© 2010 Onur Pekcan

DEVELOPMENT OF MACHINE LEARNING BASED SPEAKER RECOGNITION SYSTEM

BY

ONUR PEKCAN

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2010

Urbana, Illinois

Advisor:

Professor Dan Roth

Abstract

In this thesis, we describe a biometric authentication system that is capable of recognizing its users' voice using advanced machine learning and digital signal processing tools. The proposed system can both validate a person's identity (i.e. verification) and recognize it from a larger known group of people (i.e. identification). We designed the entire speaker recognition system to be integrated into the Siebel Center's infrastructure, and named it "Biometric Authentication System for the Siebel Center (BASS)". The main idea is to extract discriminative characteristics of an individual's voiceprint, and employ them to train classifiers using binary classification. We formed the training data set by recording 11 speakers' voices in a laboratory environment. The majority of the speakers were from different nations, with different language backgrounds and therefore various accents. They were considered to be a subset of the Siebel Center community. We asked them to speak 13 words including numeric digits (0-9) and proper nouns, and used triplet combinations of these words as passwords. We chose Mel-Frequency Cepstral Coefficients to represent the voice signals for forming frame-based feature vectors. With these we trained Support Vector Machine and Artificial Neural Network classifiers using "One vs. all" strategy. We tested our recognition models with unseen voice records from different speakers and found them very successful based on different criteria such as equal error rate, precision and recall values. In the scope of this work, we also assembled the hardware through which the software, including the algorithm and developed models, could operate. The hardware consists of several parts such as an infrared sensor that is used to sense the presence of users, a PIC microcontroller to communicate with the software and an LCD screen to display the passwords, etc. Based on the decision obtained from the software, BASS is also capable of opening the office door, where it is built to function.

To the people who have chosen to fulfill their dreams...

Acknowledgments

I would like to thank my advisor Dan Roth for his continuous support, encouragement and understanding throughout my thesis. It is my great pleasure to acknowledge him as my mentor and a great advisor. I also would like to remember Dr. Sylvian Ray for supporting me when entering into this highly enriching Master's program of Computer Science.

Developing an entire authentication system was a challenging task and an interdisciplinary work by its nature. I learned about so many things, the existence of which I was not even aware before I started studying this subject. During this adventure, I received a great amount of help from many people, which actually formed the basis of my work. It is a pleasure to thank those who have made this thesis possible: Mark Bronsberg for his help in Electrical and Computer Engineering senior design laboratory, Celal Ziftçi for teaching me how to write computer codes proficiently, Batu Sat for sharing his experience and resources on digital signal processing, Barış Aktemur, Nejan Huvaj Sarıhan, Lale Özkahya, Çiğdem Şengül, Nazlı İkizler, Serdar Yüksel, Gabriel Garcia, and Charles Thompson for their participation in the laboratory experiments. I also would like to thank Ming-Wei Chang, Emre Akbaş and Selen Pehlivan for their useful discussions about my work and İsmail Çağrı Özcan, Nur Pehlivanoğlu and Thyago Sellmann Pinto Cesar Duque for their close friendship. Finally, I would like to show my gratitude to hard working staff of the Siebel Center for Computer Science building and the University of Illinois for generously providing me their resources.

I have to confess that I could not have succeeded in this work without the support of my wife during my entire Master's education. Last but not the least, I owe my deepest gratitude to my mother.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ADC	Analog to Digital Converter
ANN	Artificial Neural Network
ASR	Automatic Speaker Recognition
BASS	Biometric Authentication System Software
DFT	Discrete Fourier Transform
EER	Equal Error Rate
FAR	False Acceptance Rate
FFT	Fast Fourier Transform
FRR	False Rejection Rate
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
IDFT	Inverse Discrete Fourier Transform
IR Sensor	Infrared Sensor
MFCC	Mel-Frequency Cepstral Coefficients
MLP	Multi-Layer Perceptron
NTN	Neural Tree Network
QP	Quadratic Programming
PIC	Peripheral Interface Controller
RS-232	Recommended Standard 232
SVM	Support Vector Machine
VAD	Voice Activity Detection
VOIP	Voice over Internet Protocol

Chapter 1

Introduction

Biometrics refers to technologies that measure, analyze and validate human body characteristics, such as fingerprints, eye retina, iris and facial patterns for authentication purposes. In addition to these characteristics, the way a person speaks is unique since each person has a different vocal tract. In this work, we mainly focus on distinguishing one person from many others, by properly expressing the acoustic features of their voice and using machine learning methods.

1.1 Overview and Problem Statement

The faculty and staff in the Siebel Center for Computer Science building use Single Validation Entry System, which is nothing but the authentication of an identity card (called I-card) through a scanning device, to enter their offices. Since I-card carries private information about the personnel, it can also be used for other purposes such as borrowing a book from the library. Although this system, at a first glance, seems to be highly secure, it requires the personnel to carry an electronically valid card at all times. In addition, there occasionally are many scratches on both sides of the card after it is heavily used, which may prevent not only entering the office spaces but also using other facilities that require authentication of I-cards. The best solution to this problem seems to be replacing the card with a new one, which is costly. As a final point, the currently used system is also very expensive to build.

In this work, we propose a voice-based authentication system that will fundamentally change the existing system. It is mainly planned to serve for biometric verification to form a practical, secure and robust entry system for the offices in the Siebel Center. This system also eliminates the problems mentioned above. We call the proposed system Biometric Authentication System for the Siebel Center (BASS). BASS is developed using speaker recognition approach. It both

validates a speaker's identity (i.e., verification) and recognizes it from a larger known group of people (i.e., identification) using the combination of words (called "passwords"). BASS can achieve these by searching the voice pattern of a speaker in the database previously formed by collecting the voiceprints of all the authenticated users. The biggest advantage is that, since the biometric data is unique to a person, it is difficult to fake the system. In addition, the personnel do not need to carry anything to authenticate them to enter the offices. It is also capable of opening the office door when the person is verified. Finally, the system is intended to operate very fast and require very little work by the user.

1.2 Motivation

The main objective of this thesis is to build a robust biometric authentication system, which can be achieved at both software and hardware levels. First, the hardware is built to provide flexibility to the public so that it requires almost no effort to open the door when a person is authenticated. The hardware should include five parts: (i) an infrared sensor that can initiate the recognition process when the speaker appears in front of it, (ii) a microphone to record the user's voice, (iii) a screen where predefined passwords are shown, (iv) a computer where the software operates, and (v) an electronic door security system to open the door. Finally, the system needs to be easily integrated into the existing security infrastructure of the Siebel Center.

The second part of this thesis involves the development of speaker recognition software that operates using the built hardware. It needs to perform both speaker verification and identification effectively. For this purpose, speaker models should be developed using machine learning tools, which are also required to be validated with a comprehensive database to check for robustness and reliability. After the successful development of each component, the hardware and software parts are combined together to form a fully functional authentication system.

1.3 Research Methodology

The implementation of machine learning algorithms and the development of corresponding software is the main part of this work, although the building of hardware setup also plays an important role.

The final product is originally intended to be used for Professor Dan Roth's office, who would like to have an intelligent door that is opened only when he is in front of it. As a result, the model study is first performed to validate his identity. For this purpose, speech files from many speakers are collected in the laboratory, which is thought to be representing his office environment. Speakers are chosen from the Siebel Center staff and many other volunteers. During the experiments, they are asked to speak different words including numeric digits and proper nouns. The voice recordings are done carefully to form a database of individuals, in which both Dan Roth's and impostors' voice models are intended to be stored. Digital signal processing is performed to extract the embedded acoustic features from each speaker's voice. For each word, frame-based feature vectors are then formed to be fed into SVMs and ANNs as inputs to perform binary classification. The developed classifiers are finally tested using unseen voiceprints chosen from the database.

The identification of each speaker recorded in the database is the next step in this thesis. This is carried out by developing several models for each speaker in the database. For each word, a new voice record is tested using the developed models to make a decision about the identity of the person by calculating the matching scores of every single model. The one producing the highest score is then selected.

1.4 Thesis Organization

In Chapter 2, we present the overview of past studies on speaker recognition and give a brief explanation of pattern recognition techniques used in this field. We then explain the details of two classification methods, SVMs and ANNs, which have been widely used to build robust speaker recognition systems. In Chapter 3, the details of the hardware setup are presented. Then the architecture of software is given, explaining the details of processes required for speaker recognition. In Chapter 4, we describe the experiments performed in the Siebel Center. We provide the performances of authentication models developed using SVMs and ANNs for both speaker verification and identification purposes. Discussion of the results is also provided at the end of Chapter 4. Finally, we conclude by providing the summary of our work, conclusions and future research concepts in Chapter 5.

Chapter 2

Literature Survey

In this chapter, we first talk about the evaluation of biometric measures and why we chose voice-authentication for the users of BASS. We then briefly present evolution of speech processing studies, which form the basis of automatic speaker recognition (ASR) systems. We also provide a review of the studies performed in the area of ASR with an emphasis on pattern recognition techniques. Next, we explain the theory of Support Vector Machines (SVMs) and Artificial Neural Networks (ANNs) in detail, which we use as primary tools to develop speaker recognition models. Finally, we provide the applications of these techniques in the field of speaker recognition.

2.1 Voice as a Biometric Tool

The vulnerability of a physical or a behavioral trait to be used in a biometric application can be determined using various criteria. Table 2.1 provides the comparison of biometrics using 7 criteria that are explained below [57]. (In addition to the research articles, one of the best places to find the latest progress in biometrics and its recently discovered measures are the online resources [1, 2, 3]).

1. *Universality:* how commonly a biometric is found in each individual.
2. *Uniqueness:* how well the biometric separates one individual from another.
3. *Permanence:* how well a biometric resists in time.
4. *Performance:* how accurate, fast and robust a biometric is.
5. *Collectability:* how easy it is to acquire a biometric for measurement.
6. *Acceptability:* how much it penetrates into daily life.
7. *Circumvention:* how difficult it is to fake the authentication system.

It can easily be seen that voiceprint has a high degree of acceptability, although it has a lower degree of uniqueness and performance compared to other biometrics. In addition, voice-based recognition systems have been studied exhaustively in the last four decades and the outcomes of these works have already been used successfully in various commercial [5, 7] and open source products [4, 6]. A further use may be to increase the security of existing biometric systems through hybridization, which is generally called multi-biometrics [87]. Therefore, for all practical purposes, voice is still one of the most practical biometric tools.

Table 2.1: Comparison of biometrics using various criteria [57]: the performance of each biometric is categorized as either low (L), medium (M), or high (H).

Biometrics	**Universality**	**Uniqueness**	**Permanence**	**Collectability**	**Performance**	**Acceptability**	**Circumvention**
Face	H	L	M	H	L	H	L
Fingerprint	M	H	H	M	H	M	H
Hand Geometry	M	M	M	H	M	M	M
Keystroke Dynamics	L	L	L	M	L	M	M
Hand Vein	M	M	M	M	M	M	H
Iris	H	H	H	M	H	L	H
Retinal Scan	H	H	M	L	H	L	H
Signature	L	L	L	H	L	H	L
Voiceprint	M	L	L	M	L	H	L
Facial Thermogram	H	H	L	H	M	H	H
DNA	H	H	H	L	H	L	L

2.2 Automatic Speech Recognition

The understanding of spoken language by machines is an extensive research field. It requires the contribution from many different scientific fields such as Artificial Intelligence, Computational Lin-

guistics, etc. The key purposes of automatic speech recognition are: (i) to build a system that is capable of converting acoustic signal to a string of words (so-called transcription) and (ii) to understand the meaning contextually rather than just the words, which is one of the major topics of Natural Language Processing.

The earliest work reported in automatic speech recognition field was encountered in the 1920s. The first system that could understand speech was, interestingly, a toy named "Radio Rex". It could move via a spring that released whenever the word "Rex" was pronounced; it used the acoustic energy of the first formant of the vowel [35]. Although it was very primitive, Radio Rex functioned successfully since it moved whenever it was called. More qualified speech recognition systems started to appear at the end of the 1940s. Bell Labs designed a system that can recognize 10 digits from a single speaker [36]. In this work, 97-99% accuracy was obtained by storing unique patterns for each of the words corresponding to the first two vowel formants of the digits. Another milestone work was the speech recognizer capable of recognizing four vowels and nine consonants based on a similar pattern recognition principle [39, 46]. The importance of this work is that it marked the first use of phoneme transition probabilities to constrain a recognizer.

Starting from the 1960s there have been numerous significant developments in the field of speech recognition systems. Novel feature extraction algorithms, the concept of cepstral processing and warping of signals, were introduced at that time. In the 1970s, Hidden Markov Models (HMMs) started to be used and penetrated slowly into technologies to be used for automatic speech recognition systems. The use of ANNs also became widespread in the 1980s because of their desired properties. In the last two decades, SVMs, and Gaussian Mixture Models (GMMs) have dominated the field in addition to HMMs. In spite of the successful implementation of several techniques, the problem of speech transcription and understanding of context has been solved only to a limited extent. Since the focus of this work is the recognition of speakers rather than the speech, we will not go into further details of the development in speech recognition area. A comprehensive historical overview of developments and the major steps taken in this area can be found in [59].

2.3 Automatic Speaker Recognition (ASR)

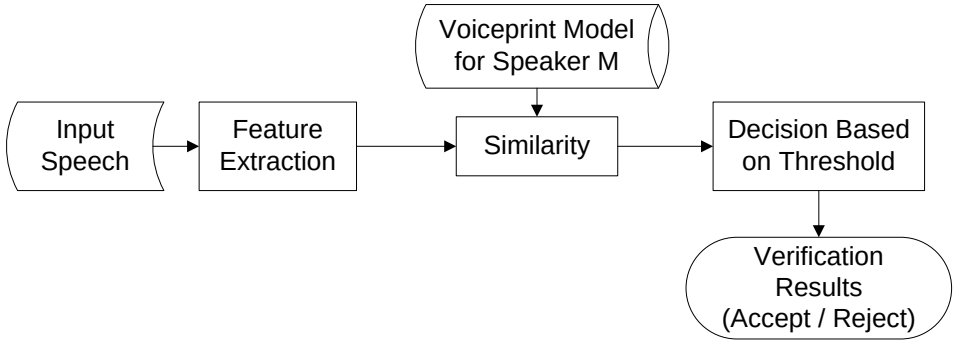
Another field of speech processing and the main subject of this thesis is speaker recognition. It can be defined as the process of automatically recognizing a person on the basis of information captured by interpreting speech signals. ASR can be defined more precisely as the use of a machine to recognize a person from a spoken phrase [25]. There are two main paradigms used in speaker recognition (Figure 2.1):

1. *Speaker verification:* A given speaker is verified who s/he claims to be. A typical verification system asks the user who claims to be the speaker (so-called client) to provide an identification. It then verifies the user by comparing codebook of given the speech utterance with that given by user. If it matches the set threshold then the user is accepted as client otherwise the user is labeled as impostor and rejected (Figure 2.1(a)).

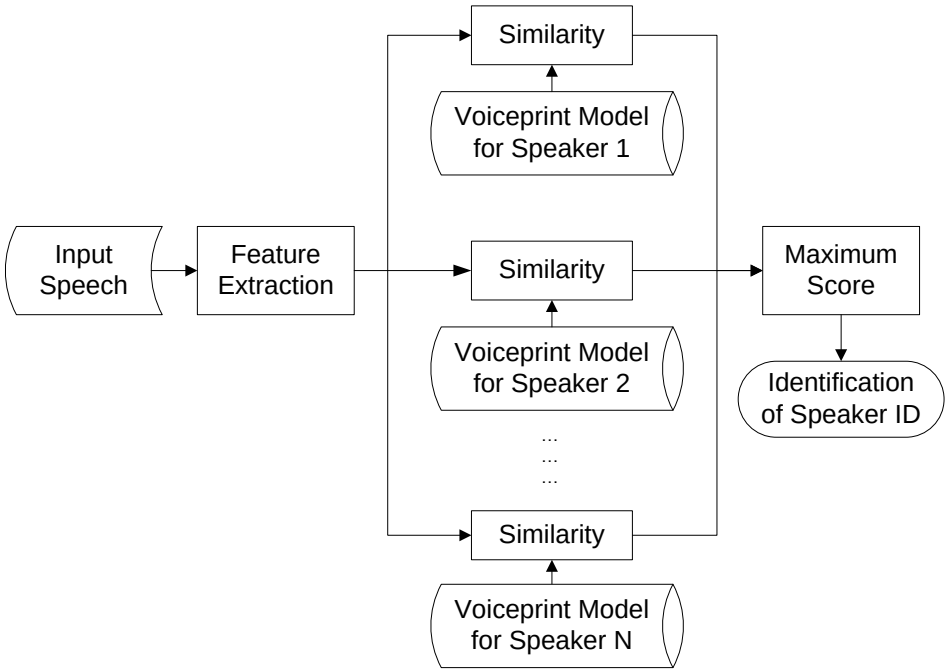
2. *Speaker identification:* A particular speaker is detected from a known population. An identifier system prompts the user to provide speech utterance. It then identifies the user by comparing the codebook of the speech utterance with those stored in the database and outputs the most likely speaker who could have given that speech utterance (Figure 2.1(b)).

One more way of classifying speaker recognition systems is based on text and shown in (Figure 2.2). The systems can be either *text dependent*, where speakers' speech corresponds to a previously defined text (so-called password) and the user is cooperative (i.e., s/he volunteers to be recognized), or *text independent* where there is no constraints on what the speakers speak and the user is potentially uncooperative [25]. Finally, further classification can be made for identification systems based on the characteristics of the group of people to be recognized. If the individual to be recognized is previously known to be in the database of biometric characteristics, the identification is called closed-set, otherwise it is called open-set identification.

In general, there are two important problems in speaker recognition: using unique feature sets and finding a technique to distinguish these for easily recognizing a user. Since this thesis uses SVMs and ANNs to find the best possible solution to our classification problem, we will discuss the theory of these in the next section. We leave the discussion of details for finding a good feature set to the next chapter.



(a) Speaker verification



(b) Speaker identification

Figure 2.1: Illustration of speaker recognition categories

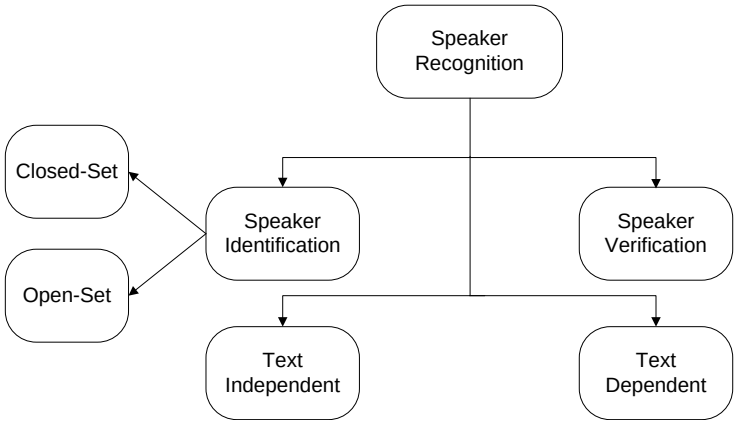


Figure 2.2: Types of speaker recognition

2.4 Machine Learning Algorithms

The area of machine learning provides a variety of tools for recognition of speakers, which generally use the information extracted from speech records using digital signal processing. An overview of these tools for speaker recognition is given in [95]. Among these, SVMs and ANNs play an important role due to several of their desired properties.

2.4.1 Support Vector Machines (SVMs)

SVMs are powerful classifiers that have gained much attention in the last decade. Applications of SVMs for pattern recognition problems can be found in [23]. Their mathematical theory and other margin based methods are thoroughly explained in [21, 97]. SVMs make decisions based on constructing a linear decision boundary (so-called hyperplane) that generally separates two classes (in some cases, there are more than two classes).

When two classes are linearly separable, the decision boundary is defined by:

$$\mathbf{w} \cdot \mathbf{x} + b = 0, \tag{2.1}$$

where $\mathbf{w}$ is the normal to the hyperplane (Figure 2.3). For linearly separable data that are labeled $\{\mathbf{x_i}, y_i\} \in R^d$ and $y_i \in \{-1, 1\}, i = 1..N$, the optimal hyperplane is chosen according to the maximum margin criterion (N is the number of data and d is dimension of the problem), which separates the

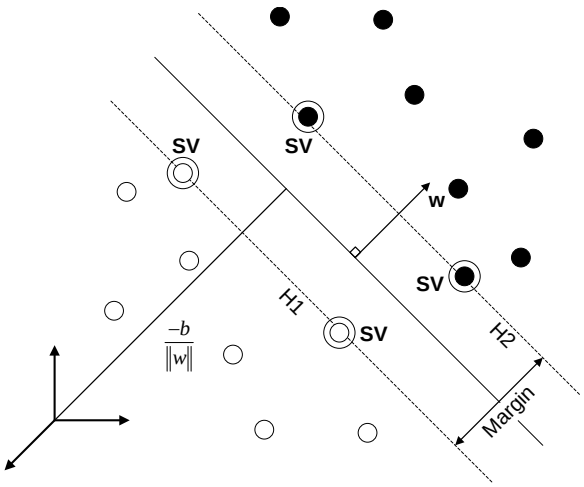


Figure 2.3: Linear hyperplanes (H1,H2) and support vectors (SVs) for separable data [21]

points by maximizing the perpendicular distance to the plane. The solid line in Figure 2.3 is the optimal hyperplane and the closest data points are called Support Vector (SVs).

The hyperplane can be found by minimizing the objective function given in Equation 2.2, i.e., the square of L2 norm of **w**, which is subject to constraints in Equation 2.3 to separate the data correctly. This implies $(\mathbf{w}\cdot\mathbf{x}+b)y_i \geq 1$ for the separable examples. As a result, SVs satisfy the equalities given in Equation 2.4.

$$\phi(\mathbf{w}) = \frac{\| w \|^2}{2} \tag{2.2}$$

$$y_i(\mathbf{w}\cdot\mathbf{x_i}+b)-1 \geq 0, \forall_i \tag{2.3}$$

$$y_s(\mathbf{w}\cdot\mathbf{x_s}+b) = 1 \tag{2.4}$$

This is a Quadratic Programming (QP) optimization problem the solution of which will be described after we extend the discussion to the case of non-separable data. In many cases, the data is not linearly separable, that is, there exists no hyperplane that satisfies the inequality given in Equation 2.3. This can be solved by introducing slack variables ξ_i into Equation 2.3 to relax the constraints such that some points are allowed to lie within the margin or they can even be misclassified. Equation 2.3 then becomes:

$$y_i(\mathbf{w}\cdot\mathbf{x_i}+b) \geq 1-\xi_i, \forall_i. \tag{2.5}$$

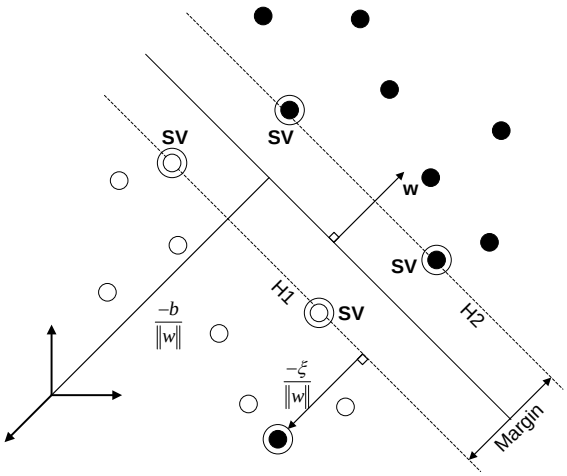


Figure 2.4: Linear hyperplanes (H1,H2) and support Vectors (SVs) for non-separable data [21]

The resulting objective function to minimize then becomes:

$$\phi(\mathbf{w}) = \frac{\| w \|^2}{2} + C\sum_{i=1}^{N} L(\xi_i). \tag{2.6}$$

where the term on the right hand side is the empirical risk associated with the marginal or misclassified points, L is the loss function and C is a parameter to specify the balance between the effects of minimizing the empirical risk and maximizing the margin [101]. The most commonly used loss function is the linear error-cost function ($L(\xi_i) = \xi_i$), since it is robust to outliers. A larger C means a higher penalty to misclassification errors. Finally, minimizing Equation 2.6 with constraints in Equation 2.5 gives a "Generalized Separating Hyperplane", which is still a QP problem. The case of non-separable data is shown in Figure 2.4.

The dual formulation of Equation 2.6 with ($L(\xi_i) = \xi_i$) is given by:

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x_i x_j} \tag{2.7}$$

which is subject to constraints:

$$0 \leq \alpha_i \leq C, \tag{2.8}$$

$$\sum_{i=1}^{N} \alpha_i y_i, \tag{2.9}$$

where α_i is the Lagrange multiplier of i^{th} constraint in the primal optimization problem.

Finally, the optimal plane $\mathbf{w}_0$ is given by Equation 2.10, where N_s is the number of SVs. It is a linear combination of all points in the feature space that have $\xi_i > 0$ as well as those that lie on the margin (i.e.,$\alpha_i \neq 0$).

$$\mathbf{w}_0 = \sum_{i=1}^{N_s} \alpha_i y_i \mathbf{x_i} \tag{2.10}$$

SVMs can be extended to nonlinear classification problems by the so-called "Kernel Trick". Instead of applying the linear methods directly to the input space R^d, they are applied to a higher dimensional feature space F, which is nonlinearly related to the input space via the mapping $\Phi : R^d \rightarrow F$. In other words, each data point is mapped onto a manifold embedded into some feature space, defined implicitly by the kernel, which can be of significantly higher dimension than the input space. The training algorithm then only depends on the data through the dot product in F of the form: $\phi(\mathbf{x_i}) \cdot \phi(\mathbf{x_j})$. The computation of the dot product is prohibitive if the number of training vectors $\phi(\mathbf{x_i})$ is large, and since ϕ is not known a priori, the Mercer's theorem [34] for positive definite functions allows to replace $\phi(\mathbf{x_i}) \cdot \phi(\mathbf{x_j})$ by a positive definite symmetric kernel function $K(\mathbf{x_i}, \mathbf{x_j})$ such that $K(\mathbf{x_i}, \mathbf{x_j}) = \phi(\mathbf{x_i}) \cdot \phi(\mathbf{x_j})$. The hyperplane then is constructed in the feature space and intersects with the manifold creating a nonlinear boundary in the input space [101]. This way, the data can be made linearly separable in the feature space, although the original input is not linearly separable. Various kernel functions are used in the literature; some common examples are provided in Equations 2.11 - 2.14:

$$Linear : K(\mathbf{x_i}, \mathbf{x_j}) = \mathbf{x_i} \cdot \mathbf{x_j}, \tag{2.11}$$

$$Polynomial : K(\mathbf{x_i}, \mathbf{x_j}) = (\gamma \mathbf{x_i} \cdot \mathbf{x_j} + r)^n, \gamma > 0, \tag{2.12}$$

$$RadialBasisFunction : K(\mathbf{x_i}, \mathbf{x_j}) = \exp\left[-\gamma\left(\left\|\mathbf{x_i} - \mathbf{x_j}\right\|\right)^2\right], \gamma > 0, \tag{2.13}$$

$$Sigmoid : K(\mathbf{x_i}, \mathbf{x_j}) = \tanh(\gamma \mathbf{x_i} \cdot \mathbf{x_j} + r), \tag{2.14}$$

where n is the order of polynomial, γ and r are kernel parameters.

For the solution of QP problems described above, well known techniques such as Lagrange Multipliers and Wolfe Dual can be efficiently used [90, 91]. Although, these techniques can be

applied for small to medium-scaled problems, they are not suitable for large size problems because of the following reasons [63]:

- Kernel matrix should be computed first, which requires large memory to store
- The matrix operations such as Cholesky decomposition to solve kernel matrix is computationally expensive.

Various algorithms were proposed to tackle the above problems [51, 79, 82, 63].

2.4.2 Artificial Neural Networks (ANNs)

ANNs, or Neural Networks, are a biologically inspired class of machine learning methods, which have a variety of applications in many scientific fields. There are a number of ANNs that have been developed to solve classification and regression problems [15, 10]. Among these, Multi-Layer Perceptrons (MLPs) with trained backpropagation learning algorithm are the ones generally used for solving classification problems.

2.4.2.1 Multi-Layer Perceptrons (MLPs)

MLPs, also known as Feedforward Neural Networks, consist of artificial neurons that are interconnected to each other to mimic the behavior of biological neurons. A typical MLP structure is shown in Figure 2.5. Input and output layers represent the features that are inputs and outputs of the problem, respectively. The hidden layer connects these two layers and ensures information flow. The design of hidden layer is performed by changing the number of layers and their neurons, depending on the problem complexity.

In MLPs, artificial neurons communicate through signals that are sent through weighted connections resulting in high degree of interconnections. The connections between the neurons are generally defined using weights W_{ij}, which determine the effect a signal of neuron i (in the input layer) has on neuron j (in the hidden layer). Each neuron has a threshold above which it is activated (or fired). The activation is updated based on the input coming from the other neurons of different layers and a bias term that represents an external resource to the system. The activation is calculated

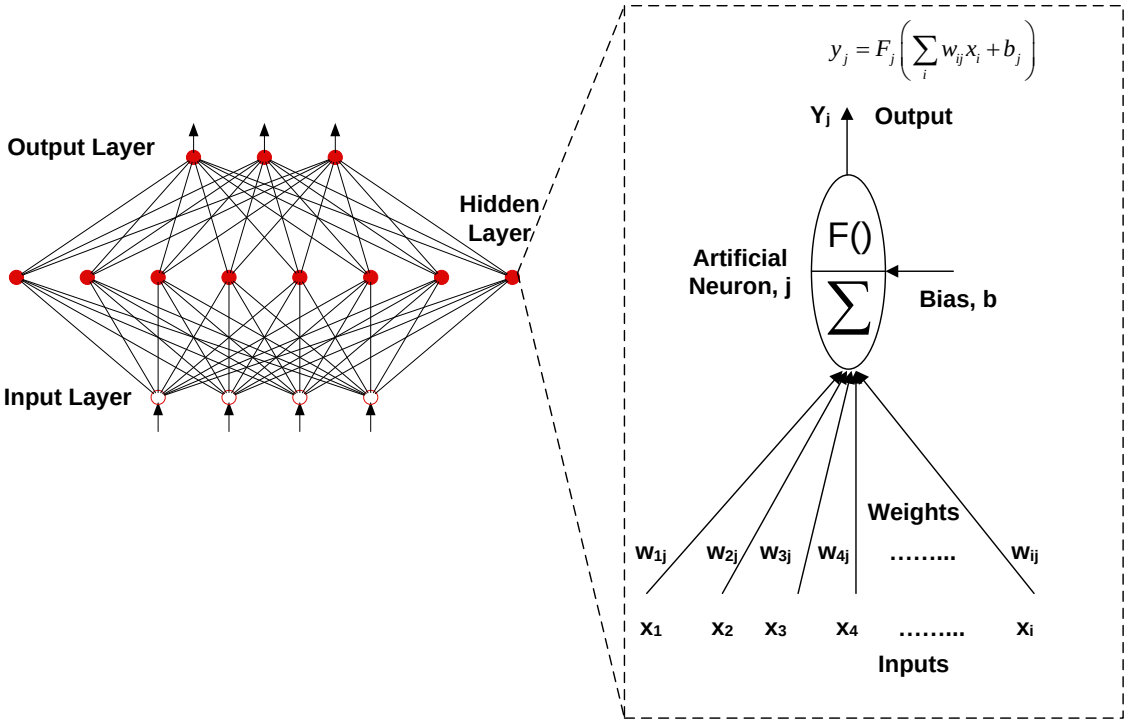


Figure 2.5: Structure of multi-layer perceptrons

using Equation 2.19, where the term $\sum_i w_{ij}x_i + b_j$ is usually called net signal for neuron j.

$$y_j = F_j\underbrace{\left(\sum_i w_{ij}x_i + b_j\right)}_{net_j} \tag{2.15}$$

There are three kinds of activation functions that can be used to operate on the net signal coming to a neuron j: linear, sigmoid and hyperbolic tangent sigmoid, the definitions of which are given in Equations 2.16, 2.17 and 2.18, respectively.

$$F_j(x) = x \tag{2.16}$$

$$F_j(x) = \frac{2}{(1+e^{-x})} \tag{2.17}$$

$$F_j(x) = \frac{2}{(1+e^{-2x})} - 1 \tag{2.18}$$

Similarly, each output unit computes its net activation based on the hidden unit signals as:

$$net_k = \sum_{j=1}^{n_H} y_j w_{kj} + b_k. \tag{2.19}$$

2.4.2.2 Backpropagation Learning Algorithm

The MLP operates using in two phases: (i) Feedforward Phase and (ii) Learning Phase. The feedforward processes consist of presenting a pattern to the input units and passing (or feeding) the signals through the network in order to get output units, while learning is a supervised one that consists of presenting an input pattern and modifying the network parameters (weights) to reduce distances between the computed output and the desired output [15].

Let t_k be the k^{th} target (or desired) output and z_k be the k^{th} computed output with $k = 1, ..., c$ (c being the number of outputs) and w represents all the weights of the network, and then training error becomes [41]:

$$J(w) = \frac{1}{2} \sum_{k=1}^{c} (t_k - z_k)^2 = \frac{1}{2} \|\mathbf{t} - \mathbf{z}\|^2. \tag{2.20}$$

The backpropagation learning rule is based on gradient descent, i.e., the weights are initialized with random values and they are changed in a direction that will reduce the error. The direction is based on gradient as:

$$\Delta \mathbf{w} = -\eta \frac{\partial J}{\partial \mathbf{w}}, \tag{2.21}$$

where η is the *learning rate* which indicates the relative size of the change in weights. The update of weights is given by:

$$\mathbf{w}(m+1) = \mathbf{w}(m) + \Delta \mathbf{w}(m), \tag{2.22}$$

where m is the m^{th} pattern presented to the network, and $\mathbf{w}(m)$ and $\mathbf{w}(m+1)$ are the weights in the current and updated states, respectively. In component form, the change in weights can be rewritten as:

$$\Delta w_{mn} = -\eta \frac{\partial J}{\partial w_{mn}}. \tag{2.23}$$

Each of the c output units operates in the same manner as the hidden units do, computing net_k as the inner product of the hidden unit signals and weights at the output unit:

$$net_k = \sum_{j=1}^{n_H} y_j w_{kj} + b_k, \tag{2.24}$$

where the subscript k indexes units in the output layer and n_H denotes the number of hidden units in the hidden layer. Then using the chain rule for differentiation, the change in the error with respect to the weights can be re-written as:

$$\frac{\partial J}{\partial w_{kj}} = \frac{\partial J}{\partial net_k} . \frac{\partial net_k}{\partial w_{kj}} = -\delta_k \frac{\partial net_k}{\partial w_{kj}}, \tag{2.25}$$

where the sensitivity, δ_k, of unit k is defined as:

$$\delta_k = -\frac{\partial J}{\partial net_k}. \tag{2.26}$$

δ_k describes how the overall error changes with the activation of the unit's net:

$$\delta_k = -\frac{\partial J}{\partial net_k} = -\frac{\partial J}{\partial z_k} . \frac{\partial z_k}{\partial net_k} = (t_k - z_k) f'(net_k). \tag{2.27}$$

Since $net_k = w_k^T \cdot y$,

$$\frac{\partial net_k}{\partial w_{kj}} = y_j, \tag{2.28}$$

and hence the weight update (or learning rule) for the hidden-to-output weights is:

$$\delta w_{kj} = \eta \delta_k y_j = \eta (t_k - z_k) f'(net_k) y_j. \tag{2.29}$$

Then, the error on the input-to-hidden units:

$$\frac{\partial J}{\partial w_{ji}} = \frac{\partial J}{\partial y_j} . \frac{\partial y_j}{\partial net_j} . \frac{\partial net_j}{\partial w_{ji}}, \tag{2.30}$$

$$\begin{aligned}
\frac{\partial J}{\partial y_j} &= \frac{\partial}{\partial y_j}\left[\frac{1}{2}\sum_{k=1}^{c}(t_k - z_k)^2\right] \\
&= -\sum_{k=1}^{c}(t_k - z_k)\frac{\partial z_k}{\partial y_j} \\
&= -\sum_{k=1}^{c}(t_k - z_k)\frac{\partial z_k}{\partial net_k}.\frac{\partial net_k}{\partial y_j} \\
&= -\sum_{k=1}^{c}(t_k - z_k)f'(net_k)w_{kj}, \qquad (2.31)
\end{aligned}$$

where the sensitivity for a hidden unit is defined as simply the sum of the individual sensitivities at the output units weighted by the hidden-to-output weights w_{kj}, multiplied by $f(net_j)$:

$$\delta_j \equiv f'(net_j)\sum_{k=1}^{c} w_{kj}\delta_k. \qquad (2.32)$$

Thus the learning rule for the input-to-hidden weights is:

$$\Delta w_{ji} = \eta x_i \delta_j = \eta \underbrace{\left[\Sigma w_{kj}\delta_k\right] \; f'(net_j)}_{\delta_j} x_i. \qquad (2.33)$$

2.4.3 SVMs and ANNs in Speaker Recognition

SVMs and ANNs have been successfully used for speaker recognition problems thus far. In fact, every ANN has an equivalent SVM formulation as pattern classifier [11]. There are many research studies available in the literature showing that the two top-level neural network approximation frameworks are the applications of support vector machine theory and regularization theory to neural networks [21, 48, 71].

One of the earliest works performed using SVMs was published in 1996 [89]. In this work, two significant advantages of support vector classifiers were discussed. First, it was stated that computing polynomial classifiers from thousands of points was computationally doable. Second, by minimizing the support vector criterion function, the capacity of the classifier was reduced, resulting in fewer test errors. They used Switchboard corpus, which has various problems such as

excessive noise. They present various results with different training techniques such as one vs. all and pairwise training. They used a polynomial classifier with degree up to 5.

Another work with SVMs was published in 2000 [101]. The YOHO database [24] was used to assess the performance of SVMs for text independent speaker verification task. They utilized regular polynomial and RBF kernels and also developed normalized polynomial kernels. The minimum equal error rate (EER) they reached was 0.34% using normalized polynomial kernel with a degree of 10. The performance of RBF kernel was reported to be EER of 1.47%. Finally, they also presented speaker identification performances of the proposed system.

Another typical approach to integrate SVMs into speaker recognition task is to discriminate between entire utterances rather than frames. The utterances naturally have different lengths and, therefore, require a mapping from a variable length pattern to a fixed size vector is needed, which is generally a challenging problem [62]. Several methods have been successfully used in speaker recognition problem by solving the above problem, which include generalized linear discriminant sequence kernel [26, 99, 100, 60], Fisher kernel methods [45, 102], n-gram kernels [27, 52], Maximum Likelihood Linear Regression transform kernels [94, 43, 61, 44], and GMM supervector kernels [30].

An ANN with a non-linear transfer function and sufficiently large number of nodes in the hidden layer may approximate any functional mapping from input to output [68, 72]. This is why ANNs are considered as powerful tools. They have been applied successfully in the field of speech processing [73] and speaker recognition [77]; generally, MLPs with backpropagation algorithm (as we described above) were used [13, 47]. In fact, we used an approach similar to one described in these studies. Another successful implementation of ANNs occurred with the use of neural tree network (NTNs), which is basically a separate neural network in each node of a tree [42]. It is a combination of tree methods together with ANNs. The nodes are used to determine which branch of the tree is preferred. NTNs were used in commercial applications using text dependent speaker recognition [95]. In addition to these, time delay neural networks [12], radial basis function networks [78] and binary-pair neural networks [88] can also be counted as successful applications of ANNs using various databases. Some of these techniques were replaced with the use of GMM approach [95].

Many of the above studies using SVMs and ANNs reached less than 10.0% EER values based on the public databases. These include NIST 2002 cellular speaker recognition evaluation (Switchboard, cellular), NIST 2003 extended data SRE (Switchboard-II, landline), FISHER Corpus [33], NIST 2004 SRE (Mixer), and NIST 2005 SRE (Mixer) [8]. All of these databases use conversational speeches recorded over landline or cellular networks.

Chapter 3

Components of BASS

In this chapter, we present individual components of BASS and explain how they operate. Development of BASS contains two major tasks: (i) Assembly of hardware components, and (ii) Design of speaker recognition software. First, we explain the technology of the hardware, i.e., individual components, their communication with the software, and details of connections, etc. After that, we provide the software architecture and discuss in detail how the speech processing component operates. Finally, we talk about the performance of our hardware and software assembly, since successful combination of these parts is crucial to have a robust working system.

The flow of how the entire system works and the main functions of its individual components are illustrated in Figure 3.1. The operation of BASS can be summarized as follows:

1. An infrared sensor senses the presence of an individual trying to access the system.
2. It sends a signal to a computer that stores speaker dependent models and passwords, to start the authentication system.
3. The software sends randomly chosen passwords to an LCD screen and asks the speaker to repeat them in 2 seconds.
4. It records the voiceprint, validates the identity against a previously chosen speaker and identifies it among the others.
5. The software sends the final decision of the system to an LCD screen as a text and to the door receiver as a binary signal to operate the door.
6. BASS repeats the above procedure three times starting from step 2 in the case of failed authentications. If none of these attempts produce successful results, the user is directed to the security personnel of the building.

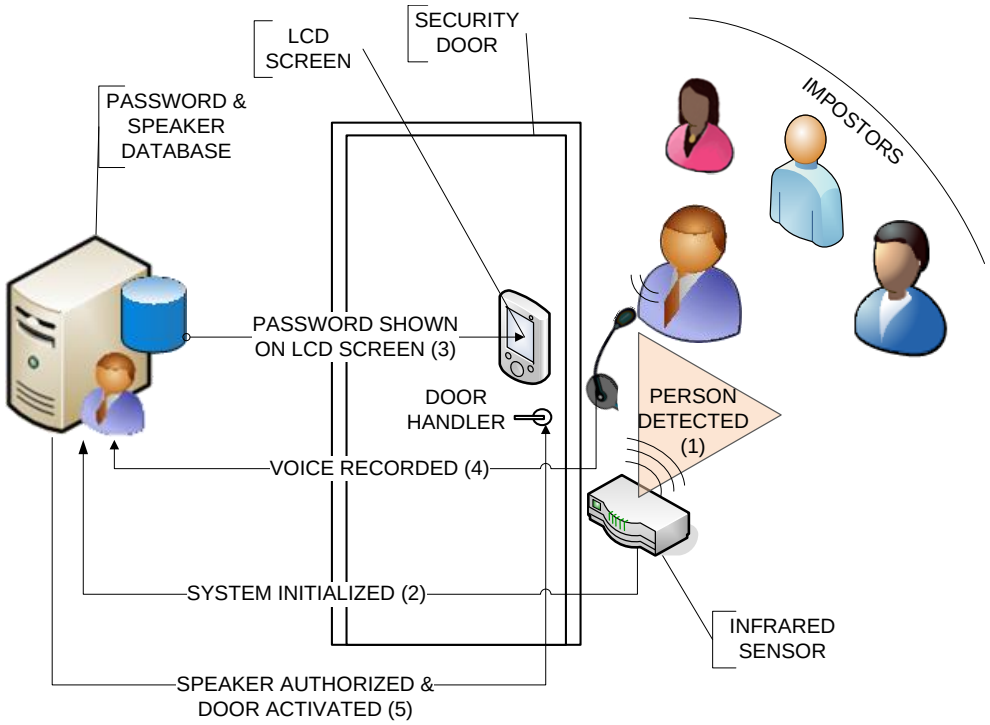


Figure 3.1: Operation steps of biometric authentication system

3.1 Hardware Components

BASS hardware mainly consists of five parts:

1. Infrared (IR) Sensing System
2. Microphone
3. LCD screen
4. Computer
5. Door security system

These components are designed with minimal interdependencies: in the case of malfunctioning, individual parts can be replaced without affecting the performance of the whole system. We, first, focus on the IR sensor and the LCD screen, since their integration to the system was a challenging task due to the assembly of several electric circuits. The assembly of other components was straightforward.

3.1.1 Infrared (IR) Sensing System

IR sensing system is used to sense the presence of individuals currently accessing the system. It has to be active all the time in order for BASS to function properly. We used three hardware components for the sensing system: (i) IR Sensor, (ii) PIC Microcontroller and (iii) Recommended Standard 232 (RS-232) port. Figure 3.2 shows the flow of communications among these components.

Figure 3.2: Communication in infrared sensing system

3.1.1.1 IR Sensor

We chose "SHARP GP2D12" sensor that can sense anything in between 10 to 80 cm distance. It has a power input and an analog voltage output. The voltage output is inversely proportional to the distance between an object and the sensor. It uses triangulation to make measurements. Different sensors could also be used for different sensitivities, which would not affect the overall design. The IR sensor is shown in Figure 3.3.

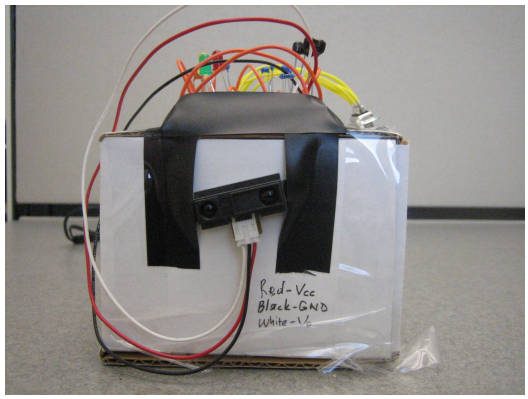


Figure 3.3: Infrared sensor of BASS

3.1.1.2 PIC Microcontroller

Proper communication is ensured through PIC (peripheral interface controller) microcontroller, which is the heart of the sensing system. It receives an analog input from the sensor, then passes it

through an Analog to Digital Converter (ADC) and finally transfers the digitized input to RS-232 serial port. For microcontroller, we chose a widely used "PIC16F876 chip", the schematic of which is shown in Figure 3.4. We used the "RA0 (PIN2)" port for the analog input from the sensor and the "RC6 and RC7 (PIN17 and 18, respectively)" ports for the serial communication.

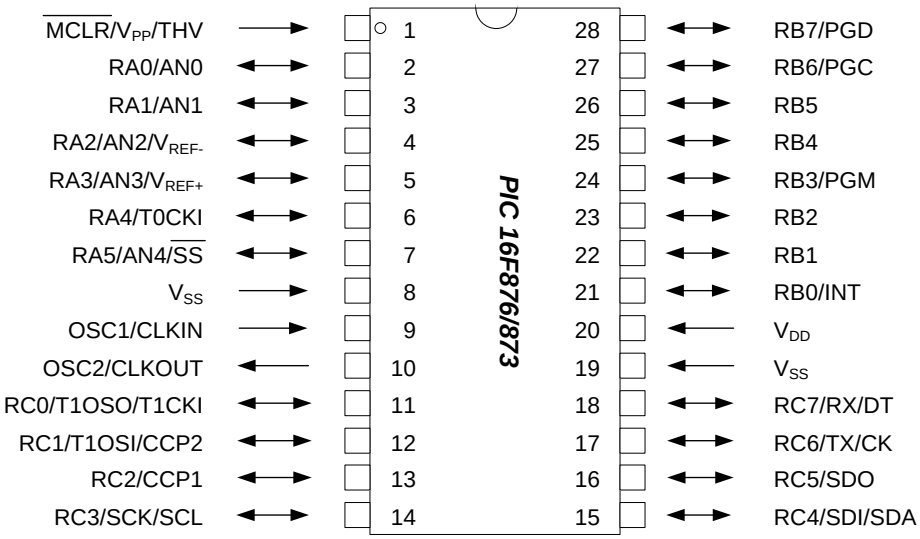


Figure 3.4: The schematic of PIC16F876 chip

3.1.1.3 RS-232 Serial Port

The IR sensor readings that are collected and passed through an ADC then are transmitted through an RS-232 serial port. RS-232 is a standard for serial binary data signals. It is a link between the computer and the IR sensor system. It consists of a standard 9-pin D-subminiature (so-called DSUB) port that connects to the RS-232 port on the PC and a "HIN232CP" chip that serves as the interface between the port and the PIC microcontroller. Figure 3.5 shows the schematic of the "HIN232CP" chip.

The connection diagram for the whole sensing system is shown in Figure 3.6(a), while Figure 3.6(b) shows the actual implementation in the laboratory. In our design, we used a "X0-543" oscillator in the circuit. The values of capacitors were carefully chosen for the system to function properly. The use of other chips may require different values for capacitors. The computer code for PIC microcontroller was written in C to synchronize all components. To finish, MATLAB®

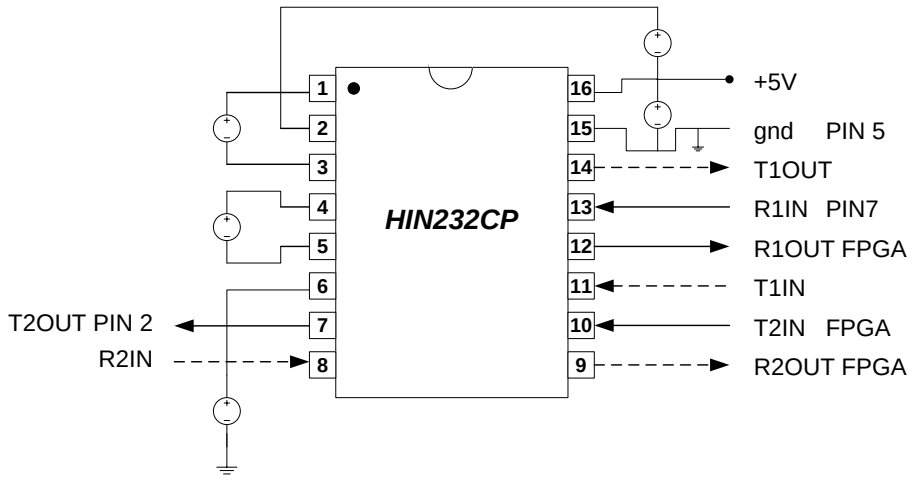


Figure 3.5: The schematic of HIN232CP chip

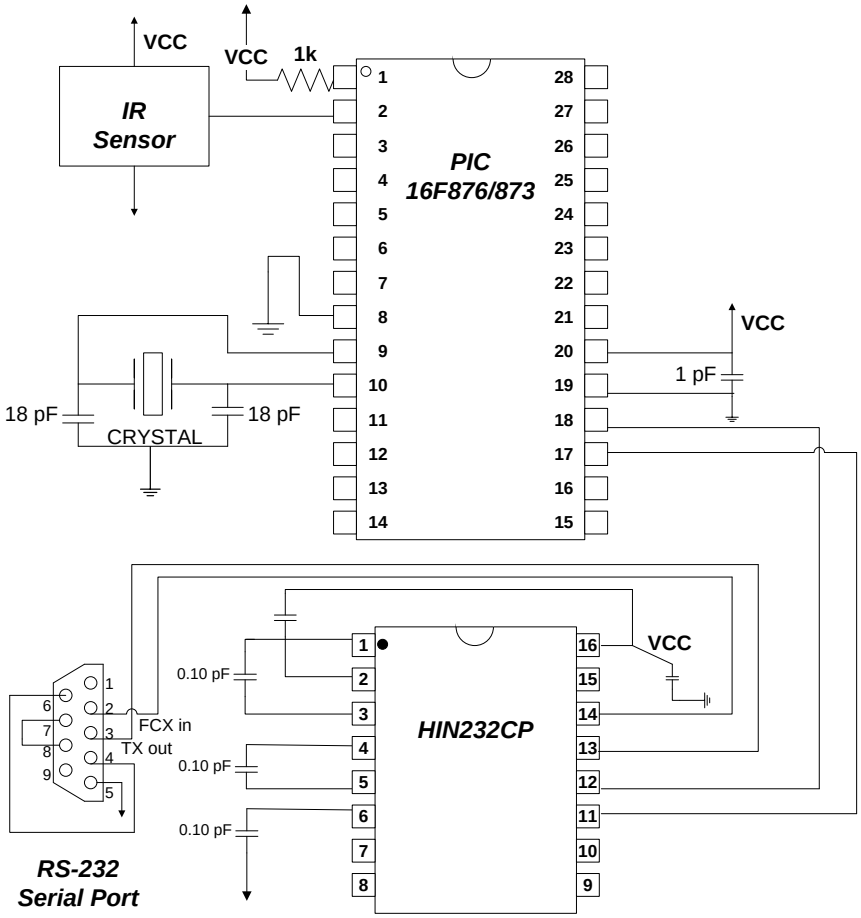
external interfaces for sending serial port signals and assigning pins provided us with an extensive flexibility.

3.1.2 LCD screen

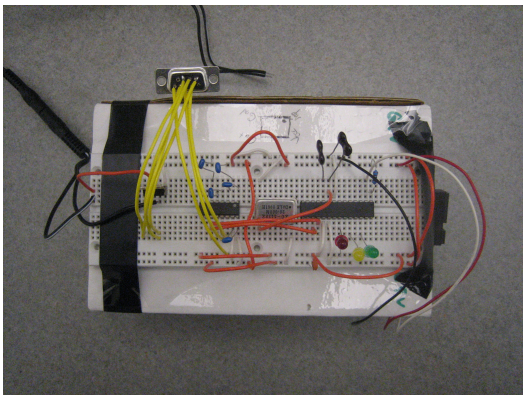
A liquid crystal display (LCD) screen is added to BASS to show randomly chosen passwords to the users. We used "MOS-AL162A-YX" LCD screen produced by Matrix Orbital®. It contains many favorable features for our system such as being capable of conveying data at a rate of at least 9.6 kbps and showing lengthy passwords. It is connected to the computer using the same serial port that IR sensor uses: however, it can only receive signals for display purposes. The connection diagram of LCD screen is provided in Figure 3.7. BASS starts out by sending an introductory message to this screen after a user is detected by the IR sensor. The passwords and the decision of speaker recognition software are also shown to the user in the same way. Figures 3.8(a) and 3.8(b) show the LCD screen when it is unattached and in use, respectively.

3.1.3 Microphone

The recordings for BASS were made using a unidirectional, noise canceling microphone, with a frequency response between 100 Hz and 16 kHz (Logitech Model Number: 980240-0914). The input sensitivity of the microphone is $-67dbV/\mu bar$ ($-47dBV/Pa \mp 4dB$). Despite the fact that a head-mounted microphone should be used to eliminate reverberation and standardize the speaking



(a) Design



(b) Implementation

Figure 3.6: Design and implementation of PIC microcontroller, HIN232CP chip and RS-232 serial port assembly

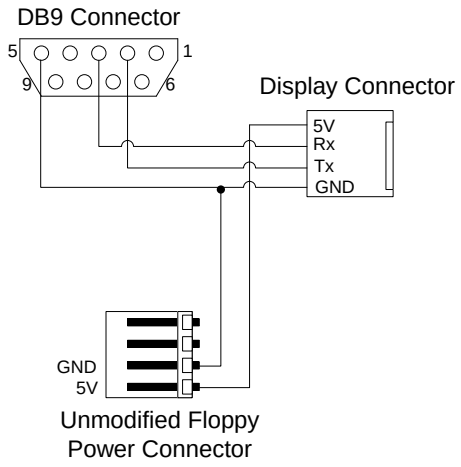


Figure 3.7: Connection diagram for the LCD screen and RS-232 serial port

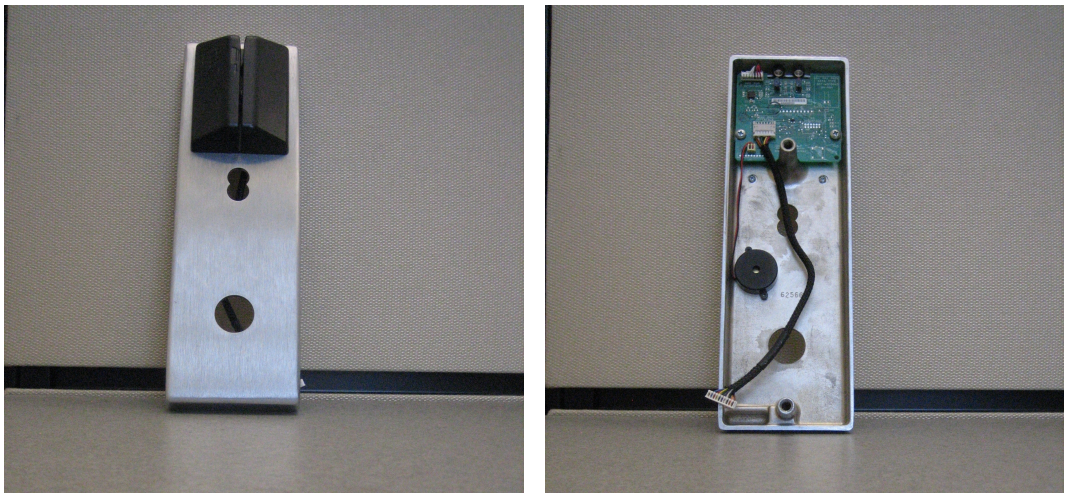
(a) Disconnected (b) In use

Figure 3.8: The LCD screen to display passwords

distance, this microphone was selected for all practical reasons. In particular, an omnidirectional microphone was not selected since it could easily capture the noise from all the directions, which could have made the pre-processing of recorded sound files much harder.

3.1.4 The Door Security System

The whole hardware assembly explained above can be integrated to existing door lock systems, which actually operate when the users slide their cards. These systems are designed such that each door can be controlled by a centralized security system, on which the security personnel have full access. Further details of the door security system and its connections cannot be provided here for



(a) Card Sliding Mechanism (b) Cable Connections

Figure 3.9: The door security system

security purposes, since it is actively used in the Siebel Center. Figures 3.9(a) and 3.9(b) show the card sliding mechanism and its cable connections when it is detached from the centralized system.

Finally, the communications among hardware components are successfully made through a computer, where the speaker recognition software is installed. The hardware verification experiments were performed to check if the system worked fast and robustly. No malfunctioning was reported even when the system was used excessively. In the case of electricity cut-off, however, the system is automatically deactivated and the doors are kept locked to redirect the users to the authorized personnel. The overview of the hardware, including the computer, is shown in Figure 3.10.

3.2 Software Architecture

In this section, we provide an overview of the software component of BASS. We explain the details of tasks for performing speaker recognition. We also provide the background information we used in digital signal processing, since there are different design choices that can be made in authentication studies.

Figure 3.10: Biometric authentication system for the Siebel Center

The main function of the software is to process the speech for creating feature vectors and to use them for training or testing classifiers. For this purpose, there are 5 main tasks that need to be performed sequentially:

1. Speaker Data Collection
2. Front-end Processing
3. Enrollment or Authentication
4. Pattern Matching and Scoring
5. Decision

The flow of these tasks in software component of BASS is also given in Figure 3.11.

Speaker data collection can be performed in a controlled or natural environment, both of which can easily be affected by the ambient noise. Pre-processing is necessary to eliminate the noise from the signal and cut the silenced regions of the recording to form feature vectors representing speaker information. After successful extraction of features, any individual's voiceprint can either

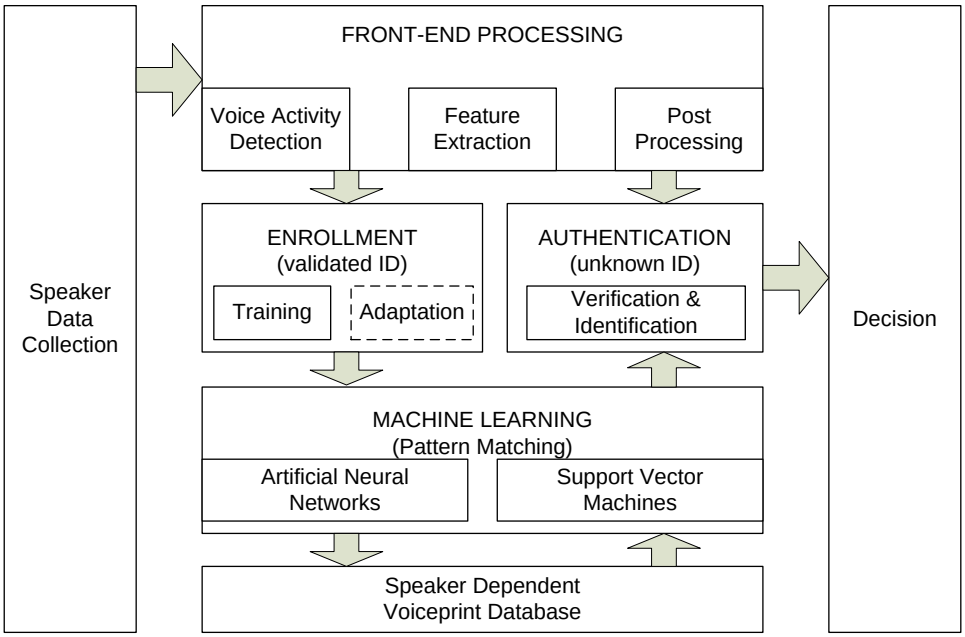


Figure 3.11: Tasks performed by the software component of BASS

be enrolled or authenticated in BASS using the machine learning algorithms, SVMs and ANNs. Finally, the decision is made according to the principles of verification or identification. In the next section, we will explain the steps for feature extraction (so-called "Front-end Processing"), which plays a crucial role amongst all.

3.2.1 Front-end Processing

The front-end processing is done to extract the speaker dependent information reliably and use this either to construct a voiceprint model for the speakers or to test the existing authentication system. The characteristics of the information obtained from speakers can be classified using three aspects [95]:

- *Temporal Span:* The speaker dependent information stored in speech signal can be captured at different time spans and rates. In addition, the features should be able to follow the variations at these time spans and rates. To ensure this, the vocal tract is analyzed by separating it into tiny portions stretched on short time spans and using the frequency spectrum of the speech.
- *Discrete and Continuous Values:* Features that contain speech frequency spectrum samples are typical continuous features while the ones including the number of word usage counts or

the counting of any event in a signal illustrate discrete features. The conversion among these features is possible in both ways.

- *Information Level:* The information included in speech can be at different levels such as the semantic meaning of the words or the speaker's vocal tract, etc. Features at a low level generally extract acoustic characteristics while the high level information may include pronunciations that may be retrieved from a word recognition system. Proper use of high level information and finding new representative features are challenging in both speech and speaker recognition areas [18].

The space of these features is illustrated in Figure 3.12. The main difference between different speakers' voice is due to the complexities of co-articulation, where the spectral characteristics of a word depends upon the surrounding phones or words. However, several other factors may also contribute to variability in speech signal. They can be summarized as [22]:

- Speaker Differences
- Speaking Style
- Channel/Environment

These sources of variability should be handled by the speaker recognition system successfully. For this to happen, front-end processing needs to be performed accurately.

The front-end processing generally includes three sub-processes (i) voice activity detection, (ii) feature extraction and (iii) post processing, the details of which are explained in the next sections.

3.2.1.1 Voice Activity Detection (VAD)

VAD, also known as Endpoint or Speech Detection, is the task of determining speech portions of a continuous signal with the background noise, and removing the silenced parts. VAD is an integral part of any speech recognition application in the sense that it provides efficient computation of speech features and also prevents the need for transmission of silence packages through telephone networks. In addition to these, it has variety of benefits in speech related applications [75, 56].

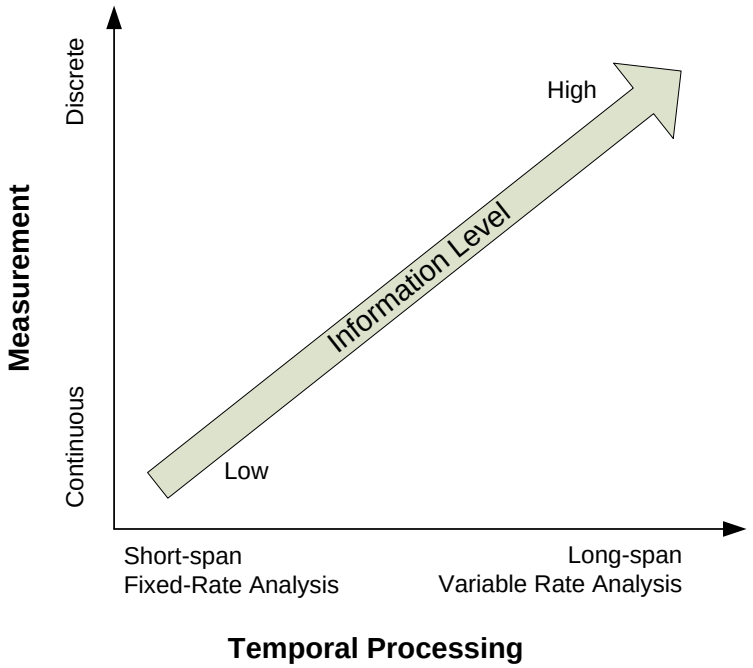


Figure 3.12: Feature space used in speaker recognition systems [95]

In general, word isolated, text dependent systems are very sensitive to accurate determination of voiced portions of speech signals. However, certain recognition systems model the silenced and voiced portions as a continuous signal using HMMs, which allows the determination of precise locations of the speech [54, 92]. Most recently, SVMs have been used for this purpose [58].

VAD is a challenging task. The algorithms developed for this purpose need to handle certain situations that may happen in a given record. For example, specific sounds (i.e., /f/ as in "thief" and /h/ as in "happy"), weak plosives (i.e., /p/ as in "pot", /t/ as in "loot" and /k/ as in "cow"), final nasals (i.e., "pin" or "calm"), trailing of sounds (i.e., "three"), and voiced fricatives becoming unvoiced, (i.e., "has") [22] may create difficulties when removing silenced parts. In addition to these, an efficient VAD should work even in the case of extremely noisy conditions. To overcome the difficulties encountered, many methods have been proposed. A comprehensive overview of these is given in [98].

Although VAD by itself is a major research topic, it can be said that most VADs work by observing the zero crossing rates and/or the spectral energy of the speech signal. The short-term energy of the signal is computed as the sum of the squares of the amplitude of the signal samples

in a frame [80, 38]. Another feature that can be used for VAD is a measure of zero crossings. This feature represents the number of times a signal has changed its sign within the frame. Different types of audio signals have different zero crossing measurements. Generally, voiced speech has fewer zero crossing measurements than unvoiced signals. We implemented a well-known algorithm [86] in this work, the details of which are explained in the next chapter.

3.2.1.2 Feature Extraction

In this section, we explain the formation of acoustic features using the preprocessed voice records. The term *features* represents the vector of numbers which represent one time slice of a speech signal. There are many ways of representing voice signal at a low level such as using Linear Predictive Coding [83, 67, 96], Warped Linear Predictive Coding [64, 65], Perceptual Linear Prediction [53] and Mel-Frequency Cepstrum based features. A comparative study about the use of some of these features in speech recognition can be found in [37]. In addition, there are various features that can be used as representative features for recognition purposes. For example, codebook quantized spectral entries measure the approximate location of spectrum in acoustic space [85]. Pitch and energy [9] also provide low level information about speakers. Finally, various high level information such as prosodic statistics [81] and word and phone tokenization have been utilized in speaker recognition systems [40, 74, 70, 28, 95].

3.2.1.3 Mel-Frequency Cepstral Coefficients (MFCCs)

Among many possible low level features, the most commonly used ones are MFCCs. They have proved to robust representations of a speech signal and successfully used in both speech and speaker recognition studies [84, 50]. In this section, we explain how to obtain MFCC coefficients from a row waveform.

There are mainly eight steps to retrieve the MFCCs from a given raw signal, which are shown in Figure 3.13.

1. *Pre-emphasis:* The raw speech signal generally includes more energy at lower frequencies compared to higher frequencies, especially for voiced segments like vowels. In order to enhance (or lift) the amount of energy in the higher frequencies, the signal is sent to a high-

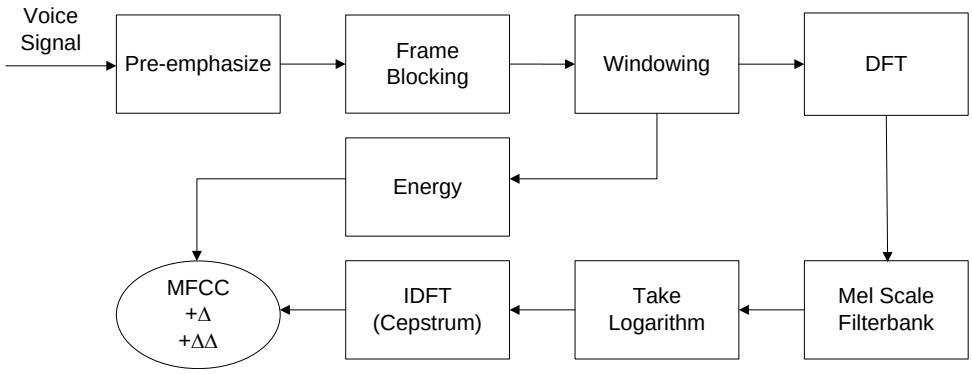


Figure 3.13: Calculation of mel-frequency cepstral coefficients

pass filter, which results in emphasizing the information on higher formant to be used in the acoustic model. The filter is designed to be a first order high pass filter such that:

$$y(n) = x(n) - \beta x(n-1) \tag{3.1}$$

where $x(n)$ and $y(n)$ are the input and output signals, respectively, and the value of β is usually between 0.9 and 1.0. We used $\beta = 0.97$ in the scope of this thesis.

2. *Frame Blocking:* The spectrum of a signal for the whole utterance changes very rapidly. However, within a given region, the desired features of the speech signal should be constant as much as possible (so-called stationary signal) so that these can be used to characterize that portion of signal. Extracting this information can be achieved using a window, which only operates on the region where it is specified. It is not functional (i.e., zero) anywhere else. The extracted speech from a window is called a frame, the duration of the sampling is called the frame size and the time between successive frames is called frame overlap (frame shift). Generally, the input speech signal is segmented into frames of 20 to 30 ms with overlap size from $\frac{1}{3}$ to $\frac{1}{2}$ of the frame size. Furthermore, the number of sample points need to be equal powers of 2 in order to facilitate the use of Fourier Transform. Otherwise, zero padding operation is performed to the nearest length of power of 2. For example, if the sample rate is 8 kHz and the frame size is 160 sample points, then the frame duration is $(160/8000) \cdot 1000 =$ 20 ms. In addition, if the overlap is 80 points, then the frame rate is $8000/(160-80) = 100$ frames per second.

3. *Windowing:* The information from a given signal, $s(n)$ at a time n can be extracted using the windowing function $w(n)$ in Equation 3.2.

$$y(n) = w(n)s(n) \tag{3.2}$$

There are mainly three types of windowing functions namely: (i) rectangular, (ii) hamming and (iii) hanning, the definitions of which are given in Equations 3.3, 3.4 and 3.5, respectively.

$$w(n) = \begin{cases} 1 & if\ 0 \leqslant n \leqslant L-1 \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

$$w(n) = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{L}\right) & if\ 0 \leqslant n \leqslant L-1 \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

$$w(n) = \begin{cases} 0.5\left(1 - \cos\left(\frac{2\pi n}{L}\right)\right) & if\ 0 \leqslant n \leqslant L-1 \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

4. *Discrete Fourier Transform:* Discrete Fourier Transform (DFT) is used to extract the spectral information from a portion of a signal windowed at the previous step. The input to DFT is windowed signal $x(n)$ to $x(m)$ for each N frequency bands and the output is a complex number $X(k)$ that represents the magnitude and the phase of that component in the original signal, which is given in Equation 3.6 [59].

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi}{N}kn} \tag{3.6}$$

where j is the imaginary unit on Euler's Formula, which is given in Equation 3.7.

$$e^{j\theta} = \cos(\theta) + j\sin(\theta) \tag{3.7}$$

Finally, Fast Fourier Transform (FFT) is used for efficient computation of DFT.

5. *Mel Scale Filterbank:* The result of applying FFT to a given windowed signal is actually a representation of energy for each frequency range. However, human hearing is not equally responsive to all these ranges: it is more sensitive to frequencies lower than 1 kHz. In fact, the frequency scale of cochlea in the human ear is actually non-linear and known as mel scale (A mel is the unit of a pitch). This scale has linear frequency spacing below 1 kHz and a logarithmic spacing above this value. The mel-frequency can be calculated using the Equation 3.8.

$$mel(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{3.8}$$

Proper use of the above fact may improve the performance of the recognition systems. Generally is a bank of filters based on the total energy in critical bands around the mel-frequencies are created to be fed into Inverse Discrete Fourier Transform (IDFT) by applying a half overlapped triangular window with increasing length centered on the mel frequencies [22] (To illustrate the concept, the mel filter bank obtained using the Praat speech processing software [16] is shown in Figure 3.14).

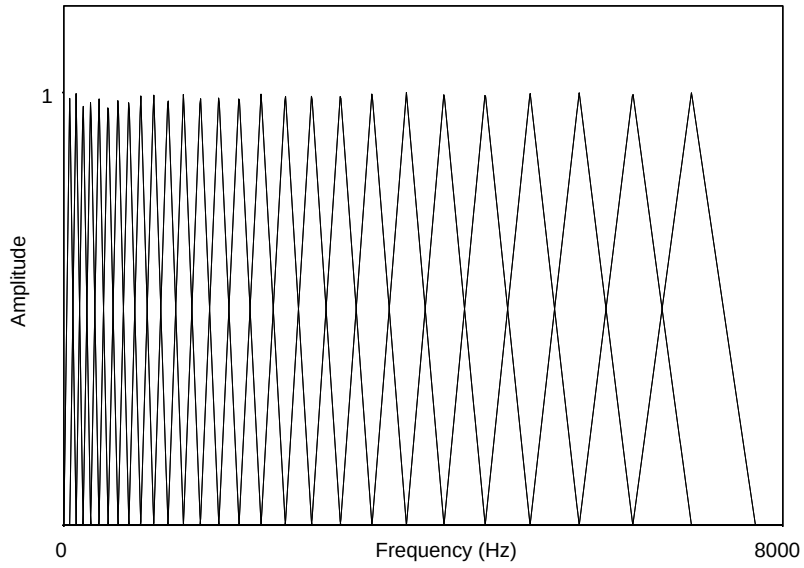


Figure 3.14: Mel-frequency filter bank obtained using the Praat software [16]

6. *Inverse Discrete Fourier Transform (Cepstrum):* The final step to calculate MFCCs is to calculate the cepstrum. Cepstrum (anagram of spectrum) is a way to separate the source and the filter. The human speech signal is the convolution of the voiced excitation sequence and impulse response due to the vocal system. The convolution in the time domain is the equivalent of multiplication in the frequency domain, therefore:

$$S(w) = E(w) \cdot H(w) \tag{3.9}$$

where $S(w)$ denotes DFT of the speech signal, $E(w)$ represents DFT of the excitation and $H(w)$ is DFT of the vocal system impulse response. After taking the logarithm of Equation 3.9 on both sides, the term $\log|S(w)|$ can be interpreted as a periodic signal made up of two linearly combined parts. The first term, $\log|E(w)|$, can be thought as a high frequency component, while the second one, $\log|H(w)|$, can be thought as a low frequency component. Since this is not the formal frequency domain, it is referred to as *quefrencies*. Since $\log|S(w)|$ is periodic, we can determine the Fourier series coefficients corresponding to the harmonics of the signal. The term $\log|S(w)|$ is a real and even function of w. Therefore, we can equivalently use the IDFT to determine Fourier series coefficients, which is the standard definition of real cepstrum [22].

   For the purposes of feature extraction in speaker recognition applications, we are interested in low quefrency components which represent the voice system response. They can be extracted using a window that can skip the excitation impulses, or equivalently we can take the first N cepstrum coefficients. The main advantage of use of MFCCs is that normal speech waveform may vary from time to time depending on the physical condition of speakers vocal cord. However, MFFCs are less susceptible to these variations [84].

7. *Energy and Deltas:* In general, there are 12 MFCCs for each frame. In addition, the energy of a frame can be added as a separate feature, which may be used for recognition using phone lines. With the addition of energy feature (the definition of which is given in VAD section), there are 13 features that can effectively be used in speaker recognition.

Finally, to measure the change of cepstral features from frame to frame, two different features namely Delta (Δ) and Delta Delta ($\Delta\Delta$) can be added to the feature vector, which are also known as velocity and acceleration features, respectively. As the names imply, Δ and $\Delta\Delta$ represent the derivatives with respect to a given time. Δ represents the change in cepstral and energy features while $\Delta\Delta$ represents the change of Δ features in time. Since there were originally 13 feature vectors, with the addition of these, total of 39 features can be obtained.

3.2.1.4 Post Processing

The last step in front-end processing is to handle the channel compensation. If the recording is made using different input devices (i.e., different microphones, phones handsets etc.), then this will impose different spectral characteristics on the speech signal such as bandlimiting. In order to create a robust recognition system, it should be independent of the input device and these channel effects need to be removed. To do this, cepstral mean subtraction is generally applied to the acoustic features obtained. Other than the linear channel compensation in the feature domain, there are other compensation techniques that can be used at a model and/or match score domains. However, these are not considered in the scope of this study.

3.2.2 Enrollment or Authentication

After successfully obtaining feature vectors, the next step in developing the software is the utilization of these for the purpose of speaker recognition. This can be done in two ways, which are explained below:

1. *Enrollment:* BASS learns characteristics of a person's voice when a new recording is done in the system. During enrollment (i.e., training), the recognition system first analyzes recording and characterizes the voice in a voiceprint model. The identification of speaker should have been previously validated by some other means. After a successful enrollment, the system stores models in the database for use during authentication. It also has the capability to adapt the developed models based on the new input coming from the existing users and/or new impostors. The adaptation part, however, is not included in the scope of this work.

2. *Authentication:* Once a speaker is enrolled, BASS can authenticate the identity of that speaker using new recordings. The system compares the characteristics of the current speaker with the voice model for that speaker. It calculates the scores for new recordings and compares them to a threshold value for acceptance/rejection decision. If the score is above the threshold, then the user is granted access. BASS can also identify the speaker based on the scores and decide to report "none of the speakers", which is a more generalized version of closed-set identification, called "open-set speaker identification".

The technical details of these steps will be explained next.

3.3 Machine Learning Algorithms

A review of the learning algorithms we use in this study was provided in Chapter 2. In this section, we explain the details of training and testing procedures for both speaker verification and identification. We focus on the details of the score calculation from given feature vectors based on a properly trained algorithm, since it plays a crucial role in the performance of the whole system. We, finally, review the performance measures we utilized to quantify the success of our models.

3.3.1 Training and Testing

BASS is capable of calculating several features given a record from a speaker. In BASS, for a specific frame, the number of features is at most 39. However, the actual number to be used in training is generally found by experiments with a classifier trained using various feature combinations. Previous works produced successful results with different number of MFCCs in a given training vector [55, 32, 103]. We, therefore, tried different arrangements of MFCC, Δ and $\Delta\Delta$ features in training of SVM and ANN classifiers. In general, increasing the number of MFCCs results in increasing complexity of the learning problem, given a classifier. However, depending on the discriminative characteristics of features, the recognition problem may become easier to solve by increasing the number of features [30]. The formation of feature vectors from a given record is shown in Figure 3.15.

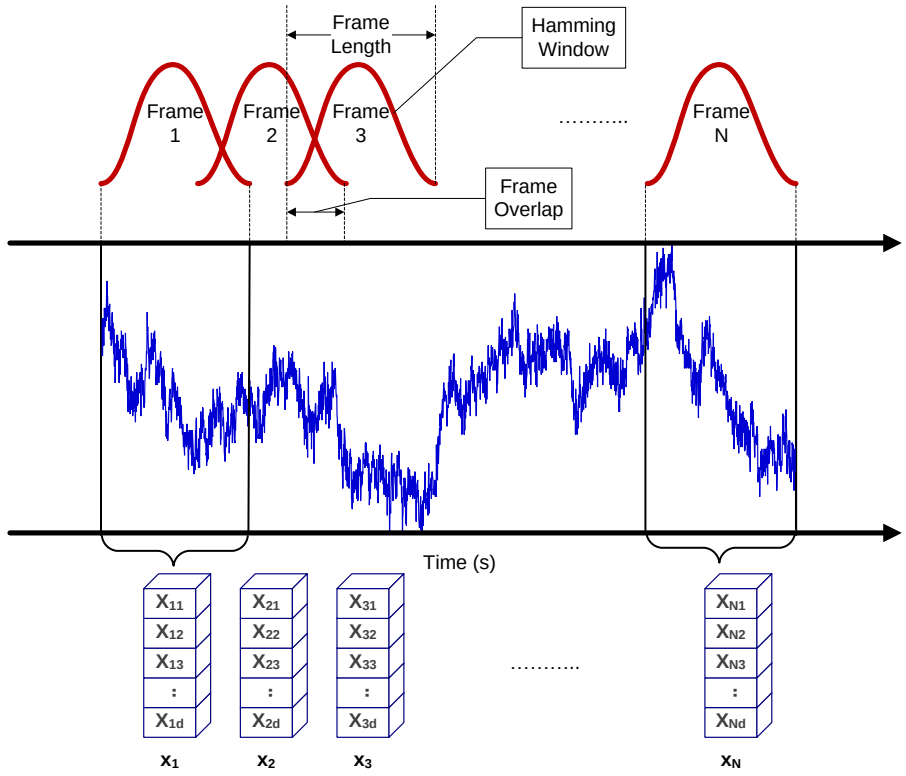


Figure 3.15: Feature extraction with windowing

Properly extracted feature vectors are then collected to form feature vectors for training purposes. Since the length of an utterance differs from speaker to speaker, the length of feature vectors will be different if they are combined sequentially. To eliminate this, there are many methods proposed in the literature which have proved to be successful in experiments performed even with large databases [102, 103, 100, 26]. In this study, unlike in many others, the sequential ordering of feature vectors is not preserved during training of neither SVMs nor ANNs. This, at first, may seem contradictory to what have been proposed in recently published works (especially the ones using HMMs, GMMs or combination of these methods with SVMs). However, the power of SVMs and ANNs to learn complex relations should not be underestimated. This is especially be emphasized in closed-set identification problems and verification of limited speaker data sets. To illustrate this, the resultant feature vectors are labeled as $+1$ for a speaker whose identity needs to be validated (generally called client) and -1 (or 0) for impostors. This is called "One vs. all" strategy and ideally, test vectors for client should have $+1$ response, while the ones for the remaining speakers have -1 (or 0).

There are many choices for the selection of kernel function (Equations 2.11 - 2.14) to train speaker verification models. Researchers have written problem specific kernels that are slightly modified forms of existing ones, which have proven to be successful in certain databases [101]. In this study, we chose Radial Basis Function (RBF) kernel for the training of SVMs. Since the scalability of SVMs with large training sets is limited, it may be better to train SVMs with small subset of all examples. Otherwise, the data may be inseparable resulting in very large number of support vectors after training. This may also include misclassified data points as the support vectors. Finally, the storage requirements and therefore the computational power need to be properly handled in the case of very large data sets.

The training of ANNs is very similar to that of SVMs at an abstract level. In this thesis, we used backpropagation learning algorithm to train a MLP classifier with the data previously used to train SVMs. We also used adaptive learning rate and momentum rather than implementing steepest descent that keeps the learning rate constant. This makes the performance less sensitive to choosing the learning rate and momentum coefficients. ANNs were trained on 2 layers, one being hidden. We used hyperbolic tangent sigmoid (so-called "tansig") transfer function, given in Equation 2.18, between layers of ANN, which converts the inputs to $[-1, 1]$ interval. We started with a large neural network that is capable of learning relations with high degree of complexity, and then we gradually reduced the size of ANN and stopped developing the models at a level where we were able to preserve the models' generalization capabilities. Mean square error criterion was chosen to stop the training. We used 4-fold cross validation to determine the threshold values that maximizes the distance between the positive and negative examples. We also limited number of epochs to 2000, however, the training never reached to this number due to the cross validation.

Using frame-based acoustic feature vectors without considering their sequence also causes differences in interpreting the results of both verification and identification. Frame-based evaluation of feature vectors during testing will not yield accurate results due to both limited data obtained from client during training of classifiers (i.e., unbalanced training data) and not preserving the sequence of these vectors for robust modeling of clients. Therefore, we used a scoring policy based on utterances rather than features for both SVMs and ANNs. In the next section, we explain how to calculate the scores for both clients and impostors to make decisions.

3.3.2 Score Calculation

The decision criterion for SVM classifier is given in Equation 3.10. In addition, the activation of SVM is provided in Equation 3.11. Since the decision of SVM is made using features for each frame, we take the average of the activations of SVM for each acoustic vector to calculate the score of an utterance [101] (Equation 3.12). A detailed expression to calculate the score of an utterance is also given in Equation 3.13, where α_s is the Lagrange multiplier for s^{th} support vector and y_s, is the classification label, which is $+1$ and -1 (or 0) for client and impostor, respectively.

$$sign(\mathbf{w} \cdot \mathbf{x} + b) \tag{3.10}$$

$$\mathbf{w} \cdot \mathbf{x} + b \tag{3.11}$$

$$\text{Utterance Score} = \frac{1}{N} \sum_{i=1}^{N} (w \cdot x_i + b) \tag{3.12}$$

$$\text{Utterance Score} = \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{s} \alpha_s y_s K(\mathbf{x}_s, \mathbf{x}_i) + b \right) \tag{3.13}$$

Using utterance score has several advantages when compared to the frame-based scores. In general, the data obtained from both client and impostors is unbalanced since there are many impostors and only one client. This causes classifiers to learn more of the characteristics of impostors than those of client. In addition, using the sign operator on activation of SVM makes the decisions less certain around the decision boundaries (activation is close to zero) since using the sign operator strengthens the signal and should not be permitted to prevent creating an impostor biased model. Utterance score overcomes these problems since the feature vector that is strongly classified as either client or impostor and contributes more to the mean. This also eliminates the need for penalizing the features coming from impostors during training.

ANNs that used mean squared error or cross entropy criteria [15] model the posterior probability, $p(speaker|\mathbf{x_i})$ where $\mathbf{x_i}$ is the feature vector that represents speech of a speaker [19, 49]. The scoring criterion for ANNs is usually the weighted average of posterior for all the frames that represents an utterance of a speaker (Equation 3.14). Similar to SVMs, ANNs may have the tendency to learn impostor speakers better, since the distribution of the speaker population is biased towards "-1" labeled ones. To eliminate this, (i) Vector Quantization can be used to compress the number

of "-1" labels, (ii) noisy data labeled with "+1" can be added to training data set [42] (iii) random sampling with prior equalization can be done in the selection of training data set (iv) the outputs in testing can be scaled with target prior [95].

$$\text{Utterance Score} = \frac{1}{N}\sum_{i=1}^{N} p(speaker|\mathbf{x_i}) \tag{3.14}$$

Calculating utterance score alone is not sufficient for decision purposes. It needs to be compared to a value (so-called threshold) such that if "Utterance Score $>$ Threshold" then the system accepts the speaker, otherwise, the speaker is assumed to be an impostor and rejected. Selection of the threshold is a challenging task as it varies from speaker to speaker. In this work, we chose the threshold values for each speaker during the training process such that it provides the maximum distance between the client and impostor speakers using the cross validation examples randomly selected from the training data. These threshold values were then used to calculate the performances of classifiers on unseen data. The training and testing procedures for speaker verification are given in Algorithms 1 and 2, respectively.

Algorithm 1 Training (Enrollment) Algorithm of BASS for Speaker Verification: During enrollment voice-print model for the client, a speaker whose identity needs to be validated, is created given a set of speakers

for each word, w **do**
  for each speaker, sp **do**
    for each voice record, r **do**
      pre-process the signal to obtain the feature vectors $\mathbf{x_i^{w,sp,r}}$, for each frame $i = 1..N$ (see Figure 3.15)
      normalize $\mathbf{x_i^{w,sp,r}}$ using the mean and standard deviation : $\frac{\mathbf{x_i^{w,sp,r}}-\mu}{\sigma}$
      label each $\mathbf{x_i^{w,sp,r}}$ as $+1$ for the client, and -1 for the impostors
    end for
  end for
  separate the data into 4 parts to perform 4-fold cross validation
  for each fold, f **do**
    train both SVMs and ANNs using samples $\{\mathbf{x_i^{w,sp,r}}, \pm 1\}$, without considering sequence of feature vectors
    test the developed models using a validation set chosen from the training data
    choose a threshold ($T_{w,f}$) for each word that maximizes the distance between the client and the impostors
  end for
  choose the best threshold (T_w) among others that better separates the client from impostors
end for

Algorithm 2 Testing (Authentication) Algorithm of BASS for Speaker Verification: During authentication voice-print model of the client is compared with the one obtained from a new record

for each password **do**
  for each word "w" in the password **do**
    pre-process the signal to obtain the feature vectors $\mathbf{x_i^w}$, for each frame $i = 1..N$ (see Figure 3.15)
    normalize $\mathbf{x_i^w}$ using the mean and standard deviation: $\frac{\mathbf{x_i^w}-\mu}{\sigma}$, the values of μ and σ are calculated during enrollment using the training data
    if classifier (or trained model) is SVM for word "w" **then**
      calculate the utterance score, $U_k = \frac{1}{N}\sum_{i=1}^{N}\left(\sum_s \alpha_s y_s K(\mathbf{x}_s, \mathbf{x}_i) + b\right)$
    else if classifier (or trained model) is ANN for word "w" **then**
      calculate $U_w = \frac{1}{N}\sum_{i=1}^{N} p(speaker|\mathbf{x_i})$
    end if
    if $U_w \geq T_w$ **then**
      label the whole utterance as $C_w = +1$ for the client
    else
      label the whole utterance as $C_w = -1$ for the impostors
    end if
  end for
  if $\sum_{w=1}^{allwords} C_w > 0$ **then**
    accept the speaker
  else
    reject the speaker
  end if
end for

Another attempt is made in this study for identification of speakers given the set of people (i.e., closed-set identification). The easiest way to do this is to develop a separate classifier for each speaker. If there are "n" speakers then "n" classifiers must be trained and the identity of speaker (either client or impostor) is determined from the classifier that yields the largest utterance score [101]. This can be calculated using Equation 3.15 for SVMs.

$$\arg\max_j \frac{1}{N}\sum_{i=1}^{N}\left(\sum_s \alpha_{sj} y_{sj} K(\mathbf{x}_{sj}, \mathbf{x}_i) + b\right), \tag{3.15}$$

where $\mathbf{x}_{sj}$ are the support vectors of the j^{th} classifier and α_{sj} and y_{sj}, are the corresponding Lagrange multipliers and classes.

A similar strategy for ANNs needed to be developed. Separate classifiers were trained for each speaker using again "One vs. all" strategy. Then a given record (i.e., unseen data or test vector)

was applied to each ANN model and outputs were collected. The identity of speaker was then decided based on ANN model producing the maximum accumulated output for the given utterance (Equation 3.16).

$$\arg\max_{j} \frac{1}{N} \sum_{i=1}^{N} p(speaker|\mathbf{x_i}) \tag{3.16}$$

3.3.3 Performance Measures

In speaker recognition applications, the performances of proposed models can be measured in several ways. One way is using F Measure, precision, recall and accuracy, the definitions of which are given in Equations 3.17 - 3.20. These criteria can be used to evaluate the accuracy of authentication models.

$$F\ Measure = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{3.17}$$

$$precision = \frac{True\ Genuines}{True\ Genuines + False\ Genuines} \tag{3.18}$$

$$recall = \frac{True\ Genuines}{True\ Genuines + FalseImpostors} \tag{3.19}$$

$$Accuracy = \frac{True\ Genuines + True\ Impostors}{All\ Examples} \cdot 100 \tag{3.20}$$

Another and more popular way to interpret the performance of a biometric system is to use False Rejection Rate (FRR), also known as Type I error and False Acceptance Rate (FAR), also known as Type II Error. The definitions of FAR and FRR are given in Equations 3.21 and 3.22. A *false acceptance* means that an impostor is incorrectly authenticated, and a *false rejection* is a client who is incorrectly identified as an impostor. Once these are determined, we can also determine Equal Error Rate (EER) of the system, the error rate at which FAR and FRR are equal, the calculation of which is explained in the next chapter.

$$FAR = \frac{Accepted\ Impostors}{Total\ Impostors} \cdot 100 \tag{3.21}$$

$$FRR = \frac{Rejected\ Genuines}{Total\ Genuines} \cdot 100 \tag{3.22}$$

Chapter 4

Experiments

In this section, we describe the experiments performed in a laboratory to validate the performance of BASS. First, we start with explaining the details of data collection. We then present illustrative results of pre-processing of the collected voice records at various stages such as hum removal, speech enhancement, etc. We also demonstrate the results of applying different VAD procedures to remove silenced parts. Next, we give the details of forming the acoustic feature vectors from pre-processed records and post-processing of these vectors. Using these feature vectors, we then explain details of training SVM and ANN models for the purpose of speaker verification, together with the results we obtained. We also provide the models to be used for speaker identification. We conclude with the discussion of results and some challenges we faced during the development of BASS.

4.1 Voice Database

To validate the proposed method for speaker recognition, we needed to collect voice samples from different speakers. Since BASS will be used in the Siebel Center, the experiments were performed in one of its computer laboratories. It is an isolated (presumably noise free) place and it simulates the office environment where BASS will be built after its performance is verified. A total of 11 speakers participated in the experiments, whose names and gender information are given in Table 4.1. Some speakers were selected amongst the people working in the Siebel Center, while the others volunteered to participate. Considering the multi-cultural environment of the Siebel Center, most of the speakers were from different nations, and therefore had different accents. Among the volunteers, 3 were native English speakers, while the remaining ones were fluent in English, although English was their second language.

Table 4.1: Participants of BASS experiments

Name	Sex
Barış	M
Çiğdem	F
Dan	M
Gabriel	M
Lale	F
Nazlı	F
Nejan	F
Onur	M
Özgül	F
Serdar	M
Thompson	M

4.1.1 Passwords

When BASS performs verification in text-dependent mode, the speaker enrolls with a password phrase that is subsequently used for authentication. In addition to the security benefit of a secret password phrase, the audio characteristics of the password phrase can significantly affect recognition performance. In general, the password phrases should have the following characteristics [76]:

- *Number of enrollment repetitions:* Usually, three repetitions may provide a good balance between accuracy, processing time, and user convenience.
- *The duration of utterance:* It should be at least 1.5 seconds in length. A longer phrase generally provides better accuracy as it may include more speaker dependent features.
- *Number of syllables:* The words should have at least five syllables to enrich phonetic variability.

To determine the passwords of BASS, the individuals were asked to speak 13 words including numeric digits (0-9) as shown in Table 4.2. Each word was repeated 50 times. In addition, we obtained extra recordings to replace the ones that may be corrupt for some particular reason or possibly have extensive noise. Within the scope of this thesis, we intended to use the combination of these single words as passwords to help better verification of the client's voice. Although our intention was to use the random triplets for authentication, the appearance of some words in triples may be limited. The main reason behind this was to reduce the rate of false acceptance and false rejection

Table 4.2: Words used in BASS experiments

Word	Number of Repetitions
Zero, One, ... Nine	50
Trial - Siebel - Dan Roth	50

due to poor performances of recognition models for such words. In short, we chose which words were to be used more frequently in passwords after the success of the each model was validated with unseen test data.

4.1.2 Recordings

Many speaker recognition systems use 8 kHz frequency for recordings, the objective of which is to replicate phone bands. In this thesis, however, we aimed to create a single entry system to be used only for offices in the Siebel Center. Therefore, we did not restrict ourselves to phone band frequencies. As a result, we used higher quality settings for recordings: the speakers' voices were recorded at a frequency of 44.1 kHz and using 16 bits. The duration of the recording for each word was 2 seconds. Speakers also had a chance to take a break after every 25 repetitions to refresh themselves. An example of a record is shown in Figure 4.1, which was obtained using the Praat software [16].

As mentioned in the previous chapter, MATLAB was utilized to develop specific components of BASS as well as for recordings during the experiments. Its toolboxes, which include Digital Signal Processing Toolbox, were also highly utilized. Although some individual tasks such as detection of voice activity were explicitly coded for BASS, well established voice processing tools such as MFCC calculation were directly embedded into the software. Finally, MATLAB was also used to communicate between the software and hardware components using the serial port, as explained in Chapter 3.

4.2 Front-End Processing of Voice Records

One of the most important steps of speaker (or speech) recognition is to eliminate the noise and to deal with reverberation, which will affect the performance adversely if not handled properly. Therefore we spent a considerable amount of time investigating our records by listening them individually.

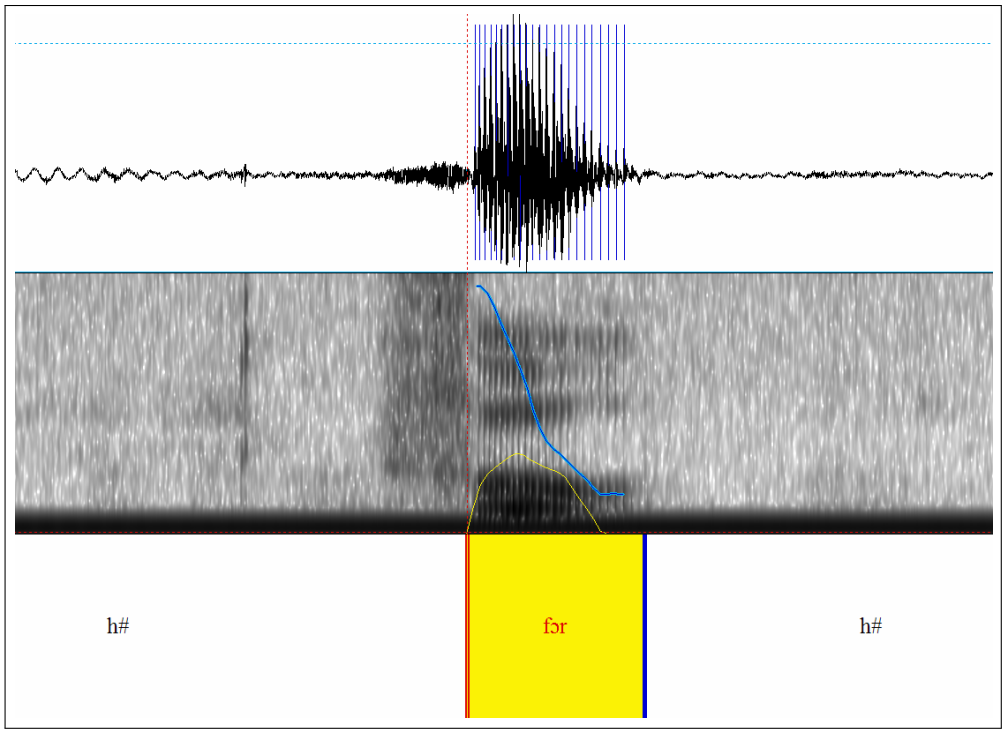


Figure 4.1: The acoustic wave record for the word “Four” spoken by Dan Roth: the spectrogram, the phonetic transcription, pitches and intensity are shown.

We processed them to obtain good quality records and used them as inputs to form acoustic feature vectors. We observed that the error rate of speaker recognition changed considerably according to the design choices we made in each step of pre-processing since the actions of pre-preprocessing were relying on the results of the previous one.

4.2.1 Removal of Hum

The laboratory environment was thought to represent the conditions where the system is intended to be built. The real environment where the system will be used may include many types of noise that may vary from a hum to audible noises such as stationary noise (fan or motor noise), or non-stationary ones (music or varying background speech). Therefore, we first applied 60 Hz hum filter to all records. The comparison of the records obtained before and after this filter was applied suggested that there was no need to eliminate hum from the waveform.

4.2.2 Spectral Subtraction

This step was mainly performed to remove the stationary noise and to enhance the speech before determining the voice activity. There are many spectral subtraction methods available in the literature with their own strengths and weaknesses. We first implemented the spectral averaging and residual noise reduction in MATLAB according to the algorithm that was proposed in [17]. The first 0.25 secs of the speech signal were assumed to be noise only and were used to model the noise signal. Then, we used the algorithm available in Voicebox for MATLAB [20, 14, 69] to verify our findings and to improve the robustness of the first implementation. We compared the resultant speech files after denoising and found that the correlation coefficient between the outcome signals as a result of these procedures was 0.976. Since both approaches produced very similar results, the latter approach was then decided to be used in BASS to denoise all the recordings. An example for denoising of the speech file for "Zero" spoken by the speaker "Barış" is given in Figure 4.2.

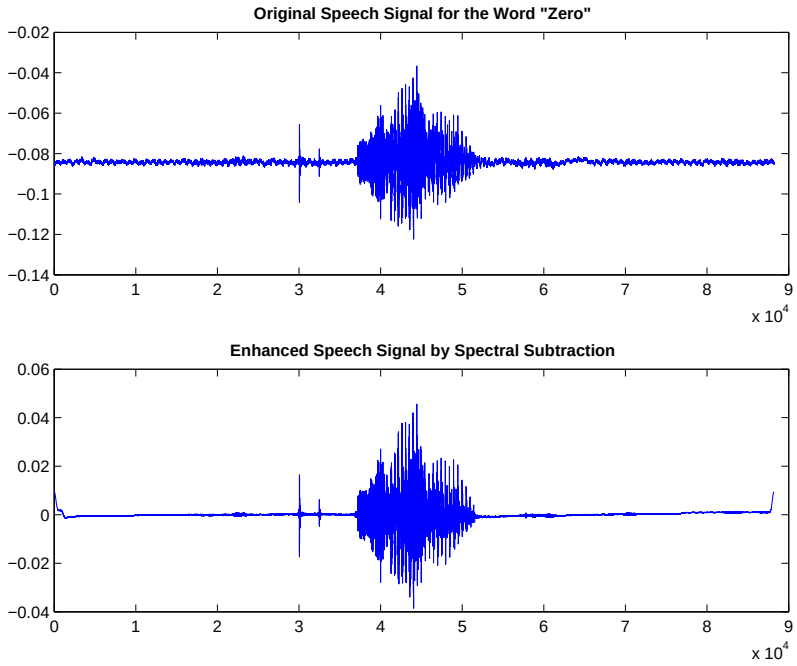


Figure 4.2: Enhancement of speech file for the word: "Zero" using spectral subtraction.

4.2.3 Voice Activity Detection

As mentioned in the previous chapter, a successful speaker recognition system requires a reliable VAD algorithm to identify the voiced and silent regions in a speech file. It is the most important step in developing a text dependent speaker recognition system. In this study we implemented 3 VAD algorithms [86, 93, 66] and compared their performances. These 3 VAD implementations are based on:

1. Short term energy and zero crossing rate [86]
2. The decision rule and the noise statistic estimation algorithm [93]
3. Minimum mean-squared error, a posteriori estimation of noise [66]

Our first implementation [86] uses the short term energy of a signal and the number of zero crossings measured over a frame of 10 ms duration. In this algorithm, the upper and lower thresholds of energy are chosen by analyzing the first 100 ms of a signal, which are assumed to be silence. A single threshold for zero crossings is also determined for this period. The algorithm first operates on the energy by searching the points at which energy goes above the upper energy threshold, and then back tracks to find the points at which the energy crossed the lower threshold. This determines the beginning of the signal. The same process is repeated starting from the other end of the speech signal to determine the end point. Finally, the algorithm effectively uses zero crossing rates to search backwards from the beginning point over 25 frames counting the number of intervals greater than zero crossing threshold. If this is greater than a constant (3 was chosen in the original paper), the beginning point is modified by recording the new one which first surpasses the zero crossing threshold. The end point can also be updated using a similar approach.

The details of the second and the third algorithm we used for comparison were discussed in the corresponding papers [93, 66]. The results we obtained, as shown in Figures 4.4 - 4.5, indicated that the performances of all three implementations were acceptable. The success of VAD based on the short term energy and zero crossing rate was dependent on the threshold values which were selected differently for each speaker. The second and third algorithms also produced very good estimations of voiced regions without any adaptive approach. Since it proved to be successful in many applications, we chose the first implementation [86] for further VAD processes in BASS.

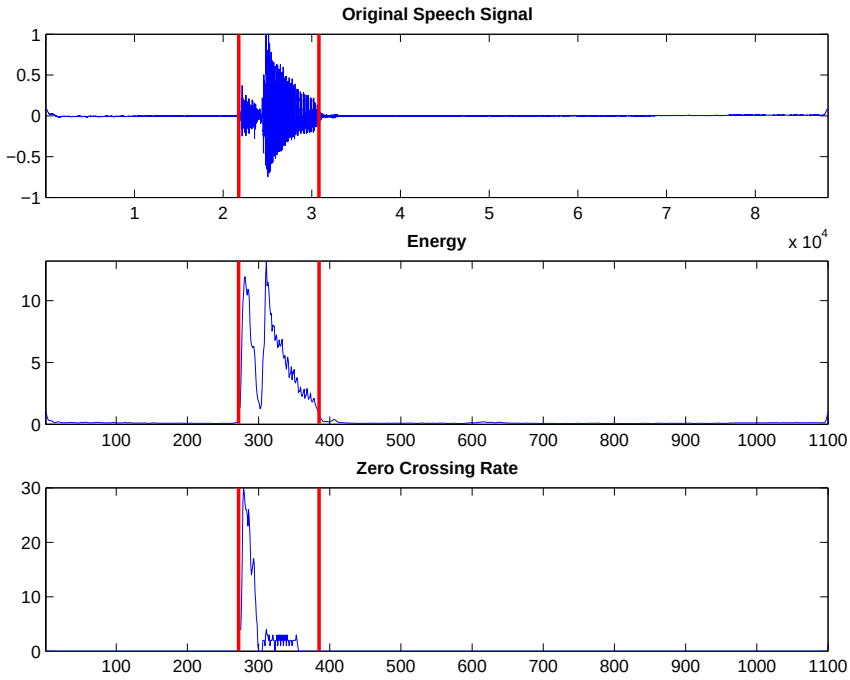


Figure 4.3: Voice activity detection based on energy and zero crossing rate for the word: "Trial"

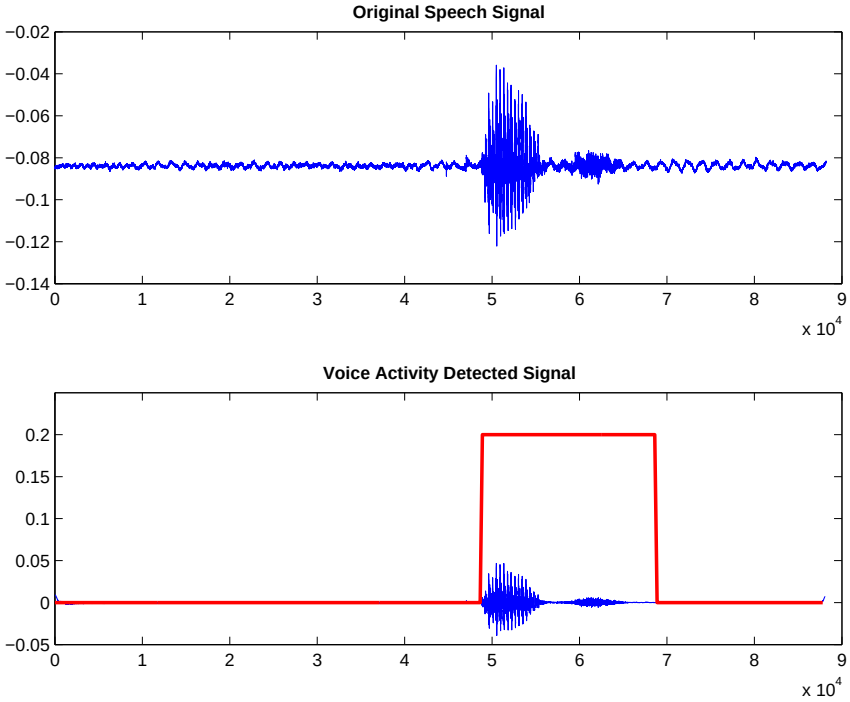


Figure 4.4: Voice activity detection based on the decision rule and the noise statistic estimation algorithm for the word: "Eight"

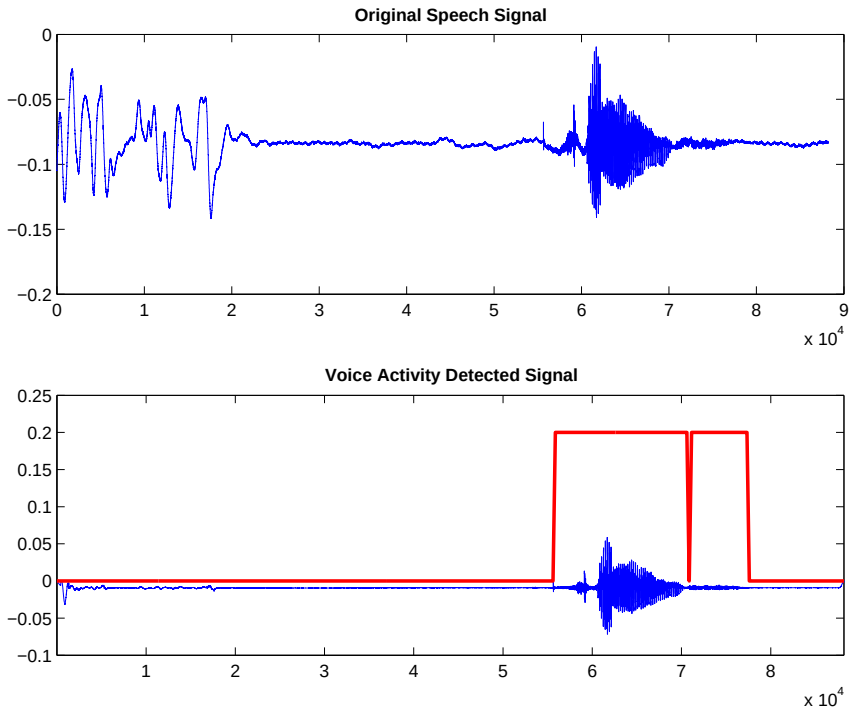


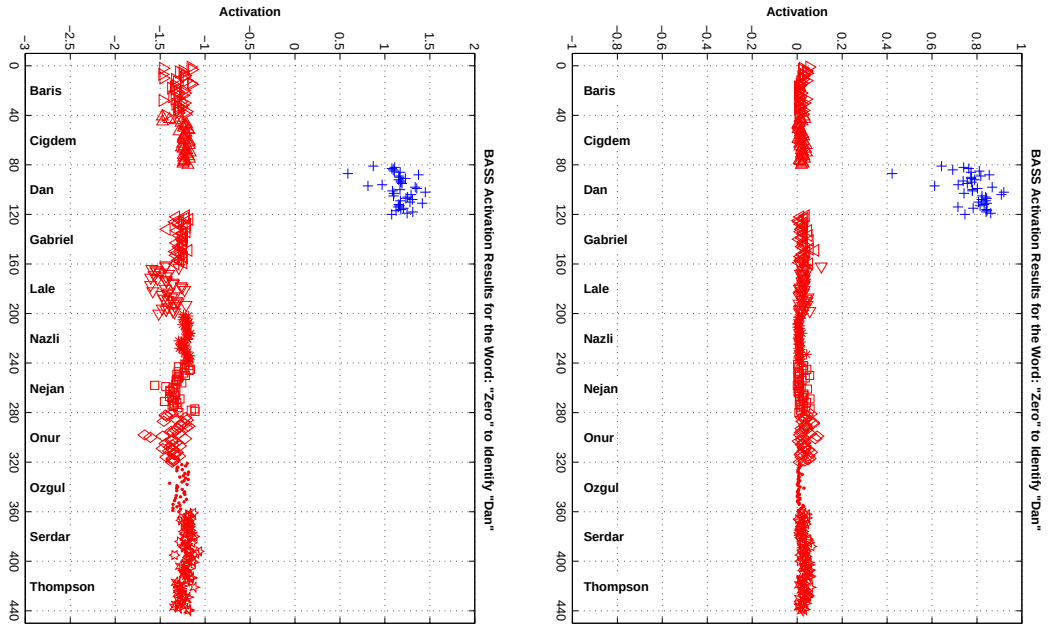
Figure 4.5: Voice activity detection based on minimum mean-squared error, a posteriori estimation of noise for the word: "Five"

4.2.4 Acoustic Feature Selection and Post-processing

We implemented a standard procedure [29] for extracting feature vectors from the voiced parts of the records. The procedure is as follows: First, we extracted 12 dimensional MFCCs from the pre-emphasized speech signal every 10 ms using a 25 ms Hamming window. We computed the mel cepstral vector using a simulated triangular filterbank on the discrete fourier transform spectrum. We performed bandlimiting by retaining only the filterbank outputs from the frequency range 300 Hz - 3140 Hz. We added the energy component of a frame as an additional feature. We also computed Δ cepstral coefficients computed over $\mp$ 2 frame span and appended to the cepstral vector. In addition, we added $\Delta\Delta$ coefficients to form a 39 dimensional vector including the energy, as explained in the previous chapter (Figure 3.13). Finally, we used mean and variance normalization that was applied to every individual frame for post processing of the acoustic feature vectors. Since we used only one microphone in the experiment, we did not include any algorithm to compensate channel effects.

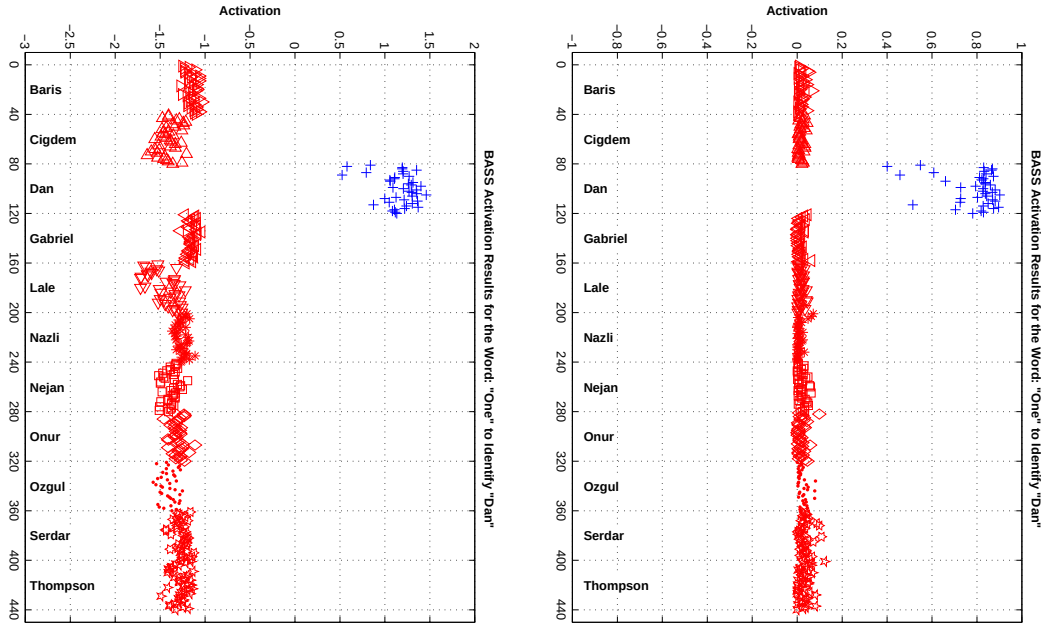
4.3 Training Classifiers

Our first task was to verify the client's identity (Dan Roth) against the others. For this purpose, we trained a large number of classifiers using the data set collected previously in the laboratory. We used the LIBSVM software package [31] for SVM training and the MATLAB Neural Network Toolbox for training of ANNs. The total number of training examples was 440 for each word (40 records from each speaker). In order to obtain the best performance, we tried various combinations of features: MFCC, MFCC + Energy, MFCC + Energy + Δ and MFCC + Energy + Δ + $\Delta\Delta$ to train the classifiers. We, however, only present the results that belong to the recognition models trained with 12 MFCC features since their performance were found to be adequate. We first used a validation set consisting of 10 records selected from 40 records of training data (i.e., 4-fold cross validation) to determine threshold values that will be used to decide when an unseen data is used to test the trained model. After determining the threshold values, we trained the classifiers using all the training examples. For each word, the training performances of SVM and ANN models are shown in Figures 4.6 - 4.18. (The figures on the left correspond to the results of SVM training while the ones on the right are for ANN training.)



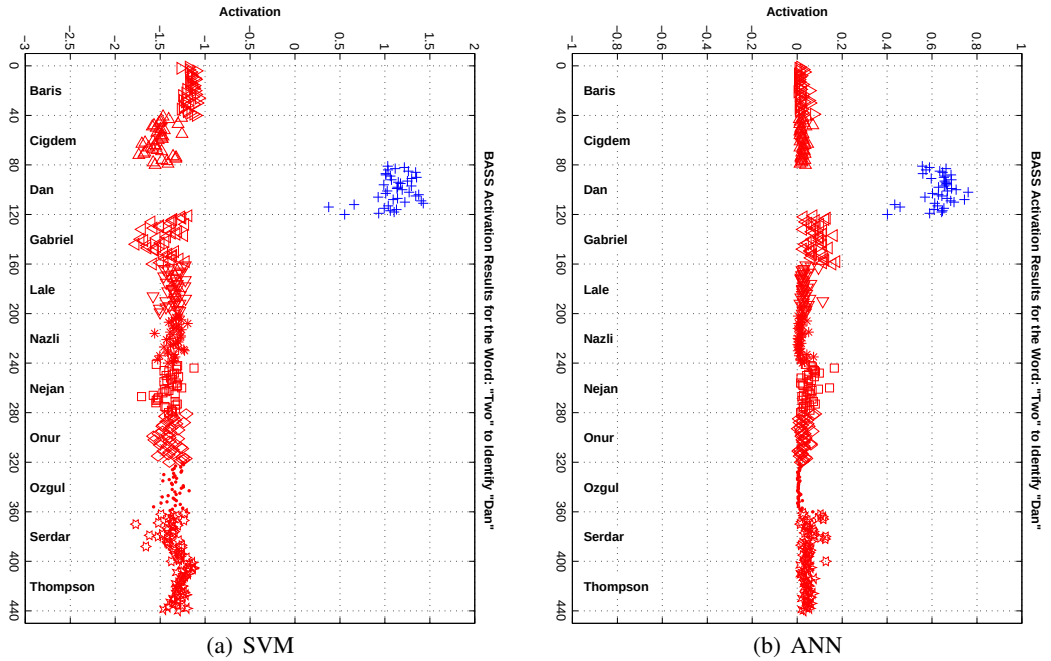
(a) SVM (b) ANN

Figure 4.6: Results for training of SVM and ANN classifiers for the word: "Zero"



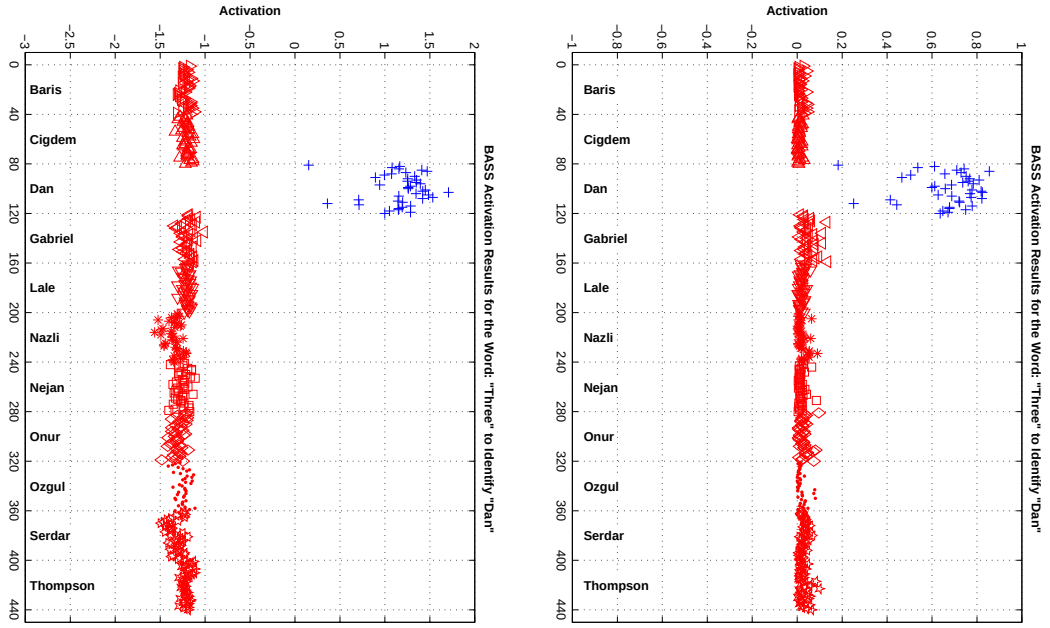
(a) SVM (b) ANN

Figure 4.7: Results for training of SVM and ANN classifiers for the word: “One”



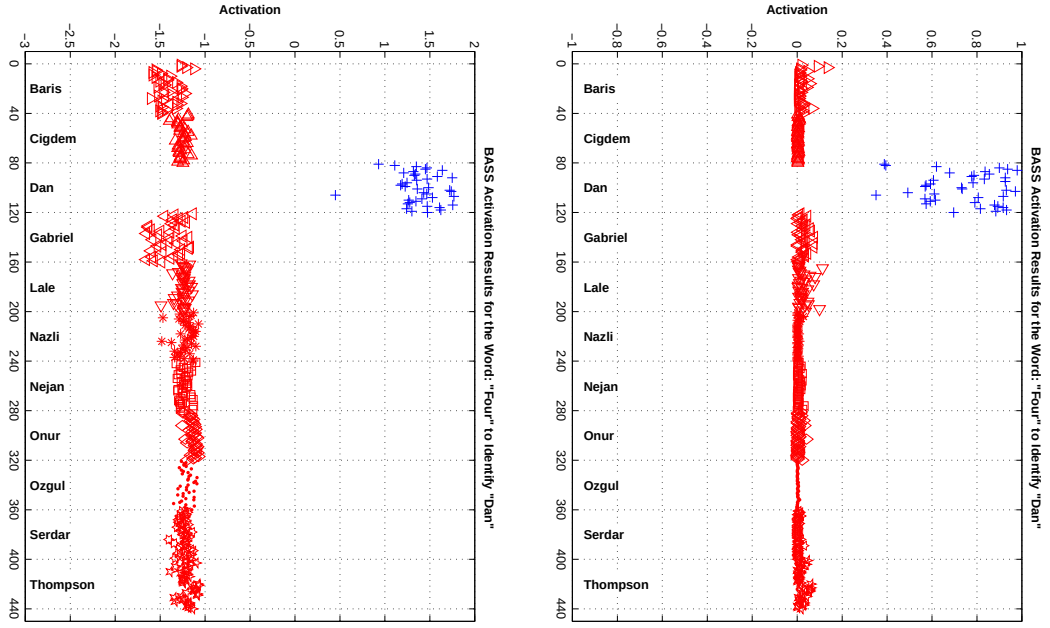
(a) SVM (b) ANN

Figure 4.8: Results for training of SVM and ANN classifiers for the word: “Two”



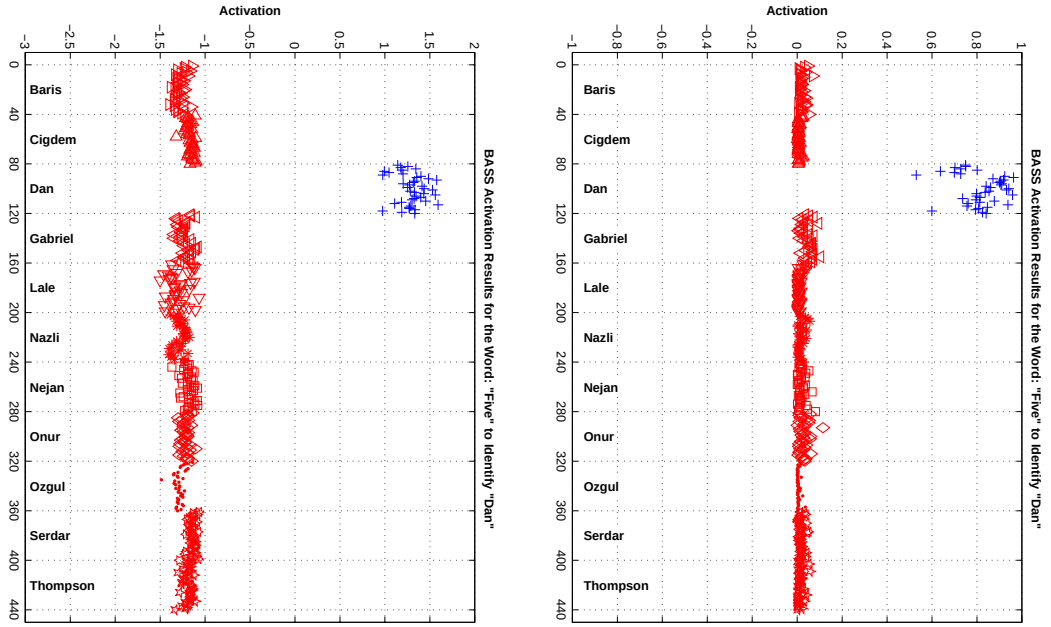
(a) SVM (b) ANN

Figure 4.9: Results for training of SVM and ANN classifiers for the word: “Three”



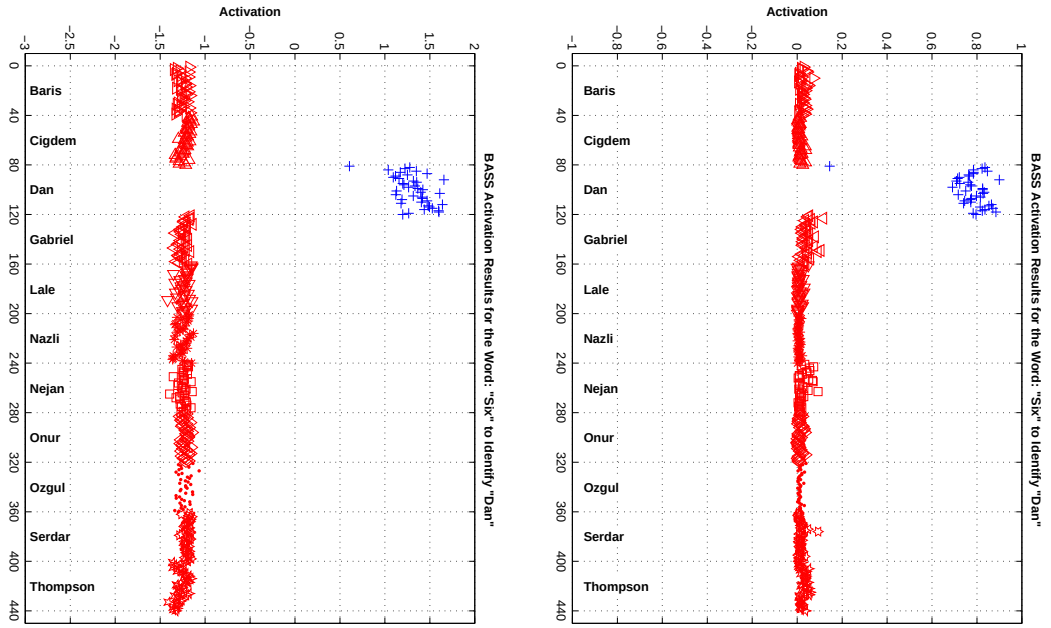
(a) SVM (b) ANN

Figure 4.10: Results for training of SVM and ANN classifiers for the word: “Four”



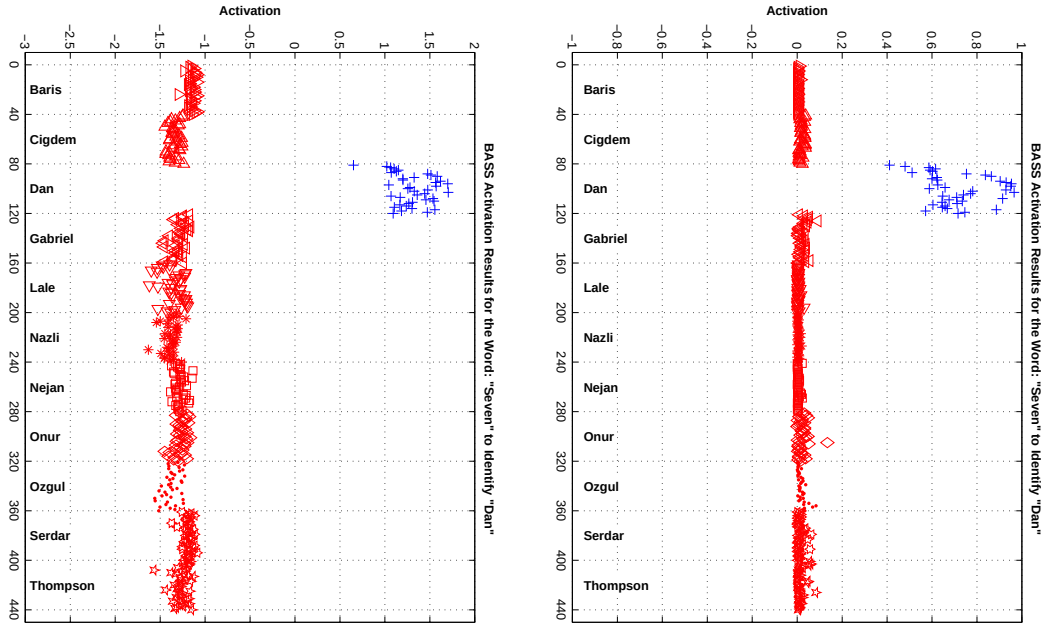
(a) SVM (b) ANN

Figure 4.11: Results for training of SVM and ANN classifiers for the word: “Five”



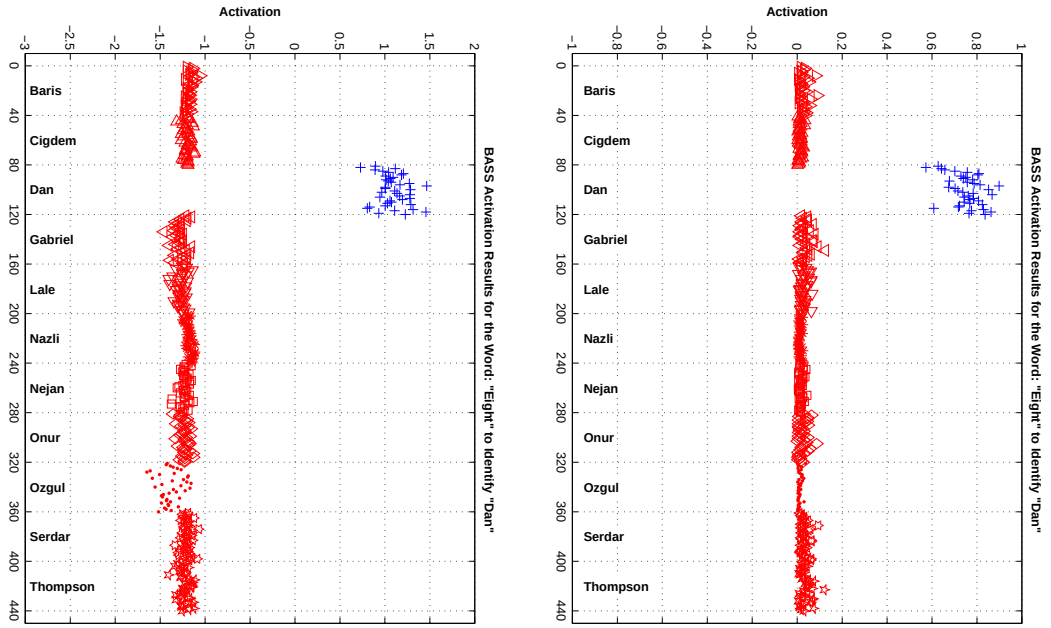
(a) SVM (b) ANN

Figure 4.12: Results for training of SVM and ANN classifiers for the word: “Six”



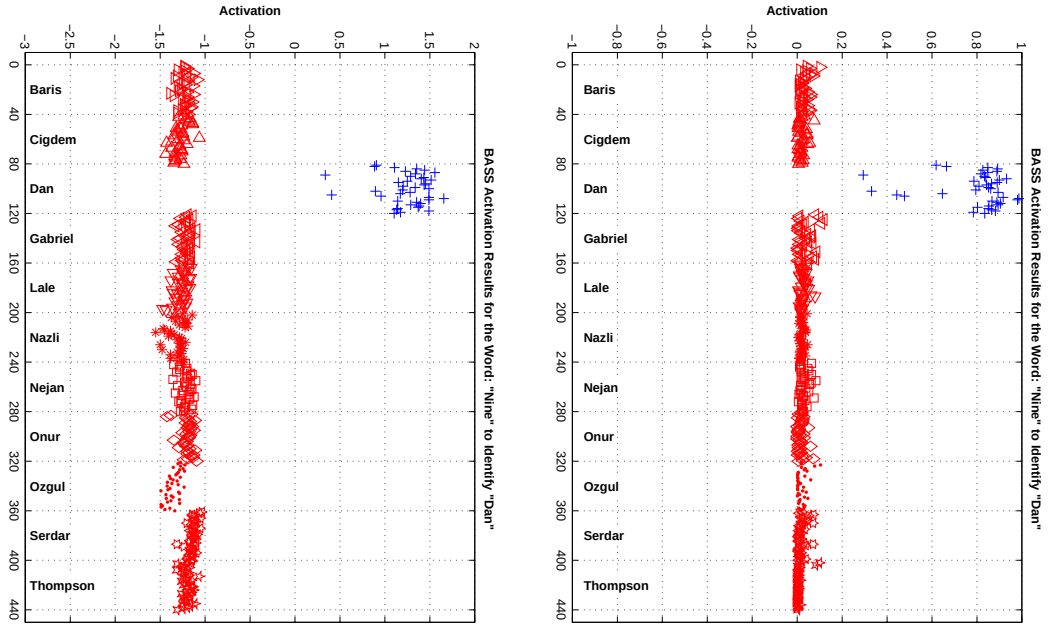
(a) SVM (b) ANN

Figure 4.13: Results for training of SVM and ANN classifiers for the word: “Seven”



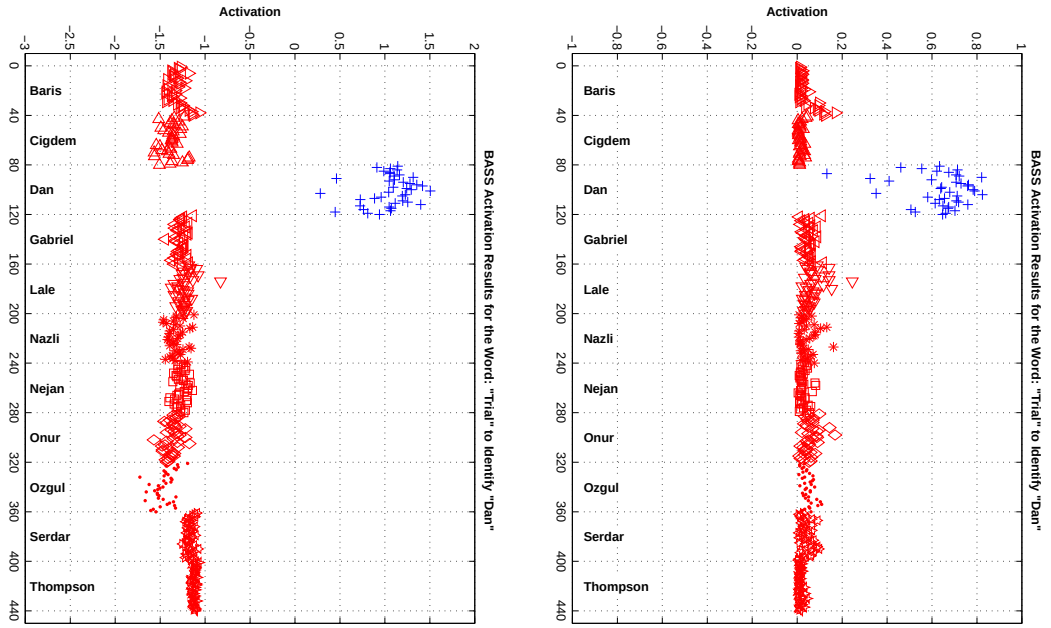
(a) SVM (b) ANN

Figure 4.14: Results for training of SVM and ANN classifiers for the word: “Eight”



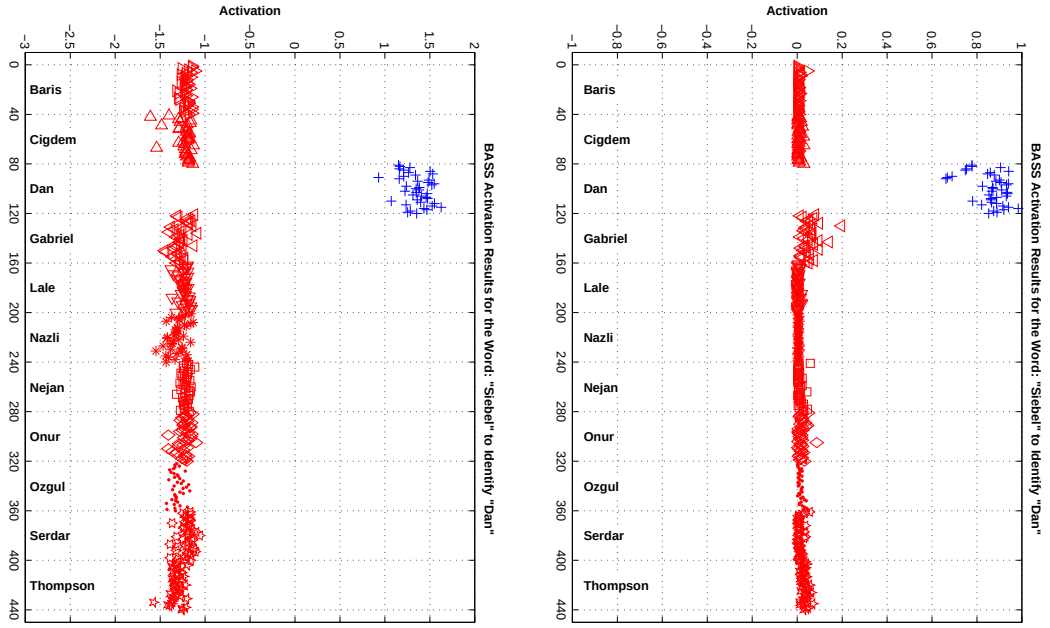
(a) SVM (b) ANN

Figure 4.15: Results for training of SVM and ANN classifiers for the word: “Nine”



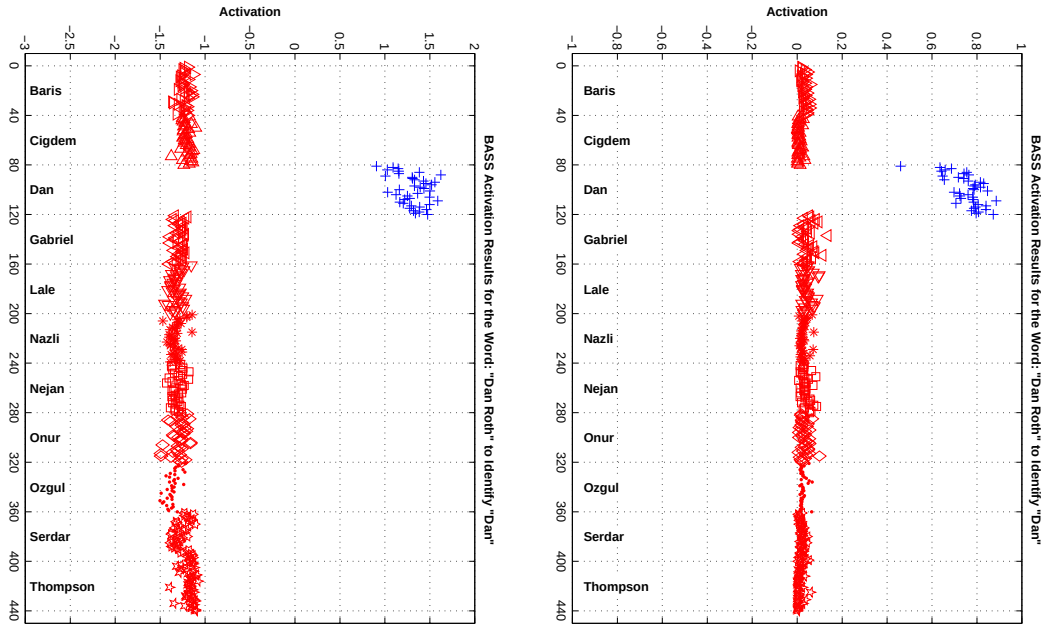
(a) SVM (b) ANN

Figure 4.16: Results for training of SVM and ANN classifiers for the word: “Trial”



(a) SVM (b) ANN

Figure 4.17: Results for training of SVM and ANN classifiers for the word: “Siebel”



(a) SVM (b) ANN

Figure 4.18: Results for training of SVM and ANN classifiers for the word: “Dan Roth”

4.4 Results

4.4.1 Speaker Verification

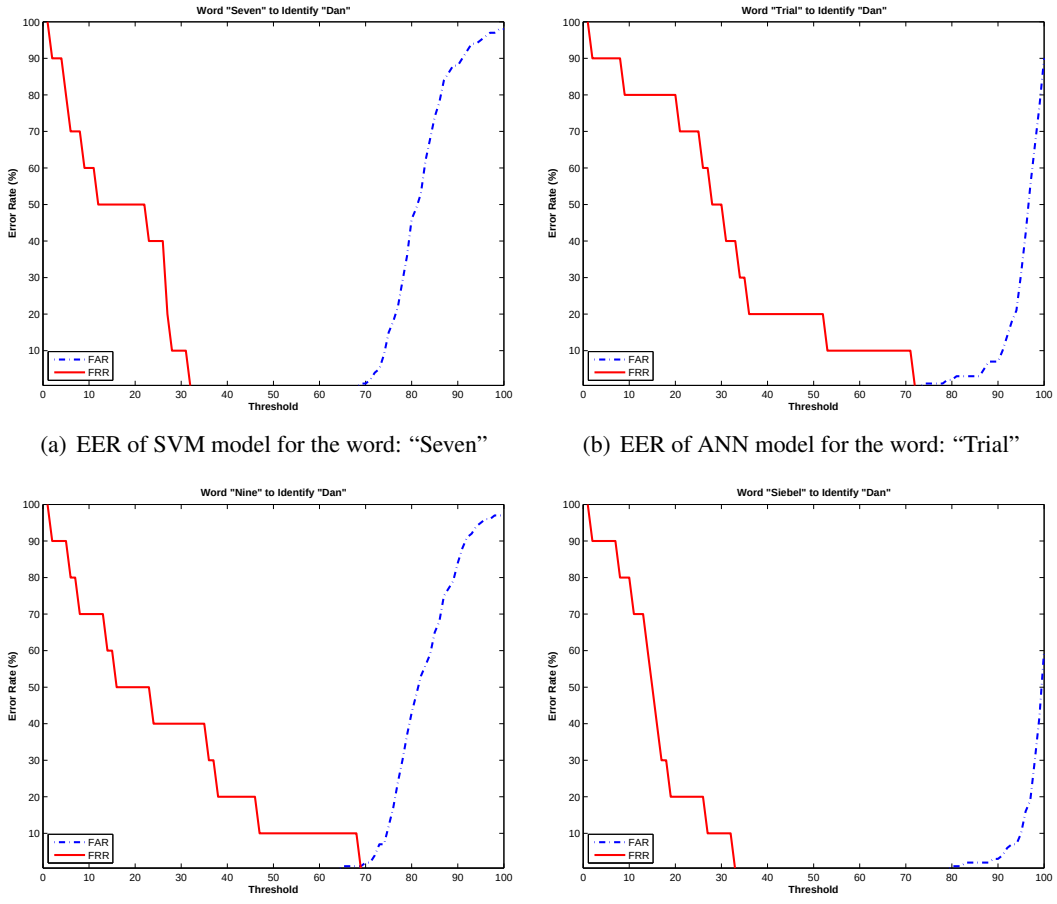
After successfully training both SVM and ANN classifiers, we verified their performances using a testing set. The steps for testing of speaker verification models were provided in Algorithm 2. We performed the testing using 10 records of each speaker, which were not seen during training. We then compared the utterance scores with the threshold values previously set for each word based on the training performance of SVM and ANN classifiers. We used the F measure (F), precision (p), recall (r) and accuracy (A) values to measure the performance of the developed models, which were calculated based on the utterances.

The results for verification of Dan Roth using both SVMs and ANNs are given in Table 4.3. The lowest value of the F measure we obtained using SVM models was 0.95 for the words "One" and "Nine". Similarly, we found the lowest values of the p and r values to be 0.91 (for the word "Nine") and 0.90 (for the word "One"), respectively. Meanwhile, the accuracy of SVM based models were generally higher than 99%. In the case of ANN models, we obtained perfect results: the accuracy value of each model was 100%. Likewise, the values of the F measure, p and r were all 1. Although the performance of both SVM and ANN models were satisfactory, we found that ANN based models were slightly better in validating the client's identity compared to SVM based models.

Table 4.3: BASS results of SVM and ANN classifiers for the verification of Dan Roth

	SVM				**ANN**			
	F	**p**	**r**	**A(%)**	**F**	**p**	**r**	**A(%)**
Zero	1	1	1	100	1	1	1	100
One	0.947	1	0.9	99.09	1	1	1	100
Two	1	1	1	100	1	1	1	100
Three	1	1	1	100	1	1	1	100
Four	1	1	1	100	1	1	1	100
Five	1	1	1	100	1	1	1	100
Six	1	1	1	100	1	1	1	100
Seven	1	1	1	100	1	1	1	100
Eight	1	1	1	100	1	1	1	100
Nine	0.952	0.909	1	99.09	1	1	1	100
Trial	1	1	1	100	1	1	1	100
Siebel	1	1	1	100	1	1	1	100
Dan Roth	1	1	1	100	1	1	1	100

We also calculated the false acceptance rate (FAR), false rejection rate (FRR) and therefore equal error rate (EER) of each model to measure the performance of models developed for each word. Our emphasis is to present the EER more since it is a widely accepted measure of the success for biometric authentication systems. In order to determine values of EER, we generated the "Error Rate vs. Threshold Value" plots for each word. Figure 4.19 shows four of these plots for various words and two classifiers. The maximum value of EER we calculated for SVM models was 5% (for the words "One" and "Nine"), while we obtained 0.0% EER of ANN models for every word. However, since BASS uses triplets, i.e., the combination of the individual words as passwords, the success rate of the whole system was increased to a level where EER was 0% for recognition models developed with both classifiers.



(a) EER of SVM model for the word: "Seven"

(b) EER of ANN model for the word: "Trial"

(c) EER of SVM model the word: "Nine"

(d) EER of the ANN model for the word: "Siebel"

Figure 4.19: Determination of equal error rate for SVM and ANN models

4.4.2 Speaker Identification

As an extension of the studies to validate Dan Roth's identity, further verification models needed to be built for the other speakers in the experiments. For each speaker, we assumed that s/he was a client whose identity needed to be verified while the others were impostors. As a result, we developed a total of 143 verification models (13 words x 11 speakers) for each classifier. We again tested the performance of these models using a testing set consisting of 10 records for each speaker, which were not seen during training. For both SVM and ANN classifiers, we present the result of testing for each word in Tables 4.4 - 4.13, using the previously defined statistical measures: "F, p, r and A".

In general, we obtained very good results for both SVM and ANN models. Amongst all SVM models, we found the lowest value for the F measure as 0.76 for the model developed to verify " Çiğdem" for the word "One". This specific model also produced the lowest values of p and r as 0.72 and 0.80, respectively. Interestingly, when the same data was used to train ANN models, the results proved to be very successful. The effect of lower success rates produced by certain models (especially those based on SVMs) was eliminated by using the combinations of the words having superior performances in the same password. As a result, we achieved an accuracy of 100%, which also causes EER values to decrease to 0%.

In the case of ANN models, we obtained the lowest value for the F measure as 0.82 for the model developed to verify "Gabriel" for the word "Siebel". This model also produced the lowest value of r to be 0.70. On the other hand, we obtained the lowest value of p to be 0.90 from the model developed to verify " Çiğdem" for the word "Trial". In general, ANN models proved to be better working compared to SVM ones. However, since we used triplets of different words each time when a user needs to be authenticated, the performances increased significantly for both classifiers.

Finally, the identification of the speakers was relatively easy since verification models developed for each speaker worked successfully. As explained in the previous chapter, we made decisions based on the model producing the maximum accumulated output for the given utterance for each classifier (see Equations 3.15 and 3.16 for the decision criteria of SVMs and ANNs, respectively). If BASS cannot decide about the identity of the speaker, then it displays an "Unknown Identity" message to the user.

Table 4.4: BASS results of SVM and ANN classifiers for the verification of Barış

	SVM				ANN			
	F	**p**	**r**	**A(%)**	**F**	**p**	**r**	**A(%)**
Zero	1	1	1	100	1	1	1	100
One	0.778	0.875	0.7	96.36	0.824	1	0.7	97.27
Two	1	1	1	100	1	1	1	100
Three	1	1	1	100	1	1	1	100
Four	1	1	1	100	1	1	1	100
Five	1	1	1	100	1	1	1	100
Six	1	1	1	100	1	1	1	100
Seven	1	1	1	100	1	1	1	100
Eight	1	1	1	100	1	1	1	100
Nine	1	1	1	100	1	1	1	100
Trial	1	1	1	100	1	1	1	100
Siebel	1	1	1	100	1	1	1	100
Dan Roth	1	1	1	100	1	1	1	100

Table 4.5: BASS results of SVM and ANN classifiers for the verification of Çiğdem

	SVM				ANN			
	F	**p**	**r**	**A(%)**	**F**	**p**	**r**	**A(%)**
Zero	1	1	1	100	1	1	1	100
One	0.762	0.727	0.8	95.46	1	1	1	100
Two	1	1	1	100	1	1	1	100
Three	1	1	1	100	1	1	1	100
Four	1	1	1	100	1	1	1	100
Five	0.952	0.909	1	99.09	1	1	1	100
Six	1	1	1	100	1	1	1	100
Seven	1	1	1	100	1	1	1	100
Eight	1	1	1	100	1	1	1	100
Nine	0.889	1	0.8	98.18	1	1	1	100
Trial	0.9	0.9	0.9	98.18	0.9	0.9	0.9	98.18
Siebel	1	1	1	100	1	1	1	100
Dan Roth	1	1	1	100	1	1	1	100

Table 4.6: BASS results of SVM and ANN classifiers for the verification of Gabriel

	SVM				ANN			
	F	**p**	**r**	**A(%)**	**F**	**p**	**r**	**A(%)**
Zero	1	1	1	100	1	1	1	100
One	0.947	1	0.9	99.09	1	1	1	100
Two	1	1	1	100	1	1	1	100
Three	1	1	1	100	1	1	1	100
Four	1	1	1	100	1	1	1	100
Five	1	1	1	100	1	1	1	100
Six	1	1	1	100	1	1	1	100
Seven	1	1	1	100	1	1	1	100
Eight	0.952	0.909	1	99.09	0.952	0.909	1	99.09
Nine	1	1	1	100	1	1	1	100
Trial	1	1	1	100	1	1	1	100
Siebel	0.824	1	0.7	97.27	0.824	1	0.700	97.27
Dan Roth	0.947	1	0.9	99.09	0.947	1	0.900	99.09

Table 4.7: BASS results of SVM and ANN classifiers for the verification of Lale

	SVM				ANN			
	F	**p**	**r**	**A(%)**	**F**	**p**	**r**	**A(%)**
Zero	1	1	1	100	1	1	1	100
One	0.778	0.875	0.7	96.36	1	1	1	100
Two	0.9	0.9	0.9	98.18	1	1	1	100
Three	0.952	0.909	1	99.09	1	1	1	100
Four	1	1	1	100	1	1	1	100
Five	1	1	1	100	1	1	1	100
Six	1	1	1	100	1	1	1	100
Seven	1	1	1	100	1	1	1	100
Eight	1	1	1	100	1	1	1	100
Nine	0.952	0.909	1	99.09	1	1	1	100
Trial	1	1	1	100	1	1	1	100
Siebel	1	1	1	100	1	1	1	100
Dan Roth	1	1	1	100	1	1	1	100

Table 4.8: BASS results of SVM and ANN classifiers for the verification of Nazlı

	SVM				ANN			
	F	**p**	**r**	**A(%)**	**F**	**p**	**r**	**A(%)**
Zero	1	1	1	100	1	1	1	100
One	1	1	1	100	1	1	1	100
Two	1	1	1	100	1	1	1	100
Three	1	1	1	100	1	1	1	100
Four	1	1	1	100	1	1	1	100
Five	1	1	1	100	1	1	1	100
Six	0.947	1	0.9	99.09	1	1	1	100
Seven	1	1	1	100	1	1	1	100
Eight	1	1	1	100	1	1	1	100
Nine	1	1	1	100	1	1	1	100
Trial	0.947	1	0.9	99.09	1	1	1	100
Siebel	1	1	1	100	1	1	1	100
Dan Roth	1	1	1	100	1	1	1	100

Table 4.9: BASS results of SVM and ANN classifiers for the verification of Nejan

	SVM				ANN			
	F	**p**	**r**	**A(%)**	**F**	**r**	**q**	**A(%)**
Zero	1	1	1	100	1	1	1	100
One	0.952	0.909	1	99.09	1	1	1	100
Two	1	1	1	100	1	1	1	100
Three	1	1	1	100	1	1	1	100
Four	1	1	1	100	1	1	1	100
Five	1	1	1	100	1	1	1	100
Six	1	1	1	100	1	1	1	100
Seven	1	1	1	100	1	1	1	100
Eight	1	1	1	100	1	1	1	100
Nine	1	1	1	100	1	1	1	100
Trial	1	1	1	100	1	1	1	100
Siebel	1	1	1	100	1	1	1	100
Dan Roth	1	1	1	100	1	1	1	100

Table 4.10: BASS results of SVM and ANN classifiers for the verification of Onur

	SVM				ANN			
	F	**p**	**r**	**A(%)**	**F**	**p**	**r**	**A(%)**
Zero	1	1	1	100	1	1	1	100
One	1	1	1	100	1	1	1	100
Two	1	1	1	100	1	1	1	100
Three	1	1	1	100	1	1	1	100
Four	1	1	1	100	1	1	1	100
Five	1	1	1	100	1	1	1	100
Six	1	1	1	100	1	1	1	100
Seven	1	1	1	100	1	1	1	100
Eight	1	1	1	100	1	1	1	100
Nine	1	1	1	100	1	1	1	100
Trial	1	1	1	100	1	1	1	100
Siebel	1	1	1	100	1	1	1	100
Dan Roth	0.952	0.909	1	99.09	1	1	1	100

Table 4.11: BASS results of SVM and ANN classifiers for the verification of Özgül

	SVM				ANN			
	F	**p**	**r**	**A(%)**	**F**	**p**	**r**	**A(%)**
Zero	1	1	1	100	1	1	1	100
One	1	1	1	100	1	1	1	100
Two	1	1	1	100	1	1	1	100
Three	1	1	1	100	1	1	1	100
Four	1	1	1	100	1	1	1	100
Five	1	1	1	100	1	1	1	100
Six	1	1	1	100	1	1	1	100
Seven	1	1	1	100	1	1	1	100
Eight	1	1	1	100	1	1	1	100
Nine	1	1	1	100	1	1	1	100
Trial	1	1	1	100	1	1	1	100
Siebel	1	1	1	100	1	1	1	100
Dan Roth	1	1	1	100	1	1	1	100

Table 4.12: BASS results of SVM and ANN classifiers for the verification of Serdar

	SVM				ANN			
	F	**p**	**r**	**A(%)**	**F**	**p**	**r**	**A(%)**
Zero	1	1	1	100	1	1	1	100
One	0.870	0.769	1	97.27	1	1	1	100
Two	0.889	1	0.8	98.18	1	1	1	100
Three	1	1	1	100	1	1	1	100
Four	1	1	1	100	1	1	1	100
Five	1	1	1	100	1	1	1	100
Six	1	1	1	100	1	1	1	100
Seven	1	1	1	100	1	1	1	100
Eight	1	1	1	100	1	1	1	100
Nine	1	1	1	100	1	1	1	100
Trial	1	1	1	100	1	1	1	100
Siebel	1	1	1	100	1	1	1	100
Dan Roth	0.947	1	0.9	99.09	1	1	1	100

Table 4.13: BASS results of SVM and ANN classifiers for the verification of Thompson

	SVM				ANN			
	F	**p**	**r**	**A(%)**	**F**	**p**	**r**	**A(%)**
Zero	1	1	1	100	1	1	1	100
One	1	1	1	100	0.952	0.909	1	99.09
Two	1	1	1	100	1	1	1	100
Three	1	1	1	100	1	1	1	100
Four	1	1	1	100	1	1	1	100
Five	1	1	1	100	1	1	1	100
Six	0.952	0.909	1	99.09	1	1	1	100
Seven	1	1	1	100	1	1	1	100
Eight	1	1	1	100	1	1	1	100
Nine	1	1	1	100	1	1	1	100
Trial	1	1	1	100	1	1	1	100
Siebel	1	1	1	100	1	1	1	100
Dan Roth	1	1	1	100	1	1	1	100

4.4.3 Challenges

BASS is designed to give users at least three attempts to verify their password phrase. Our experience showed that acceptance rates may improve on the second try since some records show that the speakers were confused about how to start saying the passwords. For example, Figure 4.20 shows the failed attempt of a speaker when a female speaker was asked to speak the password "Dan Roth". The problem here was that the speaker did not wait for the system to start recoding and the system missed the initial part of the record.

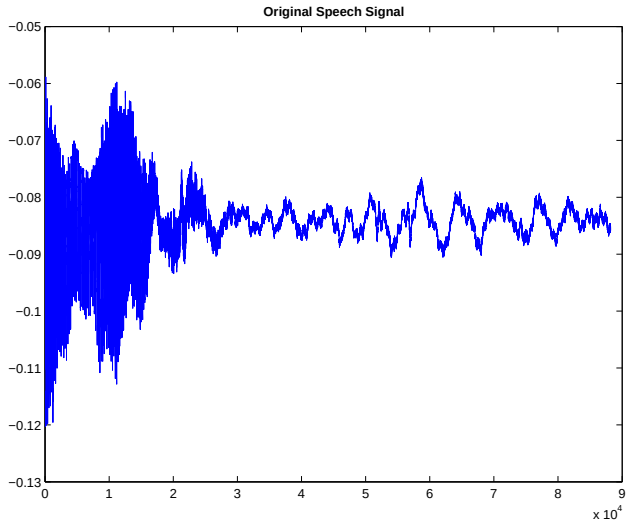


Figure 4.20: The unsuccessful capturing of a word "Dan Roth" spoken by Çiğdem in her first trial

Finally, since no biometric application can provide 100% accuracy, it is essential to develop contingent authentication strategies when designing the verification application. For example, if a person had difficulty gaining access to the system using the authentication failure, he or she is redirected to a staff member of the Siebel Center or allowed to swipe the ID card for a more traditional identity assessment.

Chapter 5

Summary, Conclusions and Future Work

5.1 Summary

In this study, we developed a complete Biometric Authentication System, which we named BASS, to be used in the Siebel Center for Computer Science building. BASS includes both hardware and software components. The hardware part mainly consists of a computer, microphone, infrared sensor, LCD screen and a connection output to a real door lock system. These components are designed to be almost independent such that the entire system can flexibly and efficiently be integrated into any infrastructure with only minor modifications.

The software component was mainly designed to perform speaker recognition. It is capable of doing both speaker verification and identification. The decision mechanism for authentication is based on machine learning models trained using voiceprint of different speakers. To develop a comprehensive database, experiments were performed in controlled laboratory conditions, where a representative subset of the Siebel Center staff was asked to speak passwords. A total of 11 speakers were involved and 13 words including numeric digits (0-9) were used during the experiments. Each speaker's voice was recorded and analyzed using digital signal processing tools. Front-end processing to speech signals was achieved by (1) pre-emphasizing the signal, (2) spectral subtraction and (3) voice activity detection. Several algorithms from the literature were implemented to obtain a clean speech signal. Based on the preprocessed signals, feature vectors were obtained using Mel-Frequency Cepstral Coefficients (MFCCs). The combinations of MFCCs together with delta (Δ) and delta-delta ($\Delta\Delta$) coefficients were considered. Finally, the feature vectors were further processed using mean and standard deviation normalization to be fed into the machine learning algorithms as inputs.

Two machine learning algorithms were used for classification purposes: Support Vector Machines (SVMs) and Artificial Neural Networks (ANNs). Binary classification was made using both techniques. Radial Basis Function kernel was chosen to train SVMs while backpropagation learning algorithm with adaptive learning rate and momentum was utilized in training of ANNs. "One vs. all" strategy was used in developing these models. The features used in training do not consider the order of voice segments, which was typical for ANNs but not for SVMs. The decision was made according to utterance scores that were calculated using the sum of frame-based posterior probabilities in ANNs and activation values in SVMs.

Among 11 speakers, Dan Roth's identity was first validated using both SVM and ANN models that were developed separately for every word spoken in the experiments. In addition, each participant's identity was found by developing separate models for all speakers and all words. As a result, a total of 286 models for both SVM and ANN classifiers (i.e., 11 speakers x 13 passwords = 143 models for SVM) were developed. A total of 40 out of 50 records were used in training. The remaining 10 were used to test the performance of each classifier. Thresholds that were used to make decisions were adaptive and speaker dependent. They were chosen based on data used to validate the performance of the training data (10 out 40 training records). To identify each speaker, the models developed for each speaker and for each word were tested, and the one producing the highest match score was selected.

5.2 Conclusions

The success in the implementation of our system was dependent on those of individual components, i.e., hardware and software. Assembly of BASS hardware was simple in the sense that finding the appropriate components and connecting them properly were straightforward. Hardware verification tests proved that BASS was quite responsive to inputs from the infrared sensor (i.e., the signal sent due to presence of speakers in front of the door). Selected combinations of words and the decision output of authentication system were also successfully displayed on LCD screen. The communication between hardware and software was also fast and reliable, and no malfunctioning was observed.

The speaker recognition part was the main contribution of this work. The results showed that Dan Roth's identity can be validated using both SVM or ANN based models. The performance of these models was measured using F measure (F), precision (p), recall (r) and accuracy (A) values, calculated based on the utterances. False acceptance rate (FAR), false rejection rate (FRR) and equal error rate (EER) were also reported for the whole system. Based on these values, the overall performance of BASS was quite satisfactory for both verification and identification. However, we concluded that ANN models were more precise and accurate compared to SVM models based on training of feature vectors without considering ordering of phoneme. As the number of speakers increase, complexity of the learning problem increases and therefore the recognition task will be more difficult. This may, for example, result in drastic increase in the number of support vectors for training data. The scalability of SVMs to large databases should then be questioned, and precautions should be taken.

Setting a low threshold for SVM activations and proper value to make decisions about utterance based posterior probabilities produced by ANN models played an important role in increasing the performance since adjusting of weights due to unbalanced data (i.e., penalizing negative examples) during training was not performed separately. This could have also been handled by properly choosing the training data such that the ratio of positive examples to negative ones was higher.

With regard to speech processing, we also concluded that since the verification system is text dependent, the detection of voiced regions in processing speech signal was the most important part of the front end processing. Similarly, the elimination of noise was another challenge. 12 MFCCs were found to produce relatively good results for both verification and identification tasks. There was no need to increase the number of features by adding Δ and $\Delta\Delta$ features commonly used in speech processing.

Finally, BASS is intended for the use of all academic personnel and staff members of the Siebel Center. The development of BASS has positive effects on the personnel who are responsible for maintaining the I-cards security and efficiency since it is going to eliminate the damage due to the heavy use of existing system. Since this model study reached very successful results, a much wider database needs to be developed to include voiceprints for all academic personnel and staff for facilitating entering their offices.

5.3 Future Work

Since the development of the whole system is composed of many tasks, there may be many improvements that can carry the currently designed system to the next level. We provide several items (not in the order of importance) below as future work when there is a need for a better authentication system.

- *Microphone Sensitivity:* For practical purposes, we used a unidirectional microphone in our experiments. Since the success rates of the classifiers are very much dependent on the robust and accurate formation of acoustic feature vectors, the acoustic characteristics of the microphone used in an authentication system need to be investigated. Performance of classifiers should be compared with those obtained using head-mounted or omnidirectional microphones.

- *Adaptivity:* The adaptation of the system to new users or with the existing ones considering different conditions such as illnesses, aging, etc should be taken into account in the design of BASS. Although there is currently a small module implemented in BASS for this purpose, it needs to be extended. For example, the modeling of impostors can be enriched using various databases.

- *Environmental Effects:* The noise cancelation was not in the scope of this work although it is a well studied subject in speech processing. Having carefully analyzed the speech files, we realized that the acoustics of the room should be understood well. In the end, the environment where the system is employed may be totally different than the experimental conditions in terms of acoustic quality. For this purpose, noise analysis should be performed even when speakers are not present to use the system.

- *Longer Duration of Speech:* The distinguishing characteristic of feature vectors for a typical voice file is dependent on the duration of recording. The characteristics of speakers can be defined more easily as duration of recording is increased. In this study, we used typical short duration samples for training; however, longer durations of recordings need to be considered in further development of BASS.

- *Phone Based Authentication:* Provided the tasks considered for future work are finished, the whole system can be carried onto *Voice over Internet Protocols (VOIP)*. In that case, the experiments should be performed using frequencies typical to phone lines, 8 kHz. This will enable individuals to use the developed system effectively by just speaking to a microphone in their cell phones, which will send the signals through VOIP protocol enabled internet browsers.

References

[1] Biometrics.gov. http://www.biometrics.gov, February, 2006.

[2] The Biometric Consortium. http://www.biometrics.org, February, 2006.

[3] The Biometrics Catalog. http://www.biometricscatalog.org, February, 2006.

[4] Hidden Markov Model Toolkit. http://htk.eng.cam.ac.uk, January, 2009.

[5] Dragon NaturallySpeaking 10 Standard, Speech Recognition Software. http://www.nuance.com, October, 2008.

[6] CMU Sphinx, Open Source Toolkit For Speech Recognition. http://cmusphinx.sourceforge.net, October, 2009.

[7] Microsoft Windows Speech Recognition Tool. http://www.microsoft.com/enable/products/windowsvista/speech.aspx, October, 2009.

[8] NIST Corpus. http://www.nist.gov/speech, October, 2009.

[9] ADAMI, A., MIHAESCU, R., REYNOLDS, D., AND GODFREY, J. Modeling prosodic dynamics for speaker recognition. In *Proceedings of Acoustics, Speech, and Signal Processing, ICASSP-03* (2003), vol. 4, pp. 788–791.

[10] ALPAYDIN, E. *Introduction to Machine Learning*. The MIT Press, 2004.

[11] ANDRAS, P. The equivalence of support vector machine and regularization neural networks. *Neural Processing Letters 15*, 2 (2002), 97–104.

[12] BENNANI, Y., AND GALLINARI, P. On the use of TDNN-extracted features information in talker identification. In *Proceedings of Acoustics, Speech, and Signal Processing, ICASSP-91* (1991), pp. 385–388.

[13] BENNANI, Y., AND GALLINARI, P. Neural networks for discrimination and modelization of speakers. *Speech Communication 17*, 1-2 (1995), 159–175.

[14] BEROUTI, M., SCHWARTZ, R., AND MAKHOUL, J. Enhancement of speech corrupted by acoustic noise. In *Proceedings of Acoustics, Speech, and Signal Processing, ICASSP-79* (1979), vol. 4, pp. 208–211.

[15] BISHOP, C. *Neural Networks for Pattern Recognition*. Oxford University Press, 2005.

[16] BOERSMA, P. Praat, a system for doing phonetics by computer. *Glot International 5*, 9/10 (2001), 341–345.

[17] BOLL, S. A spectral subtraction algorithm for suppression of acoustic noise in speech. In *Proceedings of Acoustics, Speech, and Signal Processing, ICASSP-79* (1979), vol. 4, pp. 200–203.

[18] BORYS, S. E. An SVM Front-end Landmark Speech Recognition System. Master's thesis, University of Illinois at Urbana-Champaign, 2008.

[19] BRIDLE, J. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing: Algorithms, Architectures and Applications* (1989), pp. 227–236.

[20] BROOKES, M. Voicebox: Speech processing toolbox for matlab. `http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html`, 2000.

[21] BURGES, C. J. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery 2* (1998), 121–167.

[22] BURKE, D. *Speech Processing for IP Networks: Media Resource Control Protocol (MRCP)*. Wiley, 2007.

[23] BYUN, H., AND LEE, S. Applications of support vector machines for pattern recognition: a survey. In *Pattern Recognition with Support Vector Machines. Proceedings of First International Workshop, SVM 2002. (Lecture Notes in Computer Science Vol.2388)* (2002), Springer Berlin / Heidelberg, pp. 213–236.

[24] CAMPBELL, J. Testing with the YOHO CD-ROM voice verification corpus. In *Proceedings of Acoustics, Speech, and Signal Processing, ICASSP-95* (1995), vol. 1, pp. 341–344.

[25] CAMPBELL, J.P., J. Speaker recognition: a tutorial. *Proceedings of the IEEE 85*, 9 (1997), 1437–1462.

[26] CAMPBELL, W. Generalized linear discriminant sequence kernels for speaker recognition. In *Proceedings of Acoustics, Speech, and Signal Processing, ICASSP-02* (2002), vol. 1, pp. 161–164.

[27] CAMPBELL, W., CAMPBELL, J., REYNOLDS, D., JONES, D., AND LEEK, T. High-level speaker verification with support vector machines. In *Proceedings of Acoustics, Speech, and Signal Processing, ICASSP-04* (2004), pp. 73–76.

[28] CAMPBELL, W., CAMPBELL, J., REYNOLDS, D., JONES, D., AND LEEK, T. Phonetic speaker recognition with support vector machines. *Advances in Neural Information Processing Systems 16* (2004), 1377–1384.

[29] CAMPBELL, W., CAMPBELL, J., REYNOLDS, D., SINGER, E., AND TORRES-CARRASQUILLO, P. Support vector machines for speaker and language recognition. *Computer Speech & Language 20*, 2-3 (2006), 210–229.

[30] CAMPBELL, W., STURIM, D., REYNOLDS, D., AND SOLOMONOFF, A. SVM based speaker verification using a GMM supervector kernel and NAP variability compensation. In *Proceedings of Acoustics, Speech, and Signal Processing, ICASSP-06* (2006), vol. 1, pp. 97–100.

[31] CHANG, C.-C., AND LIN, C.-J. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[32] CHEN, S., AND LUO, Y. Speaker verification using MFCC and support vector machine. In *Proceedings of the International MultiConference of Engineers and Computer Scientists* (2009), vol. 1.

[33] CIERI, C., MILLER, D., AND WALKER, K. The Fisher corpus: a resource for the next generations of speech-to-text. In *Fourth International Conference on Language Resources and Evaluation* (2004).

[34] COURANT, R., AND HILBERT, D. *Methods of Mathematical Physics*, vol. 1. Wiley, 1953.

[35] DAVID, E., AND SELFRIDGE, O. Eyes and ears for computers. vol. 50, pp. 1093–1101.

[36] DAVIS, K. H., BIDDULPH, R., AND BALASHEK, S. Automatic recognition of spoken digits. *The Journal of the Acoustical Society of America 24*, 6 (1952), 637–642.

[37] DAVIS, S., AND MERMELSTEIN, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing 28*, 4 (1980), 357–366.

[38] DELLER, JR., J. R. P. J. G., AND HANSEN, J. H. *Discrete Time Processing of Speech Signals*. Prentice Hall PTR, 1993.

[39] DENES, P. The design and operation of the mechanical speech recognizer at university college, london. *Journal of the British Institution of Radio Engineers 19* (1959), 219–229.

[40] DODDINGTON, G., ET AL. Speaker recognition based on idiolectal differences between speakers. In *Proceedings of Seventh European Conference on Speech Communication and Technology* (2001).

[41] DUDA, R., HART, P., AND STORK, D. *Pattern Classification*. Wiley, 2001.

[42] FARRELL, K., MAMMONE, R., AND ASSALEH, K. Speaker recognition using neural networks and conventional classifiers. *IEEE Transactions on Acoustics, Speech and Signal Processing 2*, 1 (1994), 194–205.

[43] FERRAS, M., LEUNG, C., BARRAS, C., AND GAUVAIN, J. Constrained mllr for speaker recognition. In *Proceedings of Acoustics, Speech and Signal Processing, ICASSP-07* (2007), vol. 7, pp. 53–56.

[44] FERRAS, M., LEUNG, C., BARRAS, C., AND GAUVAIN, J. MLLR techniques for speaker recognition. In *Proceedings of IEEE Speaker Odyssey* (2008).

[45] FINE, S., NAVRATIL, J., AND GOPINATH, R. A hybrid GMM/SVM approach to speaker identification. In *Proceedings of Acoustics, Speech and Signal Processing, ICASSP-01* (2001), vol. 1.

[46] FRY, D. Theoretical aspects of mechanical speech recognition. *Journal of the British Institution of Radio Engineers 19*, 4 (1959), 211–218.

[47] GANCHEV, T., FAKOTAKIS, N., AND KOKKINAKIS, G. Text-independent speaker verification based on probabilistic neural networks. In *Proceedings of the Acoustics* (2002), pp. 159–166.

[48] GIROSI, F. An equivalence between sparse approximation and support vector machines. *Neural Computation 10* (1997), 1455–1480.

[49] GISH, H. A probabilistic approach to the understanding and training ofneural network classifiers. In *Proceedings of Acoustics, Speech and Signal Processing, ICASSP-90* (1990), pp. 1361–1364.

[50] GISH, H., AND SCHMIDT, M. Text-independent speaker identification. *IEEE Signal Processing Magazine 11*, 4 (1994), 18–32.

[51] GUTSCHOVEN, B., AND VERLINDE, P. Multi-modal identity verification using support vector machines (svm). In *Proceedings of the International Conference on Information Fusion, FUSION* (2000), IEEE Press, pp. 3–8.

[52] HATCH, A., STOLCKE, A., AND PESKIN, B. Combining feature sets with support vector machines: application to speaker recognition. In *IEEE Workshop on Automatic Speech Recognition and Understanding* (2005), pp. 75–79.

[53] HERMANSKY, H. Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustical Society of America 87*, 4 (1990), 1738–1752.

[54] HIRSCH, H., AND EHRLICHER, C. Noise estimation techniques for robust speech recognition. In *Proceedings of Acoustics, Speech and Signal Processing, ICASSP-95* (1995), vol. 1, pp. 153–156.

[55] HOSSAIN, M., AHMED, B., AND ASRAFI, M. A real time speaker identification using artificial neural network. In *Proceedings of 10th International Conference on Computer and Information Technology* (2007), pp. 1–5.

[56] HUANG, J.-H., SU, S.-L., AND CHEN, J.-H. Design and performance analysis for data transmission in gsm/gprs system with voice activity detection. *IEEE Transactions on Vehicular Technology 51*, 4 (2002), 648–656.

[57] JAIN, A., BOLLE, R., AND PANKANTI, S. *Biometrics: Personal Identification in Networked Society*. Kluwer Academic Publishers, 1999.

[58] JO, Q.-H., CHANG, J.-H., SHIN, J., AND KIM, N. Statistical model-based voice activity detection using support vector machine. *IET Signal Processing 3*, 3 (2009), 205–210.

[59] JURAFSKY, D., AND MARTIN, J. *Speech and Language Processing : An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 2nd ed. 2009.

[60] KAJAREKAR, S. Four weightings and a fusion: a cepstral-svm system for speaker recognition. In *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding* (2005), pp. 17–22.

[61] KARAM, Z., AND CAMPBELL, W. A new kernel for SVM MLLR based speaker recognition. In *Proceedings of Interspeech* (2007), pp. 290–293.

[62] KATZ, M., KRUGER, S., SCHAFFONER, M., ANDELIC, E., AND WENDEMUTH, A. Speaker identification and verification using support vector machines and sparse kernel logistic regression. In *Advances in Machine Vision, Image Processing, and Pattern Analysis* (2006), pp. 176–184.

[63] KEERTHI, S., SHEVADE, S., BHATTACHARYYA, C., AND MURTHY, K. Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation 13*, 3 (2001), 637–649.

[64] KRUGER, E., AND STRUBE, H. Linear prediction on a warped frequency scale (speech processing). *IEEE Transactions on Acoustics, Speech and Signal Processing 36*, 9 (1988), 1529–1531.

[65] LAINE, U., KARJALAINEN, M., AND ALTOSAAR, T. Warped linear prediction (wlp) in speech and audio processing. In *Proceedings of Acoustics, Speech and Signal Processing, ICASSP-94* (1994), vol. 3, pp. 349–352.

[66] LEE, B., AND HASEGAWA-JOHNSON, M. Minimum mean-squared error a posteriori estimation of high variance vehicular noise. *Biennial on DSP for In-Vehicle and Mobile Systems* (2007).

[67] MAKHOUL, J. Linear prediction: a tutorial review. *Proceedings of the IEEE 63*, 4 (1975), 561–580.

[68] MAKHOUL, J., SCHWARTZ, R., AND EL-JAROUDI, A. Classification capabilities of two-layer neural nets. In *Proceedings of Acoustics, Speech and Signal Processing, ICASSP-89* (1989), pp. 635–638.

[69] MARTIN, R. Noise power spectral density estimation based on optimal smoothing and minimum statistics. *IEEE Transactions on Speech and Audio Processing 9*, 5 (2001), 504–512.

[70] MATSUI, T., AND FURUI, S. Concatenated phoneme models for text-variable speaker recognition. In *Proceedings of Acoustics, Speech and Signal Processing, ICASSP-93* (1993), vol. 2.

[71] MATTERA, D., PALMIERI, F., AND HAYKIN, S. Simple and robust methods for support vector expansions. *IEEE Transactions on Neural Networks 10*, 5 (1999), 1038–1047.

[72] MIRCHANDANI, G., CAO, W., AND BOSWORTH, B. Efficient implementation of neural nets using an optimal relationship between number of patterns, input dimension and hidden nodes. In *Proceedings of Acoustics, Speech and Signal Processing, ICASSP-89* (1989), pp. 2521–2523.

[73] MORGAN, D. P., AND SCOFIELD, C. L. *Neural Networks and Speech Processing*. Kluwer Academic Publishers, 1991.

[74] NAVRATIL, J., JIN, Q., ANDREWS, W., AND CAMPBELL, J. Phonetic speaker recognition using maximum-likelihood binary-decision tree models. In *Proceedings of Acoustics, Speech and Signal Processing, ICASSP-03* (2003), vol. 4, pp. 796–799.

[75] NEWSON, P., AND HEATH, M. The capacity of a spread spectrum cdma system for cellular mobile radio with consideration of system imperfections. *IEEE Journal on Selected Areas in Communications 12*, 4 (1994), 673–684.

[76] NUANCE. *Application Developer's Guide*, April, 2009.

[77] OGLESBY, J., AND MASON, J. Optimisation of neural models for speaker identification. In *Proceedings of Acoustics, Speech and Signal Processing, ICASSP-90* (1990), vol. 1, pp. 261–264.

[78] OGLESBY, J., AND MASON, J. S. Radial basis function networks for speaker recognition. In *Proceedings of Acoustics, Speech and Signal Processing, ICASSP-91* (1991), pp. 393–396.

[79] OSUNA, E., FREUND, R., AND GIROSI, F. Training support vector machines: an application to face detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (1997), pp. 130–136.

[80] PARSONS, T. *Voice and Speech Processing*. McGraw-Hill College, 1987.

[81] PESKIN, B., NAVRATIL, J., ABRAMSON, J., JONES, D., KLUSACEK, D., REYNOLDS, D., AND XIANG, B. Using prosodic and conversational features for high-performance speaker recognition: report from jhu ws'02. In *Proceedings of Acoustics, Speech and Signal Processing, ICASSP-03* (2003), vol. 4, pp. 792–795.

[82] PLATT, J. Sequential minimal optimization: a fast algorithm for training support vector machines. *Advances in Kernel Methods-Support Vector Learning 208* (1999).

[83] QUATIERI, T. *Discrete-Time Speech Signal Processing: Principles and Practive*. Prentice Hall, 2002.

[84] RABINER, L., AND JUANG, B. Introduction to hidden markov models. *IEEE ASSP Magazine. 3*, 1 (1986), 4–16.

[85] RABINER, L., AND JUANG, B. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[86] RABINER, L., AND SAMBUR, M. An algorithm for determining the endpoints of isolated utterances. *The Bell System Technical Journal 54*, 2 (1975), 297–315.

[87] ROSS, A., NANDAKUMAR, K., AND JAIN, A. *Handbook of Multibiometrics*. Springer, 2006.

[88] RUDASI, L., AND ZAHORIAN, S. A. Text-independent talker identification with neural networks. In *Proceedings of Acoustics, Speech and Signal Processing, ICASSP-91* (1991), pp. 389–392.

[89] SCHMIDT, M., AND GISH, H. Speaker identification via support vector classifiers. In *Proceedings of Acoustics, Speech and Signal Processing, ICASSP-96* (1996), pp. 105–108.

[90] SCHÖLKOPF, B., BURGES, C., AND SMOLA, A. *Advances in Kernel Methods: Support Vector Learning*. The MIT press, 1998.

[91] SMOLA, A., AND SCHÖLKOPF, B. A tutorial on support vector regression. *Statistics and Computing 14*, 3 (2004), 199–222.

[92] SOHN, J., KIM, N. S., AND SUNG, W. A statistical model-based voice activity detection. *IEEE Signal Processing Letters 6*, 1 (1999), 1–3.

[93] SOHN, J., AND SUNG, W. A voice activity detector employing soft decision based noise spectrum adaptation. In *Proceedings of Acoustics, Speech and Signal Processing, ICASSP-98* (1998), vol. 1, pp. 365–368.

[94] STOLCKE, A., FERRER, L., KAJAREKAR, S., SHRIBERG, E., AND VENKATARAMAN, A. MLLR transforms as features in speaker recognition. In *Ninth European Conference on Speech Communication and Technology* (2005), ISCA.

[95] STURIM, D. E., CAMPBELL, W. M., AND REYNOLDS, D. A. Classification methods for speaker recognition. In *Speaker Classification I: Fundamentals, Features, and Methods* (2007), Springer Berlin / Heidelberg, pp. 278–297.

[96] TIERNEY, J. A study of lpc analysis of speech in additive noise. *IEEE Transactions on Acoustics, Speech and Signal Processing 28*, 4 (1980), 389–397.

[97] VAPNIK, V. *The nature of Statistical Learning Theory*. Springer Verlag, 2000.

[98] VASEGHI, S. *Advanced Digital Signal Processing and Noise Reduction*. Wiley, 2005.

[99] WAN, V. *Speaker verification using support vector machines*. PhD thesis, University of Sheffield, 2003.

[100] WAN, V. *Building sequence kernels for speaker verification and word recognition*. IGI Publishing, 2007.

[101] WAN, V., AND CAMPBELL, W. M. Support vector machines for speaker verification and identification. In *Proceeding of Neural Networks for Signal Processing* (2000), pp. 775–784.

[102] WAN, V., AND RENALS, S. SVMSVM: support vector machine speaker verification methodology. In *Proceedings of the International Conference on Acoustics Speech and Signal Processing* (2003), vol. 2, pp. 221–224.

[103] WAN, V., AND RENALS, S. Speaker verification using sequence discriminant support vector machines. *IEEE Transactions on Speech and Audio Processing 13*, 2 (2005), 203–210.