REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Information Technologies

IDENTIFICATION OF THE SPEAKER'S VOICE USING MACHINE LEARNING

Noor Sabah Shandookh ASSAFI

Master`s Thesis

Supervisor

Asst. Prof. Dr. Oğuz KARAN

Istanbul, 2022

IDENTIFICATION OF THE SPEAKER'S VOICE USING MACHINE LEARNING

Noor Sabah Shandookh ASSAFI

Information Technologies

Master`s Thesis

ALTINBAŞ UNIVERSITY

2022

The thesis titled Identification of the SPEAKERS VOICE USING MACHINE LEARNING prepared by NOOR SABAH SHANDOOKH ASSAFI and submitted on 12/12/2022 has been **accepted unanimously** for the degree of Master of Science in Information Technologies.

Asst. Prof. Oğuz KARAN

The Supervisor

Thesis Defense Committee Members:

Asst. prof. Dr. Oğuz KARAN	Department of Software Engineering, Altınbaş University	_____
Asst. prof. Dr. Sefer KURNAZ	Department of Computer Engineering, Altınbaş University	_____
Assoc. prof. Dr. Adil Deniz DURU	Department of Trainer Education, Marmara University	_____

I hereby declare that this thesis meets all format and submission requirements of a Master's thesis.

Submission date of the thesis to Institute of Graduate Studies: ____/____/____

I hereby declare that all information presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Noor Sabah Shandookh ASSAFI

Signature

DEDICATION

I would like to thank Allah Almighty for my success in my studies. My dear parents, I know very well that you loved science and study, I am sure that you are now happy with my studies. Thank you for all you have done for me so that I can reach this level of study today. My husband, I would like to offer you all my thanks, love, and appreciation. Thank you for your support, standing by my side, and your great help for me.

ACKNOWLEDGEMENT

I would like to thank very much to my supervisor, Assistant Professor Asst. Prof. Dr. Oğuz KARAN. for his assistance throughout, the research period and for answering many questions and he was the best help for me throughout the study period. Without his time and help, this research would not have been completed and I am very grateful to him.

ABSTRACT

IDENTIFICATION OF THE SPEAKER'S VOICE USING MACHINE LEARNING

ASSAFI, Noor

M.Sc., Information Technologies, Altınbaş University,

Supervisor: Asst. Prof. Dr. Oğuz KARAN

Date: December / 2022

Pages: 69

The modern society of today has an ever-increasing demand for identity and safety measures, which has resulted in the development of a large number of different safety and identification measures. This demand is driving innovation in the field. Biometrics is a term that refers to many techniques that can be used to determine an individual's identity based on the unique characteristics that they have. The primary contribution that we have made with this paper is that we have designed and implemented a system that makes use of machine learning techniques in order to classify into labels features that have been collected from an audio file of a speaker. These features include the speaker's name, the speaker's gender, the speaker's age, and the speaker's location. The purpose of this research is to finally give a comprehensive understanding of the many approaches to Speech Recognition that are now accessible. This will be accomplished by performing a thorough examination of the relevant literature in order to achieve this goal.

Keywords: Support Victor Machine , Convolutional Neural Network, Machine Learning, Deep Learning.

TABLE OF CONTENTS

Pages

LIST OF FIGURES

ABBREVIATIONS

RNN	:	Recurrent Neural Networks
SVM	:	Support Vector Machine
FTT	:	Fast Fourier Transform
ML	:	Machine Learning
MFCC	:	Mel-frequency cepstral coefficients
CNN	:	Convolutional Neural Network
DCF	:	Discounted Cash-Flow
NB	:	Naïve bayes

1. INTRODUCTION

The use of automatic techniques for speaker recognition is not recurrent in today's world just for convenience and practicality, but mainly for the security that the system offers. Techniques for both speaker recognition and word or phoneme recognition have been used for at least 30 years, undergoing sudden changes in this period [1].

The human being has several inherent and unique characteristics, a good example for understanding is the fingerprint, or biometrics. However, their systems, although with great effectiveness nowadays, may present security flaws, or even inaccessibility for certain people. When we deal with voice, in addition to inferring a unique characteristic of each one, we take into account practicality and convenience (recurring trend of technological advancement), as well as a non-transferable means of identification. Therefore, in several applications, the use of such a feature is significantly more coherent, taking into account that it needs to be treated with extreme caution to mitigate errors or undesirable eventualities.

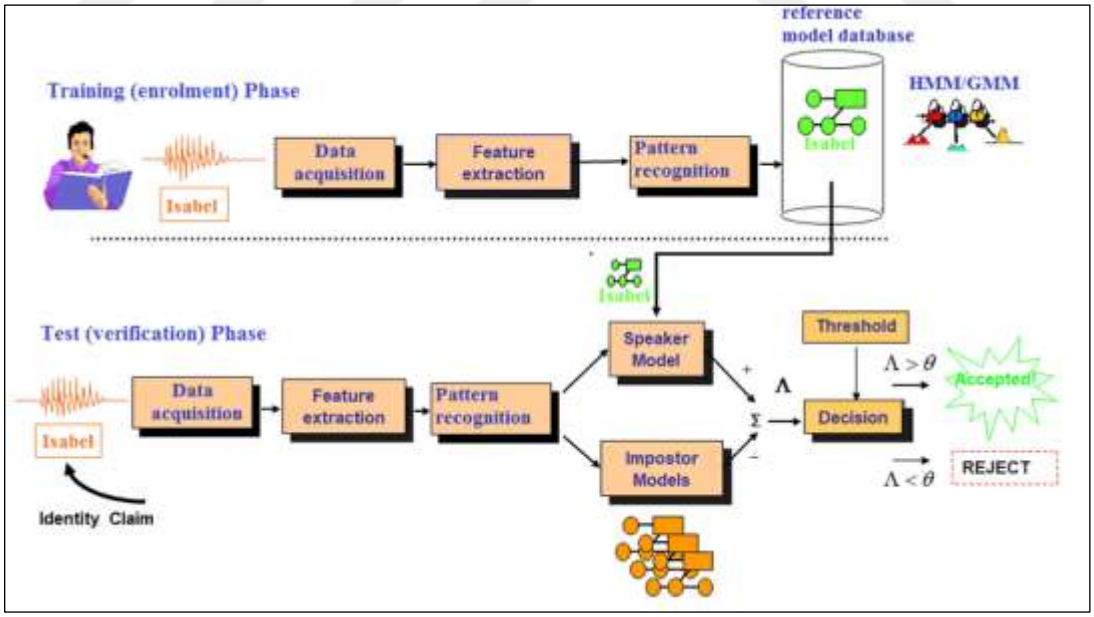

Figure 1.1: Automatic Techniques For Speaker Recognition.

What seems like an intuitive and trivial task for the human being, is a big challenge for the field of technology and signal processing. Word recognition or speaker recognition face problems that were never fully solved, but the improvement in efficiency in these times was noticeable.

The speech of a human being, due to variations and inflections, is a rather complex signal to be processed [20]. For this, we must teach a computer to extract characteristics of the voice of a certain person, and thus make a decision, which would basically be how that signal is classified in the application.

A basic and extremely important concept for intelligent systems is that of machine learning. This consists of teaching a system or a computer to find solutions to a given problem. More specifically, this solution is found by extracting patterns from input data and identifying them, in order to make a decision possible. Such a concept, despite seeming specific, is an offshoot of artificial intelligence and covers a large area of knowledge. Several applications use resources offered by machine learning, among them we have image processing, DNA classification sequence and speech recognition [9].

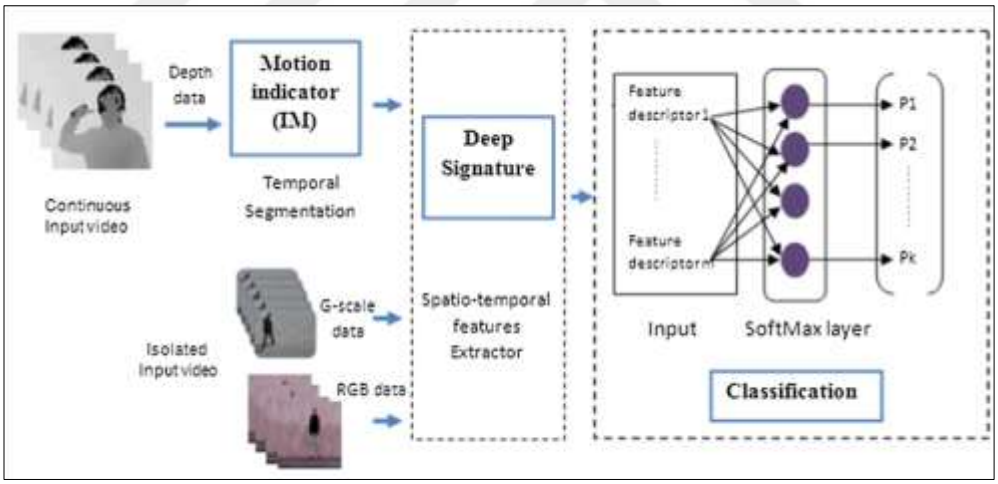

Figure 1.2 : Isolated recognition and continuous recognition.

Recognition of words or phonemes can be divided into two main strands: isolated recognition and continuous recognition. For isolated recognition systems, the user provides the system with isolated words, so that the system will relate these inputs to previously stored patterns. For continuous recognition systems, the user provides as input a sequence of words in the form of their natural speech, and the system must be able to recognize all of them [20]. This model is undoubtedly more complex, because before dealing with patterns previously informed to the System, it must initially provide the correct separation of words, which can be complex taking

into account the speech variations of each individual, and the difference in energy. that some words have at their beginning and end in pronunciation. On the other hand, continuous speech recognition is the most desired, due to the human being's natural way of communication.

Among the continuous and isolated recognition models, we have two more aspects, which deal with supervised and unsupervised systems. Supervised systems need training, that is, a previous data entry so that the labels are defined and the system which of these should sort the subsequent entries. In unsupervised systems, which are notoriously more complex, the system must identify patterns in the inputs so that a later input similar to one already provided can be identified [27].

In order to carry out an adequate recognition, then we need resources to train or test our system. The resources used in voice recognition are the cepstral coefficients, which consist of treating the voice signal in frequency using some mathematical manipulations to extract speech characteristics. Such coefficients proved to be very efficient for this type of application due to the ability to preserve information [23] [17]. These resources are used by different classification systems in recognition, which are basically how the computer or machine will make the decision (about which rules) [1].

A classification technique frequently used in the field of speech recognition in its beginnings is the Hidden Markov Model (HMM), which basically treats the voice signal as well as its variations (belonging to the same individual) in a statistical and temporal manner. This technique has a variety of works and diversified studies, but with little implementation in embedded systems made [7]. This reason is due to the need for a large database for training, considering in isolated systems that the error rate in this model is given by the number of unclassified trained terms. In continuous systems, this error rate is obtained through the trained segments that contain recognition error [22].

An efficient and frequently used classifier is the support vector machine (SVM) algorithm, which consists of tracing a hyperplane in order to divide a set of given features in a binary way. Working with such a classifier, it is possible to trace several hyperplanes in order to differentiate n desired classifications, taking as input the extracted cepstral coefficients [1] [6].

Another area of machine learning studies is the ensemble classifiers, which perform a series of classifications using weak classifiers, generating a robust classifier, which is also a binary classifier, and with it it makes the classification of the results, which allows the evaluation of the performance.

1.2 PROBLEM DEFINITION

The security of information or valuable assets is always an issue to be addressed, as no security system works effectively for any case. Even with countless efforts to improve the reliability of systems using digital thumb identification and/or access cards with passwords (however complex they are nowadays), there are still flaws and security breaches that allow undue access of content or material, or otherwise block access to a user who would have the proper permission. An alternative is the use of voice signals to perform access control. Voice recognition systems may have identification failures, but it is a method that is becoming increasingly comfortable in certain applications and, if well developed and implemented, can offer a considerable and sufficient level of security. Unlike a card, password or even digital ID that can be stolen, voice is always inherent to the individual. On the other hand, voice is also not a method that guarantees security in all cases, because voice recordings can be used, making the system accessible. An alternative to increase security efficiency would be to use voice signals (using suitably efficient methods and classifiers) in conjunction with some of the security systems already in use. making the system accessible. An alternative to increase security efficiency would be to use voice signals (using suitably efficient methods and classifiers) in conjunction with some of the security systems already in use. making the system accessible. An alternative to increase security efficiency would be to use voice signals (using suitably efficient methods and classifiers) in conjunction with some of the security systems already in use.

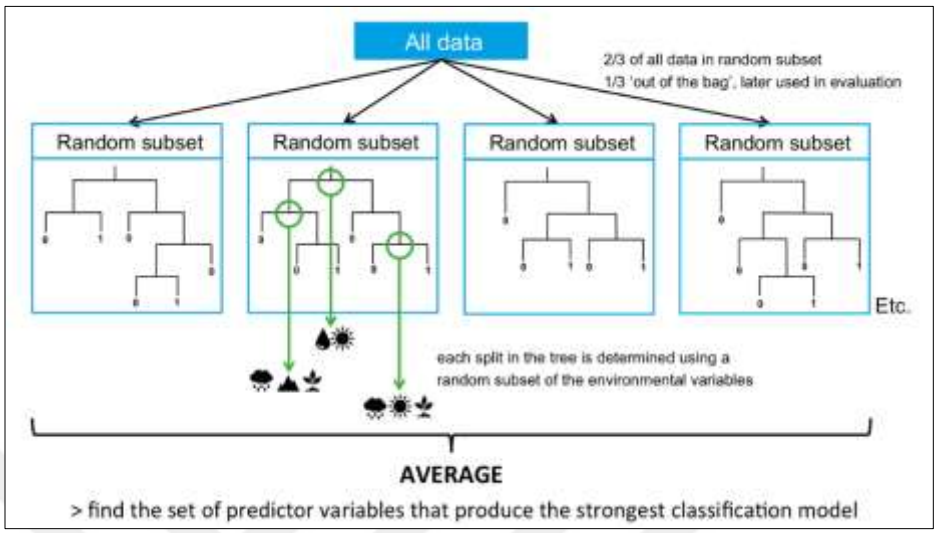

Figure 1.3: Classification process in ML.

1.3 CONTRIBUTION

Analyzing the technological environment that the world is in today, it is remarkable to see that the use of voice for various applications has been increasingly frequent and convenient. Smartphones [15], vehicular applications and others, as it is a system that will offer a greater level of security, accessibility, convenience and comfort for those who use it.

It is then proposed to develop a system that, upon receiving a given sound signal emitted by the user (phoneme or word), will process it in order to analyze characteristics extracted by the classifier for training and, thus, be able to identify the user. or not, if it is not in the voicebank. For the identification of patterns, fundamentals and techniques of machine learning will be applied, responsible for processing the previously acquired signal, and comparing it with the voice signal received at that given moment. This learning is done through adaboost-SVM and ensemble classifiers through supervised training, in which the voices (training) of all those who will be granted access will be registered.

1.4 THESIS OBJECTIVES

We divided our thesis objectives into two parts as following .

1.4.1 GENERAL OBJECTIVE

The present work aims to develop an access control prototype to verify the speech of a certain registered speaker using different classifiers, comparing their performance.

1.4.2 SPECIFIC OBJECTIVES

For the general development of the proposal, some specific objectives were listed:

a. Develop and implement the cepstral coefficients extraction algorithm.
b. Develop and implement classification algorithms (supervised and text-independent).
c. Train systems from a database.
d. Perform the correct identification of the speaker for the developed systems. V. Train systems with real voice signal.
e. Check the identification of the specified speaker in all cases, that is, use metrics (accuracy, precision and others) to validate the performance of each classifier.
f. Compare the performance of the developed classifiers.

1.5 THESIS MOTIVATION

The main limitation in security systems using sound identification is the difficulty of the machine to recognize a word, because the human being naturally inserts variations in his way of speaking that differentiates him from any other, whether by language, accent, rhythm, dialects or simply language vices. It is easy to see that a vocal signal of a word, when processed by a computer, will never be subsequently the same. Thus, techniques need to be developed to create a model that will better define the vocal characteristics of that individual or that language. Furthermore, for a human being, the beep is not the only form of communication in a conversation. Several non-verbal resources are used, such as gestures,20].

Taking into account the existing limitations, it is important to carry out a security application research for the area of speech recognition with the use of machine learning, which has had a development advance mainly in the last decade [20]. And with that, prove that you can use sound signals for access control systems, without errors in order to make the use of speech unfeasible.

1.6 THESIS STRCUTURE

The present dissertation develops as it will be described. In the second chapter describes the literature review of the past works. In the third chapter the theoretical foundation about the basic concepts to be worked on is presented. It initially describes how speech production occurs and its characteristics from a physiognomic point of view. The extracted characteristics that will be used in this work are deepened in this section. The theoretical and mathematical foundations of the classifiers to be used are described, as well as the relationship and application of the characteristics extracted from these systems. At the end of the section, the general concept of speaker recognition as a system is discussed. The types of speaker recognition and their application needs are presented. This last topic basically deals with how the classifier interacts with the extracted features (labeling, validations. The fourth describes the entire methodology used to implement the system. In the first part of this section, which covers the experimental methodology, the system is described at a high level from the hardware point of view, that is, which basic components are needed, their functions and how they will compose the project as a whole. Also presented in this section is a description of the implementation of the classifier algorithms and the extraction of features (language, software, libraries and others). The second part deals with the experimental methodology, which describes all the tests performed (as well as how they were done) that will allow the system's reproducibility and falsifiability. Tests, classifiers and worked parameters are also presented. For the tests, which database was used and its main characteristics (number of people, age, gender and acquisition conditions) are detailed. The last part of this section, the analysis methodology, describes how the tests performed will compose a valid result, that is, with which metrics (statistical results) we can reach a valid classification conclusion.

In the fifth section of this work, the collected and validated results are presented in fact, as described by the analysis methodology. Possible improvements and limitations of the implemented algorithm are also discussed. This section presents, graphically and in tables, comparisons between the chosen classifiers with regard to the metrics presented in the analysis methodology. Finally, the last section concludes the dissertation with respect to the collected results and the entire system itself. This includes the schedule of tasks and studies carried out.

2. LITERATURE REVIEW

2.1 OVERVIEW

Speech recognition and speaker recognition are the two primary areas of automatic speech recognition technology. Recognition of spoken language is the more important of the two. The researchers followed an approach that was rational in the sense that they first constructed systems that were capable of interpreting speech before attempting to determine the identity of the speaker in the actual world. This was done before attempting to determine who the speaker was. Because of this, researchers began researching voice recognition well over a decade before to the very first study that was ever completed on speaker recognition. This was a direct result of this. Davis was working at Bell Laboratories in 1952 when he and some of his coworkers [2] came up with a method that could determine the numbers that a person was saying aloud. This method was able to identify the numbers. By leveraging formant frequencies that were measured in vowels, this approach was able to distinguish and differentiate between spoken digits. The initial attempt at automatic voice recognition was a big step forward in the field, but at the time, the system could only distinguish digits. This did not stop the field from making significant progress, though. It was unable to discern between individual words, whole phrases, and even numbers when utilizing this method, as demonstrated by the findings of the study. It was extremely sluggish and had very restricted capabilities because of the lack of computing power that was available at the time.

2.2 RELATED WORKS

Traditional methods of speaker recognition are included in both analyses of speaker identification strategies, which were written by Saquib et al. (2010) and by Singh and Singh, respectively (2017). A considerable number of researchers who were working on this topic employed conventional approaches to speaker detection and sound comparison for most of the work that they did. The exploration of deep learning algorithms for speaker detection and voice comparison has received an insufficient amount of effort up to this point. In order to obtain a higher degree of comprehension of the current state of the art, it is necessary to carry out a

survey comparing traditional and deep learning-based strategies to speaker recognition (identification and verification), as well as voice comparison.

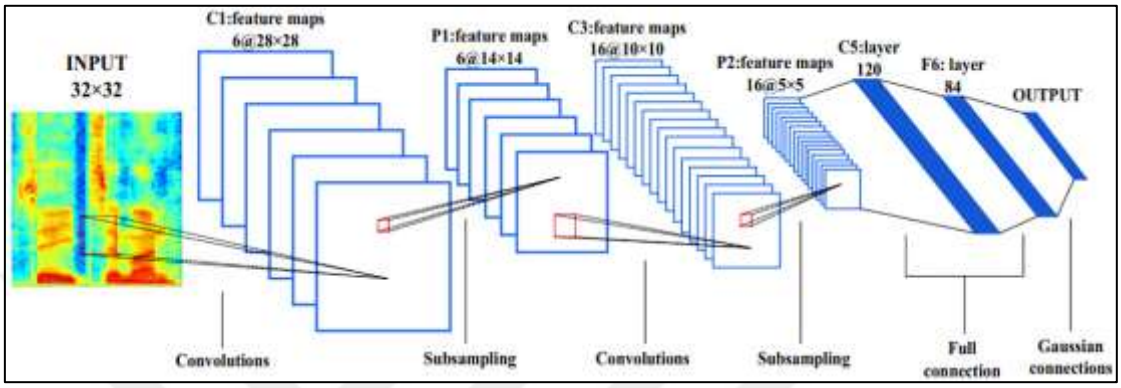

Figure 2.1: Spectrum Extraction By CNN.

Researchers Cardoso et al. [1] working on the voice comparison system developed a novel method to extract Vq (Voice Quality) aspects from speech and display them graphically utilizing MFCC features. This was accomplished by creating a new technique. They chose 97 participants at random from the DyViS corpus to take part in their investigation for the purposes of their study. [1] 32 speakers are utilized for training, 33 speakers are utilized for testing, and 32 speakers are utilized as references out of a total of 97 speakers. During the process of producing a DyViS corpus, speech is transcribed in four distinct ways: high-quality (HQ), telephone (TEL), mobile high-quality (MOBHQ), and low-quality (MOBLQ). High-quality transcriptions are referred to with the abbreviation "HQ" (mobile low-quality recording). The author made the decision to utilize the MFCC as a feature in order to make the process of feature extraction easier to complete. The incorporation of Vq capabilities into the MFCC was done with the intention of enhancing the system's overall performance, which was the primary motivation behind the decision. The following four unique approaches are utilized by us in order to arrive at an appropriate value for Vq: The fundamental frequency (F0), the cepstral peak prominence (CPP), the harmonic-to-noise ratio (HNR), and the fundamental frequency (F0) are some examples of these (harmonic to acoustic ratio). There is evidence that humans possess all four of these properties in their vocal features. These attributes are listed in the order they appear in the evidence. These four MFCC features will be extracted from spoken words by a voice

recognition system for the purpose of further investigation. This is the case for a number of reasons, the most important of which is that the contribution of Vq to system performance increased as the quality of transmission decreased from HQ to TEL, MOBHQ, and MOBLQ. There are a number of other reasons why this is the case, but these are the most important ones. The Equal Error Rate (EER) for the standard Chinese triphthong /iau/ was 2.85 percent when MFCC was used on its own, but it reduced to 0.91 percent when the Vq. trajectory technique was paired with it. In order to determine the characteristics of Chinese tokens, an acoustic-phonetic approach, which is applied in conjunction with machine learning, is used. The researchers used a method known as logistic regression fusion in order to combine the test results that were received from both the acoustic-phonetic and automated systems. In a previous work [38] that was quite similar to this one, Morrison et al. created and implemented a forensic voice comparison system for the standard Chinese monophthongs /i/, /e/, and /a/.

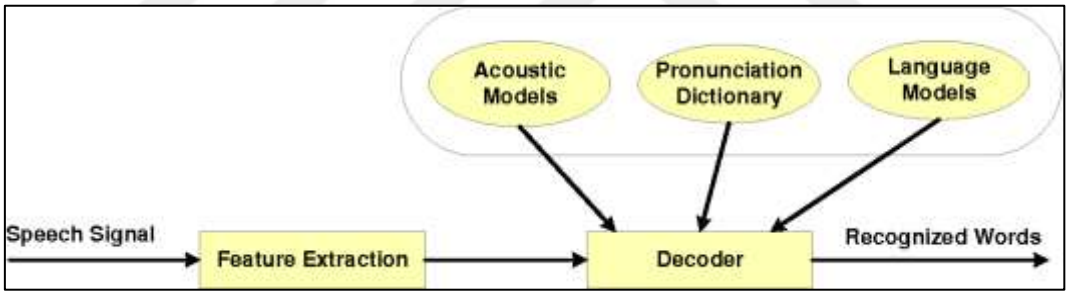

Figure 2.2: Method Proposed By [39]

In the research paper that they had published [39], Variani and his colleagues suggested applying a Deep Neural Network (DNN) model to the task of speaker verification. The acoustic characteristics of each video frames can serve as the foundation for the DNN's training to teach it how to discern between the many different kinds of people who can be heard speaking. After it has been identified which d-vector features represent the typical characteristics of these speakers, those features are then used to validate the validity of new speakers. This process continues until all speakers have been checked. Lukic et al. [12] conducted research on the TIMIT dataset in which they explored a new method for enhancing the They were able to publish their findings and share them with the scientific community because the authors used a

Convolution Neural Network (CNN) on spectrograms to learn speaker-specific characteristics from a wide variety of audio sources. This allowed the authors to share their findings with the scientific community. Many layers of convolutional neural networks can be applied to smaller local input regions of the network by utilizing CNNs, which are also known as convolutional neural networks (e.g. a 3X3-area, which is then repeated throughout the entire input space). Following the convolution layer is the max pooling layer, which lowers the resolution of the activations produced by the convolution layer by deleting the full filter activation from a 2X2 window. This occurs after the convolution layer. A version of the activations created by the convolution layer with a reduced resolution is produced as a result of this. Following that step, the convolution layer of the image will be removed from the file. After the last max-pooling layer, the outputs of all of the systems are combined to produce a speaker classification system. This system is generated by combining the outputs of the systems.

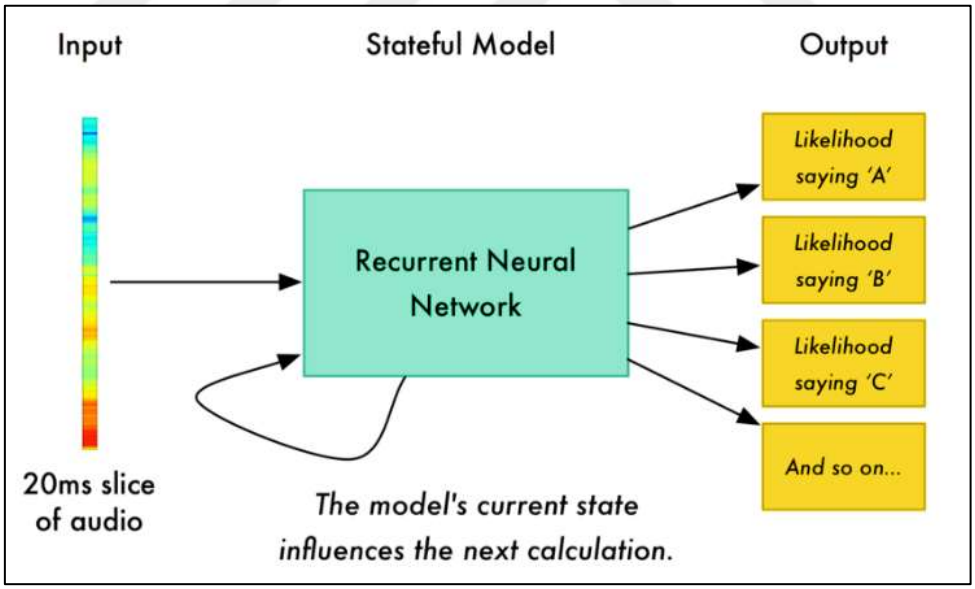

Figure 2.3: RNN Based Model Proposed By [40].

In the context of speech recognition systems that make use of microphones and data that is contaminated by noise, Plchot et al. presented a DNN-based auto-encoder that was able to differentiate between a range of different speakers (DAE). One of the fundamental responsibilities of the auto-primary encoder is to improve the sound quality of the spoken signal.

This is one of the tasks that it is responsible for (i.e., to de-noise and de-reverberate). However, multi-condition training is especially helpful in the situation of dealing with distortions that are created by additive noise. It is feasible to apply strategies for audio enhancement in order to correct for the distortions that are caused by reverberation; these techniques include: The 3D-CNN architecture, which was proposed by Torfi et al. [42], is a unique approach to text-independent speaker verification. This architecture was developed by the researchers. Convolutional neural networks are incorporated into the design of this architecture. They proposed an adaptive learning feature that could be accomplished by first directly generating a speaker model with 3D-CNN, and then subsequently integrating it into the system. This would allow the system to learn new information in a manner that is more tailored to the individual user. Each speaker gives the network the same total number of contributions as did the speaker before them in the line of speakers.

Chen et al. (2019) have developed a real-time class-based language model (CLM) for enterprise-to-enterprise (E2E) systems by employing a class-based language model (Kneser and Ney, 1993) and a token passing decoder with efficient token recombination. This was accomplished by using a class-based language model. This paradigm was conceived with enterprise-to-enterprise, or E2E, communications in mind (Hall et al., 2015). The community concluded that Wang et al(2020recommendation)'s that a Scout Network and a Recognition Network might be merged into a single system in order to achieve low-latency speech recognition was correct. Word boundaries can be identified with the assistance of the Scout Network, which is flexible enough to deal with any kind of neural network. This makes it possible for word boundaries to be determined. The recognition network will use the information from the frame before the projected boundary in addition to the information from the frame before that in order to create an educated prediction about the following subword. Because it is difficult to capture the lengthy background information of words in language models for big vocabulary continuous speech recognition (LVCSR), it can be challenging to create language models for large vocabulary continuous voice recognition (LVCSR). Because of this, developing language models for LVCSR is a challenging endeavor.

In order to successfully build a model for a subword unit, it is essential to perform estimations of the model parameters using a training set consisting of continuous speech utterances. This collection of continuous speech utterances needs to include all pertinent subword units occuring "sufficiently" frequently in order to be considered complete. It is possible to directly attribute, at least in part, the effectiveness of the identification system as a whole to the manner in which training is carried out. When it comes to training, determining the appropriate size of the training set is one of the most important decisions that must be made. Because of constraints on the amount of processing power available, it is impractical to construct training sets of an indefinite size. As a result, we will need to make do with sets of a size that is easier to work with.

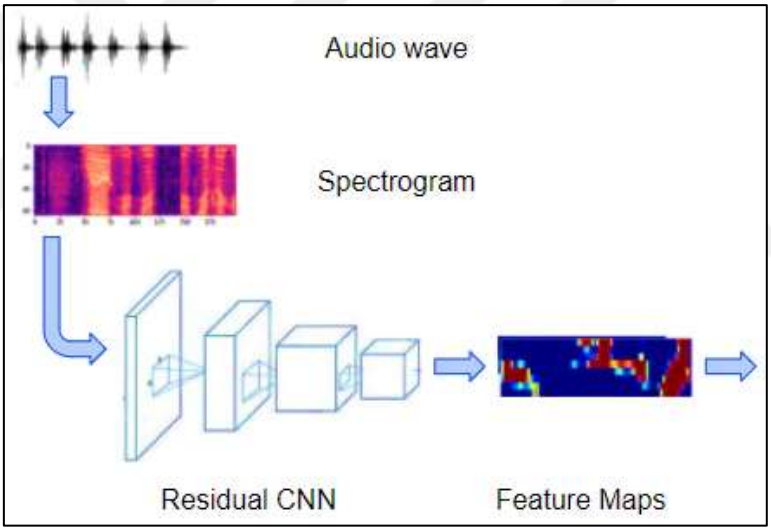

Figure 2.4: Sub-world Based Recognition System.

Sub word units, in contrast to other parts of speech, may not necessarily occur as frequently as other parts of speech do. [Case in point:] [Case in point:] When constructing a phrase, there is a trade-off between using more sub word units, which offers better coverage for each individual unit but less context resolution, and using fewer sub word units, which provides better coverage for the context but less individual unit coverage (which provides better unit coverage but less context resolution). Increasing the number of sub word units leads to improved coverage of individual units (where we get poor coverage of the infrequently occurring units, but good resolution of linguistic context). When initial sub word unit models are used rather than a

massive training set, it is possible for the models to be fine-tuned to the activity, speaker, and/or environment over the course of time. This is made possible because initial sub word unit models require less data to train than massive training sets (with new training material, which may be drawn from real-world test utterances). These methods make it possible to generate application-specific models from a set of models that is more generic in nature. They are especially helpful when coping with new people, activities, and environments (which are speaker, environment, task, and context independent). Not only are speech patterns highly variable with regard to the spectral properties of the sound, but they are also quite variable with relation to the passage of time. This is because speech is a highly dynamic process. In the field of speech modeling, it can be difficult to find modeling approaches for the speech signal that are theoretically valid and computationally manageable at the same time. This can be a challenge for those working in this area. Within the field of speech recognition, the modeling technique that has seen the most extensive application and successful implementation is known as hidden Markov models (HMMs). Rabiner (1989) offers not only an introduction of the HMM approach and its many applications but also a step-by-step guidance to the subject matter that is being discussed. As a different kind of framework and paradigm, ANN techniques [Bourlard and Wellekens, 1992; Bourlard and Morgan, 1994; Robinson, 1994] have proven useful to modeling and computation [Bourlard and Wellekens, 1992; Bourlard and Morgan, 1994; Robinson, 1994]. [Bourlard and Wellekens, 1992; Bourlard and Morgan, 1994; Robinson, 1994] Hidden Markov models and its expansions are used in the great majority of today's automatic speech recognition systems to model speech units. These models are utilized in virtually every single one of the processes that are carried out by these systems. The HMM framework will be broken down into its component parts and discussed in further depth in the subsequent section of this paper. When compiling a dictionary that can actually be used, it is vital to take into account not only the number of different ways a word can be pronounced, but also its baseform. This is because a word's pronunciation might vary depending on the context. This is because the way a word is spoken may shift depending on the context in which it is used (or standard). In terms of how a word is pronounced, the baseform pronunciation functions as a kind of guide, while the number of alternative pronunciations reveals how various regional accents and talker populations say the

word. The baseform pronunciation is the most common pronunciation. When a word is employed in continuous conversation, the way it is spoken may be very different from the way it is pronounced in its root form. This is especially true with regard to word borders. This is particularly true with the beginnings and finishes of words. It has been shown that making use of various pronunciations or networks of pronunciations makes it possible to manage lexical variants in a way that is both simpler and more straightforward (43).

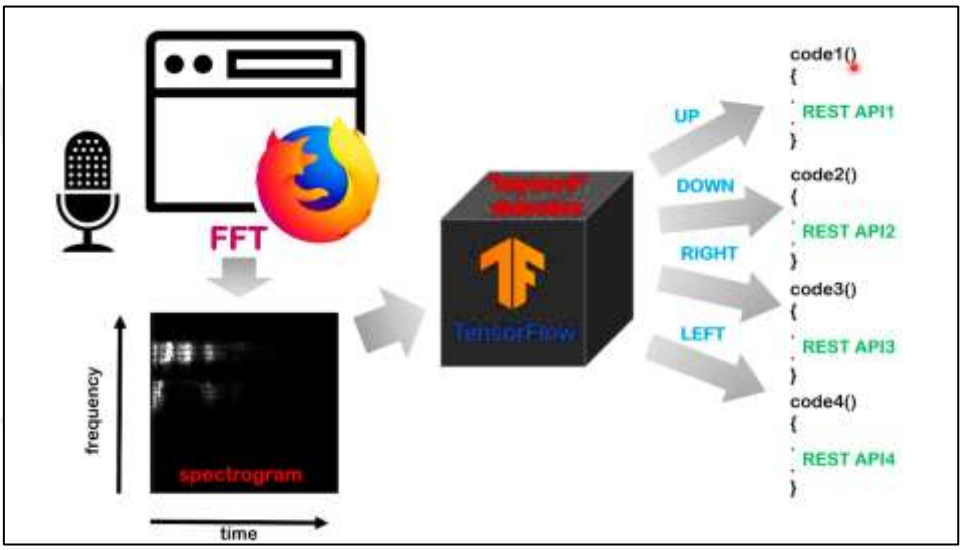

Figure 2.5: FFT based model proposed by [43].

It is required to add language-specific phonological rules in order to model lexical variability. Additionally, it is necessary to build acoustic-linguistic mapping rules that pertain to the selection and modeling of subworld units. Finally, it is necessary to develop word models. In addition to this, it is essential to incorporate morphological principles that are particular to the language. A fascinating new avenue of research is the application of probabilistic word modeling, which directly characterizes lexical variability and is able to be used to detect patterns in text. This new avenue of research has the potential to significantly advance the field.

3. MATERIALS AND METHODS

3.1 VOICE SIGNALS

3.1.1 The Human Vocal System

Speech production basically has two main components. The first of these is the source of excitement. The source of excitation corresponds to the flow of air that occurs leaving the lungs, conducted by the trachea, when we want to produce speech. The second component is the "vocal tract". This corresponds to the entire system that functions as a filter (pharynx, larynx, epiglottis, esophagus, oral cavity, nasal and others, shown in Figure3.1), changing the spectrum of the acoustic wave (frequency domain) from the excitation process. This entire set (and each of them itself) carries information inherent and unique to each individual that will be extracted for speaker recognition [4]. A good example of this is the possibility of recognizing a person only by his breath, because what is produced in the process of arousal carries characteristics of a particular human being, similar to what happens in speech (which proves that such characteristics are not are only a result of the vocal tract, but in conjunction with the source of excitation) [17].

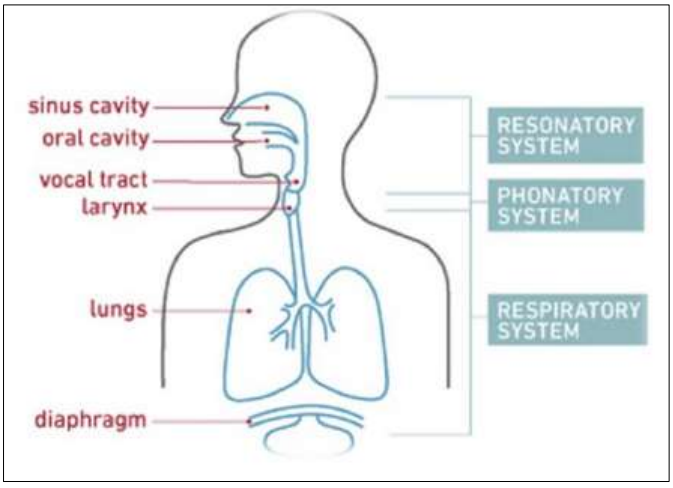

Figure 3-1: Human Vocal System.

Therefore, for every voice signal, considering the randomness and inflections in the speech production process, it is possible to extract and process its characteristics and, applying it to a classifier system, we can differentiate such a signal from any other. Such characteristics are the cepstral coefficients (LPCC).

The modern society of today has an ever-increasing demand for identity and safety measures, which has resulted in the development of many different safety and identification measures. This demand is driving innovation in the field. Biometrics is a term that refers to many techniques that can be used to determine an individual's identity based on the unique characteristics that they have. It is possible to identify a person based on his or her physiological or behavioral characteristics, such as a person's stride (which belongs to the category of physiological characteristics) or facial features (which belong to the category of behavioral characteristics) (behavioral) Natural language processing is broken down into several subfields, one of which is called speech recognition. Speech recognition is concerned with the automatic translation and recognition of spoken language. This field of research incorporates concepts and theories from a wide range of academic subfields. It is not possible to sufficiently convey how important voice recognition is to the digital transformation process as a whole. In recent years, many applications of the Internet of Things and machine learning have been created and implemented in a range of different business sectors. These sectors include the education sector, the industrial sector, and the healthcare sector, amongst others. The process of recognizing words that have been pronounced is considered to be one of the most challenging problems to solve in the field of computer science. Voice recognition is a common term used to describe this process. The most efficient method of voice recognition has not yet been identified, despite the considerable quantity of research that has been conducted on the topic. This is due to the fact that natural languages have a huge diversity of characteristics, and that each language has its own unique set of characteristics that distinguish it from other languages.

3.1.2 Cepstral Coefficients

Cepstrum was first described in 1963 as a technique for recognizing time-resonant frequencies in voice signals [5]. It consists of the representation of the frequency power amplitude in a

Short-period spectrum, considering a time-invariant signal [17] [1]. Spectrum representation is a more refined mathematical technique that allows the extraction of short-period speech signal formant frequencies without significant loss of information useful for classification. The math consists of applying the transform (frequency domain representation) and then applying the modulus and logarithm operations to the signal, ensuring that it can be correctly separated within the useful frequency range that you want to work with in a non-linear way. . That is, the human voice reaches a certain frequency range, but from this range, a restricted portion represents the signal from which we can extract a phoneme or even a characteristic of a certain individual.

After such processing, the inverse transform is performed to obtain the coefficients. It is worth mentioning that for an efficient signal processing, an adequate sampling of the original signal must be done, respecting the Nyquist sampling theorem. The block diagram of such a process is represented in Figure3.2.

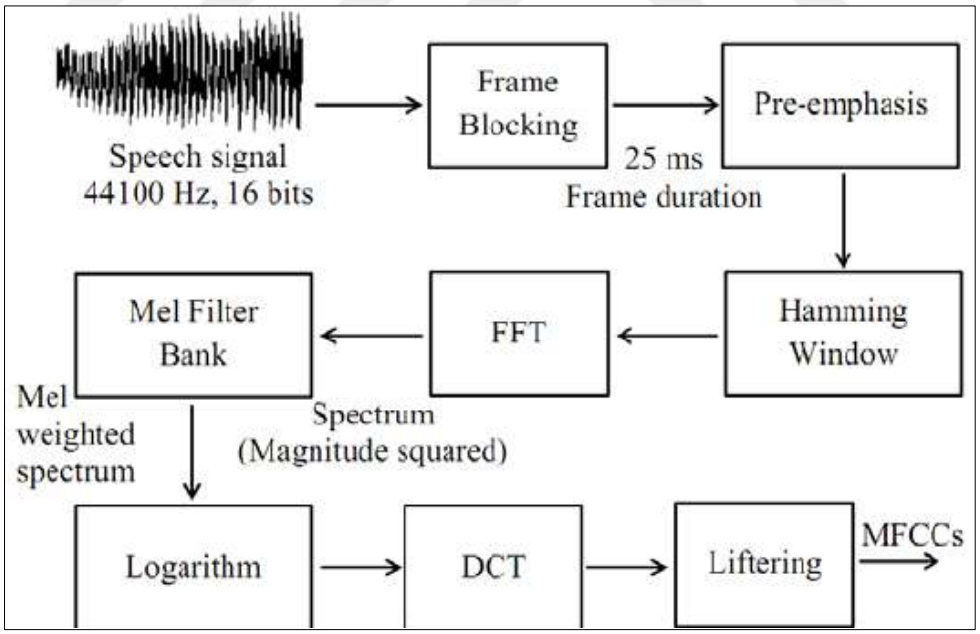

Figure 3.2: Block Diagram of Cepstral Coefficients Extraction.

The diagram above shows how the cepstral coefficients are extracted in stages. Early in the speech process, excitation occurs from highly stochastic factors. This excitation is modified by what we call the vocal tract. The vocal tract consists of the physiological characteristics of each

One (rib cage, lungs, diaphragm, trachea physiognomy, mouth and others), and then produces an output signal that we will be concerned with treating. Such a signal then corresponds to a time domain convolution between the excitation and the vocal tract system.

As shown in the diagram, we apply the Fourier transform, which implies a multiplication in the frequency domain. What we want to do, in fact, is to apply the logarithm in the equation as a mathematical technique so that we can extract the coefficients of smaller values without significant loss of information (manipulating the multiplication to a sum). Before that, then, we must take the module of the equation, considering that we want to deal with the exclusive contingencies of the logarithm (negative numbers, for example).

Finally, we duly apply the logarithm function, and select the lowest-valued coefficients, which are sufficient for our further classification. The inverse Fourier transform is then applied, to return to the time domain. A suitable experimentally defined number of (smaller) samples is extracted.

Another way to calculate cepstral coefficients is through Linear Prediction Coding or LPC coefficients, these in turn are calculated through a method using autocorrelation coefficients [16], to extract cepstral coefficients from LPC coefficients, the following recursion is used

$$c_m = \begin{cases} \ln \sigma^2 & m = 0 \\ a_m + \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) c_k a_{m-k} & 1 \leq m \leq p \\ \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) c_k a_{m-k} & m > p \end{cases} \tag{3.1}$$

with*am*being the LPC coefficients,*cm*the ceptral coefficients and$\sigma^{\text{two}}$is the gain term associated with the LPC model, remembering that (3.1) shows that a given number*P*of LPC coefficients, can generate a greater number*Q*of cepstral coefficients, usually something around

$$Q \simeq \left(\frac{3}{2}\right) p \tag{3.2}$$

the ceptral coefficients*cm*generated by the above recursion are called Linear Predictive Cepstral Coefficients (LPCC).

3.2 CLASSIFIERS

For the classification of sound signals, it is essential that pattern recognition be approached and applied, through the construction of good models of a given data set, in the case of this work, sound signals.

This dataset, in machine learning, consists of feature vectors (or training), where each is an object description using a series of features, for example, the classification vectors for beeps are numerical values that describe the object, sound signal, through a series of characteristics (intensity, vocal tract, etc.). The number of features in a dataset is called its dimension or dimensionality, the attribute features are the instance training vectors, and the dataset are the samples.

From the samples, a model can then be built. This process of generating models from data is called learning or training, which is accompanied by a given algorithm. There are several models for machine learning, the most common of which are the ones already mentioned: supervised and unsupervised [27].

For supervised training, the main objective is to obtain values with invisible instances. This model obtained is called a predictor, because we predict the values as being "square" or "cross". This is called a label, and the predictor must be able to categorize an instance within a given label, if it is categorical. The learning process is given the name of classification and the algorithm is called a classifier and if the label is numerical, the process is a regression, and the algorithm is then an adaptive linear regression. In unsupervised training, there is no use of information from labels. A clustering algorithm is made, capable of discovering patterns for the data obtained, classifying them in this way.

In order to define whether a given classification (or regression) model for supervised training works properly, its generalization must be evaluated, since generalizing well indicates that the learning process is evaluating well the data received from the invisible instances, generating a low training or prediction error [27].

3.2.1 Decision Trees

A decision tree consists of a classifier structure structure that works in a divide-and-conquer architecture, where each intermediate node corresponds to a feature test, so for each new training data, branches will be made according to the values of the characteristics, and the final node of each branch has an associated label. To carry out the validation, a series of tests is performed on the data so that the validation data is associated with some of the final nodes of the tree, thus generating a label for that data [27], a simple example of a decision tree can be seen in Figure3.3.

Decision trees are generally recursive processes, because there is a whole procedure to make which branch to follow and because, that is, the key to the good functioning of the algorithm is how to select the tests of characteristics.

It is a common case that decision trees that have a good efficiency for training samples do not have a bad generalization of information, while

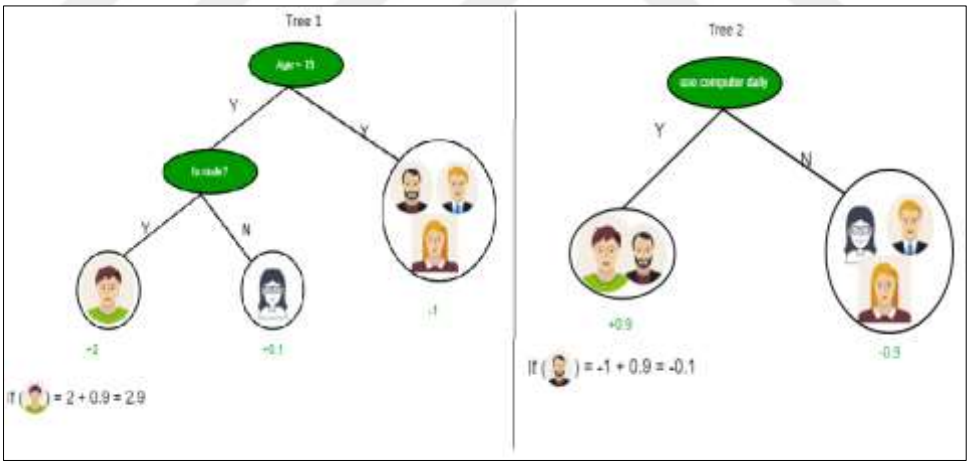

Figure 3.3: Example of decision tree.

some trees that don't work so well for training end up validating better, generating the case of overfitting, this can happen because of noise present in the training samples that ends up generating an inefficient feature test [27].

Due to the low generalization, decision trees are often used as a base classifier for ensemble classifiers, which, through the iteration process, manages to generate and combine several trees, creating a process without overfit and with high generalization capacity.

3.2.2 Support Vector Machine

Support vector machines (SVM) are binary classifiers that use machine learning to separate a set of training vectors. The idea was officially developed in the mid-1990s [6], however, previous studies about the creation of machine learning using neural networks to create linear planes [26] and creating optimal separation hyperplanes [25] were made, which enabled the development of the algorithm proposed by Vapnik.

The SVMs then implement the following idea: to map the input vectors in a multidimensional space Z through a nonlinear mapping chosen in principle (kernel). In this space, the linear decision surface is constructed in order to guarantee the SVM's ability to classify correctly.

This decision is made by creating the optimal separation hyperplanes, which consist of the linear decision function with the greatest distance in space between the 2-class vectors. To build hyperplanes, few samples of the trained data are needed, the so-called support vectors, which are located at the ends of the set of values in space. The minimum distance between them is then calculated, and in the middle of this distance, the optimal hyperplane is generated [6]. It is demonstrated in the experiment that, if the classification is done correctly through an optimal separation hyperplane, the probability of occurrence of a classification error is limited by the ratio between the number of support vectors and the number of training vectors,

$$E[P(Erro)] = \frac{E[Vs]}{Vt}, \tag{3.3}$$

on what*AND*[P(Error)]is the error probability estimate,*AND*[P(V s)]number of support vectors used to classify and*V t*the number of training vectors.

In the figure3.4we can see an example of a simple SVM with only 2 classes. In this example, one can clearly see the position of the support vectors, the distribution of trained data for a given feature and the position of the hyperplane.

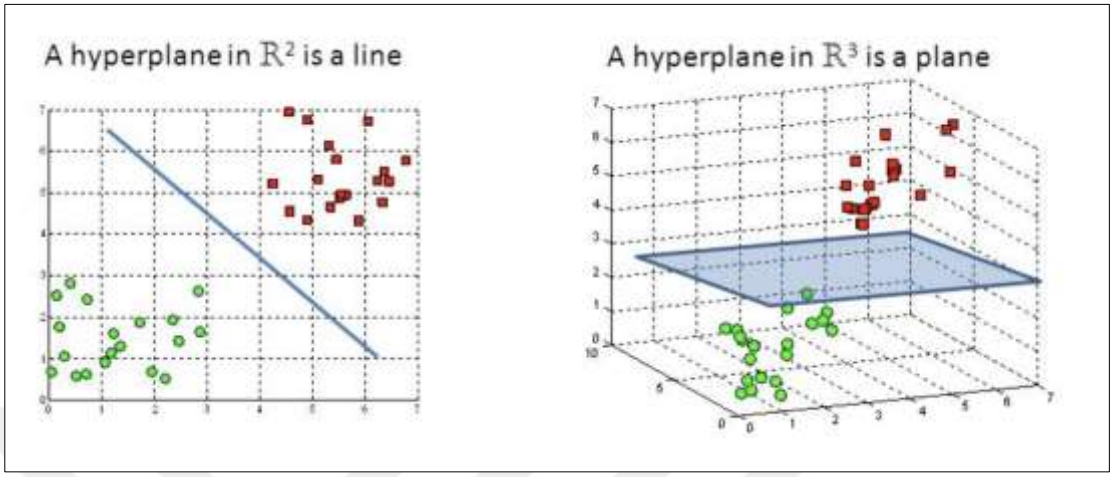

Figure 3.4: Example of SVM with 2 features and linear hyperplane in 2 Dimensions

To calculate an optimal linear hyperplane of an SVM considering a training with 2 labels shown in the figure3.4, if there is a vector*w*~and a scalar*B*, considering the values trained as*xi*and labels like*yi*, which satisfy the following inequalities,

$$w\sim\cdot\sim xi + b \geq 1, yi = 1 \tag{3.4}$$

$$w\sim\cdot\sim xi + b < -1, yi = -1 \tag{3.5}$$

manipulating (3.4) and (3.5) is obtained,

$$and(w\sim\cdot\sim xi + b) > \tag{3.6}$$

1or

$$and(w\sim\cdot\sim xi + b) - 1 = 0 \tag{3.7}$$

for vectors at the ends of the projections.

The hyperplane shown in Fig.3.4is represented by

$$(w0\cdot x + b0) = 0 \tag{3.8}$$

which separates the trained values with the maximum distance between the values of each label. Calculating the distance between the projections of the training vectors we have

$$d = (S_M - S_m)\frac{w}{|w|} = \frac{2}{|w|} \tag{3.9}$$

where d is the distance vector, $\frac{w}{\|w\|}$ corresponds to the vector~*x*normalized.*SM*and*sm* correspond to the support vectors at the ends of each region.

To maximize this distance, Lagrange multipliers are used.*αi*, then we get

$$W = \frac{1}{2}\|w\|^2 - \sum_i \alpha(y_i(\vec{w}_i \cdot \vec{x}_i + b) - 1)) \tag{3.10}$$

with W corresponding to the maximized distance.

To calculate the value of the vector*w*~, one takes the derivative of the maximized distance with respect to itself (*w*~, getting

$$\vec{w} = \sum_i \alpha \vec{x_i}\ y_i \tag{3.11}$$

Using this value, we can say that*w*~is nothing more than the sum of some linear vectors. Now computing the derivative of L with respect to*B*, it is obtained

$$\sum_i \alpha y_i = 0 \tag{3.12}$$

Substituting now the values found in the equations (3.11) and (3.12) above in (3.10)

$$W = \frac{1}{2}\left(\sum_i \alpha_i \vec{x}_i\ y_i\right)\left(\sum_j \alpha_j y_j \vec{x}_j\right) - \left(\sum_i \alpha y_i \vec{x}_i\right)\left(\sum_j \alpha_j y_j \vec{x}_j\right) - \sum_i \alpha y_i + \sum_i \alpha_i \tag{2.13}$$

rearranging the equation, we have

$$W = \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j \tag{3.14}$$

which is the equation that, in fact, maximizes the distance between the two vectors.

Now putting this representation of*w*in (3.11) in the equation (3.5),

$$\sum_i \alpha_i y_i \vec{x_i} \cdot \vec{u} \geq 0 \tag{3.15}$$

it is then observed that the decision rule depends only on the inner product between~*xi*and~*u*.

However, this maximization of the distances between the hyperplanes of the ends often cannot be done without a classification error being generated, this error corresponds to the training samples that were between the hyperplane of one end and the optimal hyperplane, in which

case you want -to obtain a separation with as few errors as possible, to deal with this approach we introduce the function

$$\phi(\xi) = \sum_{i=1}^{l} \xi_i^{\sigma} \tag{3.16}$$

that, for a low value ofσ >0, subject to the conditions

$$and(\mathrm{w}\sim \cdot \sim \mathrm{xi} + \mathrm{b}) \geq 1 - \xi \mathrm{i} \tag{3.17}$$

$$\xi i \geq 0 \tag{3.18}$$

describes the number of training errors, whereξi= 0for vectors outside the regions between hyperplanes, minimizing it describes a space with the least number of training errors.

If these data are excluded from the training set, it is then possible to generate a set that will not generate training errors, thus being able to generate an ideal separation hyperplane. For this, we minimize the equation

$$\frac{1}{2}\left\|w^2\right\| + F(u)C\sum_{i=1}^{l} \xi_i^{\sigma} \tag{3.19}$$

with$F$(u)being a convex monotonic function and C a constant.

For a large enough C,σsmall enoughσ= 1, the vector$w$0and$B$0that minimize (3.19), considering (3.17) and (3.18), then the optimal hyperplane is determined that minimizes the errors of the training dataset and separates the remaining data by maximizing the distance as described earlier in (3.18), considering now the training errors, the equation becomes,

$$W = \sum_i \alpha_i - \frac{1}{2}\left(\sum_i \sum_j \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j + \frac{\alpha_{max}^2}{C} \right) ! \tag{3.20}$$

being$\alpha max$the value of the largest Lagrangian multiplier.

For cases in which a linear hyperplane cannot classify properly, the vectors must be transformed to another space of N dimensions, in which it is possible to perform the linear separation without errors. To move both vectors~$xi$and~$u$the function is usedphi, which transforms a function from an initial dimension m to an M, so that in this dimension it is

possible to separate the training samples linearly, different from the initial dimension, thus a (3.15) stay

$$\sum_i \alpha i y i \varphi(\sim xi) \cdot \varphi(\sim u) \geq 0 \tag{3.21}$$

This point-to-point product between the two transformed vectors is the kernel of the system, and is defined as

$$K(\sim x, \sim u) = \varphi(\sim x) \cdot \varphi(\sim u) \tag{3.22}$$

ie

$$\sum_i a i y i K(\sim x, \sim u) - b \geq 0 \tag{3.23}$$

There are numerous types of kernels for spaces with more N dimensions. In the figure3.5there is an example that uses an rbf kernel, which has the format,

$$K(u, v) = \exp\left(-\frac{|u - v|}{\sigma}\right) \tag{3.24}$$

SVMs model class decision limits using hyperplanes. Like, As shown in Figure 9, the hyperplane separator divides the Cartesian plane into two classes. SVM uses margin, the shortest distance between decision limits and any other sample, to handle multiple limit solutions. The decision limit must maximize margin. Support vectors are the hyperplane's maximum margin. Figure 3.13 shows vectors X a and X b. After training, discard the vectors that do not affect the decision limit decision and find the one with the fewest generalization errors. Speaker verification uses two classes: target speaker training vectors and imposter speaker vectors. SVM finds a hyperplane separator that maximizes class separation using these vectors labeled by class. Figure 3.9 shows the separation of two classes that are linearly separable in the original input plane (X 1 and X 2), but this is rare. In these cases, kernel functions compute the internal products of two vectors in the kernel feature space, which is larger than the input space. Hyperplanes separate classes better on larger spaces. Somewhat intuitively, a linear hyperplane

in the kernel characteristic space corresponds in decision limit in the original input space, such as the MFCC space.

3.2.3 Ensemble Classifiers

Ensemble methods train multiple classifiers, unlike other machine learning methods that build just one classifier containing all the training data, ensemble methods build a series of classifiers and combine them creating a robust classifier, Figure3.6shows the default architecture of an ensemble classifier [27].

An ensemble contains a number of base classifiers, which are generated from the training data by a base machine learning algorithm, which can be a neural network, decision tree, KNN, or other pre-specified algorithm. In ensemble classifiers, the most normal thing is to use only one base algorithm for classification, guaranteeing homogeneity of the classifiers, also leading to the use of homogeneous ensembles [27].

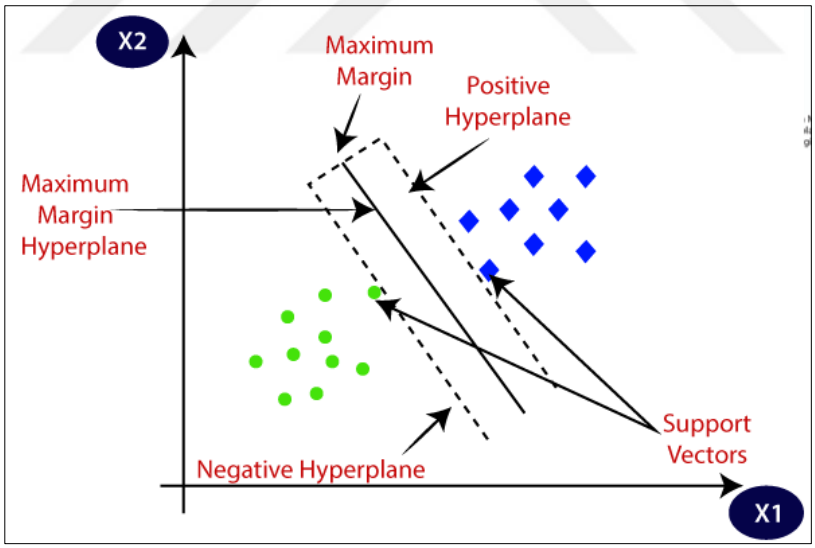

Figure 3.5: SVM nonlinearly separable in 2 dimensions, trained using rbf kernel.

The various approaches that can now be utilized, text-independent voice recognition is the one that offers the most degree of adaptability. In addition to this, it is the one and only practicable modality that can be used to all speech recognition systems that are now available on the market. A message ID should be returned by the system, and it shouldn't matter who the speaker Is, the ID should be completely determined by the contents of the tract. Even when the identity

of the speaker is known, this should still be the case. It is important to merge the closed set and the check into a single object in order to accurately identify an open set. This is done by combining the two sets. Following the application of the closed set method in order to repeatedly identify the same speaker, the verification model that corresponds to that ID is then employed in order to carry out the voice test on the verification model while applying the closed set methodology. After the verification process has been carried out effectively and to completion, the user will be provided with a unique identification number. When a person's identification is established by listening to their speech, there are no presumptions made about what they are saying, and the system runs in the same manner as if they had spoken a PIN. This is because listening to a person's voice is a biometric method. In order to successfully con someone, you need to use a technique that does not rely on language, and any recording of high enough quality is capable of accomplishing this goal

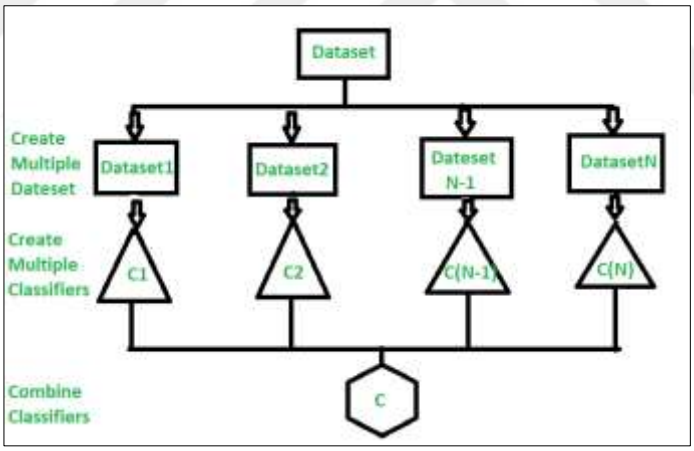

Figure 3.6: Schematic of an ensemble classifier

The generalization ability of an ensemble is generally much greater than that of the base classifiers separately. Therefore, one of the most common applications for ensemble classifiers is to boost weak classifiers, in order to obtain results similar to those of a robust classifier, that is, in this area the base classifiers are also known as weak classifiers [27]. In the area of ensemble methods there are three types of approaches that are commonly used [27].

a. Classifier combination: Studied in the area of pattern recognition, he works with robust classifiers and tries to organize his results and create combination rules to generate combinations of robust classifiers.

b. Ensembles of weak classifiers: Studied in the area of machine learning, it works using a weak classifier and through robust algorithms, it boosts the performance from weak to robust. This study area gave rise to Boosting and Bagging ensemble classifiers such as: Adaboost , RobustBoost , Bag.

c. Expert mix: Studied in the field of deep learning, ensemble tries to learn a mix of parametric models together and uses matching rules to get an average solution.

3.2.4 Boosting

The term boosting refers to the idea of transforming a weak classifier into a robust one using an iterative and sequential process, that is, each classifier (except the first) uses data from previous learning. Intuitively, the weak classifier is better for random learning, while the robust classifier has a performance closer to the ideal. [24] proved that it is possible for a weak classifier to become robust, thus developing the first boosting algorithm seen in Figure3.7.

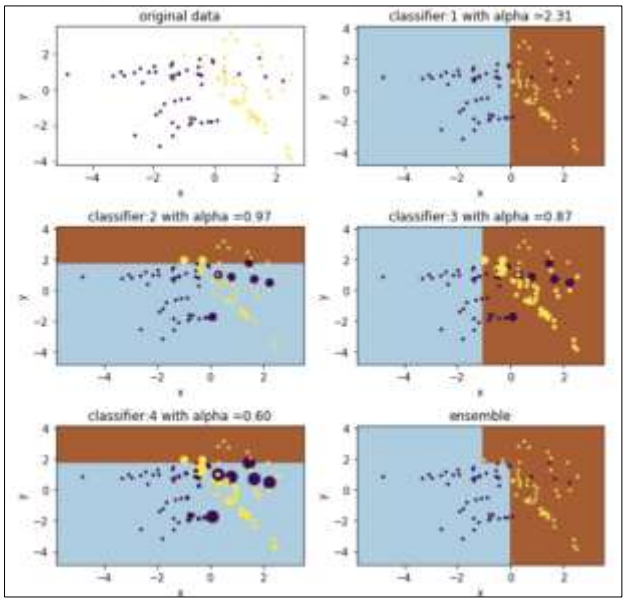

Figure 3.7: Generic boosting algorithm

The general idea of a boosting algorithm is quite simple, suppose a given distribution (D1) with three equally distributed spaces for example, and a classifier (h1) is trained with a 50% chance of error (quite undesirable) when classifying each of these 3 distributions, 2 were classified

correctly (X1 and X2) and 1 failed (X3), to correct this 1/3 of failure, one can, for example, make a derivation of this space, so that the error seen in the first classifier becomes more evident, training and classifying again, there was a hit in X1 and X3 but error in X2, this second classifier (h2) is also a weak classifier and has a high capacity to fail, and if we combined it with h1, we still wouldn't have the robust classifier desired, as there would be some flaws in the classifications of X3 and X2. If one more derivation is made,27]. The generic boosting algorithm that generates strong learning is shown in Figure3.7, the algorithm is not real because there are functions mentioned without specifications such as:Adjust_Distribution and Combine_output, some applications of this algorithm are Adaboost and Robustboost.

A weak classifier commonly used to perform ensemble classifications is the decision tree, which consists of an algorithm that.

3.2.5 Adaboost

The Adaboost(Adaptive Boosting) algorithm developed in 1995 by Yoah Freund and Robert E. Schapire [13], is the most common boosting method currently and follows the line of reasoning discussed above, since training is performed using a weak classifier (line 3 of the generic boosting algorithm described in3.7and the error estimate of the Dt distribution classifier (line 4), made by evaluating which samples were correctly classified from the distribution and if the error is above 50%, it means that the classifier cannot be better than a classifier random, that is, completely invalidates the classifier by discarding it.

The Adjust_Distribution function described earlier in Adaboost is composed by calculating alpha coefficients, these will be calculated according to the classifier error through the equation,

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right) \tag{3.25}$$

and will later be used to weight each of the classifiers generated in the loop. In addition, the weights for each sample must be updated, these will be defined according to the classification and the alpha value, these weight values will be higher for samples that were incorrectly classified and lower for the correct classifications, the initial value default is 1, and varies with

each training, the formula used to carry out the weighting of the weights for the samples are described in .

$$D_{t+1} = \frac{D_t \exp(-\alpha_t f(x) h_t(x))}{Z_t} \tag{3.26}$$

With *Zt*corres ponding to a normalization factor.

After several N iterations of the algorithm, creating a series of classifiers*ht*, the procedure represented by the Combine_output function in the previous section is performed, which in Adaboost corresponds to the sum of alphas by multiplying the values of a sample x of all classifiers and evaluating the correct class through the response sign, in the case of a classifier with two classes, the formula that describes the final classification for each sample is described in (3.27), the general Adaboost algorithm is described in Figure3.8

$$H(x) = sign\left(\sum_{t}^{T} \alpha_t h_t(x)\right) \quad ! \tag{3.27}$$

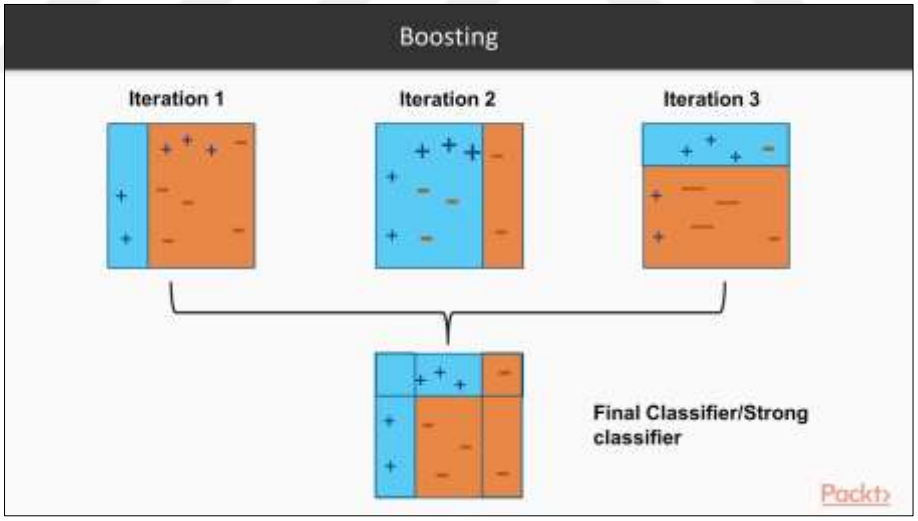

Figure 3.8 : Adaboost Algorithm.

3.2.6 Robustboost

Since the invention of the Adaboost algorithm, which was a great advance in the area of machine learning, for being an algorithm that managed to avoid overfit, that is, a training that Only classifies correctly within the training samples , without generalization. However, the algorithm had a problem with the insertion of random noise in the labels of the training samples,

which ended up completely deteriorating the results. Friedman proposed an algorithm to circumvent the problem called gentle Adaboost or Logitboost, which had more noise tolerance, this was due to the equal weight given to all samples that were poorly trained by the weak learning algorithms, unlike Adaboost which increases the weight , which ends up generating a greater tolerance for noise.

The algorithm called Robustboost was then created by Yoav Freund in 2009 [12], is a variation of the Brownboost algorithm [11], which is based on the Boost by Majority (BBM) algorithm [13], this algorithm is based on the idea that the number of iterations until the creation of the strong classifier is defined by the algorithm from a user-defined error input. Most of the more common boosting algorithms like Adaboost, put less weight on samples that were already quite right by previous classifiers, in the case of BBM, it puts less weight on samples that have many misclassifications, putting them in the measurements of training errors. A disadvantage of this algorithm is that it is not adaptive, that is, it does not have the idea of putting weights on each of the trained h classifiers, to generate the output, Brownboost is precisely the adaptive BBM algorithm runningçtimes, beingçpre-computed parameter via error.

The Robustboost algorithm is very similar to the Brownboost algorithm, except that, instead of minimizing the classifier error, it seeks to minimize the number of training samples for which the normalized margins mare less than a given valueθ >0.

Its loss function has the form

$$\phi(m,t) = 1 - erf\left(\frac{m-\mu(t)}{\sigma(t)}\right) \tag{3.28}$$

and with*erf*being the error function

$$erf(a) = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{a} \mathrm{e}^{-x^2}\,\mathrm{d}x \tag{3.29}$$

μ(t)andσ(t)are defined by the equations

$$\sigma 2(t) = (\sigma f2(t) + 1)e2(1-t) - 1 \tag{3.30}$$

(3.31)

$$\mu(t) = (\theta - 2\rho)e(1-t) + 2\rho\text{deriving}$$

(3.28) we obtain the weight function for the samples

$$w(m,t) = \exp\left(-\frac{(m-\mu(t))^2}{2\sigma(t)^2}\right) \tag{3.32}$$

the parameter θ is the objective, and increasing this value reduces the performance difference between the data used in training and validation. The algorithm finishes running when *t* reaches the value 1, if the error value is too low, the algorithm does not finish running, the idea is to find the minimum value for the error, which is previously defined, which guarantees a reasonable amount of iterations. With$c\ c\ \sigma_f = 0.1$to avoid numerical instability with*t*close to 1, calculate the value ofρ, allowing the calculation shown in (3.32). After the iterations are done, the final hypothesis is made in a similar way to Adaboost.

3.2.7 Bagging

According to the way in which the base classifiers are generated, there are two types of paradigms in ensemble methods, in one of them these classifiers are generated sequentially (as already explained in the case of boosting algorithms), and in another method they are generated in parallel, which this is the case for bagging algorithms. The main motivation in the use of bagging algorithms is to explore the independence of the base classifiers, unlike boosting classifiers that exploit dependence (training with weights on the samples), and through that to minimize the error by combining these classifiers [27].

Through Hoeffding's inequality, it can be seen that the generalization error reduces exponentially according to the number T of independent classifiers used [27]. However, it is practically impossible to generate independent base classifiers as they are generated from the same training data, one way to reduce dependence is to randomly choose the training data. Another advantage of using parallel ensemble methods is that the training speed can be quite high if you are using computers with a multi-core processor.

The Bootstrap Aggregating (Bagging) algorithm was developed in 1996 by Breiman [3] and uses the idea of independent base classifiers discussed earlier. In order to generate different base classifiers, bootstrap sampling is applied [8], to get the data subsets to use to train these

classifiers. This database works as follows, suppose a training database with a number M of training examples, using the technique of sampling with repetition, a sample with M training examples will be generated from the previous base, however some training examples will be repeated while others will be excluded, doing this process T times, T samples of M training examples are generated, each one being used to train a base classifier. robust

$$H(x) = sign\left(\sum_{i=1}^{T} h_i(x)\right) \qquad ! \tag{3.33}$$

In order to group the outputs of the base classifiers, bagging adopts voting for classification and calculating the average for regression. In the case of classification, the label that has the highest number of votes is given as the correct classification. bagging can work perfectly with binary or multiclass classifiers. The Bagging algorithm is shown in Fig.3.9.

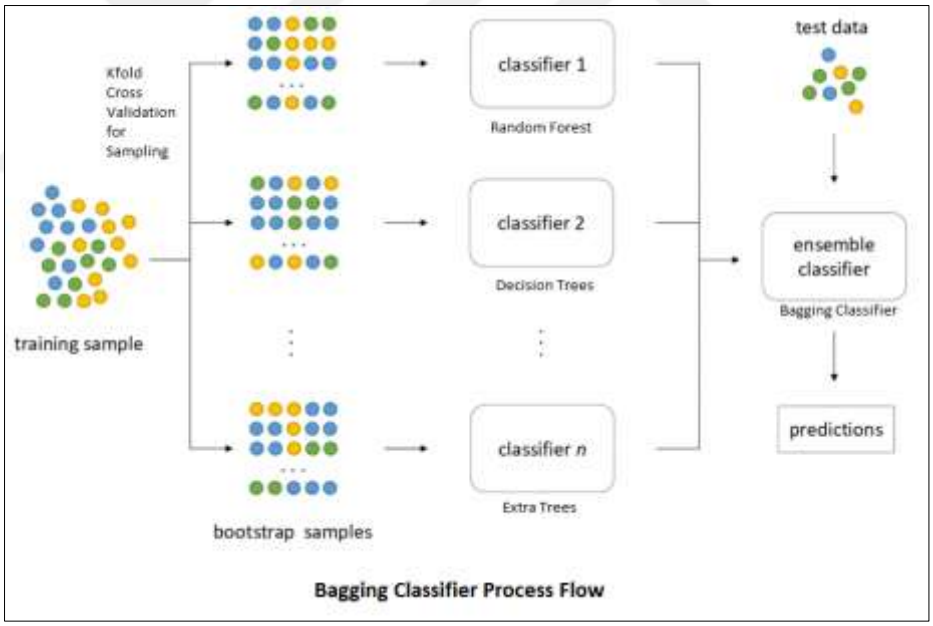

Figure 3.9: Bagging Algorithm.

3.3 AUTOMATIC SPEAKER RECOGNITION

Automatic speaker recognition systems can be developed in order to meet different requirements of different applications. Basically, such systems can be divided into text-dependent or text-independent. A text-dependent system is one that requires a certain previously known phoneme (or sentence) to be provided as input for classification. On the other

hand, a text-independent system does not require the provision of a particular phoneme or sentence, as it only extracts the features and performs the classification, so the latter needs a considerably larger volume of data for training for good effectiveness. [4].

A basic characteristic in the training of an automatic speaker recognition system is the supervision or non-supervision of the same. In supervised systems, the system is given labels (groups) to which the entries may belong, so that the system will classify which of these labels (identifications) the given entry belongs to [27]. In unsupervised systems, the inputs are provided and mapped, so that the algorithm itself must perform the labeling of the provided characteristics in order to relate the inputs with some specific identification (not previously informed). A simple representation of these types of training can be seen in Figure3.10.

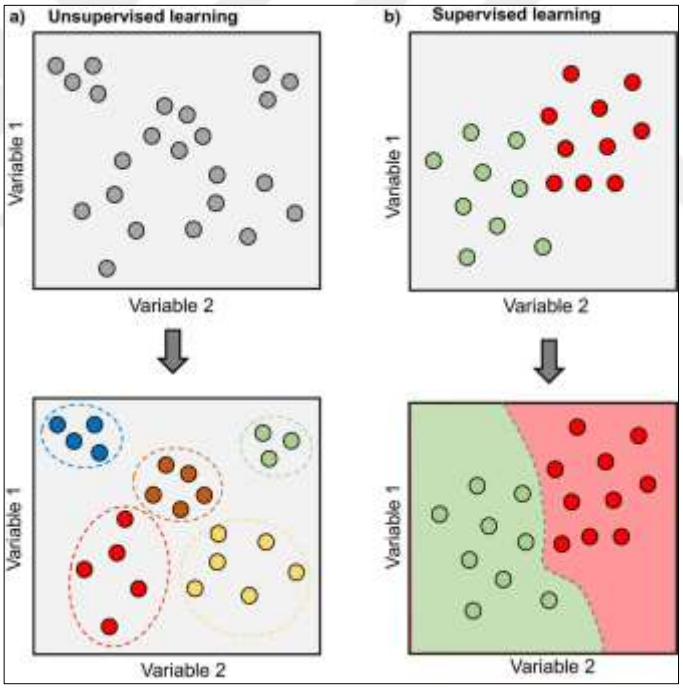

Figure 3.10 : Basic Graphic Representation of Supervised and Unsupervised Learning.

Another bifurcation that we find in the projects of such systems is the speaker identification and verification. In speaker identification, a test entry is provided and the system identifies which group speaker (already trained) that entry belongs to. The speaker identification can still be a system with or without rejection. In systems with rejection, the level of similarity of the characteristic entered with the label must reach a certain threshold, or the classification will be

rejected. On the other hand, in talker verification, an input is provided and the system must accept or reject the talker's identity [18]. The difference between systems with and without rejection is the number of decisions to be made. While talker identification needs to make a number of decisions corresponding to the number of labels trained in the system, talker verification takes only one, whether or not the input belongs to the label [19].

In this work, we will deal with text-independent, supervised systems that perform speaker verification. Disregarding the supervision feature, we can classify the speaker's automatic recognition systems according to Figure3.11

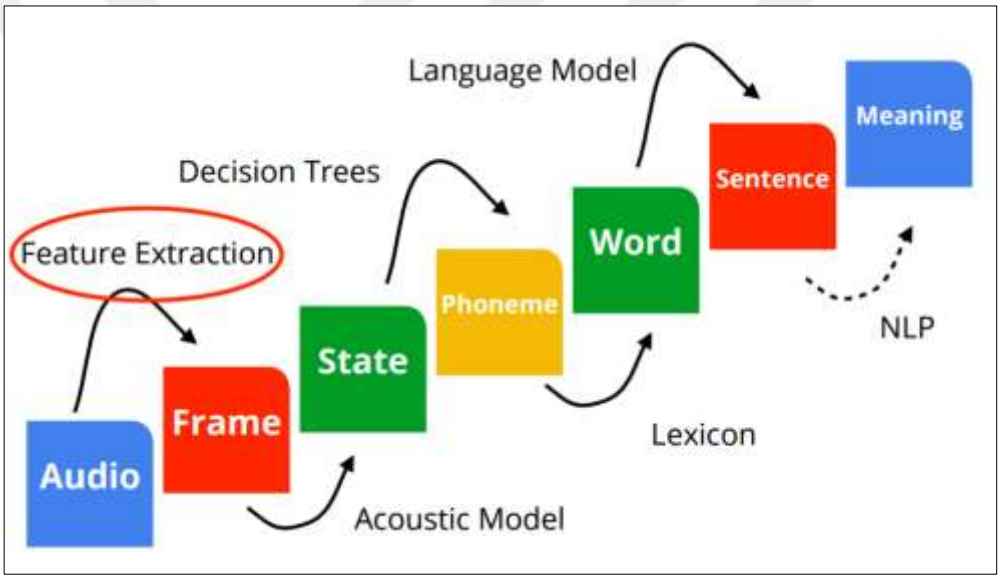

Figure 3.11: Types of Automatic Speaker Recognition.

With the individuality in the characteristics that are extracted from the voice signals resulting from the vocal tract (different for each human being), we were able to distinguish individuals previously labeled in the system. The quality and efficiency of the system that will perform the speaker recognition is directly linked with the correct extraction of the features to be classified, and with the choice of the classifier itself [15].

One method that was used in the early days of speech and speaker recognition is the Hidden Markov Model. The implementation of such a technique for classification, however, is not the most efficient nowadays. Using the cepstral coefficients in [21], this system had a 75% success rate. Another traditional method (even today) of classification is the Gaussian Mixture (GMM),

which estimates the adjacent probability distribution predominant in the extracted features. Such classifier, using the cepstral coefficients as characteristics, presented a rate of 70.8% of accuracy in [15].

Evaluating now the results present in the literature on the classifiers to be analyzed in the present work, we verify in the implementation of an SVM in [21] an accuracy rate of 84%, still using the cepstral coefficients as chosen characteristics.

4. PROPOSED METHOD

4.1 OVERVIEW

To build a voice user interface using Bangla language, our main task was to develop a speech recognition system. Though, many works have been done in speech recognition However, we chose to use Convolutional Neural Network (CNN) as a basis for our speech recognition algorithm. To use CNN, we needed a lot of sample data of Bangla speech. But unfortunately, there was no available dataset in the internet which we can use. So, we planned to collect the dataset ourselves. We collected the speech data from most of the students of our batch, then processed the data and used them for training our neural network. After that, we built an user interface which can interact with the user through audio commands.

in this project, our aim is converting an analog signal to digital signal and process it in the computer environment using MATLAB. firstly, we sampled the signal with a specific sampling frequency then we determined the speech's character by taking its Fourier transform and store the information in an array and lastly compare the newly pronounced digit with the array which includes the feature vectors of the digits. in this project the main problem is that to characterize the speech and store them, then characterize the testing speech as well and finally compare them to understand whether they are the same or not, and understand the digit said by the user

First, WE sampled the wav format voice, which received by MATLAB with method wavread, then WE divided the sampled signal into frames which are each 25ms for a 1.5 second signal, and secondly most crucially WE detected the start point of the speech the first instance where magnitude of the signal is greater than the threshold level, that provides us to eliminate the noisy parts. With this elimination process our new signals length is 10000 samples because WE take 10000 signals from the start point of the speech. With 16000 Hz sample rate MATLAB listened 1.5 seconds of voice it corresponds 24000 samples in discrete domain and then WE only took the 10000 parts as mentioned above, then the new signals length is 0.625 seconds in time domain, 0.625 / 0.025 = 25 frames. Then WE multiplied each frame with hamming window to minimize the maximum side lobe. Then WE take the FFT of the frame, then WE calculate power

estimate of the frame and then WE create Mel filter bank which consists of 26 filters and multiply power estimate with this filter bank to divide the signal into frequency bands. WE take the log and DCT of each filters results respectively that give us the Mel Frequency Cepstral Coefficients for one frame WE do this operation for 25 frames thus our feature vector size is 650 for one digit that pronounced.

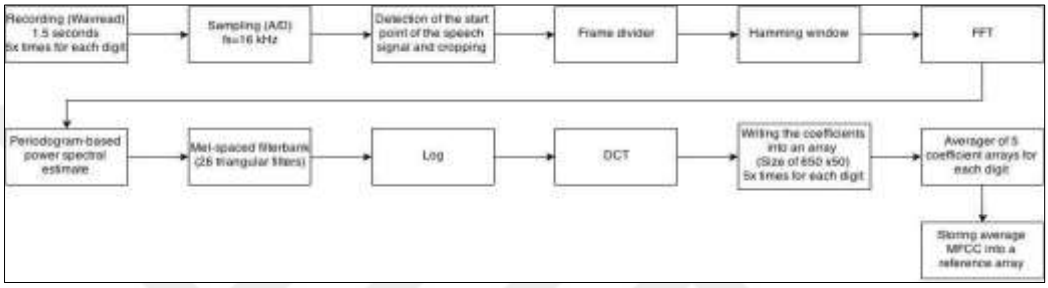

Figure 4-1 : Stock price model with a non-linear rise weekly

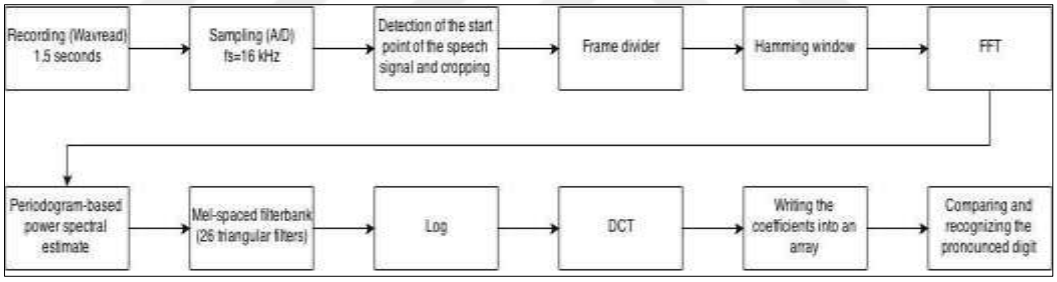

Figure 4.2: The Block Diagram of Testing Process

4.2 DATA COLLECTION

For data collection, we used simple microphone. Each person (Trump, Blair, Obama) is asked to say their roll number digit by digit and then say the digits from zero to one repeatedly. Each student's sample was 40 seconds long.

Additionally, to make our neural network more robust, we needed to train it with noise signal too, so that it can distinguish between actual speech signal and noise signal. So, we recorded some random noise at random places and used them as samples to train the network. Five recorded wav sound samples were stored in the database, and we wish to recognize them using a correlation test.wav in MATLAB.

STEP1: We create a test file command

STEP2: The code represents the sample one.wav, do this repeatedly for as many samples you want to be present. In this case, we have five samples. Therefore, we repeated this code five times with every place that say 1 replaced with 2, 3, 4 and 5 respectively.

STEP3: We note that the test has x value, it is this value that is used to compare the y value of the sample. Hence, Z1=xcorr(x,y);

STEP4: We create a conditional statement where m6=300; and if m=max the machine will sound the allowed

4.2.1 Data Processing

After we have collected the data, we had to make the collected data suitable for training the neural network. Firstly, we had to crop the different digit from the audio files that we have collected. But these were of different lengths. So, we padded zeros to make them of same length. It was possible to do because adding zeros to a signal does not change its frequency spectrum.

A. Recording (Wavered) Block

Recording the speech using MATLAB's wavered command and WE record for 1.5 second, 16000 sampling frequency, then, WE store the speech in an array size of 24000.

B. Sampling (A/D) Block

The recorded speech signal is sampled using 16000 Hz sampling frequency. WE recorded for 1.5 seconds so WE have 24000 samples. Moreover, the human vocal range is 300 Hz – 3400 Hz and it can include cosines with frequency of 3400 Hz, so to avoid aliasing WE choose 16000 Hz which is much greater than 2*3400=7200 Hz (Nyquist rate).

Detection of the start point of the speech signal and cropping

The aim of this block is determining the start point of the speech where the first instance's magnitude of the signal is greater than the threshold level. This approach provides us to eliminate the noisy parts. and the threshold can be modified according to the environment's noise. Using this elimination process, our new signal's length is 10000 samples, WE take 10000

samples from the start point of the speech to catch the part of the signal where the digit exactly spoken. Briefly, this process increases the recognition rate.

C. Frame Divider Block

Taking 25 ms frames of the speech signal. The cropped signals length is 0.625 seconds in time domain, 0.625 / 0.025 = 25 frames. There are 10000/25 = 400 samples in each frame. The reason WE select 25ms as frame durations in time domain is to make the signal stationary in each frame to increase the precision of the feature vector.

D. Hamming Window

Using the hamming command of the MATLAB, WE created a fixed point hamming window of size 400 which equals to sample number for each frame to ease the implementation to the FPGA. Then WE multiply each frame with hamming window to minimize the maximum side lobe. Hamming window is better than the other types of filters in terms of suppression of the maximum side lobe.

E. FFT Block

We take 400 point fixed point FFT of the frame using fft command in MATLABbecause our frames are 400 samples long. Taking FFT determine the speech characteristic of the specific frame. As a result of the FFT operation WE have frequency characteristic of the frame.

Periodogram-based power spectral estimate

We take the periodogram-based power spectral estimate for the speech frame by taking the absolute value of the FFT values then take the square of the absolute value of the FFT then divide it by the length of specific frame. The aim of this process is to understand the power estimate of the specific frame, moreover; it provides information about power estimates of different frequency components for a specific frame.

This formula gives the power estimate of the frame.

Mel-spaced filter bank (26 triangular filters) Block

As a result of our research about Mel- filter banks, it is common that 26 filters are used in the implementation of filter bank. Therefore, WE decided to create 26 overlapping triangular filters increasing in size, using the formula above.

4.2.2 Calculating Spectrograms

Spectrograms are a very efficient way of representing speech signals. Specially, in neural network applications, speech signal is nearly always represented using spectrograms.

Spectrograms are a two dimensional array more like an image. To compute spectrograms, the signal is first segmented into chunks of very small time lengths like 20ms. Then, for each of these segments, DFT is done. Then, an image is formed in which the horizontal axis represents time, the vertical axis represents frequency, and the color of a point represents the power of the corresponding frequency component of the signal segment at the corresponding time.

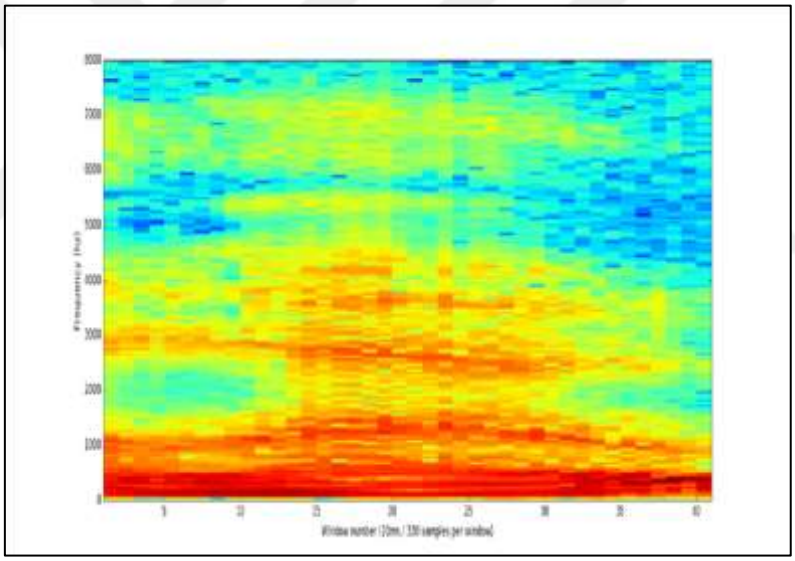

Figure 4.3: Example of a Spectrogram

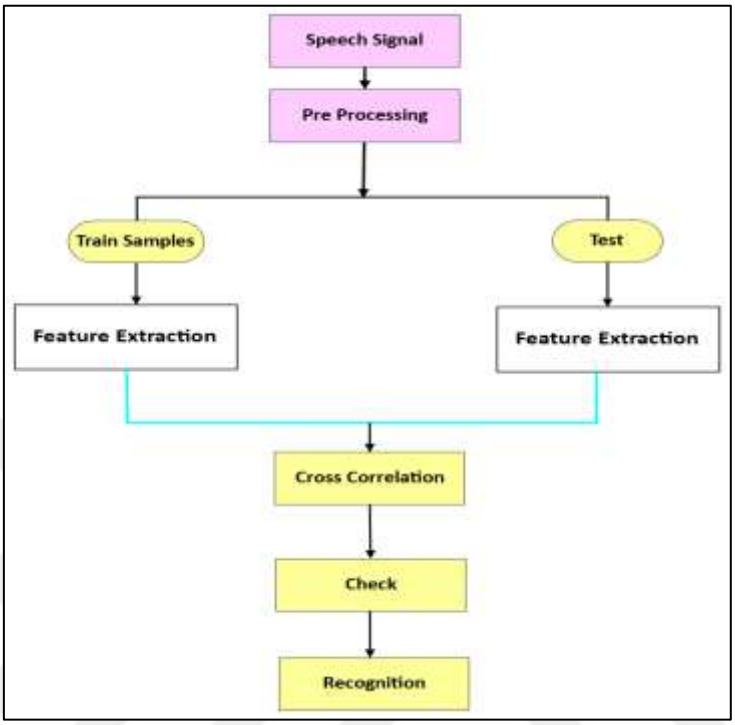

Figure 4.4 : Diagram of The Training Process

We have performed this operation in our project using built-in functions of MATLAB.

Log Operation Block

We take the log of the values that came out from the summation of values from the filter bank using log command in MATLAB. This process scales the summation values to reasonable values to store. For example, log101000000=6. WE can say that this process imitates the human ear's scaling function.

a. DCT Operation Block

Using the dct command in MATLAB we take the DCT's of the log values, WE use DCT because DCT operates using only real numbers, so it eases our process by preventing any imaginary numbers in the feature vector.

b. Writing the coefficients into an array

After getting the coefficients from DCT, for each filter there is one coefficient as a result of log and DCT processes and 26 coefficients for each frame, lastly there are 25 frames so there is 25x 26=650 coefficients for each speech, this forms the feature vector. For training code this process is a little complicated, as WE record 5 times for each digit to improve the feature vector, which is for one digit WE have [650x5] matrix which includes 5 feature vectors that will be averaged in next block.

c. Average Block

We sum the first elements of the five feature vectors then WE divide the sum with five and WE store the average value in another array's first column which corresponds that for digit '1's feature vector. WE do this operation for 650 times to form the improved feature vector for digit '0' to '9'

4.3 TRAINING THE NEURAL NETWORK

MATLAB Neural Network Toolbox trained the neural network. Layered network architecture was invented. Convolutional, batch normalization, and max pooling layers down sampled feature maps "spatially" (in time and frequency). A final max pooling layer pools the input feature map globally over time. This enforces (approximate) time-translation variance in the input spectrograms, which seems sensible if we expect the network to classify speech regardless of time. The final fully connected layer has fewer parameters due to global pooling. Add a little bit of dropout to the inputs of the layers with the most parameters to prevent the network from memorizing training data features. These convolutional layers have the most filters. Each final convolutional layer has 36864 weights (plus biases). The final completely connected layer weighs 3840.

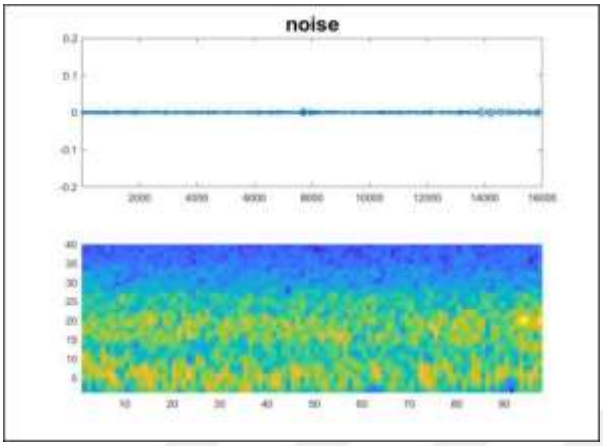

Figure 4.4: Diagram of the Training Process.

Next, we applied weighted cross entropy classification loss. We utilized inversely proportionate class weights to give each class equal weight in the loss. When training with the Adam optimizer, class weight normalization should not affect training. With a 128-mini-batch size and 5e-4 learning rate, we employed the Adam optimizer. After 25 epochs, we halved the learning rate.

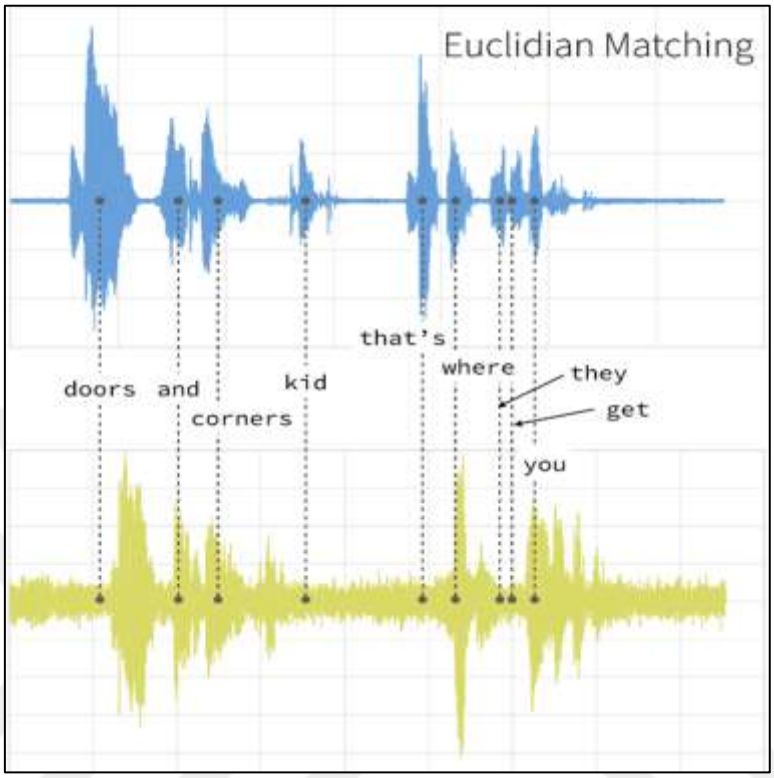

Figure 4.5: Spectrum of the Training in Progress

After the training process has been done, then we saved the trained network in a .mat file so that we can use it anytime we want without the need to train the network each time.

4.4 TESTING THE NETWORK

Our freshly trained command detection network was tested on microphone-streaming audio. We said "zero," "one," etc. It usually understood speech commands.

But sometimes it got it wrong. It continuously read data from microphone and sends a data of length 1 second to the network. The network classifies the signal and then the most probable command is set through a thresholding operation. It is shown as a title of the figure. In the figure, we can also see the signal and its spectrogram in the figure which is constantly changing as new speech signal is being input to it.

4.5 PSO FEATURE SELECTION

The developed solution, due to its ability to adapt to the user's voice using the Dynamic Time Warping method, has a fairly wide range of applications. After recording the software on the smartphone, the final product can correspond to the work of a virtual assistant or, after creating an appropriate interface, can execute commands chosen by the user to work with the hardware (eg: turn on/off the lighting in a certain place). The solution may also apply to alternative input for disabled users. Also, as previously mentioned, this can be used in multimedia (computer games, art installations, etc.).

4.6 SOLUTION DEVELOPMENT TOOLS

The MATLAB programming environment was used for the development of the solution due to its versatility; and the Dynamic Time Warping recognition method due to its fast operation without using a lot of computer resources. Requirements

This program can be implemented in various forms, but currently a web page is used because it makes the software widely and quickly available. The following requirements are currently planned for this software:

A. Internet access (if the software is not downloaded).
B. Access to microphone (audio input).
C. Access to the speaker (audio output).

4.7 METHODOLOGY USED

This chapter discusses the main audio equalization algorithm, Dynamic Time Warping.

A. Dynamic Time Warping Algorithm

Dynamic Time Warping (DTW) is an algorithm for estimating the correlation between two data sets with unequal time parameters. This algorithm can be applied to data analysis in many areas - to compare financial statistics between months with an unequal number of days (eg April - 30 days, May - 31 days); in pedometers - to count the number of steps taken when the user's walking speed is not stable, etc.[3]

In this work, DTW will be applied to speech recognition - we will look for correlation between recordings of different durations. The user will submit three voice recordings, the content of which must contain the same spell. These entries, once confirmed to be sufficiently similar, will be considered benchmarks - spells understood by the system. As the program continues, the benchmark with the highest correlation will decide what word the user submitted to the system. The power condition is that a minimum correlation value must be reached. If this value is not reached, the system will let the user know that the word (spell) they are saying is not valid. After saying the correct spell, the corresponding colored light assigned to the specific spell will light up.

The DTW method was chosen because it solves the problem of uneven speech speed. Searching for correlation consistently makes speech recognition unreliable. DTW is an effective solution because it allows for time manipulation (Figure 4 – Sequential Search, Figure 5 – DTW)

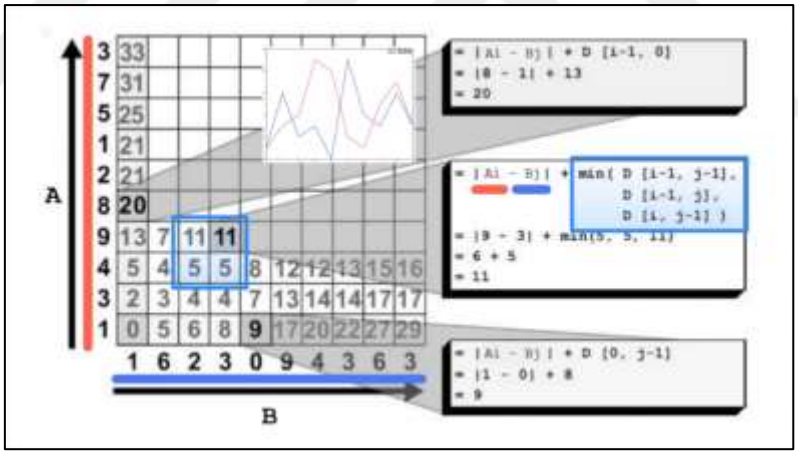

Figure 4.6: Sequential Correlation Search.

B. ALGORITHM PERFORMANCE

Calculation matrices are created from the record provided by the user and standards ,The example below shows how to get the numbers 20, 11 and 9.

The distances between the amplitudes of the two records are calculated. The corresponding values are taken from record A and record B. They are subtracted from each other, then the modulus is applied. Finally, the lowest value from the previous calculations is added. The value

can be selected from previously calculated values nearby: from the left, below, or diagonally left down.

In the example, the number 20 is obtained from 8 by subtracting 1 and adding 13, the only suitable previously calculated value (there are no previously obtained values on the left and diagonally). The resulting equality: |8-1|+13=20.

11 is obtained by subtracting 3 from 9 and adding the smallest suitable previous result (possible numbers are 11, 5 and 5; the smallest is chosen, ie 5). The resulting equality: |9-3|+5=11.

9 is obtained in an analogous way. 1 is subtracted from zero, the applied modulus. The only valid value to add is on the left and that is 8. There are no valid values below and diagonally. Resulting equality: |1-0|+8=9.

In this way, the entire matrix is obtained. When the full matrix is obtained, the shortest path from the upper right point of the matrix to the lower left point is searched (Figure 4.7). Starting from the upper right corner, the lowest value of the 3 adjacent ones is selected. This travels to the lower left corner. After all, the smallest possible distance between the amplitudes of the two records is found. The smallest distance corresponds to the largest correlation. This determines which record is most similar. If the similarity is high enough, the system considers the word spoken by the user to be recognized and a response is executed - a light of a certain color lights up, depending on which word was spoken by the user.

4.8 RESULTS

From the above discussion, it is evident that our Interaction Tool works as per our plan. But, due to time constraints, the user interface could not build completely. We had further plan to improve the interface by adding purchase options and cart system like online shopping websites. Then, we could have added a billing system to it. However, being an initial development, our tool is quite successful. For spectrum graphs, we typed "speech recognition (‘test.wav')" in the terminal window and pushed enter. Test.wav represents 2 and test2.wav represents 3. We test test2.wav. Figure 3 in figure 4.8 is the most accurate spectrum because test2.wav sounds 3.

Test.wav, number 2, was repeated. Figure 4.9 shows that graph 2 is more accurate because test.wav is two samples.

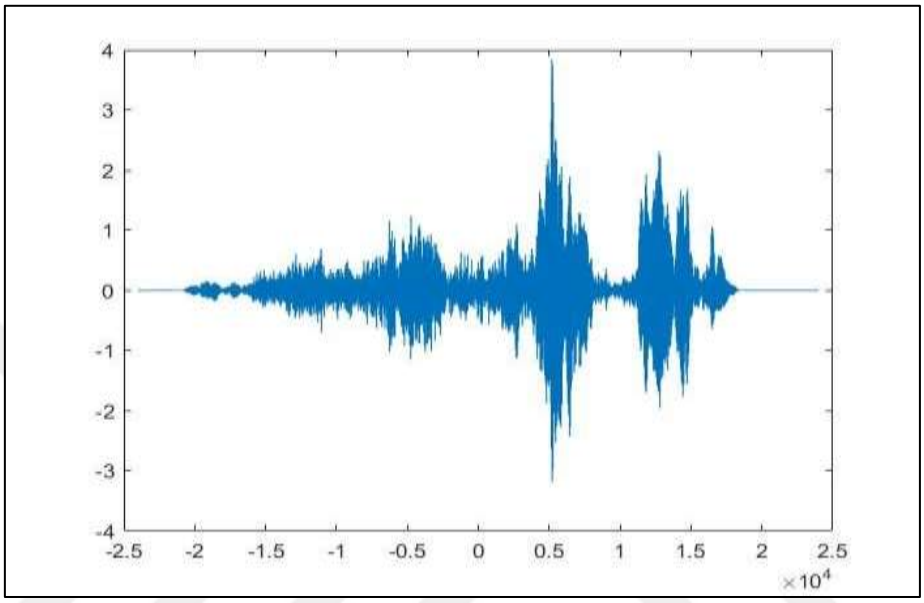

Figure 4-7: Spectrum of the Blair.wav input

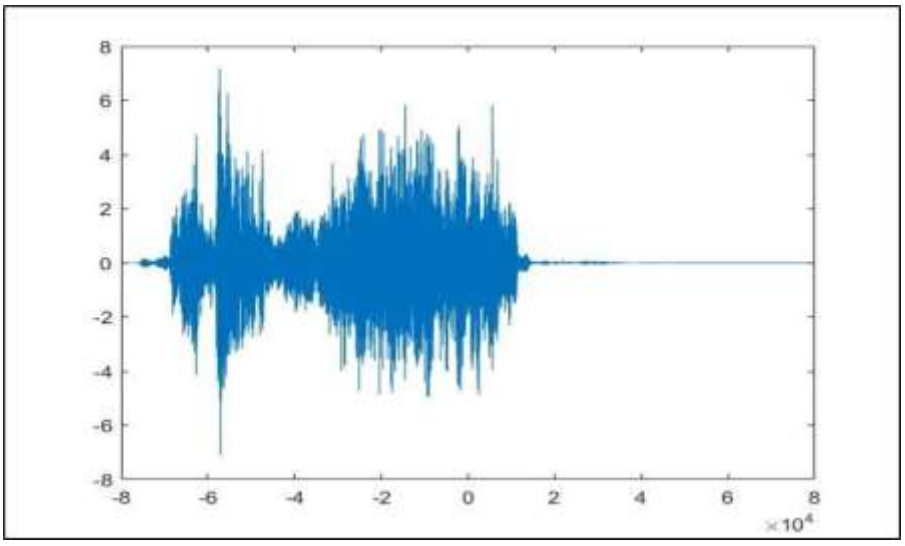

Figure 4.8: Spectrum of the Obama.Wav Input

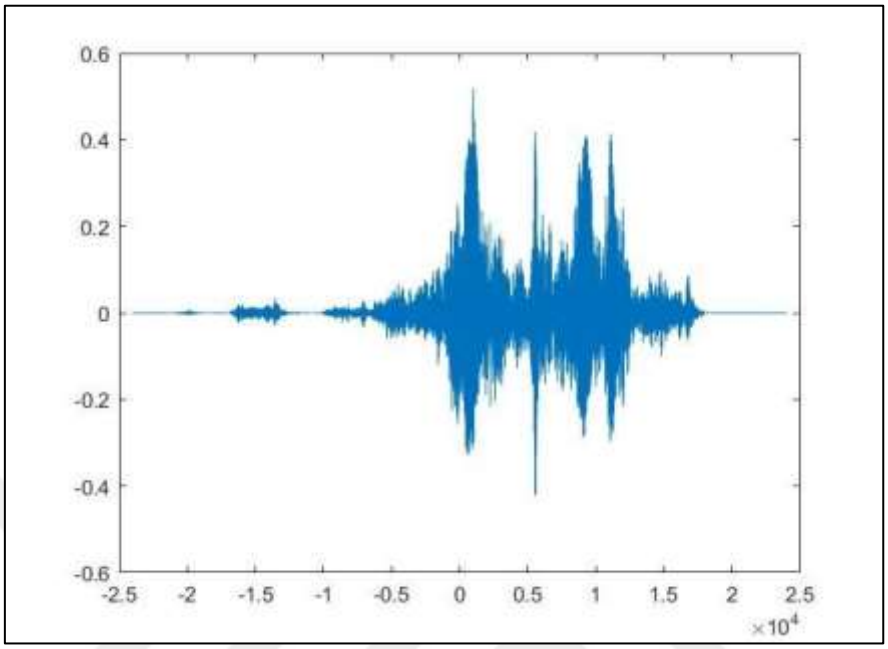

Figure 4.9: Spectrum of the Trump.wav input

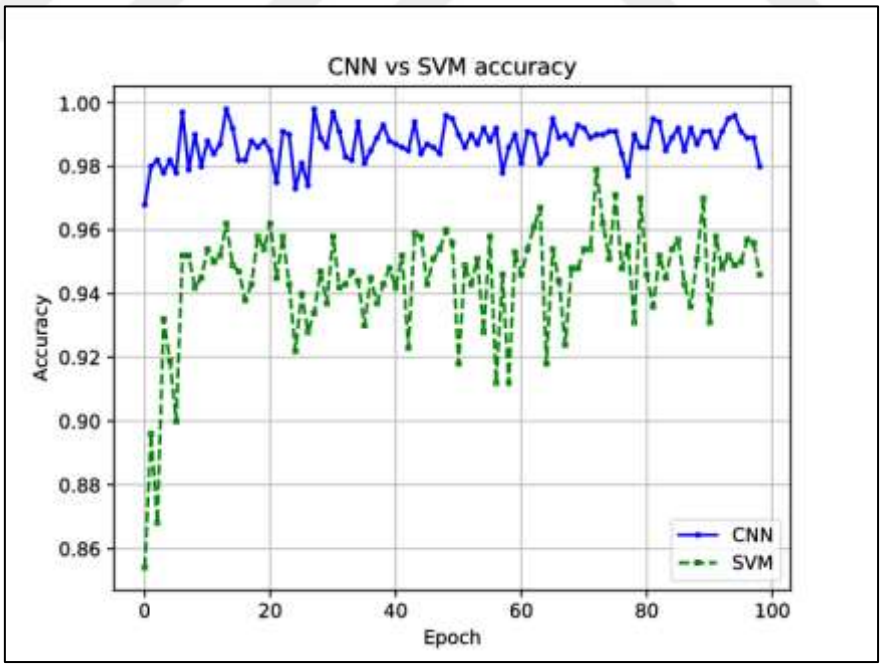

Figure 4.10: CNN and SVM accuracy compared (With MFCC)

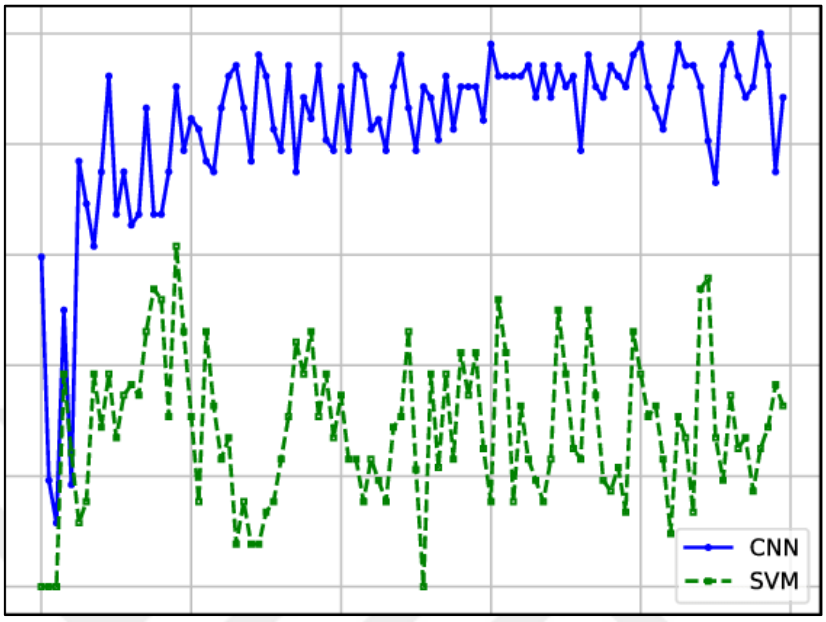

Figure 4.11: CNN and SVM accuracy compared without MFCC

4.9 ADVANTAGES OF THE PROPOSED SYSTEM

A. Using voice instructions rather than typing a message is a much more time-efficient method.
B. Voice activation and recognition technology is now under development for a wide variety of different reasons.
C. Decrease the costs of operations
D. Voice can communicate feelings.
E. It led to a decrease in both costs and errors.
F. This piece of technology eliminates inefficiencies and cuts down on the amount of time that is lost.
G. The use of speech recognition might make it possible to cut down on the amount of overtime that transcriptionists work and/or to avoid sending dictation work to companies outside the hospital.
H. It will save time and money for everyone involved, making the process more efficient overall.

4.10 DISADVANTAGES OF THE PROPOSED SYSTEM

a. It is challenging to construct a system that is flawless.
b. Speech is audible to others and encourages them to make noise.
c. The process of filtering out background noise is a challenging one, and there is simply too much of it for people to be able to handle on their own.
d. This biometric is susceptible to external factors such as ambient noise, which may affect its accuracy.
e. It doesn't retain privacy it does not suit a busy atmosphere.
f. An error was made, as well as a misunderstanding of what was said.

   It is quite evident that visually impaired people face some difficulties in public places where visual interaction is necessary. For this reason, a speech based interaction tool was very much necessary for them. But not much work has been done about it, largely due to lack of research in Bangla speech recognition. So, it can be said that our project may become the start of a trend about researching in Bangla speech recognition algorithms

5. CONCLUSIONS

Accent detection, speech-to-text conversion, and other applications of a similar nature are becoming an increasingly popular choice for future academic specializations in educational institutions and research centers. To develop an acoustic model of the Speech Recognition System employing MFCC and LPC characteristics for the DataSet, Support Vector Machines were put to use. This change was implemented so that it would be compatible with the language. Using a Multilayer Artificial Neural Network that was trained with this DataSet has shown some promise in terms of speech recognition. This DataSet was used to train the network. The integration of SVM techniques into the Azerbaijan Speech Recognition System is the primary goal of this thesis . While training, we conduct comparisons of the relative performance of a large number of SVMs that make use of a variety of Kernel functions. In terms of identification accuracy, Support Vector Machines that have been trained via radial basis functions and polynomial kernels surpass Multilayer Neural Networks. [Case in point:] [C]consider the example of recognizing a handwritten letter. Because of the availability of vast amounts of sensor data, cloud computing for the processing and training of deep neural networks, and the increasing sophistication of mobile and embedded technologies, the next generation of intelligent systems is poised to revolutionize both personal and business computing. This is since the next generation of intelligent systems is poised to revolutionize both personal and business computing. This is because cloud computing is readily available for the processing of deep neural networks as well as their training. In the first half of this thesis , we are going to investigate the beginnings of some of the most well-known deep learning models for high-level vision and voice systems, as well as the following evolution of these models. This thesis provides what we believe to be one of the most in-depth analyses of recent developments in the field of intelligent voice and vision applications. To achieve this goal, it is necessary to address both the software and hardware aspects of the problem. There are a number of these cutting-edge deep neural network technologies that have a tremendous amount of promise to boost the development of future vision and voice systems. This dissertation presents a hybrid model that identifies speakers based on their gender, accent, and keywords using a combination of Convolutional Neural Network (CNN) and Support Vector Machine (SVM) (SVM). The results

obtained by using the hybrid model are of a higher quality than those obtained by using either CNN or SVM on their own. It is common knowledge that training a hybrid model involves significantly more effort than training a regular CNN or SVM model would require. To categorize the characteristics that were extracted from a spectrogram image representation of speech using a support vector machine (SVM), a convolutional neural network (CNN) is first utilized. In comparison to the CNN model on its own, the CNN-SVM fusion model displays faster convergence while also reducing the amount of overfitting. According to the findings, the proposed system was able to successfully do a variety of tasks all at once while preserving an extremely high level of recognition accuracy. "One vs all" is the name of the multiclass classification approach that was applied in this piece of work. The training method makes use of the numerical input that is provided by seventy different individuals. Following this step, the LPC and MFCC algorithms are applied to each voice stream to extract speech features from those streams. Even though the lengths of individual audio signals change from utterance to utterance, the SVM is still applied to feature vectors that are of equal length. This is done to get the highest possible level of precision. For the purpose of synchronizing the volume levels of the various voices, Lagrange interpolation is necessary.

5.1 FUTURE WORK

The main contribution of this work consists in proposing a method for classifying the trend of assets using different algorithms. The research sought to reproduce results already present in the literature but applied to the Brazilian market. The results obtained show that the high hit rates found in the bibliography are probably not reproducible on a daily basis. This fact is due to the validation method used, which randomly separates training and testing. The SVM-ANN function proved to be able to circumvent the problem of financial predictions, reinforcing the statement in relation to its application in other contexts, in addition to image recognition.

REFERENCES

[1] Dong, Yanchao, et al. "Driver inattention monitoring system for intelligent vehicles: A review." , IEEE transactions on intelligent transportation systems vol.12, no.2, pp.596-614, 2010.

[2] McCall, Joel C., and Mohan M. Trivedi. "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation" , IEEE transactions on intelligent transportation systems vol.7 , no.1, pp.20-37 ,2006

[3] N. Buch, S. a. Velastin, and J. Orwell, "A Review of Computer Vision Techniques for the Analysis of Urban Traffic" , IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 3, pp. 920-939, 2011.

[4] E. Ohn-Bar and M. M. Trivedi, "Looking at Humans in the Age of Self-Driving and Highly Automated Vehicles" , IEEE Transactions on Intelligent Vehicles, vol. 1, no. 1, pp. 90-104, 2016.

[5] M. Bojarski et al., "End to End Learning for Self-Driving Cars," arXiv:1604, pp. 1-9, 2016.

[6] H. Woo et al., "Lane-Change Detection Based on Vehicle-Trajectory Prediction" , IEEE Robotics and Automation Letters, vol. 2, no. 2, pp. 1109-1116, 2017.

[7] W. Ouyang, X. Zeng, and X. Wang, "Single-pedestrian detection aided by two-pedestrian detection" , IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 9, pp. 1875-1889, 2015.

[8] W. Huang, G. Song, H. Hong, and K. Xie, "Deep Architecture for Traffic Flow Prediction: Deep Belief Networks With Multitask Learning," IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 5, pp. 2191-2201, 2014.

[9] X. Wang, R. Jiang, L. Li, Y. Lin, X. Zheng, and F.-Y. Wang" ,Capturing Car-Following Behaviors by Deep Learning," IEEE Transactions on Intelligent Transportation Systems, pp. 1-11, 2017.

[10] A. Ferdowsi, U. Challita, and W. Saad ,"Deep Learning for Reliable Mobile Edge Analytics in Intelligent Transportation Systems: An Overview" , IEEE vehicular technology magazine, vol. 14, no. 1, pp. 62-70, 2019.

[11] M. Havaei et al., "Brain tumor segmentation with Deep Neural Networks," Medical Image Analysis, vol. 35, pp. 18-31, 2017.

[12] S. Liu et al., "Multimodal Neuroimaging Feature Learning for Multiclass Diagnosis of Alzheimer's Disease" ,IEEE Transactions on Biomedical Engineering, vol. 62, no. 4, pp. 1132-1140, 2015.

[13] E. Putin et al., "Deep biomarkers of human aging: Application of deep neural networks to biomarker development" ,Aging, vol. 8, no. 5, pp. 1021-1033, 2016.

[14] Deo, Rahul C., et al. "An end-to-end computer vision pipeline for automated cardiac function assessment by echocardiography" , CoRR (2017).

[15] M. R. Alam, M. B. I. Reaz, and M. A. M. Ali, "A review of smart homes—Past, present, and future" , IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 6, pp. 1190-1203, 2012.

[16] R. S. Cooper, J. F. McElroy, W. Rolandi, D. Sanders, R. M. Ulmer, and E. Peebles, "Personal virtual assistant," ed: Google Patents, 2011.

[17] E. W. Ngai, L. Xiu, and D. C. Chau, "Application of data mining techniques in customer relationship management: A literature review and classification," Expert systems with applications, vol. 36, no. 2, pp. 2592-2602, 2009.

[18] S. Goswami, S. Chakraborty, S. Ghosh, A. Chakrabarti, and B. Chakraborty, "A review on application of data mining techniques to combat natural disasters," Ain Shams Engineering Journal, pp. 1-14, 2016.

[19] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," Artificial Intelligence Review, vol. 43, no. 1, pp. 1-54, 2015.

[20] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1653-1660, 2014.

[21] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in Advances in neural information processing systems, vol. 7, pp. 1799-1807, 2014

[22] S. Srivastava, A. Bisht, and N. Narayan, "Safety and security in smart cities using artificial intelligence—A review," in Cloud Computing, Data Science & Engineering-Confluence, 2017 7th International Conference on, IEEE, pp. 130-133. 2017

[23] Y. Yang, C. Fairbairn, and J. F. Cohn, "Detecting depression severity from vocal prosody," IEEE Transactions on Affective Computing, vol. 4, no. 2, pp. 142-150, 2013.

[24] L. D. Shriberg, R. Paul, J. L. McSweeny, A. Klin, D. J. Cohen, and F. R. Volkmar, "Speech and prosody characteristics of adolescents and adults with high-functioning autism and Asperger syndrome," Journal of Speech, Language, and Hearing Research, vol. 44, no. 5, pp. 1097-1115, 2001.

[25] M. El Ayadi, M. S. Kamel, and F. Karray, "Survey on speech emotion recognition: Features, classification schemes, and databases," Pattern Recognition, vol. 44, no. 3, pp. 572-587, 2011.

[26] H. M. Fayek, M. Lech, and L. Cavedon, "Evaluating deep learning architectures for Speech Emotion Recognition," Neural Networks, vol. 92, pp. 60-68, 2017.

[27] Y. Kim, H. Lee, and E. M. Provost, "Deep learning for robust feature generation in audiovisual emotion recognition,", pp. 36873691, doi: 10.1109/ICASSP.2013.

[28] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," Neural computation, vol. 18, no. 7, pp. 1527-1554, 2006.

[29] G. E. Hinton, "Learning multiple layers of representation," Trends in cognitive sciences, vol. 11, no. 10, pp. 428-434, 2007.

[30] R. M. Cichy, A. Khosla, D. Pantazis, A. Torralba, and A. Oliva, "Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence," Scientific reports, vol. 6 , no. 27755, pp. 1-13, 2016.

[31] N. Kruger et al., "Deep hierarchies in the primate visual cortex: What can we learn for computer vision?," IEEE transactions on pattern analysis and machine intelligence, vol. 35, no. 8, pp. 1847-1871, 2013.

[32] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural networks, vol. 61, pp. 85-117, 2015.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks" , in Advances in neural information processing systems. pp. 1097-1105, 2012.

[34] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," Journal of Machine Learning Research, vol. 11, no. Dec, pp. 3371-3408, 2010.

[35] I. Goodfellow et al., "Generative adversarial nets," in Advances in neural information processing systems, pp. 2672-2680,2014.

[36] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," arXiv preprint arXiv:1312.6114, 2013.

[37] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," arXiv preprint arXiv:1605.08803, 2016.

[38] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," arXiv preprint arXiv:1506.00019, pp. 1-38, 2015.

[39] A. Vaswani et al., "Attention is all you need," in Advances in neural information processing systems, pp. 5998-6008, 2017.

[40] M. Alam, L. Vidyaratne, and K. M. Iftekharuddin, "Novel hierarchical Cellular Simultaneous Recurrent neural Network for object detection," in Neural Networks (IJCNN), 2015 International Joint Conference on, pp. 1-7, 2015.

[41] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," in Proceedings of the 24th international conference on Machine learning, ACM, pp. 791-798. , 2007.

[42] R. Salakhutdinov and G. Hinton, "Deep boltzmann machines," in Artificial Intelligence and Statistics, pp. 448-455,2009

[43] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, pp. 3377-3381, 2013.

[44] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in Proceedings of the 25th international conference on Machine learning, ACM, pp. 1096-1103, 2008.

[45] G. B. Huang, H. Lee, and E. Learned-Miller, "Learning hierarchical representations for face verification with convolutional deep belief networks," in Computer Vision and Pattern Recognition (CVPR),IEEE Conference on, IEEE, pp. 2518-2525,2015.

[46] Z. You, X. Wang, and B. Xu, "Investigation of deep boltzmann machines for phone recognition," in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, pp. 7600-7603, 2013.

[47] G. Wen, H. Li, J. Huang, D. Li, and E. Xun, "Random Deep Belief Networks for Recognizing Emotions from Speech Signals," Computational intelligence and neuroscience, vol. 2017, pp. 1-9, 2017.

[48] C. Huang, W. Gong, W. Fu, and D. Feng, "A research of speech emotion recognition based on deep belief network and SVM," Mathematical Problems in Engineering, vol. 2014, pp. 1-7, 2014.

[49] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in Proceedings of the 26th annual international conference on machine learning, ACM, pp. 609-616, 2009.

[50] X. Yan, J. Yang, K. Sohn, and H. Lee, "Attribute2image: Conditional image generation from visual attributes," in European Conference on Computer Vision, 2016: Springer, pp. 776-791, 2016.

[51] J. Walker, C. Doersch, A. Gupta, and M. Hebert, "An uncertain future: Forecasting from static images using variational autoencoders," in European Conference on Computer Vision, Springer, pp. 835-851, 2016

[52] S. Semeniuta, A. Severyn, and E. Barth, "A hybrid convolutional variational autoencoder for text generation," arXiv preprint arXiv:1702.02390, 2017.

[53] K. Akuzawa, Y. Iwasawa, and Y. Matsuo, "Expressive speech synthesis via modeling expressions with variational autoencoder," arXiv preprint arXiv:1804.02135, 2018.

[54] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," arXiv preprint arXiv:1605.05396, 2016.

[55] C. Ledig et al., "Photo-realistic single image super-resolution using a generative adversarial network," arXiv preprint, 2017.

[56] M. Mirza and S. Osindero, "Conditional generative adversarial nets," arXiv preprint arXiv:1411.1784, 2014.

[57] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans" , in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8798-8807, 2018.

[58] [58] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning" , arXiv preprint arXiv:1605.09782, 2016.

[59] J. Donahue and K. Simonyan, "Large scale adversarial representation learning," in Advances in Neural Information Processing Systems, pp. 10541-10551,2019.

[60] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks" , arXiv preprint arXiv:1701.04862, 2017.

[61] I. Goodfellow, "NIPS 2016 tutorial: Generative adversarial networks," arXiv preprint arXiv:1701.00160, 2016.

[62] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," arXiv preprint arXiv:1802.05957, 2018.

[63] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in International conference on machine learning, pp. 214-223, 2017.

[64] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in Advances in neural information processing systems, pp. 5767-5777, 2017.

[65] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in Proceedings of the IEEE International Conference on Computer Vision, pp. 2794-2802, 2017.

[66] A. Razavi, A. v. d. Oord, and O. Vinyals, "Generating Diverse High-Fidelity Images with VQ-VAE-2," arXiv preprint arXiv:1906.00446, 2019.

[67] L. Dinh, D. Krueger, and Y. Bengio, "Nice: Non-linear independent components estimation," arXiv preprint arXiv:1410.8516, 2014.

[68] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in Advances in Neural Information Processing Systems, pp. 10215-10224, 2018.

[69] A. v. d. Oord et al., "Wavenet: A generative model for raw audio," arXiv preprint arXiv:1609.03499, 2016.

[70] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," arXiv preprint arXiv:1601.06759, 2016.

[71] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, pp. 3617-3621,2019.

[72] S. Pascual, A. Bonafonte, and J. Serra, "SEGAN: Speech enhancement generative adversarial network," arXiv preprint arXiv:1703.09452, 2017.

[73] N. Adiga, Y. Pantazis, V. Tsiaras, and Y. Stylianou, "Speech Enhancement for Noise-Robust Speech Synthesis Using Wasserstein GAN}}," Proc. Interspeech 2019, pp. 1821-1825, 2019.

[74] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional lstm-cnns-crf," arXiv preprint arXiv:1603.01354, 2016.

[75] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," arXiv preprint arXiv:1409.1259, 2014.

[76] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," IEEE transactions on neural networks and learning systems, vol. 28, no. 10, pp. 2222-2232, 2016.

[77] V. Mnih, N. Heess, and A. Graves, "Recurrent models of visual attention," in Advances in neural information processing systems, pp. 2204-2212. 2014.

[78] H. Larochelle and G. E. Hinton, "Learning to combine foveal glimpses with a third-order Boltzmann machine," in Advances in neural information processing systems, pp. 1243-1251, 2010

[79] M. A. Ranzato, "On learning where to look," arXiv preprint arXiv:1405.5488, 2014.

[80] M. Denil, L. Bazzani, H. Larochelle, and N. de Freitas, "Learning where to attend with deep architectures for image tracking," Neural computation, vol. 24, no. 8, pp. 2151-2184, 2012.

[81] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, "Draw: A recurrent neural network for image generation," arXiv preprint arXiv:1502.04623, 2015.

[82] K. Fu, J. Jin, R. Cui, F. Sha, and C. Zhang, "Aligning Where to See and What to Tell: Image Captioning with Region-Based Attention and Scene-Specific Contexts," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2321-2334, 2017.

[83] E. Mansimov, E. Parisotto, J. L. Ba, and R. Salakhutdinov, "Generating images from captions with attention," arXiv preprint arXiv:1511.02793, 2015.

[84] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," arXiv preprint arXiv:1410.5401, pp. 1-26, 2014.

[85] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," arXiv preprint arXiv:1508.04025, pp. 1-11, 2015.

[86] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on, IEEE, pp. 49604964, 2016.

[87] J. Liu, A. Shahroudy, D. Xu, A. C. Kot, and G. Wang, "Skeleton-based action recognition using spatio-temporal LSTM network with trust gates," IEEE transactions on pattern analysis and machine intelligence, vol. 40, no. 12, pp. 3007-3021, 2018.

[88] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," arXiv preprint arXiv:1805.08318, 2018.