is also observed that in all those instances, *transfer learning* outperforms the baseline.

**Table 56:** Summary of average PTSD diagnostic accuracies across all the *transfer learning* experiments which have input feature combination of prosodic, vocal-tract and excitation features.

| **Method: Transfer Learning (Architecture) Input Prosodic, Vocal-tract and Excitation Features** | **Mean Segment-Wise Acc (%) on Youtube** | **Mean Segment-wise Acc (%) on Ohio** | **Mean Subject-wise Acc (%) on Youtube** | **Mean Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|
| 162-100-50, Layer 1 | 76.42<br>($\sigma = 2.06$) | 54.92<br>($\sigma = 2.64$) | 85.03 | 44.86 | **64.95** |
| 162-100-50, Layer 2 | 73.77<br>($\sigma = 1.94$) | 53.23<br>($\sigma = 1.32$) | 84.61 | 47.43 | **66.02** |
| Average: | 75.09<br>($\sigma = 1.87$) | 54.07<br>($\sigma = 1.19$) | 84.82 | 46.14 | **65.48** |
| 2430-100-50, Layer 1 | 74.23<br>($\sigma = 2.28$) | 61.48<br>($\sigma = 2.64$) | 77.34 | 62.81 | **70.08** |
| 2430-100-5), Layer 2 | 68.89<br>($\sigma = 2.35$) | 60.09<br>($\sigma = 3.36$) | 71.36 | 67.94 | **69.65** |
| Average: | 71.56<br>($\sigma = 3.77$) | 60.78<br>($\sigma = 0.98$) | 74.35 | 65.37 | **69.86** |
| 2430-1000-1000-50, Layer 1 | 76.46<br>($\sigma = 2.12$) | 61.02<br>($\sigma = 2.28$) | 78.20 | 62.81 | **70.50** |
| 2430-1000-1000-500, Layer 2 | 73.48<br>($\sigma = 1.68$) | 57.41<br>($\sigma = 1.16$) | 77.34 | 59.39 | **68.37** |
| 2430-1000-1000-500, Layer 3 | 71.56<br>($\sigma = 1.88$) | 58.21<br>($\sigma = 1.00$) | 78.19 | 60.67 | **69.43** |
| Average: | 73.83<br>($\sigma = 2.46$) | 58.88<br>($\sigma = 1.89$) | 77.91 | 60.95 | **69.43** |
| 2430-500-500-500-500-100, Layer 1 | 75.34<br>($\sigma = 2.85$) | 61.31<br>($\sigma = 2.21$) | 78.62 | 64.52 | **71.57** |
| 2430-500-500-500-500-100, Layer 2 | 73.05<br>($\sigma = 1.24$) | 59.19<br>($\sigma = 2.03$) | 76.49 | 61.53 | **69.01** |
| 2430-500-500-500-500-100, Layer 3 | 70.50<br>($\sigma = 1.02$) | 60.34<br>($\sigma = 1.83$) | 75.63 | 66.23 | **70.93** |
| 2430-500-500-500-500-100, Layer 4 | 69.34<br>($\sigma = 2.03$) | 60.26<br>($\sigma = 1.92$) | 73.92 | 66.66 | **70.29** |
| 2430-500-500-500-500-100, Layer 5 | 69.44<br>($\sigma = 2.57$) | 59.98<br>($\sigma = 1.92$) | 74.35 | 66.23 | **70.29** |
| Average: | 71.53<br>($\sigma = 2.60$) | 60.21<br>($\sigma = 0.76$) | 75.80 | 65.03 | **70.41** |
| Overall Average: | 72.70<br>($\sigma = 2.72$) | 58.95<br>($\sigma = 2.59$) | 77.59 | 60.92 | **69.25** |

With regard to hypothesis testing between *transfer learning* and *sparse coding*, it can be seen from Table 64 that in 10 of 27 cases, the results are significantly different with *p-values* less than the significance level. From Table 65 it is observed that in 3 of 27 cases, the classification accuracies are also significantly different. For these cases, the performance of *transfer learning* exceeds the baseline result.

In the case of applying hypothesis testing corresponding to different feature categories,

**Table 57:** Summary of average PTSD diagnostic accuracies across all the *transfer learning* experiments which have MFCC input features.

| **Method: Transfer Learning (Architecture) Input MFCC Features** | **Mean Segment-Wise Acc (%) on Youtube** | **Mean Segment-wise Acc (%) on Ohio** | **Mean Subject-wise Acc (%) on Youtube** | **Mean Subject-wise Acc (%) on Ohio** | **Overall Subject-wise Accuracy (%)** |
|---|---|---|---|---|---|
| 585-500-500-500-500-100, Layer 1 | 77.41<br>($\sigma = 2.69$) | 52.31<br>($\sigma = 2.04$) | 74.35 | 64.95 | **69.65** |
| 585-500-500-500-500-100, Layer 2 | 76.24<br>($\sigma = 1.98$) | 52.41<br>($\sigma = 3.26$) | 76.91 | 57.68 | **67.30** |
| 585-500-500-500-500-100, Layer 3 | 75.11<br>($\sigma = 3.05$) | 52.42<br>($\sigma = 3.53$) | 80.76 | 56.40 | **68.58** |
| 585-500-500-500-500-100, Layer 4 | 75.95<br>($\sigma = 2.78$) | 52.20<br>($\sigma = 3.83$) | 79.90 | 53.84 | **66.87** |
| 585-500-500-500-500-100, Layer 5 | 79.17<br>($\sigma = 2.26$) | 52.16<br>($\sigma = 2.43$) | 88.03 | 55.97 | **72.00** |
| Average: | 76.77<br>($\sigma = 1.57$) | 52.30<br>($\sigma = 0.11$) | 79.99 | 57.76 | **68.88** |
| 585-2000-2000-2000, Layer 1 | 78.94<br>($\sigma = 2.27$) | 53.14<br>($\sigma = 1.94$) | 88.03 | 56.46 | **63.02** |
| 585-2000-2000-2000, Layer 2 | 76.98<br>($\sigma = 1.97$) | 52.45<br>($\sigma = 1.53$) | 82.47 | 56.83 | **69.65** |
| 585-2000-2000-2000, Layer 3 | 76.53<br>($\sigma = 2.45$) | 51.82<br>($\sigma = 1.58$) | 81.18 | 55.97 | **68.58** |
| Average: | 77.48<br>($\sigma = 1.28$) | 52.47<br>($\sigma = 0.66$) | 83.75 | 56.42 | **67.01** |
| Overall Average: | 77.04<br>($\sigma = 1.41$) | 52.36<br>($\sigma = 0.37$) | 81.40 | 57.26 | **68.17** |

we can see from Table 66 that the absolute value of t-values for E only results are much larger than the rest of the t-values, and the p-values are within the range that enables the rejection of the null hypothesis. Using the significance level of 5%, we observed that the p-values for excitation features were clearly found to lie within the significance level and was declared to be statistically significant.

#### 6.3.2.2 WILCOXON SIGNED-RANK TEST

Since the normal distribution assumption of Paired t-test may not be true, we introduce a non-parametric statistical hypothesis test—Wilcoxon signed-rank test, which does not require any underlying distribution assumptions. For two paired samples, it is to assess whether the population mean ranks differ. After calculating the test statistic W, also called as the sum of the signed ranks, we compare it to the critical value from the reference table based on the degree of freedom. The null hypothesis is that there is no significant difference of the mean ranks in the two paired samples, which denotes that the W statistic is smaller than the critical value at a certain degree of freedom. However, if the test statistics W is greater than or equal to the

**Table 58:** Summary of best PTSD diagnostic accuracies by different models across all the *transfer learning* experiments which have input feature combination of prosodic, vocal-tract and excitation features.

| **Method** | **Frame Length (sec)** | **Frame Shift (sec)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Best Subject-Wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| 162-100-5, Layer 1 | 3.0 | 0.1 | 78.93 ($\sigma = 28.46$) | 54.02 ($\sigma = 28.72$) | 88.46 | 42.30 | **65.38** |
| 162-100-50, Layer 2 | 3.0 | 0.1 | 75.81 ($\sigma = 27.07$) | 53.08 ($\sigma = 28.04$) | 84.61 | 42.30 | **63.45** |
| | | Average: | 77.37 ($\sigma = 2.20$) | 53.55 ($\sigma = 0.66$) | 86.53 | 42.30 | **64.41** |
| 2430-100-50-2, Layer 1 | 3.0 | 1.0 | 77.04 ($\sigma = 35.97$) | 64.93 ($\sigma = 33.71$) | 80.76 | 69.23 | **74.99** |
| 2430-100-50-2, Layer 2 | 1.0 | 0.5 | 70.72 ($\sigma = 32.46$) | 59.14 ($\sigma = 22.86$) | 69.23 | 65.38 | **67.30** |
| | | Average: | 73.88 ($\sigma = 4.46$) | 62.03 ($\sigma = 4.09$) | 74.99 | 67.30 | **71.14** |
| 2430-1000-1000-50, Layer 1 | 1.0 | 1.0 | 79.11 ($\sigma = 36.14$) | 62.79 ($\sigma = 33.37$) | 80.76 | 69.23 | **74.99** |
| 2430-1000-1000-500, Layer 2 | 1.0 | 0.5 | 75.18 ($\sigma = 36.64$) | 57.73 ($\sigma = 32.37$) | 80.76 | 61.53 | **71.14** |
| 2430-1000-1000-500, Layer 3 | 3.0 | 0.5 | 73.69 ($\sigma = 36.86$) | 59.49 ($\sigma = 32.58$) | 80.76 | 65.38 | **73.07** |
| | | Average: | 75.99 ($\sigma = 2.80$) | 60.00 ($\sigma = 2.56$) | 80.76 | 65.38 | **73.06** |
| 2430-500-500-500-500-100, Layer 1 | 1.0 | 1.0 | 78.86 ($\sigma = 35.42$) | 62.80 ($\sigma = 32.97$) | 84.61 | 69.23 | **76.92** |
| 2430-500-500-500-500-100, Layer 2 | 3.0 | 0.5 | 74.14 ($\sigma = 36.11$) | 61.86 ($\sigma = 30.03$) | 76.92 | 69.23 | **73.07** |
| 2430-500-500-500-500-100, Layer 3 | 2.0 | 0.5 | 71.94 ($\sigma = 36.36$) | 62.34 ($\sigma = 28.73$) | 76.92 | 69.23 | **73.07** |
| 2430-500-500-500-500-100, Layer 4 | 2.0 | 0.5 | 71.26 ($\sigma = 35.61$) | 62.80 ($\sigma = 26.83$) | 76.92 | 73.07 | **74.99** |
| 2430-500-500-500-500-100, Layer 5 | 1.0 | 0.5 | 71.86 ($\sigma = 36.91$) | 60.25($\sigma = 23.59$) | 73.07 | 69.23 | **71.15** |
| | | Average: | 73.61 ($\sigma = 3.13$) | 62.01 ($\sigma = 1.05$) | 77.68 | 69.99 | **73.84** |

critical value, we would reject the null hypothesis, and there is considered to be a significant difference between the means of the two paired groups. Table 67 shows the results of this statistical non-parametric test.

**Table 59:** Summary of best PTSD diagnostic accuracies by different models across all the *transfer learning* experiments which have input MFCC features.

| **Method** | **Frame Length (sec)** | **Frame Shift (sec)** | **Segment-wise Acc (%) on Youtube** | **Segment-wise Acc (%) on Ohio** | **Subject-wise Acc (%) on Youtube** | **Subject-wise Acc (%) on Ohio** | **Overall Best Subject-Wise Accuracy (%)** |
|---|---|---|---|---|---|---|---|
| 585-500-500-500-500-100, Layer 1 | 3.0 | 0.1 | 80.69 ($\sigma$ = 25.95) | 54.39 ($\sigma$ = 25.22) | 84.61 | 61.53 | **73.07** |
| 585-500-500-500-500-100, Layer 2 | 3.0 | 0.5 | 80.02 ($\sigma$ = 22.95) | 54.58 ($\sigma$ = 24.02) | 80.76 | 61.53 | **71.14** |
| 585-500-500-500-500-100, Layer 3 | 3.0 | 0.1 | 78.11 ($\sigma$ = 21.68) | 55.11 ($\sigma$ = 20.12) | 84.61 | 57.69 | **71.15** |
| 585-500-500-500-500-100, Layer 4 | 3.0 | 0.1 | 79.03 ($\sigma$ = 24.68) | 54.82 ($\sigma$ = 21.64) | 84.61 | 53.84 | **69.22** |
| 585-500-500-500-500-100, Layer 5 | 3.0 | 0.5 | 81.87 ($\sigma$ = 25.28) | 55.58 ($\sigma$ = 25.70) | 88.46 | 61.53 | **74.99** |
| | | Average: | 79.94 ($\sigma = 1.45$) | 54.89 ($\sigma = 0.46$) | 84.61 | 59.22 | **71.91** |
| 585-2000-2000-2000, Layer 1 | 3.0 | 0.5 | 81.72 ($\sigma$ = 25.11) | 55.58 ($\sigma$ = 25.97) | 88.46 | 61.53 | **65.37** |
| 585-2000-2000-2000, Layer 2 | 3.0 | 0.1 | 79.31 ($\sigma$ = 27.19) | 53.90 ($\sigma$ = 25.90) | 84.61 | 57.69 | **66.60** |
| 585-2000-2000-2000, Layer 3 | 3.0 | 0.5 | 79.76 ($\sigma$ = 24.76) | 53.80 ($\sigma$ = 24.87) | 80.76 | 57.69 | **69.22** |
| | | Average: | 80.26 ($\sigma = 1.28$) | 54.42 ($\sigma = 1.0$) | 84.61 | 58.97 | **67.06** |

Still, only two pairs of data show significance. We then deduced that the original data is not normally distributed as we assumed, so we conducted Wilcoxon's signed-rank test as an alternative for paired t-test since it does not require the sample data to be normally distributed. The method is calculating the difference between two sets of data, as we did in paired t-test. Then we ranked them by the absolute value of the differences. After that, we multiplied the ranks by the signs of the corresponding original differences. For instance, a subject's segment-wise accuracy increased after excluding some features, and then the sign of difference must be positive. Similarly, if subject's segment-wise accuracy decreased, then the sign of difference must be negative. Finally, we sum up all the signed rank together to get the w test statistic and compare it to the critical value in Table 6 based on the degree of freedom, which is the number of subjects that have a sign in their difference, that is, compared to the subjects that have a change of 0 in testing accuracy. However, only E only tests for both Youtube and Ohio show significance as shown below in the table, for the w-value falls below the critical value for the w statistic. By

**Table 60:** Application of hypothesis testing to compare the *transfer learning* against SVM on Youtube subjects, for PTSD diagnosis. The architecture is 2430-1000-1000-500. There are 2430 input features. These features are a combination of vocal-tract, prosodic and excitation feature categories. The level of significance, $\alpha$ is 5%. P-values in boldface are significantly different results. The null hypothesis is that the accuracies from both methods come from a normal population distribution consisting of independent random samples with equal means and unknown variances. The alternative hypothesis is that the means are unequal.

| **PTSD Input and Network Architecture** | **Hidden Layer No.** | **Frame Size (sec)** | **Frame Shift (sec)** | **p-value** | **Segment-wise (subject-wise) Classification Accuracy (%) Transfer Learning** | **Segment-wise (subject-wise) Classification Accuracy (%) SVM** |
|---|---|---|---|---|---|---|
| Youtube (26 subjects) | 1 | 1.0 | 0.1 | **0.0379** | 73.94(76.92) | 56.04(57.69) |
| | | 1.0 | 0.5 | **0.4964** | 75.63(76.92) | 60.97(61.53) |
| | | 1.0 | 1.0 | **0.0342** | 79.11(80.76) | 53.57(53.84) |
| | | 2.0 | 0.1 | **0.0485** | 74.65(76.92) | 55.73(57.69) |
| | | 2.0 | 0.5 | **0.0499** | 75.83(76.92) | 56.34(57.69) |
| | | 2.0 | 1.0 | **0.0493** | 79.50(80.76) | 59.53(61.53) |
| | | 3.0 | 0.1 | **0.0472** | 74.87(76.92) | 53.03(53.84) |
| | | 3.0 | 0.5 | **0.0486** | 75.71(76.92) | 61.93(61.53) |
| | | 3.0 | 1.0 | **0.0201** | 78.92(80.76) | 50.12(50.00) |
| Youtube (26 subjects) | 2 | 1.0 | 0.1 | **0.0491** | 72.95(76.92) | 56.04(57.69) |
| | | 1.0 | 0.5 | 0.4979 | 75.18(80.76) | 60.97(61.53) |
| | | 1.0 | 1.0 | **0.0448** | 71.32(73.07) | 53.57(53.84) |
| | | 2.0 | 0.1 | **0.0491** | 73.78(76.92) | 55.73(57.69) |
| | | 2.0 | 0.5 | 0.0903 | 75.72(80.76) | 56.34(57.69) |
| | | 2.0 | 1.0 | 0.2735 | 72.04(76.92) | 59.53(61.53) |
| | | 3.0 | 0.1 | **0.0495** | 74.12(80.76) | 53.05(53.84) |
| | | 3.0 | 0.5 | 0.2267 | 74.98(80.76) | 61.93(61.53) |
| | | 3.0 | 1.0 | 0.0703 | 71.23(69.23) | 50.12(50.00) |
| Youtube (26 subjects) | 3 | 1.0 | 0.1 | 0.1240 | 70.18(76.92) | 56.04(57.69) |
| | | 1.0 | 0.5 | 0.2658 | 73.36(80.76) | 60.97(61.53) |
| | | 1.0 | 1.0 | 0.1278 | 71.44(76.92) | 53.57(53.84) |
| | | 2.0 | 0.1 | 0.1581 | 71.24(80.76) | 55.73(57.69) |
| | | 2.0 | 0.5 | 0.1208 | 74.13(80.76) | 56.34(57.69) |
| | | 2.0 | 1.0 | 0.3915 | 69.30(76.92) | 59.53(61.53) |
| | | 3.0 | 0.1 | 0.0907 | 71.83(80.76) | 53.03(53.84) |
| | | 3.0 | 0.5 | 0.4251 | 70.62(76.92) | 61.93(61.53) |
| | | 3.0 | 1.0 | 0.0980 | 68.92(37.74) | 50.12(50.36) |

comparing the experimental w-values with the critical values for the w-statistic, we can see that only when excitation features remain, the critical value surpasses the w-value, which suggests significance of this result.

Since all of the statistical tests show that the reduction of both prosodic features and

**Table 61:** Application of hypothesis testing to compare the *transfer learning* against SVM, on Ohio subjects for PTSD diagnosis. The architecture is 2430-1000-1000-500. There are 2,430 input features. These features are a combination of vocal, prosodic and excitation feature categories. The level of significance, $\alpha$ is 5%. P-values in boldface are significantly different results.The null hypothesis is that the accuracies from both methods come from a normal population distribution consisting of independent random samples with equal means and unknown variances. The alternative hypothesis is that the means are unequal.

| **PTSD Input and Network Architecture** | **Hidden Layer No.** | **Frame Size (sec)** | **Frame Shift (sec)** | **p-value** | **Segment-wise (subject-wise) Classification Accuracy (%) Transfer Learning** | **Segment-wise (subject-wise) Classification Accuracy (%) SVM** |
|---|---|---|---|---|---|---|
| Ohio (26 subjects) | 1 | 1.0 | 0.1 | 0.5895 | 57.82(61.53) | 52.13(53.84) |
| | | 1.0 | 0.5 | 0.3245 | 60.23(65.38) | 69.78(76.92) |
| | | 1.0 | 1.0 | 0.3444 | 62.79(69.23) | 52.98(53.84) |
| | | 2.0 | 0.1 | 0.5098 | 58.20(57.69) | 51.28(53.84) |
| | | 2.0 | 0.5 | 0.3771 | 62.40(61.53) | 52.95(53.84) |
| | | 2.0 | 1.0 | **0.0484** | 63.05(69.23) | 47.03(5.00) |
| | | 3.0 | 0.1 | 0.6458 | 58.67(53.84) | 53.57(57.69) |
| | | 3.0 | 0.5 | **0.0490** | 62.96(61.53) | 42.65(46.15) |
| | | 3.0 | 1.0 | **0.0429** | 63.13(65.38) | 40.81(42.30) |
| Ohio (26 subjects) | 2 | 1.0 | 0.1 | 0.6907 | 56.24(57.69) | 52.13(53.84) |
| | | 1.0 | 0.5 | 0.2269 | 57.73(61.53) | 69.78(76.92) |
| | | 1.0 | 1.0 | 0.7656 | 55.96(57.69) | 52.98(49.61) |
| | | 2.0 | 0.1 | 0.5539 | 57.57(61.53) | 51.28(53.84) |
| | | 2.0 | 0.5 | 0.5711 | 59.04(61.53) | 52.95(53.84) |
| | | 2.0 | 1.0 | 0.3617 | 57.43(57.69) | 47.03(50.00) |
| | | 3.0 | 0.1 | 0.7013 | 57.91(61.53) | 53.57(57.69) |
| | | 3.0 | 0.5 | 0.1357 | 58.92(61.53) | 42.65(46.15) |
| | | 3.0 | 1.0 | 0.1629 | 55.94(53.84) | 40.81(42.30) |
| Ohio (26 subjects) | 3 | 1.0 | 0.1 | 0.6858 | 56.29(57.69) | 52.13(53.84) |
| | | 1.0 | 0.5 | 0.2541 | 58.52(61.53) | 69.78(41.25) |
| | | 1.0 | 1.0 | 0.6767 | 57.26(61.53) | 52.98(53.84) |
| | | 2.0 | 0.1 | 0.5322 | 57.79(57.69) | 51.28(53.84) |
| | | 2.0 | 0.5 | 0.5577 | 59.25(61.53) | 52.95(53.84) |
| | | 2.0 | 1.0 | 0.3250 | 57.95(57.69) | 47.03(50.00) |
| | | 3.0 | 0.1 | 0.6494 | 58.64(61.53) | 53.57(57.69) |
| | | 3.0 | 0.5 | **0.0482** | 59.49(65.38) | 42.65(46.15) |
| | | 3.0 | 1.0 | 0.1051 | 58.77(61.53) | 40.81(42.30) |

vocal-tract features has a considerable effect on the accuracy of classification, we concluded that excitation features are the least significant among all three categories [91]. However, considering there are only 2 features in this category, while there are 12 in prosodic features and 40 in vocal-tract features, there may be biases confounded in the results. Also, we noticed that after excluding the vocal-tract features, the performance rate increased by a little for Youtube data.

**Table 62:** Application of hypothesis testing to compare the *transfer learning* against DBN on Youtube subjects, for PTSD diagnosis. The architecture is 2430-1000-1000-500 for transfer learning and 2430-1000-1000-500-2 for DBN. There are 2430 input features. These features are a combination of vocal-tract, prosodic and excitation feature categories. The level of significance, $\alpha$ is 5%. P-values in boldface are significantly different results. The null hypothesis is that the accuracies from both methods come from a normal population distribution consisting of independent random samples with equal means and unknown variances. The alternative hypothesis is that the means are unequal.

| **PTSD Input and Network Architecture** | **Hidden Layer No.** | **Frame Size (sec)** | **Frame Shift (sec)** | **p-value** | **Segment-wise (subject-wise) Classification Accuracy (%) Transfer Learning** | **Segment-wise (subject-wise) Classification Accuracy (%) DBN** |
|---|---|---|---|---|---|---|
| Youtube (26 subjects) | 1 | 1.0 | 0.1 | 0.0734 | 73.94(76.92) | 58.85(61.53) |
| | | 1.0 | 0.5 | 0.0890 | 75.63(76.92) | 59.87(61.53) |
| | | 1.0 | 1.0 | **0.0428** | 79.11(80.76) | 60.05(61.53) |
| | | 2.0 | 0.1 | 0.2194 | 74.65(76.92) | 64.69(76.92) |
| | | 2.0 | 0.5 | 0.2666 | 75.83(76.92) | 66.72(76.92) |
| | | 2.0 | 1.0 | **0.0135** | 79.50(80.76) | 58.35(57.69) |
| | | 3.0 | 0.1 | **0.0016** | 74.87(76.92) | 47.65(34.61) |
| | | 3.0 | 0.5 | 0.1487 | 75.71(76.92) | 63.70(76.92) |
| | | 3.0 | 1.0 | 0.0783 | 78.92(80.76) | 61.81(69.23) |
| Youtube (26 subjects) | 2 | 1.0 | 0.1 | **0.0489** | 72.95(76.92) | 58.85(61.53) |
| | | 1.0 | 0.5 | 0.1008 | 75.18(80.76) | 59.87(61.53) |
| | | 1.0 | 1.0 | 0.2413 | 71.32(73.07) | 60.05(61.53) |
| | | 2.0 | 0.1 | 0.2638 | 73.78(76.92) | 64.69(76.92) |
| | | 2.0 | 0.5 | 0.2585 | 75.72(80.76) | 66.72(76.92) |
| | | 2.0 | 1.0 | 0.1191 | 72.04(76.92) | 58.35(57.69) |
| | | 3.0 | 0.1 | **0.0021** | 74.12(80.76) | 47.65(34.61) |
| | | 3.0 | 0.5 | 0.1694 | 74.98(80.76) | 63.74(76.92) |
| | | 3.0 | 1.0 | 0.3376 | 71.23(69.23) | 61.81(69.23) |
| Youtube (26 subjects) | 3 | 1.0 | 0.1 | **0.0494** | 70.18(76.92) | 58.85(61.53) |
| | | 1.0 | 0.5 | **0.0492** | 73.36(80.76) | 59.87(61.53) |
| | | 1.0 | 1.0 | 0.2204 | 71.47(76.92) | 60.05(61.53) |
| | | 2.0 | 0.1 | 0.4229 | 71.24(80.76) | 64.69(76.92) |
| | | 2.0 | 0.5 | 0.3489 | 74.13(80.76) | 66.72(76.92) |
| | | 2.0 | 1.0 | 0.2002 | 69.30(76.92) | 58.35(57.69) |
| | | 3.0 | 0.1 | **0.0043** | 71.83(80.76) | 47.65(34.61) |
| | | 3.0 | 0.5 | 0.3968 | 70.62(76.92) | 63.70(76.92) |
| | | 3.0 | 1.0 | 0.4562 | 68.92(69.23) | 61.81(69.23) |

It may be by chance, but it may also imply that there are other lurking variables hidden in the scenes. All three tests suggested that excitation features are the least important among the three categories. We rejected all three null hypotheses when comparing the control group with all features present and the experimental group with only excitation features present. It

**Table 63:** Application of hypothesis testing to compare the *transfer learning* against DBN on Ohio subjects, for PTSD diagnosis. The architecture is 2430-1000-1000-500 for *transfer learning* and 2430-1000-1000-500-2 for DBN. There are 2430 input features. These features are a combination of vocal-tract, prosodic and excitation feature categories. The level of significance, $\alpha$ is 5%. P-values in boldface are significantly different results. The null hypothesis is that the accuracies from both methods come from a normal population distribution consisting of independent random samples with equal means and unknown variances. The alternative hypothesis is that the means are unequal.

| **PTSD Input and Network Architecture** | **Hidden Layer No.** | **Frame Size (sec)** | **Frame Shift (sec)** | **p-value** | **Segment-wise (subject-wise) Classification Accuracy (%) Transfer Learning** | **Segment-wise (subject-wise) Classification Accuracy (%) DBN** |
|---|---|---|---|---|---|---|
| Ohio (26 subjects) | 1 | 1.0 | 0.1 | 0.0938 | 57.82(61.53) | 47.63(46.15) |
| | | 1.0 | 0.5 | 0.0936 | 60.23(65.38) | 50.37(53.84) |
| | | 1.0 | 1.0 | 0.1524 | 62.79(69.23) | 52.97(57.69) |
| | | 2.0 | 0.1 | 0.0672 | 58.20(57.69) | 48.43(42.30) |
| | | 2.0 | 0.5 | **0.0497** | 62.40(61.53) | 46.64(46.15) |
| | | 2.0 | 1.0 | 0.1249 | 63.05(69.23) | 53.84(65.38) |
| | | 3.0 | 0.1 | 0.2253 | 58.67(53.84) | 50.83(50.00) |
| | | 3.0 | 0.5 | 0.1532 | 62.96(61.53) | 52.15(53.84) |
| | | 3.0 | 1.0 | 0.1909 | 63.13(65.38) | 53.08(61.53) |
| Ohio (26 subjects) | 2 | 1.0 | 0.1 | 0.1094 | 56.24(57.69) | 47.63(46.15) |
| | | 1.0 | 0.5 | 0.2330 | 57.73(61.53) | 50.37(53.84) |
| | | 1.0 | 1.0 | 0.6711 | 55.96(57.69) | 52.97(57.69) |
| | | 2.0 | 0.1 | 0.0920 | 57.57(61.53) | 48.43(42.30) |
| | | 2.0 | 0.5 | 0.1319 | 59.04(61.53) | 46.64(46.15) |
| | | 2.0 | 1.0 | 0.5763 | 57.43(57.69) | 53.84(65.38) |
| | | 3.0 | 0.1 | 0.2896 | 57.91(61.53) | 50.83(50.00) |
| | | 3.0 | 0.5 | 0.3717 | 58.92(61.53) | 52.15(53.84) |
| | | 3.0 | 1.0 | 0.7266 | 55.94(53.84) | 53.08(61.53) |
| Ohio (26 subjects) | 3 | 1.0 | 0.1 | 0.1202 | 56.29(57.69) | 47.63(46.15) |
| | | 1.0 | 0.5 | 0.1666 | 58.52(61.53) | 50.37(53.84) |
| | | 1.0 | 1.0 | 0.5495 | 57.26(61.53) | 52.97(57.69) |
| | | 2.0 | 0.1 | 0.0757 | 57.79(57.69) | 48.43(42.30) |
| | | 2.0 | 0.5 | **0.0493** | 59.25(61.53) | 45.87(46.15) |
| | | 2.0 | 1.0 | 0.5112 | 57.95(57.69) | 53.84(65.38) |
| | | 3.0 | 0.1 | 0.2250 | 58.64(61.53) | 50.83(50.00) |
| | | 3.0 | 0.5 | 0.3232 | 59.49(65.38) | 52.15(53.84) |
| | | 3.0 | 1.0 | 0.4668 | 58.77(61.53) | 53.08(61.53) |

is because the data obtained from this experimental group showed significant diminishment in the classification performance that could not have happened by chance. When we only kept prosodic features or vocal-tract features, the classification performances did not show significant changes, which means the changes observed can be explained by chances. Moreover, when

**Table 64:** Application of hypothesis testing to compare the *transfer learning* against sparse coding on Youtube subjects, for PTSD diagnosis. The architecture is 2430-1000-1000-500. There are 2430 input features. These features are a combination of vocal-tract, prosodic and excitation feature categories. The level of significance, $\alpha$ is 5%. P-values in boldface are significantly different results. The null hypothesis is that the accuracies from both methods come from a normal population distribution consisting of independent random samples with equal means and unknown variances. The alternative hypothesis is that the means are unequal.

| **PTSD Input and Network Architecture** | **Hidden Layer No.** | **Frame Size (sec)** | **Frame Shift (sec)** | **p-value** | **Segment-wise (subject-wise) Classification Accuracy (%) Transfer Learning** | **Segment-wise (subject-wise) Classification Accuracy (%) Sparse Coding** |
|---|---|---|---|---|---|---|
| Youtube (26 subjects) | 1 | 1.0 | 0.1 | 0.1326 | 73.94(76.92) | 66.55(76.92) |
| | | 1.0 | 0.5 | 0.0924 | 75.63(76.92) | 64.31(65.38) |
| | | 1.0 | 1.0 | **0.0089** | 79.11(80.76) | 61.88(61.53) |
| | | 2.0 | 0.1 | 0.1255 | 74.65(76.92) | 65.04(73.07) |
| | | 2.0 | 0.5 | 0.0589 | 75.83(76.92) | 62.19(61.53) |
| | | 2.0 | 1.0 | **0.0310** | 79.50(80.76) | 62.84(57.69) |
| | | 3.0 | 0.1 | 0.0821 | 74.87(76.92) | 64.10(73.07) |
| | | 3.0 | 0.5 | 0.1043 | 75.71(76.92) | 62.98(61.53) |
| | | 3.0 | 1.0 | **0.0165** | 78.92(80.76) | 58.18(50.00) |
| Youtube (26 subjects) | 2 | 1.0 | 0.1 | 0.1658 | 72.95(76.92) | 66.55(76.92) |
| | | 1.0 | 0.5 | 0.3069 | 75.18(80.76) | 64.31(65.38) |
| | | 1.0 | 1.0 | 0.1544 | 71.32(73.07) | 61.88(61.53) |
| | | 2.0 | 0.1 | 0.1518 | 73.78(76.92) | 65.04(73.07) |
| | | 2.0 | 0.5 | 0.0603 | 75.72(80.76) | 62.19(61.53) |
| | | 2.0 | 1.0 | 0.2127 | 72.04(76.92) | 62.84(57.69) |
| | | 3.0 | 0.1 | 0.1079 | 74.12(80.76) | 64.10(73.07) |
| | | 3.0 | 0.5 | 0.1233 | 74.98(80.76) | 62.98(61.53) |
| | | 3.0 | 1.0 | 0.1038 | 71.23(69.23) | 58.18(50.00) |
| Youtube (26 subjects) | 3 | 1.0 | 0.1 | 0.4018 | 70.18(76.92) | 66.55(76.92) |
| | | 1.0 | 0.5 | 0.1565 | 73.36(80.76) | 64.31(65.38) |
| | | 1.0 | 1.0 | 0.1004 | 71.44(76.92) | 61.88(61.53) |
| | | 2.0 | 0.1 | 0.2865 | 71.24(80.76) | 65.04(73.07) |
| | | 2.0 | 0.5 | 0.0913 | 74.13(80.76) | 62.19(61.53) |
| | | 2.0 | 1.0 | 0.3425 | 69.30(76.92) | 62.84(57.69) |
| | | 3.0 | 0.1 | 0.1883 | 71.83(80.76) | 64.10(73.07) |
| | | 3.0 | 0.5 | 0.1590 | 70.62(76.92) | 62.98(61.53) |
| | | 3.0 | 1.0 | 0.1516 | 68.92(69.23) | 58.18(50.00) |

excitation features were excluded, the performance improved slightly as shown in Tables 46 and 47. Though the changes are not significant, we can still deduce from the results that this feature category might be a disturbing factor to the overall classification performance. Tables 46 and 47 included both segment-wise accuracy and subject-wise accuracies during testing, providing more

**Table 65:** Application of hypothesis testing to compare the *transfer learning* against sparse coding on Ohio subjects, for PTSD diagnosis. The architecture is 2430-1000-1000-500. There are 2430 input features. These features are a combination of vocal-tract, prosodic and excitation feature categories. The level of significance, $\alpha$ is 5%. P-values in boldface are significantly different results. The null hypothesis is that the accuracies from both methods come from a normal population distribution consisting of independent random samples with equal means and unknown variances. The alternative hypothesis is that the means are unequal.

| **PTSD Input and Network Architecture** | **Hidden Layer No.** | **Frame Size (sec)** | **Frame Shift (sec)** | **p-value** | **Segment-wise (subject-wise) Classification Accuracy (%) Transfer Learning** | **Segment-wise (subject-wise) Classification Accuracy (%) Sparse Coding** |
|---|---|---|---|---|---|---|
| Ohio (26 subjects) | 1 | 1.0 | 0.1 | 0.6503 | 57.82(61.53) | 59.54(69.23) |
| | | 1.0 | 0.5 | 0.0685 | 60.23(65.38) | 64.95(73.07) |
| | | 1.0 | 1.0 | 0.5683 | 62.79(69.23) | 65.56(69.23) |
| | | 2.0 | 0.1 | 0.2958 | 58.20(57.69) | 62.56(65.38) |
| | | 2.0 | 0.5 | 0.6054 | 62.40(61.53) | 64.31(61.53) |
| | | 2.0 | 1.0 | 0.7795 | 63.05(69.23) | 64.31(69.23) |
| | | 3.0 | 0.1 | 0.2306 | 58.67(53.84) | 63.81(73.07) |
| | | 3.0 | 0.5 | 0.8820 | 62.96(61.53) | 63.48(65.38) |
| | | 3.0 | 1.0 | 0.7150 | 63.13(65.38) | 61.40(73.07) |
| Ohio (26 subjects) | 2 | 1.0 | 0.1 | 0.3498 | 56.24(57.69) | 59.54(69.23) |
| | | 1.0 | 0.5 | 0.1051 | 57.73(61.53) | 64.95(65.38) |
| | | 1.0 | 1.0 | 0.2815 | 55.96(57.69) | 65.56(61.53) |
| | | 2.0 | 0.1 | 0.1743 | 57.57(61.53) | 62.56(65.38) |
| | | 2.0 | 0.5 | 0.1273 | 59.04(61.53) | 64.31(61.53) |
| | | 2.0 | 1.0 | 0.0987 | 57.43(57.69) | 64.31(69.23) |
| | | 3.0 | 0.1 | 0.1394 | 57.91(61.53) | 63.81(73.07) |
| | | 3.0 | 0.5 | 0.1254 | 58.92(61.53) | 63.48(65.38) |
| | | 3.0 | 1.0 | 0.1723 | 55.94(53.84) | 61.40(73.07) |
| Ohio (26 subjects) | 3 | 1.0 | 0.1 | 0.3656 | 56.29(57.69) | 59.54(69.23) |
| | | 1.0 | 0.5 | 0.1430 | 58.52(61.53) | 64.95(65.58) |
| | | 1.0 | 1.0 | 0.1066 | 57.26(61.53) | 65.56(61.53) |
| | | 2.0 | 0.1 | 0.1938 | 57.79(57.69) | 62.56(65.38) |
| | | 2.0 | 0.5 | 0.1265 | 59.25(61.53) | 64.31(61.53) |
| | | 2.0 | 1.0 | 0.1625 | 57.95(57.69) | 64.31(69.23) |
| | | 3.0 | 0.1 | 0.1874 | 58.64(61.53) | 63.81(73.07) |
| | | 3.0 | 0.5 | 0.1816 | 59.49(65.38) | 63.48(65.38) |
| | | 3.0 | 1.0 | 0.5192 | 58.77(61.53) | 61.40(73.07) |

perceptible evidence of this implication.

**Table 66:** Paired t-test results on segment-wise accuracies utilizing feature selection based on transfer learning. The architecture was 2430-500-500-500-500-100.

| **Segment-wise Accuracies** | | **P Out** | **V Out** | **E Out** | **P Only** | **V Only** | **E Only** |
|---|---|---|---|---|---|---|---|
| Youtube | t-value | -1.53 | 0.14 | 0.87 | -0.04 | -1.00 | -2.98 |
| | p-value | 0.14 | 0.89 | 0.39 | 0.97 | 0.33 | **0.01** |
| Ohio | t-value | 0.80 | 0.05 | 1.57 | 0.82 | 1.55 | -2.75 |
| | p-value | 0.43 | 0.96 | 0.13 | 0.42 | 0.13 | 0.01 |

**Table 67:** Wilcoxon signed-rank test results on segment-wise accuracies utilizing feature selection based on transfer learning. The architecture was 2430-500-500-500-500-100.

| **Segment-wise Accuracies** | | **P Out** | **V Out** | **E Out** | **P Only** | **V Only** | **E Only** |
|---|---|---|---|---|---|---|---|
| Youtube | w-value | 25 | 111 | 35 | 115 | 49 | **55** |
| | deg. of freedom | 14 | 21 | 12 | 21 | 15 | 24 |
| | critical value for w-statistic | 21 | 58 | 13 | 58 | 25 | **81** |
| Ohio | w-value | 125 | 132 | 103 | 109 | 93 | **73** |
| | deg. of freedom | 23 | 23 | 22 | 23 | 23 | 25 |
| | critical value for w-statistic | 73 | 73 | 65 | 73 | 73 | **89** |

## 6.4 CONCLUSION OF THE PROPOSED APPROACH

*Transfer learning* achieved a best of 84.62% for Youtube and 73.08% for Ohio. In the majority of the cases using transfer learning, the first hidden layer of features has the best performance compared to the other hidden layers in the network. This could be attributed to the fact that lower layers have more primitive feature detectors that are more adapted to the input data. The hypothesis tests shows that in more than 31% of the cases that were tested, *transfer learning* performed significantly better than the baseline. *Transfer learning* also showed significantly better performance compared to the *deep belief network* method in nearly 19% of the cases that were tested. Comparing *transfer learning* with *sparse coding*, nearly 6% of the cases showed *transfer learning* to perform significantly better than *sparse coding*. When comparing between *transfer learning* with SVM, in almost 92% of the cases where the difference was not significant, *transfer learning* had a better accuracy than SVM. When comparing *transfer learning* against DBN, in 100% of the cases where the difference was not significant, *transfer learning* outperformed the DBN. Between *transfer learning* and *sparse coding*, in nearly 49% of the cases where the difference was not significant, *transfer learning* had better accuracy

**Table 68:** The critical value table based on degree of freedom and significance level.

| Two Tailed significance levels | | | |
|---|---|---|---|
| **deg. of freedom** | **0.05** | **0.02** | **0.01** |
| 12 | 14 | 10 | 7 |
| 13 | 17 | 13 | 10 |
| 14 | 21 | 16 | 13 |
| 15 | 25 | 20 | 16 |
| 16 | 30 | 24 | 20 |
| 17 | 35 | 28 | 23 |
| 18 | 40 | 33 | 28 |
| 19 | 46 | 38 | 32 |
| 20 | 52 | 43 | 38 |
| 21 | 59 | 49 | 43 |
| 22 | 66 | 56 | 49 |
| 23 | 73 | 62 | 55 |
| 24 | 81 | 69 | 61 |
| 25 | 89 | 77 | 68 |

than *sparse coding*. The proposed transfer learning method showed significant difference when compared to all other models in the proposed method. The overall effectiveness of *transfer learning* for PTSD detection is evident compared to the baseline and the other models in the proposed method.

In the feature selection based experiments, we conducted transfer learning through DBN to identify whether a patient has PTSD based on speech, and obtained up to 80% accuracy for the PTSD data set. The importance of the features used in order to eliminate the disturbing factors were also investigated, by removing specific features and comparing the results. Finally, we concluded that excitation feature category is the least significant as indicated by multiple statistical tests.

# CHAPTER 7

# CONCLUSIONS

An average subject-wise accuracy of 73.07% on Youtube and 56.53% on Ohio was achieved by SVM. *Sparse coding* achieved average subject-wise accuracies of 81.61% on *youtube* and 57.26% on Ohio. The mean subject-wise accuracy of 79.90% on Youtube and 63.67% on Ohio was obtained by the *deep belief network*. The architecture was 585-2000-2000-2000-2. There was a mean subject-wise accuracy of 75.63% on Youtube and 66.23% on Ohio using the *transfer learning* strategy was achieved by the third hidden layer. The corresponding architecture was 2430-500-500-500-500-100. A mean subject-wise accuracy of 84.62% for Youtube and 73.08% for Ohio after feature category selection was achieved by the first hidden layer in *transfer learning*. The feature category selection was carried out by zeroing the *excitation* feature category. The architecture was 2430-500-500-500-500-100.

In this dissertation, an efficient speech-driven *sparse coding* framework was developed for emotion recognition which did not exist before. The proposed system, evaluated on the SUSAS data set, outperformed other state-of-the-art algorithms. A speech-driven *sparse coding* and *deep belief net* framework was developed for PTSD detection for the first time. It addressed the limitation of current clinical diagnostic methods which heavily depend on assessment based on structured interviews, conducted in clinics. Novel feature extraction techniques were performed for PTSD detection. Excitation features were found not to be useful whereas the vocal-tract and prosodic feature categories proved to be superior in detecting PTSD. The small data size challenge was resolved by adopting a *transfer learning* strategy. *Transfer learning* achieved statistically significant performances compared to the other models in the proposed method and proved its overall effectiveness in PTSD detection. The proposed PTSD detection system surpassed the current clinical diagnostic accuracy. Overall, the proposed models proved to be promising and are recommended for PTSD diagnosis.

This work could be extended in the future to include application of deep convolutional networks for emotion recognition and PTSD diagnosis. *Transfer Learning* could be applied to the deep convolutional networks. In addition, recurrent neural networks may be explored in the context of emotion recognition and PTSD diagnosis.

# BIBLIOGRAPHY

[1] S. R. Krothapalli, S. G. Koolagudi, “Characterization and recognition of emotions from speech using excitation source information, ” *Int. J. Speech Technol.*, vol. 16, no. 2, pp. 181-201, 2012.

[2] R. A. Calvo, and S. D’Mello, “Affect detection: An interdisciplinary review of models, methods, and their applications,“ *IEEE Trans. Affective. Comput.*, vol. 1, no. 1, pp. 18-37, 2010

[3] S. Casale, A. Russo, and S. Serrano, “Multistyle classification of speech under stress using feature subset selection based on genetic algorithms,” *Speech Commun.*, vol. 49, no. 10-11, pp. 801-810, 2007.

[4] A. Nogueiras, A. Moreno, A. Bonafonte, and J. B. Marino, “Speech emotion recognition using hidden markov models, ” *Speech Commun.*, vol. 41, no. 4, pp. 603-623, 2003.

[5] S. G. Koolagudi, K. S. Rao, “Emotion recognition from speech: A review,” *Int. J. Speech Technol.*, vol. 15, no. 2, pp. 99-117, 2012.

[6] E. Bozkurt, E. Erzin, C. E. Erdem, A. T. Erdem, “Improving automatic recognition from speech signals,” *INTERSPEECH*, pp. 324-327, 2009.

[7] A. Chauhan, S. G. Koolagudi, S. Kafley, and K. S. Rao, “Emotion recognition using LP residual,” *Proc. IEEE Tech Sym Conf.*, pp. 255-261, 2010.

[8] S. Yildrim, M. Bulut, C. M. Lee, A. Kazemzadeh, C. Busso, Z. Deng, S. Lee, and S. Narayanan, “An acoustic study of emotions expressed in speech”, *Int. Conf. Spoken Language Process.*, pp. 2193-2196, 2004.

[9] F. Dellaert, T. Polzin, and A. Waibel, “Recognizing emotions in speech,” *Int. Conf. Spoken Language Process.*, pp. 1970-1973, 1996.

[10] E. Moore II, M. A. Clements, J. W. Peifer and L. Weisser, ”Critical Analysis of the Impact of Glottal Features in the Classification Of Clinical Depression in Speech“, *IEE Trans. on Biomed. Engineering*, vol 55. pp. 96-106, 2008.

[11] R. L.Rabiner, ”Digital Processing of Speech Signals“, *Pearson Education*, 1978.

[12] M. E. Ayadi, M. S. Kamel and F. Karray, ”Survey on Speech Emotion Recognition: Features, Classification Schemes, and Databases,“ *Pattern Recognition*, vol. 44, pp. 572-587, 2011.

[13] T. L. Nwe, S. W. Foo and L. C. DeSilva ”Speech Emotion Recognition Using Hidden Markov Models,“ *Speech Commun.*, vol. 41, pp. 603-623, 2003.

[14] C. M. Lee, S. Yildrim, M. Bulut and A. Kazemzadeh, ”Emotion Recognition Based On Phoneme Classes“, *ICSLP*, 2004.

[15] L. Leinonen, T. Hiltunen, I. Linnankoski and M. J. Laakso, ”Expression Of Emotional Motivational Connotations With a One Word Utterance,“ *Journal Acoust. Soc. Am.*, vol. 102, no. (3), pp. 1853-1863, 1997.

[16] I. Luengo, E. Navas, I. Hernaez, and J. Sanchez, ”Automatic Emotion Recognition using Prosodic Parameters,“ *Journal Of InterSpeech*, pp. 493-496, 2005.

[17] S. Wu, T. H. Falk and W. Y. Chan, ”Automatic Recognition Of Speech Emotion Using Long-Term Spectro-Temporal Features“, *16th Int'l Conf. Digital Signal Process.*, 2009.

[18] T. L. Nwe, S. W. Foo and L. C. DesSilva, ”Classification Of Stress In Speech Using Linear And Non-Linear Features,“ *Int'l Conf. Acoustics Speech and Signal Proc.*, 2003.

[19] D. Neiberg, K. Elenius, I. Karlsson and K. Laskowski, ”Emotion Recognition in Spontaneous Speech Using GMMs,“ *Int'l Conf. Spoken Language Process.* , 2006.

[20] I. R. Murray, J. L. Arnott and E. A. Rohwer, ”Emotional Stress In Synthetic Speech: Progress And Future Directions“, *Speech Commun.*, vol. 20, pp. 85-91, 1996.

[21] E. Bozkurt, E. Erzin, C. E. Erdem and A. T. Erdem, ”Improving Automatic Emotion Recognition from Speech Signals“, *InterSpeech Emotion Challenge*, 2009.

[22] K. Guojun Zhou, J. L¿ Hansen and J. F. Kaiser, ”Nonlinear Feature Based Classification of Speech Under Stress,“ *IEEE Trans. Speech and Audio Process.*, vol. 9, no. 3, pp. 201-216, 2001.

[23] S. G. Koolagudi, S. Devyilal, B. Chawla and A. Barthwal, ”Recognition Of Emotions From Speech Using Excitation Source Features“, *Int'l Conf. Modeling, Optimization and Computing*, 2012.

[24] C. M. Lee, S. Narayanan and R. Pieraccini, ”Recognition of Negative Emotions from the Speech Signal,“ *IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 240-243, 2001.

[25] M. E. Hoque, M. Yeasin and M. M. Louwerse, ”Robust Recognition of Emotion from Speech,“ *Springer Verlag*, vol. 4133, pp. 42-53, 2006.

[26] A. I. Iliev, M. S. Scordillis, J. P. Papa and A. X. Falcao, ”Spoken emotion recognition through optimum-path forest classification using glottal features“, *Computer Speech And Language*, vol. 24, pp. 445-460, 2010.

[27] M. Lugger and B. Yang, ”The Relevance Of Voice Quality Features in Speaker Independent Emotion Recognition,“ in *Int'l Conf. Acoustics Speech and Signal Process.*, 2007.

[28] T. Iliou, C. N. Anagnostopoulos, ”Comparison Of Different Classifiers for Emotion Recognition,“ *Panhellenic Conf. on Informatics, IEEE Computer Society*, 2009.

[29] A. Nilsonne and L. Wetterberg, ”Acoustic Analysis Of Speech Variables During Depression And After Improvement,“ *Acta Psychiatr Scand.*, 1987.

[30] B. T. Harel, M. S. Cannizzaro, H. Cohen, N. Reilly and P. J. Snyder, ”Acoustic characteristics of Parkinsonian speech: a potential biomarker of early disease progression and Treatment,“ *Journal of Neurolinguistics*, vol. 17, pp. 439-453, 2004.

[31] B. T. Harel, M. S. Cannizzaro, H. Cohen, N. Reilly and P. J. Snyder, ”Voice acoustical measurement of the severity of major depression,“ *Brain And Cognition*, vol. 56, pp. 30-35, 2004.

[32] Y. Kim, H. Lee and E. M. Provost, ”Deep Learing For Robust Feature Generation In AudioVisual Emotion Recognition,“ *Int'l Conf. Acoustics Speech and Signal Process.*, 2013.

[33] A. Stuhlsatz, C. Meyer, F. Eyben, T. Zielke, G. Meier and B. Schuller, ”Deep Neural Networks for Acoustic Emotion Recognition Raising the Benchmarks,“ *Int'l Conf. Acoustics Speech and Signal Process.* 2011.

[34] E. B. Foa, L. Cashman, L. Jaycox and K. Perry, ”The Validation of a Self-Report Measure of Post Traumatic Stress Disorder: The Posttraumatic Diagnostic Scale“, *Psychological Assessment*, vol. 9, pp. 445-451, 1997.

[35] M. J. Friedman, ”PTSD History and Overview“,
http://www.ptsd.va.gov/professional/PTSD-overview/ptsd-overview.asp Accessed on: July, 10, 2016.

[36] ”National Institute Of Health Fact Sheet“,
https://report.nih.gov/nihfactsheets/ViewFactSheet.aspx?csid=58. Accessed on: July 10, 2016.

[37] D. Vergyri, B. Knoth, E. Shriberg, V. Mitra, M. McLaren, L. Ferrer, P. Garcia, C. Marmar, ”Speech-based assessment of PTSD in a military population using diverse feature classes“, *Interspeech*, pp. 3729-3733, 2015.

[38] X. Zhuang, V. Rozgic, M. Crystal and B. P. Marx, ”Improving Speech-Based PTSD Detection via Multi-View Learning “, *IEEE Spoken Language Technol. Workshop*, pp. 260-265, 2014

[39] F. Liu, B. Xie, Y. Wang, W. Guo, J. P. Fouche, Z. Long, W. Wang, H. Chen, M. Li, X. Duan, J. Zhang, M. Qiu, and H. Chen, ”Characterization of Post-traumatic Stress Disorder Using Resting-State fMRI with a Multi-level Parametric Classification Approach“, *Brain Topography, Springer*, pp. 221-237, 2014.

[40] Q. Zhang, Q. Wu, H. Zu, L. He, H. Huang, J. Zhang and W. Zhang, ”Multimodal MRI-Based Classification of Trauma Survivors with and without Post-Traumatic Stress Disorder“, *Frontiers in Neuroscience*, 2016.

[41] S. Brown, A. Webb, R. S. Mangoubi, J. G. Dy, ”A Sparse Combined Regression-Classification Formulation for Learning aPhysiological Alternative to Clinical Post-Traumatic Stress Disorder Scores“, *Proc. of the Twenty-Ninth Assoc. Advancement of Artificial Intelligence Conference*, pp. 1700-1706, 2015.

[42] K. I. Karstoft, I. R. G. Levy, A. Statnikov, Z. Li, A. Y. Shalev, ”Bridging a translational gap: using machine learning to improve the prediction of PTSD“, *Biomed Central*, 2015.

[43] I. R. G. Levy, K. I. Karstoft, A. Statkinov and A. Y. Shalev, ”Quantitative Forecasting of PTSD from Early Trauma Responses: A Machine Learning Application“, *Journal Of Psychiatric Research*, pp. 68-76, 2014.

[44] M. El. Ayadi, M. S. Kamel, and F. Karray, “Survey on speech Emotion Recognition: Features, Classification Schemes And Databases,” *Pattern Recog.*, vol. 44, no. 3, pp. 572-587, 2011.

[45] S. Wu, T. H. Falk, and W. Y. Chan, “Automatic recognition of speech emotion using long-term spectro-temporal features,” *Int’l. Conf. Digital Signal Process.*, 2009, pp. 1-6.

[46] C. Busso, S. Mariooryad, A. Metallinou, S. Narayanan, “Iterative feature normalization scheme for automatic emotion detection from speech,” *IEEE Trans. Affect. Comput.*, vol. 4, no. 4, pp. 386-397, 2013.

[47] C. Busso, S. Lee, and S. Narayanan, “Analysis of emotionally salient aspects of fundamental frequency for emotion detection,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 17, no. 4, pp. 582-596, 2009.

[48] A. Coates, and A. Y. Ng, “The importance of encoding versus training with sparse coding and vector quantization,” *Proc. Int’l. Conf. Machine Learning*, 2011, pp. 921-928.

[49] K. Patel, and R.K. Prasad, “Speech recognition and verification using MFCC and VQ,” *Int’l. Journal of Emerging Science And Engineering*, vol. 1, no. 7, 2013, pp. 33-37.

[50] Z. Tychtl, and J. Psutka, “Speech production based on the mel-frequency cepstral coefficients,” *EUROSPEECH*, 1999, pp. 2335-2338.

[51] J. H. L. Hansen, W. Kim. M. Rahurkar. E. Ruzanski. and J. Myerhoff, “Robust Emotional Stressed Speech Detection Using Weighted Frequency Subbands,” *Hindawi Publishing Corporation*, vol. 2011, no. 10.1155/2011/906789.

[52] J. H. L. Hansen, C. Swail, A. South, R. K. Moore, H. Steeneken, E. J. Cupples, T. Anderson, C. R. A. Vloeberghs, I. Trancoso, and P. Verlinde, “The impact of speech under stress on military speech technology,” *NATO IST/TG-01*, 2000.

[53] E. Moore II, M. A. Clements, J.W. Peifer, and L. Weisser, “Critical analysis of the impact of glottal features in the classification of clinical depression in speech,” *IEEE Trans. Biomed. Engineering.*, vol. 55, no. 1, 2008, pp. 96-107.

[54] P. Sprechmann, and G. Sapiro, “Dictionary learning and sparse coding for unsupervised clustering,” *Int’l. Conf. Acoustics, Speech, Signal Process.*, 2010, pp. 2042-2045.

[55] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, “Discriminative learned dictionaries for local image analysis,” *IEEE Conf. Comput. Vision*, Pattern Recog., 2008, pp. 1-8.

[56] J. Mairal, M. Leordeanu, F. Bach, M. Hebert and J. Ponce, “Discriminative sparse image models for class-specific edge detection and image interpretation,” *Lec. Notes. Comput. Science*, vol. 5304, pp. 43-56, 2008.

[57] R. Raina, A. Battle, H. Lee, B. Packer, A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," *Int'l. Conf. Machine Learning*, 2007, pp. 759-766.

[58] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Machine Intell.*", IEEE Trans. Pattern Anal. Machine Intell., vol. 31, no. 2, pp. 210-227, 2009.

[59] E. Ogusulu, K. Iftekharuddin, and J. Li, "Sparse coding for hyperspectral images using random dictionary and soft thresholding," *Visual Inf. Process.*, 2012.

[60] C. Cortes, and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273-297, 1995.

[61] J. Hansen, "Getting Started With SUSAS: Speech Under Simulated And Actual Stressed Database," *Int'l Conf. Speech Comm. And Technol.* , vol. 4, 1997.

[62] S. E. Bou-Ghazale, and J. H. L. Hansen, "A comparitive study for traditional and newly proposed features for recognition of speech under stress," *IEEE Trans. Speech, Audio Process.*, vol. 8, no. 4, pp. 429-442, 2000.

[63] D.J. Folds, J.M. Gerth, W.R. Engelman, "Enhancement of Human Performance in Manual Target Acquisition and Tracking," *USAF School of Aerospace Medicine.*, Brooks AFB., TX., 1986.

[64] A. Montanari, "Linear Discriminant Analysis and Transvariation," *Journal Of Classification*, vol. 21, pp. 71-88, 2004.

[65] X. Li, J. Tao, M. T. Johnson, J. Soltis, A. Savage, K. M. Leong, and J. D. Newman, "Stress and emotion classification using jitter and shimmer features," *IEEE Int'l. Conf. Acousics, Speech And Signal Process.*, 2007, vol. 4, pp. 1081-1084.

[66] W. B. Liang and C. H. Wu, "Emotion recognition of affective speech based on multiple classifiers using acoustic-prosodic information and semantic labels," *IEEE Trans. Affect. Comput.*, vol. 2, no. 1, pp. 10-21, 2011.

[67] E. Vayrynen, J. Kortelainen, and T. Seppanen, "Classifier-based learning of nonlinear feature manifold for visualization of emotional speech prosody," *IEEE Trans. Affect. Comput.* , vol. 4, no. 1, pp. 47-56, 2013.

[68] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz and J.G. Taylor, "Emotion recognition in human computer interaction," *IEEE Signal Process. Mag.*, vol. 18, no. 1, pp. 32-80, 2001.

[69] I. Luengo, E. Navas and I. Hernaez, "Feature analysis and evaluation for automatic emotion identification in speech," *IEEE Trans. Multimedia*, vol.12, no. 6, pp. 490-501, 2010.

[70] N. A. A. B. Johari, M. Hariharan, A. Saidatul and S. Yaacob, "Multistyle Classification of Speech Under Stress using Wavelet Packet Energy and Entropy Features," in *IEEE Conf. on Sustainable Utilization and Development in Engineering and Technol.*, 2011.

[71] J. H. L. Hansen, J. F. Kaiser and G. Zhou, "Nonlinear Feature Based Classification of Speech Under Stress," in *IEEE Trans. Speech And Audio Process.*, 2001.

[72] O.W. Kwon, K. Chan, J.Hao and T.W. Lee, "Emotion Recognition by Speech Signals," in *8th European Conf. Speech Comm. and Technol.*, Geneva, Switzerland, 2003.

[73] G. P. Kafentzis, T. Yakoumaki, A. Mouchtrais and Y. Stylianou, "Analysis Of Emotional Speech Using An Adaptive Sinusoidal Model", *22nd European Signal Process. Conf.*, Lisbon, 2104.

[74] C. O. Resa, I. L. Moreno, D. Ramos and J. G. Rodriguez, "Anchor Model Fusion for Emotion Recognition in Speech", *Springer-Verlag*, 2009.

[75] B. Vlasenko, B. Schuller, A. Wendenmuth and G. Rigoll, "Frame vs Turn-Level: Emotion Recognition from Speech Considering Static and Dynamic Processing", *Springer-Verlag*, Berlin, 2007.

[76] L. Caponetti, C. A. Buscicchio and G. Castellano, "Biologically inspired emotion recognition from Speech", *EURASIP Journal on Advances in Signal Process.*, 2011.

[77] C. A. Buscicchio, P. Gorecki and L. Caponetti, "Speech Emotion Recognition Using Spiking Neural Networks", *Springer-Verlag*, 2006.

[78] J. S. Garofolo, L. F. Lamel, J. G. Fiscus, D. S. Pallett and N. L. Dahlgren, "DARPA TIMIT - Acoustic-Phonetic Continuous Speech Corpus", *NISTIR-4930*, February, 1993.

[79] G. E. and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks", *Science*, vol. 313, pp. 504-507, 2006.

[80] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Over Fitting", *Journal of Machine Learning Research*, vol 15, 2014.

[81] L. Deng, J. Li, J. Huang, K. Yao, D. Yu, F. Seide, M. L. Seltzer, G. Zweig, X. He, J. Williams, Y. Gong and A. Acero, "Recent Advances in Deep Learning for Speech Research at Microsoft", *Int'l. Conf. Acoustics Speech Signal Process.*, 2013.

[82] S. Dieleman, B. Schrauwen, "End-to-end Learning for Music Audio", *Int'l. Conf. Acoustics Speech Signal Process.*, 2014.

[83] Y. Bengio, "Learning deep architectures for AI," *Foundations and trends in Machine Learning*, vol. 2, no. 1, pp. 1-127, 2009.

[84] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, pp. 504-507, 2006.

[85] S. J. Pan and Q. Yang, "A Survey on Transfer Learning", *IEEE Trans.Knowledge and Data Engineering*, vol. 22, No. 10, 2010.

[86] X. Jiang, "Representational Transfer in Deep Belief Networks", *Springer*, pp. 338-342, 2015.

[87] R. Raina, A. Battle, H. Lee, and A. Y. Ng, “Self-Taught Learning: Transfer Learning from Unlabaled Data”, *Proc. 24th Int’l Conf. Machine Learning*, pp. 759-766, 2007.

[88] J. Baxter, “Learning Internal Representations”, *In Proc. of Eighth Annual Conference on Computational Learning Theory*, pp. 311-320, 1995.

[89] S. Gutstein, S. Fuentes, E. Freudenthal, “Knowledge Transfer in Deep Convolutional Neural Nets”, *Int’l. Journal On Artificial Intelligence Tools*, 17(03), pp. 555-567, 2008.

[90] N. Y. Li, D. Banerjee and J. Li, “A Comparitive Study Of Classification Schemes in Transfer Learning for PTSD Diagnosis”, *SpringSim*, Society for Modeling and Simulation Int’l, 2017.

[91] N. Y. Li, D. Banerjee and J. Li, “Speech Feature Investigation in Transfer Learning for Improved Post-Traumatic Stress Disorder Diagnosis”, *SpringSim*, Society for Modeling and Simulation Int’l, 2017.

# APPENDIX A

Tables 69 through 78 are listed below. These tables show the optimal parameter selection for sparse coding using trial-error method.

**Table 69:** Classification performance for variation in the number of basis functions, for the following fixed set of parameters. The basis function size is 10 and the stride step size is 5. A random number of 20,000 patches and a soft-thresholding parameter, $\alpha_{st}$ of 0.8 are used. An *average quadrant* type of pooling is used.

| **Fixed Parameter Values** | **Basis Function Size** | **No of Basis Functions** | **Training Accuracy (%)** | **Test Accuracy (%)** |
|---|---|---|---|---|
| No of patches = 20,000, $\alpha_{st} = 0.8$, pooling type is *average quadrant*, . | 10 | 1000 | 91.66 | 91.66 |
| | | 2000 | 91.66 | 91.66 |
| | | 3000 | 91.66 | 91.51 |
| | | 5000 | 91.66 | 91.51 |

**Table 70:** Classification performance for variation in the number of random patches used to create the dictionary, for the following fixed set of parameters. The basis function size is 10 and the stride step size is 5. The soft-thresholding parameter, $\alpha_{st}$ is 0.8. The number of basis functions is 3000. An *average quadrant* pooling parameter is used.

| **Fixed Parameter Values** | **Basis Function Size** | **No of Patches** | **Training Accuracy (%)** | **Test Accuracy (%)** |
|---|---|---|---|---|
| No of basis functions = 3,000, $\alpha_{st} = 0.8$, pooling type is *average quadrant.* | 10 | 10,000 | 91.66 | 91.66 |
| | | 20,000 | 91.66 | 91.66 |
| | | 30,000 | 91.66 | 91.51 |
| | | 50,000 | 91.66 | 91.51 |

**Table 71:** Classification performance for variation in stride step size for fixed basis function sizes. The number of basis functions is 3000 selected from a random number of 20,000 patches. The soft-thresholding parameter, $\alpha_{st}$ is 0.8. An *average quadrant* pooling parameter is used. The basis function sizes are varied between 20 and 45 in increments of 5.

| **Fixed Parameter Values** | **Basis Function Size** | **Stride Step Size** | **Training Accuracy (%)** | **Test Accuracy (%)** |
|---|---|---|---|---|
| No of basis functions = 3000, No of patches = 20,000, $\alpha_{st} = 0.8$, pooling type is *average quadrant.* | 20 | 5 | 91.35 | 91.67 |
| | | 10 | 89.21 | 88.89 |
| | | 15 | 91.98 | 91.67 |
| | 25 | 5 | 91.35 | 94.44 |
| | | 10 | 89.52 | 88.89 |
| | | 15 | 91.43 | 91.67 |
| | | 20 | 90.63 | 88.89 |
| | 30 | 5 | 90.16 | 91.67 |
| | | 10 | 87.70 | 88.89 |
| | | 15 | 93.10 | 91.67 |
| | | 20 | 84.84 | 83.33 |
| | | 25 | 92.78 | 91.67 |
| | 35 | 5 | 89.52 | 86.11 |
| | | 10 | 86.51 | 86.11 |
| | | 15 | 91.59 | 91.67 |
| | | 20 | 86.43 | 88.89 |
| | | 25 | 93.65 | 94.44 |
| | 40 | 5 | 88.33 | 88.89 |
| | | 10 | 88.10 | 88.89 |
| | | 15 | 91.35 | 91.67 |
| | | 20 | 90.95 | 88.89 |
| | | 25 | 91.59 | 91.67 |
| | | 30 | 94.29 | 94.44 |
| | | 35 | 91.98 | 91.67 |
| | 45 | 5 | 89.76 | 91.67 |
| | | 10 | 88.02 | 88.89 |
| | | 15 | 90.79 | 91.67 |
| | | 20 | 89.21 | 88.89 |
| | | 25 | 92.78 | 91.67 |
| | | 30 | 94.29 | 94.44 |
| | | 35 | 91.35 | 91.67 |
| | | 40 | 91.67 | 91.67 |

**Table 72:** Classification performance for variation in stride step size for fixed basis function sizes. The number of basis functions is 3000 selected from a random number of 20,000 patches. The soft-thresholding parameter, $\alpha_{st}$ is 0.8. An *average quadrant* pooling parameter is used. The segment sizes are varied between 50 and 60 in increments of 5.

| **Fixed Parameter Values** | **Basis Function Size** | **Stride Step Size** | **Training Accuracy (%)** | **Test Accuracy (%)** |
|---|---|---|---|---|
| No of basis functions = 3000, No of patches = 20,000, $\alpha_{st}$ = 0.8, pooling type is *average quadrant.* | 50 | 5 | 88.81 | 88.89 |
| | | 10 | 88.02 | 88.89 |
| | | 15 | 88.81 | 88.89 |
| | | 20 | 88.25 | 88.89 |
| | | 25 | 91.19 | 91.67 |
| | | 30 | 94.13 | 94.44 |
| | | 35 | 91.83 | 94.44 |
| | | 40 | 93.65 | 94.44 |
| | | 45 | 91.35 | 91.67 |
| | 55 | 5 | 88.41 | 88.89 |
| | | 10 | 85.71 | 86.11 |
| | | 15 | 89.29 | 88.89 |
| | | 20 | 87.70 | 88.89 |
| | | 25 | 89.92 | 88.89 |
| | | 30 | 93.65 | 94.44 |
| | | 35 | 93.41 | 94.44 |
| | | 40 | 92.30 | 94.44 |
| | | 45 | 91.03 | 91.67 |
| | | 50 | 89.68 | 88.89 |
| | 60 | 5 | 87.78 | 88.89 |
| | | 10 | 85.63 | 86.11 |
| | | 15 | 89.68 | 88.89 |
| | | 20 | 88.57 | 88.89 |
| | | 25 | 90.95 | 88.89 |
| | | 30 | 94.05 | 94.44 |
| | | 35 | 93.65 | 94.44 |
| | | 40 | 94.60 | 94.44 |
| | | 45 | 91.27 | 91.67 |
| | | 50 | 91.67 | 91.67 |
| | | 55 | 89.37 | 86.11 |

**Table 73:** Classification performance for variation in stride step size for fixed basis function sizes. The number of basis functions is 3000 selected from a random number of 20,000 patches. The soft-thresholding parameter, $\alpha_{st}$ is 0.8. An *average quadrant* pooling parameter is used. The segment sizes are varied between 65 and 70.

| **Fixed Parameter Values** | **Basis Function Size** | **Stride Step Size** | **Training Accuracy (%)** | **Test Accuracy (%)** |
|---|---|---|---|---|
| No of basis functions = 3000, No of patches = 20,000, $\alpha_{st}$ = 0.8, pooling type is *average quadrant.* | 65 | 5 | 86.27 | 88.89 |
| | | 10 | 85.56 | 86.11 |
| | | 15 | 90.40 | 88.89 |
| | | 20 | 89.21 | 88.89 |
| | | 25 | 89.92 | 88.89 |
| | | 30 | 94.13 | 94.44 |
| | | 35 | 91.51 | 91.67 |
| | | 40 | 95.00 | 94.44 |
| | | 45 | 91.35 | 91.67 |
| | | 50 | 92.06 | 91.67 |
| | | 55 | 91.59 | 91.67 |
| | | 60 | 95.48 | 97.22 |
| | 70 | 5 | 85.56 | 86.11 |
| | | 10 | 84.21 | 86.11 |
| | | 15 | 90.95 | 91.67 |
| | | 20 | 87.86 | 88.89 |
| | | 25 | 88.89 | 88.89 |
| | | 30 | 94.37 | 94.44 |
| | | 35 | 91.75 | 91.67 |
| | | 40 | 93.97 | 94.44 |
| | | 45 | 91.03 | 91.67 |
| | | 50 | 91.03 | 88.89 |
| | | 55 | 91.35 | 91.67 |
| | | 60 | 95.00 | 94.44 |
| | | 65 | 93.97 | 94.44 |

**Table 74:** Classification performance for variation in stride step size for fixed basis function sizes. The number of basis functions is 3000 selected from a random number of 20,000 patches. The soft-thresholding parameter, $\alpha_{st}$ is 0.8. An *average quadrant* pooling parameter is used. The segment sizes are varied between 75 and 80.

| **Fixed Parameter Values** | **Basis Function Size** | **Stride Step Size** | **Training Accuracy (%)** | **Test Accuracy (%)** |
|---|---|---|---|---|
| No of basis functions = 3000, No of patches = 20,000, $\alpha_{st}$ = 0.8, pooling type is *average quadrant.* | 75 | 5 | 85.63 | 86.11 |
| | | 10 | 85.24 | 86.11 |
| | | 15 | 91.03 | 91.67 |
| | | 20 | 88.02 | 88.89 |
| | | 25 | 90.56 | 91.67 |
| | | 30 | 94.13 | 94.44 |
| | | 35 | 92.54 | 91.67 |
| | | 40 | 93.65 | 94.44 |
| | | 45 | 90.08 | 91.67 |
| | | 50 | 91.67 | 88.89 |
| | | 55 | 91.51 | 91.67 |
| | | 60 | 94.37 | 94.44 |
| | | 65 | 94.37 | 94.44 |
| | | 70 | 90.24 | 88.89 |
| | 80 | 5 | 85.63 | 86.11 |
| | | 10 | 83.97 | 86.11 |
| | | 15 | 89.29 | 91.67 |
| | | 20 | 87.70 | 88.89 |
| | | 25 | 89.21 | 88.89 |
| | | 30 | 91.98 | 91.67 |
| | | 35 | 92.30 | 91.67 |
| | | 40 | 93.89 | 94.44 |
| | | 45 | 91.43 | 91.67 |
| | | 50 | 92.14 | 88.89 |
| | | 55 | 91.27 | 91.67 |
| | | 60 | 94.60 | 94.44 |
| | | 65 | 94.13 | 94.44 |
| | | 70 | 90.56 | 91.67 |
| | | 75 | 95.24 | 97.22 |

**Table 75:** Classification performance for variation in stride step size for a fixed basis function size of 100. The number of basis functions is 3000 selected from a random number of 20,000 patches. The soft-thresholding parameter, $\alpha_{st}$ is 0.8. An *average quadrant* pooling parameter is used.

| **Fixed Parameter Values** | **Basis Function Size** | **Stride Step Size** | **Training Accuracy (%)** | **Test Accuracy (%)** |
|---|---|---|---|---|
| No of basis functions = 3000, No of patches = 20,000, $\alpha_{st}$ = 0.8, pooling type is *average quadrant.* | 100 | 5 | 85.63 | 86.11 |
| | | 10 | 85.16 | 83.33 |
| | | 15 | 90.79 | 91.67 |
| | | 20 | 89.76 | 88.89 |
| | | 25 | 91.19 | 91.67 |
| | | 30 | 94.21 | 94.44 |
| | | 35 | 90.16 | 91.67 |
| | | 40 | 92.46 | 91.67 |
| | | 45 | 90.79 | 91.67 |
| | | 50 | 91.51 | 91.67 |
| | | 55 | 91.59 | 91.67 |
| | | 60 | 96.83 | 97.22 |
| | | 65 | 92.06 | 91.67 |
| | | 70 | 94.13 | 94.44 |
| | | 75 | 94.44 | 94.44 |
| | | 80 | 92.62 | 91.67 |
| | | 85 | 94.92 | 97.22 |
| | | 90 | 92.46 | 91.67 |
| | | 95 | 94.05 | 94.44 |

**Table 76:** Classification performance for variation in basis function size for fixed stride step sizes. The soft-thresholding parameter, $\alpha_{st}$ is 0.8. An *average quadrant* pooling parameter is used. The stride step sizes are varied between 5 and 10.

| Fixed Parameter Values | Stride Step Size | Basis Function Size | Training Accuracy (%) | Test Accuracy (%) |
|---|---|---|---|---|
| No of basis functions = 3000, No of patches = 20,000, $\alpha_{st}$ = 0.8, pooling type is *average quadrant.* | 5 | 10 | 91.59 | 91.67 |
| | | 15 | 90.87 | 91.67 |
| | | 20 | 91.35 | 91.67 |
| | | 25 | 91.35 | 94.44 |
| | | 30 | 90.16 | 91.67 |
| | | 35 | 89.52 | 86.11 |
| | | 40 | 88.33 | 88.89 |
| | | 45 | 89.76 | 91.67 |
| | | 50 | 88.81 | 88.89 |
| | | 55 | 88.41 | 88.89 |
| | | 60 | 87.78 | 88.89 |
| | | 65 | 86.27 | 88.89 |
| | | 70 | 85.56 | 86.11 |
| | | 75 | 85.63 | 86.11 |
| | | 80 | 85.63 | 86.11 |
| | 10 | 15 | 91.03 | 89.21 |
| | | 20 | 89.21 | 88.89 |
| | | 25 | 89.52 | 88.89 |
| | | 30 | 87.70 | 88.89 |
| | | 35 | 86.51 | 86.11 |
| | | 40 | 88.10 | 88.89 |
| | | 45 | 88.02 | 88.89 |
| | | 50 | 88.02 | 88.89 |
| | | 55 | 85.71 | 86.11 |
| | | 60 | 85.63 | 86.11 |
| | | 65 | 85.56 | 86.11 |
| | | 70 | 84.21 | 86.11 |
| | | 75 | 85.24 | 86.11 |
| | | 80 | 83.97 | 86.11 |

**Table 77:** Classification performance for variation in basis function size for fixed stride step sizes. The soft-thresholding parameter, $\alpha_{st}$ value of 0.8 is used. An *average quadrant* pooling parameter is used. The step sizes are varied between 15 and 20.

| Fixed Parameter Values | Stride Step Size | Basis Function Size | Training Accuracy (%) | Test Accuracy (%) |
|---|---|---|---|---|
| No of basis functions = 3000, No of patches = 20,000, $\alpha_{st}$ = 0.8, pooling type is *average quadrant.* | 15 | 20 | 91.98 | 91.67 |
| | | 25 | 91.43 | 91.67 |
| | | 30 | 93.10 | 91.67 |
| | | 35 | 91.59 | 91.67 |
| | | 40 | 91.35 | 91.67 |
| | | 45 | 90.79 | 91.67 |
| | | 50 | 88.81 | 88.89 |
| | | 55 | 89.29 | 88.89 |
| | | 60 | 89.68 | 88.89 |
| | | 65 | 90.40 | 88.89 |
| | | 70 | 90.95 | 91.67 |
| | | 75 | 91.03 | 91.67 |
| | | 80 | 89.29 | 91.67 |
| | 20 | 25 | 90.63 | 88.89 |
| | | 30 | 84.84 | 83.33 |
| | | 35 | 86.43 | 88.89 |
| | | 40 | 90.95 | 88.89 |
| | | 45 | 89.21 | 88.89 |
| | | 50 | 88.25 | 88.89 |
| | | 55 | 87.70 | 88.89 |
| | | 60 | 88.57 | 88.89 |
| | | 65 | 89.21 | 88.89 |
| | | 70 | 87.76 | 88.89 |
| | | 75 | 88.02 | 88.89 |
| | | 80 | 87.70 | 88.89 |

**Table 78:** Classification performance for variation in basis function size for a fixed stride step size of 25. The number of basis functions is 3000 selected from a random number of 20,000 patches. The soft-thresholding parameter, $\alpha_{st}$ is 0.8. An *average quadrant* pooling parameter is used.

| **Fixed Parameter Values** | **Stride Step Size** | **Basis Function Size** | **Training Accuracy (%)** | **Test Accuracy (%)** |
|---|---|---|---|---|
| No of basis functions = 3000, $\alpha_{st}$ = 0.8, No of patches = 20,000, pooling type is *average quadrant.* | 25 | 30 | 92.78 | 91.67 |
| | | 35 | 93.65 | 94.44 |
| | | 40 | 91.59 | 91.67 |
| | | 45 | 92.78 | 91.67 |
| | | 50 | 91.19 | 91.67 |
| | | 55 | 89.92 | 88.89 |
| | | 60 | 90.95 | 88.89 |
| | | 65 | 89.92 | 88.89 |
| | | 70 | 88.89 | 88.89 |
| | | 75 | 90.56 | 91.67 |
| | | 80 | 89.21 | 88.89 |

**Table 79:** Classification performance for variation in the soft-thresholding parameter, $\alpha_{st}$ for the following fixed set of parameters. The number of basis functions is 3000, selected from a random number of 20,000 patches. A basis function size of 10 and stride step size of 5 are used. The pooling parameter type is *average quadrant.*

| **Fixed Parameter Values** | **Basis Function Size** | **Soft-thresholding ($\alpha_{st}$)** | **Training Accuracy (%)** | **Test Accuracy (%)** |
|---|---|---|---|---|
| No of basis functions = 3000, No of patches = 20,000, pooling type is *average quadrant* | 10 | 0.0 | 91.66 | 91.58 |
| | | 0.25 | 91.66 | 91.66 |
| | | 0.4 | 91.66 | 91.66 |
| | | 0.6 | 91.66 | 91.66 |
| | | 0.8 | 91.60 | 91.59 |
| | | 1.0 | 91.66 | 91.51 |

# VITA

Debrup Banerjee
Department of Electrical and Computer Engineering
Old Dominion University
Norfolk, VA 23529

Debrup Banerjee attended Shivaji University in India for his undergraduate degree in Mechanical Engineering in 1999 securing *distinction* standing. He completed his Master's education at Hampton University, USA in Computer Science in 2004. After that he worked with a leading global organization in information technology as a software engineer in India. He started pursuing his Ph.D. Degree in Electrical and Computer Engineering at Old Dominion University in 2009. As a graduate student, he worked as a researcher at the Old Dominion University Medical Imaging Diagnosis and Analysis lab (MIDA). He secured various accolades as a teaching assistant while pursuing his graduate studies.