COMPARATIVE STUDY ON MALAY CHILDREN VOWEL RECOGNITION USING MULTI-LAYER PERCEPTRON AND RECURRENT NEURAL NETWORKS

AFSHAN KORDI

RESEARCH PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF ENGINEERING (BIOMEDICAL)

FACULTY OF ENGINEERING
UNIVERSITY OF MALAYA
KUALA LUMPUR

2012

UNIVERSITI MALAYA
ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: *Afshan Kordi*

Registration/Matric No: *KGL090010*

Name of Degree: *Master of Biomedical Engineering*

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"):

Comparative Study on Malay Children Vowel Recognition using Multi-Layer Perceptron and Recurrent Neural Networks

Field of Study: *Signal Processing*

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date

Subscribed and solemnly declared before,

Witness's Signature Date

Name:

Designation:

ABSTRACT

Speech recognition has become popular during recent decades due to its widespread applications such as telephone systems, health care domain, data entry, speech to text processing, biometric systems, training air traffic controllers and so on. Among the technologies that have been investigated in acoustic modeling of speech, Artificial Neural Networks (ANN) have received interest from many researchers as they have shown good results in pattern recognition specially in classification. Despite of noteworthy progress in speech classification using neural networks, some unresolved issues still are remained in utilizing and performing the neural networks. Particularly less effort has been done on the speech of children which is more dynamic. There are numerous neural network architectures introduced by scientists that the most common sufficient for speech recognition include: Multi-Layer Perceptron (MLP) and Recurrent Neural Network (RNN). The purpose of this study is to compare the performance and recognition rate of these two types of neural networks in terms of signal length and number of hidden neurons for sustained Malay vowel among Malay children. Linear Predictive Coding (LPC) is used as a feature extractor to convert the speech signal into parametric coefficients. The Neural Network Toolbox™ (*nntool*) in Matlab® is used to classify the six Malay vowels (/a/, /e/ /ə/, /i/, /o/ and /u/) according to the *3-fold cross validation* technique in different signal lengths with different number of hidden neurons. Experiments were done to compare the performance of the neural networks using single frame and multiple frame approach as well. The results show that longer signal lengths perform better than those in short signal lengths. The findings indicate that MLP and RNN reached a recognition rate of 83.79% and 83.10% respectively. Vowel /i/ got the highest recognition rate in both methods.

ABSTRAK

Pengenalpasti suara telah menjadi subjek yang hangat sejak dekad yang lalu disebabkan perluasan penggunaan aplikasi seperti telefon, alat kesihatan, kemasukan data, alat suara ke tulisan, sistem biometric, latihan pengurusan trafik udara dan sebagainya. Antara teknologi yang telah dikaji dalam bidang akoustik, Artificial Neural Networks (ANN) telah mendapat minat daripada ramai pengaji disebabkan ia menunjukan prestasi yang bagus dalam bidang klasifikasi. Walaupun keberkesanan sistem Neural Networks, namun banyak lagi masalah dalam bidang klasifikasi masih tidak dapat diselesaikan. Jarang terdapat kajian yang dilakukan ke atas suara kanak-kanak yang lebih dinamik. Banyak jenis Neural Networks telah diperkenalkan oleh saintis dan antara yang paling berkesan ialah Multi-Layer Perceptron (MLP) and Recurrent Neural Network (RNN). Ini disebabkan sistem tersebut dapat mengambil kira faktor dalam kepanjangan maklumat dan nombor neurons dalam Neural Networks tersebut dalam aplikasi suara kanak-kanak Melayu. Linear Predictive Coding (LPC) adalah salah satu cara untuk mengira informasi yang terlindung dalam maklumat dari suara tersebut. Neural Network Toolbox™ (*nntool*) dalam Matlab® telah digunakan untuk mengklasifikasikan enam vocal kanak-kanak (/a/, /e/ /ə/, /i/, /o/ and /u/) dengan membahagikan data kepada 3 bahagian dan kajian terhadap kepanjangan suara dan nombor untuk neurons di lapisan terlindung telah dilakukan. Eksperimen juga dilakukan untuk mengkaji hubungan di antara jangka tunggal dan jangka ramai. Keputusan telah menunjukkan suara yang panjang lebih sesuai untuk digunakan untuk klasifikasi suara kanak-kanak Melayu. MLP dan RNN telah mencapai keputusan 83.79% and 83.10%. Vokal /i/ telah mendapat keputusan yang tertinggi di antara semua vocal yang telah dikaji.

ACKNOWLEDGEMENTS

First and for most, I would like to extend my sincere thanks to my supervisor Dr. Ting Hua Nong that have encourage, support and help me in completing my research project successfully and for his valuable guidance and advice.

Besides, my thank and appreciation goes to Mr. Yong Boon Fei who has willingly help me out with his abilities.

Last but not least I would like to express my gratitude towards my beloved parents for their understanding, endless love and encouragement when it was most required.

Contents

List of Tables

List of Figures

List of Abbreviations

ANN	**A**rtificial **N**eural **N**etwork
ASR	**A**utomatic **S**peech **R**ecognition
DTW	**D**ynamic **T**ime **W**arping
GUI	**G**raphical **U**ser **I**nterface
HMM	**H**idden **M**akrov **M**odel
LPC	**L**inear **P**redictive **C**oding
MLP	**M**ulti-**L**ayer **P**erceptron
NN	**N**eural **N**etwork
RNN	**R**ecurrent **N**eural **N**etwork
SVM	**S**upport **V**ector **M**achine

1 CHAPTER I: INTRODUCTION

1.1 Overview

Speech is one of the most common ways of communication for human being. However, the process of this phenomenon from learning relevant skills until performance is complicated. A number of efforts have been done on Automatic Speech Recognition (ASR) systems which are used to convert spoken words and statements into a form of machine response such as: Dynamic Time Warping (DTW), Hidden Markov Model (HMM), Support Vector Machine (SVM) and Neural Networks (NNs) (El Choubassi, El Khoury, Alagha, Skaf, & Al-Alaoui, 2003; Giurgiu, 1995).

The importance of brain in cognitive skills such as speech recognition has motivated the researchers to investigate the brainlike models that may lead to brainlike performance on various complex tasks (Figure 1.1). This research area is called Artificial Neural Network (ANN).

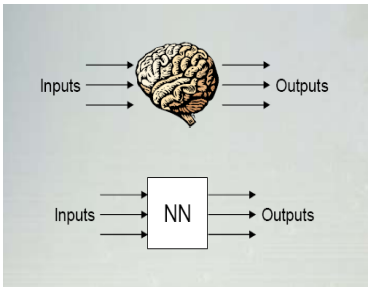

Figure 1.1: Resemblance of Brain and ANN

The human brain uses a set of simple processing units (neurons) connected by weights (synapses), that the strength of them can be adjusted with experience, to support and provide memory in learning in the biological systems. This is the true of ANNs (Zebulum, Vellasco, Perelmuter, & Pacheco, 1996).

In general, neural networks are structures consist of three different layers: input, output and at least one hidden layer. These systems are made of interrelated computational nodes functioning somehow similar to human neurons (Ala-Keturi, 2004). They can classify related data or make interpolation or extrapolation in a multi-dimensional space after a training phase to recognize nonlinear patterns and solve complex problems. So neural networks have a special position among speech recognition systems as great adaptive nonlinear classifiers.

In most of the languages, the consonants and vowels have the higher frequency than the subwords unit. So recognition of the Consonant-Vowel is essential to develop a speech recognition system with an acceptable accuracy (Nazari, Sayadiyan, & Valiollahzadeh, 2008). Since vowels are particular phenomena of each language (Thasleema, Kabeer, & Narayanan, 2007) and speaker independent of vowels is the main difficulty in speech techniques, the researchers have done many efforts to find a suitable and acceptable method to recognize vowels and furthermore words.

The purpose of collecting data and samples from children in this study is that children's speech is more challenging than adults due to higher pitch (Lee, Potamianos, & Narayanan, 1999) and rapid changes in speech features as a function of age during growth (Lee & Iverson, 2009).

1.2 Research Problem and Problem Statement

A number of studies have reported the performance of speech recognition techniques, but a few of them have compared the existing methods. This study aims to compare the performance of two different basic architectural neural networks (MLP and RNN) for sustained Malay vowel recognition of Malay children. The research attempts to find the

best frame of speech signal with the highest recognition rate and sufficient number of hidden neurons for Malay vowel recognition and discuss the performance of each method.

1.3 Significance of the Study

The application of speech recognition systems are enormous, including voice dialing, data entry, speech to text processing, biometric systems, training air traffic controllers, etc. The result of this research leads to some significant factors that are important in Malay speech recognizers to get optimal performance. Finally, this study can contribute to the general knowledge in terms of enhancing our experience in different methods of vowel recognition of Malay people and later can be developed to other languages. It will help us to develop systems that can receive spoken data and respond with higher speed and more accurately.

1.4 Scope of the Study

This research considers samples of vowels for healthy Malay children between 7-12 years old. It is a speaker independent study that is focused on six Malay vowels (/a/, /e/, /ə/, /i/, /o/ and /u/) and does not cover consonants or other languages. The performance of two basic methods of ANN including MLP and RNN are compared by using Neural Network Toolbox™ in Matlab® software (R2010a).

1.5 Objective of the Study

1.5.1 Main Objective

This research is aimed to find out the recognition rate of MLP and RNN architectures for sustained Malay vowel recognition of Malay children.

1.5.2 Specific Objectives

- Discover the appropriate signal length to extract the features of vowel signals in Malay children vowel recognition.
- Specify the proper number of hidden neurons to obtain the optimal performance of MLP and RNN architectures.

1.6 Organization of the Dissertation

This study includes five chapters. In each chapter several subtopics are discussed and some figures, tables, data and references are showed.

Chapter one introduces the Artificial Neural Network as one the main technologies in ASR systems. The importance of recognition of the Consonant-Vowel to develop a speech recognition system is mentioned in this chapter. Also, the aim and objective of this study are discussed.

Chapter two previews the literatures and focuses on the available methods in the field of speech recognition, competencies of neural network in vowel recognition and compares two basic architecture of neural networks including: MLP and RNN.

Chapter three in particular deals with a speech feature extraction method to find the parametric coefficients of the collected samples and then classifying them by training and testing via two different types of neural networks using Matlab® software to analyze the results and compare the performance of each network to find the optimal performance.

Chapter four shows the best result of data analyzing of each method and the performance of the most accurate frame of speech signal with different number of hidden neurons.

Chapter five simply concludes the results of the study. It mentions future plans to improve the speech recognizers with better performance and more accurate result in noisy environment.

2 Chapter II: LITERATURE REVIEW

2.1 Introduction

Speech is the most usual mode of communication among humans. Hence, speech recognition has become a favorite topic for many scientists to do research on Automatic Speech Recognition (ASR) systems. ASR system is a technology for computers to identify words spoken into telephone or microphone and convert it into text (Singh Gill, 2008). It consists of two separate segments (Figure 2.1), Feature Extractor and Recognizer (Kumar, Kumar, & Rajan, 2009). The Feature Extractor is used to transform a large amount of input data into a collection of feature vectors. The Recognizer's duty is to figure out the correlation between the vectors and recognize the spoken words.

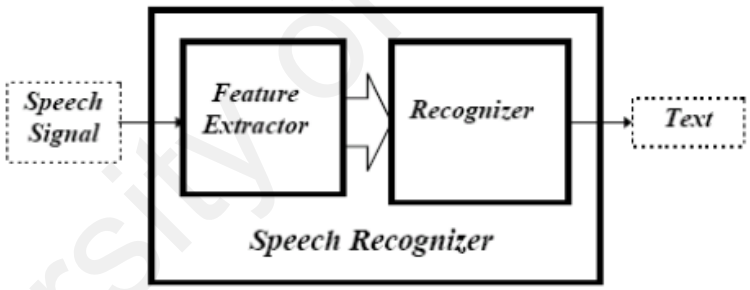

Figure 2.1: Segments of a Speech Recognition System (Kumar, et al., 2009)

Generally, these systems use dominant algorithms as Recognizers like: Dynamic Time Warping (DTW), Hidden Markov Model (HMM), Support Vector Machine (SVM) and Neural Networks (NNs) (El Choubassi, et al., 2003; Giurgiu, 1995).

DTW is one the oldest method which applies the differences between frame's times to do adjustment and recognition. Later on, ANN substituted DTW. At last, HMM and SVM were created to improve the recognition rate (Zhao & Han, 2010).

2.2 Dynamic Time Warping (DTW)

One of the important and doing-well tools in the field of isolated-word recognition is DTW that approximates the similarities and differences between two warped nonlinearly sequences like time series to figure out optimal matches (Singh Gill, 2008). In this technique, a template which is a particular utterance of the spoken word, can match a group of training utterances in the best way.

The main difficulty in applying this method is how to obtain speaker independent templates (Liu, Lee, Chen, & Sun, 1992). A solution to this problem is given by Rabiner et al. (1979) and Wilpon et al. (1985). They suggested that the data can be divided into several clusters, so one template can be achieved for each. The total number of templates for each word depends on the task ranging 10 to 30. This approach will help to have adequate number of templates to reach optimal classification performance. However, this idea has a disadvantage. It is difficult for the speaker independent system with a large number of vocabularies and 10 to 30 templates to discriminate dissimilar properties of speech signals of different speakers.

It seems the combination of DTW with other beneficial approaches in speech recognition that are discussed later may lead to more accurate results because it can extract the profits of each method. Thus, it is possible to gain from the characteristic of DTW in representation of the speech signal with temporal structure regarding to its capability in high time alignment without considering its drawback in speaker independent variations (Bourouba, Bedda, & Djemili, 2005).

2.3 Artificial Neural Network (ANN)

The other common and useful technique that has many applications in recognition systems and was applied in signal processing since the late 1980s is ANN (Wang & Zheng,

1998). Neural Networks are composed of a series of layers (input – hidden and output layers) containing neurons that work similar to human brain. The input and output layers are interconnected with each other by certain weights like synapses in the biological system (Ala-Keturi, 2004; Kumar, et al., 2009). Figure 2.2 illustrates the similarities between a physical neuron and an artificial neuron.

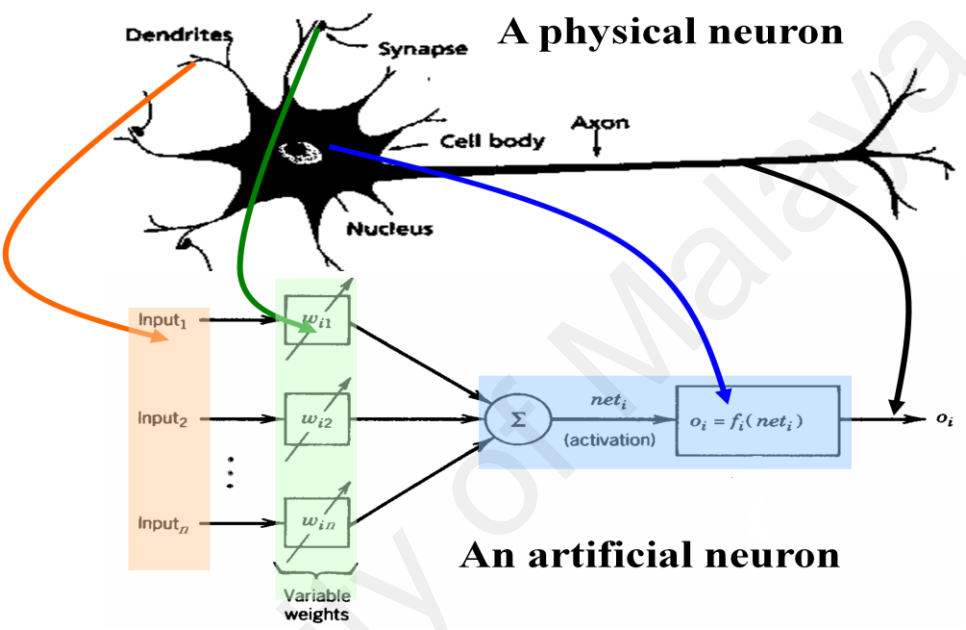

Figure 2.2: Correspondence between physical neuron and artificial neuron

These systems do the processing through two steps (Gao, Chen, Zeng, Liu, & Sun, 2009):

1- Training/Learning: is the process of initializing and altering the weights to generate a network that can execute some functions. There are some types of the algorithms for the network to learn the correlation between the inputs and outputs. The most common one is Back- Propagation. The responsibility of the training algorithm (activation function) is minimizing the prediction errors caused by the network due to differences between the actual output and target output.

2- Testing: a set of new inputs are fed to the generated network to predict output.

The superiority of the neural networks is their flexibility and adaptability to particular situations and complex non-linear tasks by adjusting the weights and connection strength to give acceptable results (Kumar, et al., 2009; Leonida, 2000). They have a great ability in discrimination between classes (Paliwal, 1991). They are very useful in parallel computation and controlling the speech recognition processing as they can learn complicated features from the inputs due to the artificial neurons' non-linear structure (Oglesby & Mason, 1990; Widrow, Winter, & Baxter, 1988). In addition, utilizing ANN as a classification technique can eliminate the complications and difficulties related to time-varying data (Giurgiu, 1995).

Taking account of that ANNs can give us desired recognition rates, they also have some drawbacks. The process of how the neurons are trained in the hidden layer is obscure and analyzing the obtained weights in the network to find a pattern is speculative and rough. Moreover, it is sometimes impossible for the neural networks to assure an optimal result which is because of the various types of training functions such as gradient descent to get local minima for the function (Leonida, 2000).

Nevertheless, ANNs are become attractive for many researchers during recent decades to improve the performance and accuracy of ASR systems. The two most common neural network architectures in the field of speech recognition includes: Multi-Layer Perceptron and Recurrent Neural Network which utilize supervised learning techniques to train the network.

2.3.1 Multi-Layer Perceptron (MLP)

Neural networks have the ability of providing high computation rates by using analog components and parallel algorithms (Huang, Lippmann, & Gold, 1988). Multi-Layer Perceptron is one of the neural network types that is widely used in static pattern

classification trained with back propagation to minimize the training errors (Ahad, Fayyaz, & Mehmood, 2002; Hao & Ravi, 1995). This network architecture which usually consists of an input layer of source neurons, at least one hidden layer of computational neurons and an output layer (Figure 2.3), is capable to approximate nonlinear functions of static models (Ghaemmaghami, Razzazi, Sameti, Dabbaghchian, & BabaAli, 2009). It has a feed forward structure that uses gradient descent algorithm to decrease the mean square errors in the output. The training will continue until it achieves the highest number of epoch or lowest amount of training error (Hua Nong & Yunus, 2004).

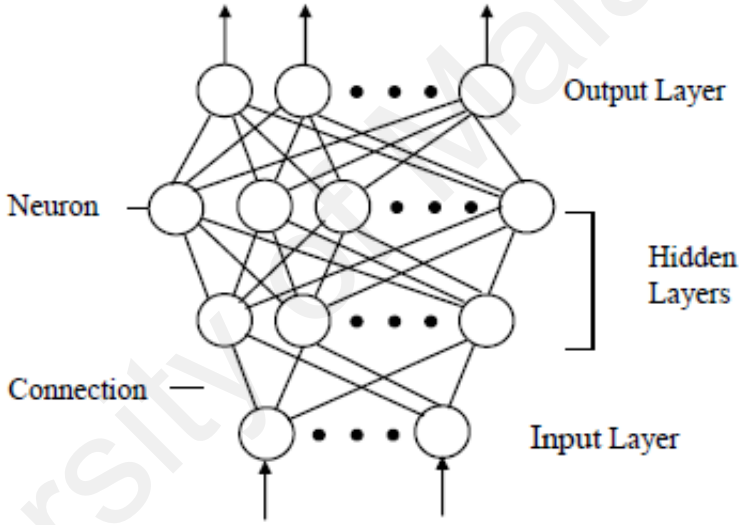

Figure 2.3: MLP Architecture (El Choubassi, et al., 2003)

There are some problems with MLPs. As they need to work with fixed-length input data, they are not useful in classification of dynamic signals such as speech. Also, adding number of connections in MLP may lead to longer training time and weak local minima (Ala-Keturi, 2004; Paliwal, 1991).

2.3.2 Recurrent Layer Neural Network (RNN)

Since time is a dimension of input feature in speech process, a dynamic structure model like RNN can have better performance than a static one (i.e. MLP) in a ASR system. In general, RNNs are a combination of feed forward network and feedback structure between units of various layers (Ahmad, Ismail, & Samaon, 2004). They have the capability to store the past information in a layer called context layer which is useful in processing arbitrary orders of inputs (Figure 2.4). So RNNs are able to deal with time varying and dynamic information which has a great importance in ASR systems (Bronzino, 2000).

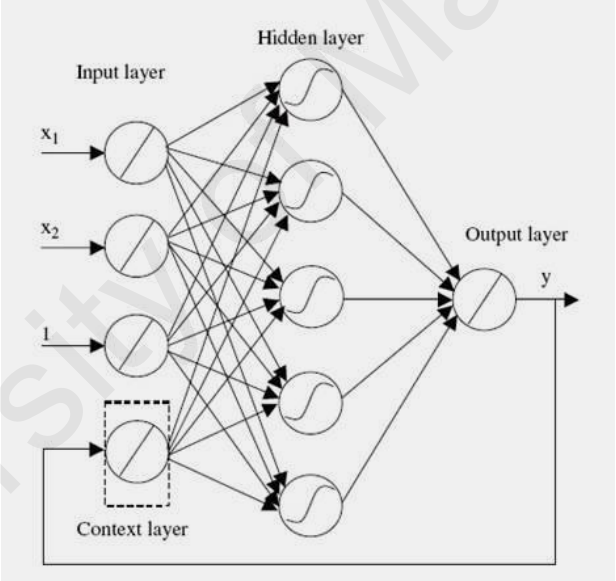

Figure 2.4: RNN Architecture (Ahmad,2004)

This type of neural network has also some advantages and weakness. Compared with the MLP, RNN has better speech recognition performance but it has a more complicated, slower and sensitive algorithm due to high amount of computation process. In addition, its effectiveness in learning extensive connected order is still uncertain (Albesano, Gemello, & Mana, 1992).

The strengths and weaknesses of the two mentioned network architecture are summarized in the following table.

Table 2.1: Comparison between MLP and RNN

Neural Network Architecture	Strength	Weakness
MLP	- Capable to estimate non-linear structures - Aptitude to learn - Robustness	- Difficulties in dealing with temporal pattern - Fixed length of inout pattern
RNN	- Storing past information in a context layer	- More complicated training algorithm - Unsure efficiency for learning long connected sequence

Although, most of the existing ANNs have high efficiency in clean environment, they cannot cope with the noisy conditions and they are weak in temporal information processing (Wan, 1990).

2.4 Hidden Makrov Model (HMM)

Hidden Makrov Modeling is one of the most successful approaches in speech recognition. It can be described as a simple dynamic Bayesian network with invisible states and visible outputs based on the states. The only parameters in HMM is the state transition likelihoods. A stochastic representation of specific utterance can be provided by HMM and according to the likelihood that a word model the observed speech, the similarities can be measured (Giurgiu, 1995). HMM has a great ability in modeling the time variability of

speech which is related to dynamic patterns and it needs small amount of calculation. On the other hand, it cannot discriminate between classes like words and phonemes (Paliwal, 1991). HMMs are more suitable for speaker recognition.

Hidden Makrov Models and Artificial Neural Networks have two main differences (Renals, McKelvie, & McInnes, 1991):

1- The static and permanent features of the speech signal can be modeled by feed forward neural networks, whereas HMMs are able to offer a clear time dependent model of speech signal through the transmission between model states. It is important to mention that recurrent neural networks are more appropriate in speech recognition by modeling sequences of arbitrary complication.

2- The training phase in neural networks is done by minimizing the errors between the target and output and maximizing the probability of correct classes, while HMMs are trained via a highest likelihood procedure. In recent years, the discrimination between classes in HMM is developed by using ANN framework.

The fact that ANNs have a great ability in short time acoustic classification and considerable limitations in long sequences to present complete sentences even when a RNN architecture is applied, has encouraged scientists to combine ANN and HMM in order to produce a new model called Hybrid HMM/ANN (Xian, 2009). Advanced research in using Hybrid HMM/ANN has shown beneficial results for ASR systems. Such a system opens the possibility of taking advantage from the features of both techniques and leads to obtain higher recognition performance.

2.5 Support Vector Machine (SVM)

The original SVM algorithm, invented in 1995(Cortes & Vapnik, 1995), performs classification and regression by separating the data into N-dimensional hyper-plane. For

instance, in the field of speech recognition, this computer algorithm learns by segmenting phonemes in continuous speech (Juneja & Espy-Wilson, 2003). Indeed, a SVM that uses kernel function can be considered same as a two-layer perceptron neural network.

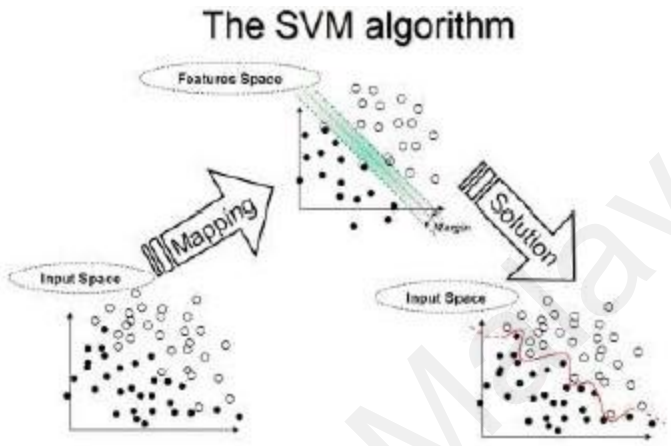

Figure 2.5: A Two Dimensional Example of SVM

From: http://www.dtreg.com/svm.htm

SVMs are capable to learn from small amount of input data and control high dimensional data with precision. But like neural networks, their ability is limited in ASR systems due to the poor model of dynamic and time varying articulation (Juneja & Espy-Wilson, 2003).

2.6 Vowel Recognition

A large amount of researches have been done on vowel speech recognition in different languages. Carlson and Glass (1992) examined the effects of speech-synthesis-like parameters on several vowel classification methods. Giurgiu (1995) investigated the capability of ANN in speaker independent vowel recognition. The obtained recognition rate with 50 hidden neurons and 5 output neurons to recognize Romanian vowels was at around 96%.

In another previous studies, DTW was used as a classifier to enhance the accuracy and recognition rate of Cantonese vowels (Fu, Lee, & Clubb, 1996). An averaged accuracy of 94% was achieved by using their methodology. Although they obtained an acceptable recognition rate, but the method they used needed considerable amount of computation especially for large number of input units.

Nazari et al. (2008) used a combination of kernel-based feature extractor and SVM classifier with non-linear dimension reduction procedure to recognize Persian vowel speech and attained a recognition rate of 93.9%.

Damien (2011) proposed a new effective method based on HMM classifier to discriminate vowel recognition from consonant recognition in classical Arabic language and obtained a recognition rate of 81.7%.

2.7 Malay Speech Recognition

In the field of Malay recognition, many efforts have been made by Malaysian researchers. Salam et al. (2001) studied the performance of neural network using generic algorithm and handcrafted (trial and error) neural network in recognizing isolated Malay digits (0 to 9). In addition, Ting et al. (2001) implemented MLP and DTW techniques to classify Malay vowels. Furthermore, Ting and Yunus (2004) investigated the recognition rate of six Malay vowels of Malay children in a speaker-independent system using MLP network fed with cepstral coefficients. Later, Ting and Mark (2008) examined the capability of NN to recognize Malay vowels of a Malay child with articulation disorders. Al-Haddad et al. (2009) attained 98% in Malay digits recognition using Hybrid HMM/DTW and Recursive Least Squares algorithm to cancel the noise. Recently, Shahrul et al. (2010) suggested a novel technique for Malay vowel recognition based on formant and spectrum envelope using single-frame analysis.

A further precise literature review has proved that development of ASR systems is still under investigation, mostly using multi-frame analysis in dependent and independent speaker systems.

3 CHAPTER III: METHOD AND PROCEDURE

3.1 Introduction

In this study, recognition rate of six Malay vowels (/a/, /e/, /ə/, /i/, /o/ and /u/) of Malay children between 7-12 years old by using two types of neural networks (MLP and RNN) are evaluated and compared. It is a speaker independent study in which the obtained data are analyzed using Matlab ® software. The reason of using this software lies on its ability to do numerical calculations without needing time consuming and massive programming. Also, there are some interface functions to transfer data between Matlab ® and C++ easily. After data collecting and analyzing, confusion matrix were produced to find out the recognition accuracy of each vowel for the frame of speech signal with the best performance. Figure 3.1 represents the methodology structure of this study.

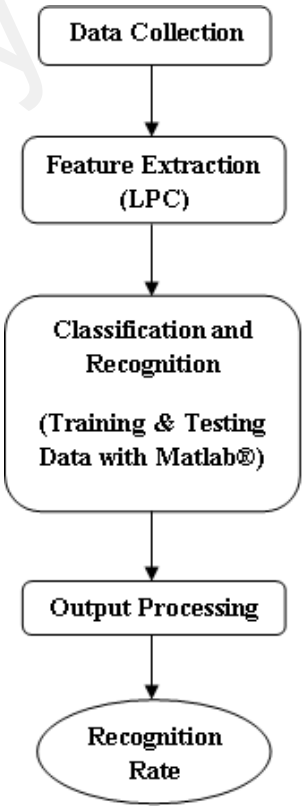

Figure 3.1: Methodology Structure

3.2 Data Collection

The database for this study was collected from 60 Malay children (30 males and 30 females) within the age range of 7 to 12 years old at 20 kHz with 16-bit resolution in normal condition (Ting, 2004). So totally we have 360 samples.

As recently mentioned in previous chapter, the first part of an ASR system is Feature Extractor. In this study, the Linear Predictive Coding (LPC) which is an autocorrelation analysis (Makhoul, 1975) was used to extract features and convert the speech signal to parametric coefficients. LPC is based on the linear arrangement of the past signal samples to predict the current sample. The LPC parameters were obtained from autocorrelation coefficients by LPC analyzing using C++ program. These parameters formed the main input data for this study.

There is a technique called *k-fold cross validation* which is used to evaluate the accuracy of data set in a predictive model to perform a practice. The data is broken up into *k* subsets and the method will run *k* times. Each time, one of the subsets is considered as a testing set and the rest (*k-1*) are placed together as a training set (Krogh & Vedelsby, 1995). This method will decrease the error due to data dividing. The drawback of this tecnique refers to k-times repetition that causes additional computation (Faisal, Taib, & Ibrahim, 2010). In this study, obtained data were divided into 3 sets according to *3-fold cross validation.*

To extract the speech features, the speech signal was evaluated as single-frame and multi-frame approaches. The examined signal length included: 10ms, 15ms, 20ms, 25ms, 30ms, 35ms, 40ms, 45ms, 50ms, 55ms, 60ms, 65ms and 70ms for a single-frame and 30ms, 40ms, 50ms, 60ms, 70ms, 80ms, 90ms and 100ms for multi-frame analysis. So, each set of collected samples contains 13 single-frame and 8 multi-frame data (Figure 3.2).

Each frame contains 360 samples that were divided, according to *3-fold cross validation* into 240 samples as training and 120 samples as testing data.

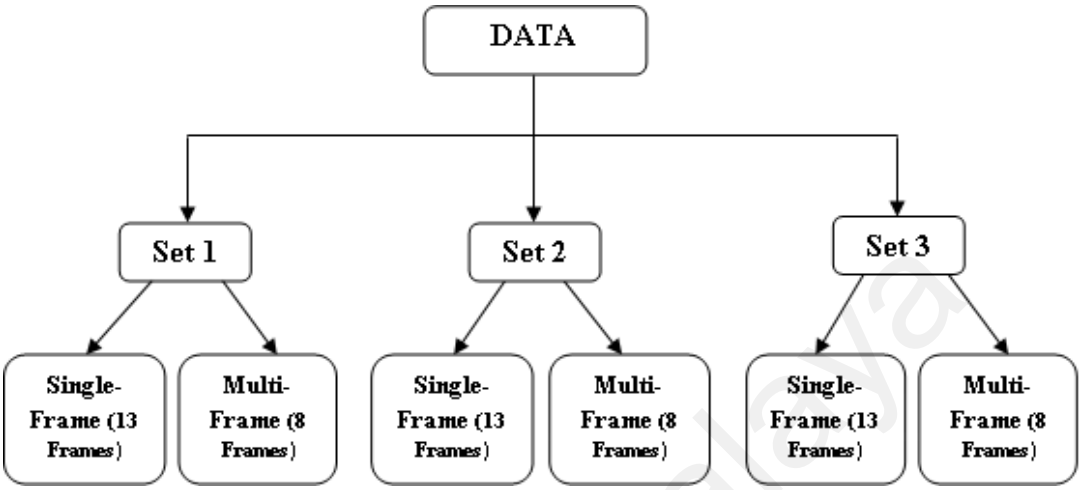

Figure 3.2: Block Diagram of Data Grouping

In this study, the LPC order for each single length of single-frame analysis was 24 according to the experiments to ache optimum performance. For the mentioned frames of the multi-frame analysis the LPC order was not fixed. It was: 48, 72, 96, 120, 144, 168, 192 and 216 respectively for each frame. Indeed, the order of LPC refers to the number of cepstral coefficients that are used to represent the various features of the signal.

3.3 Classification and Recognition

The purpose of this study is evaluating the performance of MLP and RNN and find out the best frame of the speech signal with the sufficient number of hidden neurons in each architecture. The first step was loading the data which are cepstral coefficients obtained by LPC, from C++ to Matlab®. As Matlab ® deals with matrices, so it is essential to define input, target and test matrices for each type of network before constructing a neural network model for classification.

3.3.1 Create Input Matrix

The input matrix of the training phase is a $m \times n$ matrix where m refers to the number of cepstral coefficients in each frame of the signal and n is the number of samples for each vowel. Actually, m determines the number of neurons in the input layer. As there were 240 samples and six vowels for each one in the training phase, n was equal to 1440 (240×6). The order of samples' vectors was as follows which is repeated 240 times to create the training matrix.

$$Input = [\ /a/\ \ /ə/\ \ /e/\ \ /i/\ \ /o/\ \ /u/\ ] \qquad (3.1)$$

3.3.2 Create Target Matrix

The target matrix, contains the desired output, is used to classify the six vowels via comparing it with the output matrix produced after simulating the network. Hence, the basic block of target matrix was designed as follows:

$$Target = \begin{matrix} /a/ & /ə/ & /e/ & /i/ & /o/ & /u/ \end{matrix} \begin{bmatrix} 0.9 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.9 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.9 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.9 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.9 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.9 \end{bmatrix} \qquad (3.2)$$

Where the number of rows represents the six vowels and the number of columns refers to the number of input samples. The values "0.1" and "0.9" were selected instead of "0" and "1" to prohibit necessity of large amount of weights when a sigmoid function is utilized in the hidden layer (Sorsa, Koivo, & Koivisto, 1991).

This matrix was replicated 240 times with the *repmat* command to form the final 6×1440 target matrix. In other words, *repmat (Target,1,240)* was used to create a large matrix consisting of *1-by-240* tiling copies of *Target*.

3.3.3 Create Test Matrix

The test matrix was employed in the testing phase to simulate the network after training it. The importance of this matrix is in assessing the generalizability of the neural network to predict the output. The elements of this matrix were provided by the independent testing set of data (120 samples) that were not used during training. So the test matrix had 720 columns (120 samples × 6 vowels) and the number of rows depends on the number of cepstral coefficients same as the input matrix.

3.3.4 Construct Neural Network

After importing the training and testing samples of data sets and defining the input, target and matrices, Neural Network Toolbox™ (*nntool*) was used to do the classification. Neural Network Toolbox™ is a type of Graphical User Interface (GUI) in Matlab® which has its own work area, separate from command-line workspace to model and perform training and testing neural networks. In this study, to get more accurate results in a short time, *nntool* was used mainly to develop a model of ANN.

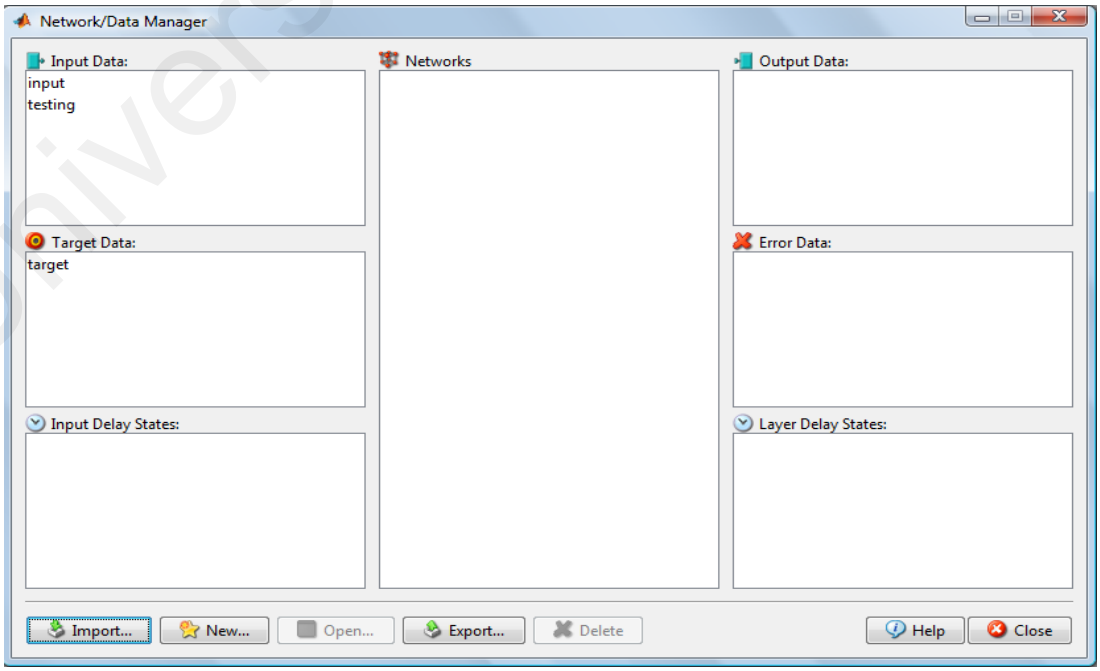

Figure 3.3: Main GUI of Neural Network Toolbox™

Network/Data Manger (Figure 3.3) is the main part of Neural Network Toolbox™ which is an interface between Matlab® console and this toolbox to create, add or subtract and manipulate data in the neural network. The second part of GUI (Figure 3.4) is used for showing the structure of the network, training, simulating, initializing the weights and etc.

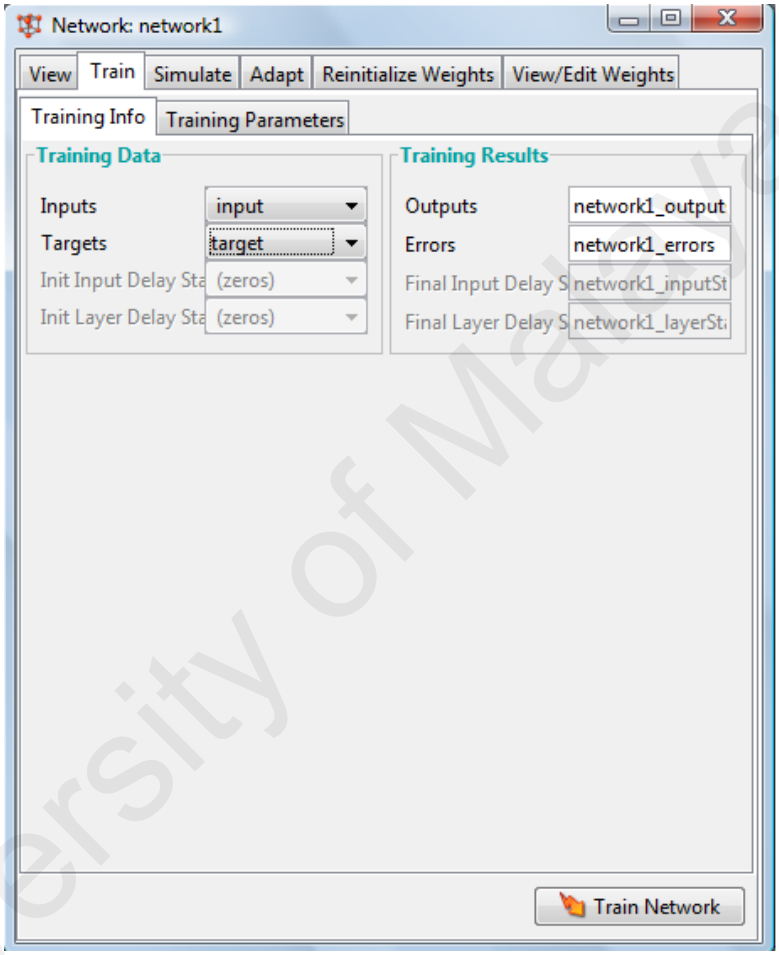

Figure 3.4: Second GUI of Neural Network Toolbox™

3.3.5 Multi-Layer Perceptron (MLP) Analysis

Alsmadi et. al. (2009) have proved that back propagation algorithm is the best one among the MLP techniques (Alsmadi, Omar, & Noah, 2009). So, Feed-Forward Back Propagation was used to train the data for this type of neural network with two layers including one hidden layer plus output layer (Figure 3.5).

The signal was examined in various lengths of single-frame and multi-frame as mentioned before.

It is obvious that the number of neurons in the hidden layer has a direct effect on the performance of the network and recognition rate. Training the ANN with small number of hidden neurons may lead to the poor results (Giurgiu, 1995). In this experiment number of hidden neurons was varied from 10 to 200 (10, 20, 40, 60, 80, 100, 120, 140, 160, 180 and 200) for single-frame analysis and multi-frame analysis of different signal lengths. So, the training and testing was done 11 times for each of the frames with different number of hidden neurons as mentioned above.

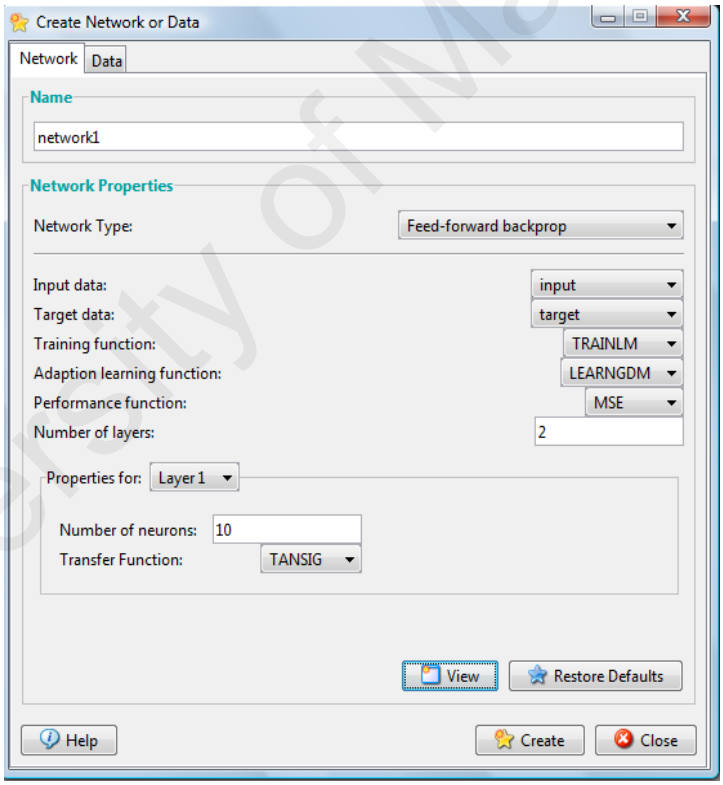

Figure 3.5: Creating Network

There are several training functions adapted in Neural Network Toolbox™. Selecting the best training function depends on some factors such as error goal, number of training input data points, weights and biases. To train the network, Levenberg-Marquardt (*trainlm*) function was used for the single-frame data, which is the fastest one in many cases to get

less mean square errors. However, storing large amount of matrices in certain problems limits the application. To cope up with this error, Gradient Descent Back-propagation with Adaptive Learning Rate algorithm (*traingdx*) was applied for the multi-frame analysis to re-edify the weights and biases pursuant to adaptive learning rate and gradient descent momentum.

TANSIG which is a Tangent Sigmoid transfer function was utilized to calculate a layer's output ranging from -1 to 1 from its net input. The structure of the network is shown in Figure 3.6.

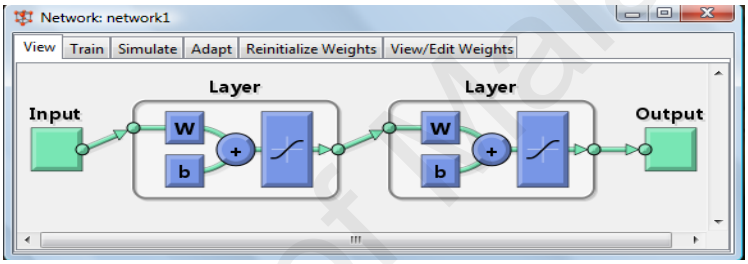

Figure 3.6: Feed Forward Back Propagation with Two Layers

The training parameters were assigned as shown in the Figure 3.7. To avoid the overfitting which can be caused by the error of learning on the training data set when drops under specific threshold, the performance goal was considered as zero.

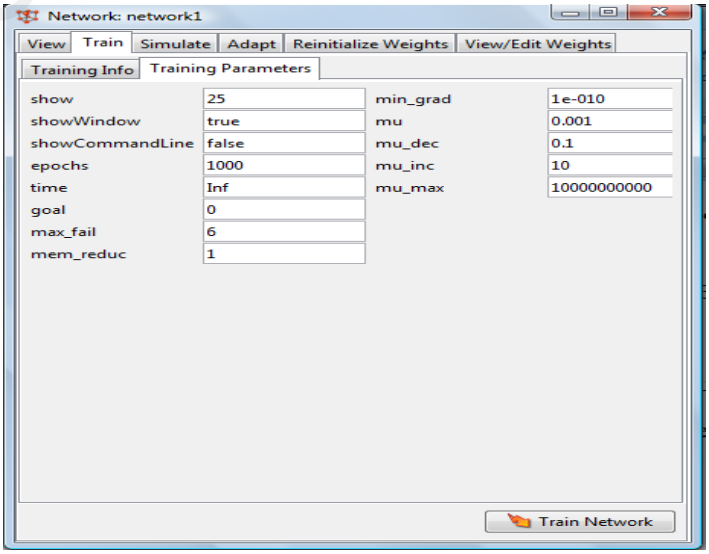

Figure 3.7: Assigning the Training Parameters

Training the network leads to open the following window (Figure 3.8) which shows some characteristics such as number of epochs and performance time.

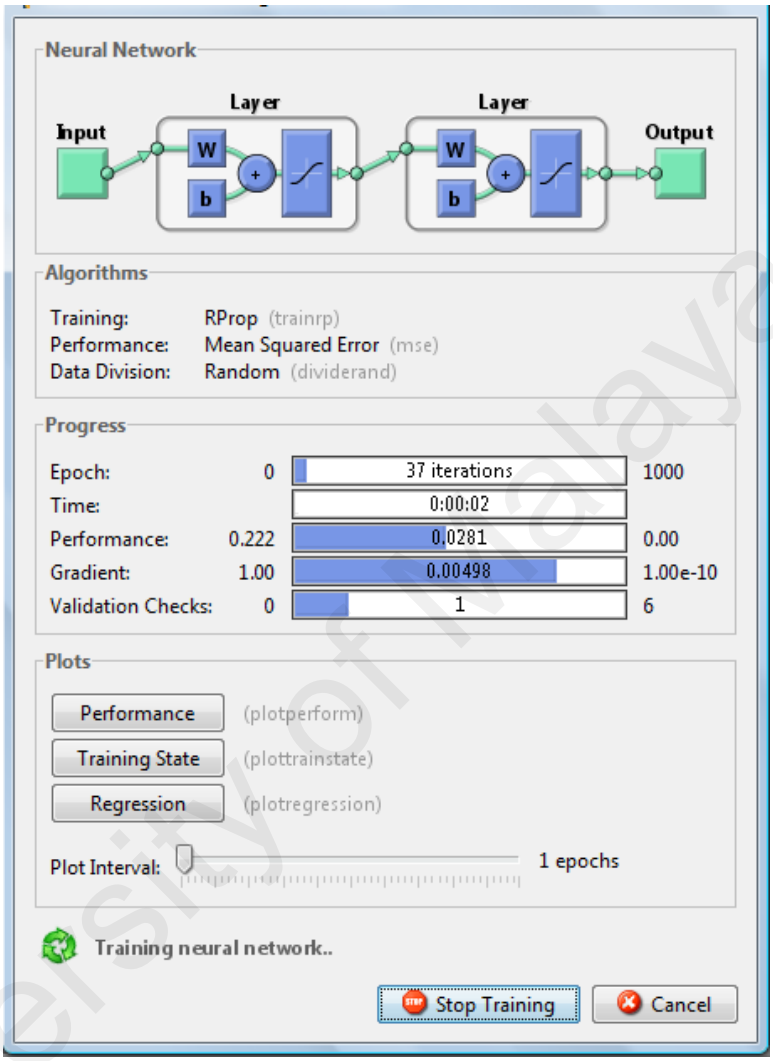

Figure 3.8: Training the Network

For most of the algorithms, the training stops when any of the following situations occur:

- The number of *epochs* reaches the maximum which was 1000 in the current study.

- The minimum training error is achieved.

To do the second step of process which is testing phase, the network should be simulated (Figure 3.9) with a new set of independent input data. This state was led to produce a 6×720 matrix as an output that was used to create the *Confusion Matrix*.

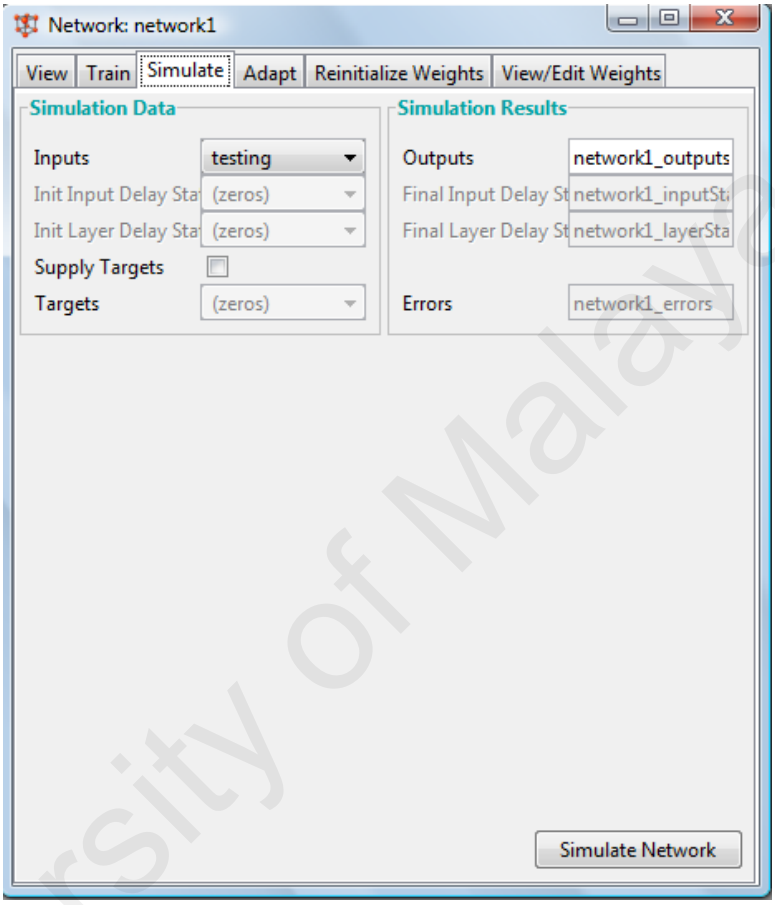

Figure 3.9: Simulating the Network

3.3.6 Recurrent Neural Network (RNN) Analysis

The same procedure was done to evaluate the performance of RNN. The only difference relates to the training function. The experiments showed that Random Order Weight/Bias Learning Rules (*trainr*) algorithm is more suitable for RNN considering the training time and the out of memory error.

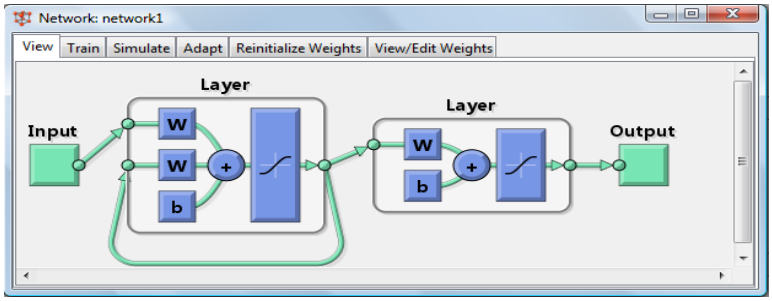

Figure 3.10: Layer Recurrent Network with Two Layers

3.3.7 Output Processing

The output matrix produced after simulation needs some modification to become more practical. The elements of this matrix were transformed to "0" and "1" by the programming code provided in the Appendix A. An example of the output matrix after and before modification is as follows:

$$\begin{bmatrix} 0.84 & 0.11 & 0.10 & 0.11 & 0.13 & 0.12 \\ 0.12 & 0.82 & 0.10 & 0.10 & 0.11 & 0.15 \\ 0.25 & 0.10 & 0.89 & 0.10 & 0.14 & 0.13 \\ 0.10 & 0.42 & 0.10 & 0.89 & 0.10 & 0.11 \\ 0.13 & 0.10 & 0.21 & 0.10 & 0.18 & 0.25 \\ 0.15 & 0.10 & 0.14 & 0.32 & 0.80 & 0.75 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Before Modification — After Modification

Then the columns of the modified output matrix were compared to the following columns to recognize the columns of the output matrix according to the order of vowels in the test matrix:

$$/a/ = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad /ə/ = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad /e/ = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad /i/ = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad /o/ = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \text{ and } /u/ = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.3)$$

In some rare cases, the output's column was not equal with any of the above columns. In this situation, it was stored as "others".

The last stage is creating the *Confusion Matrix* to figure out the recognition rate of the neural model. *Confusion Matrix* is a visualization tool to estimate the actual and predicted classes generated by a classification system. The importance of this matrix is evident while two classes are mislabeled or confused in the system. The rows of the matrix show the samples in the actual class and the columns show the samples in the predicted class. The accuracy of the system can be easily obtained from this matrix through feeding it with the output and target patterns.

4 CHAPTER IV: RESULTS AND DISCUSSIONS

4.1 Introduction

This chapter discusses the results which are achieved from Neural Network Toolbox™ (*nntool*) of Matlab® and programming codes, and compares the performance of the two architectures of neural network: MLP and RNN. The results are presented in two separate parts for each technique that are dedicated for single-frame analysis and multi-frame analysis of different signal lengths. Then the confusion matrix is shown for the frame of each method with the highest accuracy.

4.2 MLP Results

4.2.1 Single-Frame Analysis

The experiment was done for each frame of the vowel signal at different number of hidden neurons for the three sets of data. Table 4.1 represents the highest recognition rates with the number of hidden neurons for the different signal lengths of the three sets using MLP architecture. The average rates of the three sets were calculated for each frame. It is obvious from Table 4.1 that the highest average rate was obtained from single-frame 55ms.

Table 4.1: Recognition Rates (%) of each Single-Frame Data Set using MLP Architecture

Signal Length (ms)	No. of Hidden Neurons	SET 1 (Recognition Rate %)	No. of Hidden Neurons	SET 2 (Recognition Rate %)	No. of Hidden Neurons	SET 3 (Recognition Rate %)	AVERAGE (%)
10	120	80.27	180	79.86	120	77.5	79.21
15	200	81.11	180	81.25	120	79.16	80.51
20	120	81.44	120	81.38	200	80.69	81.17
25	120	81.38	120	81.8	80	81.38	81.52
30	120	81.8	200	83.61	20	81.66	82.36
35	120	82.77	180, 200	83.61	80	83.61	83.33
40	120	82.63	180	84.3	80	84.16	83.70
45	120	83.33	200	83.19	180	84.3	83.61
50	120, 180	83.19	120, 180	82.77	20	83.75	83.24
55	120	84.16	180	83.61	120	83.61	83.79
60	120	83.33	80	82.22	120	83.88	83.14
65	120	82.08	120	82.91	120	83.89	82.96
70	120	82.08	20	84.02	20	85.13	83.74

Figure 4.1 illustrates the performance of each single-frame with the highest recognition rate using MLP network. The highest recognition rate obtained is 83.79% related to the single-frame 55ms vowel signal and the lowest rate is 79.21% which was achieved from single-frame 10ms.

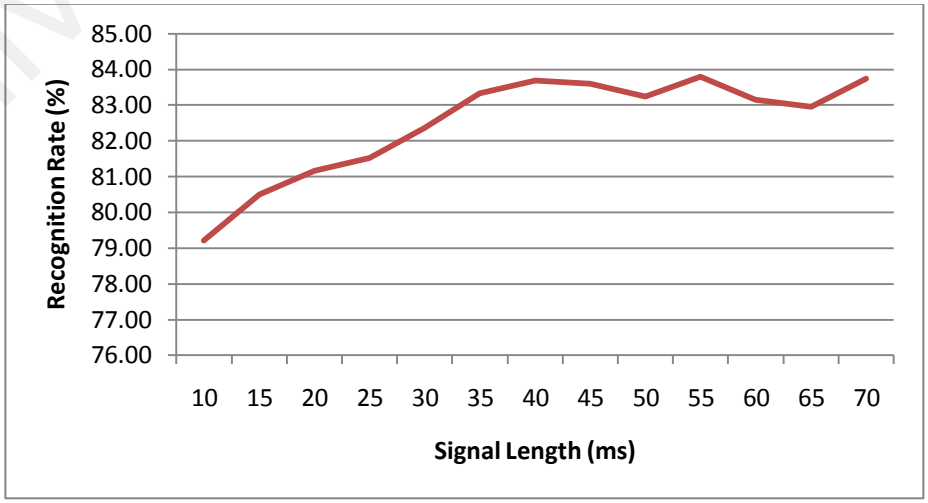

Figure 4.1: Recognition Rates (%) of each Single-Frame Vowel Signal using MLP

To study the effect of the number of hidden neurons on the performance of the network, the following chart (Figure 4.2) was plotted for SF55ms which got the best recognition rate among the other frames. The highest recognition rate was obtained at 120 hidden neurons, whereas 60 hidden neurons caused the lowest rate.

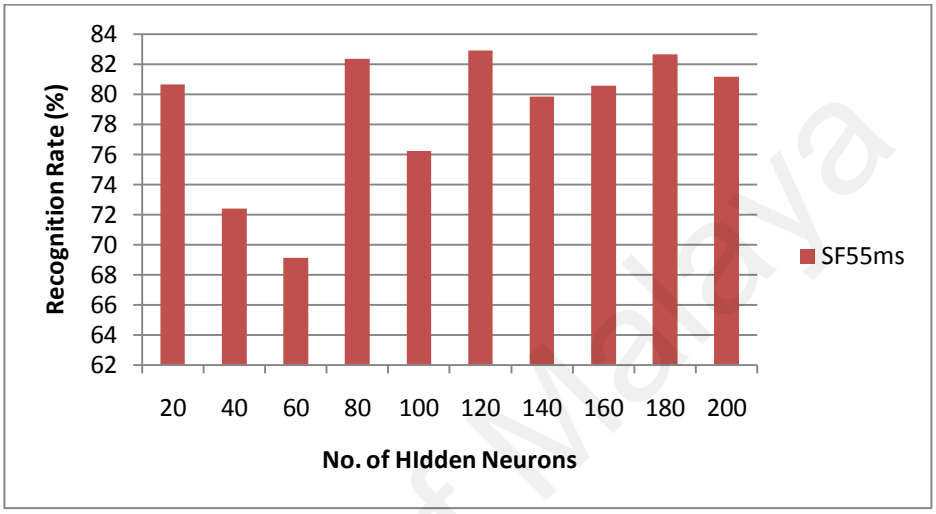

Figure 4.2: Recognition Rates (%) of SF55ms Vowel Signal at Different Number of Hidden Neurons using MLP

The confusion matrix of the single-frame 55ms data trained with 120 hidden neurons is presented in Table 4.2. The greatest accuracy was achieved by the vowels /a/ and /i/ as 112 samples out of 120 were recognized properly with the rate of 93.33%. The vowels /u/ got the worst recognition rate. This refers to the inability of the NNs in distinguishing between /o/ and /u/ as 32 samples out of 120 were recognized wrongly.

Table 4.2: Confusion Matrix of SF55ms Vowel Signal Trained using 120 Hidden Neurons in MLP

	/a/	**/ə/**	**/e/**	**/i/**	**/o/**	**/u/**	**Accuracy (%)**
/a/	**112**	0	6	2	0	0	**93.33**
/ə/	2	**107**	4	5	0	2	**89.16**
/e/	7	3	**106**	0	2	2	**88.33**
/i/	0	6	0	**112**	0	2	**93.33**
/o/	7	4	3	1	**93**	12	**77.5**
/u/	1	6	4	1	32	**76**	**63.33**
Total Recognition Rate (%)							**84.16**

4.2.2 Multi-Frame Analysis

The highest recognition rates achieved for different multi-frame vowel signal of the three sets using MLP architecture are summarized in Table 4.3. The average rates of the three sets were computed for each frame size.

Table 4.3: Recognition Rates (%) of each Multi-Frame Data Set using MLP Architecture

Signal Length (ms)	**No. of Hidden Neurons**	**SET 1 (Recognition Rate %)**	**No. of Hidden Neurons**	**SET 2 (Recognition Rate %)**	**No. of Hidden Neurons**	**SET 3 (Recognition Rate %)**	**AVERAGE (%)**
30	140	79.72	60	80.83	100	83.47	81.34
40	60	78.88	40	80.55	160	83.88	81.10
50	140	79.3	140	82.63	60	83.88	81.94
60	40	79.72	120	81.25	120	82.77	81.25
70	120	79.44	120	82.22	40	83.05	81.57
80	80	78.05	200	80.97	180,200	82.91	80.64
90	140	79.44	160	81.38	180	83.47	81.43
100	120,140	79.16	200	78.47	80	83.05	80.23

Figure 4.3 illustrates the performance of each multi-frame data with the highest recognition rate using MLP network. The data of 50ms frame size got the highest recognition rate of 81.94%. Though, the lowest recognition rate obtained is 8o.23% related to the 100ms frame size. In general, there is a smooth fluctuation around 81% in the recognition rate of different multi-frame signal lengths using MLP.

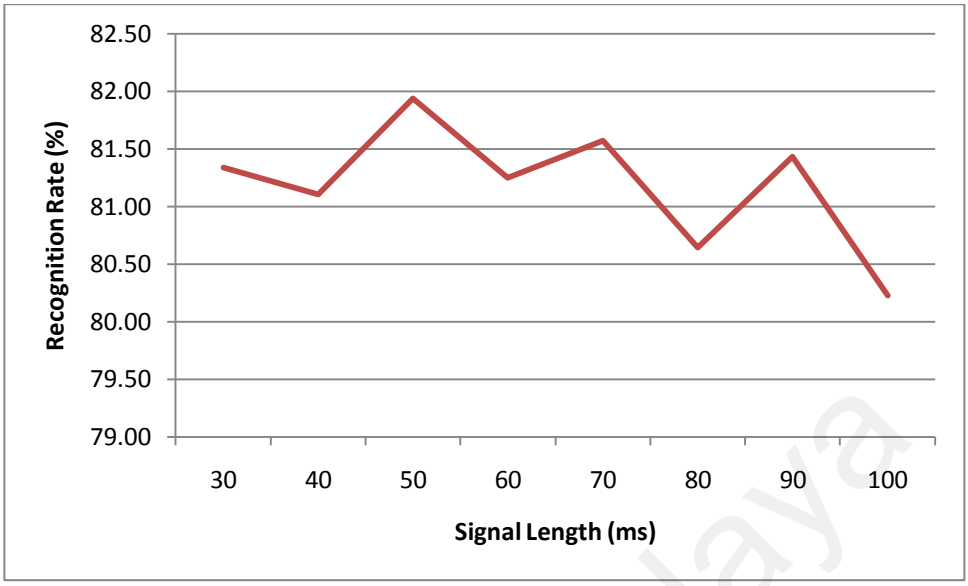

Figure 4.3: Recognition Rates (%) of each Multi-Frame Vowel Signal using MLP

Comparing the results of single-frame and multi-frame speech data indicates that single-frame data performed better than multi-frame data using MLP. This is due to the different qualification of the applied training functions. Levenberg-Marquardt (*trainlm*) algorithm was used for the single-frame data, which is the fastest training function in many cases to get less mean square errors. On the other hand, Gradient Descent Back-propagation with Adaptive Learning Rate algorithm (*traingdx*) was applied for the multi-frame analysis to cope with the memory deficiency.

Figure 4.4 shows different recognition rates of MF50ms at different number of hidden neurons. The best accuracy in this stage was obtained at 60 hidden neurons and the lowest rate was achieved when 180 neurons were used in the hidden layer. When the number of hidden neurons is less than 120, the recognition rates change between 79.20% and 89.25%. However, using more neurons in hidden layer leads to reduction in accuracy. This fact relates to the "overfitting" problem caused by the complex neural networks dealing with training data patterns and external test data.

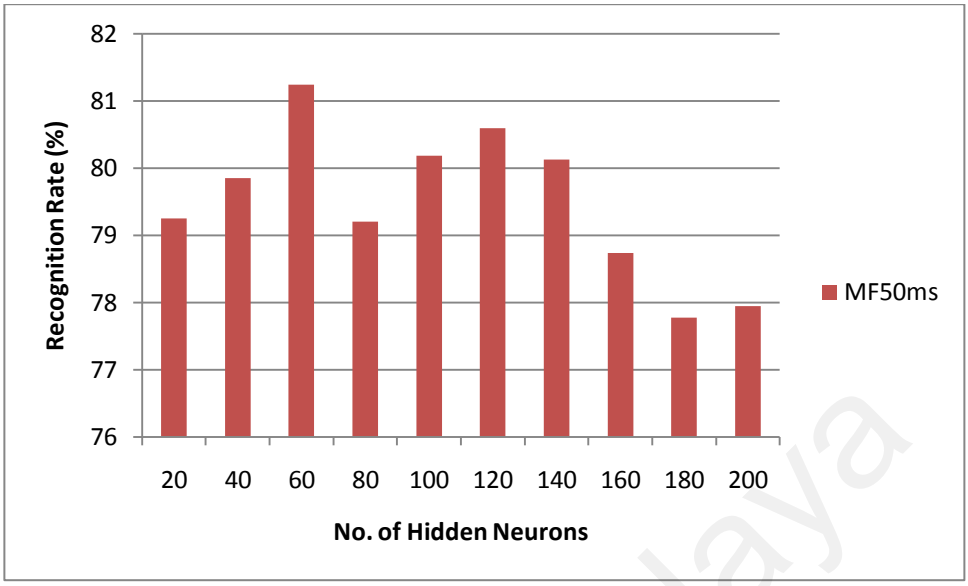

Figure 4.4: Recognition Rates (%) of MF50ms Vowel Signal at Different Number of Hidden Neurons using MLP

Table 4.4 shows the confusion matrix of 50ms multi-frame data trained with 60 hidden neurons. Vowel /i/ got the highest accuracy as 116 samples out of 120 were recognized properly with the rate of 96.66%. The vowel /o/ got the lowest rate at 64.16% as it was confused with /u/ in 38 samples out of 120.

It can be concluded that worst recognition rate in both single-frame and multi-frame analysis using MLP network relates to vowel /o/.

Table 4.4: Confusion Matrix of MF50ms Vowel Signal Trained using 60 Hidden Neurons in MLP

	/a/	**/ə/**	**/e/**	**/i/**	**/o/**	**/u/**	**Accuracy (%)**
/a/	**106**	0	9	0	5	0	**88.33**
/ə/	0	**102**	1	12	3	2	**85**
/e/	0	5	**99**	0	5	11	**82.5**
/i/	0	1	0	**116**	1	2	**96.66**
/o/	1	0	4	0	**77**	38	**64.16**
/u/	0	0	2	0	14	**104**	**86.66**
Total Recognition Rate (%)							**83.88**

4.3 RNN Results

4.3.1 Single-Frame Analysis

The highest recognition rates obtained for different single-frame vowel signal of the three sets using RNN architecture are summarized in Table 4.3. Also, the calculated average rates of the three sets are displayed for each frame size.

Table 4.5: Recognition Rates (%) of each Single-Frame Data Set using RNN Architecture

Signal Length (ms)	No. of Hidden Neurons	SET 1 (Recognition Rate %)	No. of Hidden Neurons	SET 2 (Recognition Rate %)	No. of Hidden Neurons	SET 3 (Recognition Rate %)	AVERAGE (%)
10	80	76.66	160	78.61	100	76.25	77.17
15	80	78.05	120	79.72	100	80.83	79.53
20	80	78.47	120	81.38	100	82.22	80.69
25	40	80.41	120,160	81.38	100	83.61	81.80
30	40	80.55	120	82.22	100	83.88	82.22
35	40	80.41	120	81.66	100	83.33	81.80
40	40	79.72	120	81.8	100	83.75	81.76
45	40	81.94	120	82.08	100	83.88	82.63
50	40	79.02	180	81.94	100	83.75	81.57
55	40	80.69	120	82.08	100	84.3	82.36
60	10	79.58	180	81.8	120	84.44	81.94
65	10	80.55	120	81.94	120	84.58	82.36
70	40	81.11	120	81.66	60	83.19	81.99

The highest recognition rates of each single-frame database trained by RNN are depicted in the following figure. The lowest and highest obtained recognition rates using RNN was 77.17% by single-frame 10ms and 82.63% by single-frame 45ms, respectively.

Therefore, it can be deduced that the worst frame size of speech data to recognize vowels is single-frame 10ms using MLP and RNN architectures.

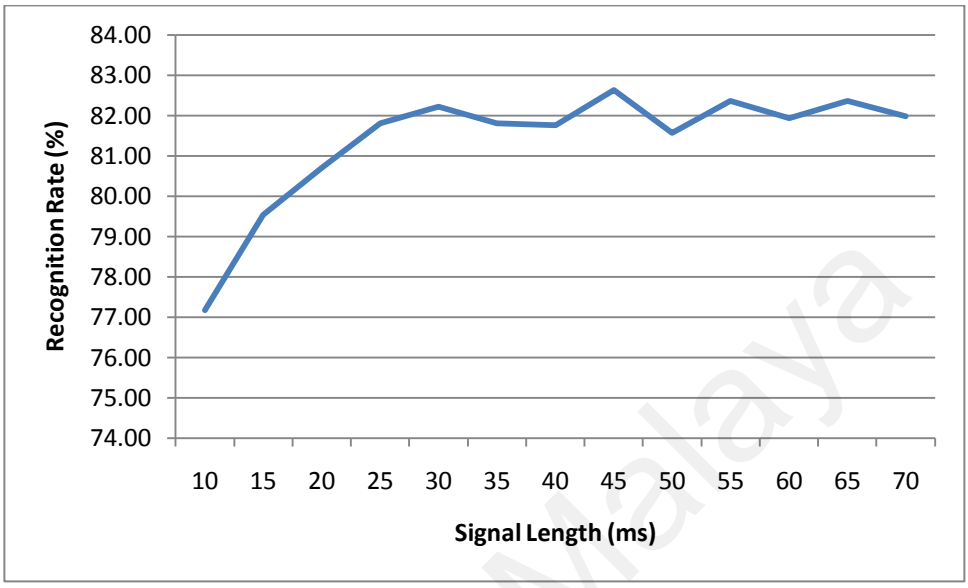

Figure 4.5: Recognition Rates (%) of each Single-Frame Vowel Signal using RNN

Different recognition rates for single-frame 45ms vowel signal trained with various neurons in hidden layer are illustrated in Figure 4.6. As shown in the chart, there are gentle changes in the rate around 80% when the number of hidden neurons ranges between 60 and 200. It can be concluded that Random Order Weight/Bias Learning Rules (*trainr*) algorithm is suitable for training the single-frame speech data using RNN. The maximum recognition rate was attained at 100 hidden neurons; whereas, the minimum rate was occurred when 20 hidden neurons were used to train the network.

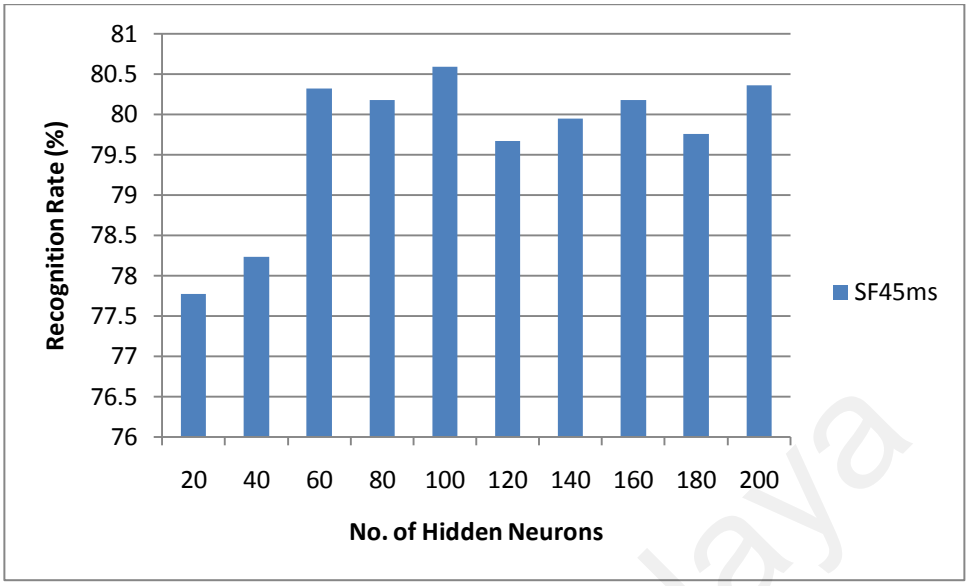

Figure 4.6: Recognition Rates (%) of SF45ms Vowel Signal at Different Number of Hidden Neurons using RNN

Table 4.6 shows the confusion matrix of 45ms single-frame data trained with 100 hidden neurons. The same as MLP, the best classification was achieved by the vowel /i/ with recognizing 118 samples out of 120 correctly and the recognition rate of 98.33%. However, 38 samples of /u/ out of 120 were recognized incorrectly due to the confusion between /o/ and /u/.

Table 4.6: Confusion Matrix of SF45ms Vowel Signal Trained using 100 Hidden Neurons in RNN

	/a/	**/ə/**	**/e/**	**/i/**	**/o/**	**/u/**	**Accuracy (%)**
/a/	**108**	0	6	0	6	0	**90**
/ə/	0	**101**	3	13	3	0	**84.16**
/e/	2	4	**98**	0	14	2	**81.66**
/i/	0	1	0	**118**	0	1	**98.33**
/o/	0	0	1	0	**105**	14	**87.5**
/u/	0	0	7	1	38	**74**	**61.66**
Total Recognition Rate (%)							**83.88**

4.3.2 Multi-Frame Analysis

Table 4.7 represents the highest recognition rates with the number of hidden neurons for the different signal lengths of the three sets using RNN architecture. The average rates of the three sets were calculated for each frame. It is obvious from the table that the highest average rate was obtained from multi-frame 100ms.

Table 4.7: Recognition Rates (%) of each Multi-Frame Data Set using RNN Architecture

Signal Length (ms)	No. of Hidden Neurons	SET 1 (Recognition Rate %)	No. of Hidden Neurons	SET 2 (Recognition Rate %)	No. of Hidden Neurons	SET 3 (Recognition Rate %)	AVERAGE (%)
30	40	80	120	82.08	100	84.02	82.03
40	160,200	79.44	40	82.36	80	84.58	82.13
50	40	81.66	200	82.5	100	84.72	82.96
60	200	80.97	40	81.94	160	84.02	82.31
70	200	81.25	40,120	81.52	160	85.13	82.63
80	180	80.97	40	81.8	160	83.05	81.94
90	60	80.27	200	81.8	200	84.16	82.08
100	120	81.38	80	82.91	80	85	83.10

The highest recognition rates of each multi-frame database trained by RNN are represented in the Figure 4.7. All the frame sizes, except MF80ms, have the accuracy rate above 82%. The maximum and minimum recognition rates were achieved 83.10% at 100ms frame size and 81.94% at 80ms frame size, respectively.

Figure 4.8 shows the different recognition rates of MF100ms frame size are shown at different number of hidden neurons. The highest rate happened when 80 hidden neurons were used to train the network.

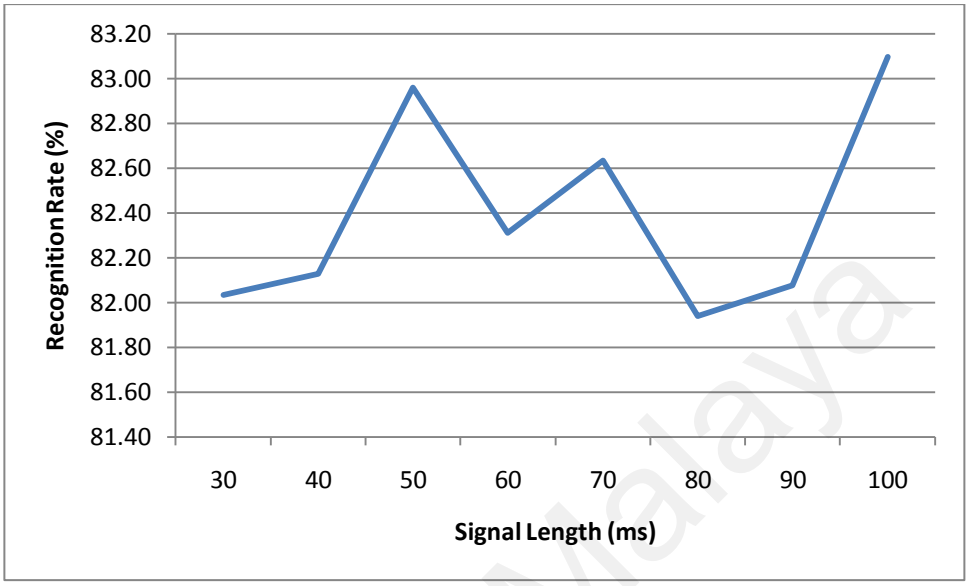

Figure 4.7: Recognition Rates (%) of each Multi-Frame Vowel Signal using RNN

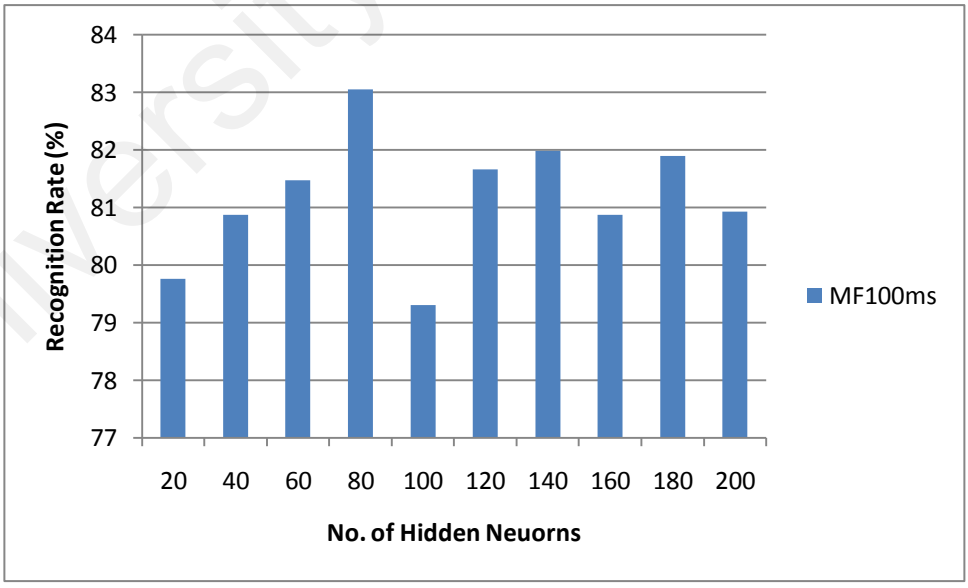

Figure 4.8: Recognition Rates (%) of MF100ms Vowel Signal at Different Number of Hidden Neurons using RNN

The confusion matrix of MF100ms vowel signal trained using 80 hidden neurons is illustrated in Table 4.8. The highest recognition rate was achieved for vowel /i/ where 116 samples out of 120 were recognized properly with the rate of 95.83%. The vowel /u/ got the lowest rate at 65.83% as it was confused with /o/ in 36 samples out of 120.

As a result, the lowest recognition rates for both single-frame and multi-frame signal analysis using RNN relates to the vowel /u/.

Table 4.8: Confusion Matrix of MF100ms Vowel Signal Trained using 80 Hidden Neurons in RNN

	/a/	**/ə/**	**/e/**	**/i/**	**/o/**	**/u/**	**Accuracy (%)**
/a/	**107**	0	6	0	7	0	**89.16**
/ə/	0	**105**	3	10	2	0	**87.5**
/e/	3	6	**98**	0	6	7	**81.66**
/i/	0	2	1	**115**	0	2	**95.83**
/o/	0	0	2	0	**108**	10	**90**
/u/	0	1	4	0	36	**79**	**65.83**
Total Recognition Rate (%)							**85**

4.4 Comparison between MLP and RNN on Malay Vowel Recognition

The comparative study on Malay vowel recognition between MLP and RNN is carried out in the aspects of different signal length and different Malay vowels.

4.4.1 Different Signal Length

Figures 4.9 and 4.10 show the recognition rates of the vowel signal in different lengths using MLP and RNN for single-frame and multi-frame analysis, respectively. Both types of network have better performance in longer signal lengths compared to short signal lengths in single-frame analysis. The recognition rate has increased rapidly until 30ms and then there is some fluctuations around 82% for MLP and around 83% for RNN.

On the other hand, MLP has achieved better result in shorter signal length than long signal length in multi-frame analysis. In contrast, RNN obtained the highest recognition rate in the longest signal length which is 100ms.

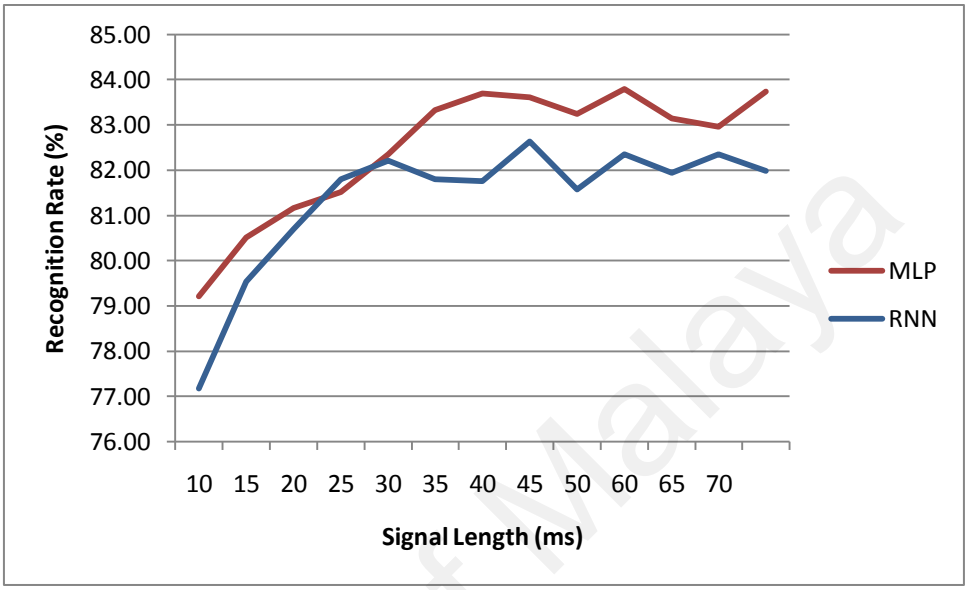

Figure 4.9: Comparison on Recognition Rates (%) of each Single-Frame Vowel Signal between MLP and RNN

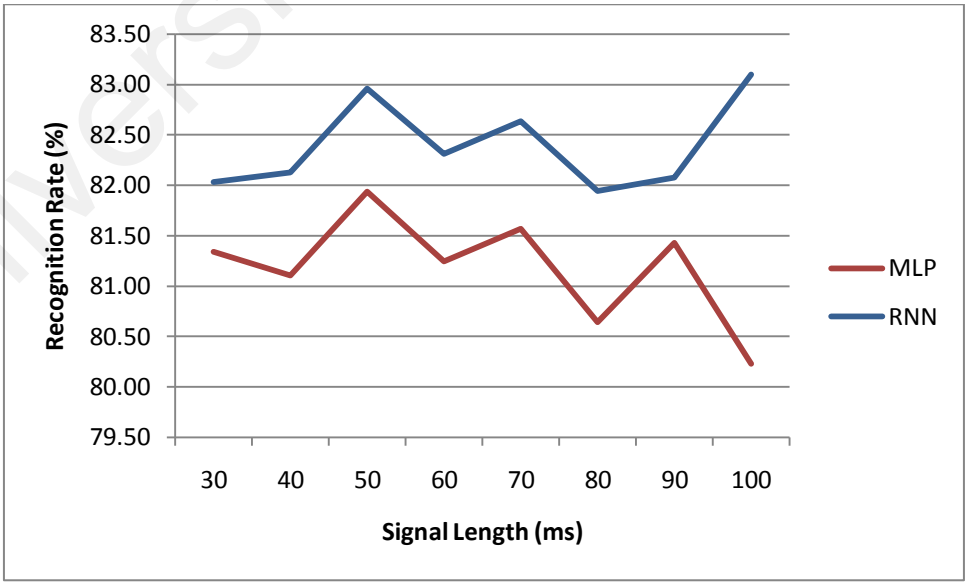

Figure 4.10: Comparison on Recognition Rates (%) of each Multi-Frame Vowel Signal between MLP and RNN

4.4.2 Vowel

The accuracy of each vowel is represented in Table 4.9 for the best signal frame of single-frame and multi-frame analysis using MLP and RNN. As mentioned before, the highest total recognition rate is achieved by single- frame vowel signal using MLP and multi-frame vowel signal using RNN.

Table 4.9: Comparison on Malay Vowel Recognition Rate (%) between MLP and RNN

Accuracy (%)	**Single-Frame**		**Multi-Frame**	
	MLP	**RNN**	**MLP**	**RNN**
/a/	93.33	90	88.33	98.16
/ə/	89.16	84.16	85	87.5
/e/	88.33	81.66	82.5	81.66
/i/	93.33	98.33	96.66	95.83
/o/	77.5	87.5	64.16	90
/u/	63.33	61.66	86.66	65.83
Total Recognition Rate (%)	84.16	83.8	83.33	85

4.5 Vowel Recognition Rate of Different Methods

According to the obtained result, the significant difference between MLP and RNN refers to the type of frame analysis. Higher total recognition rate was achieved by MLP in single-frame analysis compare to the multi frame, whereas, RNN showed better accuracy in mutli-frame analysis.

Table 4.10 summarizes the recognition rate and some of the features of this study and other recent studies on speech recognition. It addresses the matters of frame analysis, speaker type and accuracy of various speech classification methods on different database.

Table 4.10: Recent Studies on Speech Recognition

Method	Accuracy	Speaker Type	Frame Analysis	Database
ANN (MLP)	84.16%	Independent	Single-Frame	Malay Vowels
	83.88%	Independent	Multi-Frame	
ANN (RNN)	83.88%	Independent	Single-Frame	Malay Vowels
	85%	Independent	Multi-Frame	
ANN (Giurgiu, 1995)	96%	Independent	Single-Frame	Romanian Vowels
ANN (Feedforward) (Merks & Miles, 2005)	91.50%	Independent	Multi-Frame	English Vowels
SVM (Andrade Bresolin, Neto, & Alsina, 2007)	91.01%	Independent	Multi-Frame	Brazilian Vowels
	98.07%	Dependent		
SVM (based on Kernal Direct Discriminant Analysis) (Nazari et al., 2008)	93.90%	Independent	Multi-Frame	Persain Vowel
HMM (Damien, 2011)	81.70%	Independent	Multi-Frame	Arabic Vowel

The results obtained in this study are lower than those of other studies. This can be justified through the reasons such as number of vowels, methods of feature extraction and size of the feature vector, number of speakers, number of speech samples and sampling rate. All these aspects can have direct effect on the performance of ASR system. For instance, it has demonstrated by (Ssnderson & Paliwal, 1997) that a speech recognizer based on HMM can show a best performance at the sampling rate of 12kHz and feature vector size of 14 using LPC. Giurgiu (1995) achieved high ANN performance probably because he used just 4 speakers (2males and 2 females). Increasing the number of samples

may increase the error rate in training phase. Although, SVMs usually perform more accurate as classifiers in speech recognition, but large size of databases which needs to deal with huge number of training patterns has limited their application (Padrell-Sendra, Martin-Iglesias, & Diaz-de-Maria, 2006). Hence, this study has focused on neural networks.

So having a precise comparison between different classifiers and recognizers is subject to have same conditions for all methods.

Recognition rate, processing time and amount of computation in the applied algorithm are the factors that should be considered in choosing a suitable method in a ASR system.

5 CHAPTER V: CONCLUSIONS, IMPLICATIONS AND SUGGESTIONS FOR FUTURE WORK

5.1 Conclusions and Implications

The aim of this study was investigating the performance of MLP and RNN on the Malay vowel recognition. Different experiments were carried out to examine the features of networks in terms of signal length and number of hidden neurons.

The highest recognition rates obtained by MLP and RNN were 83.79% and 83.10%, respectively. The optimal performance was achieved in single-frame 55ms and multi-frame 50ms using MLP. In addition, single-frame 45ms and multi-frame 100ms got the maximum recognition rates compared to the other frames trained by RNN. In contrast, the lowest accuracy was related to the single-frame 10ms using both neural network types.

The results obtained from the research analysis output, which are depicted graphically and summarized in tables, indicates that the best recognition rates could be reached when the networks were trained with less than 120 hidden neurons. In other words, the performance of the network descends when more neurons are used in the hidden layer due to overfitting. Utilizing MLP led to greatest recognition rates for single-frame and multi-frame speech data when 120 and 60 hidden neurons were used, respectively. However, best results were attained for single-frame and multi-frame vowel signal when RNN was used with 100 and 80 hidden neurons, respectively. As a result, small number of hidden neurons is not recommended to train the neural network.

According to the processed *Confusion Matrices*, /i/ was the best vowel classified by MLP and RNN as it got the highest accuracy percentage. On the contrary, the vowels /o/ and /u/ got the lowest accuracy and were confused in classification.

In general, longer signal lengths performed better than short signal lengths. The considerable point is that the results of single-frame data using MLP was better than those in multi-frame, which refers to the training function. It means that Levenberg-Marquardt (*trainlm*) function used to train single-frame data is more effective than Gradient Descent Back-propagation with Adaptive Learning Rate algorithm (*traingdx*) applied to train multi-frame data.

5.2 Suggestions for Future Work

It is advised to implement other types of ANNs on the Malay children vowel signal in order to find out the architecture that can discriminate between /o/ and /u/ more accurately.

Additionally, it is expected that future efforts on other models of NNs such as unsupervised learning algorithms will improve the speech recognizers with better performance and more accurate results.

REFERENCES

Ahad, A., Fayyaz, A., & Mehmood, T. (2002). Speech recognition using multilayer perceptron. *IEEE Proceeding ISCON Students Conference, 1*, 103-109.

Ahmad, A. M., Ismail, S., & Samaon, D. F. (2004). Recurrent neural network with backpropagation through time for speech recognition. *IEEE International Symposium on Communications and Information Technology, 1*, 98-102.

Ala-Keturi, V. (2004). Speech recognition based on artificial neural networks. *Helsinki hnology Institute of Tec*.

Albesano, D., Gemello, R., & Mana, F. (1992). Word recognition with recurrent network automata. *International Joint Conferenc, IJCNN on Neural Networks, 2*, 308-313.

Al-Haddad, S., Samad, S., Hussain, A., Ishak, K., & Noor, A. (2009). Robust speech recognition using fusion techniques and adaptive filtering. *American Journal of Applied Sciences, 6*(2), 290-295.

Alsmadi, M. K. S., Omar, K. B., & Noah, S. A. (2009). Backpropagation algorithm: The best algorithm among the multi-layer perceptron algorithm. *Int J Comput Sci Netw Secur, 9*, 378-383.

Andrade Bresolin, A., Neto, A. D. D., & Alsina, P. J. (2007). Brazilian vowels recognition using a new hierarchical decision structure with wavelet packet and SVM. *IEEE International Conference on Acoustics, Speech and Signal Processing, 2*, 493-496.

Bourouba, E. H., Bedda, M., & Djemili, R. (2006). Isolated words recognition system based on hybrid approach DTW/GHMM. *Informatica, 30, 373-384.*

Bronzino, J. D. (2000). *The biomedical engineering handbook*. CRC Press.

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning, 20*(3), 273-297.

Carlson, R., & Glass, J. (1992). Vowel classification based on analysis-by-synthesis. *STL-QPSR, 33*(4), 017-027.

Damien, P. (2011). Visual speech recognition of Modern Classic Arabic language. *International Symposium on Humanities, Science & Engineering Research*, 50-55.

El Choubassi, M. M., El Khoury, H. E., Alagha, C. E. J., Skaf, J. A., & Al-Alaoui, M. A. (2003). Arabic speech recognition using recurrent neural networks. *Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology*, 543-547.

Faisal, T., Taib, M. N., & Ibrahim, F. (2010). Neural network diagnostic system for dengue patients risk classification. *Journal of Medical Systems*, 1-16.

Fu, S. W. K., Lee, C. H., & Clubb, O. L. (1996). Recognition of Cantonese finals using heuristic methodology. *3rd International Conference on Signal Processing, 1*, 761-764.

Gao, C., Chen, J., Zeng, J., Liu, X., & Sun, Y. (2009). A chaos-based iterated multistep predictor for blast furnace ironmaking process. *AIChE Journal, 55*(4), 947-962.

Ghaemmaghami, M. P., Razzazi, F., Sameti, H., Dabbaghchian, S., & BabaAli, B. (2009). Noise reduction algorithm for robust speech recognition using MLP neural network. *Asia-Pacific Conference on Computational Intelligence and Industrial Applications,1*, 377-380.

Giurgiu, M. (1995). On the use of neural networks for automatic vowel recognition. *International IEEE/IAS Conference on Industrial Automation and Control: Emerging Technologies*, 479-484.

Hao, R., & Ravi, S. (1995). Applying neural network to robust keyword spotting in speech recognition application. *IEEE International Conference on Neural Networks Proceedings, 2*, 2882-2886.

Huang, W., Lippmann, R., & Gold, B. (1988). A neural net approach to speech recognition. *Acoustics, Speech, and Signal Processing International Conference, 1*, 99-102.

Juneja, A., & Espy-Wilson, C. (2003). Speech segmentation using probabilistic phonetic feature hierarchy and support vector machines. *Proceedings of the International Joint Conference on Neural Networks, 1*, 675-679.

Krogh, A., & Vedelsby, J. (1995). Neural network ensembles, cross validation and active learning. *Advances in Neural Information Processing Systems*, 231–238.

Kumar, T. L., Kumar, T. K., & Rajan, K. S. (2009). Speech recognition using neural networks. *International Conference on Signal Processing Systems*, 248-252.

Lee, S., & Iverson, G. K. (2009). Vowel development in English and Korean: Similarities and differences in linguistic and non-linguistic factors. *Speech Communication, 51*(8), 684-694.

Lee, S., Potamianos, A., & Narayanan, S. (1999). Acoustics of children's speech: Developmental changes of temporal and spectral parameters. *The Journal of the Acoustical Society of America, 105*, 1455.

Leonida, M. M. G. (2000). A time-delayed neural network approach to the prediction of the hot metal temperature in a blast furnace. *Massachusetts Institute of Technology*.

Liu, Y., Lee, Y. C., Chen, H. H., & Sun, G. Z. (1992). Speech recognition using dynamic time warping with neural network trained templates. *International Joint Conference on Neural Networks, 2*, 326-331.

Makhoul, J. (1975). Linear prediction: A tutorial review. *Proceedings of the IEEE, 63*(4), 561-580.

Merkx, P., & Miles, J. (2005). Automatic vowel classification in speech. *An Artificial Neural Network Approach Using Cepstral Feature Analysis*, 1-14.

Nazari, M., Sayadiyan, A., & Valiollahzadeh, S. M. (2008). Speaker-independent vowel recognition in Persian speech. *3rd International Conference on Information and Communication Technologies: From Theory to Applications*, 1-5.

Oglesby, J., & Mason, J. S. (1990). Optimisation of neural models for speaker identification. *International Conference on Acoustics, Speech, and Signal Processing, 1*, 261-264.

Paliwal, K. K. (1991). A time-derivative neural net architecture-an alternative to the time-delay neural net architecture. *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, 367-375.

Padrell-Sendra, J., Martin-Iglesias, D., & Diaz-de-Maria, F. (2006). Support vector machines for continuous speech recognition. *14th European Signal Processing Conference, 160*, 118-122.

Rabiner, L., Levinson, S., Rosenberg, A., & Wilpon, J. (1979). Speaker-independent recognition of isolated words using clustering techniques. *IEEE Transactions on Acoustics, Speech and Signal Processing, 27*(4), 336-349.

Renals, S., McKelvie, D., & McInnes, F. (1991). A comparative study of continuous speech recognition using neural networks and hidden Markov models. *International Conference on Acoustics, Speech, and Signal Processing, 1*, 369-379.

Salam, M. S. H., Mohamad, D., & Salleh, S. H. S. (2001). Neural network speaker dependent isolated Malay speech recognition system: handcrafted vs genetic algorithm. *Sixth International Symposium on Signal Processing and its Applications, 2*, 731-734.

Sanderson, C., & Paliwal, K. K. (1997). Effect of different sampling rates and feature vector sizes on speech recognition performance. I*EEE Region 10 Annual Conference on Speech and Image Technologies for Computing and Telecommunications. 1*, 161-164.

Shahrul, A. M. Y., Siraj, F., Yaacob, S., Paulraj, M. P., & Nazri, A. (2010). Improved Malay Vowel Feature Extraction Method Based on First and Second Formants. *Second International Conference on Computational Intelligence, Modelling and Simulation (CIMSiM)*, 339-344.

Singh Gill, A. (2008). A novel low complexity speech recognition approach. *IEEE International Joint Conference on Neural Networks*, 2710-2714.

Sorsa, T., Koivo, H. N., & Koivisto, H. (1991). Neural networks in process fault diagnosis. *IEEE Transactions on Systems, Man and Cybernetics, 21*(4), 815-825.

Thasleema, T. M., Kabeer, V., & Narayanan, N. K. (2007). Malayalam vowel recognition based on linear predictive coding parameters and k-NN algorithm. *International Conference on Computational Intelligence and Multimedia Applications, 2,* 361-365.

Ting, H. N., & Mark, K. M. (2008). Speaker-dependent Malay vowel recognition for a child with articulation disorder using multi-layer perceptron. *4th International Conference on Biomedical Engineering, 21*(1), 238-241.

Ting, H. N., & Yunus, J. (2004). Speaker-independent Malay vowel recognition of children using multi-layer perceptron. *IEEE Region 10 Conference, 1,* 68-71.

Ting, H. N., Yunus, J., Salleh, S. H. S., & Cheah, E. L. (2001). Malay syllable recognition based on multilayer perceptron and dynamic time warping. *Sixth International Symposium on Signal Processing and its Applications, 2,* 743-744.

Wan, E. A. (1990). Neural network classification: a Bayesian interpretation. *Neural Networks, IEEE Transactions on, 1*(4), 303-305.

Wang, X., & Zheng, B. (1998). A new neural network oriented speech recognition. *International Conference on Communication Technology Proceedings, 2,* 4pp.

Widrow, B., Winter, R. G., & Baxter, R. A. (1988). Layered neural nets for pattern recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing, 36*(7), 1109-1118.

Wilpon, J., & Rabiner, L. (1985). A modified K-means clustering algorithm for use in isolated work recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing, 33*(3), 587-594.

Xian, T. (2009). Hybrid hidden Markov model and artificial neural network for automatic speech recognition. *Pacific-Asia Conference on Communications and Systems, 682-685*.

Zebulum, R. S., Vellasco, M., Perelmuter, G., & Pacheco, M. A. (1996). A comparison of different spectral analysis models for speech recognition using neural networks. *IEEE 39th Midwest Symposium on Circuits and Systems, 3,* 1428-1431.

Zhao, L., & Han, Z. (2010). Speech recognition system based on integrating feature and HMM. I*nternational Conference on Measuring Technology and Mechatronics Automation, 3,* 449-452.

Appendix A: Matlab® Programming Code

```
% 1) Import Data from C++ to Matlab (Training Data Set)

files = dir('*.cep');
for i=1:1440
eval(['load ' files(i). name ' -ascii']);
end


% 2) Create Input Matrix

input = [];
for i = 1:240;
    eval (['input = [ input a' num2str(i) ' ]; ' ]);
    eval (['input = [ input ae' num2str(i) ' ]; ' ]);
    eval (['input = [ input e' num2str(i) ' ]; ' ]);
    eval (['input = [ input i' num2str(i) ' ]; ' ]);
    eval (['input = [ input o' num2str(i) ' ]; ' ]);
    eval (['input = [ input u' num2str(i) ' ]; ' ]);
end


% 3) Import Data from C++ to Matlab (Testing Data Set)

files = dir('*.cep');
for i=1:720
eval(['load ' files(i). name ' -ascii']);
end

% 4) Create Test Matrix

testing = [];
for i = 1:120;
    eval (['testing = [ testing a' num2str(i) ' ]; ' ]);
    eval (['testing = [ testing ae' num2str(i) ' ]; ' ]);
    eval (['testing = [ testing e' num2str(i) ' ]; ' ]);
    eval (['testing = [ testing i' num2str(i) ' ]; ' ]);
    eval (['testing = [ testing o' num2str(i) ' ]; ' ]);
    eval (['testing = [ testing u' num2str(i) ' ]; ' ]);
end


% 5) Create Target Matrix

target1 = [.9 .1 .1 .1 .1 .1]';
target2 = [.1 .9 .1 .1 .1 .1]';
target3 = [.1 .1 .9 .1 .1 .1]';
target4 = [.1 .1 .1 .9 .1 .1]';
```

```
target5 = [.1 .1 .1 .1 .9 .1]';
target6 = [.1 .1 .1 .1 .1 .9]';
A = [target1 target2 target3 target4 target5 target6];
target = repmat (A,1,240);


% 6) Create Network

    % MLP Network Trained using trainlm

    numHiddenNeurons = 10; % Adjustable
    net =
    newff(input,target,numHiddenNeurons,{'tansig','purelin'}
    ,'trainlm','learngdm','mse');
    net.trainparam.show = 25;
    net.trainparam.epoches = 1000;
    net.trainparam.time = inf;
    net.trainparam.goal = 0;
    net.trainparam.max_fail = 6;
    net.trainparam.mem_reduc = 1;
    net.trainparam.min_grad = 1e-010;
    net.trainparam.mu = 0.001;
    net.trainparam.mu_dec = 0.1;
    net.trainparam.mu_inc = 10;
    net.trainparam.mu_max = 10000000000;


    % MLP Network Trained using traingdx

    numHiddenNeurons = 10; % Adjustable
    net =
    newff(input,target,numHiddenNeurons,{'tansig','purelin'},'
    traingdx','learngdm','mse');
    net.trainparam.show = 25;
    net.trainparam.epoches = 1000;
    net.trainparam.time = inf;
    net.trainparam.goal = 0;
    net.trainparam.max_fail = 6;
    net.trainparam.min_grad = 1e-010;
    net.trainparam.Ir= 0.01
    net.trainparam.Ir.inc= 1.05
    net.trainparam.max_pref_inc= 1.04
    net.trainparam.mc= 0.9


    % RNN Network Trained using trainr

    numHiddenNeurons = 10; % Adjustable
```

```
net =
newlrn(input,target,numHiddenNeurons,{'tansig','purelin'},
'trainr','learngdm','mse');
net.trainparam.show = 25;
net.trainparam.epoches = 100;
net.trainparam.goal = 0;
net.trainparam.time = inf;

% 7) Train and Simulate Network

[net,tr] = train(net,input,target);
output = sim(net,testing);


% 8) Output Processing

    % Getting the Maximum Value of each Column

    y=[];
    for i=1:720
    y(i)= max(output(:,i));
    end


    % Equalling the Maximum Value of each Column to 1

    for i=1:6
    for j=1:720
    if output(i,j)==y(j);
    output(i,j)= 1;
    end
    end
    end


    % Making the other Values Zeros

    for i=1:6
    for j=1:720
    if output(i,j)==1;
    else
    output(i,j)=0;
    end
    end
    end


    % Creating the 1D Output

    o=[];
```

```
if output(:,i)==[1;0;0;0;0;0];
o=[o 1];
else if output(:,i)==[0;1;0;0;0;0];
o=[o 2];
else if output(:,i)==[0;0;1;0;0;0];
o=[o 3];
else if output(:,i)==[0;0;0;1;0;0];
o=[o 4];
else if output(:,i)==[0;0;0;0;1;0];
o=[o 5];
else if output(:,i)==[0;0;0;0;0;1];
o=[o 6];
end
end
end
end
end
end
end
end


% Creating the 1D Target
y=[1 2 3 4 5 6];
t=[];
for i=1:120
t=[t y];
end


% Getting the Confusion Matrix
cm= confusionmat(t,o)


% Find out the Recognition Rate
A= sum (diag(cm))
ac= A/720*100
```

Appendix B: Recognition Rates' Results

Table B.1: Recognition rates using MLP neural network for single-frames and multi-frames of different signal lengths with different number of hidden neurons (Set 1)

Recognition Rate (%) SET 1	No. of Hidden Neurons										
	10	**20**	**40**	**60**	**80**	**100**	**120**	**140**	**160**	**180**	**200**
SF10ms	75.41	74.16	73.74	77.63	77.5	72.5	80.27	71.94	77.77	66.38	66.8
SF15ms	75.27	78.19	76.94	62.08	77.77	66.94	77.77	77.22	60.41	77.77	81.11
SF20ms	71.8	77.36	72.63	69.58	78.61	77.78	81.44	78.33	78.19	77.63	80.94
SF25ms	75.55	79.58	75.27	75.55	81.25	67.63	81.38	77.78	77.5	78.19	80.69
SF30ms	75.83	78.75	74.02	67.91	81.66	81.25	81.8	78.05	77.22	80	81.38
SF35ms	76.25	81.11	74.3	75.13	80.55	73.61	82.77	78.05	59.86	72.22	80.97
SF40ms	77.08	81.94	75.27	74.3	80.13	78.47	82.63	77.5	78.33	79.72	80.83
SF45ms	78.19	77.5	76.11	75.97	81.38	67.77	83.33	77.63	77.77	79.02	80.83
SF50ms	76.66	79.3	76.94	75.13	80	66.8	83.19	77.77	78.33	83.19	81.66
SF55ms	72.63	76.8	67.36	73.75	82.08	65.69	84.16	77.91	77.78	80.69	81.38
SF60ms	75.83	78.88	70.41	76.52	81.38	79.58	83.33	78.19	78.47	78.05	81.66
SF65ms	73.61	77.63	67.63	74.02	81.52	68.47	82.08	77.91	79.3	78.47	81.11
SF70ms	77.36	79.02	69.58	73.88	81.38	70.83	82.08	77.36	79.16	78.61	81.66
MF30ms	76.52	77.91	77.77	78.33	77.63	79.44	77.91	79.72	77.22	77.5	79.58
MF40ms	77.5	78.33	78.19	78.88	77.36	77.5	75.97	77.22	77.5	78.61	77.36
MF50ms	75	76.94	77.91	78.88	75.41	78.05	79.16	79.3	75.83	71.11	70.41
MF60ms	77.5	76.25	79.72	78.75	77.77	79.02	78.75	78.05	78.47	77.77	72.5
MF70ms	77.08	78.75	77.36	79.3	77.63	75.83	79.44	75.83	69.83	78.05	76.94
MF80ms	75.41	77.08	77.77	76.38	78.05	72.22	75.97	77.22	72.08	70.27	72.63
MF90ms	71.52	79.16	76.8	79.16	78.05	76.11	70.55	79.44	78.19	77.77	76.8
MF100ms	71.52	77.63	78.33	74.02	78.05	77.08	79.16	79.16	78.05	70.97	69.44

Table B.2: Recognition rates using MLP neural network for single-frames and multi-frames of different signal lengths with different number of hidden neurons (Set 2)

Recognition Rate (%) SET 2	No. of Hidden Neurons										
	10	**20**	**40**	**60**	**80**	**100**	**120**	**140**	**160**	**180**	**200**
SF10ms	75.13	75.97	71.66	66.94	75.55	76.52	79.3	76.25	77.63	79.86	75.83
SF15ms	73.05	77.77	79.58	74.02	79.44	71.8	80.41	78.75	79.44	81.25	79.16
SF20ms	77.36	79.3	75.97	80.69	77.77	73.33	81.38	79.44	80.83	78.75	80
SF25ms	76.94	82.22	76.38	82.36	82.5	74.86	81.8	79.72	81.52	77.91	82.91
SF30ms	70.55	81.66	79.16	81.94	83.05	75.97	83.19	80.41	81.66	82.5	83.61
SF35ms	79.44	81.94	79.02	82.08	83.47	81.8	81.83	80.83	81.8	83.61	83.61
SF40ms	79.44	81.52	76.66	76.38	83.88	81.11	81.52	80.41	81.25	84.3	83.33
SF45ms	76.25	81.38	72.91	81.32	81.94	73.33	82.22	80.69	81.92	80..41	83.19
SF50ms	79.3	81.52	73.33	81.38	82.5	80.97	82.77	80.27	81.94	82.77	81.94
SF55ms	77.77	83.47	75.27	65	81.52	81.5	80.97	80	81.11	83.61	81.8
SF60ms	67.91	80.41	79.58	69.16	82.22	81.52	82.08	80.13	80.69	70.83	81.66
SF65ms	79.72	80.13	77.77	69.58	82.08	81.25	82.91	80	81.94	82.36	82.22
SF70ms	75.55	84.02	78.05	76.94	82.91	76.94	82.77	79.72	81.94	82.63	81.8
MF30ms	79.44	78.33	79.86	80.83	78.05	80.41	80.41	79.58	80	78.33	77.5
MF40ms	78.19	77.91	80.55	77.63	80.27	79.44	78.33	80.27	72.77	80.13	79.44
MF50ms	78.61	80.55	82.22	80.97	80.27	79.02	80.27	82.63	78.05	80.41	80.27
MF60ms	77.36	79.3	78.19	78.47	78.33	78.88	81.25	77.91	73.19	80	78.33
MF70ms	77.22	79.72	80.27	80.69	80	80.69	82.22	81.11	81.66	81.38	80.55
MF80ms	78.19	76.36	79.3	80	80.13	71.8	77.77	71.25	70.55	70	80.97
MF90ms	75.69	75.97	79.44	77.36	79.44	80.41	78.19	77.63	81.38	72.08	70.55
MF100ms	76.52	77.5	77.91	78.33	70.27	78.47	77.91	71.11	72.36	73.19	78.19

Table B.: Recognition rates using MLP neural network for single-frames and multi-frames of different signal lengths with different number of hidden neurons (Set 3)

Recognition Rate (%) SET 3	No. of Hidden Neurons										
	10	**20**	**40**	**60**	**80**	**100**	**120**	**140**	**160**	**180**	**200**
SF10ms	72.63	71.11	67.22	62.91	74.44	76.38	77.5	74.44	76.25	75.69	76.11
SF15ms	70.69	78.75	71.94	77.91	79.02	78.47	79.16	76.25	77.97	76.11	79.02
SF20ms	76.94	78.05	78.05	79.44	78.33	79.02	79.86	76.94	80	79.16	80.69
SF25ms	75.27	81.25	72.91	77.63	81.38	79.02	80.97	78.47	80.97	77.5	79.44
SF30ms	74.02	81.66	78.47	81.25	80.69	78.61	81.11	79.44	81.5	77.91	81.52
SF35ms	78.61	81.38	82.63	77.36	83.61	80.83	82.22	80.97	82.08	79.3	80.55
SF40ms	78.75	82.5	79.16	79.3	84.16	82.5	83.19	80	82.77	83.61	80.83
SF45ms	76.38	83.61	77.22	74.16	83.47	80.27	81.66	80.55	82.63	84.3	81.38
SF50ms	79.58	83.75	76.38	78.19	83.05	82.19	83.19	80.97	82.77	82.08	82.63
SF55ms	80.13	81.66	74.58	68.61	83.47	81.52	83.61	81.66	82.77	83.59	80.27
SF60ms	81.8	80.97	76.11	83.33	82.22	82.91	83.88	81.66	83.33	83.75	83.05
SF65ms	81.66	80.13	80.55	82.22	82.36	82.5	83.89	81.66	82.77	83.75	83.33
SF70ms	82.36	85.13	79.3	82.36	82.77	80.83	83.05	81.8	83.33	83.47	82.91
MF30ms	79.02	80.69	79.86	80.83	80.41	83.47	81.94	81.11	80.41	81.66	81.94
MF40ms	79.58	80.69	82.5	82.91	82.5	81.25	82.77	79.44	83.88	83.61	79.86
MF50ms	78.88	80.27	79.44	83.88	81.94	83.47	82.36	78.47	82.36	81.8	83.19
MF60ms	80.55	81.66	82.22	80.97	80.97	81.8	82.77	73.88	82.36	82.22	75.13
MF70ms	80.83	82.77	83.05	81.8	80.83	81.52	81.66	81.8	82.22	82.77	81.66
MF80ms	79.16	82.5	80	80.55	80.97	73.61	79.58	75.13	82.63	82.91	82.91
MF90ms	72.63	81.52	81.25	81.52	73.75	83.05	80.83	74.44	75.13	83.47	82.91
MF100ms	77.63	78.47	80.83	81.66	83.05	73.61	79.86	78.05	72.5	80.27	74.3

Table B.4: Recognition rates using RNN for single-frames and multi-frames of different signal lengths with different number of hidden neurons (Set 1)

Recogntion Rate (%) SET 1	No. of Hidden Neurons										
	10	**20**	**40**	**60**	**80**	**100**	**120**	**140**	**160**	**180**	**200**
SF10ms	65.00	69.16	74.16	75.55	76.66	75.13	73.61	74.58	75.13	74.72	74.16
SF15ms	75.27	61.25	75.83	76.80	78.05	76.66	75.55	77.50	77.63	76.80	76.38
SF20ms	66.67	75.13	76.11	77.63	78.47	77.36	76.66	77.50	78.33	76.52	77.63
SF25ms	77.63	71.80	80.41	79.02	79.16	78.19	77.22	78.61	78.19	76.94	77.63
SF30ms	79.02	76.67	80.55	78.05	78.47	77.91	76.66	78.88	78.05	76.80	77.77
SF35ms	62.22	62.91	80.41	77.77	76.25	77.08	77.22	77.91	78.05	77.36	77.63
SF40ms	77.08	70.97	79.72	77.91	78.05	79.02	76.80	78.05	77.63	77.36	77.77
SF45ms	78.47	75.27	81.94	78.33	77.91	79.16	77.08	78.33	77.91	77.08	77.77
SF50ms	78.05	77.91	79.02	78.61	78.33	78.75	77.77	78.88	77.91	77.79	77.91
SF55ms	79.02	73.47	80.69	78.61	78.50	78.61	77.91	78.61	78.47	77.91	77.63
SF60ms	79.58	68.61	76.38	78.75	78.61	78.75	77.91	78.75	78.19	77.91	77.77
SF65ms	80.55	72.63	70.83	78.75	78.81	78.19	77.50	78.47	78.61	77.22	77.63
SF70ms	79.58	75.97	81.11	79.02	78.47	78.19	77.91	78.33	78.33	77.08	77.50
MF30ms	79.30	76.66	80.00	78.47	77.50	79.44	78.75	76.94	78.47	79.44	78.05
MF40ms	72.63	65.69	78.75	78.61	78.75	79.02	78.47	79.02	79.44	77.08	79.44
MF50ms	76.66	77.50	81.66	78.33	78.33	79.72	80.55	81.11	79.58	80.83	79.72
MF60ms	63.19	73.47	79.16	77.91	78.47	77.77	78.05	80.00	79.16	80.41	80.97
MF70ms	75.55	77.50	77.50	81.11	78.88	80.27	78.19	79.72	79.58	80.55	81.25
MF80ms	75.00	77.08	79.58	80.27	80.00	78.19	79.86	79.30	79.86	80.97	80.27
MF90ms	76.38	77.91	78.88	80.27	78.19	77.36	79.72	79.72	78.88	79.44	79.02
MF100ms	77.08	78.47	80.13	80.41	81.25	79.02	81.38	80.27	78.88	80.13	80.00

Table B.5: Recognition rates using RNN for single-frames and multi-frames of different signal lengths with different number of hidden neurons (Set 2)

Recognition Rate (%) SET 2	No. of Hidden Neurons										
	10	**20**	**40**	**60**	**80**	**100**	**120**	**140**	**160**	**180**	**200**
SF10ms	65.83	71.8	76.66	76.8	77.5	72.91	77.5	72.91	78.61	76.52	77.5
SF15ms	78.05	64.3	79.02	77.77	79.44	76.8	79.72	78.88	79.3	77.77	78.88
SF20ms	79.72	71.66	80.27	78.88	79.58	78.19	81.38	80	80.41	80	80.13
SF25ms	70.41	73.33	75	79.72	80.13	79.44	81.38	79.86	81.38	81.25	80.97
SF30ms	70.83	66.38	77.5	80	80.69	78.75	82.22	79.86	81.11	79.22	80.69
SF35ms	70.83	78.42	75.97	80	80.97	78.47	81.66	80.41	81.11	81.11	80.69
SF40ms	77.5	76.66	75.55	80	81.38	78.47	81.8	80.83	81.52	81.52	80.69
SF45ms	71.25	76.66	76.66	80	81.25	78.75	82.08	80.69	81.38	81.8	80.69
SF50ms	71.11	75.13	75.97	80.83	80.69	78.47	81.8	80.41	79.72	81.94	81.25
SF55ms	70.27	78.47	79.72	80.69	80.83	78.88	82.08	80.69	80.83	81.66	81.25
SF60ms	71.38	78.75	78.33	79.86	81.11	79.02	80.97	80.83	80.83	81.8	80.97
SF65ms	70.83	78.47	80.69	79.86	80.83	79.3	81.94	80.55	80.55	81.8	81.52
SF70ms	70.97	75.41	78.19	79.86	80.83	79.44	81.66	80.13	80.55	81.25	81.11
MF30ms	77.22	73.61	78.33	80.27	80.27	81.52	82.08	79.44	80.97	80.97	80.83
MF40ms	80.13	77.22	82.36	79.02	79.02	79.3	79.86	81.52	80.83	82.22	81.66
MF50ms	80.55	78.88	79.44	82.36	79.16	79.86	78.47	80.13	81.38	82.22	82.5
MF60ms	79.3	80.41	81.94	81.66	81.66	80.83	81.38	80.83	79.3	79.58	81.38
MF70ms	78.88	80.13	81.52	81.38	80.97	77.77	81.52	78.61	78.47	79.02	80
MF80ms	78.75	79.02	81.8	81.25	81.11	80.97	77.91	81.38	81.38	80.69	79.86
MF90ms	75.83	80.27	77.08	80.13	81.52	81.25	80.69	82.5	79.86	81.52	81.8
MF100ms	79.44	79.3	81.11	81.52	82.91	79.16	81.52	81.94	82.5	82.22	80

Table B.6: Recognition rates using RNN for single-frames and multi-frames of different signal lengths with different number of hidden neurons (Set 3)

Recognition Rate (%) Set 3	No. of Hidden Neurons										
	10	**20**	**40**	**60**	**80**	**100**	**120**	**140**	**160**	**180**	**200**
SF10ms	75.41	71.38	71.66	74.16	73.33	76.25	73.19	72.91	74.72	74.16	71.94
SF15ms	65.55	79.02	65.55	77.91	76.8	80.83	75.27	75.97	78.47	76.52	76.8
SF20ms	77.63	66.11	78.33	79.44	77.08	82.22	76.8	76.8	79.44	78.05	78.05
SF25ms	79.02	76.94	79.3	80.13	76.25	83.61	78.47	78.75	80.55	78.33	79.3
SF30ms	75.13	71.94	73.05	81.8	77.5	83.88	78.19	79.3	81.25	79.02	80
SF35ms	79.44	77.63	76.11	83.05	78.75	83.33	79.3	79.72	81.38	79.86	81.38
SF40ms	78.19	78.33	78.88	82.77	79.44	83.75	79.3	80	80.55	80.97	82.22
SF45ms	81.11	81.38	76.11	82.63	81.38	83.88	79.86	80.83	81.25	80.41	82.63
SF50ms	82.63	64.86	79.44	83.05	80	83.75	80.27	82.77	81.66	80.69	82.63
SF55ms	82.22	67.08	78.61	82.77	80.27	84.3	80	81.25	81.66	80.83	82.77
SF60ms	81.66	80.97	81.8	82.91	79.72	84.44	80.27	80.97	81.66	80.97	83.05
SF65ms	81.11	82.36	78.33	82.77	79.86	84.58	80.27	81.25	81.38	81.38	83.19
SF70ms	80.83	70.41	78.05	83.19	79.72	80.27	80.13	80.83	81.11	80.41	82.77
MF30ms	75.13	71.94	73.05	79.02	79.3	84.02	81.11	81.8	81.38	80.27	83.33
MF40ms	78.61	69.86	81.38	81.66	84.58	79.04	83.33	80.55	81.66	80.27	79.86
MF50ms	78.19	82.08	83.33	83.61	82.77	84.72	81.52	81.94	83.19	82.36	81.8
MF60ms	80	80.55	82.08	82.22	82.77	83.05	79.3	81.94	84.02	83.75	83.61
MF70ms	75.69	80	80.69	82.5	77.63	83.47	79.16	81.25	85.13	82.77	83.88
MF80ms	79.02	82.91	79.16	82.08	81.66	81.66	80.83	80.97	83.05	77.63	82.22
MF90ms	80.69	78.61	82.08	80.55	82.08	79.16	80.27	82.63	80.55	80.41	84.16
MF100ms	80.83	81.52	81.38	82.5	85	79.72	82.08	83.75	81.25	83.33	82.77