Towards The Development of Computer Aided Speech Therapy Tool in Arabic Language Using Artificial Intelligence

A dissertation submitted in partial fulfilment of the requirements for the degree of Bachelor of Science (Honours) in Computing

by
Khaled Tawfik

Department of Computing & Information Systems
Cardiff School of Management
Cardiff Metropolitan University

April 2016
Declaration

I hereby declare that this dissertation entitled “Towards The Development of Computer Aided Speech Therapy Tool in Arabic Language Using Artificial Intelligence” is entirely my own work, and it has never been submitted nor is it currently being submitted for any other degree.

Candidate: Khaled Seddik Abdel-Salam Tawfik

Signature: 

Date: 

Supervisor: Ambikesh Jayal

Signature: 

Date: 

I
Abstract

Speech disorders affecting children make their speech unclear, leading to ineffective communication and subsequent psychological problems. Articulation disorders are the most prevalent type; they are divided into four categories: substitution, omission, addition, and distortion. Interactive automatic speech monitoring tools offer a practical, adaptive and cost-effective alternative to face-to-face intervention sessions for speech therapy, however none do exist for Arabic language.

Arabic is a complicated language with a variety of different dialects. It consists of 27 consonant phonemes with a wide range of places of articulation that span the whole vocal tract from lips to glottis, in addition to eight vowel phonemes.

This dissertation is a step closer to an automatic Arabic speech therapy tool. It mainly investigates the possible features suitable for the diagnosis phase and the corresponding classification accuracy. It describes the process of detecting Arabic speech articulation disorder using the dataset found online of the letter (ر/r/).

The feature extraction method used for is a widely known for its performance in different speech analysis application; Mel-frequency Cepstrum Coefficients using Matlab application. For the final decision, the classification phase, three models were examined: Support Vector Machine (SVM), Artificial Neural Network (ANN). and K-nearest neighbour (KNN) using Weka software. The system attained overall accuracies of 85.25%, 64.18% and 74.99% for KNN, SVM and ANN respectively.
Acknowledgment

I respectfully grab this opportunity to acknowledge many people who deserve special mentions for their varied contributions in assorted ways that helped me during my research and the making of this thesis.

I am extremely grateful to Prof. Dr. Ambikesh Jayal at Cardiff Metropolitan University, for devoting much of his precious time and valuable advice for enriching and evaluation this work. I appreciate his close enthusiastic co-operation as well as his generous effort, kind guidance saving no effort or time to help me and read every word in this work.

I would like to express my deepest gratitude and sincere thanks to all my professors and staff of Computing here in Cardiff Metropolitan University, who showed a great interest in providing me with valuable data, informing guidance and encouragement, and supported me all throughout all my studies.

Last but not least, I am extremely thankful for my mother, father and my sisters for their encouragement and support. I dedicate this work to them in an attempt to express my gratitude to their priceless sacrifices.
# Table of Contents

Declaration .................................................................................................................. I
Abstract ....................................................................................................................... II
Acknowledgment ......................................................................................................... III
List of Tables .................................................................................................................. Error! Bookmark not defined.
List of Figures ............................................................................................................... VII
List of Abbreviations ..................................................................................................... VIII
Introduction .................................................................................................................... 2
  Medical Background ..................................................................................................... 2
  Articulation Disorders ................................................................................................. 3
  Speech Disorders Assessment ..................................................................................... 4
Arabic Language ............................................................................................................ 4
Problem Statement & Contribution ............................................................................. 5
Dissertation Layout ........................................................................................................ 6
Literature Review .......................................................................................................... 8
  Related work in literature .......................................................................................... 8
  Available Datasets .................................................................................................... 10
  Speech disorders detection tools ................................................................................ 10
Materials & Methods .................................................................................................... 14
  Proposed Model ......................................................................................................... 14
  Dataset ....................................................................................................................... 14
Feature Extraction .......................................................................................................... 16
  MFCC (Mel Frequency Cepstral Coefficient) ............................................................ 16
  Matlab ....................................................................................................................... 19
Classification .................................................................................................................. 21
  Support Vector Machines SVM .................................................................................. 21
  Artificial Neural Network (ANN) ................................................................................ 24
  K-Nearest Neighbour Algorithm (KNN) ..................................................................... 26
WEKA .......................................................................................................................... 26
Results ............................................................................................................................ 30
Classification Performance measures .......................................................................... 30
Cross-Validation ............................................................................................................. 30

IV
Classification Results ................................................................. 31
Analysis .......................................................................................... 35
Conclusion and Future Work .......................................................... 38
Conclusion ...................................................................................... 38
Future Work .................................................................................... 38
  1- More Data: .................................................................................. 38
  2- Application Framework: ............................................................... 39
References ....................................................................................... 40
Appendix A: Code Screen Shots ...................................................... 45
Appendix B: Ethics Form ................................................................. 48
List of Tables

Table 1 : Available Dataset for speech disorder detection .................................................. 10
Table 2: Dataset summary ........................................................................................................ 16
Table 3 : KNN classification results ....................................................................................... 32
Table 4 : SVM classification results ....................................................................................... 33
Table 5 : Multi-Layer Perceptron classification results .......................................................... 34
Table 6: Overall classification results ..................................................................................... 35
List of Figures

Figure 1 : Human speech Communication (Holmes& Holmes,2001) .............................................. 2
Figure 2 : Model phases .................................................................................................................. 14
Figure 3 : Dataset description ........................................................................................................ 15
Figure 4 : MFCC diagram .................................................................................................................. 17
Figure 5 : Pre-Emphasis step, MFCC ............................................................................................. 18
Figure 6 : Framing Step, MFCC ........................................................................................................ 18
Figure 7 : The bounding planes of a linear SVM ............................................................................. 22
Figure 8 : Choosing the hyperplane that maximizes the margin ...................................................... 23
Figure 9 : Separating the Data in a Feature Space ........................................................................... 23
Figure 10 : Hidden Layer layout ..................................................................................................... 25
Figure 11 : Multi-Layer Perceptron ................................................................................................. 25
Figure 12 : SMO parameters ........................................................................................................... 27
Figure 13 : Multi-layer Perceptron parameters .............................................................................. 27
Figure 14 : IBK parameters ............................................................................................................. 28
Figure 15 : fold cross-validation ...................................................................................................... 31
Figure 16 : Overall classification graph .......................................................................................... 36
Figure 17 : Mansora Arab Articulation Test ..................................................................................... 39
# List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC</td>
<td>Analog to Digital Conversion</td>
</tr>
<tr>
<td>ANN</td>
<td>Artificial Neural Network</td>
</tr>
<tr>
<td>ASR</td>
<td>Automatic Speech Recognition</td>
</tr>
<tr>
<td>CA</td>
<td>Colloquial Arabic</td>
</tr>
<tr>
<td>CAS</td>
<td>Childhood Apraxia of Speech</td>
</tr>
<tr>
<td>DCT</td>
<td>Discrete Cosine Transform</td>
</tr>
<tr>
<td>DN</td>
<td>Dependency Network</td>
</tr>
<tr>
<td>EPS</td>
<td>Encapsulated PostScript</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
</tr>
<tr>
<td>GMM</td>
<td>Gaussian Mixture Model</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HFCC</td>
<td>Human Factor Cepstral Coefficients</td>
</tr>
<tr>
<td>HNN</td>
<td>Hidden Markov Model</td>
</tr>
<tr>
<td>KNN</td>
<td>K Nearest Neighbor</td>
</tr>
<tr>
<td>LBPCC</td>
<td>Local Binary Pattern Cepstral Coefficient</td>
</tr>
<tr>
<td>LPC</td>
<td>Linear Prediction Coefficient</td>
</tr>
<tr>
<td>MAAT</td>
<td>Mansoura Arab Articulation Test</td>
</tr>
<tr>
<td>MFCC</td>
<td>Mel-frequency Cepstrum Coefficients</td>
</tr>
<tr>
<td>MLP</td>
<td>Multi-Layer Perceptron</td>
</tr>
<tr>
<td>NP</td>
<td>Normalized Polynomial kernel</td>
</tr>
<tr>
<td>PK</td>
<td>Polynomial Kernel</td>
</tr>
<tr>
<td>PUK</td>
<td>Pearson VII Function- Based Universal Kernal</td>
</tr>
<tr>
<td>RBF</td>
<td>Radial Basis Function Kernel</td>
</tr>
<tr>
<td>SAP</td>
<td>Speech and Audio Processing</td>
</tr>
<tr>
<td>SFS</td>
<td>Speech Filling System</td>
</tr>
<tr>
<td>SLT</td>
<td>Speech and Language Therapy</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td>VAD</td>
<td>Voice Activity Detection</td>
</tr>
<tr>
<td>VQ-LBG</td>
<td>Vector Quantization-Linde, Buzo and Gray</td>
</tr>
<tr>
<td>WR</td>
<td>Word Recognition</td>
</tr>
</tbody>
</table>
CHAPTER ONE

INTRODUCTION
Introduction

Communication is central to everything we do. It affects who we are, how we learn, how we interact with other people at home, at school, and at work and almost everywhere. Human communication is done mainly via “SPEECH”. It is an important sign of the normal growth; and hence, acquisition of speech sounds can be considered as a developmental process.

‘Speech disorders’ occur when the person’s speech sounds are not correct or fluent. This differs from ‘Language disorders’, where the person is unable to express himself (expressive language disorders) or to understand others (receptive language disorders) (Association, 1997).

One of the critical point is the treatment of articulation problems taking in concern the person’s social, emotional, educational status. A person’s everyday real life situation is affected by speech and the adequacy of it since speech is one of the most important ways of communication between people (Georgoulas and Georgopoulos, 2006).

Medical Background

Speech disorders can be seen in both children and adults; however, in children it is more harmful since ineffective communication usually lead to psychological problems. A child's communication is considered delayed when the child is noticeably behind his or her peers in the acquisition of speech and/or language skills (Shafer, no date).

Many forms of speech disorders have been reported. They can be classified according to causative factors as follows:
**Organic disorders**

**Motor Speech Disorders**
These include speech difficulties due to Childhood Apraxia of Speech (difficulty planning movements for speech) or dysarthrias (difficulty making movements for speech due to paralysis).

**Structurally-based Speech Sound Disorders**
These include speech difficulties associated with head/facial anatomy differences (e.g., cleft palate, misaligned teeth, or craniofacial differences associated with some syndromes).

**Speech Sound Disorders associated with syndromes**
These include speech difficulties associated with syndromes such as Down syndrome, metabolic conditions such as galactosemia, and sensory conditions such as hearing impairment.

**Functional disorders**

**Articulation Disorder**
This involves difficulty producing one or just few speech sounds. Another term for this is Functional Speech Disorder.

**Phonological Disorder**
This involves persistence of errors that are typical of younger children’s speech.

There is all sort of organic disorders such as; hearing disabilities, perceptual causes, oral-motor difficulties, Dysarthria, and Oral- Structural deviation. These disabilities can be treated by clinical surgeries or psychologically (Kaneshiro et al., 2014).

This research is concerned with the inorganic speech disorder, more specifically articulation related problems.

**Articulation Disorders**
Articulation simply means making sounds. Errors in producing the correct sound are usually treated through repetitive speech exercises to adjust the linguistic behavior. The given exercises are dependent on the nature of the disorder, in other words the diagnosis (Association, 1997).

The disorder usually falls in one of the following categories (Association, 1997):

- **Substitutions**: appears in pronunciation when issuing inappropriate sound for the letter instead of the desired one (replace one sound with another sound).
  
  Examples: “wed” for “red,” “thoap” for “soap,” “dut,” for “duck”, “wabbit” for "rabbit"

- **Omissions** (also known as deletions): The child omits the sound of a (letter) of the votes, which is contained in the word and then pronounces only a part of the word. These defects tend to appear in the pronunciation of consonants, which is located at the end of the word more than consonants at the beginning of the word or in the middle (Association, 1997).
  
  Examples: “p ay the piano” for “play the piano”, “g een nake” for “green snake”, "cool" for "school".

- **Additions**: Insert an extra sound (letter) to the word.
  
  Examples: “buhlack horse” for “black horse,” “doguh,” for “dog”, "pinanio" for "piano"
Distortions: Produce a very close sound to the regular one but in an unfamiliar manner. This may occur as a result of loss of teeth, the tongue is not put in its proper position during pronunciation, deviation of teeth position or loss of teeth on both sides of the lower jaw, which makes the air goes to both sides of the jaw and therefore the child cannot Pronounce the sounds.

Examples: “pencil” (nasalized—sounds more like an “m”) for “pencil,” “sun” (lisp—sounds “slushy”) for “sun”.

Speech Disorders Assessment

Evaluation of children with speech difficulties is done by a professional who examines the child to determine his overall communication functioning. Such “Speech pathologist” is trained in distinguishing speech impairments and is the one responsible for formulating a treatment plan suitable for each case. He/She often uses standardized assessment instruments like: single-word testing and connected speech sampling (Association, 1997).

For the English language, tests that diagnose speech disorders are: Denver Articulation Screening Examination, Early Language Milestone Scale, Denver II, Peabody Picture Test Revised (Kaneshiro et al., 2014).

Arabic Language

It was stated that the Arabic language is the fifth most widely used language nowadays as being spoken by approximately 400 million people throughout world. Also, it happens to be one of the official languages of the United Nations. Consequently, numerous individuals try to learn the language for formal reasons. The best books of medicine, geology, law and any subject related to sciences were all composed in Arabic during the Islamic Empire 1500 years ago (Jamil, no date).

“Arabic” is not a single linguistic variety; rather, it is a collection of different dialects. Currently, there are three forms of the Arabic language: classical Arabic, modern standard Arabic, and colloquial Arabic.

Classical Arabic is the type of Arabic utilized in the blessed Quran. Modern classical Arabic is the official dialect of writing, science, technology, education, and accordingly it is utilized generally as a part of composing, news, shows, formal discourses, and motion picture subtitles. It differs from Classical Arabic in that it is composed with the non-attendance of diacritic marks. Colloquial (or dialectal) Arabic refers to the regular talked Arabic utilized as a part of ordinary life among Arabic speakers; it is entirely talked, not composed, and it is not taught in school (Alghamdi, 2001).

As Arabic is the native language for almost 22 countries, there is multiple lingos that can separate the Arabic language talking world (Bakalla, 1984):

- Gulf Arabic, spoken by: Saudi Arabia, Kuwait, Qatar, Bahrain, United Arab Emirates, and Oman.
- Levantine Arabic, spoken by: Syria, Jordan, Lebanon, and Palestine.
- Egyptian Arabic, spoken by: Egypt and Sudan.
- Maghrebi Arabic, spoken by: Algeria, Morocco, Tunisia, and Mauritania.
- Libya Arabic, sometimes it can be included with the Maghrebi dialect.
- Yemenite Arabic
- Iraqi Arabic.
We can classify these dialects into two groups (Kaye and Rosenhouse, 1997):

- Eastern Dialects, containing Gulf Arabic, Iraqi Arabic, Levantine Arabic, Egyptian Arabic and Yemenite Arabic.
- Western Dialects, containing Maghrebi Arabic and Libya Arabic.

The Arabic language consists of 27 consonant phonemes with a wide range of places of articulation that span the whole vocal tract from lips to glottis. Its vowel system is relatively simple with eight vowel phonemes: five long (/i:, e:, a:, o:, u:/) and three short (/a , i, u/). Vowels don't happen in beginning position, and there are no diphthongs (Abou-Elsaad, Baz, and El-Banna, 2009).

The consonant stock of Colloquial Arabic CA incorporates the essential insistent phonemes /t/, /d/, /s/, /z/, which causes the Arabic dialect to be recognized from the rest dominant part of European languages. It additionally incorporates the phoneme/q/, which is regularly substituted by /ʔ/, with just couple of remarkable words ingested from traditional Arabic. The Arabic stock incorporates back consonants that are glottal (/ʔ/), velar (/x/, /γ/) or pharyngeal (/ћ/, /ʕ/) (Abou-Elsaad, Baz, and El-Banna, 2009).

As mentioned before, most articulation errors fall into one of four categories. This is valid also for the Arabic Language.

Examples for articulation disorders in Arabic languages are (Hammami et al., 2015):

1. Distortion: Madrassa (مدرسة) - pronounce – Madrtha (مدرسة) Dabet (ضابط) - pronounce – thebet (ثابط)
2. Omission: means to completely remove a letter when pronouncing: Koura (كرة) - pronounce Kouaa (كوة)
3. Substitution: replacing letter (س/s/) with (ش/sh/) or replace the letter (ر/r/) with a letter (و/w/).
4. Addition: “SSSbah...” (good morning)

**Problem Statement & Contribution**

In the light of Speech and Language Therapy (SLT) field, many systems have been created and adapted to help and support speech therapists during their sessions. The common practice in this field is the use a number of paper-based systems to reach out for the young kids that have problems or difficulty in language acquisition and help them to properly understand the language (Hammami et al., 2015).

In order to introduce the young kids to new words and vocabularies, a set of system employing card and pictures or photographs are mixed together to give the child a new storyline or context. While such a system has a great success rate, however it faces challenges such as overcoming the child fear of meeting a doctor in therapy center, the need to get the child focused all through the session duration. Such challenges motivated the used technology to move beyond the classical paper approach. It also represent a more interactive experience to the child and definitely much more fun which in return make the therapist work more easier and effective (Shahin et al., 2015).
Interactive automatic speech monitoring tools, which can be used remotely at the child’s home, offer a practical, adaptive and cost-effective alternative to face-to-face intervention sessions for children. Even though such automated therapy tools do exist but to our knowledge none do exist for Arabic language (Shahin et al., 2015). This dissertation is a step closer to an automatic Arabic speech therapy tool. It mainly investigates the possible features suitable for the diagnosis phase and the corresponding classification accuracy.

**Dissertation Layout**

This thesis is organized as follows:

- Chapter one provided an introduction of the topic along with the motivation behind the work.
- Chapter two illustrates some of the work presented in literature. It covers in details two main stages of Speech analysis process; the feature extraction stage and the classification stage along with the current software available for speech analysis.
- Chapter three gives a detailed description of the methods and tools used to build the system. It discusses MFCC, SVM, ANN, KNN.
- Chapter four describes all the experiments conducted to validate the proposed model. It also illustrates the performance indicators used in evaluating the system performance and the experimental results.
- Chapter five illustrates the conclusion and the future work.
- Finally, chapter six lists the references as well as the appendix
Literature Review

Throughout the years, many authors and researchers have carried out experiments through advanced Artificial Intelligence techniques. The first part illustrates a brief description of some of the previous scientific work done in the automatic speech analysis field; it highlights the methods and techniques used specifically for speech disorder recognition. A list of available speech datasets is provided in the second section; this list combines speeches from several languages. It was also very important to review current developed tools in order to choose the best for carrying out my diagnosis model. This is why, the thirds and last section provides a summary of the existing analysis tools.

Related work in literature

Georgoulas and Georgopoulos (2006) examined ASR using data set from young children speaking the Greek language. With the help of speech therapists, assessments and classification were made on the articulation of the kids. They used the wavelet as features extraction, which a mathematical function commonly used in voice detection process. The wavelet equation got an advantage in digital signal processing as it separates the weak signals from the actual noise. It divides the signal into several parts using time; however it does not change the signal’s shape. Changes in the time augmentation are relied upon to fit in with the analysis frequency of the function. As classification of speech sounds and detection of articulation disorders, they used the Support Vector Machines SVM method, which is a non-linear binary classifier. Experiments and tests were done with distinctive plans of SVMs which made them happy with the results.

According to Chen (2010), they created a system using photo naming task to get the subject’s speech sample. Chen used a dependency network (DN) approach in articulation recognition due to its many pros in this field. After evaluating the clinical protocol, this DN scheme was created to present the connections of the items following this protocol. This automatic speech recognition system and detection of voice disorder system was set up by incorporating three parts, the dependency network, automatic speech recognition and the pronunciation confusion network.

According to Alsulaiman, Muhammad, and Ali (2011), Arabic Language has its own attributes; hence, some features and method can be more suitable than the others. They did a comparison between 2 methods of features extractions: Mel-Frequency Cepstral Coefficient (MFCC) and Linear Prediction Coefficient (LPC). Their paper state that the best results were obtained using the MFCC method, for Arabic language. They adjusted the way of using MFCC, presenting a new enhanced way called Local Binary Pattern Cepstral Coefficients (LBpCC).

Cmejla, Rusz, Bergl and Vokral (2012), work is considered an extension of Bayesian change point detection, which is widely used in signal processing field. The novelty of the work relies on the signal segmentation into sliding window and the normalization of a posteriori probability through Bayesian evidence. Moreover, the model is executed in a recursive fashion. They validated their model on two sets of data;(a) a total of 118 subjects with variant levels of stuttering difficulties, (b) 46 subjects divided into 24,22 Parkinson’s patients and healthy subjects respectively. All of the subjects are Czech native speakers and were asked to read a short reading passage instead of single words.

Ali, Alsulaiman, Muhammad and Elamvazuthi (2013), investigate voice disorder detection in continuous speech. Disorders such as vocal fold disorders, cyst, polyp, nodules, paralysis, and sulcus, are detected through Mel-frequency cepstral coefficients (MFCC) analysis. The coefficients
are then fed to a Gaussian mixture model (GMM) to classify pathological and normal voices. The model achieved an accuracy of 91.66% with 100% specificity and 87.5% sensitivity. The author used a data set of 26 subjects having vocal fold disorder and 12 healthy subject clinically proved. The records contained three utterances of 3 vowels; /a/, /u/ and /i/.

Grzybowska and Kłaczynski (2014), created computer software (to aid articulation treatment) by relying on a modified version of the Mel Frequency Cepstral Coefficients (MFCCs). The biologically inspired version also known as Human Factor Cepstral Coefficients (HFCCs) was developed by Skowronski and Harris in 2004. HFCCs implementation differs from MFCCs by decoupling the filter bandwidth. The test was examined on twenty polish words with phonemes located in the first positions of the words. To make sure the system is correct, it was set up in two different places, the anechoic chamber at AGH University and at home in a room to visualize real life environment. Nine males and twenty females, all perfectly healthy and do not suffer from voice disorder, all participated in the data collection just to make sure the system is correct and functions perfectly. For the first set at the university, it recorded from 69% to 82% as Word Recognition rate (WR), while the set conducted at home, represent a range of 86% to 94% WR.

Salunke and Jagtap (2014) conducted a study for Signal Defects Classification using wavelet transform and Multiclass SVM but with different approach. Instead of using one wavelet, they used two wavelet transforms filters in the features extraction procedure, resulting in enhancing the results and performance. For the classification process, they create binary decision tree and use SVM method for each node in the tree.

Kumar and Lahudkar (2015) examined Automatic Speech Recognition (ASR) by focusing on distinguishing the individuals by their special voiceprint. This procedure guarantees the voice highlight extraction contains exact data that passes on the personality of the speaker. They used LPC, LPCC and MFCC as voice features extractions in their analysis and evaluation. To enhance their work, they used LPC and MFCC (LPC+MFCC) combined giving a better or more comparable results. The model was registered utilizing Vector Quantization-Linde, Buzo and Gray (VQ-LBG) technique.

Orozco-Arroyave, Hönig, Arias-Londoño, Vargas-Bonilla and Nöth (2015), conducted an automated detection of speech impairment for Parkinson’s disease patients speaking the Spanish language. Their research focus on five Spanish vowels and a total of 24 isolated words. All recordings were manually pre-processed to remove silence and produce the isolated words. Following pre-processing, recordings passed by a segmentation phase where the signal was divided into windows of 40ms with 20ms of overlap. The authors compared six sets of coefficients as potential features; LPC, LPCC, PLP, MFCC and two versions of the RASTA coefficients. For the classification phase, SVM algorithm was adopted with a Gaussian kernel through a 10-fold cross-validation strategy. Linear predictive cepstral coefficients outperformed all others.

Shahin, Ahmed, Parnandi, Karappa, McKechnie, Ballard and Gutierrez-Osuna (2015) have created a tool for the childhood apraxia of speech (CAS). This tool is based on three basic steps; (1) monitoring the child progress by giving an exercise to do created by the speech therapist for different children. (2) Data collecting step by recording the child speech sample. (3) After having the samples, it passes through speech recognition phase and analyse the patient giving a diagnosis to the therapist. The engine used in the speech recognition process relies on Voice Activity Detection (VAD) algorithm by discriminating two major points in CAS; delay in voice and absence of voice. (VAD) divides the sample in 0-10 ms frames to calculate the intensity of each frame, to estimate the silence
threshold value. Comparison are done between the intensity and silence threshold and labelled as speech and silence.

Available Datasets

Many databases have been used for speech disorder detection. Some of these are available online as shown in the following table; others are created by researchers to suit their subjects. The main purpose of researching other languages, even though completely different of Arabic language in terms of produced sounds, was to understand the language-dependant properties. Not all method produces same results and performance when applied on different languages.

<table>
<thead>
<tr>
<th>Work</th>
<th>Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Su et al., 2008</td>
<td>The TCC300 speech corpus</td>
</tr>
<tr>
<td>Miyamoto et al., 2010</td>
<td>The ATR Japanese speech database</td>
</tr>
<tr>
<td>Maier et al., 2009</td>
<td>VERBMOBIL data</td>
</tr>
<tr>
<td>Nayak et al., 2003</td>
<td>The “Baby Chilanto” database from “Instituto Nacional de Astrofisica Optica y Electronica – CONACYT, Mexico”.</td>
</tr>
<tr>
<td>Saz et al., 2005</td>
<td>The corpus recorded by the Department of Signals and Communications from the University of Las Palmas de Gran Canaria (Spain)</td>
</tr>
<tr>
<td>Kotropoulos et al., 2009; Vijayalakshmi et al., 2009; Ziogos et al., 2006</td>
<td>The Massachusetts Eye &amp; Ear Infirmary (MEEI) Voice Disorders Database</td>
</tr>
<tr>
<td>Scipioni et al., 2009</td>
<td>The ChildIt speech corpus</td>
</tr>
</tbody>
</table>

*Table 1 : Available Dataset for speech disorder detection*

Speech disorders detection tools

Numerous researches were conducted in speech disorder detection field over years using Automatic speech recognition ASR technology (Georgoulas and Georgopoulos, 2006). Starting from the 70s, Detection was based on the difference between normal speech and the abnormal one (Association and Resources, 2013). Recently, more approaches have been invented and used; there is a huge progress in this field due to the usage of Artificial Intelligence methods like artificial neural network (ANN), Hidden Markov Models (HNN) and support vector Machines (SVM) (Alsulaiman, Muhammad, and Ali, 2011).

There are numerous open-source tools for the acoustic analysis of speech that can be used on different operating system such as windows and MAC OS; they differ from each other in terms of usability, utility, research and clinical purposes.

**Matlab:** is a general purpose software with several open-source toolbars that support speech acquisitive and analysis. Examples of such toolbars include:

- Speech & audio Processing toolbar (SAP): it contains useful functions that can be used for speech and audio signal processing
  [http://mirlab.org/jang/matlab/toolbox/sap/]
• Colea: is a toolkit for speech analysis that contains a graphical user interface (GUI). This tool allows the user audio processing, speech filtering, comparing waveforms and video processing.
[http://ecs.utdallas.edu/loizou/speech/colea.htm]

• Voicebox: contains a set of functions and methods that can be used in reading speech samples and analysing it.
[http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html]

• Auditory Toolbox: increase Matlab abilities by supplying it with different auditory commands, all written in a high-level scripting language, still easy to use.
[https://engineering.purdue.edu/~malcolm/interval/1998-010/]

• Voice Sauce: a designed extension for matlab to measure voice signals over time from .wav files.
[http://www.seas.ucla.edu/spapl/voicesauce/]

**PRAAT**: a Dutch word for “Talk”, created and being updated by Paul Boersma and David Weenink, Phonetic sciences department from University of Amsterdam. PRAAT is a computer tool designed for the user to be able to analyse and adjust speech. Its features can be described as following:

• Speech analysis: contain analysis for spectral content; pitch contour, formant contours, intensity contour of existing sound sample. It also does jitter, shimmer and voice breaks and pattern excitation.

• Labelling and segmentation: to comment on existing sound objects and files, draw phonetic symbols in the picture window and the ability to view and label a sound file.

• Learning algorithms: feedforward neural network option and Optimality-Theoretic and Harmonic-Grammar learning.

• Graphics: picture window to save the files a command in the File to save the pictures as Encapsulated PostScript (EPS) file to be used by other programs and a feature to display special symbols into the TextGridEditor

• Portability: to read multiple files and write it as binary file to be able to be read again.
[http://www.fon.hum.uva.nl/praat/]

**OpenSmile**: is sound extraction software for large sound data in real life environment. The coding is done by C++ but can be used as both a standalone command line executable and as a dynamic library. It contains a friendly binary plugin interface and a comprehensive API, which gives the ability to add more new components to OpenSmile.

It was created by Florian Eyben, Martin Wöllmer, and Björn Schuller at Technische Universität München (TUM) in 2008 (Eyben, Woellmer, and Schuller, 2010).

OpenSmile has a lot of features such as:

• Works on multi operating system (Windows, Linux, Mac and android)
• Quick and effective incremental preparing in real-time
• High measured quality and reusability of components
- Plugin support
- Multi-threading support for parallel features extraction
- Audio I/O
- General audio signal processing
- Extraction of speech-related features
- Music-related features
- Moving average smoothing of feature contours
- Moving average mean subtraction and variance normalisation (e.g. for on-line cepstral mean subtraction)
- On-line histogram equalisation
- Delta Regression coefficients of arbitrary order
- Statistical functionals
- Popular I/O file formats are supported
- Fully HTK compatible: MFCC, PLP, (log-)energy, and delta regression coefficient computation

[http://audeering.com/research/opensmile/#download]

Speech Filing System (SFS:) is an open source computer program invented by University College London and being monitored and upgraded regularly. SFS allows the user to analyse speech samples and recognition activities. It includes more than twelve individual programs that are helpful for speech signal processing and clinical biofeedback. Some of these programs can be coordinated with each other to give a highly effective tool. Using SFS can be difficult for someone who is a beginner or low experience programmer.

[http://www.phon.ucl.ac.uk/resource/software.html]

Speech Analyzer: is a free and expert software that represent different graphical representations of speech and music recordings. It specializes in phonetic analysis of human voice recordings. It performs the following tasks:

- Perform fundamental frequency, spectrographic and spectral analysis, and duration of measurements.
- Add phonemic, orthographic, tone, and gloss transcriptions to phonetic transcriptions in an interlinear format.
- Perform ethnomusicological analysis of music records
- Use slowed playback, repeat loops and overlays to assist with perception and mimicry of sounds for language learning

[http://www-01.sil.org/computing/sa/index.htm]
CHAPTER THREE

Materials and Methods
Materials & Methods

Proposed Model

After an extensive survey of the widely used tools and models, I sketched my own model based on the most used feature with both an ease of implementation and the good results achieved in other datasets.

The figure below illustrates the 4 phases of the automatic diagnosis of articulation errors, starting with a data acquisition phase, a pre-processing phase, followed by a feature extraction phase and finally a classification phase where the diagnosis is stated.

![Diagram of model phases]

**Figure 2: Model phases**

Dataset

Since the main point of this study is to build up a tool or a system for programmed finding of speech disorders and restoration for Arabic language, the most essential step is to create a good corpus that can be used as a base to serve as a premise for creating a robust model.

This research was only limited to the data that could be found online and hence there was no acquisition nor pre-processing phase as the data was already downloaded from the web. There were several limitations to creating my own dataset, including: the time limit for delivering the dissertation, and my physical location outside of my home country.

The only existing Arabic dataset ((/r/-LDD) dataset) can be publically found at [https://drive.google.com/file/d/0Bw8xpk6fIOQUl0N40DRLbGpjTzg/view](https://drive.google.com/file/d/0Bw8xpk6fIOQUl0N40DRLbGpjTzg/view) (Hammami et al., 2015).ss

The main drawback of this data is that it recorded only a single letter disorders, which is the letter (/r/), not the whole alphabet letters. It is worth mentioning that their choice of the letter (/r/) was not random, but after consultation with speech specialities as it represents one of the most common letters of speech disorders in Arabic language across all dialects.

The dataset is divided into three sub-databases:

- the first is for diagnosis of the disorder in a letter (/r/) when its position is in the beginning of the word, (/r/-LDFW),
- the second when letter (/r/) is in the middle of the word, (/r/-LDMW),
- the last one when it is at end of the word. (/r/-LDFE).

All children were asked to pronounce the word rajulun, mariyam and kabeer (arabic for man, Mariam (name) and big) for each subset correspondingly.
The figure below explains the hierarchy of the system:

Each sub database contains isolated speech recordings for 60 children: 30 males and 30 females; each child repeats the voice disorder five times, in addition to another 8 males and 8 females for testing purposes.

The mother tongues of all participants was Arabic, covering most Arabic dialects; Egyptian, Levantine, Maghrebi, Iraqi and gulf.

Due to the lack of cases, this corpus acquired healthy children to simulate the speech disorders. Speech therapists were used as experts to evaluate and categorize the articulation of children.

The choice of these words was based on the different types of disorders when letter (ṣr/l) was in the beginning, middle or end respectively. Each voice disorder is repeated five times for the same word by the same child. The microphone was placed on a stand about 5 to 20 cm, slightly off-axis from the speaker with varying recording gain.

A summary of the dataset with corresponding instances is illustrated in the following table:
<table>
<thead>
<tr>
<th>Class</th>
<th># of recordings</th>
<th>Class</th>
<th># of recordings</th>
<th>Class</th>
<th># of recordings</th>
</tr>
</thead>
<tbody>
<tr>
<td>/l/- Letter disorder on the beginning of Words (/l/-LDFW)</td>
<td>Correct Word speech (Rajulun /rajulun/)</td>
<td>300</td>
<td>Correct Word speech (Mariyam /مريم/)</td>
<td>300</td>
<td>Correct Word speech (Kabeer /كبير/)</td>
</tr>
<tr>
<td></td>
<td>Substitution disorder (/l/ with /ɣ/ (/g/)) (Ghajulun /غجل/)</td>
<td>300</td>
<td>Substitution disorder (/l/ with /ɣ/ (/g/)) (Maghiam /مغيم/)</td>
<td>300</td>
<td>Substitution disorder (/l/ with /ɣ/ (/g/)) (Kabeegh /كبيغ/)</td>
</tr>
<tr>
<td></td>
<td>Substitution disorder (/l/ with /ɬ/) (Lajulun /لجل/)</td>
<td>300</td>
<td>Substitution disorder (/l/ with /ɬ/) (Maliam /مليم/)</td>
<td>300</td>
<td>Substitution disorder (/l/ with /ɬ/) (Kabee l /كبيي/)</td>
</tr>
<tr>
<td></td>
<td>Omission disorder (Julun /جُل/)</td>
<td>300</td>
<td>Addition disorder (Marereriyam /مرَرريم/)</td>
<td>300</td>
<td>Omission disorder (Kabee /كبيي/)</td>
</tr>
<tr>
<td></td>
<td>Addition disorder (Rararajulun /رَررجُل/)</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1500</td>
<td></td>
<td>1200</td>
<td></td>
</tr>
</tbody>
</table>

**Table 2: Dataset summary**

**Feature Extraction**

Generally, feature extraction is how to represent each instance in a given dataset. The choice of the features along with methods to estimate (extract or measure) them both have a great effect on the classification; as low quality features poorly describe a record. In the given problem, the objective is to discriminate between the correct sound and other miss-pronounced sound. For that specific task the Mel Frequency Cepstral Coefficient are proved more efficient than other methods (Tiwari, 2010).

**MFCC (Mel Frequency Cepstral Coefficient)**

In the 1980’s, Mel Frequency Cepstral Coefficient (MFCC), created by Davis and Mermelstein, is considered one of the best speech analysis models. People were using Linear Prediction Coefficient (LPC) and Linear Prediction Cepstral Coefficient (LPCC) before (MFCC) come along. MFCC functions by imitating the logarithmic impression of loudness and pitch, found in the human speech.
production and perception, and make an attempt to completely remove speaker dependent characteristics by removing the essential frequency and their noise and music to give a simplified representation of the timber of a person (Lutter, 2014).

MFCC depends on human listening ability which can't recognise frequencies more than 1 Khz. It contains two filters which are divided into low frequency, below 1000Hz, and high frequency, above 1000Hz (Muda, Begam, and Elamvazuthi, 2010). The MFCC calculation is accomplished in six steps as shown in figure 4:

1. Pre-emphasis
2. Framing
3. Hamming windowing
4. Fast Fourier Transform
5. Mel Filter Bank
6. Discrete Cosine Transform

Figure 4: MFCC diagram
**Pre-emphasis**
In this step, the audio signal is passed through a filter which points out higher frequencies. Figure 5 demonstrates the pre-emphasis effect and how the signal looks like before and after. This phase will build up the energy of a signal at higher frequency according to equation 1:

\[ Y(n) = X(N) - 0.95 \cdot X(N - 1) \]

*Figure 5: Pre-Emphasis step, MFCC*

**Framing**
The next procedure is dividing the wave into multiple distinct “windows” of fixed length N (20ms < N < 40ms). The framing also follows an overlapping scheme of 50% of the frame size, obtained from the analog to digital conversion (ADC). An overlap is important as it gives equal weights to all sample in the given signal; since in the windowing the end samples in the frame get attenuated. (Muda, Begam, and Elamvazuthi, 2010).

*Figure 6: Framing Step, MFCC*

**Hamming windowing**
Hamming ensures all close frequency lines are grouped together throughout the processing chain. The Hamming window equation is given as:
If the window is defined as \( W(n) \), \( 0 \leq n \leq N-1 \) where

\[ Y(n) = X(n) \times W(n) \]

\[ w(n) = 0.54 - 0.46 \cos \left( 2\pi \frac{n}{N} \right) \], \( 0 \leq n \leq N \)

Where \( N \) = number of samples in each frame, \( X(n) \) is the input signal, \( Y[n] \) is the Output signal and \( W(n) \) = Hamming window
**Fast Fourier Transform**

Simply, FFT is applied to convert the signal from its initial domain to the frequency domain or vice versa. In our case, the voice signal is translated from time domain to frequency domain following equation. The most important parameters in short-time Fourier analysis are the width and the shape of the time window (Chakraborty, Talele, and Upadhya, 2014).

\[ Y(w) = FFT \left[ h(t) \ast X(t) \right] = H(w) \ast X(w) \]

If \( X(w), H(w) \) and \( Y(w) \) are the Fourier Transform of \( X(t) \), \( H(t) \) and \( Y(t) \) respectively.

**Mel Filter Processing**

After FFT, there’s a huge gap between the frequencies in the spectrum and the voice signal detection, that’s where Mel Filter comes in. It applies the Mel-frequency scaling, which is perceptual scale that helps to simulate the way human ear works. It corresponds to better resolution at low frequencies and less at high. Using the triangular filter-bank helps to capture the energy at each critical band and gives a rough approximation of the spectrum shape, as well as smooths the harmonic structure (Chakraborty, Talele, and Upadhya, 2014). Mel filter equation is:

\[ Mel(f) = 1125 \ast \ln \left( 1 + \left( \frac{f}{700} \right) \right) \]

**Discrete Cosine transform**

The final step is to perform discrete cosine transform on the obtained output of the triangular Mel filter where every input is transformed into a sequence of acoustic vectors (Chakraborty, Talele, and Upadhya, 2014). A discrete cosine transform (DCT) expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies as in the following equation. The DCT equation is:

\[ C(n) = \sum E_k \ast \cos \left( n \ast \left( k - 0.5 \right) \ast \pi / 40 \right) \]

Where \( n = 0, 1, \ldots \) to \( N \), \( N \) is the number of triangular bandpass filters and \( L \) is the number of mel-scale cepstral coefficient.

Mel Frequency Cepstrum Coefficients are amplitudes of the resulting spectrum. It is worth mentioning that only coefficient 1 through 12 are stored and represented as the 0-th coefficient is usually ignored. Not only 12 coefficients can be extracted but they can also be combines with 12 delta coefficients and 12 delta-delta coefficients to form a set of 12, 24 or 36 coefficients in total. Delta values are the velocity whereas delta-delta values are the acceleration coefficients. Velocity and acceleration coefficients can be computed by performing first and second order derivatives on the 12 original MFCC components (Muda, Begam, and Elamvazuthi, 2010).

**Matlab**

The main purpose of Matlab is to make programming easier and widely available through several online courses and tutorials. It is also a very powerful programming language used world-wide by natural scientists, engineers and everyone and created modules and components integrated in many of our daily-use devices like cars’ software and smartphones. MATLAB is an extraordinary reason language that is typically used in:

- Math and computation
• Algorithm development
• Modelling, simulation, and prototyping
• Data analysis, exploration, and visualization
• Scientific and engineering graphics
• Application development, including Graphical User Interface building

All of the above made the Matlab the preferred choice when comparing the tools previously surveyed in chapter two. The existence of several toolboxes supported by Matlab also encouraged that choice as I was easily comparing different Matlab scripts that implemented the MFCC algorithm on one platform (MathWorks, 1994).

**VOICEBOX**

Voicebox is an open-source speech recognition toolbox created by Mike Brookes from Imperial College University in London; it includes several Matlab libraries and codes. The sample codes can be found online and are downloaded to use through the following link http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html (VOICEBOX, no date). The tool has several important functions for any speech analysis including:

- Reading and writing an Audio File, it supports different formats such as wav, xx.
- Frequency scaling and conversion between Hz, Mel, Erb and MIDI frequency scales.
- Transformations such as Fourier, Discrete cosine, Hartley Transforms and others.
- Noise generation through Random Number and Probability Distributions.
- Basic Speech Analysis methods such as Active level estimation, and Spectrograms.
- Linear Predictive Coding routines extraction.
- Text-to-speech synthesis and glottal waveform models.
- Spectral noise subtraction.
- Speech Coding; PCM coding, Vector quantisation.
- Complexity analysis with routines that calculates entropy and symbol codes.

In my model, I used one specific function called “*melcepst*”, which can be found within the “Speech Recognition” option. It calculates the MFCCs passing through the steps as mentioned in the previous section. It takes as input a wave file and returns 12 coefficients per frame. Even though other parameters can also be passed to the function to set certain values but in my analysis I used all the default values provided by the toolbox. These parameters are:

- `s` speech signal [INPUT]
- `fs` sample rate in Hz (default 11025)
- `nc` number of cepstral coefficients excluding 0'th coefficient [default 12]
- `p` number of filters in filter bank [default: floor (3* log (fs)) = approx. 2.1 per octave]
- `n` length of frame in samples [default power of 2 < (0.03*fs)]
- `inc` frame increment [default n/2]
- `fl` low end of the lowest filter as a fraction of fs [default = 0]
- `fh` high end of highest filter as a fraction of fs [default = 0.5]
- `w` mode string

My model set the time domain for the hamming window and a triangular shaped filter in the Mel domain. Only 12 main coefficients were extracted per frame and not the delta nor the delta-delta coefficients. The “*melcepst*” calls several functions within its script including:
- Enframe: separate the signal into overlapping frames.
- Melbankm: that’s the filter process by using filter bank.
- Rdct: Discrete Cosine Transform process of real data.
- Rfft: calculate the delta features of data.

Matlab Scripts created

Even though the voicebox tool can perfectly generate the needed coefficient, adjustments need to be done in order to construct the feature vector. The sample code for these classes will be found in the (Appendix A), along with the code for “melcepst” class.

- readAllIntoArray: this code is written to batch read all records per subset (1500 file for /r/LDFW and 1200 file for /r-LDMW and /r-LDFE) into an array where each element represents a single recording. It is used to be able to read the wav. files all together in a single run. It is important to note that not all wav files read were of the same size; which forced the use of an array list instead of an ordinary array to overcome the different length issue.
- Analysis: The output of “melcepst” function is very big because it gives out 12 coefficients for every frame. The number of frames per signal is again variant and ranges for the 120 frames to more. In order to create the feature vector per signal, we calculated the maximum, minimum, average and standard deviation of each coefficient over all existing frames. The total numbers of features per record are 12*4, resulting in 48 features. These 48 features are then extracted to an excel sheet for each sub-dataset, ready for the classification phase.

Classification

The machine learning models follow one of two schemes; supervised learning and unsupervised learning. Supervised machine learning is a technique that sets parameters of the classifier from training data. The task of the learning involves setting the value of its parameters for any valid input value after having seen output value (Vapnik and Chervonenkis, 1971). To validate the performance of a learned algorithm, a test dataset that consists of data that has not been introduced to model while learning is fed into the classifier. On the other hand, unsupervised learning is a machine learning technique that sets the parameters based on given data and a cost function which is to be minimized.

The classification phase is very important when discriminating speech records. It is responsible for the final decision of the diagnosis. The features extracted through the estimation of the different values serve as input to the classifier, and a training model is built to discriminate between different classes. In order to assess which classification algorithm is better, a comparison of three classifiers was made: SVM, ANN, KNN. A detailed description of each can be found next.

Support Vector Machines SVM

Support Vector Machines (SVMs) has been known for its quality performance and work in data classification in pattern recognition field. It has been used in a wide range of tasks such as Bioinformatics, text, image recognition and speech recognition. The main use of SVMs was for binary classification tasks, however, it showed great utility in overcoming more complex issues and more than binary classification tasks such is the case with the dataset; as it classifies data into multiple classes (Georgoulas and Georgopoulos, 2006).
The algorithm was firstly introduced to the public by Vladimir Vapnik and colleagues (Boser and Guyon) in 1979, but it was printed out in 1995. Generally, Support Vector Machines are a set of supervised learning strategies whose preparations method allow to show complex non-linear functions (Mo, Matteo, and A, 2012).

The constructed model plot the training set data into classes isolated by a hyperplane that can also be referred to as decision boundary using a training set sample, \( \{x_i, y_i\} \) and a set label with one of two values \( y_i \in \{1, -1\} \). The decision boundary is given by \( wt + x + b = 0 \), where \( W \) represent a vector that have the hyperplane parameters, and \( B \) represent an offset following the equation:

\[
w^T x + b = 0
\]

The data is resized so that anything higher than the boundary \( w^T x + b = 1 \) is labelled (class 1), and anything lower than the boundary \( w^T x + b = -1 \) is labelled (class 2) (Mo, Matteo, and A, 2012).

![Support Vectors are the only data points needed in defining and finding the optimal hyperplane; they are considered as the borderline cases in the decision function while building the model. The distance found between the two classes is named “the margin”, explained by \( \frac{2}{||w||^2} \). To locate the maximal margin is the same as minimizing \( ||w|| \) which can be calculated using quadratic programming, and the optimal hyperplane can be explained by: \( w = \sum \alpha_i y_i x_i \) (Meyer, 2015).](image)

**Figure 7: The bounding planes of a linear SVM**
In real word, the data cannot always be separated that easy, as the separation task does not usually follow a linear trend. However, the separation might be easier using circles or curves for example but again finding the optimal curve is difficult and computationally expensive. The support Vector Machine algorithm achieves such a complex problem by relying on the kernel trick. In order to achieve that, it firsts translates the data from its input space to a higher dimension. This translation step is essential as it makes it a lot easier to disparte the data compared to the original space. This mapping procedure requires the use of Kernel induced features. A kernel function is a method that handles the dot product of the data. The optimal hyperplane can be explained as \( f(x) = \sum a_i y_i K(x_i, x) + b \) (Meyer, 2015).

Normalized Polynomial Kernel (NP), Polynomial Kernel (PK), Radial Basis Function Kernel (RBF) and Pearson VII Function- Based Universal Kernal (PUK) are some famous kernel functions to solve nonlinear classification problems.

Figure 8 : Choosing the hyperplane that maximizes the margin

Figure 9 : Separating the Data in a Feature Space
Artificial Neural Network (ANN)

An Artificial Neural Network (ANN) refers to an information processing paradigm that is inspired by the way the nervous system (the brain) processes information (Choy, Lee, and Lo, 2003).

The ANN consists of a large number of highly interconnected neurons that work together to solve various problems. Just like the way people are on a continuous learning process, the ANN is designed for specific applications, e.g. pattern recognition or even data classification, through a learning process. In the biological system, learning involves the adjustment of the synaptic contacts that exists between the neurons. Neural network simulation seems to be a new phenomenon. However, it is worth noting that the field became into existences even before the advent of the computers and had survived many eras and set back (Choy, Lee, and Lo, 2003).

In the world today, many significant advances, especially in the scientific field, have been aided by the use of computer emulations. Neural network with their extraordinary abilities to derive meaning from complicated data can be used to extract patterns and solve complex problems. Neural networks use a different approach to problem solving than the common computers (Sonali, Maind, and Wankar, 2014).

One thing about the conventional computers is that they follow an algorithmic approach. In this case, they follow a set of procedure to solve a given problem. In an event, where the procedures that computer needs to follow to solve certain problems, are uncertain, the computer cannot solve the problem. This restricts the computer to only preprogramed instructions that it already understands. The computers would be much useful if they had the capability of solving the problems that we do not have the knowledge of how to solve. As for the neural networks, their manner of solving problems is similar to that of the human brain; they learn by example. They do not need preprogrammed to carry out their functions. On the other hand, one of the drawbacks is that the network’s operation may sometimes be unpredictable (due to its ability to solve problems by itself).

Neural networks have been applied to a broad range of data-intensive applications; some of the examples based on previous success include its application in the medical field as in medical diagnosis (Sonali, Maind, and Wankar, 2014).

Multi-Layer Perceptron

Multi-layer perceptron (MLP) refers to a feed forward neural network with multiple layers between the input and the output. A feed forward layer implies that data flow in one direction from the input to the output layer (Taher et al., 2012). They can be used to solve problems which are linearly inseparable. Such a network is often trained with the back propagation learning algorithm. The diagram below shows the perceptron neural network architecture.
The Hidden layer is a block that will sum the given set of the weighted inputs (Ng, 2011). What follows after this is that it passes in the summed response to a non-linear function to create an output node response (the Hidden layer). The bias Unit, which is constant offset is added to each node for processing. The Hidden layer (With the help of the Sigmoid-type transfer functions) can also be used to determine different functions provided that there are enough neurons in the hidden layer (Ng, 2011). However, it cannot be estimated regarding the number of layers needed for optimum performance.

Typically, neurons have more than one input. In our case, the individual inputs P1, P2, P3, P4, p5, P6, P7 are each weighted to the equivalent W1,1, W1,2, W1,3, W1,4, W1,5, W1,6, W1,7 of the weight matrix W.

The neuron has a bias b. The bias is summed up with the weighted inputs to form the gross input n

\[ n = W1,1P1 + ... + W1,7P7 \]

This can be expressed as matrix form as follows:

\[ n = WP + b \]

The matrix W for single neuron case implies only one row and, the elements of the output layer can be written as:

\[ a = f(WP + b) \]
**K-Nearest Neighbour Algorithm (KNN)**

KNN is a form of instance-based learning, also known as lazy learning. The algorithm is estimated locally, and all calculation is conceded until classification. It’s known that kNN algorithm is considered one of the easiest of all machine learning algorithms in terms of complexity and implementation (IBM knowledge center, 2013).

KNN refers to an algorithm classifier, which works by comparing the similarity between neighbour-instances. This method is used to differentiate between instances without following a similarity routine nor a matching technique between data. It relies on the distance measure between two cases; small distances mean more similarity, while different cases usually fall far from each other. Hence, classifying two cases entails the calculation of the distance between them (IBM knowledge center, 2013). A common term used in KNN algorithm is “Neighbors”, it refers to instances/records that stand next to each other. When a new case (unknown) is given, the algorithm starts by computing the separation distance from each of the cases and the new one. Then it assigns the new instance to the appropriate class (the class label that most number of close neighbours belongs to).

The function most important variable is “k” which determines the number of other instances taken into account while classifying; in other words, it answers the question of how many neighbours a new instance has. The variation in “k” value has powerful effect on KNN performance; if the value is big, everything is more accurate and classified correctly, if the value is low, it will result in more inaccurate data and unstable decision boundaries in classification (Lavrenko and Goddard, 2014). Apart from “k”, another important factor is how the distance is calculated. The most common functions used are (KNN classification, no date):

\[
\text{Euclidean} \quad \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2} \\
\text{Manhattan} \quad \sum_{i=1}^{k} |x_i - y_i| \\
\text{Minkowski} \quad \left(\sum_{i=1}^{k} (|x_i - y_i|)^q\right)^{1/q}
\]

**WEKA**

Weka is a powerful machine learning/data mining toolkit; it stands for Waikato Environment for Knowledge Analysis (WEKA), created and produced in the University of Waikato in New Zealand. Weka is free open source software that runs on any operating system (Windows, Mac, Linux) and doesn’t really need any former programming experience.

It mainly contains four extensions: GUI, a command line, an Experimenter and a Knowledge Flow. Using the WEKA toolkit, it is easy to do some data mining with the dataset provided by the user and easy to use learning algorithms on text files. It gives a variety of reading data in many formats and from numerous workplaces like files and database. The dataset can be read as both: .arff or .csv extensions. It contains many excellent packages that support classification, clustering, data visualization, feature selection techniques. The next sections highlight the packages used throughout this work (Machine learning project at the university of Waikato in New Zealand, no date).
**SVM in Weka**

Weka.Classifiers.functions: SMO package in weka, uses John Platt’s sequential minimal optimization method for support vector machine (SVM) classification. The process is made by providing a substitute to all missing data and changing nominal features into binary ones. Furthermore, it arranges all attributes by default (Concerning this situation, the coefficients in the output are defined on the normalization of the data, not the native data). Pairwise classification is the method used to overcome multi-class tasks. The functions used the polynomial function as a default kernel (PK); default values were chosen for classification values.

![Figure 12: SMO parameters](image)

**ANN in Weka**

A Classifier that uses back propagation to classify instances. This network can be built by hand, created by an algorithm or both. The network can also be monitored and modified during training time. The nodes in this network are all sigmoid (except for when the class is numeric in which case the output nodes become un-thresholded linear units). Learning rate used is 0.3, the momentum is 0.2 and training Time used is 500.

![Figure 13: Multi-layer Perceptron parameters](image)
**KNN in Weka**

The dataset was also tested through the weka KNN implementation using `package weka .classifiers. lazy. IBk`. IBK stands for instance-based-learning. The parameters used for the classification was the default values: 1 for the number of neighbours and Euclidian for the distance measurement.

![IBK parameters](image)

*Figure 14: IBK parameters*
Results

The classification problem is divided into sub-problems depending on the position of the letter in the word. In the dataset /r/-LDFW there are five classes (correct, substitution with G, Substitution with L, omission and Addition) while in /r/-LDFE and /r/-LDMW, there are only four classes; No omission, addition class in /r/-LDMW and /r/-LDFE respectively.

The same classification problems were repeated three times trying a different classifier each time; SVM, ANN and KNN.

All classifiers were evaluated by computing the statistical parameters of sensitivity, specificity and classification accuracy.

Classification Performance measures

Accuracy is a common measure of detecting performance, incorporating both a classifier’s sensitivity and specificity; it corresponds to the number of correctly classified patterns over the total number of patterns. It is defined by (Sokolova and Lapalme, 2009):

\[
\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}
\]

Where sensitivity, also called recall, is the number of correctly detected positive patterns/total number of actual positive patterns; it shows the effectiveness of a classifier to identify positive labels. While specificity is the effectiveness of a classifier to identify negative labels; it is the number of correctly detected negative patterns/total number of actual negative patterns. TP, FP, TN, FN Correspond to the elements of the resulting confusion matrix of the classification (Sokolova and Lapalme, 2009).

\[
\text{Sensitivity} = \frac{TP}{TP + FN}
\]

\[
\text{Specificity} = \frac{TN}{TN + FP}
\]

However, the above formulas apply for a binary classification problem where the model classifies the data into only two classes. In multi-classification problems the input is to be classified into one, and only one, of l non-overlapping classes (Sokolova and Lapalme, 2009). The accuracy hence represents the average per-class effectiveness of a classifier and its calculation can follow this equation:

\[
\text{Accuracy} = \frac{\sum_{i=1}^{l} TP_i + TN_i}{l}
\]

Cross-Validation

In order to assess the performance results of a machine learning algorithms, the classification model built needs to makes accurate predictions on the test set as well as on the training data. Training set
accuracy is not a good indication, however, a good indicator would be how well the classifier will perform when classifying new data outside of the training set. The validation data have to represent all of the range of inputs the classifier is likely to encounter, in other words it needs to be diverse. It is generally better to randomly select the validation examples from the existing collection of data.

There are different approaches to selecting the training and validation sets. One simple approach is to randomly select, e.g., 80% of the existing data to use for training and 20% to use for testing. The cross-validation process provides a much more accurate result of the model’s true accuracy. In cross-validation, the data is divided into a large training set and a smaller validation set, then train on the training set and use the validation set to measure our accuracy.

In this procedure, the data is divided into k parts/folds. The classifier then runs for ‘k’ rounds of cross-validation. In each round, one of the folds is used for validation, and the remaining folds for training. After training, the accuracy on each validation data is measured and averaged over the k rounds to get final cross-validation accuracy. A common value of k is 10. Figure 15 shows the detailed cross-validation process for 10-fold cross validation.

![Validation Set and Training Set](image)

**Figure 15: fold cross-validation.**

In this work, the performance of each classifier is assessed using cross-validation with 10 folds. The value of 10 was the default value used by Weka for the cross validation option. The data is divided equally into 10 subsets, a single subsample is retained as the test set for validating the model, and the remaining 9 subsamples are used as training data. The cross-validation process is then repeated 10 times (the *folds*), with each of the 10 subsets used exactly once as the validation data.

**Classification Results**

The detailed weka results for each classifier per sub-problem are demonstrated below;
### KNN

<table>
<thead>
<tr>
<th>r/- Letter disorder on the beginning of Words (r/-LDFW)</th>
<th>r/- Letter disorder on the middle of Words (r/-LDMW)</th>
<th>r/- letter disorder at the end of Words (r/-LDFE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>=== Classifier model (full training set) ===</td>
<td>=== Classifier model (full training set) ===</td>
<td>=== Classifier model (full training set) ===</td>
</tr>
<tr>
<td>IB1 instance-based classifier</td>
<td>IB1 instance-based classifier</td>
<td>IB1 instance-based classifier</td>
</tr>
<tr>
<td>using 1 nearest neighbour(s) for classification</td>
<td>using 1 nearest neighbour(s) for classification</td>
<td>using 1 nearest neighbour(s) for classification</td>
</tr>
<tr>
<td>Time taken to build model: 0 seconds</td>
<td>Time taken to build model: 0 seconds</td>
<td>Time taken to build model: 0 seconds</td>
</tr>
<tr>
<td>=== Stratified cross-validation ===</td>
<td>=== Stratified cross-validation ===</td>
<td>=== Stratified cross-validation ===</td>
</tr>
<tr>
<td>=== Summary ===</td>
<td>=== Summary ===</td>
<td>=== Summary ===</td>
</tr>
<tr>
<td>Correctly Classified Instances</td>
<td>Correctly Classified Instances</td>
<td>Correctly Classified Instances</td>
</tr>
<tr>
<td>1257</td>
<td>1058</td>
<td>1007</td>
</tr>
<tr>
<td>Incorrectly Classified Instances</td>
<td>Incorrectly Classified Instances</td>
<td>Incorrectly Classified Instances</td>
</tr>
<tr>
<td>243</td>
<td>142</td>
<td>193</td>
</tr>
<tr>
<td>Total Number of Instances</td>
<td>Total Number of Instances</td>
<td>Total Number of Instances</td>
</tr>
<tr>
<td>1500</td>
<td>1200</td>
<td>1200</td>
</tr>
<tr>
<td>=== Detailed Accuracy by Class ===</td>
<td>=== Detailed Accuracy By Class ===</td>
<td>=== Detailed Accuracy By Class ===</td>
</tr>
<tr>
<td>TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area</td>
<td>TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area</td>
<td>TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area</td>
</tr>
<tr>
<td>0.82  0.07  0.745  0.82  0.781  0.876</td>
<td>0.903  0.019  0.941  0.903  0.922  0.94</td>
<td>0.813  0.058  0.824  0.813  0.819  0.868</td>
</tr>
<tr>
<td>Addition</td>
<td>Addition</td>
<td>Correct</td>
</tr>
<tr>
<td>0.833  0.048  0.812  0.833  0.822  0.886</td>
<td>0.85  0.068  0.807  0.85  0.828  0.886</td>
<td>0.84  0.044  0.863  0.84  0.851  0.899</td>
</tr>
<tr>
<td>Correct</td>
<td>Correct</td>
<td>Omission</td>
</tr>
<tr>
<td>0.927  0.02  0.921  0.927  0.924  0.954</td>
<td>0.92  0.036  0.896  0.92  0.908  0.944</td>
<td>0.837  0.059  0.826  0.837  0.831  0.888</td>
</tr>
<tr>
<td>Omission</td>
<td>Omission</td>
<td>SubG</td>
</tr>
<tr>
<td>0.8  0.013  0.941  0.8  0.865  0.889</td>
<td>0.853  0.036  0.889  0.853  0.871  0.904</td>
<td>0.867  0.053  0.844  0.867  0.855  0.9</td>
</tr>
<tr>
<td>SubG</td>
<td>Weighted Avg.</td>
<td>SubL</td>
</tr>
<tr>
<td>0.81  0.052  0.797  0.81  0.803  0.875</td>
<td>0.882  0.039  0.883  0.882  0.882  0.918</td>
<td>Weighted Avg.</td>
</tr>
<tr>
<td>SubL</td>
<td>SubL</td>
<td>0.839  0.054  0.839  0.839  0.839  0.889</td>
</tr>
<tr>
<td>Weighted Avg.</td>
<td>Confusion Matrix</td>
<td>=== Confusion Matrix ===</td>
</tr>
<tr>
<td>0.838  0.041  0.843  0.838  0.839  0.896</td>
<td>a  b  c  d  &lt;-- classified as</td>
<td>a  b  c  d  &lt;-- classified as</td>
</tr>
<tr>
<td>=== Confusion Matrix ===</td>
<td>271  15  6  8</td>
<td>a = Addition</td>
</tr>
<tr>
<td></td>
<td>8  255  16  21</td>
<td>b = Correct</td>
</tr>
<tr>
<td></td>
<td>1  20  276  3</td>
<td>c = SubG</td>
</tr>
<tr>
<td></td>
<td>8  26  10  256</td>
<td>d = SubL</td>
</tr>
</tbody>
</table>

### Table 3: KNN classification results

The table above presents the performance metrics of the KNN classification model for three different types of letter disorders: on the beginning (r/-LDFW), middle (r/-LDMW), and end (r/-LDME) of words. The metrics include TP Rate, FP Rate, Precision, Recall, F-Measure, ROC Area, and several error rates such as Kappa statistic, Mean absolute error, and Root relative squared error. The table also includes a confusion matrix for each type of disorder, detailing the classification results. The overall accuracy and other evaluation measures are provided for each classification, highlighting the effectiveness of the KNN model in classifying these different types of letter disorders.
SMO

<table>
<thead>
<tr>
<th>/r/- Letter disorder on the beginning of Words ((/r)-LDFW)</th>
<th>/r/- Letter disorder on the middle of Words ((/r)-LDMW)</th>
<th>/r/- letter disorder at the end of Words ((/r)-LDFE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time taken to build model: 0.21 seconds</td>
<td>Time taken to build model: 0.16 seconds</td>
<td>Time taken to build model: 0.27 seconds</td>
</tr>
<tr>
<td>=== Stratified cross-validation ===</td>
<td>=== Stratified cross-validation ===</td>
<td>=== Stratified cross-validation ===</td>
</tr>
<tr>
<td>=== Summary ===</td>
<td>=== Summary ===</td>
<td>=== Summary ===</td>
</tr>
<tr>
<td>Correctly Classified Instances</td>
<td>Correctly Classified Instances</td>
<td>Correctly Classified Instances</td>
</tr>
<tr>
<td>953</td>
<td>833</td>
<td>716</td>
</tr>
<tr>
<td>Incorrectly Classified Instances</td>
<td>Incorrectly Classified Instances</td>
<td>Incorrectly Classified Instances</td>
</tr>
<tr>
<td>547</td>
<td>367</td>
<td>484</td>
</tr>
<tr>
<td>Kappa statistic</td>
<td>Kappa statistic</td>
<td>Kappa statistic</td>
</tr>
<tr>
<td>0.5442</td>
<td>0.5922</td>
<td>0.4622</td>
</tr>
<tr>
<td>Mean absolute error</td>
<td>Mean absolute error</td>
<td>Mean absolute error</td>
</tr>
<tr>
<td>0.2645</td>
<td>0.2845</td>
<td>0.3036</td>
</tr>
<tr>
<td>Root mean squared error</td>
<td>Root mean squared error</td>
<td>Root mean squared error</td>
</tr>
<tr>
<td>0.352</td>
<td>0.3618</td>
<td>0.3871</td>
</tr>
<tr>
<td>Relative absolute error</td>
<td>Relative absolute error</td>
<td>Relative absolute error</td>
</tr>
<tr>
<td>82.6417 %</td>
<td>75.8704 %</td>
<td>80.963 %</td>
</tr>
<tr>
<td>Root relative squared error</td>
<td>Root relative squared error</td>
<td>Root relative squared error</td>
</tr>
<tr>
<td>87.9991 %</td>
<td>83.559 %</td>
<td>89.3909 %</td>
</tr>
<tr>
<td>Total Number of Instances</td>
<td>Total Number of Instances</td>
<td>Total Number of Instances</td>
</tr>
<tr>
<td>1500</td>
<td>1200</td>
<td>1200</td>
</tr>
<tr>
<td>=== Detailed Accuracy by Class ===</td>
<td>=== Detailed Accuracy by Class ===</td>
<td>=== Detailed Accuracy by Class ===</td>
</tr>
<tr>
<td>TP Rate</td>
<td>TP Rate</td>
<td>TP Rate</td>
</tr>
<tr>
<td>0.517</td>
<td>0.72</td>
<td>0.707</td>
</tr>
<tr>
<td>FP Rate</td>
<td>0.113</td>
<td>0.169</td>
</tr>
<tr>
<td>Precision</td>
<td>0.534</td>
<td>0.582</td>
</tr>
<tr>
<td>Recall</td>
<td>0.517</td>
<td>0.744</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.525</td>
<td>0.893</td>
</tr>
<tr>
<td>ROC Area</td>
<td>0.749</td>
<td>0.819</td>
</tr>
<tr>
<td>Class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Addition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.523</td>
<td>0.6</td>
<td>0.533</td>
</tr>
<tr>
<td>Omission</td>
<td>0.777</td>
<td>0.705</td>
</tr>
<tr>
<td>Correct</td>
<td>0.777</td>
<td>0.777</td>
</tr>
<tr>
<td>0.79</td>
<td>0.705</td>
<td>0.911</td>
</tr>
<tr>
<td>SubG</td>
<td>0.79</td>
<td>0.745</td>
</tr>
<tr>
<td>0.57</td>
<td>0.553</td>
<td>0.806</td>
</tr>
<tr>
<td>SubL</td>
<td>0.57</td>
<td>0.562</td>
</tr>
<tr>
<td>Weighted Avg.</td>
<td>0.635</td>
<td>0.635</td>
</tr>
<tr>
<td>0.635</td>
<td>0.635</td>
<td>0.634</td>
</tr>
<tr>
<td>0.635</td>
<td>0.634</td>
<td>0.838</td>
</tr>
<tr>
<td>=== Confusion Matrix ===</td>
<td>=== Confusion Matrix ===</td>
<td>=== Confusion Matrix ===</td>
</tr>
<tr>
<td>a b c d e &lt;-- classified as</td>
<td>a b c d e &lt;-- classified as</td>
<td>a b c d e &lt;-- classified as</td>
</tr>
<tr>
<td>155 34 22 33 56</td>
<td>a = Addition</td>
<td>216 34 28 22</td>
</tr>
<tr>
<td>53 157 13 27 50</td>
<td>b = Correct</td>
<td>29 180 31 60</td>
</tr>
<tr>
<td>21 26 233 10 10</td>
<td>c = Omission</td>
<td>18 24 238 20</td>
</tr>
<tr>
<td>23 14 4 237 22</td>
<td>d = SubG</td>
<td>18 69 14 199</td>
</tr>
</tbody>
</table>

Table 4: SVM classification results
### Multi-layer Perceptron

<table>
<thead>
<tr>
<th>/r/- Letter disorder on the beginning of Words</th>
<th>/r/- Letter disorder on the middle of Words</th>
<th>/r/- letter disorder at the end of Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>(r/-LDFW)</td>
<td>(r/-LDMW)</td>
<td>(r/-LDFE).</td>
</tr>
<tr>
<td>Time taken to build model: 14.91 seconds</td>
<td>Time taken to build model: 11.86 seconds</td>
<td>Time taken to build model: 11.12 seconds</td>
</tr>
<tr>
<td>=== Stratified cross-validation ===</td>
<td>=== Stratified cross-validation ===</td>
<td>=== Stratified cross-validation ===</td>
</tr>
<tr>
<td>=== Summary ===</td>
<td>=== Summary ===</td>
<td>=== Summary ===</td>
</tr>
<tr>
<td>Correctly Classified Instances 1070</td>
<td>Correctly Classified Instances 984</td>
<td>Correctly Classified Instances 861</td>
</tr>
<tr>
<td>Incorrectly Classified Instances 430</td>
<td>Incorrectly Classified Instances 216</td>
<td>Incorrectly Classified Instances 339</td>
</tr>
<tr>
<td>Kappa statistic 0.6417</td>
<td>Kappa statistic 0.76</td>
<td>Kappa statistic 0.6233</td>
</tr>
<tr>
<td>Mean absolute error 0.1261</td>
<td>Mean absolute error 0.1036</td>
<td>Mean absolute error 0.1514</td>
</tr>
<tr>
<td>Root mean squared error 0.313</td>
<td>Root mean squared error 0.2722</td>
<td>Root mean squared error 0.3407</td>
</tr>
<tr>
<td>Relative absolute error 39.4133 %</td>
<td>Relative absolute error 27.6328 %</td>
<td>Relative absolute error 40.3758 %</td>
</tr>
<tr>
<td>Root relative squared error 62.8703 %</td>
<td>Root relative squared error 78.6884 %</td>
<td>Root relative squared error</td>
</tr>
<tr>
<td>Total Number of Instances 1500</td>
<td>Total Number of Instances 1200</td>
<td>Total Number of Instances 1200</td>
</tr>
<tr>
<td>=== Detailed Accuracy By Class ===</td>
<td>=== Detailed Accuracy By Class ===</td>
<td>=== Detailed Accuracy By Class ===</td>
</tr>
<tr>
<td>TP Rate 0.643 0.1 0.617 0.643 0.63 0.846</td>
<td>TP Rate 0.887 0.046 0.866 0.887 0.876 0.977</td>
<td>TP Rate 0.733 0.081 0.751 0.733 0.742 0.897</td>
</tr>
<tr>
<td>FP Rate 0.613 0.076 0.669 0.613 0.64 0.855</td>
<td>Precision 0.743 0.063 0.796 0.743 0.769 0.922</td>
<td>Precision 0.69 0.106 0.685 0.69 0.688 0.873</td>
</tr>
<tr>
<td>0.72 0.088 0.673 0.72 0.069 0.903 0.957</td>
<td>Recall 0.883 0.062 0.826 0.883 0.853 0.965</td>
<td>Recall 0.723 0.093 0.721 0.723 0.722 0.896</td>
</tr>
<tr>
<td>Omission 0.757 0.058 0.767 0.757 0.762 0.927</td>
<td>SubG 0.767 0.069 0.788 0.767 0.777 0.927</td>
<td>SubG 0.723 0.097 0.714 0.723 0.719 0.886</td>
</tr>
<tr>
<td>Weighted Avg. 0.713 0.072 0.715 0.713 0.713</td>
<td>SubL 0.82 0.06 0.819 0.82 0.819 0.948</td>
<td>SubL 0.718 0.094 0.718 0.718 0.718 0.888</td>
</tr>
<tr>
<td>=== Confusion Matrix ===</td>
<td>=== Confusion Matrix ===</td>
<td>=== Confusion Matrix ===</td>
</tr>
<tr>
<td>a b c d e</td>
<td>a b c d</td>
<td>a b c d</td>
</tr>
<tr>
<td>193 32 11 27 37</td>
<td>12 223 24 41</td>
<td>220 25 24 31</td>
</tr>
<tr>
<td>44 184 15 21 36</td>
<td>14 7 265 14</td>
<td>24 207 37 32</td>
</tr>
<tr>
<td>19 10 250 7 14</td>
<td>15 39 16 230</td>
<td>19 40 217 24</td>
</tr>
<tr>
<td>25 28 10 14 216</td>
<td>a = Addition</td>
<td>c = SubG</td>
</tr>
<tr>
<td>e = SubL</td>
<td>b = Correct</td>
<td>d = SubL</td>
</tr>
</tbody>
</table>

**Table 5 : Multi-Layer Perceptron classification results**
The best overall classification accuracies obtained by averaging all accuracies over all classification sub-problems are 85.25%, 64.18% and 74.99% for KNN, SVM and ANN respectively.

The results per sub-problem are shown in the following table.

<table>
<thead>
<tr>
<th>Sub-Category</th>
<th>KNN</th>
<th>SVM</th>
<th>ANN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct Speech</td>
<td>83.3</td>
<td>85</td>
<td>81.3</td>
</tr>
<tr>
<td>Sub- (/r/ with /ɣ/ (/g/))</td>
<td>80</td>
<td>92</td>
<td>83.6</td>
</tr>
<tr>
<td>Sub- (/r/ with /l/ )</td>
<td>81</td>
<td>85.3</td>
<td>86.6</td>
</tr>
<tr>
<td>Omission disorder</td>
<td>92.3</td>
<td>84</td>
<td>77.6</td>
</tr>
<tr>
<td>Addition disorder</td>
<td>82</td>
<td>90.3</td>
<td>51.6</td>
</tr>
<tr>
<td><strong>OVERALL</strong></td>
<td><strong>83.72%</strong></td>
<td><strong>88.15%</strong></td>
<td><strong>83.875%</strong></td>
</tr>
</tbody>
</table>

Table 6: Overall classification results

**Analysis**

From the table above, it can be observed that K Nearest Neighbour had a total overall accuracy, averaged over all data sub-sets, of 85.25%. It outperformed the other classifiers with almost 10-20% difference. Support Vector Machine had the least classification accuracy with only 65% while Multi-Layer Perceptron had an accuracy of almost 75%.

Another approach is to perform the analysis on the disorder level throughout all the data sub-sets. Figure 16 shows the average accuracy obtained for each type of disorder separately.
It can be concluded that the addition and omission disorders models usually have good classification accuracy. Such results are somehow logical since the repetition of the letter /r/ in the pronounced word, or its complete disappearance from the pronounced word, ensures that the overlap between disorders is minimal and hence the classification becomes easier.

On the other hand, it was regularly observed in the three sub-problems that the weakest results were obtained for both correct pronunciation and diagnosing the disorder characterized by switching the letter /r/ with the letter /l/, especially when the disorder is in the middle of or at the end of the word. Such overlap is due to the shared points of articulation; both /r/ and /l/ share the same points of articulation or “output”. Such a phenomena is called the “like letter” in the field of phonetics, meaning that the two letters are very similar, which ensures that interference between them is very likely to occur. The process of phonetically distinguishing them is difficult and also their automated classification (Sokolova and Lapalme, 2009).
CHAPTER FIVE

CONCLUSION AND FUTURE WORK
Conclusion and Future Work

Conclusion

In this dissertation, MFCC was evaluated as feature extraction method to analyse the recorded speech samples. Also, different classification methods were discussed, the best results were achieved by Ibk method (KNN) with an 85.25% accuracy overall using 1 for the number of neighbors and Euclidian equation for the distance measurements. Multi-Layer Perceptron (ANN) showed an average of 74.99% accuracy using a learning rate of 0.3 and momentum 0.2. SMO (SVM) showed the least accuracy of 64.18% using Polynomial Kernel (PK).

Generally, the results require improvement; more features need to be investigated and even other classification algorithms. Even for the current model, it needs to be validated on a larger dataset and a crucial step to fully declare the model as an automated tool is to include all Arabic letters.

Future Work

Based on previous research, the greater intended goal was to create a mobile/tablet application for speech disorder diagnosis and treatment.

To achieve this, there are two main steps to be taken:

1- More Data:

More data should be acquired to cover all Arabic letters alphabet. The choice of these words should be the same as the traditional speech session protocol followed by the therapist.

In Egypt, there was recently a standard Articulation test followed by phonetics specialties: the MAAT.

*Mansoura Arab Articulation Test (MAAT)*

It is the result of a recent pilot study that developed an Arabic verbalization test utilizing well known, socially based and outwardly straightforward words. It is utilized as a basis for contrasting phonemes of both ordinary also, phonologically disarranged Arabic-talking youngsters (Abou-Elsaad, Baz, and El-Banna, 2009). MAAT is a valid and reliable test that can be applied to collect the phonetic inventory of Arabic-speaking young children with phonological and speech disorders. The table below show the words used in the test and its equivalent English synonyms:
Figure 17: Mansoura Arab Articulation Test

### 2- Application Framework:

The application will follow the traditional picture-based technique. Moreover, it should be more fun, as the app will take a game style and not a therapy. It should ask the child to pronounce the word, upon correct pronunciation, the kid is awarded points. A wrong pronunciation is not awarded but also implies that the app categorizes the disorder into a certain diagnosis and the choice of the following word is based on a defined therapy. The design of the app will be very similar to tabby talk (Shahin et al., 2015), but the pictures will be inspired from the MAAT.
References


APPENDICES
Appendix A: Code Screen Shots

This a screenshot of the function readAllIntoArray used in Feature extraction

This a screenshot of the function EndAnalysis used in Feature extraction
The following 4 pictures are the screenshots of melcepst function used in feature extraction.
if any(w=='P')
    vr=(4:1:4)/60;
    af=(1:1:1-1)/2;
    vv=ones(5,1);
    cx=c([vv,1]); c=[nn*[vv,1]];
    vx=reshape(filter(vf,1,cx(:,1)),nf+10,nc);
    vx(1:8,:)=[];
    ax=reshape(filter(af,1,vx(:,1)),nf+2,nc);
    ax(1:2,:)=[];
    vx([1 nf+2],1)=[];
    if any(w=='P')
        c=[c vx ax];
    else
        c=[c ax];
    end
else
    if any(w=='D')
        vr=(0:1:4)/60;
        vv=ones(4,1);
        cx=c([vv,1]); c=[nn*[vv,1]];
        vx=reshape(filter(vf,1,cx(:,1)),nf+8,nc);
        vx(1:8,:)=[];
        c=[c vx];
    end
end
if normout<1
    [mr,nc]=size(c);
    t=((0:nf-1)*inc+(n-1)/2)/fs;
    cl=(1:nc)-any(w=='E')-any(w=='U');
    lnh = imagesc(tc/fs,cl,c.');
Appendix B: Ethics Form

When undertaking a research or enterprise project, Cardiff Met staff and students are obliged to complete this form in order that the ethics implications of that project may be considered.

If the project requires ethics approval from an external agency (e.g., NHS), you will not need to seek additional ethics approval from Cardiff Met. You should however complete Part One of this form and attach a copy of your ethics letter(s) of approval in order that your School has a record of the project.

The document Ethics application guidance notes will help you complete this form. It is available from the Cardiff Met website. The School or Unit in which you are based may also have produced some guidance documents, please consult your supervisor or School Ethics Coordinator.

Once you have completed the form, sign the declaration and forward to the appropriate person(s) in your School or Unit.

PLEASE NOTE:
Participant recruitment or data collection MUST NOT commence until ethics approval has been obtained.

PART ONE

<p>| Name of applicant:                  | Khaled Seddik Tawfik                                      |
| Supervisor (if student project):    | Ambikesh Jayal                                           |
| School / Unit:                      | CSM                                                      |
| Student number (if applicable):     | St20072973                                               |
| Programme enrolled on (if applicable): | BSc (Hons) Computing                                    |
| Project Title:                      | Towards the development of Computer aided speech therapy tool in Arabic language using artificial intelligence |
| Expected start date of data collection: | 19 March 2016                                           |
| Approximate duration of data collection: | 2 weeks                                                 |
| Funding Body (if applicable):       | N/A                                                     |
| Other researcher(s) working on the project: | N/A                                                      |
| Will the study involve NHS patients or staff? | No                                                      |</p>
<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Will the study involve taking samples of human origin from participants?</td>
<td>No</td>
</tr>
<tr>
<td>Does your project fall entirely within one of the following categories:</td>
<td></td>
</tr>
<tr>
<td>Paper based, involving only documents in the public domain</td>
<td>Yes</td>
</tr>
<tr>
<td>Laboratory based, not involving human participants or human tissue samples</td>
<td>No</td>
</tr>
<tr>
<td>Practice based not involving human participants (eg curatorial, practice audit)</td>
<td>No</td>
</tr>
<tr>
<td>Compulsory projects in professional practice (eg Initial Teacher Education)</td>
<td>No</td>
</tr>
<tr>
<td>A project for which external approval has been obtained (e.g., NHS)</td>
<td>No</td>
</tr>
</tbody>
</table>

If you have answered YES to any of these questions, expand on your answer in the non-technical summary. If you have answered NO to all of these questions, you must complete Part 2 of this form.

In no more than 150 words, give a non-technical summary of the project

My project tackles the problem of speech articulation disorder; this involves difficulty producing one or few speech sounds typical at young age. Interactive and automatic speech therapy tools offer a practical and cost-effective alternative to regular therapist visits. Such speech therapy tools do exist but the novelty of the project relies in developing a similar system for Arabic language. My project will be concerned mainly with the first stage of the therapy; the diagnosis phase. It should be able to give a reliable decision about the type of disorder good of an input speech. To achieve such a goal, my model is a typical AI problem; data acquisition, feature extraction and a classifier. For the data acquisition phase, I will use public data and dummy data created by myself.

DEMONSTRATION:
I confirm that this project conforms with the Cardiff Met Research Governance Framework

I confirm that I will abide by the Cardiff Met requirements regarding confidentiality and anonymity when conducting this project.

STUDENTS: I confirm that I will not disseminate any material produced as a result of this project without the prior approval of my supervisor.

Signature of the applicant:  
Date:  

FOR STUDENT PROJECTS ONLY

Name of supervisor:  
Date:  

Signature of supervisor:
PART TWO

A RESEARCH DESIGN

A1 Will you be using an approved protocol in your project? | No
A2 If yes, please state the name and code of the approved protocol to be used

A3 Describe the research design to be used in your project

A4 Will the project involve deceptive or covert research? | No
A5 If yes, give a rationale for the use of deceptive or covert research

A6 Will the project have security sensitive implications? | No
A7 If yes, please explain what they are and the measures that are proposed to address them

B PREVIOUS EXPERIENCE

B1 What previous experience of research involving human participants relevant to this project do you have?

B2 Student project only

What previous experience of research involving human participants relevant to this project does your supervisor have?

C POTENTIAL RISKS

C1 What potential risks do you foresee?

---

1 An Approved Protocol is one which has been approved by Cardiff Met to be used under supervision of designated members of staff; a list of approved protocols can be found on the Cardiff Met website here
C2 How will you deal with the potential risks?

When submitting your application you **MUST** attach a copy of the following:

- All information sheets
- Consent/assent form(s)

An exemplar information sheet and participant consent form are available from the Research section of the Cardiff Met website.